Kuliah 1 — Pengantar IMK: Mengapa Antarmuka Penting

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 1

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita berkenalan dengan disiplin ini, melihat mengapa ia penting di Indonesia pada 2026, menyiapkan meja kerja, dan mengubah perdebatan desain menjadi angka. Buku: Bab 1, Subbab 1.1–1.16.

Isi Pertemuan Ini

1. **Mengapa antarmuka penting** — halaman warung Bu Ratna, dan cerita yang sama di puskesmas, ATM, pangkalan ojek, dan laundry.
2. **Apa itu IMK** — *usability*, pengalaman pengguna, dan aksesibilitas menurut ISO; dua jurang Norman.
3. **IMK pada 2026** — bahasa desain baru, agen AI, CHI 2026, serta pengguna dan hukum Indonesia di balik layar.
4. **Meja kerja** — JDK 25, Python 3.14, dan VS Code di Windows, macOS, dan Linux.
5. **Program IMK pertama** — *Keystroke-Level Model* memprediksi berapa detik yang dihemat layar yang lebih baik.

Setiap program hari ini ada dalam Java dan Python, dan setiap keluaran yang ditampilkan adalah keluaran yang benar-benar dicetak program. Buku: Bab 1.

Titik Awal: Warung Bu Ratna Masuk Daring

Dimas membuatkan bibinya halaman pesanan: *dropdown* 42 menu, kotak jumlah, *Add*, dan *Save*.

Setiap pesanan tersimpan dengan benar, tetapi pada Sabtu pertama ia kembali ke buku catatannya pukul sebelas.

Satu pesanan berarti: buka *dropdown*, gulir, klik, klik kotak, ketik, pindahkan tangan kembali, klik *Add*, ulangi.

Tidak ada yang rusak kecuali kecocokan antara halaman dan orang yang memakainya di bawah tekanan. Buku: Subbab 1.1.

Yang Berubah adalah Antarmukanya

Dimas mengamati selama satu jam lalu menggambar ulang halaman itu sebagai **ubin besar**, menu tersering lebih dulu; satu klik menambah satu porsi.

Pesanan yang sama turun dari lebih dari dua puluh detik menjadi sekitar sepuluh, dan Bu Ratna memakainya tanpa disuruh.

Basis data, jaringan, dan bahasanya tidak berubah. Hanya **antarmukanya** yang berubah.

Antarmuka adalah bagian sistem yang dilihat, didengar, dan disentuh orang, dan melaluinya orang memberi tahu sistem apa yang harus dilakukan. Buku: Subbab 1.1.

Cerita yang Sama, di Mana-Mana

Kios puskesmas dengan dua belas label kecil, dan pasien lansia yang mencari poli gigi.

ATM yang dulu menelan kartu setiap orang yang pergi membawa uangnya.

Aplikasi ojek yang titik jemputnya jatuh di seberang sungai; menu WhatsApp **laundry** yang nomornya tidak diingat siapa pun.

Buku ini mengajarkan cara melihat masalah semacam itu sebelum pengguna mengalaminya, cara merancang lebih baik, dan cara membuktikannya. Buku: Subbab 1.1.

Apa Itu Interaksi Manusia dan Komputer

ACM SIGCHI: *disiplin yang berkaitan dengan perancangan, evaluasi, dan implementasi sistem komputasi interaktif untuk digunakan manusia.*

Perancangan sebelum membangun, **evaluasi** dengan orang dan tugas sungguhan, **implementasi** dalam kode, keberhasilan diukur dari **penggunaan manusia**.

IMK menimba dari ilmu komputer, psikologi, desain, sosiologi, ergonomi, dan linguistik.

Dalam bahasa Inggris bidang ini disebut Human-Computer Interaction (HCI). Buku: Subbab 1.2.1.

Tiga Kata dengan Makna yang Tepat

Usability (ISO 9241-11:2018): efektivitas, efisiensi, dan kepuasan bagi pengguna, tujuan, dan konteks penggunaan *tertentu*.

Pengalaman pengguna (ISO 9241-210:2019): persepsi dan tanggapan sebelum, selama, dan sesudah penggunaan.

Aksesibilitas: dapat dipakai orang dengan rentang kebutuhan dan kemampuan seluas-luasnya, termasuk penyandang disabilitas.

Standar mengubah pujian samar (“ramah pengguna”) menjadi hal yang dapat diperiksa. Buku: Subbab 1.2.1.

“Tertentu”: Mudah Dipakai bagi Siapa?

Halaman pertama Dimas sangat mudah dipakai oleh Dimas di mejanya.

Halaman itu gagal bagi pengguna tertentu (Bu Ratna, lambat memakai mouse), tujuan tertentu (pesanan tiga menu), dan konteks tertentu (dua puluh mahasiswa lapar).

Setiap metode dalam perkuliahan ini dimulai dengan bertanya **siapa**, **untuk apa**, dan **di mana**.

Ubah pengguna, tujuan, atau konteksnya, dan penilaiannya dapat berubah. Buku: Subbab 1.2.1.

Mengukur Tiga Mutu

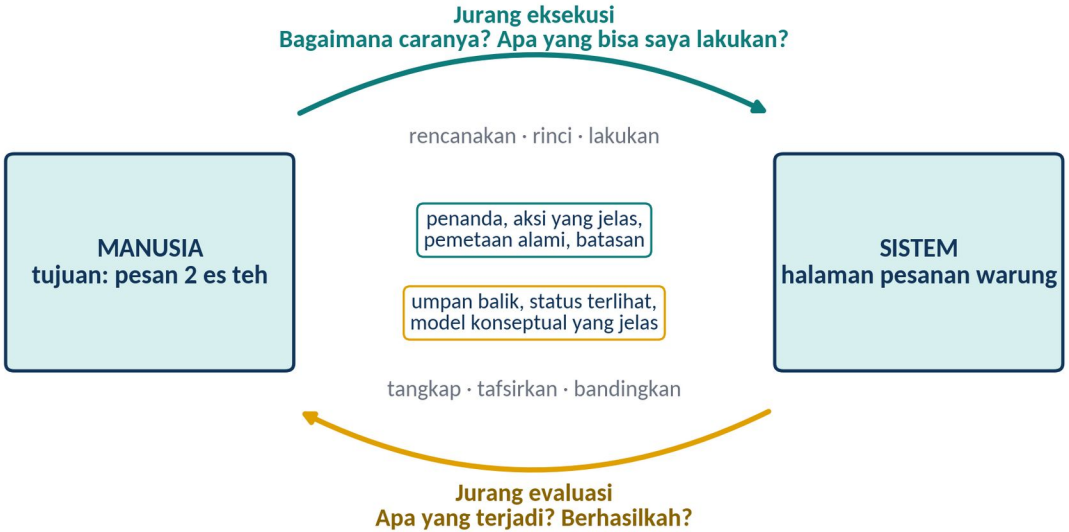
Efektivitas: persentase pesanan yang dimasukkan tanpa kesalahan.

Efisiensi: detik per pesanan.

Kepuasan: kuesioner singkat seperti *System Usability Scale* (Bab 7).

Pengalaman pengguna lebih luas lagi: kecemasan Bu Ratna pada Jumat malam dan rasa senangnya ketika antrean kosong. Buku: Subbab 1.2.1.

Dua Jurang Norman



Gambar 1.1 Dua jurang Norman: eksekusi (bagaimana caranya?) dan evaluasi (berhasilkah?), serta mutu desain yang menjembatani masing-masing.

Gambar 1.1: jurang eksekusi memisahkan tujuan dari tindakan yang ditawarkan sistem; jurang evaluasi memisahkan apa yang ditampilkan sistem dari apa yang perlu diketahui orang. Buku: Subbab 1.2.2.

Menjembatani Kedua Jurang

Eksekusi: penanda yang menunjukkan apa yang bisa dilakukan, pemetaan alami, batasan yang mencegah tindakan keliru.

Evaluasi: umpan balik segera, status yang terlihat, model konseptual yang jelas.

Bu Ratna mengeklik *Add*, tidak melihat perubahan, mengeklik lagi: dapur membuat empat es teh.

Bank menjembatani jurang evaluasi ATM dengan mengembalikan kartu sebelum uang keluar. Buku: Subbab 1.2.2.

Salahkan Jurangnya, Bukan Penggunanya

Ketika seseorang berbuat salah, tanyakan **jurang mana yang terlalu lebar?**, bukan apa yang salah dengan orangnya.

Menyalahkan pengguna tidak mengubah apa pun; mempersempit jurang mengubah perilaku setiap pengguna berikutnya.

Satu kebiasaan ini membedakan IMK dari "nanti juga terbiasa". Buku: Subbab 1.2.2.

Apa yang Bukan IMK

Bukan **membuat tampilan cantik**: halaman indah yang menyembunyikan aksi utamanya tetap gagal.

Bukan **menanyakan kepada pengguna apa yang mereka mau**: orang buruk meramalkan perilakunya sendiri; amatilah mereka (Bab 5).

Bukan **akal sehat**: yang jelas bagi pembuatnya justru yang tidak lagi terlihat olehnya.

Disiplin ini mengganti titik buta pembuat dengan bukti. Buku: Subbab 1.2.3.

Siapa Penggunanya, dari Era ke Era

Tabel 1.1 Era-era interaksi manusia dan komputer dan apa yang dituntut masing-masing dari penggunanya.

Era	Antarmuka yang khas	Yang harus dipelajari pengguna
1950-an–1970-an	Pekerjaan *batch*, baris perintah	Sintaksis perintah yang persis
1980-an–1990-an	Jendela, ikon, menu, penunjuk	Metafora desktop
1990-an–2000-an	Web	Tautan, formulir, pencarian
2007–2020-an	Layar sentuh, aplikasi seluler	Gestur, konvensi aplikasi
2020-an	Suara, AI generatif, agen, spasial	Apa yang ditanyakan, kapan percaya

Tabel 1.1: sejarah IMK adalah sejarah tentang siapa penggunanya, dari spesialis terlatih hingga hampir semua orang. Buku: Subbab 1.3.

2025–2026: Bahasa Desain Diperbarui

Liquid Glass (Apple, 9 Juni 2025): material tembus pandang di iOS 26 dan saudara-saudaranya.

Material 3 Expressive (Google, 2025): 46 studi, lebih dari 18.000 partisipan; elemen kunci ditemukan hingga empat kali lebih cepat.

Fluent 2: *design system* bersama Microsoft untuk Windows dan Microsoft 365.

Semuanya penerapan prinsip, bukan mode untuk ditiru; Bab 3 menunjukkan di mana mereka menerapkan dan di mana meregangkan prinsip. Buku: Subbab 1.3.

Melampaui Layar Datar, dan AI sebagai Antarmuka

Apple Vision Pro (Februari 2024) menjadikan **komputasi spasial** istilah umum; jam tangan, *earbud*, dan mobil punya antarmukanya sendiri.

ChatGPT melaporkan **900 juta pengguna mingguan** pada Februari 2026; kopilot dan **agen** bertindak atas nama pengguna.

Jurangnya kembali: *apa yang harus saya tanyakan?* dan *benarkah ini, dan apa yang baru dilakukan agen itu?*

Pedoman interaksi manusia-AI dari Amershi dkk. (CHI 2019) tetap menjadi titik awal; Bab 13 mengembangkannya. Buku: Subbab 1.3.

Komunitas Riset Lebih Besar daripada Sebelumnya

CHI 2026, Barcelona, 13–17 April 2026: 1.702 makalah penuh diterima dari 6.730 kiriman (25,3%), 5.254 peserta di lokasi.

Figma Config 2026 (23–24 Juni): agen AI di dalam kanvas, *code layers*, desain gerak.

Alat berubah cepat; perkuliahan ini memakainya untuk prototipe (Bab 6) tetapi tidak pernah membiarkannya menggantikan pengamatan terhadap orang sungguhan. Buku: Subbab 1.3.

Siapa yang Daring di Indonesia

APJII 2026: 235,2 juta pengguna, **81,72%** dari 287,9 juta jiwa; dari 85,95% di Jawa hingga 69,74% di Maluku dan Papua. Dari yang masih luring, 34% tidak punya perangkat dan **31,5% kurang literasi digital**.

Angka kedua adalah angka IMK: sepertiga orang yang belum terhubung tertahan oleh antarmuka yang menuntut terlalu banyak dari pengguna pemula. Buku: Subbab 1.4.1.

Dengan Perangkat Apa? Tanyakan Apa yang Dihitung

StatCounter, Agustus 2026: Android **77,07%**, iOS 22,88% dari tayangan halaman seluler di Indonesia.

Layanan yang sama: 60,59% dari seluruh tayangan halaman berasal dari desktop, 38,98% dari ponsel.

Yang dihitung adalah **tayangan halaman di situs yang terlacak**, bukan orang; sebagian besar pemakaian sehari-hari terjadi di dalam aplikasi.

Selalu baca metodenya sebelum merancang berdasarkan sebuah angka. Buku: Subbab 1.4.1.

Penggunanya Makin Tua

BPS (SUPAS 2025): **11,97%** penduduk berusia enam puluh tahun ke atas; Indonesia memasuki fase penuaan penduduk.

Dari 17,83% di DI Yogyakarta hingga 6,71% di Papua Tengah.

Layar yang dirancang anak dua puluhan untuk anak dua puluhan gagal melayani bagian negeri yang terus bertambah (Bab 2).

Lansia termasuk pengguna paling sering layanan publik seperti puskesmas. Buku: Subbab 1.4.1.

Hukum Kini Menuntut Antarmuka yang Baik

UU 8/2016 tentang Penyandang Disabilitas: hak atas informasi dan komunikasi yang aksesibel.

PP 42/2020: aksesibilitas pelayanan publik bagi penyandang disabilitas.

UU PDP 27/2022: persetujuan eksplisit yang sah untuk tujuan tertentu; **PP 33/2026**, ditandatangani 16 Juli 2026, berlaku 16 Januari 2027.

Layar persetujuan yang tidak dapat dipahami adalah desain yang buruk dan dasar hukum yang lemah (Bab 12). Buku: Subbab 1.4.2.

Di Luar Negeri, dan Tolok Ukur Teknisnya

European Accessibility Act berlaku sejak 28 Juni 2025 untuk perdagangan elektronik, perbankan, buku elektronik, dan tiket yang dijual di Uni Eropa.

WCAG 2.2 adalah *Recommendation* W3C yang berlaku; **WCAG 3.0** masih *working draft* (draf tinjauan terakhir September 2026), bertahun-tahun dari rampung.

Bab 11 membahas WCAG 2.2 kriteria demi kriteria. Buku: Subbab 1.4.2.

Cara Kerja Buku Ini

Situasi sehari-hari → gagasan dengan kata sederhana → teori dengan bukti → metode praktis → program pendamping dalam Java dan Python.

Lalu *IMK dalam Praktik: Profitina*, kesalahan yang sering terjadi, ringkasan, soal latihan dengan penyelesaian lengkap, daftar pustaka.

Program membuat IMK terukur: KLM (Bab 1), Fitts dan Hick (Bab 2), laporan heuristik (Bab 3), papan afinitas (Bab 4).

Programnya sengaja kecil; ia mengukur desain, bukan membangun sistem. Buku: Subbab 1.4.3.

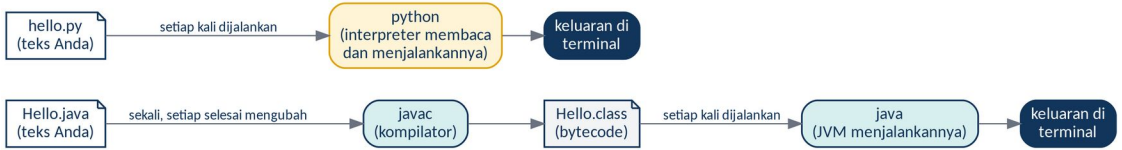
Meja Kerja Anda

Tabel 1.2 Alat-alat meja kerja dan versi yang dipakai di seluruh buku.

Alat	Versi yang dipakai
JDK (Temurin)	25 LTS, pembaruan 25.0.4
Python	3.14.7
Visual Studio Code	1.139
Extension Pack for Java, ekstensi Python	dari tampilan Extensions

Tabel 1.2: setiap program juga berjalan di JDK 17+ dan Python 3.11+, sehingga pemasangan lama tetap dapat dipakai. Buku: Subbab 1.5.

Bagaimana Java dan Python Menjalankan Program



Gambar 1.2 Bagaimana program Java dan program Python berjalan: javac mengompilasi sekali menjadi bytecode untuk JVM; Python membaca dan menjalankan berkasnya setiap kali.

Gambar 1.2: Java menemukan banyak kesalahan saat kompilasi; Python menemukannya ketika sampai di baris yang keliru. `java Hello.java` *menjalankan satu berkas dalam satu langkah. Buku: Subbab 1.5.*

Memasang di Windows

```
PS> winget install EclipseAdoptium.Temurin.25.JDK
PS> py install 3.14
PS> winget install Microsoft.VisualStudioCode
```

Buka PowerShell **baru** setelah memasang; periksa `java -version`, `javac -version`, dan `python --version`.

“not recognized” berarti foldernya tidak ada di `PATH`: jalankan ulang pemasang dengan **Add to PATH**.

Python di Windows kini dipasang melalui Python install manager. Buku: Subbab 1.6.

Memasang di macOS dan Linux

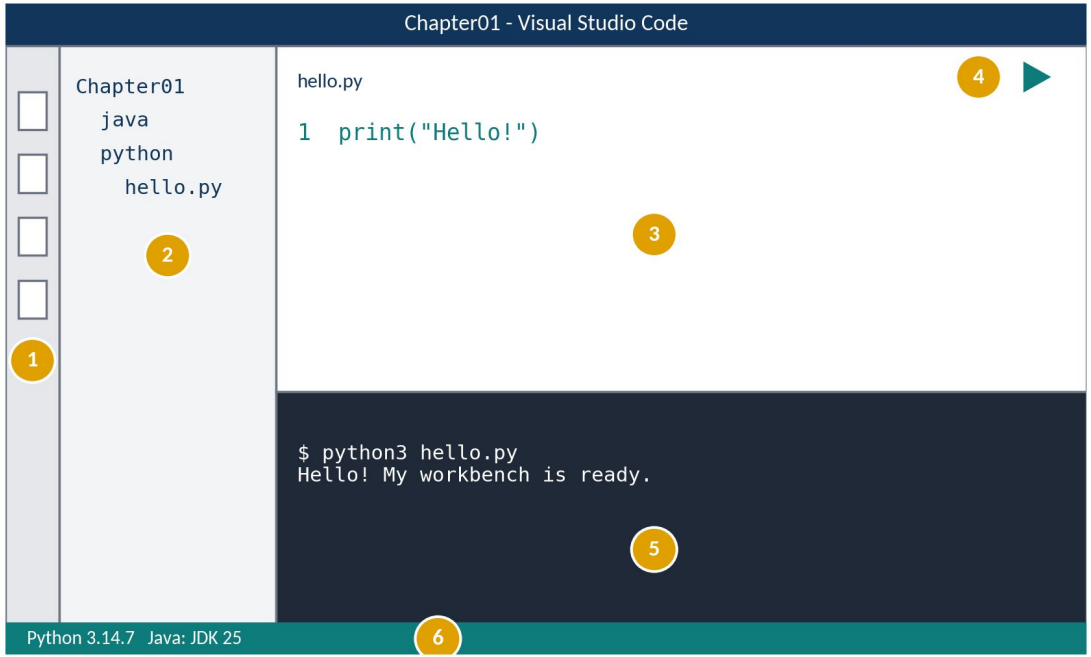
```
you@mac ~ % brew install --cask temurin@25
$ sudo apt install openjdk-25-jdk python3 python3-venv python3-pip
```

macOS: pemasang Python dari python.org, lalu **Install Certificates.command** sekali; pakai `python3`.

Ubuntu 26.04 LTS membawa OpenJDK 25 dan Python 3.14; jangan pernah menghapus Python sistem di Linux.

Fedora: `sudo dnf install java-25-openjdk-devel python3 python3-pip`. *Buku: Subbab 1.7 dan 1.8.*

Tur VS Code



Gambar 1.3 Bagian-bagian Visual Studio Code yang dipakai di buku ini: (1) Activity Bar, (2) Explorer, (3) editor, (4) tombol Run, (5) terminal terpadu, (6) Status Bar.

Gambar 1.3: pasang Extension Pack for Java dan ekstensi Python, dan selalu buka folder, bukan satu berkas. Buku: Subbab 1.9.

Program Pertama, dan Pemeriksaan

```
$ python3 check_setup.py
```

```
Python version : 3.14.7
System         : Linux (x86_64)
Virtual env.   : no (fine for this book)
Java version   : 25.0.4
Result         : your workbench is ready
```

`Hello.java` dan `hello.py` menyapa dan mencetak versinya; `CheckSetup.java` memeriksa dari sisi Java. Baris terakhirlah yang dibaca. Buku: Subbab 1.10.

Keystroke-Level Model

Tabel 1.3 Operator Keystroke-Level Model dan waktu satuan bakunya.

Operator	Arti	Waktu (s)
K	Menekan tombol	0.28
P	Menunjuk dengan mouse	1.10
B	Menekan atau melepas tombol mouse	0.10
H	Memindahkan tangan antara keyboard dan mouse	0.40
M	Persiapan mental	1.35

Tabel 1.3: Card, Moran, dan Newell (1980) memprediksi waktu seorang ahli sebelum apa pun dibangun; satu M ditaruh sebelum setiap bongkahan yang memerlukan keputusan. Buku: Subbab 1.11.1.

Hitung Langkahnya, Jumlahkan Waktunya

$$T = n_K t_K + n_P t_P + n_B t_B + n_H t_H + n_M t_M \tag{1.1}$$

Hitung setiap jenis operator, kalikan dengan waktu satuannya, lalu jumlahkan.

Persamaan (1.1) adalah seluruh modelnya; seninya terletak pada menuliskan tugas sebagai operator. Buku: Subbab 1.11.1.

Desain A dengan Tangan: Formulir Dropdown

Per menu: M, P B B (buka daftar), P B B (pilih), P B B (kotak jumlah), H, K, H, P B B (*Add*).

Tiga menu dan *Save*: $n_M = 4$, $n_P = 13$, $n_B = 26$, $n_H = 6$, $n_K = 3$.

$T_A = 5.40 + 14.30 + 2.60 + 2.40 + 0.84 = 25.54$ s.

Menunjuk saja memakan 14,30 detik dari 25,54. Buku: Subbab 1.11.2.

Desain B dengan Tangan: Ubin Menu

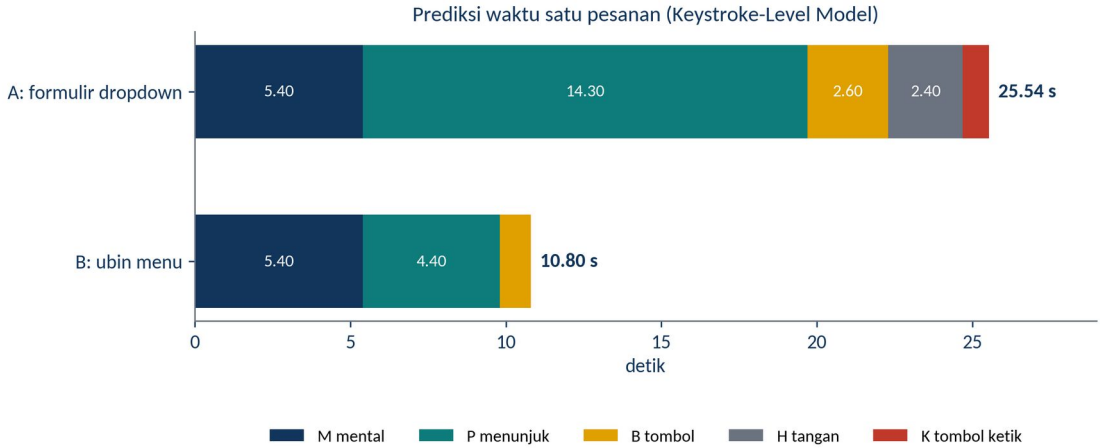
Es teh: M P B B B B; nasi goreng: M P B B; ayam geprek: M P B B; *Save*: M P B B.

$T_B = 4(1.35) + 4(1.10) + 10(0.10) = 10.80$ s.

Penghematan: $25.54 - 10.80 = 14.74$ s, atau 57,7%, pada setiap pesanan seperti itu.

Keempat operator M sama pada kedua desain: proses berpikirnya tidak berubah, menunjuknya yang berubah. Buku: Subbab 1.11.2.

Ke Mana Waktunya Pergi



Gambar 1.4 Prediksi waktu satu pesanan pada kedua halaman, dipecah per operator: menunjuk (P) mendominasi desain A.

Gambar 1.4 membuat argumennya terlihat oleh orang yang tidak membaca persamaan, seperti Bu Ratna. Buku: Subbab 1.11.2.

Program Membaca Desain sebagai Teks

Kode 1.1 `data/orders.txt`: kedua desain yang ditulis sebagai operator Keystroke-Level Model.

```
# Pesanan "2 es teh, 1 nasi goreng, 1 ayam geprek" pada dua desain.
# K tombol ketik, P menunjuk, B tombol mouse ditekan atau dilepas, H tangan
# pindah antara keyboard dan mouse, M persiapan mental. "2B" berarti "B B".
A: dropdown form
  M P 2B P 2B P 2B H K H P 2B     # es teh: daftar menu, jumlah, Add
  M P 2B P 2B P 2B H K H P 2B     # nasi goreng
  M P 2B P 2B P 2B H K H P 2B     # ayam geprek
  M P 2B                          # Save
B: menu tiles
  M P 4B                          # ubin es teh, diklik dua kali
  M P 2B                          # ubin nasi goreng
  M P 2B                          # ubin ayam geprek
  M P 2B                          # Save
```

Kode 1.1: nama di margin memulai sebuah desain; baris menjorok menambah operator; `2B` berarti `B B`. Buku: Subbab 1.11.3.

Inti `klm.py`

Kode 1.2 Dari `klm.py`: menghitung operator dan menerapkan modelnya.

```python
def count(ops):
    """Hitung operator dalam teks seperti 'M P 2B H K'."""
    counts = dict.fromkeys(ORDER, 0)
    for token in ops.split():
        times, op = int(token[:-1] or 1), token[-1]   # '2B' -> 2, 'B'
        if op not in TIMES:
            raise ValueError(f"unknown operator {token!r}")
        counts[op] += times
    return counts


def seconds(counts):
    """Prediksi waktu: jumlah tiap operator kali waktu satuannya."""
    return sum(counts[op] * TIMES[op] for op in ORDER)
```

Kode 1.2: huruf yang tidak dikenal memicu galat, karena salah ketik yang diam-diam akan membuat desain tampak lebih cepat. `Klm.java` melakukan hal yang sama dengan larik int. Buku: Subbab 1.11.3.

Menjalankannya

```
$ python3 klm.py
$ java -cp out Klm
```

```
Keystroke-Level Model, times in seconds (Card, Moran and Newell)
design              K   P   B   H   M   time
A: dropdown form    3  13  26   6   4  25.54
B: menu tiles       0   4  10   0   4  10.80
B saves 14.74 s per order (57.7%).
At 120 orders a day: 29.5 minutes a day, 12.8 hours a month (26 days).
```

Kedua bahasa mencetak baris yang sama. Pada 120 pesanan sehari, ubin mengembalikan setengah jam sehari dan 12,8 jam kerja sebulan. Buku: Subbab 1.11.4.

Yang Tidak Dikatakan Model

Model ini memprediksi waktu **ahli yang tidak berbuat salah**; pemula lebih lambat dan setiap kesalahan memakan lebih banyak waktu.

Model ini tidak mengatakan apa pun tentang belajar, kepuasan, atau benarnya pesanan.

Ia perkiraan pertama yang murah sebelum ada kode; ujilah dengan orang sungguhan sesudahnya.

Di sini perkiraannya menegaskan apa yang ditunjukkan pengamatan terhadap Bu Ratna: desainnya yang lambat, bukan penggunanya. Uji: 4 di Python, 5 pemeriksaan di Java. Buku: Subbab 1.11.4 dan 1.11.5.

IMK dalam Praktik: Profitina

Keterbacaan skor: setiap sinyal *AI visibility* dijelaskan dengan kata sederhana beserta tren mingguan.

Persetujuan sesuai UU PDP: persetujuan eksplisit, ekspor data, ganti kata sandi, dan hapus akun sebagai tindakan swalayan.

Waktu umpan balik: tulisan langsung muncul; jika gagal terkirim, teks yang diketik dikembalikan.

Aplikasi memberi saran dan pengguna yang memutuskan; ketika model lokal tidak tersedia, aplikasi mengatakannya. Buku: Subbab 1.12.

Kesalahan yang Sering Terjadi

Merancang untuk diri sendiri; menyalahkan pengguna; mengatakan “mudah dipakai” tanpa menyebut bagi siapa.

Memercayai statistik tanpa bertanya apa yang dihitungnya; menganggap KLM sebagai kebenaran.

Memasang dari situs sembarangan; membuka satu berkas, bukan folder, di VS Code.

Setiap kesalahan punya obat murah di dalam bab. Buku: Subbab 1.13.

Ringkasan

1. IMK merancang, mengevaluasi, dan mengimplementasikan sistem interaktif untuk digunakan manusia; *usability*, UX, dan aksesibilitas punya makna ISO.

2. Jurang eksekusi dan jurang evaluasi Norman menjelaskan sebagian besar masalah antarmuka.

3. Di Indonesia: 81,72% daring, Android 77,07% tayangan halaman seluler, 11,97% berusia enam puluh tahun ke atas; hukum kini menuntut antarmuka yang baik.

4. Meja kerjanya adalah JDK 25, Python 3.14, dan VS Code.

5. KLM (Persamaan (1.1)) memprediksi 10,80 s alih-alih 25,54 s untuk pesanan Bu Ratna.

Buku: Subbab 1.14.

Pertemuan Berikutnya

Bab 2 — Faktor Manusia. Persepsi, perhatian, dan memori; hukum Fitts dan hukum Hick; kios puskesmas yang dirancang ulang dengan angka.

Sebelum itu: pasang meja kerja, jalankan `check_setup.py` dan `klm.py`, lalu kerjakan Soal Latihan 1.1, 1.3, 1.4, dan 1.6.

Bu Siti, enam puluh delapan tahun, akan menunjukkan mengapa tombol 20 mm masih bisa terlalu kecil. Buku: Subbab 1.15 dan 1.16.

Kuliah 2 — Faktor Manusia: Persepsi, Kognisi, Memori, Hukum Fitts dan Hick

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 2

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita mempelajari tubuh dan pikiran yang memakai setiap antarmuka, lalu mengubah dua hukum psikologi eksperimental menjadi kalkulator untuk kios puskesmas. Buku: Bab 2, Subbab 2.1–2.16.

Isi Pertemuan Ini

1. **Manusia sebagai pemroses informasi** — tahap persepsi, kognitif, dan motorik, beserta batasnya.
2. **Persepsi** — fovea, mata yang menua, penglihatan warna, pengelompokan Gestalt, apa yang menonjol.
3. **Perhatian dan memori** — satu hal pada satu waktu; sekitar empat bongkahan; mengenali, bukan mengingat.
4. **Hukum Fitts dan hukum Hick** — berapa lama mencapai sasaran dan berapa lama memilih.
5. **Sebuah kalkulator** — mencocokkan kedua hukum pada log dan membandingkan dua tata letak kios dalam Java dan Python.

Buku: Bab 2.

Titik Awal: Bu Siti di Depan Kios

Kios Puskesmas Kenanga: dua belas tombol teks kecil dalam kisi di bagian atas layar.

Bu Siti, 68 tahun, tidak dapat membaca hurufnya dengan kacamata mana pun; jarinya menutupi *Lansia* dan menekan *Anak*.

Tiketnya bertuliskan “A-017, Anak”; ia baru sadar empat puluh menit kemudian, di ruangan yang salah.

Petugas berkata pasien lansia “belum terbiasa dengan teknologi”; dr. Rina bertanya apa yang dituntut kios secara berlebihan. Buku: Subbab 2.1.

Apa yang Dituntut Kios Secara Berlebihan?

Mata: teks kecil pada jarak yang canggung.

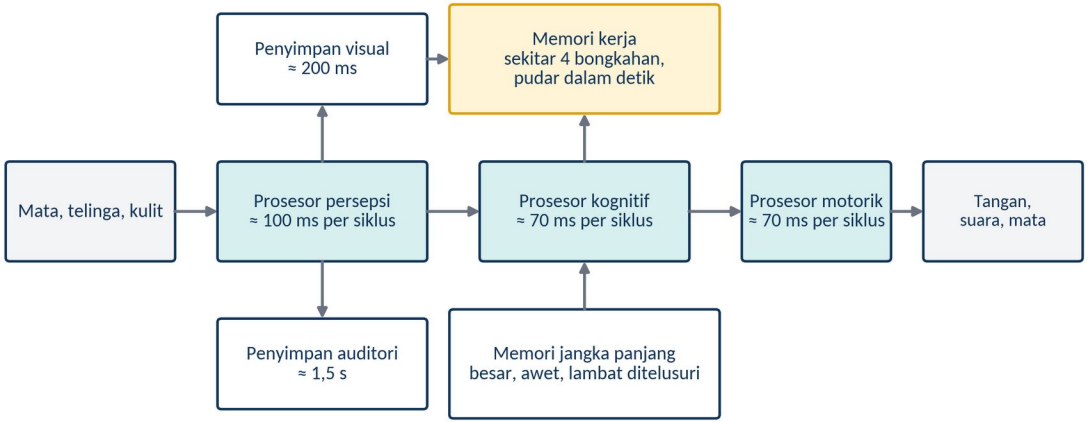
Perhatian: dua belas label yang tampak serupa.

Memori: menahan kata yang dicari sambil membaca yang lain.

Tangan: sasaran 20 mm sejauh jangkauan lengan; dan tidak ada kesempatan kedua untuk menyadari kekeliruan.

11,97% penduduk Indonesia berusia enam puluh tahun ke atas pada 2025 (BPS): pasien poli lansia adalah pengguna tetap kios itu. Buku: Subbab 2.1.

Model Human Processor



Gambar 2.1 Model Human Processor: prosesor persepsi, kognitif, dan motorik dengan waktu siklus yang khas, penyimpan sensorik yang singkat, dan dua jenis memori.

Gambar 2.1: Card, Moran, dan Newell (1983). Reaksi paling sederhana pun memakan beberapa ratus milidetik; setiap tahap dapat kelebihan beban sementara yang lain menunggu. Buku: Subbab 2.2.

Daftar Periksa Tiga Pertanyaan

Bisakah mereka melihatnya? (persepsi)

Bisakah mereka memutuskan? (kognisi dan memori)

Bisakah mereka menjangkaunya? (motorik)

Tanggapan dalam sekitar sepersepuluh detik terasa seketika; satu detik terasa lambat.

Psikologi rekayasa modern punya model yang lebih kaya; tiga tahap itu tetap daftar periksa yang baik. Buku: Subbab 2.2.

Penglihatan: Tajam Hanya di Satu Titik Kecil

Fovea mencakup sekitar dua derajat, selebar kuku ibu jari pada jarak sepanjang lengan; sisanya penglihatan tepi.

Mata melompat beberapa kali per detik (**sakade**); teks hanya terbaca di tempat mata mendarat.

Galat di bagian atas formulir panjang terlewat oleh pengguna yang berada di bagian bawah.

Letakkan pesan di tempat mata berada. Buku: Subbab 2.3.1.

Merancang untuk Orang Lanjut Usia

Ukuran teks yang lega dan kontras yang kuat; layar doff yang jauh dari silau.

Sasaran besar dengan jarak di antaranya; tanpa gestur yang menuntut kecepatan atau ketelitian.

Lebih sedikit opsi per layar, kata-kata sederhana, dan waktu untuk membaca serta membetulkan.

Aturan-aturan yang dihimpun Czaja dan rekan-rekannya ini membantu semua orang; pengguna lanjut usia hanya lebih dulu merasakan ketiadaannya. Buku: Subbab 2.3.1.

Mata yang Menua, dan Warna

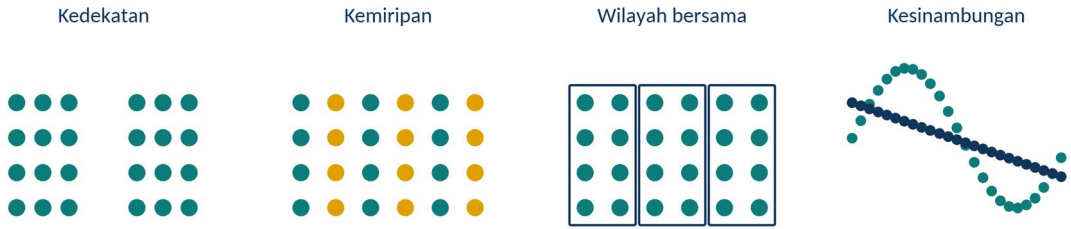
Sejak usia empat puluhan: **presbiopia**; lensa menguning; perlu lebih banyak cahaya dan kontras; lebih mudah silau.

Sekitar **1 dari 12 laki-laki dan 1 dari 200 perempuan** mengalami defisiensi penglihatan warna, kebanyakan merah-hijau.

Jangan pernah menjadikan warna satu-satunya pembawa makna (WCAG 2.2, 1.4.1).

Teks kecil, kontras rendah, dan layar mengilap di ruang yang terang pertama-tama merugikan pengguna lanjut usia. Buku: Subbab 2.3.1.

Mengelompokkan Sebelum Membaca: Gestalt



Gambar 2.2 Empat prinsip Gestalt: kedekatan, kemiripan, wilayah bersama, dan kesinambungan.

Gambar 2.2: dua belas tombol identik dengan jarak identik mengatakan bahwa kedua belas layanan setara; memisahkan yang umum membuat persepsi mengerjakan separuh pencarian. Buku: Subbab 2.3.2.

Yang Menonjol, serta Pendengaran dan Sentuhan

Satu butir merah, besar, atau bergerak langsung ditemukan (fitur **praatensi**); kombinasi fitur diperiksa satu per satu.

Material 3 Expressive: elemen kunci ditemukan hingga empat kali lebih cepat; kesenjangan usia menyempit.

Nomor yang dipanggil harus **ditampilkan dan diumumkan**; layar sentuh perlu umpan balik yang terlihat atau terasa.

Sorotan yang dipakai di mana-mana tidak menyorot apa pun. Buku: Subbab 2.3.3 dan 2.3.4.

Perhatian: Satu Hal pada Satu Waktu

Perhatian selektif dan terbatas; berpindah tugas memakan waktu dan menambah kesalahan.

Inattentional blindness: penonton yang menghitung operan tidak melihat gorila yang melintas.

Letakkan umpan balik di tempat perhatian berada; interupsi dengan sengaja hanya bila kekeliruan mahal akibatnya.

Bu Siti, yang sedang mencari sebuah kata, tidak membaca tiket bertuliskan Anak. Buku: Subbab 2.4.

Memori Kerja: Sekitar Empat Bongkahan

Miller (1956): “tujuh, plus atau minus dua”; Cowan (2001): sekitar **empat bongkahan**, pudar dalam detik.

Pembongkahan: 081234567890 adalah dua belas butir; 0812-3456-7890 adalah tiga.

Tampilkan nomor rekening, *virtual account*, dan kode sekali pakai secara **berkelompok**.

Rancang untuk sekitar empat bongkahan dan biarkan antarmuka membawa sisanya. Buku: Subbab 2.5.1.

Mengenali, dan Model Mental

Mengingat tanpa petunjuk itu lambat; **mengenali** apa yang ditampilkan itu mudah.

Taruh pengetahuan **di dunia**, bukan di kepala: menu, opsi yang terlihat, butir terbaru.

Antrean yang terlihat (“nomor 17 untuk Poli Lansia; 6 orang sebelum Anda”) membangun **model mental** yang benar.

Kode keyboard menuntut pengingatan, ubin menawarkan pengenalan (Soal Latihan 1.3). Buku: Subbab 2.5.2 dan 2.5.3.

Hukum Fitts

$$ID = \log_2\left(\frac{D}{W} + 1\right) \tag{2.1}$$

D: jarak ke sasaran; W: lebarnya; ID: indeks kesulitan dalam bit.

Persamaan (2.1): kesulitan bergantung pada rasio D/W; menggandakan jarak menambah jumlah tetap, tidak menggandakan waktu. Buku: Subbab 2.6.1.

Mengapa Rasio, Bukan Jarak?

Tangan mula-mula bergerak cepat dan kasar, lalu melakukan koreksi-koreksi kecil sampai mendarat di dalam sasaran.

Setiap koreksi memperkecil sisa galat dengan faktor yang kira-kira tetap, sehingga banyaknya koreksi tumbuh dengan $\log(D/W)$.

Dua kali lebih jauh dan dua kali lebih lebar memerlukan koreksi yang sama, sehingga waktunya sama.

Inilah intuisi di balik logaritmanya; hukumnya sendiri adalah hasil empiris, terbukti untuk jari, mouse, stilus, dan mata. Buku: Subbab 2.6.1.

Waktu Gerak dan Throughput

$$MT = a + b \cdot ID \tag{2.2}$$

a dan b bergantung pada orang, perangkat, dan postur, dan **diukur**.

Throughput $TP = 1/b$ bit per detik (ISO 9241-411 memakai penghalusan lebar efektif).

Persamaan (2.2) adalah garis lurus dalam ID, sehingga a dan b diperoleh dengan mencocokkan garis pada ketukan terukur. Buku: Subbab 2.6.1.

Contoh yang Dikerjakan: Tombol Umum

Kios lama: titik tengah $(45, 225)$ mm, tangan di $(110, -40)$: $D = \sqrt{74450} = 272.9$ mm, $W = 20$ mm.

$ID = \log_2(272.9/20 + 1) = 3.872$ bit; $MT = 0.156 + 0.147 \times 3.872 = 0.725$ s.

Kios baru: 95 kali 60 mm, $D = 245.7$ mm, $ID = 2.349$ bit, $MT = 0.501$ s.

W adalah sisi terpendek: jari dari bawah harus berhenti di dalam tinggi tombol. Tombol yang lebih besar menghemat seperempat detik dan tiga kali lebih tinggi. Buku: Subbab 2.6.2.

Ukuran Sasaran dalam Pedoman Terkini

Tabel 2.1 Ukuran minimum sasaran sentuh dalam pedoman terkini.

Sumber	Ukuran sasaran minimum
WCAG 2.2, 2.5.8 (AA)	24 kali 24 piksel CSS, atau ruang yang cukup
WCAG 2.2, 2.5.5 (AAA)	44 kali 44 piksel CSS
Apple Human Interface Guidelines	44 kali 44 *point*
Android dan Material Design	48 kali 48 dp, sekitar 9 mm

Tabel 2.1: lantai, bukan sasaran. Tombol 20 mm kios lama melampaui semuanya dan tetap terlalu kecil bagi Bu Siti. Buku: Subbab 2.6.3.

Hukum Hick

$$RT = a + b \log_2(n+1) \tag{2.3}$$

n opsi yang sama mungkin; $+1$ adalah opsi tidak menanggapi.

Konstanta latihan: dua belas tombol memakan $0.227 + 0.205 \times 3.700 = 0.986$ s; tujuh memakan 0.842 s.

Persamaan (2.3): Hick (1952) dan Hyman (1953); waktu keputusan tumbuh sebanding dengan logaritma banyaknya pilihan. Buku: Subbab 2.7.1.

Pilihan yang Tidak Sama Mungkin: Entropi

$$H = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \tag{2.4}$$

$RT = a + b\,H$: pasien tetap yang baginya *Umum* jauh lebih mungkin membawa ketidakpastian yang lebih kecil.

Persamaan (2.4): Soal Latihan 2.3 menemukan $H = 2.806$ *bit untuk kedua belas porsi kios. Buku: Subbab 2.7.1.*

Kapan Hukum Hick Tidak Berlaku

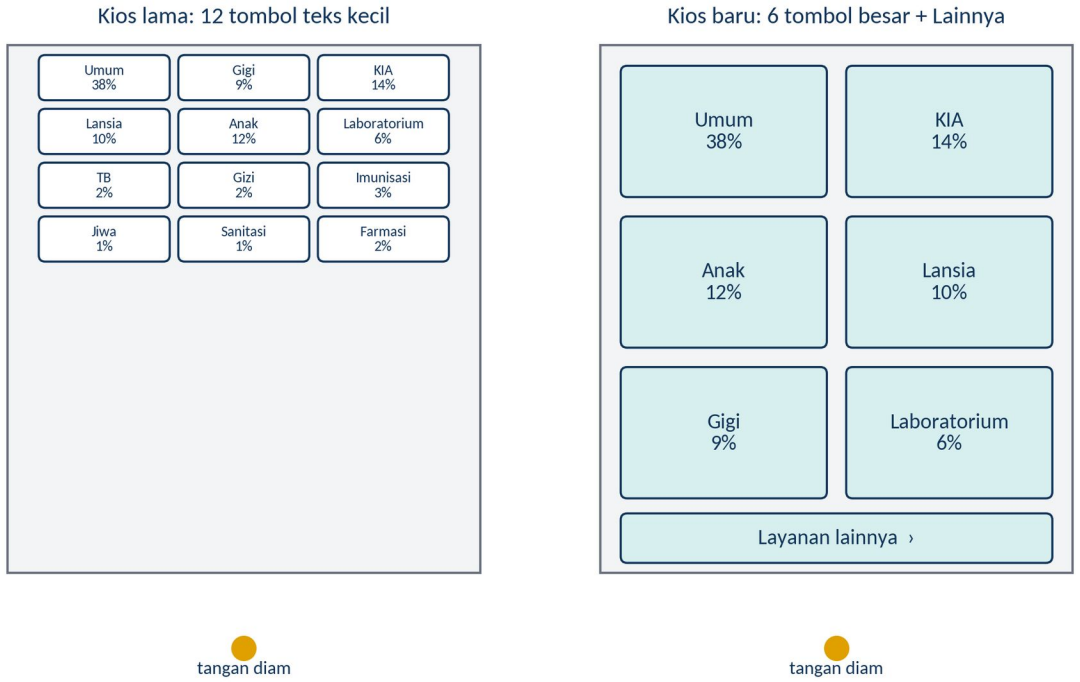
Hukum ini menggambarkan **pilihan yang terlatih** di antara opsi yang dikenal.

Pendatang baru yang **mencari** di antara label asing membacanya satu per satu: waktunya tumbuh kira-kira linear.

Jangan memakainya untuk membela menu panjang; pilih yang lebar dan dangkal daripada yang sempit dan dalam.

Opsi lebih sedikit, yang sering lebih dulu, yang berkaitan berdekatan. Buku: Subbab 2.7.2.

Usulannya: Enam Tombol Besar dan *Lainnya*



Gambar 2.3 Layar pertama kios antrean yang lama dan yang baru, sesuai skala, dengan porsi pasien setiap layanan dan titik diam tangan.

Gambar 2.3: enam layanan di layar pertama melayani 89% pasien. Lebih cepatkah, bagi siapa, dan seberapa banyak? Buku: Subbab 2.8.

Mencocokkan Garis: Kuadrat Terkecil

$$b = \frac{\sum_i (x_i - \acute{x})(y_i - \acute{y})}{\sum_i (x_i - \acute{x})^2}, a = \acute{y} - b\,\acute{x} \tag{2.5}$$

Fitts: $x = ID$, $y = MT$; Hick: $x = \log_2(n+1)$, $y = RT$.

r^2 menyatakan seberapa banyak variasi yang dijelaskan garis.

*Persamaan (2.5). Log di folder bab adalah log **latihan** yang dibangkitkan dari kedua hukum ditambah derau; Soal Latihan 2.6 menggantinya dengan pengukuran Anda sendiri. Buku: Subbab 2.8.1 dan 2.8.2.*

Waktu Harapan Sebuah Tata Letak

Layanan layar 1: Hick atas semua tombol layar 1 + Fitts dari tangan ke tombol.

Layanan layar 2: putuskan dan gerak ke *Other*, lalu putuskan dan gerak dari *Other* ke layanan.

Waktu harapan per pasien: waktu setiap layanan dibobot porsinya.

Model ini mengabaikan pergantian layar dan membaca untuk pertama kali. Buku: Subbab 2.8.3.

Inti `humanfactors.py`

Kode 2.1 Dari `humanfactors.py`: kedua hukum dan pencocokan kuadrat terkecil.

```
def fitts_id(distance, width):
    """Indeks kesulitan dalam bit (bentuk Shannon)."""
    return math.log2(distance / width + 1)


def hick_bits(choices):
    """Bit sebuah pilihan di antara n opsi sama mungkin (plus 'tidak ada')."""
    return math.log2(choices + 1)


def fit_line(xs, ys):
    """Kuadrat terkecil: a dan b dari y = a + b*x, serta r kuadrat."""
    n = len(xs)
    mx, my = sum(xs) / n, sum(ys) / n
    sxy = sum((x - mx) * (y - my) for x, y in zip(xs, ys))
    sxx = sum((x - mx) ** 2 for x in xs)
    b = sxy / sxx
    a = my - b * mx
    ss_res = sum((y - a - b * x) ** 2 for x, y in zip(xs, ys))
    ss_tot = sum((y - my) ** 2 for y in ys)
    return a, b, 1 - ss_res / ss_tot
```

Kode 2.1: `HumanFactors.java` sama, dengan `Math.log(x) / Math.log(2)` untuk logaritma basis 2 dan pembaca CSV kecil. Buku: Subbab 2.8.4.

Menjalankannya

```
$ python3 humanfactors.py
$ java -cp out HumanFactors
```

```
Fitts: MT = 0.156 + 0.147 * ID   (r2 = 0.932, 60 taps)
       throughput 1/b = 6.81 bits/s
Hick:  RT = 0.227 + 0.205 * log2(n+1)   (r2 = 0.922, 48 choices)
kiosk_old: expected 1.70 s per patient; slowest General 1.71 s
  General              38%   1.71 s
  Mother and child     14%   1.71 s
  Children             12%   1.68 s
  Elderly              10%   1.69 s
  Dental                9%   1.70 s
  Laboratory            6%   1.69 s
  Immunization          3%   1.67 s
  Tuberculosis          2%   1.67 s
  Nutrition             2%   1.66 s
  Pharmacy              2%   1.65 s
  Mental health         1%   1.65 s
  Sanitation            1%   1.64 s
```

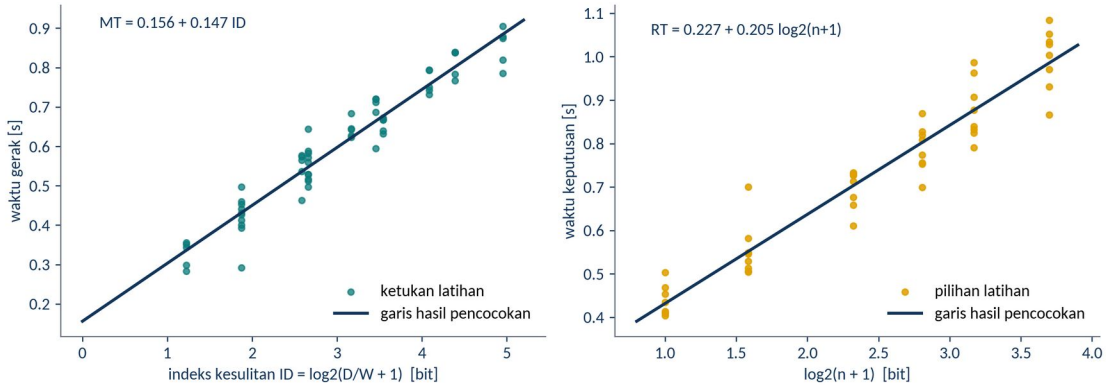
Tiga baris pertama adalah hukum hasil pencocokan; kios lama memerlukan 1,70 s per pasien, hampir sama untuk setiap layanan. Buku: Subbab 2.9.

Tata Letak Baru

```
kiosk_new: expected 1.44 s per patient; slowest Immunization 2.52 s
  General             38%   1.34 s
  Mother and child    14%   1.34 s
  Children            12%   1.29 s
  Elderly             10%   1.29 s
  Dental               9%   1.22 s
  Laboratory           6%   1.22 s
  Immunization         3%   2.52 s
  Tuberculosis         2%   2.52 s
  Nutrition            2%   2.45 s
  Pharmacy             2%   2.45 s
  Mental health        1%   2.38 s
  Sanitation           1%   2.38 s
```

1,44 s per pasien, 15% lebih sedikit; enam layanan umum memakan 1,22–1,34 s; 11% pasien di layar 2 membayar sekitar satu detik lebih. Buku: Subbab 2.9.

Hukum Hasil Pencocokan



Gambar 2.4 Log latihan dengan garis hasil pencocokan: hukum Fitts (kiri) dan hukum Hick (kanan).

Gambar 2.4: garis-garisnya menemukan kembali konstanta pilihan dengan dekat tetapi tidak persis, sebagaimana mestinya bila titik berderau dicocokkan. Buku: Subbab 2.9.

Yang Tidak Terlihat oleh Angka

Membaca label asing: jauh lebih lambat daripada 1,7 s bagi pengguna pertama kali.

Kesalahan: tiket keliru Bu Siti berasal dari sasaran yang tertutup jari; tombol yang lebih tinggi mencegahnya.

Pakai model untuk apa yang diukurnya dan uji *usability* (Bab 7) untuk sisanya.

Uji: 3 di Python, 6 pemeriksaan di Java, pada kasus yang jawabannya pasti. Buku: Subbab 2.9 dan 2.10.

Rekam Log Anda Sendiri

Lima orang atau lebih dari kelompok pengguna sebenarnya, dengan persetujuan mereka, pada ketinggian kios.

Ketukan: empat jarak, tiga lebar, masing-masing lima ketukan, urutan acak; catat jarak, lebar, waktu, dan yang meleset.

Pilihan: sebuah kata dan n tombol, n dari 1 hingga 12, masing-masing delapan kali; cocokkan per orang dan per kelompok.

Dua puluh ketukan dan dua puluh pilihan dari tiga orang seusia pengguna sasaran sudah mengalahkan konstanta buku teks mana pun untuk kios Anda (Soal Latihan 2.6). Buku: Subbab 2.11.

Tinjauan Faktor Manusia, Tahap demi Tahap

Persepsi: ukuran, kontras, warna plus kata, pengelompokan.

Perhatian: satu aksi utama; umpan balik di tempat pengguna memandang.

Memori: sekitar empat bongkahan; mengenali; angka berkelompok.

Keputusan: opsi jarang keluar dari layar pertama. **Gerak**: sasaran besar, dekat, yang berbahaya berjauhan.

Daftar periksa di bab mengubah tiga pertanyaan menjadi tinjauan; ganti log latihan dengan pengukuran pengguna Anda sendiri. Buku: Subbab 2.11.

IMK dalam Praktik: Profitina

Dasbor mengumpulkan langkah berikutnya yang lazim dalam satu layar, masing-masing dapat diklik: lebih sedikit keputusan, lebih sedikit gerakan.

Skor mingguan berupa **garis**, sehingga persepsi, bukan memori, yang menjawab “apakah saya membaik?”.

“N baru” hanya menghitung tulisan tingkat teratas; tugas akun yang jarang menurut UU PDP menjadi aksi terpisah yang berlabel jelas.

Buku: Subbab 2.12.

Kesalahan yang Sering Terjadi

Merancang teks dan sasaran untuk mata dan tangan muda; warna sebagai satu-satunya sinyal.

Memercayai “tujuh plus atau minus dua”; memakai hukum Hick untuk membela menu panjang.

Memasukkan konstanta buku teks ke hukum Fitts; mengambil lebar pada arah yang salah; membaca model waktu sebagai kebenaran seutuhnya.

Buku: Subbab 2.13.

Ringkasan

1. Tahap persepsi, kognitif, dan motorik masing-masing punya batas (Gambar 2.1).

2. Fovea, mata yang menua, 1 dari 12 laki-laki berdefisiensi warna; Gestalt mengelompokkan sebelum kita membaca.

3. Perhatian terbatas; memori kerja menampung sekitar empat bongkahan.

4. Fitts: $MT = a + b\,ID$ (Persamaan (2.2)); Hick: $RT = a + b \log_2(n+1)$ (Persamaan (2.3)).

5. Kios baru: 1,44 s alih-alih 1,70 s, dan jauh lebih sedikit tiket keliru.

Buku: Subbab 2.14.

Pertemuan Berikutnya

Bab 3 — Prinsip dan Heuristik Desain. Norman, Shneiderman, ISO 9241-110, dan Nielsen; Liquid Glass dan Material 3 Expressive; evaluasi heuristik halaman pesanan laundry.

Sebelum itu: jalankan `humanfactors.py` dan kerjakan Soal Latihan 2.1, 2.2, 2.4, dan 2.7.

Empat evaluator, 26 temuan, 14 masalah: halaman pesanan pertama Profitina Laundry. Buku: Subbab 2.15 dan 2.16.

Kuliah 3 — Prinsip dan Heuristik Desain

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 3

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini: prinsip yang menjelaskan mengapa desain berhasil, heuristik untuk memeriksanya, dan evaluasi heuristik halaman pesanan laundry yang diubah menjadi laporan berperingkat. Buku: Bab 3, Subbab 3.1–3.15.

Isi Pertemuan Ini

1. Prinsip Norman — penanda, pemetaan, batasan dan fungsi pemaksa, umpan balik, model konseptual.

2. Tiga keluarga aturan — aturan emas Shneiderman, ISO 9241-110, sepuluh heuristik Nielsen.

3. Prinsip bertemu bahasa desain 2025–2026 — Liquid Glass, Material 3 Expressive, Fluent 2.

4. Evaluasi heuristik — prosedur, keparahan, dan berapa evaluator.

5. Pembuat laporan — menggabungkan 26 temuan dari empat evaluator dalam Java dan Python.

Buku: Bab 3.

Titik Awal: Profitina Laundry Meninggalkan WhatsApp

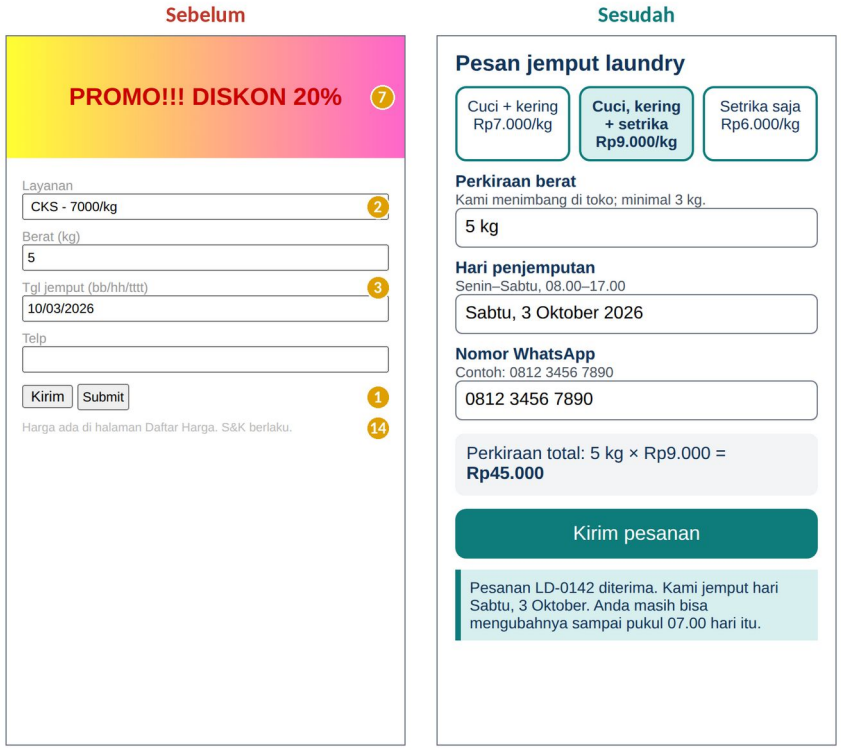
Pesanan Mbak Dewi datang lewat WhatsApp dengan menu bernomor: satu jam setiap malam, dan “2” tertukar dengan “3”.

Halaman pesanan web yang baru berjalan secara teknis, tetapi pada minggu pertama: pesanan ganda, penjemputan hari Minggu, pertanyaan soal harga.

Keponakannya mengusulkan **evaluasi heuristik**: empat orang, daftar aturan singkat, satu sore.

Profitina Laundry adalah studi kasus pengajaran buku ini. Buku: Subbab 3.1.

Sebelum dan Sesudah



Gambar 3.1 Halaman pesanan Profitina Laundry sebelum dan sesudah evaluasi heuristik; penanda bernomor menunjuk masalah P01, P02, P03, P07, dan P14.

Gambar 3.1: kedua halaman adalah halaman pengajaran kita sendiri yang dirender di Chromium. Di akhir pertemuan kita tahu bagaimana halaman kanan dicapai. Buku: Subbab 3.1.

Prinsip, Pedoman, Heuristik

Prinsip: umum, awet, menjawab *mengapa* (orang perlu umpan balik).

Pedoman: khusus platform, menjawab *apa* (sasaran 48 kali 48 dp).

Heuristik: aturan praktis untuk *memeriksa* layar apa pun (visibilitas status sistem).

Evaluasi heuristik: beberapa evaluator memeriksa secara mandiri terhadap heuristik, mencatat lokasi dan keparahan, lalu menggabungkannya. Buku: Subbab 3.2.

Afordans dan Penanda

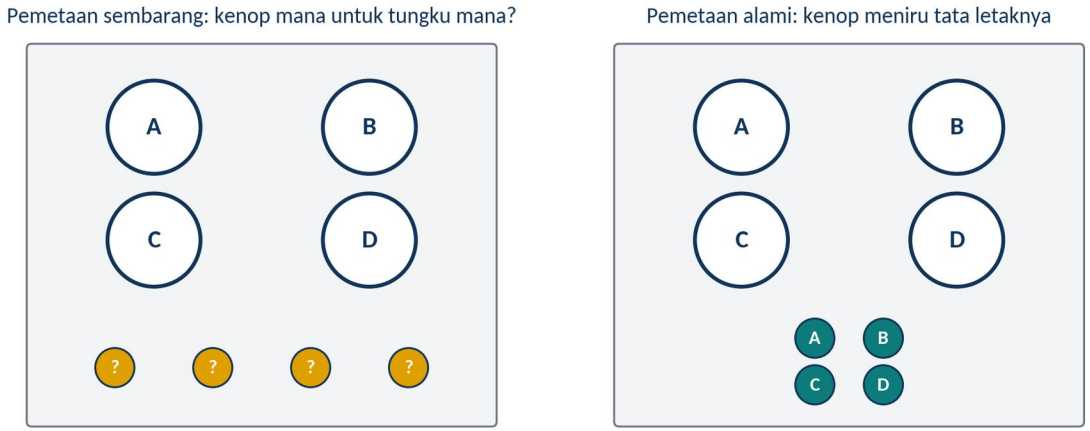
Afordans adalah apa yang dimungkinkan sebuah benda bagi seseorang; **penanda** menunjukkan apa dan di mana.

Di layar hampir apa pun bisa bereaksi, sehingga penandalah yang menentukan.

Kirim dan *Submit* berdampingan adalah penanda yang bertentangan; “CKS” tidak menandai apa pun bagi pelanggan.

“Pengguna tidak melihat bahwa itu tombol” adalah keluhan tentang penanda yang hilang. Buku: Subbab 3.3.1.

Pemetaan



Gambar 3.2 Pemetaan pada kompor: kenop yang berjajar memberi pemetaan sembarang, kenop dalam persegi yang sama dengan tungkunya memberi pemetaan alami.

Gambar 3.2: pemetaan alami tidak memerlukan label. Ubin layanan memetakan tiga layanan ke tiga pilihan yang terlihat beserta harganya. Buku: Subbab 3.3.2.

Kesalahan Pascapenyelesaian di Mana-Mana

Mesin fotokopi kampus: orang mengambil salinannya dan meninggalkan dokumen asli di atas kaca.

Fungsi pemaksa: keluarkan salinan terakhir hanya ketika tutupnya diangkat, atau pakai pengumpan yang mengembalikan dokumen asli bersama salinannya.

Tutup tangki yang tertinggal terbuka setelah membayar bensin: tutup yang tertambat ke mobil adalah batasan yang mencegahnya hilang.

Setiap kali tujuan utama datang sebelum langkah terakhir, perkirakan langkah itu akan terlupa, lalu rancang ulang urutannya (Soal Latihan 3.4). Buku: Subbab 3.3.3 dan 3.15.

Batasan dan Fungsi Pemaksa

Batasan membuat tindakan keliru mustahil: pemilih tanggal yang mengabu-abukan hari Minggu.

Fungsi pemaksa menahan langkah berikutnya sampai langkah sebelumnya selesai.

ATM mengembalikan kartu **sebelum** uang: kesalahan pascapenyelesaian hilang.

Tanpa pelatihan, tanpa papan peringatan: perubahan urutan menghapus kesalahan itu. Buku: Subbab 3.3.3.

Umpan Balik dan Model Konseptual

Umpan balik: cepat dan jelas — “Pesanan LD-0142 diterima” — atau pengguna mengulang tindakan.

Model konseptual: harga per kilogram, berat perkiraan, nomor pesanan, dapat diubah sampai waktu yang disebutkan.

Bersama penanda, pemetaan, dan batasan, keduanya menghasilkan **ketertemuan**.

Umpan balik yang tidak ada itulah sebab pelanggan mengirim pesanan dua kali. Buku: Subbab 3.3.4.

Shneiderman dan ISO 9241-110

Delapan aturan emas: konsistensi; kebergunaan universal; umpan balik informatif; penyelesaian; cegah kesalahan; mudah dibatalkan; pengguna memegang kendali; kurangi beban memori.

ISO 9241-110:2020: kesesuaian dengan tugas; kejelasan diri; kesesuaian dengan harapan; mudah dipelajari; keterkendalian; ketahanan terhadap kesalahan penggunaan; keterlibatan pengguna.

Nilai ISO: tender atau kontrak dapat mensyaratkan kepatuhan terhadapnya. Buku: Subbab 3.4.

Keluarga-Keluarga itu Sejalan

Umpan balik informatif = kejelasan diri = **H1**.

Mudah dibatalkan = keterkendalian = **H3**; cegah kesalahan = ketahanan terhadap kesalahan penggunaan = **H5**.

Kurangi beban memori jangka pendek = **H6**, mengenali, bukan mengingat.

Pilih satu daftar per evaluasi agar semua evaluator berbicara dengan bahasa yang sama. Buku: Subbab 3.4.

Sepuluh Heuristik Nielsen

Tabel 3.1 Sepuluh heuristik usability Nielsen beserta contoh dari halaman pesanan pertama Profitina Laundry.

Kode	Heuristik	Contoh pada halaman pesanan pertama
H1	Visibilitas status sistem	Tidak ada konfirmasi; tidak ada total harga
H2	Kecocokan dengan dunia nyata	Kode “CKS”, “CK”, “SL”
H3	Kendali dan kebebasan pengguna	Tidak bisa mengubah atau membatalkan
H4	Konsistensi dan standar	Dua tombol kirim; tanggal bulan/hari
H5	Pencegahan kesalahan	Tanggal lampau dan hari Minggu diterima
H6	Mengenali, bukan mengingat	Harga di halaman lain
H7	Fleksibilitas dan efisiensi	Tidak ada “ulangi pesanan terakhir”
H8	Desain estetis dan minimalis	Promo beranimasi
H9	Pulih dari kesalahan	“Error 422”
H10	Bantuan dan dokumentasi	Tidak ada penjelasan penimbangan atau minimal 3 kg

Tabel 3.1: diturunkan pada 1994 dari 249 masalah usability dan dipakai hampir tanpa perubahan sejak itu. Ketiga keluarga jauh lebih banyak sepakat daripada berbeda. Buku: Subbab 3.4.

Liquid Glass: Prinsiplah yang Menang

Material tembus pandang Apple (Juni 2025) melayani **H8**: konten dulu, kontrol mundur.

Ia meregangkan **keterbacaan**: teks di atas konten ramai kehilangan kontras, terutama bagi mata yang menua.

iOS 26.1 menambahkan *Settings › Display and Brightness › Liquid Glass*: **Clear** atau **Tinted** — kendali pengguna (H3, H7).

Ketika gaya berbenturan dengan prinsip, prinsiplah yang akhirnya menang. Buku: Subbab 3.5.

Artinya bagi Indonesia

Sebagian besar pengguna memakai ponsel Android, sebagian kecil yang berarti memakai iPhone (Bab 1).
Ikuti konvensi terkini setiap platform (H4): pengguna berharap layar berperilaku seperti aplikasi lain di ponselnya.
Periksa setiap pilihan gaya untuk kontras dan ukuran sasaran bagi pengguna lanjut usia.

Buku: Subbab 3.5.

Material 3 Expressive dan Fluent 2

Material 3 Expressive: warna, bentuk, ukuran, dan gerak yang lebih kuat; 46 studi, 18.000+ partisipan, elemen kunci ditemukan hingga empat kali lebih cepat.

Fluent 2: konsistensi di Windows dan Microsoft 365 (H4).

Ikuti konvensi platform, dan periksa setiap pilihan gaya terhadap prinsip-prinsipnya.

Design system tidak menjamin usability; heuristik memeriksa hasil rakitannya. Buku: Subbab 3.5.

Evaluasi Heuristik dalam Lima Langkah

1. Siapkan heuristik, cakupan, dan templat pencatatan.

2. Evaluasi secara mandiri, setidaknya dua kali penelusuran.

3. Bahas dan gabungkan temuan menjadi masalah-masalah yang berbeda.

4. Nilai keparahan. **5. Laporkan dan prioritaskan**.

Kemandirian penting: evaluator yang saling berbicara menemukan masalah yang sama. Buku: Subbab 3.6.1.

Skala Keparahan Nielsen

Tabel 3.2 Skala keparahan Nielsen untuk masalah usability.

Nilai	Arti
0	Bukan masalah *usability*
1	Kosmetik: perbaiki bila ada waktu
2	Kecil: prioritas rendah
3	Besar: prioritas tinggi
4	Bencana: perbaiki sebelum rilis

Tabel 3.2: keparahan menggabungkan frekuensi, dampak, dan kegigihan masalah. Buku: Subbab 3.6.1.

Berapa Banyak Evaluator?

$$\text{Found}(i) = N\left(1 - (1-\lambda)^{i}\right) \tag{3.1}$$

Setiap evaluator menemukan bagian λ; sebuah masalah terlewat oleh semua i evaluator dengan peluang $(1-\lambda)^{i}$.

Rata-rata Nielsen dan Landauer: $\lambda \approx 0.31$.

Persamaan (3.1): banyaknya masalah yang diharapkan ditemukan oleh i evaluator mandiri. Buku: Subbab 3.6.2.

Memperkirakan λ dan Ukuran Tim

$$\hat{\lambda} = \frac{F}{n \cdot P} \tag{3.2}$$

F temuan oleh n evaluator tentang P masalah berbeda; agak terlalu optimistis.

Persamaan (3.2): untuk halaman laundry, $26/(4 \times 14) = 0.464$*. Buku: Subbab 3.6.2.*

Evaluator yang Diperlukan

$$i \geq \frac{\ln(1-t)}{\ln(1-\lambda)} \tag{3.3}$$

$\lambda = 0.31$: empat evaluator untuk 75% (3.74), tujuh untuk 90% (6.21).

Persamaan (3.3): kedua logaritma negatif, sehingga arah ketaksamaannya tetap. Buku: Subbab 3.6.2.

Efek Evaluator

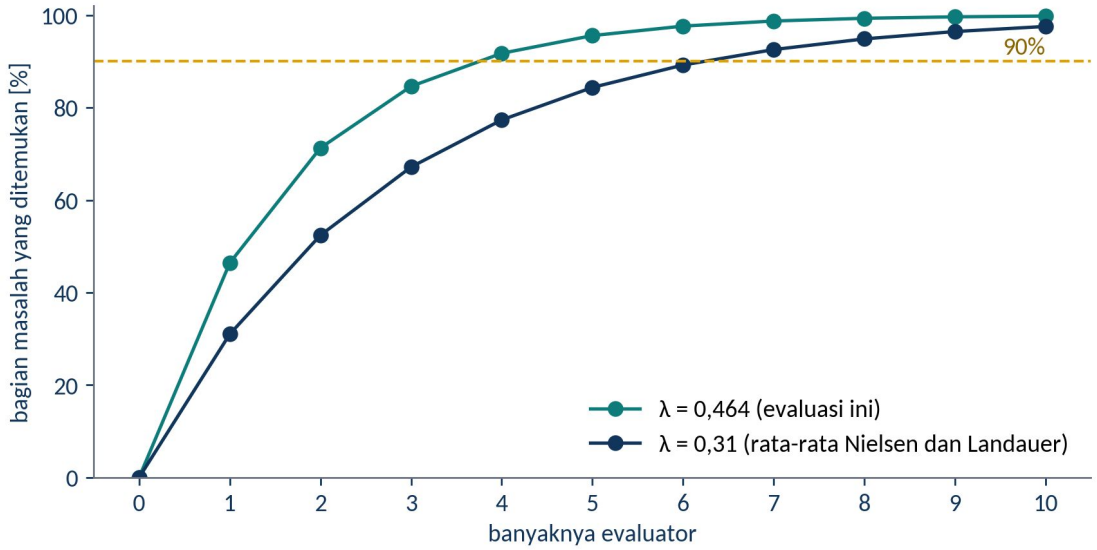
Evaluator yang berbeda menemukan **masalah yang berbeda**: pada halaman laundry tujuh dari empat belas hanya ditemukan satu orang.

Mereka juga **menilai** masalah yang sama secara berbeda, kadang berselisih dua tingkat.

Menggabungkan dan merata-ratakan penilaian mandiri itulah yang membuat daftarnya dapat dipercaya.

Hertzum dan Jacobsen menyebutnya fakta yang mengkhawatirkan tentang metode evaluasi usability. Buku: Subbab 3.6.2.

Hasil yang Makin Berkurang



Gambar 3.3 Bagian masalah yang ditemukan oleh i evaluator untuk λ = 0,464 (evaluasi ini) dan λ = 0,31 (rata-rata Nielsen dan Landauer).

Gambar 3.3: kurva yang melandai adalah alasan saran umum tiga hingga lima evaluator. Efek evaluator menambahkan bahwa penilaian juga berbeda, maka rata-ratakan. Buku: Subbab 3.6.2.

Kerabat: Penelusuran Kognitif dan Tinjauan Pedoman

Penelusuran kognitif: satu tugas, langkah demi langkah; akankah pengguna mencoba, menyadari, mengaitkan, dan melihat kemajuan?

Tinjauan pedoman: daftar panjang aturan khusus, seperti WCAG 2.2; teliti tetapi mekanis.

Urutan praktis: evaluasi heuristik, penelusuran tugas utama, tinjauan aksesibilitas, lalu uji pengguna.

Masing-masing memakan jam, bukan minggu. Buku: Subbab 3.6.3.

Datanya: Masalah dan Temuan

Kode 3.1 Baris-baris pertama `data/findings.csv`: evaluator, masalah, dan keparahan.

```
evaluator,id,severity
E1,P01,4
E1,P02,3
E1,P03,3
E1,P04,3
E1,P08,3
E1,P11,4
E1,P12,2
E1,P14,2
```

Kode 3.1: `problems.csv` menyimpan empat belas masalah hasil pembahasan; `findings.csv` menyimpan 26 penilaian. Penggabungan tetap pekerjaan manusia; menghitung adalah tugas program. Buku: Subbab 3.7.1.

Inti `heuristics.py`

Kode 3.2 Dari `heuristics.py`: penggabungan, pengurutan, dan model Nielsen–Landauer.

```python
def merge(problems, findings):
    """Per masalah: siapa yang menemukan dan dengan keparahan berapa (0-4)."""
    merged = {p["id"]: dict(p, severities=[]) for p in problems}
    for f in findings:
        merged[f["id"]]["severities"].append(int(f["severity"]))
    return merged


def rank(merged):
    """Paling parah dulu; lalu yang ditemukan lebih banyak evaluator; lalu id."""
    def key(p):
        s = p["severities"]
        return (-sum(s) / len(s), -len(s), p["id"])
    return sorted((p for p in merged.values() if p["severities"]), key=key)


def found_share(lam, i):
    """Nielsen dan Landauer: bagian masalah yang ditemukan i evaluator."""
    return 1 - (1 - lam) ** i


def evaluators_needed(lam, target):
    """Banyak evaluator terkecil yang prediksi bagiannya >= target."""
    return math.ceil(math.log(1 - target) / math.log(1 - lam))
```

Kode 3.2: pengurutan dengan tuple (rata-rata keparahan, evaluator, id). `Heuristics.java` menambahkan pemecah CSV karakter demi karakter untuk koma berkutip. Buku: Subbab 3.7.2.

Menjalankannya

```
$ python3 heuristics.py
$ java -cp out Heuristics
```

```
Heuristic evaluation: 4 evaluators, 14 problems, 26 findings
rank id   heur found mean max location
   1 P01  H1   3/4   3.7   4  Send button
   2 P11  H1   4/4   3.5   4  Total price
   3 P08  H3   2/4   3.0   3  After sending
   4 P02  H2   3/4   2.7   3  Service list
   5 P03  H5   2/4   2.5   3  Pickup date
   6 P04  H9   2/4   2.5   3  Error message
   7 P14  H8   2/4   2.5   3  Small print
   8 P12  H5   2/4   2.0   2  Phone number
   9 P05  H6   1/4   2.0   2  Prices
  10 P07  H8   1/4   2.0   2  Promo banner
  11 P09  H7   1/4   2.0   2  Regular customers
  12 P13  H4   1/4   2.0   2  Date format
  13 P06  H4   1/4   1.0   1  Buttons
  14 P10  H10  1/4   1.0   1  Weight
```

Tiga masalah punya rata-rata keparahan 3 atau lebih: tidak ada konfirmasi, tidak ada total harga, tidak bisa mengubah pesanan — penyebab kekacauan minggu pertama. Buku: Subbab 3.8.

Sisa Laporan

```
Fix before launch (mean severity 3 or more): P01, P11, P08
Problems per heuristic: H1 2, H2 1, H3 1, H4 2, H5 2, H6 1, H7 1, H8 2, H9 1, H10 1
Each evaluator found 46.4% of the problems on average (lambda).
Predicted share found by i evaluators:
  1: 46%  2: 71%  3: 85%  4: 92%  5: 96%  6: 98%
Evaluators needed for 90%: 4
```

Hanya satu masalah ditemukan keempat evaluator dan tujuh hanya oleh satu; $\hat{\lambda} = 0.464$ *memprediksi sekitar 92% ditemukan oleh empat. Buku: Subbab 3.8.*

Evaluator Kelima

```
$ python3 heuristics.py data5
```

```
Fix before launch (mean severity 3 or more): P01, P11, P08
Problems per heuristic: H1 2, H2 1, H3 1, H4 2, H5 2, H6 1, H7 1, H8 2, H9 1, H10 1
Each evaluator found 44.3% of the problems on average (lambda).
Predicted share found by i evaluators:
  1: 44%  2: 69%  3: 83%  4: 90%  5: 95%  6: 97%
Evaluators needed for 90%: 4
```

E5 menemukan lima masalah, tidak ada yang baru: $\hat{\lambda}$ *turun menjadi* $31/(5\times14)=0.443$ *dan daftar empat belas masalah tampak hampir lengkap (Soal Latihan 3.3). Buku: Subbab 3.15.*

Mengapa Tidak Hanya Bertanya kepada Pengguna?

Evaluasi heuristik tidak memerlukan pengguna, laboratorium, dan hanya sedikit waktu: inilah *usability* **diskon**.
Ia menemukan apa yang dapat diramalkan para ahli, bukan apa yang hanya terungkap oleh pengguna sungguhan.
Pakailah lebih dulu untuk menghapus masalah yang mencolok, lalu ujilah bersama orang (Bab 7).

Buku: Subbab 3.2.

Dari Laporan ke Desain Ulang

Ubin layanan yang terlihat dengan harga (P02, P05); penimbangan dijelaskan (P10).

Hari jemput dengan jam buka (P03, P13); contoh nomor telepon (P12); total sebelum mengirim (P11).

Satu tombol kirim (P06); konfirmasi dengan nomor pesanan dan batas perubahan (P01, P08); tanpa spanduk atau cetakan abu-abu (P07, P14).

Halaman kanan Gambar 3.1 adalah hasilnya; Bab 7 memeriksanya dengan pelanggan sungguhan. Uji: 3 di Python, 5 pemeriksaan di Java. Buku: Subbab 3.8 dan 3.9.

Templat Pencatatan

Tabel 3.3 Templat pencatatan untuk evaluasi heuristik.

Kolom	Contoh
Evaluator	E2
Lokasi	Halaman pesanan, kolom tanggal jemput
Masalah	Menerima hari Minggu, padahal toko tutup
Heuristik	H5
Keparahan	3
Usulan	Tawarkan hanya hari buka

Tabel 3.3: satu baris per masalah; setelah pembahasan baris-baris itu menjadi `problems.csv` dan `findings.csv`. Buku: Subbab 3.10.

Memakainya Sungguh-sungguh

Tiga hingga lima evaluator, masing-masing dua jam, satu templat: lokasi, masalah, heuristik, keparahan, usulan. Tuliskan masalah sebagai **pengamatan**, bukan solusi; catat juga temuan positif; evaluasi lagi setelah perbaikan. Untuk fitur AI, tambahkan pedoman interaksi manusia-AI sebagai heuristik tambahan (Bab 13).

Buku: Subbab 3.10.

IMK dalam Praktik: Profitina

H1: pengiriman optimistis; tulisan yang gagal terkirim mengembalikan teksnya dengan pesan yang jelas.

H5 dan H9: Link Safety menandai tautan berisiko dan menjelaskan alasannya.

H3: ekspor, ganti kata sandi, dan hapus akun secara swalayan menurut UU PDP; **H2**: sinyal dalam bahasa sederhana; batas kanal dinyatakan di samping tombolnya.

Buku: Subbab 3.11.

Kesalahan yang Sering Terjadi

Mengevaluasi bersama sejak awal; memakai satu evaluator; menulis solusi alih-alih masalah.

Menilai keparahan sendirian; menganggap evaluasi heuristik sebagai uji pengguna.

Mengikuti *design system* membabi buta; memaksakan setiap masalah ke sebuah heuristik; berhenti setelah satu putaran.

Buku: Subbab 3.12.

Ringkasan

1. Prinsip menjawab mengapa, pedoman menjawab apa, heuristik memungkinkan kita memeriksa.

2. Norman: penanda, pemetaan (Gambar 3.2), batasan, umpan balik, model konseptual.

3. Shneiderman, ISO 9241-110, dan Nielsen (Tabel 3.1) sejalan.

4. Bahasa desain menerapkan dan meregangkan prinsip; Liquid Glass mendapat opsi *Tinted*.

5. i evaluator menemukan $1-(1-\lambda)^i$ (Persamaan (3.1)); empat evaluator menemukan 14 masalah pada halaman laundry.

Buku: Subbab 3.13.

Pertemuan Berikutnya

Bab 4 — Proses Berpusat pada Pengguna dan Design Thinking. ISO 9241-210, Double Diamond, *design thinking*; diagram afinitas, persona, peta perjalanan; pangkalan ojek yang ingin punya aplikasi.

Sebelum itu: jalankan `heuristics.py`, evaluasi halaman yang Anda pakai sehari-hari dengan templatnya, dan kerjakan Soal Latihan 3.1, 3.2, 3.4, dan 3.6.

Buku: Subbab 3.14 dan 3.15.

Kuliah 4 — Proses Berpusat pada Pengguna dan Design Thinking

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 4

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini: cara menata proyek agar membangun hal yang tepat, alat-alat yang mengubah wawancara menjadi kebutuhan, dan program yang mengerjakan pembukuan sintesis. Buku: Bab 4, Subbab 4.1–4.15.

Isi Pertemuan Ini

1. **Desain berpusat pada pengguna** — tiga kebiasaan Gould dan Lewis; prinsip dan aktivitas ISO 9241-210.
2. **Double Diamond dan design thinking** — serta Lean UX dalam tim *agile*.
3. **Alat-alat kerja** — diagram afinitas, persona, peta perjalanan, pertanyaan HMW, matriks dampak-upaya, kebutuhan terukur.
4. **AI dalam proses, 2026** — apa yang bisa dan tidak bisa dilakukannya; data riset menurut UU PDP.
5. **Program sintesis** — dari 37 catatan ke prioritas dalam Java dan Python.

Buku: Bab 4.

Titik Awal: Pangkalan Ojek Ingin Punya Aplikasi

Sebuah pangkalan ojek di pasar di Jawa Timur: dua puluh pengemudi, penumpang tetap, banyak di antaranya orang tua yang menelepon pangkalan.

Pak Darto meminta mahasiswa membuat “aplikasi seperti yang besar-besar, tetapi milik kita”; mereka mulai menggambar peta.

Dosen mereka bertanya: **masalah siapa yang akan dipecahkan peta itu?** Tidak ada yang tahu.

Banyak penumpang tetap tidak dapat membaca peta; pengemudi mengenal kota lewat masjid dan warung, bukan nama jalan. Buku: Subbab 4.1.

Apa yang Dilakukan Tim Sebagai Gantinya

Tiga pagi di pangkalan; enam penumpang dan empat pengemudi; setiap pengamatan pada selembar catatan.

Temuan: saat terburuk adalah **menunggu tanpa kepastian** dan **menyepakati harga**; pengemudi khawatir soal **giliran yang adil**.

Usulan: tabel tarif, pesan “pengemudi sedang menuju”, tautan berbagi perjalanan, kemudian antrean yang adil — bukan aplikasi lengkap.

Sebagian kecil biaya semula, dan menjawab apa yang benar-benar dikatakan orang. Buku: Subbab 4.1.

Gagasannya dalam Tiga Kebiasaan

Fokus sejak awal pada pengguna dan tugas.

Ukur secara empiris bagaimana orang memakai simulasi dan prototipe.

Rancang secara iteratif: perbaiki apa yang diungkap pengukuran lalu ukur lagi.

Gould dan Lewis (1985); setiap proses modern menata ketiga kebiasaan ini. Lawannya adalah merancang dari dalam ke luar: dari teknologi, pesaing, atau keinginan klien. Buku: Subbab 4.2.

Permintaan adalah Hipotesis

"Aplikasi seperti yang besar-besar" adalah usulan **solusi**, bukan kebutuhan.

Pak Darto mengenal penumpangnya lebih baik daripada siapa pun dan merupakan peserta yang penting.

Proses dimulai dengan memahami masalah di balik permintaan itu.

Desain berpusat pada pengguna, dalam istilah ISO desain berpusat manusia, membuat sistem mudah dipakai dan berguna dengan berfokus pada pengguna, kebutuhan, dan persyaratannya. Buku: Subbab 4.2.

ISO 9241-210: Enam Prinsip

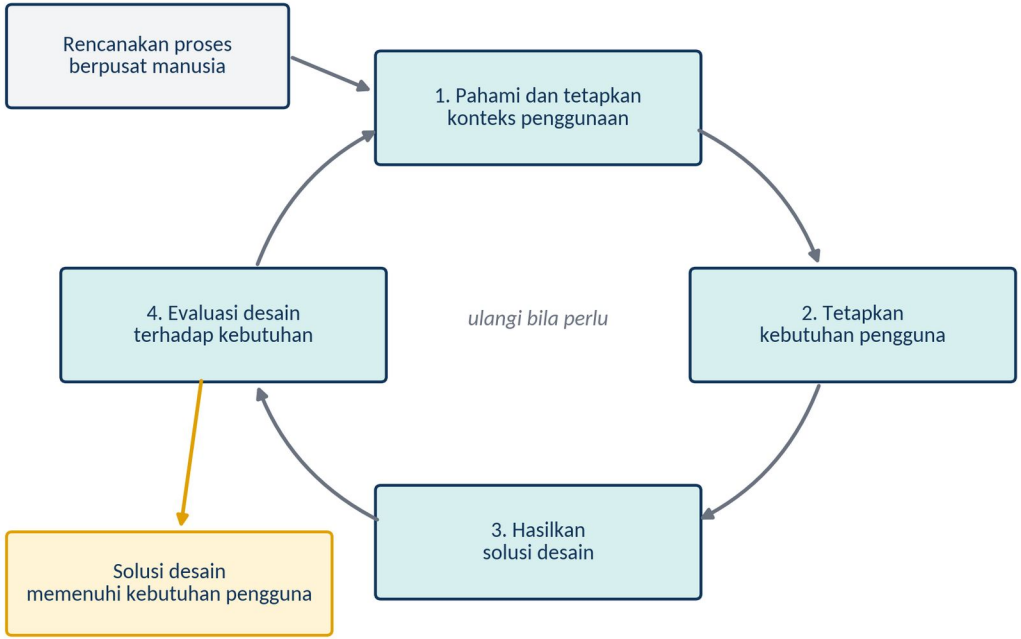
Pemahaman eksplisit tentang pengguna, tugas, dan lingkungan; pengguna dilibatkan di sepanjang proses.

Desain digerakkan dan diperhalus oleh evaluasi berpusat pengguna; proses yang iteratif.

Seluruh pengalaman pengguna; tim multidisiplin (seorang pengemudi dan seorang penumpang ikut sesi desain).

Prinsip menyatakan apa yang harus berlaku pada proyek; aktivitas menyatakan apa yang dikerjakan tim. Buku: Subbab 4.3.

ISO 9241-210: Empat Aktivitas dalam Siklus



Gambar 4.1 Aktivitas desain berpusat manusia dalam ISO 9241-210: rencanakan, lalu berputar melalui konteks, kebutuhan, desain, dan evaluasi hingga solusinya memenuhi kebutuhan pengguna.

Gambar 4.1: evaluasi dapat mengirim tim kembali ke aktivitas mana pun, bahkan yang pertama, bila terungkap bahwa konteksnya disalahpahami. Buku: Subbab 4.3.

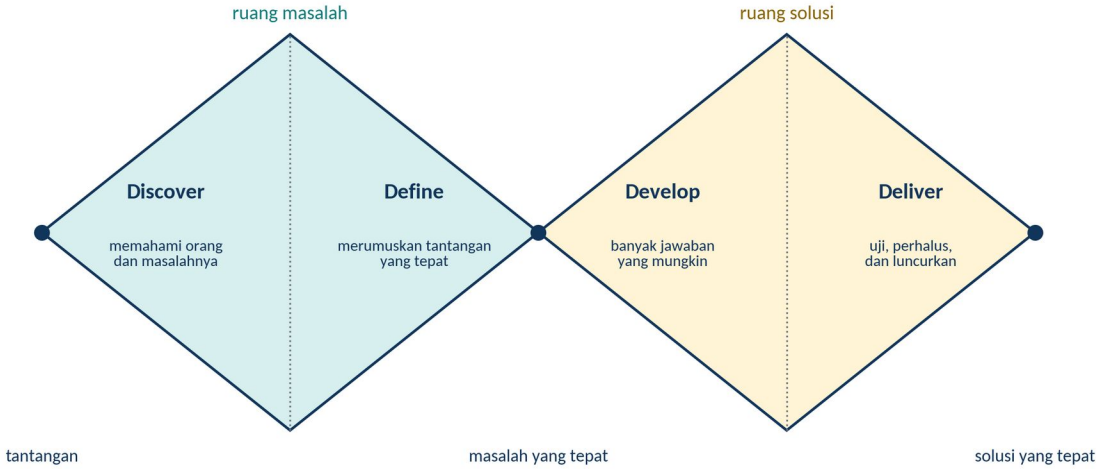
Konteks Penggunaan dan Kebutuhan Pengguna

Konteks penggunaan: pengguna, tujuan, tugas, sumber daya, dan lingkungan — terik di depan pagar, ponsel murah berkuota kecil, kebisingan pasar.

Kebutuhan pengguna: apa yang harus dapat dilakukan pengguna, dan seberapa baik, ditulis agar dapat **diuji**.

Konteks "tertentu" dalam definisi usability di Bab 1. Buku: Subbab 4.3.

Double Diamond



Gambar 4.2 Double Diamond: discover dan define di ruang masalah, develop dan deliver di ruang solusi; setiap berlian mula-mula melebar, lalu menyempit.

Gambar 4.2: UK Design Council, 2004; Framework for Innovation 2019; Systemic Design Framework 2021. Kesalahan paling umum adalah melompati berlian pertama. Buku: Subbab 4.4.1.

Design Thinking dan Lean UX

Lima mode d.school: *empathize*, *define*, *ideate*, *prototype*, *test* — bukan urutan satu arah.

Lean UX: setiap keputusan desain adalah hipotesis dengan ukuran keberhasilan.

Dual-track: jalur penemuan berjalan satu atau dua siklus di depan jalur penghantaran.

“Kami yakin menampilkan tarif sebelum memesan mengurangi tawar-menawar; kami akan tahu bila makin sedikit penumpang yang menanyakan harga lewat telepon.” Buku: Subbab 4.4.2 dan 4.4.3.

Satu Siklus, Tiga Kosakata

Tabel 4.1 Siklus yang sama dalam tiga kosakata, beserta metode yang lazim untuk setiap langkah.

ISO 9241-210	Double Diamond	Design thinking	Metode
Konteks penggunaan	Discover	Empathize	Wawancara, pengamatan (Bab 5)
Kebutuhan pengguna	Define	Define	Afinitas, persona, peta perjalanan
Solusi desain	Develop	Ideate, prototype	Sketsa, prototipe (Bab 6)
Evaluasi	Deliver	Test	Heuristik (Bab 3), uji (Bab 7)

Tabel 4.1: pilih satu kosakata agar tim berbicara dengan satu bahasa; yang lain adalah terjemahannya. Buku: Subbab 4.4.3.

Diagram Afinitas

Satu pengamatan per catatan, sebagai **fakta dengan kata-kata partisipan**, bukan tafsiran.

Kelompokkan catatan dari bawah ke atas tanpa kategori yang ditetapkan; namai kelompoknya: **tema**.

37 catatan membentuk delapan tema, dari “Waktu tunggu tak pasti” hingga “Giliran yang adil”.

Memindahkan catatan adalah penafsiran dan tetap pekerjaan manusia. Buku: Subbab 4.5.1.

Hitung Orang, Bukan Catatan

$$r_t = \frac{|P_t|}{|P|} \tag{4.1}$$

P_t: partisipan dengan catatan dalam tema t; P: semua partisipan.

Sepuluh catatan dari satu orang yang banyak bicara adalah satu pandangan; lima catatan dari lima orang adalah pola.

Persamaan (4.1): jangkauan sebuah tema. Buku: Subbab 4.5.1.

Persona yang Dibangun dari Catatan

Tabel 4.2 Persona yang dibangun dari catatan riset, beserta implikasi desain yang disarankannya.

Kolom	Isi
Nama dan konteks	Mbah Sri, 72 tahun, berjualan sayur di pasar
Tujuan	Sampai di pasar pukul 05.30, pulang tengah hari, tanpa bergantung pada anaknya
Kekesalan	Berdiri di bawah terik tanpa tahu berapa lama; tidak dapat membaca peta
Implikasi desain	Tanpa peta; nama pengemudi dan waktu tiba; patokan, bukan alamat

Tabel 4.2: “Akankah Mbah Sri memakai ini?” punya jawaban konkret; “akankah pengguna memakai ini?” tidak. Persona tanpa bukti adalah stereotipe. Buku: Subbab 4.5.2.

Peta Perjalanan

Tahap: **Mencari** pengemudi, **Sepakat** soal tempat dan harga, **Menunggu**, **Naik**, **Bayar**.

Untuk setiap tahap: tindakan, pikiran, perasaan, titik sentuh, dan keluhan dari 1 (ringan) sampai 5 (berat).

$$\acute{p}_s = \frac{1}{\left|N_s\right|} \sum_{n \in N_s} p_n \tag{4.2}$$

Persamaan (4.2): rata-rata keluhan tahap s atas catatan-catatannya N_s. Buku: Subbab 4.5.3.

Peta Empati dan Jobs to Be Done

Peta empati: dikatakan, dipikirkan, dilakukan, dirasakan — Mbah Sri *mengatakan* tidak keberatan menunggu tetapi *berdiri* di depan pagar dua puluh menit lebih awal.

Jobs to be done: tidak ada yang menginginkan aplikasi ojek; orang ingin “sampai di pasar sebelum sayur yang bagus habis”.

Merumuskan kebutuhan sebagai pekerjaan memisahkannya dari satu solusi tertentu. Buku: Subbab 4.5.4.

Pernyataan Masalah dan “Bagaimana Kita Bisa”

Pernyataan masalah: pengguna, kebutuhan, wawasan — penumpang tetap lanjut usia perlu tahu kapan pengemudi tiba, karena menunggu tanpa kepastian membuat mereka berangkat terlalu awal.

HMW: “Bagaimana kita bisa menunjukkan kepada penumpang kapan pengemudinya tiba?”

Terlalu sempit: “membangun aplikasi”; terlalu luas: “memperbaiki transportasi”.

Pertanyaan HMW yang baik cukup luas untuk banyak jawaban dan cukup sempit untuk memberi arah. Buku: Subbab 4.5.5.

Matriks Dampak-Upaya

$$v = \frac{I}{E} \tag{4.3}$$

Menang cepat: dampak tinggi, upaya rendah. **Taruhan besar**: dampak tinggi, upaya tinggi.

Pengisi: dampak rendah, upaya rendah. **Tunda dulu**: dampak rendah, upaya tinggi.

Persamaan (4.3): nilai per upaya mengurutkan gagasan di dalam kuadran; suara memecah hasil seri. Skor-skornya adalah pertimbangan yang dibuat terlihat oleh matriks. Buku: Subbab 4.5.6.

Kebutuhan Pengguna yang Terukur

“Penumpang tetap berusia 60+ memesan dari tautan tersimpan dalam kurang dari 30 detik tanpa bantuan, pada 4 dari 5 percobaan.”

“Setelah memesan, 9 dari 10 penumpang dapat menyebutkan nama pengemudi dan perkiraan waktu tibanya.”

“Dalam sebulan, telepon yang menanyakan di mana pengemudinya turun separuh.”

Masing-masing dapat diperiksa dalam uji usability (Bab 7) atau dengan data (Bab 14). Buku: Subbab 4.5.7.

AI dalam Proses, 2026

Develop: agen Figma dari Config 2026, varian tata letak, draf teks, prototipe yang dapat diklik, ringkasan transkrip.

Discover: “pengguna sintetis” mencerminkan apa yang lazim dalam data latih, bukan apa yang benar tentang orang-orang ini.

AI boleh membantu Anda **mengolah** apa yang dikatakan orang; AI tidak boleh **mengarang** apa yang akan mereka katakan.

Tidak ada model bahasa yang akan menemukan kekhawatiran pengemudi soal giliran yang adil. Buku: Subbab 4.6.

Data Riset adalah Data Pribadi

Catatan, rekaman, dan foto partisipan tunduk pada **UU PDP 27/2022**; **PP 33/2026** berlaku mulai Januari 2027.

Jelaskan tujuannya, minta persetujuan untuk mencatat atau merekam, pakai kode (P1, D1), hapus rekaman setelah analisis.

Menempelkan transkrip mentah lengkap dengan nama ke alat AI berbasis awan adalah pengalihan data pribadi.

Buku: Subbab 4.6.

Datanya: Catatan

Kode 4.1 Baris-baris pertama `data/notes.csv`: partisipan, peran, tahap, tema, keluhan, dan catatannya.

```
id,participant,role,stage,theme,pain,note
N01,P1,passenger,Find,T3,4,Calls the base but the phone is busy at 7 am
N02,P1,passenger,Agree,T1,4,Never sure the price is the same as last time
N03,P1,passenger,Wait,T2,5,Waits at the gate not knowing if anyone is coming
N04,P2,passenger,Find,T3,3,Does not have the base number saved on the new phone
N05,P2,passenger,Agree,T4,4,"Explains ""after the mosque, green fence"" twice"
N06,P2,passenger,Pay,T7,3,Driver has no change for a 50 thousand note
N07,P3,passenger,Find,T6,3,Tried a big app; too many screens and promotions
```

Kode 4.1: 37 catatan; `themes.csv` menyimpan label dan frasa HMW, `ideas.csv` menyimpan dampak, upaya, dan suara. Buku: Subbab 4.7.1.

Inti `synthesis.py`

Kode 4.2 Dari `synthesis.py`: tema beserta orangnya, perjalanan, kuadran, dan peringkat gagasan.

```
def affinity(notes):
    """Kelompokkan catatan per tema: jumlah catatan, orang berbeda, peran."""
    board = {}
    for n in notes:
        t = board.setdefault(n["theme"], {"notes": 0, "people": set(),
                                          "roles": set()})
        t["notes"] += 1
        t["people"].add(n["participant"])
        t["roles"].add(n["role"])
    return sorted(board.items(),
                  key=lambda kv: (-len(kv[1]["people"]), -kv[1]["notes"], kv[0]))


def journey(notes):
    """Rata-rata keluhan (1 ringan-5 berat) dan jumlah catatan per tahap."""
    result = []
    for stage in STAGES:
        pains = [int(n["pain"]) for n in notes if n["stage"] == stage]
        result.append((stage, len(pains), sum(pains) / len(pains)))
    return result


def quadrant(impact, effort):
    """Matriks dampak-upaya: dampak tinggi 4 atau 5, upaya rendah 1 atau 2."""
    if impact >= 4:
        return "Quick win" if effort <= 2 else "Big bet"
    return "Fill-in" if effort <= 2 else "Avoid for now"
```

```
def prioritise(ideas):
    """Kuadran dulu, lalu nilai per upaya, lalu suara."""
    rows = []
    for i in ideas:
        impact, effort = int(i["impact"]), int(i["effort"])
        rows.append((quadrant(impact, effort), impact, effort,
                     impact / effort, int(i["votes"]), i["idea"]))
    return sorted(rows, key=lambda r: (QUADRANTS.index(r[0]), -r[3], -r[4]))
```

Kode 4.2: himpunan menghitung orang yang berbeda. `Synthesis.java` memakai record dan TreeSet, serta pembaca CSV untuk koma berkutip. Buku: Subbab 4.7.2.

Menjalankannya: Papan Afinitas

```
$ python3 synthesis.py
$ java -cp out Synthesis
```

```
Affinity board: 37 notes from 10 people (6 passengers, 4 drivers)
theme label                  notes people roles
T2    Unknown waiting time       7      7  passenger+driver
T1    Unclear price              5      5  passenger+driver
T3    Calling the base           5      5  passenger+driver
T4    Explaining the place       5      5  passenger+driver
T7    Cash and change            5      5  passenger+driver
T5    Safety at night            4      4  passenger+driver
T6    Big apps feel hard         3      3  passenger+driver
T8    Fair turns                 3      3  driver
```

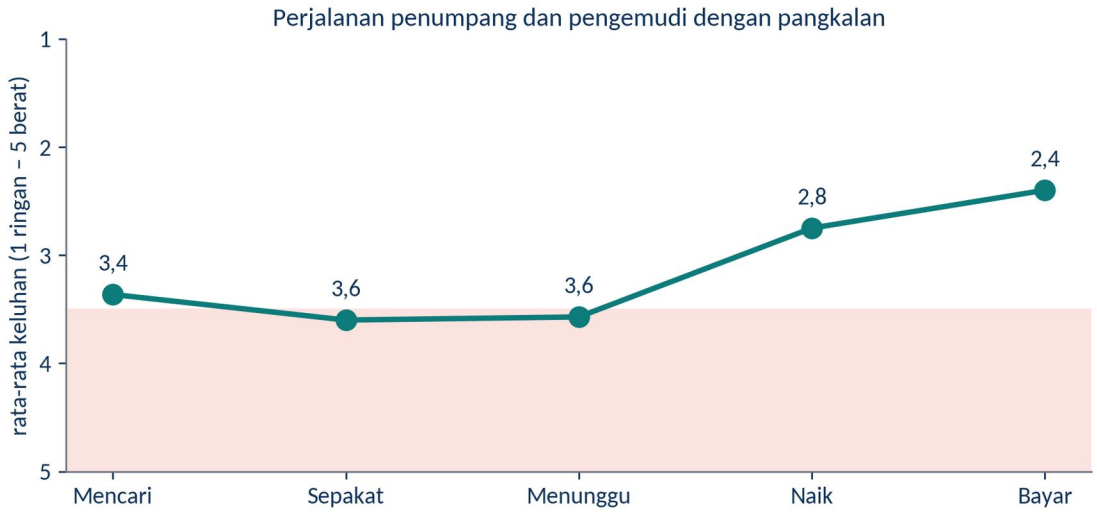
Menunggu tanpa kepastian menjangkau tujuh dari sepuluh orang, $r = 70\ \%$; Fair turns hanya datang dari pengemudi dan akan terlewat tanpa mereka. Buku: Subbab 4.8.

Perjalanan dan Pertanyaannya

```
Journey: mean pain per stage (1 mild, 5 severe)
  Find   11 notes  3.4  #######
  Agree  10 notes  3.6  #######
  Wait    7 notes  3.6  #######
  Ride    4 notes  2.8  ######
  Pay     5 notes  2.4  #####
How might we ...
  1. ... show passengers when their driver will arrive? (T2, 7 people)
  2. ... make the fare clear before the ride starts? (T1, 5 people)
  3. ... let passengers book without a phone call? (T3, 5 people)
```

Agree dan Wait berrata-rata 3,6: untuk Agree, $36/10 = 3.6$*. Keluhannya ada sebelum perjalanan, bukan selama perjalanan. Buku: Subbab 4.8.*

Garis Emosi



Gambar 4.3 Garis emosi peta perjalanan: rata-rata keluhan per tahap dari 37 catatan; sumbunya dibalik dan pita berarsir menandai keluhan berat.

Gambar 4.3: temuan yang perlu disampaikan terus terang kepada Pak Darto — perjalanan yang dibanggakan pengemudi sudah baik; yang menyakitkan adalah semua yang sebelumnya. Buku: Subbab 4.8.

Gagasan yang Diprioritaskan

```
Ideas by impact and effort
quadrant      imp eff ratio votes idea
Quick win       5   2  2.50     9  Fixed fare table by zone shown before booking
Quick win       5   2  2.50     8  Driver-on-the-way message with name and photo
Quick win       4   2  2.00     6  Share-my-ride link for family on night trips
Big bet         5   3  1.67     7  Automatic fair queue for the base
Big bet         4   3  1.33     7  One-tap booking page from a saved WhatsApp link
Big bet         4   4  1.00     5  Pick the pickup point from photos of landmarks
Fill-in         3   2  1.50     4  Cooperative QRIS with the amount shown
Avoid for now   3   4  0.75     4  Voice booking by phone with a menu
Avoid for now   2   3  0.67     3  Loyalty points for regular passengers
Avoid for now   3   5  0.60     2  Own full ride-hailing app like the big ones
```

Tiga gagasan menang cepat, tiga taruhan besar; “aplikasi transportasi lengkap milik sendiri” berakhir terakhir dengan dua suara. Buku: Subbab 4.8.

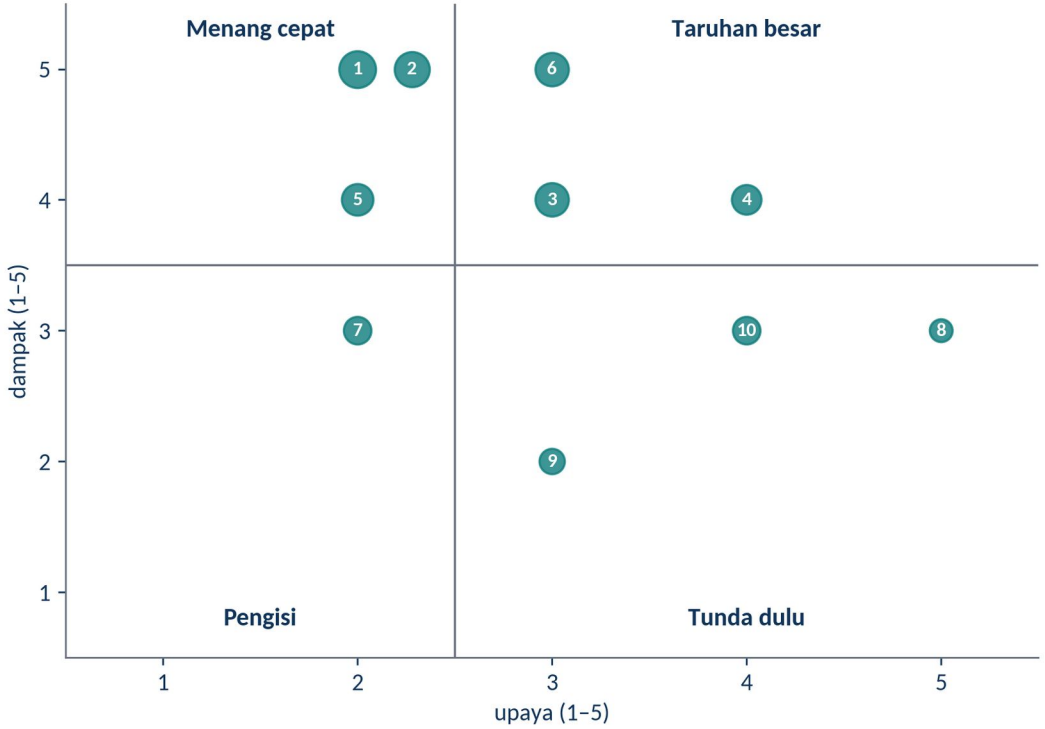
Gagasan Seorang Pengemudi di Sesi Kedua

“Kartu tarif tercetak di pangkalan dan di toko-toko”: dampak 4, upaya 1, lima suara.

quadrant	imp	eff	ratio	votes	idea
Quick win	4	1	4.00	5	Printed fare card at the base and in shops
Quick win	5	2	2.50	9	Fixed fare table by zone shown before booking
Quick win	5	2	2.50	8	Driver-on-the-way message with name and photo

Gagasan itu mendarat di puncak dengan nilai per upaya $4/1 = 4.00$*, dan menjangkau penumpang tanpa ponsel pintar, yang tidak dijangkau tabel tarif sebelum memesan (Soal Latihan 4.3). Buku: Subbab 4.15.*

Matriksnya


Menang cepat
Taruhan besar
Pengisi
Tunda dulu
dampak (1–5)
upaya (1–5)
1
2
3
4
5
6
7
8
9
10

Gambar 4.4 Kesepuluh gagasan dalam matriks dampak-upaya, dinomori sesuai urutan berkas gagasan; ukuran lingkaran menunjukkan suara.

Gambar 4.4: menang cepat di kiri atas, taruhan besar di kanan atas. Buku: Subbab 4.8.

Rekomendasinya

Mulai dengan tiga gagasan menang cepat: tabel tarif, pesan pengemudi sedang menuju, tautan berbagi perjalanan.

Rencanakan antrean yang adil bersama pengemudi; buat prototipe penjemputan dengan foto patokan bersama penumpang lanjut usia.

Jangan membangun aplikasi transportasi lengkap; lampirkan kebutuhan terukur pada setiap langkah.

Siklus berikutnya pada Gambar 4.1 memeriksa apakah rekomendasi itu berhasil. Uji: 3 di Python, 5 pemeriksaan di Java. Buku: Subbab 4.8 dan 4.9.

Lokakarya Sintesis Satu Hari

Pagi: bacakan dan kelompokkan catatan; namai tema; beri tahap dan keluhan; jalankan program; tulis persona dan pertanyaan HMW.

Siang: ideasi tanpa menilai; beri skor dampak dan upaya; kumpulkan suara partisipan; jalankan lagi; pilih prototipe; tulis kebutuhan.

Ganti tahap dan perannya dengan milik Anda sendiri: laundry, puskesmas, layanan kampus. Buku: Subbab 4.10.

Dua Kehati-hatian

Angka-angka itu merangkum **pertimbangan**: diskusikan perbedaan pendapat alih-alih merata-ratakannya begitu saja. Tema dengan jangkauan rendah bisa saja **kritis**, misalnya masalah keselamatan yang disebut satu perawat sif malam.

Angka mengarahkan perhatian; ia tidak menggantikannya. Buku: Subbab 4.10.

IMK dalam Praktik: Profitina

Alur umpan balik yang transparan: kirim, saring, dataset, latih dan ukur Qwen2.5 lokal, rilis — lingkar berpusat pengguna yang terlihat.

Penyederhanaan berdasarkan bukti: paket Agency dihapus pada Agustus 2026, satu langganan Pro 30 hari.

“Aplikasi memberi saran, pengguna yang memutuskan” sebagai kebutuhan: setiap saran AI dapat dinilai dan ditolak.

Buku: Subbab 4.11.

Kesalahan yang Sering Terjadi

Berangkat dari solusi; menganggap permintaan klien sebagai kebutuhan; mewawancarai hanya satu jenis pengguna.
Menghitung catatan alih-alih orang; mengarang persona atau pengguna sintetis.
Kebutuhan yang tidak dapat diuji; menjalankan proses sekali saja; salah menangani data partisipan.

Buku: Subbab 4.12.

Ringkasan

1. Fokus awal pada pengguna, pengukuran empiris, desain iteratif.

2. ISO 9241-210: enam prinsip, empat aktivitas dalam siklus (Gambar 4.1).

3. Double Diamond dan *design thinking* menggambarkan siklus yang sama (Tabel 4.1).

4. Jangkauan (Persamaan (4.1)), keluhan perjalanan (Persamaan (4.2)), nilai per upaya (Persamaan (4.3)).

5. Untuk pangkalan ojek: menunggu (jangkauan 70%) dan harga paling menyakitkan; tiga gagasan menang cepat alih-alih aplikasi lengkap.

Buku: Subbab 4.13.

Pertemuan Berikutnya

Bab 5 — Metode Riset Pengguna. Wawancara, pengamatan, survei, dan studi buku harian: bagaimana catatan di bab ini dikumpulkan, dan kapan berhenti.

Sebelum itu: jalankan `synthesis.py`, tulis persona dari catatan P1, dan kerjakan Soal Latihan 4.2, 4.3, 4.4, dan 4.6.

Buku: Subbab 4.14 dan 4.15.

Kuliah 5 — Metode Riset Pengguna

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 5

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita belajar mengganti tebakan tentang pengguna dengan bukti: memilih metode berdasarkan pertanyaan, merencanakan studi kecil dengan persetujuan yang jujur, mewawancarai tanpa menggiring, menyurvei dengan ketelitian yang diketahui, dan memutuskan kapan wawancara sudah cukup. Buku: Bab 5, Subbab 5.1–5.16.

Cakupan Pertemuan Ini

1. **Masalahnya** — sebuah laundry kehilangan pelanggan tetap dan setiap orang punya tebakan berbeda.
2. **Gagasannya** — apa yang dikatakan versus apa yang dilakukan orang; mengapa versus berapa banyak; empat pertanyaan untuk memilih metode.
3. **Perencanaan** — rencana riset satu halaman, merekrut orang yang sudah pergi, dan persetujuan menurut UU PDP 27/2022 dan PP 33/2026.
4. **Wawancara dan inkuiri kontekstual** — bertanya tentang kejadian spesifik terakhir, dan sikap guru-murid magang.
5. **Survei** — butir yang baik, margin galat, selang kepercayaan, *top-2-box*, dengan Java dan Python.
6. **Analisis kualitatif** — pengodean, diagram afinitas, dan aturan saturasi yang dinyatakan.
7. **AI dalam riset, 2026** — apa yang membantu, dan mengapa pengguna sintetis bukan bukti.

Setiap program hari ini tersedia dalam Java dan Python, dan setiap keluaran yang ditampilkan adalah cetakan program yang sebenarnya. Buku: Bab 5.

Pelanggan Bu Rina yang Menghilang

Laundry Bersih, dekat kampus di Surabaya: pesan lewat WhatsApp, dijemput sepeda motor, pakaian kembali dalam dua hari.

Tahun ini: pesanan turun empat bulan berturut-turut, pelanggan tetap berkurang sekitar sepertiga.

Empat tebakan, empat investasi: aplikasi (keponakan), harga lebih murah (tetangga), motor kedua (kurir), “mereka pulang kampung” (pemilik).

Jika penyebab sebenarnya lain, setiap investasi sia-sia dan pelanggan tetap pergi.

Riset pengguna: cara disiplin untuk *mencari tahu* sebelum membelanjakan uang.

Kasus ini juga mengikuti kita di Bab 6 dan 7: pesan-pesan yang dibuat prototipenya dan diuji berasal dari studi ini. Buku: Subbab 5.1.

Mengapa Penting di Indonesia, 2026

APJII (Mei 2026): penetrasi internet **81,7%**, sekitar 235 juta dari 287 juta jiwa; 84,75% di perkotaan, 78,18% di perdesaan.

StatCounter (Agustus 2026): dari *tayangan halaman* yang dihitungnya dari Indonesia, 60,59% desktop, 38,98% ponsel.

Bukan pertentangan: APJII bertanya kepada **orang**; StatCounter menghitung **tayangan halaman** di situs yang memasang kodenya.

Pelajaran untuk seluruh bab: setiap angka menjawab pertanyaannya sendiri. Ketahui yang mana sebelum mengutipnya.

Peneliti yang mengutip "kebanyakan orang Indonesia berselancar dengan desktop" dari data tayangan halaman telah mencampuradukkan dua pengukuran. Buku: Subbab 5.1.

Apa Itu Riset Pengguna

> **DEFINISI.** *Riset pengguna* adalah kajian sistematis tentang orang yang memakai, atau mungkin memakai, sebuah produk: perilaku, kebutuhan, tujuan, konteks, dan sikap mereka. *Pertanyaan riset* adalah hal yang perlu diketahui tim; *metode* adalah cara mencari tahu; *temuan* adalah apa yang ditunjukkan studi, dinyatakan bersama buktinya.

ISO 9241-210:2019: pahami dan rumuskan **konteks penggunaan** sebelum merancang.

Studi dimulai dari pertanyaan dan keputusan, tidak pernah dari metode. “Ayo survei” bukan rencana.

Inilah berlian pertama siklus design thinking di Bab 4: memperluas pandangan tentang masalah sebelum menyempit ke sebuah rumusan. Buku: Subbab 5.2.1.

Dikatakan versus Dilakukan

Manusia adalah saksi yang buruk atas perilakunya sendiri: “tiap minggu” bisa berarti tiap sepuluh hari dengan jeda di bulan Juli.

Manusia buruk meramal masa depannya: “apakah Anda akan memakai aplikasi dengan pelacakan?” dijawab ya karena sopan.

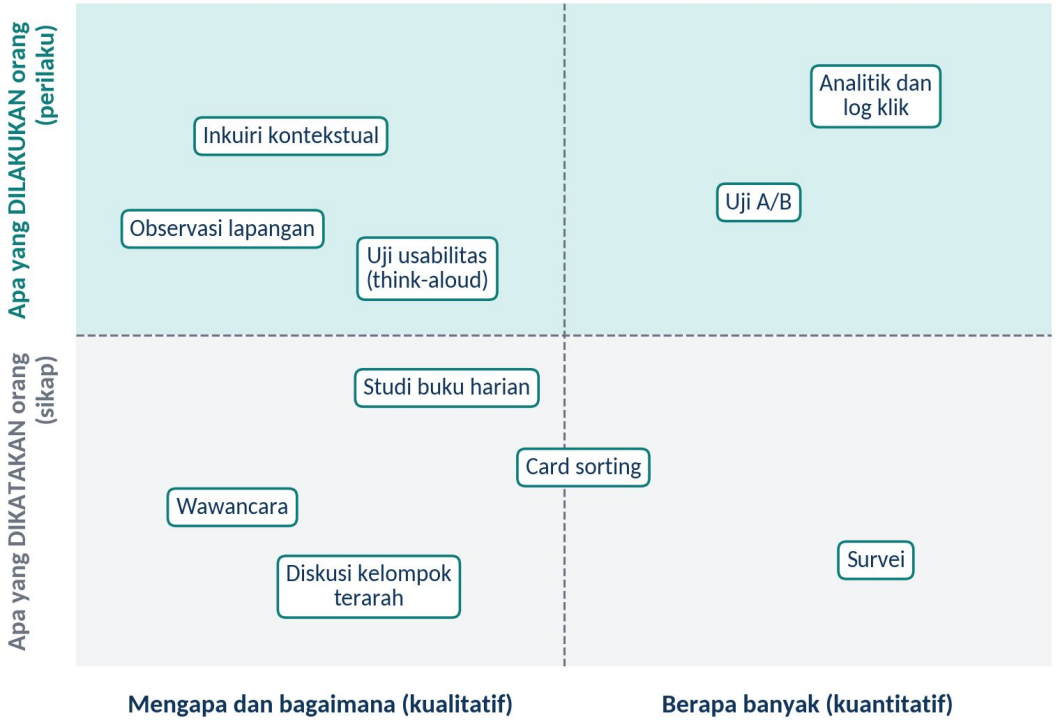
Dua sumbu: **sikap** (apa yang dikatakan) versus **perilaku** (apa yang dilakukan); **kualitatif** (mengapa, bagaimana) versus **kuantitatif** (berapa banyak).

Triangulasi: ketika wawancara, survei, dan buku pesanan sepakat, bertindaklah dengan yakin.

Kedua jawaban di butir pertama bukan kebohongan; ingatan menyimpan kasus yang lazim, bukan kasus sebenarnya. Buku: Subbab 5.2.2.

Lanskap Metode

Gambar 5.1 menempatkan metode-metode yang umum pada kedua sumbu.



Gambar 5.1 Lanskap metode riset pengguna: metode perilaku (atas) mengamati apa yang dilakukan orang, metode sikap (bawah) mencatat apa yang dikatakan orang; metode kualitatif (kiri) menjelaskan mengapa, metode kuantitatif (kanan) menghitung berapa banyak.

Tidak ada kuadran yang lebih baik; studi yang baik memadukan dua. Buku: Subbab 5.2.2 dan Gambar 5.1.

Empat Pertanyaan untuk Memilih Metode

1. **Keputusan apa yang dibantu?** Bu Rina: ke mana uang dibelanjakan dalam tiga bulan ke depan.
2. **Apa yang sudah diketahui, dan dari mana?** Buku pesanan: pelanggan tetap berkurang, tetapi tidak menjelaskan sebabnya.
3. **Mengapa atau berapa banyak?** “Mengapa pelanggan pergi?” bersifat kualitatif.
4. **Apa yang sanggup dilakukan?** Beberapa malam, daftar pelanggan, voucer kecil: selusin wawancara dan survei singkat.

Bentuk lazim studi kecil: **kualitatif dahulu untuk menemukan, kuantitatif kemudian untuk mengukur.**

Riset yang tidak dapat mengubah keputusan tidak layak dilakukan. Buku: Subbab 5.2.3 dan Tabel 5.1.

Rencana Riset Satu Halaman

Keputusan — harga, kapasitas penjemputan, komunikasi, atau aplikasi.

Pertanyaan — mengapa pelanggan tetap berhenti? Apa yang terjadi antara pesan dan kembalinya pakaian?

Metode — 12 wawancara semi-terstruktur, satu pagi inkuiri kontekstual, lalu survei 6 butir.

Partisipan — yang berhenti *dan* yang setia; mahasiswa, pekerja, keluarga, pemilik kos; sekurang-kurangnya dua orang berusia 60+.

Etika dan data — persetujuan sebelum perekaman; kode alih-alih nama; rekaman dihapus setelah diperiksa.

Keluaran — catatan berkode, diagram afinitas, ringkasan survei, lima wawasan, rekomendasi satu halaman.

Nilainya bukan pada dokumen, melainkan pada percakapan yang dipaksanya terjadi sebelum ada yang berbicara dengan pelanggan. Buku: Subbab 5.3.1 dan Tabel 5.2.

Merekrut Orang yang Tepat

Pelanggan yang sudah pergi paling sulit dijangkau dan paling penting didengar.

Screener empat pertanyaan: pesanan terakhir, frekuensi sebelumnya, pekerjaan, kesediaan berbincang 30 menit.

Libatkan kelompok pinggiran: pelanggan lanjut usia, penyandang disabilitas (UU 8/2016 menjamin layanan dan informasi yang aksesibel).

Berapa wawancara? Untuk satu kelompok yang cukup seragam, sebagian besar tema muncul dalam 6–12 wawancara (Guest, Namey, dan Chen, 2020).

Studi yang tidak melibatkan seseorang akan merancang tanpa orang itu. Buku: Subbab 5.3.2.

Persetujuan menurut UU PDP dan PP 33/2026

UU PDP 27/2022: dasar pemrosesan yang sah; riset lazimnya memakai persetujuan eksplisit.

PP 33/2026: persetujuan **bebas, sadar, spesifik, tidak ambigu**; informasi diberikan *sebelum* persetujuan; tidak digabung dengan syarat dan ketentuan; tidak dengan cara menipu.

Penarikan: pemrosesan berhenti dalam **72 jam** setelah permintaan terverifikasi; menolak tidak boleh menurunkan layanan yang tidak membutuhkan data itu.

Kewajiban rincinya berlaku mulai 16 Januari 2027, tetapi studi hari ini sebaiknya sudah bekerja demikian.

Memperlakukan partisipan seperti ini segera menjadi kewajiban hukum dan sejak sekarang memang pantas. Buku: Subbab 5.3.3.

Teks Persetujuan Enam Bagian

1. Siapa yang melakukan studi, dan untuk apa.

2. Apa yang akan terjadi, dan berapa lama.

3. Apa yang direkam, siapa yang melihatnya, kapan dihapus.

4. Sukarela; dapat berhenti kapan saja; tanpa pengaruh pada layanan.

5. Ucapan terima kasih, baik partisipan menyelesaikan maupun tidak.

6. Dua pertanyaan terpisah: “Setuju ikut serta?” dan “Setuju suara Anda direkam?”

> **PERINGATAN:** Jangan mengumpulkan yang tidak dibutuhkan: tanpa alamat, rincian keluarga, atau penghasilan untuk studi pemesanan. Pakai kode partisipan sejak awal.

Ikut serta dan direkam adalah dua persetujuan berbeda; “spesifik” dalam PP 33/2026 berarti keduanya ditanyakan terpisah. Buku: Subbab 5.3.3.

Wawancara Semi-Terstruktur

Percakapan terarah, bukan kuesioner yang dibacakan: topik dan pertanyaan pembuka, lalu ikuti partisipan.

Empat bagian: **pemanasan**, **kisah kali terakhir**, **topik khusus** (penjemputan, harga, menunggu, membayar, masalah), **penutup**.

Jantungnya: *kejadian spesifik yang baru terjadi*. “Biasanya lancar” → “ceritakan saat terakhir yang tidak lancar.”

Peneliti menjadikan partisipan sebagai ahlinya dan jauh lebih banyak mendengar (Portigal).

Hal umum menghasilkan jawaban umum; hari Selasa tertentu menghasilkan detail. Buku: Subbab 5.4.1.

Bertanya Tanpa Menggiring

Tabel 5.1 Jebakan wawancara dan pertanyaan yang lebih baik untuk studi laundry. ::: {custom-style="SlideBullet"} Tabel 5.1 memperlihatkan jebakannya. Tiga kebiasaan: **diam** (hitung sampai lima), **gema** (ulangi kata terakhir), **minta artefaknya** ("boleh saya lihat chat-nya?"). :::

Jebakan	Buruk	Lebih baik
Menggiring	"Jam jemputnya terlalu sempit, kan?"	"Ceritakan bagaimana Anda mengatur penjemputan kemarin."
Ramalan	"Apakah Anda akan memakai aplikasi dengan pelacakan?"	"Apa yang Anda lakukan sambil menunggu kemarin?"
Rancangan	"Fitur apa yang harus ada di aplikasi kami?"	"Apa yang menjengkelkan pada pesanan terakhir?"
Ganda	"Harganya jelas dan wajar?"	"Bagaimana Anda tahu harganya?" lalu "Apa pendapat Anda?"

Riwayat WhatsApp di ponsel partisipan memperlihatkan jam dan kata-kata yang persis, jauh lebih baik daripada ingatan. Buku: Subbab 5.4.2 dan Tabel 5.3.

Inkuiri Kontekstual: Guru dan Murid Magang

Wawancara di tempat pekerjaan terjadi, ketika pekerjaan itu terjadi (Holtzblatt dan Beyer).

Prinsip: **konteks**, **kemitraan**, **interpretasi**, **fokus**.

Satu pagi membonceng Dimas: di tiga rumah kos, kantong dititipkan kepada penjaga yang tidak tahu kapan cucian kembali.

Satu sore di kasir: pesan harga datang 3–4 jam setelah penjemputan; pesan “sudah selesai” kadang setelah pukul 21.00.

Tidak ada yang mengeluh dengan kata-kata, tetapi chat penuh dengan “Bu, sudah selesai belum?”

Observasi menemukan hal yang tidak dapat ditemukan wawancara: masalah yang sudah tidak lagi disadari orang. Buku: Subbab 5.4.3.

Catatan Lapangan dan Studi Buku Harian

Satu pengamatan per baris, masing-masing ditandai **kutipan**, **pengamatan**, atau **tafsiran**.

Memisahkan ketiganya adalah kebiasaan terpenting: tafsiran kelak dipertanyakan; kutipan dan pengamatan adalah buktinya.

Studi buku harian mencatat entri setiap kali sesuatu terjadi, selama berhari-hari atau berminggu-minggu.

Pemilik kos memesan setiap Senin tetapi memutuskannya Minggu malam: pengingat pada Minggu malam mungkin disambut baik.

Wawancara satu jam memadatkan irama sehari-hari; studi buku harian memulihkannya. Buku: Subbab 5.4.4.

Survei: Kapan dan Bagaimana

Setelah wawancara, untuk mengukur *seberapa luas* setiap calon masalah.

Butir ditulis dengan kata-kata pelanggan sendiri dari wawancara.

Aturan: satu gagasan per butir; konkret dan baru terjadi; kata-kata netral (hindari bias persetujuan); skala dan arah yang sama; sedikit butir; uji coba dengan tiga orang.

Survei laundry: enam butir Likert (1–5), ditambah kelompok usia dan saluran.

Survei yang ditulis sebelum wawancara mengukur asumsi tim. Buku: Subbab 5.5.1 dan 5.5.2.

Seberapa Teliti Sebuah Survei?

Untuk proporsi p dari n jawaban, margin galat 95% (Persamaan (5.1)):

$$E = z\sqrt{\frac{p(1-p)}{n}} \tag{5.1}$$

Kasus terburuk $p = 0.5$. Untuk $n = 60$: $1.96 \times \sqrt{0.25/60} = 0.127$, **±12,7 poin**. Untuk $n = 400$: ±4,9 poin.

Sampel yang dibutuhkan: $n = z^2\,p(1-p)/E^2$; untuk $E = 0.10$, 97 jawaban.

Sampel empat kali lipat membuat margin menjadi separuhnya, karena n berada di bawah akar kuadrat. Buku: Subbab 5.5.3.

Selang Kepercayaan Rata-rata

Untuk butir Likert dengan rata-rata $\acute{x}$, simpangan baku s, dan n jawaban (Persamaan (5.2)):

$$\acute{x} \pm t_{0.975,\,n-1} \cdot \frac{s}{\sqrt{n}} \tag{5.2}$$

Program membutuhkan t tanpa tabel; deret di sekitar z teliti mulai 5 derajat bebas (Persamaan (5.3)):

$$t \approx z + \frac{z^3 + z}{4\,v} + \frac{5\,z^5 + 16\,z^3 + 3\,z}{96\,v^2} \tag{5.3}$$

Untuk $v = 59$: $1.960 + 0.040 + 0.001 = 2.001$, sama dengan tabel cetak.

***Top-2-box*:** bagian jawaban 4 atau 5 — mudah dijelaskan, dan merupakan proporsi, sehingga Persamaan (5.1) memberi margin galatnya.

Simpangan baku sampel dibagi $n - 1$ karena enam puluh jawaban mewakili populasi yang lebih besar. Buku: Subbab 5.5.3.

Perangkum Survei

`survey_stats.py` / `SurveyStats.java`: membaca `survey.csv` dan `items.csv`; mencetak rata-rata, selang, *top-2-box*, rincian Q2 menurut usia, dan margin galat.

Java memasang `Locale.US` agar kedua program mencetak titik desimal di mana pun.

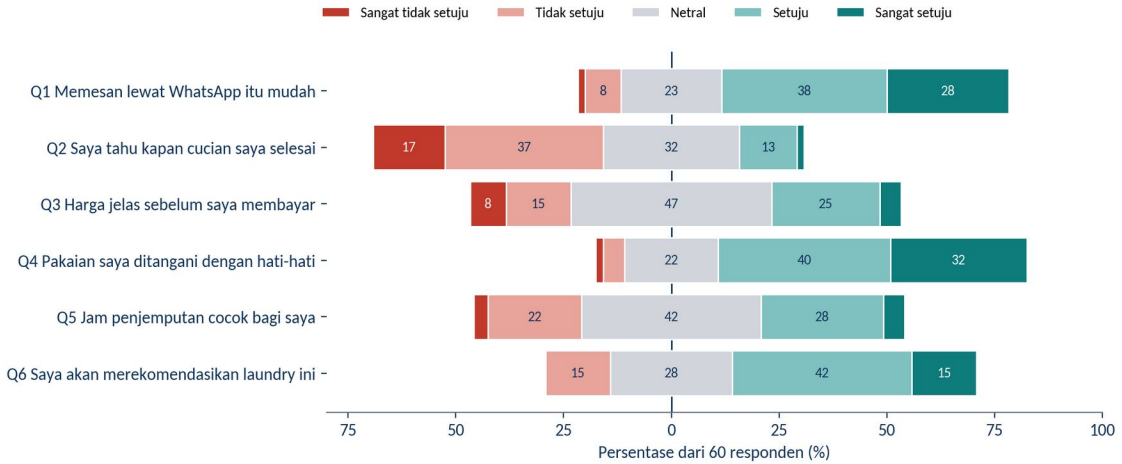
Keluaran:

```
60 respondents, 1 = strongly disagree ... 5 = strongly agree
item   mean         95% CI  top-2  statement
Q1     3.83   [3.58, 4.09]    67%  Ordering by WhatsApp is easy
Q2     2.47   [2.21, 2.72]    15%  I know when my laundry will be ready
Q3     3.03   [2.78, 3.28]    30%  The price is clear before I pay
Q4     3.95   [3.71, 4.19]    72%  I trust my clothes are handled with care
Q5     3.10   [2.86, 3.34]    33%  The pickup times suit me
Q6     3.57   [3.33, 3.81]    57%  I would recommend the laundry to a friend
```

Teks butir dibaca dari berkas, sehingga program yang sama merangkum survei berikutnya tanpa diubah. Buku: Subbab 5.6.2–5.6.4.

Menggambar Data Likert

Gambar 5.2: diagram batang bertumpuk divergen — ketidaksetujuan ke kiri, persetujuan ke kanan.



Gambar 5.2 Enam butir survei sebagai diagram batang bertumpuk divergen (60 responden); butir Q2 paling condong ke sisi tidak setuju.

Inilah gambaran tunggal yang paling jelas untuk data Likert. Buku: Subbab 5.6.4 dan Gambar 5.2.

Membaca Survei

Saluran dan layanan inti baik: Q1 3,83 (dua pertiga setuju), Q4 3,95 (72%).

Butir terlemah jauh: **Q2 “Saya tahu kapan cucian saya selesai”**, 2,47, hanya 15% setuju; selang [2,21; 2,72] seluruhnya di bawah 3.

Rincian usia: 45+ pada 2,14 ± 0,63, 18–29 pada 2,68 ± 0,42 — selangnya bertumpang tindih: **petunjuk**, bukan bukti.

±12,7 poin pada $n = 60$: “15% setuju” berarti “beberapa persen sampai sekitar seperempat”.

> **WAWASAN KUNCI:** Wawancara memberi tahu *apa* dan *mengapa*; survei memberi tahu *seberapa luas*. Salah satunya saja tidak cukup.

Laporkan petunjuk sebagai petunjuk; wawancara berikutnya dapat mengujinya. Buku: Subbab 5.6.5.

Dari Catatan ke Kode dan Tema

Pengodean: label pendek pada bagian catatan, misalnya `ready-time-unknown`, `reply-slow-evening`; buku kode tumbuh selama analisis.

Analisis tematik (Braun dan Clarke): mengakrabi data, mengodekan, mencari tema, meninjau, mendefinisikan dan menamai, menuliskan dengan kutipan.

Diagram afinitas: satu gagasan per kertas tempel, kelompokkan tanpa bicara, namai setiap kelompok dengan kalimat dari sudut pandang pelanggan.

"Masalah komunikasi" hanyalah label; "Saya tidak pernah tahu kapan pakaian saya kembali" adalah temuan yang dapat dirancangkan solusinya.

Tema yang didukung hanya satu partisipan adalah opini, bukan temuan. Buku: Subbab 5.7.1 dan 5.7.2.

Kapan Wawancara Sudah Cukup?

Guest, Namey, dan Chen (2020): ukuran dasar $b = 4$, panjang deret $r = 2$, ambang $\tau = 5\ \%$.

B = kode berbeda dalam dasar; N_{run} = kode yang pertama kali muncul dalam deret; saturasi pada deret pertama yang memenuhi (Persamaan (5.4)):

$$\frac{N_{\text{run}}}{B} \leq \tau \tag{5.4}$$

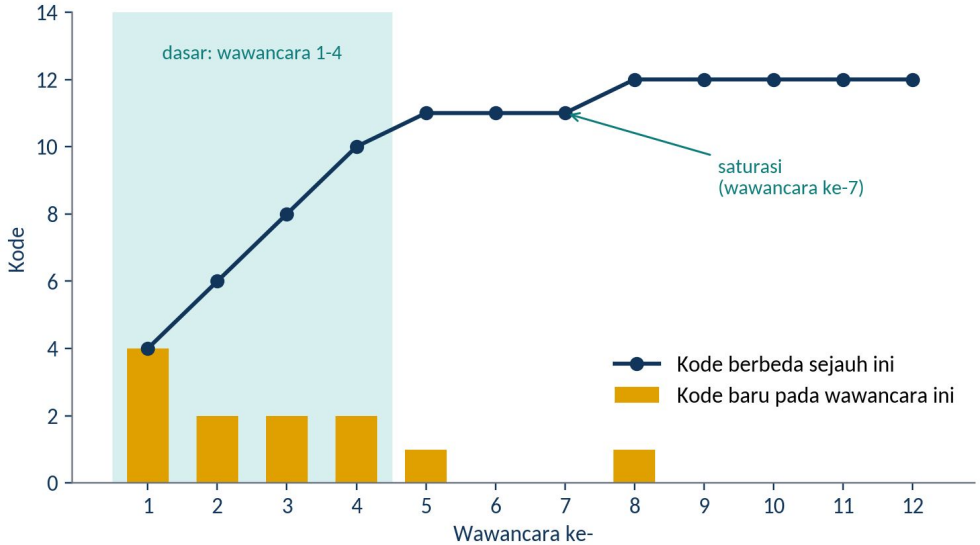
Studi laundry: dasar 10 kode; deret 5–6 menambah 1 (10%); deret 6–7 menambah 0 → **saturasi pada wawancara 7**.

Aturan ini membuat gagasan yang kabur menjadi tepat dan dapat dilaporkan. Buku: Subbab 5.7.3.

Program Saturasi dan Keluarannya

`saturation.py` / `Saturation.java`: frekuensi kode, kode baru per wawancara, dan aturannya.

```
Base: 10 codes in interviews 1-4. Saturation reached at interview 7 (run of 2 with <= 5% new codes).
```



Gambar 5.3 Kode baru per wawancara (batang) dan kode berbeda yang ditemukan sejauh ini (garis); aturan menyatakan saturasi pada wawancara 7.

Gambar 5.3: kurva naik pada dasar lalu mendatar. Kaus kaki hilang di wawancara 8 datang *setelah* saturasi.

Saturasi berarti tema-tema umum sudah ditemukan, bukan bahwa setiap cerita sudah terdengar. Buku: Subbab 5.7.4 dan Gambar 5.3.

Hitungan dalam Data Kualitatif

`ready-time-unknown` dalam 11 dari 12 wawancara: bukti kuat masalah itu umum *di antara jenis orang yang diwawancarai*.

Itu **bukan** persentase semua pelanggan; untuk itulah survei ada.

Masalah langka membutuhkan lebih banyak partisipan dan aturan berhenti yang lain (misalnya studi keselamatan).

Biarkan kedua jenis bukti tetap di jalurnya masing-masing. Buku: Subbab 5.7.4.

Menguji Program Analisis

Rumus yang salah menghasilkan angka yang tampak sangat masuk akal — karena itu uji.

`test_research.py` / `ResearchTest.java`: $t(59) \approx 2.001$, $t(9) \approx 2.262$; rata-rata dan *top-2-box* hitungan tangan; margin pada $n = 400$ sebesar 4,9; saturasi pada 7 dengan dasar 10; tidak pernah saturasi bila setiap wawancara baru.

```
7 checks passed
```

Pengujian Java tidak memakai pustaka, sehingga berjalan hanya dengan JDK. Buku: Subbab 5.8.

Triangulasi dan Wawasan

Buku pesanan (perilaku, kuantitatif) + wawancara dan observasi (kualitatif) + survei (sikap, kuantitatif) sama-sama menunjuk ke **waktu selesai yang tidak diketahui**.

Tebakan-tebakan: pesaing muncul dua kali; libur menjelaskan sebagian penurunan mahasiswa; kunjungan larut malam hanyalah gejala; **tidak ada yang meminta aplikasi**.

Bentuk wawasan: *[siapa] membutuhkan [apa] karena [mengapa], tetapi saat ini [hambatannya]*.

Contoh: pelanggan perlu tahu kapan cucian kembali *saat menyerahkannya*, karena mereka merencanakan pakaian seminggu, tetapi saat ini baru tahu ketika Dimas hendak berangkat.

Setiap wawasan menjadi pertanyaan "Bagaimana kita bisa…?" untuk pencarian gagasan di Bab 4. Buku: Subbab 5.9.1 dan 5.9.2.

Apa yang Dilakukan Bu Rina

Bukan aplikasi — empat perubahan pada layanan WhatsApp:

pesan penjemputan berisi perkiraan rentang harga, janji waktu selesai, dan jumlah potong;

pesan “sudah selesai” paling lambat pukul 16.00 dengan dua pilihan jam antar, salah satunya setelah 18.00;

foto cucian terlipat beserta jumlahnya;

daftar harga berhuruf besar yang dikirim sebagai teks, dengan pesan suara dipersilakan.

Riset tidak memberi tahu apa yang harus dibangun; riset memberi tahu masalah mana yang harus diselesaikan. Bab 6 dan 7 membuat prototipe dan menguji pesan-pesan ini. Buku: Subbab 5.9.3.

AI dalam Riset Pengguna, 2026

Membantu: transkripsi (periksa nama dan angka), pengodean tahap awal, pengelompokan jawaban teks bebas — manusia yang memutuskan.

Studi CHI 2026 tentang model sumber terbuka lokal untuk pengodean: dihargai karena cepat dan mampu mengekstrak makna permukaan, kurang dipercaya untuk makna tersirat; pengguna meminta tautan ke bagian sumber dan buku kode yang dapat dikunci.

Merugikan: "pengguna sintetis". NN/g (2024): berguna untuk riset meja atau menyusun panduan, tidak pernah untuk menggantikan partisipan sungguhan atau memvalidasi rancangan — terlalu mengiyakan, tanpa prioritas, tanpa pengalaman nyata.

Privasi: mengunggah rekaman ke AI awan berarti pemroses lain menurut UU PDP dan PP 33/2026; sebutkan dalam persetujuan, utamakan alat lokal.

> **PERINGATAN:** Ringkasan AI adalah tafsiran, bukan kutipan. Jangan pernah menaruhnya dalam tanda kutip.

Tanyakan kepada pelanggan sintetis tentang aplikasi dengan pelacakan dan ia akan menjawab ya — persis jawaban yang akan menyesatkan Bu Rina. Buku: Subbab 5.10.

Menerapkannya dalam Kenyataan

Antrean puskesmas: amati ruang tunggu pukul 07.00; jangan bertanya tentang penyakit (data kesehatan adalah data pribadi *spesifik*); arahkan keluhan ke perawat atau dokter.

Pemesanan warung: observasi jam makan siang, lima wawancara, hitungan dari riwayat chat dengan persetujuan pemilik.

Pangkalan ojek: wawancara saat menunggu, 15 menit, bayar waktunya.

Daftar periksa: keputusan dan pertanyaan tertulis; metode mengikuti pertanyaan; orang yang pergi dan kelompok pinggiran; persetujuan terpisah; pertanyaan tentang kejadian terakhir; survei setelah wawancara; aturan berhenti dan margin dinyatakan; setiap temuan menunjuk ke buktinya.

Bentuk studi yang sama cocok untuk sebagian besar layanan kecil. Buku: Subbab 5.11.

IMK dalam Praktik: Profitina

Skor yang harus terbaca: tanyakan kepada pemilik merek “ceritakan saat terakhir Anda melihat skor; apa yang Anda lakukan sesudahnya?”, bukan “apakah Anda suka dasbornya?”.

Persetujuan yang dipahami: Profitina menerapkan hak swalayan UU PDP; wawancara singkat setelah pendaftaran (“apa yang barusan Anda setujui?”) menguji pemahaman lebih baik daripada hitungan klik.

Alur umpan balik sebagai riset berkelanjutan: kirim → kurasi → *dataset* → latih dan ukur Qwen2.5 lokal → rilis; kodekan umpan baliknya seperti catatan wawancara.

Pemrosesan AI lokal menghindarkan pengiriman data itu ke pihak ketiga.

Profitina memberi saran dan penggunalah yang memutuskan; riset adalah cara tim mengetahui apakah saran itu dipahami. Buku: Subbab 5.12.

Kesalahan yang Sering Terjadi

Memulai dari metode, bukan pertanyaan.

Meminta orang meramal atau merancang.

Hanya berbicara dengan pelanggan yang puas.

Pertanyaan menggiring; mencampur kutipan, pengamatan, dan tafsiran.

Menulis survei sebelum wawancara.

Persentase tanpa margin; hitungan wawancara dibaca sebagai persentase populasi.

Saturasi tanpa aturan; pengguna sintetis sebagai bukti; satu persetujuan untuk segalanya.

Setiap kesalahan punya perbaikan satu baris di dalam bab. Buku: Subbab 5.13.

Ringkasan

1. Riset dimulai dari keputusan dan pertanyaan; metode mengikutinya (Gambar 5.1).

2. Dikatakan versus dilakukan, mengapa versus berapa banyak; lakukan triangulasi.

3. Rencanakan dalam satu halaman; rekrut orang yang pergi dan kelompok pinggiran; persetujuan yang diinformasikan, spesifik, tidak digabung, dapat ditarik (UU PDP, PP 33/2026).

4. Wawancarai tentang kejadian spesifik terakhir; amati dalam konteks.

5. Survei: Persamaan (5.1), Persamaan (5.2), Persamaan (5.3); Q2 pada 2,47 dengan persetujuan 15%.

6. Kodekan, temukan tema, dan berhenti pada saturasi dengan aturan yang dinyatakan (Persamaan (5.4)).

7. AI membantu; manusia memutuskan; pengguna sintetis bukan bukti.

Masalah sebenarnya laundry itu adalah waktu selesai yang tidak diketahui; tidak ada yang meminta aplikasi. Buku: Subbab 5.14.

Pertemuan Berikutnya

Bab 6 — Prototipe: dari kertas hingga prototipe interaktif dan berbantuan AI.

Kios antrean puskesmas: sesi kertas, pemeriksa alur dalam Java dan Python, dan prototipe berkode yang dilayani Flask dan server HTTP JDK.

Sebelum itu: kerjakan Soal Latihan 5.2, 5.3, 5.4, dan 5.6, lalu jalankan `survey_stats` dan `saturation` dalam kedua bahasa.

Pesan-pesan laundry yang direkomendasikan hari ini dibuat prototipenya sebagai Wizard of Oz di Subbab 6.9. Buku: Subbab 5.15 dan 5.16.

Kuliah 6 — Prototipe: dari Kertas hingga Interaktif (dan Berbantuan AI)

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 6

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini prototipe menjadi pertanyaan yang dapat kita pegang: kita memilih fidelitasnya berdasarkan pertanyaan, menguji layar kertas dengan "komputer" manusia, memeriksa alur dengan program, membangun kios yang dapat diklik dalam Java dan Python, dan melihat apa yang ditambahkan sekaligus disembunyikan alat AI tahun 2026. Buku: Bab 6, Subbab 6.1–6.15.

Cakupan Pertemuan Ini

1. **Masalahnya** — antrean puskesmas sebelum fajar dan brosur kios dari vendor.
2. **Gagasannya** — prototipe menjawab pertanyaan: peran, tampilan dan rasa, atau implementasi; fidelitas; horizontal dan vertikal.
3. **Prototipe kertas dan Wizard of Oz** — fasilitator, komputer yang diam, penyihir yang tersembunyi.
4. **Memeriksa alur** — penelusuran melebar menemukan layar tak terjangkau, jalan buntu, dan ketukan paling sedikit.
5. **Prototipe berkode** — enam layar, nomor antrean sungguhan, log, dilayani Flask dan server HTTP JDK; tangkapan layar dengan Playwright.
6. **Prototipe berbantuan AI** — Figma Make, Google Stitch, dan enam aturan.

Pesan-pesan laundry dari Bab 5 juga dibuat prototipenya, tanpa satu layar pun. Buku: Bab 6.

Antrean Sebelum Fajar

Sebuah puskesmas di Jawa Timur: pasien mengantre sejak pukul enam lewat seperempat; pasien lanjut usia paling awal karena nomor poli umum habis menjelang siang.

Rencana dr. Wulan: kios layar sentuh; brosur vendor menampilkan login NIK, pertanyaan BPJS, sebelas layanan berhuruf kecil, dan konfirmasi.

Tim mahasiswa meminta waktu seminggu dan memakai kertas, gunting, spidol, dan kardus.

Mereka menemukan tiga masalah yang tak mungkin diperlihatkan brosur: “BPJS atau Umum?” dibaca sebagai pertanyaan poli; Bantuan tanpa jalan kembali; jam poli tak terjangkau.

Ketiganya mudah diperbaiki setelah terlihat dan mahal setelah kios terpasang. Buku: Subbab 6.1.

Prototipe adalah Pertanyaan

> **DEFINISI.** *Prototipe* adalah representasi apa pun dari sebuah rancangan, dibuat sebelum produk jadi, yang memungkinkan orang merasakan atau menilai suatu aspeknya. *Fidelitas* adalah seberapa dekat ia menyerupai produk jadi dalam tampilan, interaksi, dan isi.

Houde dan Hill: tanyakan *apa yang diprototipekan oleh prototipe itu*.

Peran — apa yang dilakukannya dalam kehidupan seseorang (apakah kios memperpendek penantian subuh?).

Tampilan dan rasa — bagaimana rasanya dilihat dan disentuh (apakah tombol cukup besar untuk jari yang gemetar?).

Implementasi — dapatkah dibangun (dapatkah peramban mengendalikan printer termal?).

Kios kardus menjawab peran dan banyak bagian tampilan dan rasa, tidak menjawab implementasi. Buku: Subbab 6.2.1.

Sketsa atau Prototipe? Hal yang Tepat atau Hal yang Benar?

Buxton: **sketsa** cepat, murah, banyak, sekali pakai — *apakah kita merancang hal yang tepat?*

Prototipe halus, lebih sedikit, lebih mahal — *apakah kita merancang hal itu dengan benar?*

Orang mengomentari warna ketika melihat hasil yang rapi, dan mengomentari gagasan ketika melihat sketsa pensil.

> **WAWASAN KUNCI:** Sesuaikan fidelitas dengan pertanyaannya, bukan dengan alat Anda. Pertanyaan awal layak dijawab prototipe kasar; pertanyaan akhir layak dijawab prototipe halus.

Langsung melompat ke hasil yang rapi menjawab pertanyaan kedua sebelum pertanyaan pertama. Buku: Subbab 6.2.2.

Spektrum Fidelitas

Gambar 6.1 menyusun prototipe dari pekerjaan beberapa menit sampai beberapa hari.



Gambar 6.1 Spektrum fidelitas: dari sketsa yang bertanya apakah ini hal yang tepat sampai prototipe berkode yang bertanya apakah hal itu sudah benar; alat berbantuan AI mencapai fidelitas tinggi dengan cepat, sehingga pertanyaan di sisi kiri mudah terlewat.

Biaya membuat dan mengubah naik dari kiri ke kanan. Buku: Subbab 6.2.2 dan Gambar 6.1.

Horizontal, Vertikal, dan Bentuk T

Horizontal: banyak fitur, masing-masing hanya permukaan (setiap layar kios ada; “Cetak” hanya menampilkan gambar).

Vertikal: satu jalur secara mendalam (jalur poli gigi sungguh menerbitkan nomor dan memperkirakan waktu tunggu).

Bentuk T: lebar dan dangkal di mana-mana, dalam di sepanjang satu jalur yang paling penting.

Kios berkode pertemuan ini berbentuk T.

Tentukan jalur mana yang layak didalami sebelum menulis kode apa pun. Buku: Subbab 6.2.3.

Prototipe Kertas: Bahan dan Peran

Bahan: kertas A5 atau kartu, kertas tempel untuk jendela sembul, spidol, gunting, selotip yang mudah dilepas, bingkai perangkat (muka kios dari kardus).

Fasilitator: menyambut, menjelaskan bahwa yang diuji rancangannya, bukan orangnya; membacakan tugas; meminta berpikir dengan bersuara.

Komputer: tidak pernah berbicara; menukar layar persis sesuai rancangan, termasuk untuk sentuhan “salah”; kartu kosong bertanda “?” untuk layar yang belum ada.

Pengamat: satu pengamatan per baris, seperti di Bab 5.

Kartu kosong itu sendiri sebuah temuan: partisipan mengharapkan sesuatu yang tidak ada di rancangan. Buku: Subbab 6.3.1.

Sebuah Sesi di Puskesmas

Tugas: ambil nomor poli umum; cari jam buka poli gigi hari ini; pulih setelah tak sengaja menekan Bantuan.

Delapan partisipan selama dua pagi: lima pasien lanjut usia, dua ibu, satu pekerja; persetujuan seperti di Bab 5.

Ketiga masalah besar muncul dalam empat sesi.

Temuan lebih kecil: judul disentuh seolah-olah tombol; daftar yang tidak dapat digulir; “berapa lama saya menunggu?” ditanyakan sebelum hal lain.

Tidak satu pun membutuhkan sebaris kode. Buku: Subbab 6.3.2.

Wizard of Oz

Seorang manusia tersembunyi menghasilkan tanggapan sistem secara manual; partisipan mengira, atau setuju berpura-pura, bahwa sistemlah yang menghasilkannya.

Kelley (1984) merancang aplikasi perkantoran berbahasa alami dengan cara ini sebelum kemampuan memahami bahasanya ada.

Di puskesmas: pasien berkata “saya mau ke poli gigi”; mahasiswa di balik kardus memasang layar konfirmasi.

Temuan: konfirmasi harus *diucapkan*, karena pasien yang tidak dapat membaca layar juga tidak dapat membaca konfirmasinya.

Wizard of Oz juga cara membuat prototipe AI assistant sebelum membangunnya (Bab 13). Buku: Subbab 6.3.3.

Alur adalah Graf

Layar adalah simpul, tombol adalah panah.

Tak terjangkau: tanpa jalan masuk. **Jalan buntu:** tanpa jalan keluar (kecuali tujuan). **Ketukan menuju tujuan:** panjang lintasan terpendek.

Format: `screen -> next : label`, ditambah baris `start` dan `goal` (Kode 6.1).

Kode 6.1 Cuplikan `flow_v1.txt`: alur kios pada uji kertas, satu baris per tombol.

```
start welcome
goal ticket
welcome -> service : Ambil antrean / Take a number
welcome -> help : Bantuan / Help
service -> bpjs : Umum / General
bpjs -> nik : Pakai BPJS / Use BPJS
bpjs -> confirm : Bayar sendiri / Pay myself
nik -> confirm : Lanjut / Next
confirm -> ticket : Cetak / Print
schedule -> service : Kembali / Back
```

Tidak ada yang menggambar graf dengan sempurna di atas kertas; tuliskan alurnya agar dapat diperiksa program. Buku: Subbab 6.4.1.

Penelusuran Melebar

Kunjungi layar menurut jaraknya: awal, lalu semua yang berjarak satu ketukan, lalu dua ketukan, dengan antrean.

Saat pertama kali sebuah layar dicapai, lintasan terpendek ke sana sudah ditemukan.

Setelah penelusuran: tak pernah dicapai = tak terjangkau; tanpa tombol keluar dan bukan tujuan = jalan buntu.

`flow_check.py` memakai `collections.deque`; `FlowCheck.java` memakai `ArrayDeque` dan `TreeMap` agar keduanya mencetak dalam urutan yang sama.

Program memeriksa struktur; manusia memeriksa makna. Keduanya dibutuhkan. Buku: Subbab 6.4.2.

Menjalankan Pemeriksa

Versi 1 (uji kertas):

```
../data/flow_v1.txt: 8 screens, 9 buttons
  goal 'ticket': 4 taps  welcome > service > bpjs > confirm > ticket
  unreachable screens: schedule
  dead ends (no way on or back): help
```

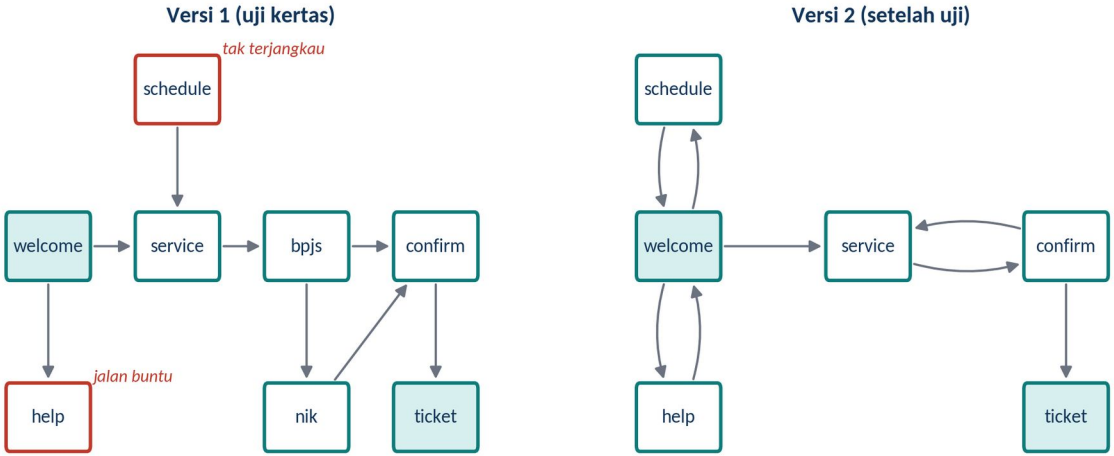
Versi 2 (BPJS dipindah ke meja depan, Bantuan dan jam poli dapat dicapai dengan Kembali, KIA dan Lansia ditambahkan):

```
../data/flow_v2.txt: 6 screens, 11 buttons
  goal 'ticket': 3 taps  welcome > service > confirm > ticket
  unreachable screens: none
  dead ends (no way on or back): none
```

Pemeriksa menemukan dua dari tiga masalah; pertanyaan BPJS yang disalahpahami adalah masalah makna yang hanya ditemukan manusia. Buku: Subbab 6.4.3.

Sebelum dan Sesudah

Gambar 6.2 menggambar kedua alur.



Gambar 6.2 Alur kios sebelum dan sesudah uji kertas: versi 1 memiliki layar tak terjangkau dan jalan buntu serta membutuhkan empat atau lima ketukan menuju tiket; versi 2 membutuhkan tiga, dan setiap layar mempunyai jalan kembali.

Empat atau lima ketukan menjadi tiga, dan setiap layar punya jalan maju atau kembali. Buku: Subbab 6.4.3 dan Gambar 6.2.

Wireframe dan Mockup yang Dapat Diklik

Wireframe: struktur tanpa gaya — satu pertanyaan per layar, tombol utama selebar layar ≥ 72 px, huruf dan nama layanan pada setiap tombol, pita kuning “PROTOTIPE”.

Mockup yang dapat diklik pada 2026: Figma (Config 2025: Figma Make, Sites, Draw, Buzz, Grid; Config 2026: Figma Motion, *code layers*, pembaruan agen).

Penpot: sumber terbuka, berbasis peramban, dapat dipasang sendiri — berguna bagi lembaga publik.

Mockup memperlihatkan alur, bukan bahwa nomor diterbitkan dengan benar atau bahwa dua pasien tidak pernah mendapat nomor yang sama.

Untuk perilaku dan implementasi, bangunlah prototipe berkode. Buku: Subbab 6.5.

Apa yang Harus Dilakukan Prototipe Berkode

Enam layar `flow_v2.txt` dalam bahasa Indonesia atau Inggris (`?lang=id|en`).

Nomor antrean sungguhan yang terus naik per layanan; perkiraan tunggu = orang di depan × menit per pasien.

Tombol setinggi ≥ 72 px (minimum SC 2.5.8 WCAG 2.2 adalah 24 × 24 px CSS), teks gelap di atas putih, garis fokus yang terlihat.

Pita “PROTOTIPE” di setiap layar; setiap tampilan layar dicatat dengan cap waktu.

Prototipe ini **tidak** mencetak, tidak menyentuh rekam medis, dan tidak menyimpan apa pun tentang orang yang menekan.

Jumlah orang yang menunggu adalah nilai awal rekaan; berpura-pura sebaliknya berarti menipu partisipan. Buku: Subbab 6.6.1.

Layar-layarnya: Satu Fungsi

`render(screen, lang, code, number)` mengembalikan halaman HTML lengkap; `kiosk_pages.py` dan `KioskPages.java` menghasilkan halaman yang identik byte demi byte.

Teks kedua bahasa dalam satu kamus: terjemahan adalah perubahan data, bukan kode.

Label melewati `escape`; halaman menyatakan `<html lang="...">` untuk pembaca layar.

Pengujian dapat memeriksa halaman tanpa menyalakan server.

Memisahkan layar dari server membuat keduanya dapat diuji dan dibandingkan. Buku: Subbab 6.6.2.

Dua Server

Flask (`kiosk_proto.py`, porta 4060): satu fungsi `show` kecil per rute; `lambda s=s` menangkap setiap nama layar.

Server HTTP JDK (`KioskProto.java`, porta 4061): satu penangan untuk setiap jalur, 404 untuk jalur yang tak dikenal, penguraian *query*.

Penghitung nomor dan log bersifat `synchronized` di Java: dua pasien tidak boleh menerima nomor yang sama.

```
 * Running on http://127.0.0.1:4060
```

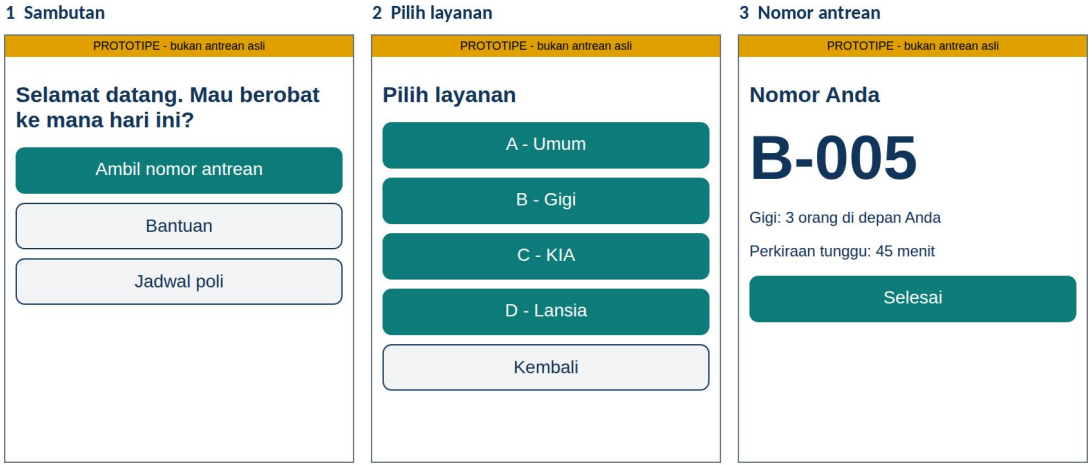
`com.sun.net.httpserver` *menjadi bagian JDK sejak Java 6 dan tidak membutuhkan pustaka. Buku: Subbab 6.6.3 dan 6.6.4.*

Tangkapan Layar dengan Playwright

`shoot.py` mengendalikan Chromium sungguhan: membuka layar sambutan, mengambil nomor poli gigi, menyimpan tangkapan layar pada setiap langkah.

```
title: Health Center Queue
ticket: B-004
saved 3 screenshots in ../screens/
```

Terhadap server Java, tiga baris yang sama muncul.



Gambar 6.3 Tiga layar prototipe kios yang sedang berjalan, ditangkap oleh `shoot.py`: layar sambutan, pilihan layanan, dan tiket dengan banyaknya orang di depan serta perkiraan waktu tunggu.

Poli gigi memiliki tiga orang yang menunggu, sehingga tiket pertama B-004 dan waktu tunggunya 3 × 15 = 45 menit. Buku: Subbab 6.6.5 dan Gambar 6.3.

Log dan Pengujian

Setiap tampilan menambahkan `time,lang,screen` ke `data/events.csv`; selisih waktunya adalah lama di sebuah layar (Bab 7 memakai log semacam ini).

`test_proto.py`: kedua alur; atribut `lang` dan lima tombol di halaman layanan; melalui *test client* Flask, B-004 lalu B-005 dan dua baris log (log dialihkan ke berkas sementara).

`ProtoTest.java`: pemeriksaan yang sama tanpa server.

```
8 checks passed
```

Prototipe sekali pakai pun tidak boleh menyesatkan. Buku: Subbab 6.7.

Prototipe Berbantuan AI pada 2026

Figma Make (Mei 2025): *prompt* atau rancangan → prototipe atau aplikasi yang bekerja.

Google Stitch (Mei 2025): teks dan gambar, termasuk sketsa → rancangan UI dan kode *front-end*, dapat diekspor ke Figma, dengan model Gemini.

Prompt tim menghasilkan empat layar menarik dalam kurang dari semenit — dengan login NIK, pertanyaan BPJS, *carousel*, dan tautan Bantuan kecil: setiap masalah yang baru saja dihilangkan uji kertas.

Alat hanya dapat mengikuti pengetahuan yang diberikan kepadanya; temuannya ada di catatan pengamat. Buku: Subbab 6.8.1.

Enam Aturan Prototipe AI

1. Buat sketsa dan uji lebih dahulu; libatkan AI setelah alurnya mantap.

2. Tuliskan temuan riset di dalam *prompt*.

3. Periksa layar bangkitan terhadap berkas alur dan daftar periksa aksesibilitas.

4. Perlakukan kode bangkitan sebagai prototipe: perlu tinjauan dan pengujian.

5. Perhatikan apa yang diunggah: tanpa data pasien sungguhan; utamakan alat lokal atau yang dipasang sendiri.

6. Uji dengan manusia, seperti versi kertas.

> **PERINGATAN:** Prototipe rapi yang dibuat dalam semenit tampak sudah jadi dan mengundang persetujuan yang sopan. Tampilkan prototipe AI tahap awal dalam skala abu-abu atau di atas kertas.

Kecepatan di ujung kanan spektrum membuat ujung kiri mudah terlewat. Buku: Subbab 6.8.2.

Membuat Prototipe Pesan: Laundry

Rekomendasi Bab 5 berupa *pesan*, sehingga prototipenya kartu-kartu pesan yang diperankan lewat WhatsApp: Wizard of Oz untuk layanan pesan.

Temuan: rentang harga butuh satu baris tentang apa yang membuatnya lebih mahal; dua pilihan jam antar perlu diberi nomor (“balas 1 atau 2”).

Dua minggu dikirim manual dari templat WhatsApp Business; otomasikan hanya setelah kata-katanya mantap.

Buat prototipe kata-katanya, lalu perilakunya, baru otomasinya. Buku: Subbab 6.9.

Menerapkannya dalam Kenyataan

Sebuah kartu: pertanyaannya; fidelitas terendah yang menjawabnya; berkas alur yang diperiksa setiap putaran; 3–5 tugas dengan akhir yang jelas; 5–8 partisipan termasuk kelompok pinggiran; kejujuran (label, persetujuan, tanpa data sungguhan); iterasi.

ATM: muka kardus, layar kertas, dan papan tombol sungguhan untuk urutan langkah penarikan.

Aplikasi ojek: mockup yang dapat diklik diuji di pangkalan pengemudi: apakah titik jemput terlihat sebelum menerima pesanan?

Pemesanan warung lewat WhatsApp: kartu pesan yang diperankan bersama pelanggan tetap.

Dua atau tiga putaran pendek lebih baik daripada satu putaran panjang. Buku: Subbab 6.10.

IMK dalam Praktik: Profitina

Pengiriman seketika yang optimistis: kolom tulis langsung kosong dan kiriman langsung muncul; jika gagal, teksnya dikembalikan. Pengaturan waktu hanya dapat dibuat prototipenya dalam kode, dengan tunda dan kegagalan buatan.

Gerbang mutu: Content Engine hanya menyajikan draf AI (Qwen2.5 lokal) bila mengalahkan templat bawaan — dibangkitkan bukan sinonim lebih baik.

Widget lewat SSE: apakah sanggup mengikuti umpan pada koneksi ponsel yang lambat adalah pertanyaan implementasi untuk prototipe berkode kecil.

Profitina memberi saran dan penggunalah yang memutuskan; prototipe mencari tahu sebelum berkomitmen. Buku: Subbab 6.11.

Kesalahan yang Sering Terjadi

Memulai dengan fidelitas tinggi; membuat prototipe tanpa pertanyaan.

Membela prototipe selama sesi.

Layar tak terjangkau dan jalan buntu — jalankan `flow_check`.

Lupa label “PROTOTIPE”; data pribadi sungguhan dalam prototipe.

Memercayai keluaran AI sebagai riset; memperlakukan kode prototipe sebagai kode produk.

Mengotomasi pesan sebelum kata-katanya berhasil.

Setiap kesalahan punya obat satu baris di dalam bab. Buku: Subbab 6.12.

Soal Terselesaikan: Terpendek Bukan yang Lazim

Soal 6.3: dalam `flow_v1.txt`, pasien BPJS menempuh `welcome > service > bpjs > nik > confirm > ticket` — **lima** ketukan.

`flow_check` melaporkan **empat**, karena penelusuran melebar mengembalikan lintasan *terpendek*, melalui "bayar sendiri".

Di puskesmas ini sebagian besar pasien pagi memakai BPJS, sehingga jalur yang lazim justru yang panjang.

Soal 6.4 menambahkan penelusuran mendalam untuk lintasan sederhana *terpanjang*; pada `flow_v1.txt` hasilnya jalur BPJS lima ketukan.

Penelusurannya eksponensial pada kasus terburuk: tidak masalah untuk sepuluh layar, tetapi tidak untuk situs dengan ribuan halaman.

Laporkan jalur yang ditempuh pengguna yang paling umum, bukan hanya batas bawahnya. Buku: Subbab 6.14 dan 6.15, Soal 6.3 dan 6.4.

Soal Terselesaikan: Menambah Layar Bahasa

Soal 6.2: tambahkan `welcome -> language : English / Bahasa Indonesia`.

Pemeriksa kini melaporkan `dead ends (no way on or back): language` — layar baru tidak punya jalan keluar.

Tambahkan `language -> welcome : Kembali / Back` dan laporannya kembali menjadi `none`.

Layar menjadi 7, tombol menjadi 12 lalu 13; jalur menuju tiket tetap tiga ketukan.

Pelajarannya: setiap layar baru membutuhkan jalan maju atau kembali, dan pemeriksa menangkap kelalaian itu dalam sedetik.

Jalankan pemeriksa setelah setiap perubahan alur, bukan hanya sebelum putaran uji. Buku: Subbab 6.15, Soal 6.2.

Ringkasan

1. Prototipe menjawab pertanyaan: peran, tampilan dan rasa, implementasi.

2. Fidelitas sesuai dengan pertanyaannya (Gambar 6.1).

3. Prototipe kertas dengan komputer yang diam; Wizard of Oz untuk perilaku yang sulit.

4. Alur adalah graf; penelusuran melebar menemukan masalah dan memperpendek jalur dari empat atau lima ketukan menjadi tiga (Gambar 6.2).

5. Kios berkode menerbitkan nomor sungguhan, mencatat tampilan, memakai tombol besar, dalam Flask dan server JDK (Gambar 6.3).

6. Alat AI mencapai fidelitas tinggi dalam hitungan menit; tuliskan temuan di *prompt*, periksa, dan uji dengan manusia.

Buku: Subbab 6.13.

Pertemuan Berikutnya

Bab 7 — Evaluasi dan Pengujian Usabilitas: evaluasi heuristik, uji *think-aloud*, SUS dan UMUX-Lite, serta uji A/B, dengan Java dan Python.

Halaman pesan-dulu Warung Bu Tini, delapan pelanggan, empat tugas.

Sebelum itu: kerjakan Soal Latihan 6.2, 6.3, dan 6.5, lalu jalankan kedua server kios dan `shoot.py`.

Log tampilan dari prototipe hari ini menjadi waktu pengerjaan tugas di Bab 7. Buku: Subbab 6.14 dan 6.15.

Kuliah 7 — Evaluasi dan Pengujian Usabilitas

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 7

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita memeriksa klaim bahwa orang dapat memakai sebuah rancangan: inspeksi dengan heuristik, uji think-aloud bersama pengguna sungguhan, ukuran keberhasilan, waktu, dan kepuasan dengan selang yang jujur, berapa partisipan yang cukup, dan uji A/B pada ratusan pengunjung sungguhan. Buku: Bab 7, Subbab 7.1–7.17.

Cakupan Pertemuan Ini

1. Kasusnya — halaman pesan-dulu Warung Bu Tini, buatan Arif, anaknya.

2. Gagasannya — efektivitas, efisiensi, kepuasan; inspeksi versus empiris; formatif versus sumatif.

3. Evaluasi heuristik — evaluator mandiri dan nilai keparahan.

4. Uji usabilitas — tugas, *think-aloud*, dan apa yang terjadi.

5. Ukuran — selang *adjusted-Wald*, waktu rata-rata geometrik, kesalahan, SUS, UMUX-Lite.

6. Berapa partisipan — model penemuan masalah, dan batas-batasnya.

7. Uji A/B — uji-z dua proporsi dan ukuran sampel yang ditetapkan di muka.

Setiap rumus dikerjakan dengan tangan lebih dahulu, lalu dihitung program Java dan Python dengan keluaran identik. Buku: Bab 7.

Pesan Dulu di Warung Bu Tini

Nasi campur dekat gedung kantor dan kampus di Surabaya; antrean makan siang mencapai jalan.

Halaman buatan Arif: pilih lauk dan jumlah, pilih jam ambil, bayar dengan QRIS atau di kasir, terima konfirmasi.

Ia sudah mengujinya sendiri; teman-temannya bilang bagus.

Pertanyaannya: *dari mana kamu tahu pelanggan ibumu bisa memakainya?* Seorang pensiunan guru yang membeli makan siang untuk suaminya; pekerja kantor dengan ponsel kecil di antara dua rapat.

“Bisa dipakai oleh saya” bukan bukti tentang orang lain. Buku: Subbab 7.1.

Apa Arti “Mudah Dipakai”

> **DEFINISI.** *Usabilitas* adalah sejauh mana sebuah sistem dapat dipakai oleh pengguna tertentu untuk mencapai tujuan tertentu dengan *efektivitas*, *efisiensi*, dan *kepuasan* dalam konteks penggunaan tertentu (ISO 9241-11:2018).

Bukan satu angka: efektif tetapi lambat, atau cepat tetapi menjengkelkan, membutuhkan ukuran terpisah.

Hanya untuk pengguna, tujuan, dan konteks *tertentu*: setiap evaluasi menyebut siapa, melakukan apa, di mana.

Inspeksi (pakar, prinsip) versus **empiris** (pengguna sungguhan, tugas sungguhan).

Formatif (apa yang salah?) versus **sumatif** (seberapa baik?).

Uji Arif bersifat formatif; uji A/B di akhir bersifat sumatif. Buku: Subbab 7.2.

Evaluasi Heuristik

Nielsen dan Molich (1990): beberapa evaluator menilai kesesuaian dengan sekumpulan kecil prinsip, heuristik di Bab 3.

Prosedur: **beri pengarahan** tentang pengguna dan konteks; **periksa mandiri**, dua kali; **gabungkan**; **nilai keparahan** secara mandiri lalu rata-ratakan.

Satu evaluator menemukan sebagian kecil; beberapa evaluator menemukan jauh lebih banyak — karena itu tiga sampai lima.

Teman-teman Arif menemukan enam masalah dalam satu sore, termasuk ikon menu tanpa label yang menyembunyikan “ubah pesanan” dan pesan galat “Invalid input”.

Mereka melewatkan tiga masalah yang ditemukan pengguna, misalnya konfirmasi yang tidak menyebut kapan makanan siap. Buku: Subbab 7.3.1.

Nilai Keparahan

Tabel 7.1 Skala keparahan masalah usabilitas, dengan contoh dari halaman pemesanan. ::: {custom-style="SlideBullet"} Tabel 7.1: keparahan memadukan **frekuensi**, **dampak**, dan **kegigihan**. :::

Nilai	Arti	Contoh
0	Bukan masalah usabilitas	Warna yang tidak disukai evaluator
1	Kosmetik	Foto menu lambat dimuat
2	Kecil	Harga baru berubah setelah keluar dari kolom
3	Besar	Kode QRIS terlalu kecil untuk dipindai
4	Bencana	Tombol kembali mengosongkan keranjang

Masalah yang dialami sedikit pengguna setiap kali dan tak dapat diatasi bisa lebih parah daripada yang dialami semua orang sekali. Buku: Subbab 7.3.2 dan Tabel 7.1.

Merencanakan Uji

Pertanyaan riset: dapatkah pelanggan memesan, mengubah pesanan, dan membayar tanpa bantuan? Sama mudahnya bagi pelanggan lanjut usia?

Partisipan: delapan pelanggan tetap — empat pekerja kantor, dua mahasiswa, dua pensiunan di atas 60 tahun (P3, P6).

Tempat: ponsel mereka sendiri, di warung setelah jam ramai.

Tugas: **T1** cari menu dan harga hari ini; **T2** pesan dua porsi untuk pukul 12.00; **T3** ubah menjadi tiga porsi; **T4** bayar dengan QRIS (kode uji).

Ukuran: keberhasilan, waktu, kesalahan, SUS dan UMUX-Lite, masalah yang teramati.

Tugas adalah tujuan dengan kata-kata pelanggan, tidak pernah instruksi dengan kata-kata antarmuka. Buku: Subbab 7.4.1.

Berpikir dengan Bersuara

Ericsson dan Simon: orang dapat *mengucapkan* isi memori kerjanya sambil bekerja tanpa banyak mengubahnya — minta mereka mengucapkan, bukan menjelaskan.

Fasilitator sebagian besar diam; dorongan netral: “apa yang sedang Anda pikirkan?”

Kembalikan pertanyaan: “menurut Anda di mana seharusnya?”; bantu hanya bila buntu, dan hitung tugasnya sebagai gagal.

Momen: P3 memesan lagi alih-alih mengubah (tagihan ganda); P6 menggulir jam sejak 07.00; tombol kembali P1 mengosongkan keranjang; P5 merasa semuanya “biasa saja”.

Pengguna yang percaya diri menunjukkan rancangan berfungsi bagi mereka; mereka tidak dapat menunjukkan bahwa rancangan berfungsi bagi semua orang. Buku: Subbab 7.4.2 dan 7.4.3.

Keberhasilan Tugas dengan Selang yang Jujur

Dengan n kecil, selang biasa gagal (“8 dari 8” menghasilkan 100% sampai 100%).

Adjusted Wald (Sauro dan Lewis) menambahkan $z^2/2$ keberhasilan dan z^2 percobaan (Persamaan (7.1)):

$$\hat{p}_{\text{adj}} = \frac{x + z^2/2}{n + z^2}, \hat{p}_{\text{adj}} \pm z\sqrt{\frac{\hat{p}_{\text{adj}}(1-\hat{p}_{\text{adj}})}{n+z^2}} \tag{7.1}$$

T1, 8 dari 8: $9.921/11.842 = 0.838$, setengah lebar 0,210 → **62,8% sampai 100%**.

T3, 4 dari 8: 0,500 ± 0,285 → **21,5% sampai 78,5%**.

Selang yang lebar adalah pernyataan jujur tentang seberapa banyak yang dapat dikatakan delapan orang. Buku: Subbab 7.5.1.

Waktu Pengerjaan dan Kesalahan

Waktu tugas menceng; untuk *n* di bawah sekitar 25 pakai rata-rata geometrik (Persamaan (7.2)):

$$\acute{t}_{\text{geo}} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \ln t_i\right) \tag{7.2}$$

Tujuh waktu T4: rata-rata logaritma 4,021 → **55,8 detik** (rata-rata aritmetika 58,4 detik).

Percobaan gagal tidak dimasukkan ke waktu; sudah terhitung dalam penyelesaian.

Kesalahan: tindakan yang tidak mengarah ke tujuan — memperlihatkan *di mana* rancangan menyesatkan.

Satu partisipan yang sangat lambat dapat mendominasi rata-rata aritmetika delapan orang. Buku: Subbab 7.5.2 dan 7.5.3.

Hasil Tugas

```
Task  success  95% CI (adj. Wald)  geo-mean time  errors
T1    8/8     62.8% - 100.0%       37.4 s     0.25
T2    7/8     50.8% -  99.9%       82.9 s     1.25
T3    4/8     21.5% -  78.5%       97.9 s     1.75
T4    7/8     50.8% -  99.9%       55.8 s     0.50
```

T3, mengubah pesanan, adalah masalahnya: separuh gagal, sisanya paling lambat dan paling banyak salah.

Yang gagal: P1 (tombol kembali), P3 (memesan lagi), P6 (menyerah), P8 (tak menemukan tautannya).

Tabel ini berasal dari `usability_metrics.py` dan `UsabilityMetrics.java`. Buku: Subbab 7.5.4.

System Usability Scale

Brooke (1996): sepuluh butir, 1–5, bergantian positif (ganjil) dan negatif (genap).

Skornya (Persamaan (7.3)):

$$\mathrm{SUS} = 2.5\left(\sum_{i \in \{1,3,5,7,9\}} (a_i - 1) + \sum_{i \in \{2,4,6,8,10\}} (5 - a_i)\right) \tag{7.3}$$

P1 menjawab 4, 2, 4, 1, 3, 2, 4, 2, 4, 2: ganjil 14, genap 16, jumlah 30 → **75**.

Bukan persentase: produk rata-rata mendapat **68** (Sauro); nilai berkurva, misalnya 65,0–71,0 = C, 80,8–84,0 = A, 84,1+ = A+.

Ganti “sistem” dengan nama produknya dan periksa terjemahan dengan penerjemahan balik. Buku: Subbab 7.6.1.

UMUX-Lite

Lewis, Utesch, dan Maher (2013): dua butir skala 7 titik — “kemampuannya memenuhi kebutuhan saya”, “mudah dipakai”.

Skor mentah dan regresinya ke skala SUS (Persamaan (7.4)):

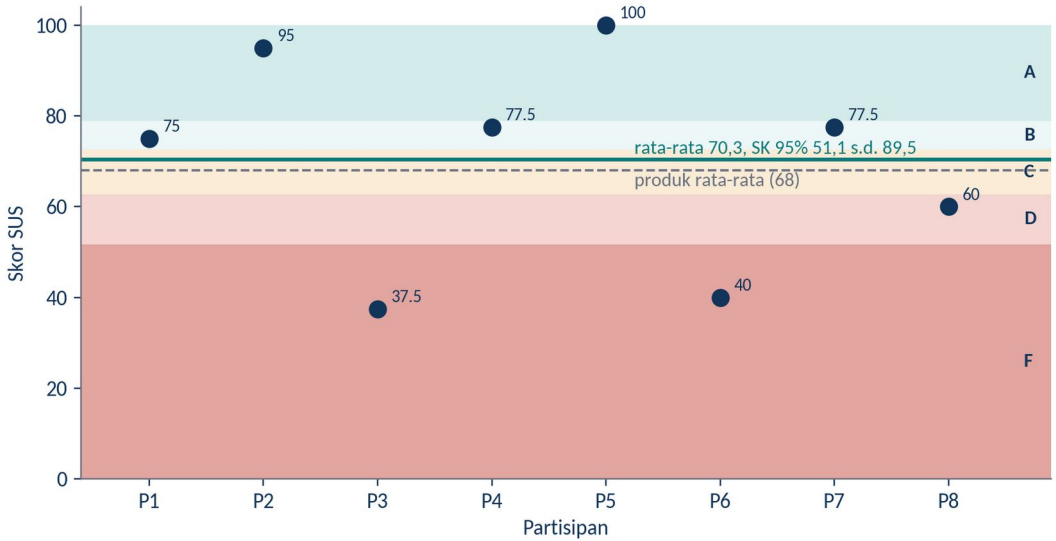
$$\mathrm{UMUX}_{\mathrm{raw}} = \frac{(c-1)+(e-1)}{12} \times 100\,, \widehat{\mathrm{SUS}} = 0.65\,\mathrm{UMUX}_{\mathrm{raw}} + 22.9 \tag{7.4}$$

P1: $c=6$, $e=5$ → mentah 75,0 → 71,65 pada skala SUS.

Pakai di tempat sepuluh butir akan dilewati, misalnya jendela sembul setelah memesan. Buku: Subbab 7.6.2.

Hasil Kepuasan: Lihat Setiap Orang

```
SUS mean 70.3 (95% CI 51.1 to 89.5), grade C; the average product scores 68
UMUX-Lite raw mean 74.0, on the SUS scale 71.0
```



Gambar 7.1 Skor SUS kedelapan partisipan terhadap pita nilai berkurva; rata-ratanya (70,3) sedikit di atas produk rata-rata (68), tetapi kedua partisipan lanjut usia berada di pita F.

Gambar 7.1: enam skor dari 60 sampai 100; **P3 37,5 dan P6 40** — partisipan lanjut usia, dua orang yang sama yang gagal di T3.

> **WAWASAN KUNCI:** Laporkan sebaran, bukan hanya rata-rata. Rata-rata dapat menyembunyikan justru orang-orang di pinggiran.

Buku: Subbab 7.6.3 dan Gambar 7.1.

Program Metrik

Satu fungsi per rumus: `adjusted_wald`, `geometric_mean`, `sus_score`, `umux_lite`, `discovery`.

Skala nilai berupa daftar batas bawah; `grade` mengembalikan yang pertama tercapai.

`sus_score` memakai posisi dalam daftar: posisi genap adalah butir bernomor ganjil yang positif.

Java membaca CSV dengan `split(",")`, cukup karena tak ada kolom yang memuat koma; `Locale.US` menjaga titik desimal.

Kedua versi mencetak laporan yang identik. Buku: Subbab 7.7.

Berapa Banyak Partisipan?

Nielsen dan Landauer: jika setiap partisipan mengalami sebuah masalah dengan peluang p, n partisipan menemukannya dengan peluang (Persamaan (7.5)):

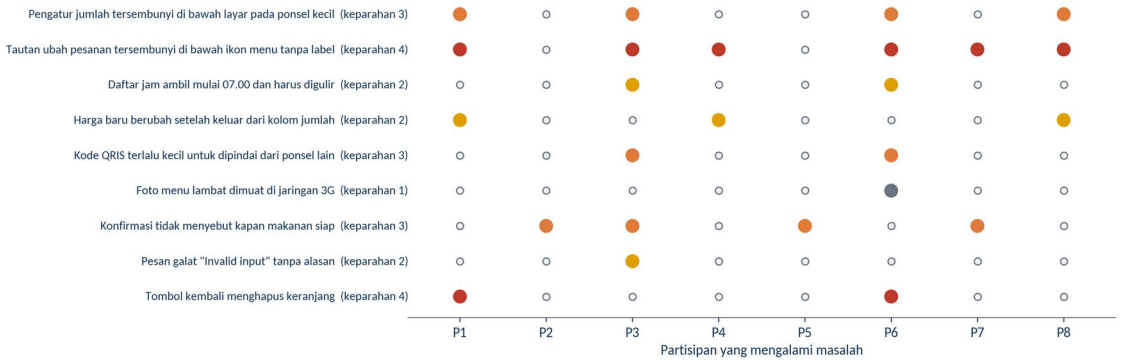
$$P(\text{found}) = 1 - (1 - p)^n \tag{7.5}$$

Dengan $p = 0.31$: lima partisipan menemukan $1 - 0.69^5 = 0.844$, sekitar **85%**.

Anjuran Nielsen: uji sekitar lima, perbaiki, lalu uji lagi.

Hasil terbaik datang dari beberapa putaran kecil, bukan satu putaran besar. Buku: Subbab 7.8.1.

Menaksir p dari Studi

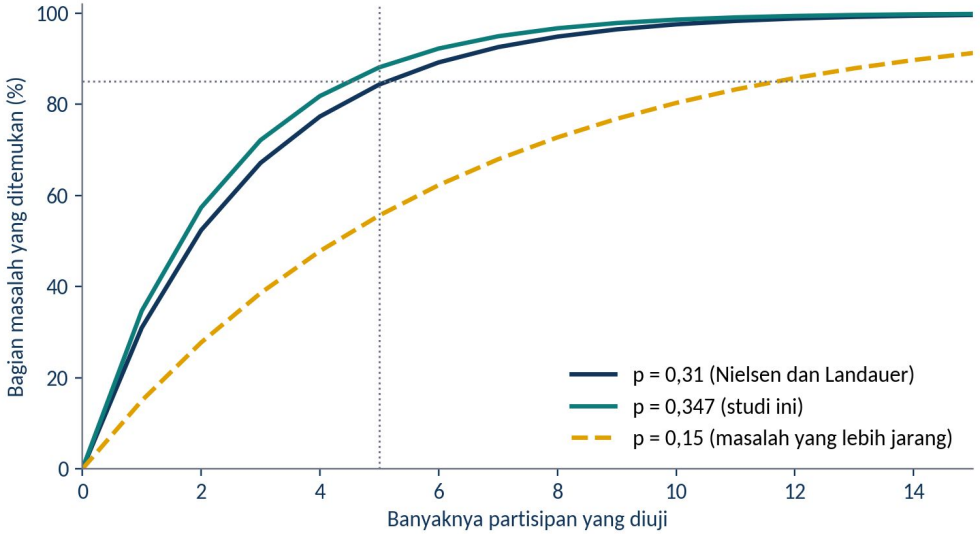


Gambar 7.2 Sembilan masalah dalam uji warung dan partisipan yang mengalami masing-masing; P3 dan P6 mengalami masalah terbanyak.

Gambar 7.2: 9 masalah × 8 partisipan = 72 sel, 25 kejadian → $\hat{p} = 0.347$.

Matriks ini sudah memperlihatkan bahwa partisipan tidak serupa. Buku: Subbab 7.8.2 dan Gambar 7.2.

Tiga Peringatan tentang “Lima Pengguna”



Gambar 7.3 Bagian masalah yang diharapkan ditemukan seiring bertambahnya partisipan: dengan p sekitar 0,3, lima partisipan menemukan sekitar 85%; dengan masalah yang lebih jarang (p = 0,15), sekitar 56%.

Gambar 7.3. **1.** $\hat{p}$ dari studi kecil bias ke atas (Lewis, 2001). **2.** Model menganggap partisipan serupa — lima pekerja kantor yang percaya diri akan menyembunyikan masalah pelanggan lanjut usia. **3.** Masalah yang lebih jarang butuh lebih banyak: dengan $p=0.15$, lima hanya menemukan 56%.

Lima pengguna menemukan sebagian besar masalah dari jenis pengguna yang diuji. Uji setiap kelompok yang berbeda. Buku: Subbab 7.8.2 dan Gambar 7.3.

Menguji Metrik

Jawaban yang diketahui: SUS 100 dan 0 pada titik ekstrem, 50 untuk semua tiga; batas-batas nilai; UMUX-Lite tertinggi (100; 87,9); *adjusted Wald* 8/8 mulai 62,8%; rata-rata geometrik 10 dan 1000 adalah 100; penemuan 84% untuk lima; $\Phi(1.96) = 0.975$; uji A/B.

```
11 checks passed
```

Rumus yang tampak masuk akal ketika salah layak diuji. Buku: Subbab 7.9.

Dari Delapan Orang ke Ratusan: Uji A/B

Setelah perbaikan (keranjang tetap, tombol “Ubah pesanan”, jam mulai 11.30, QRIS besar, “Siap pukul 12.00”), putaran kedua dengan lima partisipan baru tidak menemukan kegagalan T3.

Pertanyaan tersisa: label tombol mana yang membuat lebih banyak pengunjung memesan? A *Kirim* versus B *Pesan sekarang – ambil jam 12.00*.

> **DEFINISI.** *Uji A/B* menugaskan pengunjung secara acak ke versi A atau B dan membandingkan metrik yang dipilih sebelumnya; penugasan acak memungkinkan selisih yang lebih besar daripada kebetulan dikaitkan dengan perubahan itu (Kohavi, Tang, dan Xu).

Aturan: pilih metrik lebih dahulu; tetapkan ukuran sampel atau durasi di muka; periksa bahwa ukuran kelompok sesuai rencana.

Selisih kecil hanya dapat diukur pada banyak orang, dan pengguna sungguhan adalah orang yang tepat. Buku: Subbab 7.10.1.

Uji-z Dua Proporsi

A: 157 pesanan dari 1.210 pengunjung (12,98%); B: 198 dari 1.187 (16,68%).

Gabungan $\acute{p} = 355/2397 = 0.1481$; statistiknya (Persamaan (7.6)):

$$z = \frac{\hat{p}_B - \hat{p}_A}{\sqrt{\acute{p}(1-\acute{p})\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}} \tag{7.6}$$

Galat baku 0,01451, $z = 2.55$, p dua sisi ≈ 0.011.

Selang 95% selisihnya (tanpa penggabungan): sekitar +0,9 sampai +6,6 poin.

Seandainya label tidak berpengaruh, selisih sebesar ini hanya muncul sekitar sekali dalam seratus uji. Buku: Subbab 7.10.2.

Ukuran Sampel Ditetapkan di Muka

Pengunjung per versi untuk mendeteksi $p_1 \to p_2$ pada taraf 5% dengan kuasa 80% (Persamaan (7.7)):

$$n = \frac{\left(z_{\alpha/2}\sqrt{2\,\acute{p}(1-\acute{p})} + z_{\beta}\sqrt{p_1(1-p_1)+p_2(1-p_2)}\right)^2}{(p_2-p_1)^2} \tag{7.7}$$

Memperkecil selisih yang dideteksi menjadi separuhnya melipatempatkan sampel.

Φ dihitung dengan Abramowitz–Stegun 7.1.26 (galat di bawah 1.5×10^{-7}) karena Java tidak mempunyai fungsi galat bawaan.

Hampiran yang sama di kedua bahasa, keluaran identik. Buku: Subbab 7.10.3 dan 7.10.4.

Hasil A/B, dengan Jujur

```
difference B - A: +3.71 points, 95% CI +0.86 to +6.55
z = 2.554, two-sided p = 0.0107
at the 5% level the difference is significant
visitors per variant to detect 13% -> 16%: 2162
visitors per variant to detect 13% -> 14%: 18330
```

B sangat mungkin lebih baik; tetapi 2.162 per versi dibutuhkan untuk kenaikan 3 poin dan warung hanya punya sekitar 1.200 — **kuasa kurang, selang lebar**.

Kenaikan 1 poin butuh 18.330 per versi: berbulan-bulan lalu lintas — nilailah penyempurnaan kecil dengan uji usabilitas.

> **PERINGATAN:** Jangan mengintip lalu berhenti. Memeriksa setiap hari dan berhenti pada $p<0.05$ pertama menyatakan pemenang palsu jauh lebih sering daripada 5%.

Laporkan “B sangat mungkin lebih baik; seberapa besar, kami belum tahu dengan tepat.” Buku: Subbab 7.10.4.

Evaluasi pada 2026

Uji **jarak jauh** menjangkau orang di luar kota; uji **tanpa pemandu** menjangkau banyak orang dengan murah; keduanya kehilangan pertanyaan yang tepat pada saat yang tepat.

Fitur AI: transkrip, penanda keraguan, ringkasan, bahkan moderator AI.

Pakai AI untuk menemukan momen yang perlu ditonton, bukan menggantikan menontonnya; periksa ringkasan terhadap rekaman; rekaman adalah data pribadi menurut UU PDP.

Ringkasan AI “pengguna merasa mudah” akan menyembunyikan P3 dan P6 persis seperti rata-rata. Buku: Subbab 7.11.

Menerapkannya dalam Kenyataan

Rencana satu halaman: tujuan; inspeksi lebih dahulu; 5–8 per putaran dan per kelompok; tugas realistis; ukuran dengan selang; gambarkan setiap skor; iterasi; A/B hanya untuk pertanyaan berlalu lintas besar dengan Persamaan (7.7) lebih dahulu.

Kios puskesmas: tugas yang sama pada kios sungguhan, pasien lanjut usia diperbanyak; log layar memberi waktu.

Aplikasi ojek: pengemudi, ponsel sendiri, di luar ruangan, satu tangan.

Desain ulang ATM: tinjauan heuristik terhadap pedoman aksesibilitas, uji bersama nasabah tunanetra dan lanjut usia dengan colokan audio, penyelesaian per transaksi dari log mesin.

Buku: Subbab 7.12.

IMK dalam Praktik: Profitina

Skor yang harus dipahami: tugas seperti “cari tahu mengapa skor Anda turun minggu ini” memperlihatkan apakah pemilik dapat berpindah dari angka ke tindakan.

Gerbang rilis: seluruh rangkaian pengujian otomatis, jauh lebih dari seribu termasuk penjaga UI JavaScript, harus berhasil; rilis bertahap baru melapor sukses setelah pemeriksaan kesehatan menjawab.

Pengujian otomatis menjaga perbaikan usabilitas kemarin agar tidak rusak; uji usabilitas menemukan masalah esok hari.

Jawaban AI diukur bersama pengguna: setiap rilis model lokal dibandingkan dengan versi yang berjalan sebelum disajikan.

Profitina memberi saran dan penggunalah yang memutuskan; evaluasi memperlihatkan apakah sarannya membantu. Buku: Subbab 7.13.

Kesalahan yang Sering Terjadi

Menguji dengan teman; tugas yang membocorkan jawaban; membantu selama uji.

“8 dari 8” dilaporkan 100%; rata-rata aritmetika waktu yang menceng.

SUS dibaca sebagai persentase; butir negatif tidak dibalik.

“Lima pengguna selalu cukup.”

Mengintip uji A/B; menganggap signifikan berarti tepat; ringkasan AI alih-alih menonton.

Buku: Subbab 7.14.

Soal Terselesaikan: SUS dan Lima Pengguna

Soal 7.1 — P3 menjawab 3, 4, 2, 4, 3, 3, 3, 4, 2, 3.

Butir ganjil (3, 2, 3, 3, 2) menyumbang $2+1+2+2+1=8$; butir genap (4, 4, 3, 4, 3) menyumbang $1+1+2+1+2=7$; $2.5\times 15=37.5$, sama dengan cetakan.

Soal 7.4 — partisipan untuk 90% masalah: $n \geq \ln 0.10 / \ln(1-p)$.

$p=0.31$: $-2.3026/-0.3711=6.2$, jadi **7**; $p=0.15$: $-2.3026/-0.1625=14.2$, jadi **15**.

Jawaban kedua menunjukkan mengapa “lima pengguna” tidak dapat menjadi aturan tetap: masalah yang lebih jarang melipatduakan kebutuhan.

Buku: Subbab 7.17, Soal 7.1 dan 7.4.

Soal Terselesaikan: Uji yang Tidak Signifikan

Soal 7.6 — dua tata letak menu: A 96 dari 800 (12,00%), B 108 dari 790 (13,67%).

Selisih +1,67 poin, selang 95% −1,62 sampai +4,96, $z = 0.996$, $p = 0.32$: tidak signifikan.

Kesimpulan jujur: “belum ada bukti selisih *dengan sampel ini*”, bukan “kedua tata letak sama saja”.

Langkah berikutnya: jalankan sampai sampel yang direncanakan tercapai, atau nilai tata letaknya dengan uji usabilitas.

Tidak adanya bukti bukan bukti ketiadaan, terutama pada uji yang kuasanya kurang. Buku: Subbab 7.17, Soal 7.6.

Ringkasan

1. Usabilitas: efektivitas, efisiensi, kepuasan bagi pengguna, tujuan, dan konteks tertentu.

2. Evaluasi heuristik dengan nilai keparahan (Tabel 7.1).

3. Uji *think-aloud* dengan tujuan dalam kata-kata pengguna; fasilitator diam.

4. Penyelesaian dengan Persamaan (7.1), waktu dengan Persamaan (7.2); T3 berhasil bagi 4 dari 8.

5. SUS (Persamaan (7.3)) dan UMUX-Lite (Persamaan (7.4)): rata-rata 70,3 (C), tetapi P3 dan P6 di F (Gambar 7.1).

6. Persamaan (7.5): lima sampai delapan per putaran per kelompok (Gambar 7.3).

7. A/B dengan Persamaan (7.6) dan Persamaan (7.7): 12,98% → 16,68%, $p = 0.011$, selang +0,9 sampai +6,6.

Buku: Subbab 7.15.

Pertemuan Berikutnya

Bab 8 — Desain Visual, Design System, dan Arsitektur Informasi: Gestalt, huruf dan grid, kontras WCAG 2.2 dari kode heksadesimal, *design token*, Material 3 Expressive, Liquid Glass, Fluent 2, dan *card sorting* dengan pengelompokan.

Sebelum itu: kerjakan Soal Latihan 7.1, 7.2, 7.4, dan 7.6; jalankan `usability_metrics` dan `ab_test` dalam kedua bahasa.

Aplikasi bank di Bab 8 akan diperiksa dengan kepekaan yang sama terhadap orang-orang di pinggiran. Buku: Subbab 7.16 dan 7.17.

Kuliah 8 — Desain Visual, Design System, dan Arsitektur Informasi

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 8

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita membuat struktur terlihat dan menjaganya tetap begitu: pengelompokan Gestalt dan hierarki, huruf dan grid, kontras yang dihitung dari kode heksadesimal, design token dan bahasa desain platform tahun 2026, serta menu yang dibangun dari cara nasabah mengelompokkan sesuatu. Buku: Bab 8, Subbab 8.1–8.16.

Cakupan Pertemuan Ini

1. Kasusnya — aplikasi bank perkreditan rakyat yang tak dapat dibaca dan dinavigasi nasabah.

2. Melihat kelompok — kedekatan, kesamaan, wilayah bersama, ketertutupan; hierarki visual dan uji menyipit.

3. Huruf dan tata letak — ukuran, panjang baris, skala modular, jarak dan grid.

4. Warna dan kontras — WCAG 2.2, luminans relatif, rasio kontras, pemeriksa yang membaca *design token* W3C.

5. *Design system* — token, komponen, pedoman; Material 3 Expressive, Liquid Glass, Fluent 2.

6. Arsitektur informasi — organisasi, label, navigasi, pencarian; *card sorting* dan pengelompokan; *tree testing*.

Programnya `contrast` dan `cardsort`, dalam Java dan Python dengan keluaran identik. Buku: Bab 8.

Aplikasi Sebuah BPR

BPR Tani Makmur, Jawa Timur: petani, pedagang pasar, guru, pensiunan pegawai negeri; banyak yang berusia di atas lima puluh.

Keluhan tiap minggu: “hurufnya terlalu kecil”, “tulisan putih di atas hijau tidak terbaca”, “bayar angsuran di mana?”

Layar beranda: lima belas ikon sama besar dengan lima belas warna cerah, kebanyakan tanpa label; saldo berhuruf abu-abu kecil.

Tak ada yang salah bila dilihat satu per satu; bersama-sama, setiap butir berteriak sama keras, sehingga tak ada yang terdengar.

Tiga langkah: buat struktur terlihat dan periksa kontras; tuliskan sebagai *design system* kecil; susun ulang menu dari *card sorting*.

Card sorting akan menunjukkan bahwa nasabah menaruh angsuran pinjaman di bawah tagihan. Buku: Subbab 8.1.

Tiga Gagasan

DEFINISI. *Desain visual* menata huruf, warna, ruang, dan gambar agar struktur dan prioritas terlihat sekilas. *Arsitektur informasi* mengorganisasi, memberi label, menavigasi, dan mencari isi serta fungsi. *Design system* adalah sekumpulan keputusan bersama yang terdokumentasi: token, komponen, dan aturan.

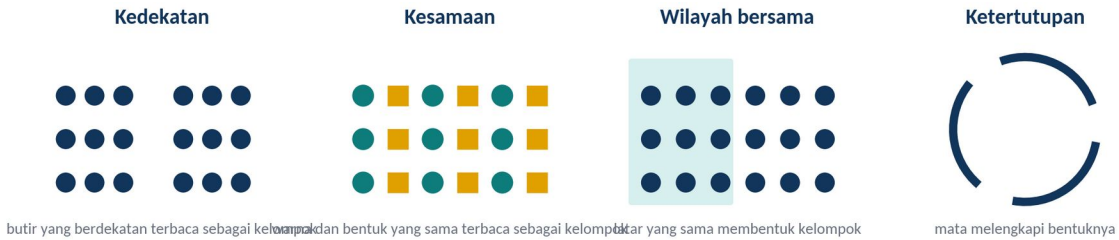
Desain visual: *apa yang saya lihat lebih dahulu, dan apa berpasangan dengan apa?*

IA: *di mana letaknya, dan apa namanya?*

Design system: *bagaimana mengambil keputusan yang sama dengan cara yang sama setiap kali?*

Layar indah dengan struktur salah akan gagal; begitu pula struktur baik yang tak dapat dibaca. Buku: Subbab 8.2.

Gestalt: Cara Orang Melihat Kelompok



Gambar 8.1 Empat prinsip Gestalt yang dipakai dalam desain antarmuka: kedekatan, kesamaan, wilayah bersama, dan ketertutupan.

Gambar 8.1: sistem penglihatan mengelompokkan unsur sebelum kita sadar melihatnya (riset seabad, diulas Wagemans dkk., 2012).

Buku: Subbab 8.3.1 dan Gambar 8.1.

Memakai Keempat Prinsip

Kedekatan: jarak antarkelompok jelas lebih besar daripada jarak di dalam kelompok.

Kesamaan: semua yang dapat diketuk tampak serupa; warna ikon acak merusak pengelompokan.

Wilayah bersama: latar kartu mengelompokkan isinya lebih kuat daripada jarak saja.

Ketertutupan: daftar yang terpotong di bawah berarti “masih ada lagi”.

Beranda baru bank: saldo dan dua tindakan sehari-hari dalam satu kartu; fungsi yang lebih jarang di kartu kedua; pengaturan dan bantuan di tab tersendiri; hanya yang dapat diketuk berwarna *teal*.

Kesamaan berlaku dua arah: yang tidak dapat diketuk tidak boleh tampak dapat diketuk. Buku: Subbab 8.3.1.

Hierarki Visual dan Uji Menyipit

Hierarki = urutan mata menangkap layar: ukuran, ketebalan, warna, posisi, keterpisahan.

Urutan visual harus sesuai kepentingan untuk tugas paling umum: saldo dahulu, lalu transfer dan bayar.

Uji menyipit: buramkan layar. Lama: lima belas gumpalan setara. Baru: angka besar, dua tombol, daftar yang tenang.

Jika layar yang buram tidak memperlihatkan urutan kepentingan, layar yang tajam pun tidak. Buku: Subbab 8.3.2.

Huruf yang Dapat Dibaca

Ukuran: teks isi minimal 16 px CSS di ponsel; hormati Dynamic Type dan skala huruf Android.

Panjang baris: sekitar 45–75 karakter.

Penekanan: ketebalan, bukan deretan panjang huruf kapital.

Angka: *Rp1.250.000*, format yang dipakai nasabah — bukan *1250000* atau *IDR 1,250,000*.

Bagi bank, angka adalah teks terpenting di layar. Buku: Subbab 8.4.1.

Skala Huruf Modular

Pilih dasar b dan rasio r; pakai hanya (Persamaan (8.1)):

$$s_k = b \cdot r^k, k = \ldots, -1, 0, 1, 2, \ldots \tag{8.1}$$

$b = 16$, $r = 1.25$: 12,8, 16, 20, 25, 31,25, 39,06 px → bank memakai 13, 16, 20, 25, 39.

Setiap langkah 25% lebih besar: terlihat sebagai tingkatan, bukan kekeliruan.

Ukuran yang berbeda satu atau dua piksel tampak seperti galat, bukan tingkatan. Buku: Subbab 8.4.2.

Ruang, Grid, dan Target

Skala jarak kelipatan 4 atau 8 px: 8 px di dalam kelompok, 24 px antarkelompok, 16 px di tepi.

Grid ponsel: empat kolom, margin dan celah 16 px.

Target: SC 2.5.8 WCAG 2.2 mensyaratkan minimal 24 × 24 px CSS (atau jarak yang cukup); platform menganjurkan sekitar 44–48 untuk kontrol utama.

Ruang adalah cara kedekatan diungkapkan. Buku: Subbab 8.4.3.

Mengapa Kontras Diukur

WCAG 2.2 (Rekomendasi W3C sejak Oktober 2023): **4,5:1** teks biasa, **3:1** teks besar (SC 1.4.3); **3:1** untuk batas kontrol, ikon, dan indikator fokus (SC 1.4.11).

Oktober 2025: WCAG 2.2 disahkan sebagai **ISO/IEC 40500:2025**.

WCAG 3.0 masih **Working Draft** (draf baru untuk ditinjau, September 2026); aturan hukum hari ini merujuk WCAG 2.x.

SC 1.4.1: warna tidak pernah menjadi *satu-satunya* tanda — bank menambahkan “Berhasil”/“Gagal” dan ikon pada hijau dan merah.

Defisiensi penglihatan warna merah-hijau dialami beberapa persen laki-laki pada banyak populasi. Buku: Subbab 8.5.1.

Dari Heksadesimal ke Luminans

Linearkan setiap kanal sRGB (Persamaan (8.2)):

$$c = \frac{c_8}{255}, \qquad C = \begin{cases} c/12.92 & \text{jika } c \leq 0.04045 \\ \left(\frac{c+0.055}{1.055}\right)^{2.4} & \text{jika tidak} \end{cases} \tag{8.2}$$

Beri bobot pada kanal, hijau terbesar (Persamaan (8.3)):

$$L = 0.2126\,R + 0.7152\,G + 0.0722\,B \tag{8.3}$$

Pengodean sRGB menyimpan kecerahan secara tidak linear; luminans membutuhkan cahaya linear. Buku: Subbab 8.5.2.

Rasio Kontras, Dihitung

Dengan L_1 yang lebih terang dan L_2 yang lebih gelap (Persamaan (8.4)):

$$\mathrm{CR} = \frac{L_1 + 0.05}{L_2 + 0.05} \tag{8.4}$$

Navy `#12355B` = (18, 53, 91): linear 0,00605, 0,0356, 0,1046 → $L = 0.0343$; di atas putih \$1.05/0.0843 = \$ **12,46:1**.

Putih di atas *amber* `#E0A100`: $L = 0.4134$ → \$1.05/0.4634 = \$ **2,27:1** — gagal bahkan untuk teks besar.

Skala dari 1:1 sampai 21:1; 0,05 memodelkan cahaya ruangan yang dipantulkan layar.

Amber untuk sorotan dan ikon di atas navy, bukan latar untuk teks putih. Buku: Subbab 8.5.2.

Warna sebagai *Design Token* W3C

Design Tokens Format Module **2025.10**, versi stabil pertama, 28 Oktober 2025.

JSON; setiap token punya `$value` dan `$type`; nilai warna menyebut ruang warna dan komponennya, dengan `hex` opsional.

```
"navy": {
  "$value": {
    "colorSpace": "srgb",
    "components": [0.0706, 0.2078, 0.3569],
    "hex": "#12355B"
  }
}
```

`pairs.csv` mendaftar setiap pasangan yang benar-benar dipakai di layar, dengan pemakaiannya: `text`, `large`, `ui`.

Satu berkas token dapat melayani pemeriksa, CSS, dan sumber daya Android. Buku: Subbab 8.6.1.

Pemeriksa Kontras

`contrast.py` / `Contrast.java`: hitung Persamaan (8.2), Persamaan (8.3), Persamaan (8.4) untuk setiap pasangan; bandingkan dengan minimum untuk pemakaiannya.

Untuk pasangan yang gagal: gelapkan warna yang *lebih gelap* dengan langkah 1% sampai minimum tercapai — selisih luminans selalu membesar.

Java tidak punya pengurai JSON: ekspresi reguler membaca berkas datar ini; pakai pustaka JSON untuk yang lebih umum.

Saran itu memperlihatkan seberapa jauh warna harus bergeser; perancang tetap boleh memilih rona lain. Buku: Subbab 8.6.2 dan 8.6.3.

Keluaran Pemeriksa

```
grey       grey-light  text    4.39:1  FAIL (needs 4.5:1)
                               try foreground #69707D instead of #6B7280: 4.53:1
white      green-old   text    2.78:1  FAIL (needs 4.5:1)
                               try background #3A853D instead of #4CAF50: 4.56:1
teal       teal-light  text    4.18:1  FAIL (needs 4.5:1)
                               try foreground #0E7676 instead of #0F7C7C: 4.52:1
6 pair(s) fail WCAG 2.2 AA
```

Tombol lama putih-di-atas-hijau: 2,78:1 — “tulisan putih di atas hijau tidak terbaca”.

Abu-abu lulus di atas putih (4,83) tetapi gagal di atas kartu abu-abu muda (4,39).

Teal di atas *teal* muda gagal tipis; `#0E7676` memperbaikinya tanpa terlihat.

Buku: Subbab 8.6.4.

Pasangan sebagai Contoh Warna



Gambar 8.2 Sepuluh pasangan warna aplikasi bank sebagai contoh warna, dengan pemakaian, rasio kontras, dan hasil WCAG 2.2 AA.

Gambar 8.2: enam dari sepuluh pasangan gagal.

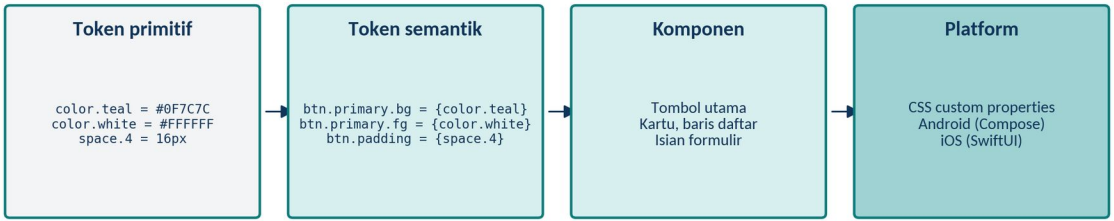
> **PERINGATAN:** Periksa pasangan yang benar-benar dipakai — teks di atas kartu, gambar, keadaan nonaktif dan *hover* — bukan hanya teks utama di atas latar utama.

Buku: Subbab 8.6.4 dan Gambar 8.2.

Design System dan Token

Tiga lapis: **token** (nilai), **komponen** (dengan keadaan: normal, ditekan, nonaktif, terfokus, galat), **pedoman**.

Token **primitif** menamai nilai; token **semantik** menamai tujuan dan menunjuk ke primitif; komponen hanya memakai token semantik.



Gambar 8.3 *Design token* dalam *design system*: token primitif memuat nilai, token semantik menamai tujuan, komponen memakai token semantik, dan alat *build* mengubah satu berkas token menjadi kode untuk setiap platform.

Gambar 8.3: gelapkan satu *teal* primitif dan setiap komponen ikut berubah; tema berkontras tinggi mengarahkan ulang token semantik.

Implementasi rujukan format 2025.10: Style Dictionary, Tokens Studio, Terrazzo; pengadopsinya termasuk Figma dan Penpot. Buku: Subbab 8.7.1 dan Gambar 8.3.

Bahasa Desain Tahun 2026

Material 3 Expressive (Mei 2025): bentuk beragam, huruf tegas, warna kaya, gerak berbasis fisika; Google melaporkan 46 studi, 18.000+ partisipan, unsur kunci ditemukan hingga 4× lebih cepat, pengguna tua secepat yang muda — temuan Google tentang sistemnya sendiri.

Liquid Glass (9 Juni 2025; iOS 26 dan saudara-saudaranya): material tembus pandang yang membiaskan dan menyesuaikan diri dengan isi; iOS 26.1 menambah pilihan *Clear* versus *Tinted* (opasitas dan kontras lebih tinggi) di samping Reduce Transparency.

Fluent 2 (Microsoft): token, komponen, dan pedoman aksesibilitas terbuka untuk Windows, web, iOS, dan Android.

Di atas latar yang berubah, jamin kontras dengan lapisan penyangga atau rona, dan uji dengan isi sungguhan. Buku: Subbab 8.7.2.

Apa yang Sebaiknya Dilakukan Tim Kecil

Sistem yang *kecil*: sekitar dua puluh warna, skala huruf, skala jarak, tiga radius, selusin komponen dengan keadaannya, satu halaman aturan.

Bangun di atas konvensi platform agar nasabah menemukan segala sesuatu di tempat yang diajarkan ponselnya.

Aturan pertama bank: “setiap pasangan teks dan ikon lulus `contrast.py` sebelum dirilis”.

Beri sistem itu seorang pemilik.

Sistem tanpa pemilik membusuk menjadi folder berisi berkas lama. Buku: Subbab 8.7.3.

Arsitektur Informasi

Organisasi: menurut tugas, topik, kelompok pengguna, abjad, waktu — kebingungan datang dari mencampur skema tanpa sadar.

Pelabelan: label adalah janji; “Layanan” tidak menjanjikan apa-apa, “Bayar tagihan” menjanjikan sesuatu yang dikenali.

Navigasi: tab bawah untuk tiga sampai lima tujuan utama, daftar dan kartu di dalamnya, jalan kembali yang jelas.

Pencarian: gagal ketika kata pencariannya kata perancang, bukan kata pengguna.

Aplikasi lama: satu grid datar, label internal (“Mutasi”, “Inquiry”, “Setor Angsuran”), tanpa pencarian.

Empat sistem menurut Rosenfeld, Morville, dan Arango. Buku: Subbab 8.8.1.

Card Sorting dan *Tree Testing*

***Card sorting* terbuka:** partisipan mengelompokkan kartu dan menamai tumpukannya — menemukan cara berpikir dan kata-kata mereka.

***Card sorting* tertutup:** kartu ke kategori yang ditentukan — memeriksa struktur usulan.

15–30 partisipan dari satu kelompok pengguna biasanya memberi gambaran stabil (Spencer).

***Tree testing*:** hanya teks hierarki menu; “di mana Anda akan membayar angsuran pinjaman?”

Bank: 20 nasabah di dua cabang, 15 kartu; `cardsort.csv` mencatat partisipan, kartu, dan nama tumpukan.

Card sorting mengusulkan; tree testing memeriksa. Buku: Subbab 8.8.2.

Kemiripan dan *Average Linkage*

Kemiripan kemunculan bersama kartu i dan j (Persamaan (8.5)):

$$S_{ij} = \frac{n_{ij}}{N} \tag{8.5}$$

Mulai dengan setiap kartu sendirian; berulang kali gabungkan dua kelompok yang paling mirip, dengan *average linkage* (Persamaan (8.6)):

$$S(A, B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} S_{ij} \tag{8.6}$$

Average linkage menghindari perangkaian *single linkage* dan ketatnya *complete linkage*.

Nama tumpukan tidak masuk ke kemiripan; yang penting hanya apakah dua kartu berbagi tumpukan. Buku: Subbab 8.9.1 dan 8.9.2.

Program *Card Sorting*

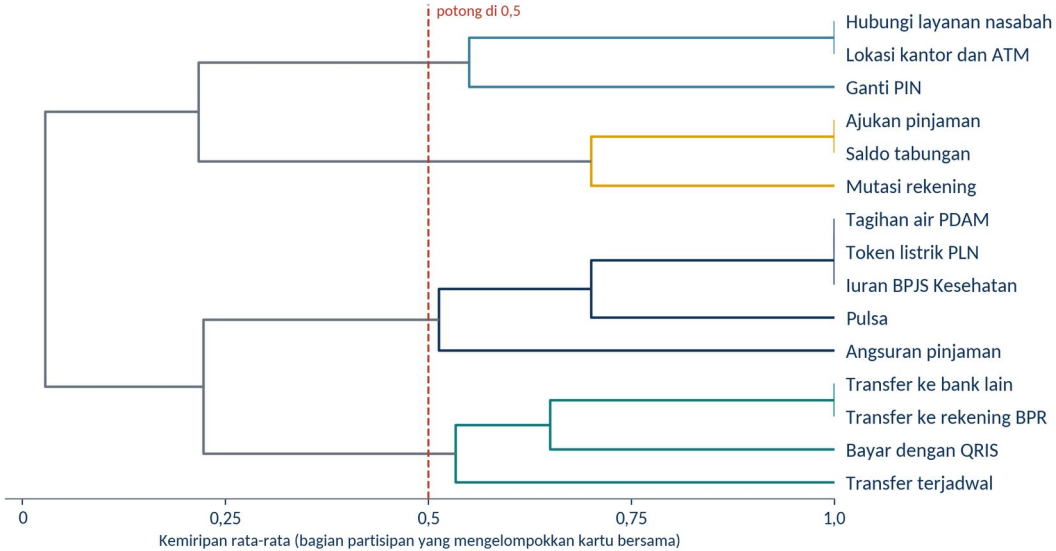
`cardsort.py` / `CardSort.java`: kemiripan, penggabungan bernama G1, G2, …, dan kelompok pada suatu potongan.

```
  0.70  G6 = Phone credit + G3
  0.70  G7 = Account statement + G4
  0.65  G8 = Pay by QRIS + G1
  0.55  G9 = Change PIN + G5
  0.53  G10 = Scheduled transfer + G8
  0.51  G11 = Loan installment + G6
  0.22  G12 = G10 + G11
```

Ditulis agar jelas, bukan cepat: cukup untuk puluhan kartu; toleransi `1e-12` membuat Java dan Python memecah seri dengan cara yang sama.

Buku: Subbab 8.9.3 dan 8.9.4.

Dendrogram



Gambar 8.4 Dendrogram *card sorting* terbuka (20 partisipan, 15 kartu, *average linkage*); memotong pada kemiripan 0,5 menghasilkan empat kelompok, dan angsuran pinjaman bergabung dengan tagihan.

Gambar 8.4: makin ke kanan dua kartu bergabung, makin banyak partisipan yang sepakat.

Buku: Subbab 8.9.4 dan Gambar 8.4.

Membaca *Card Sorting*

Empat kelompok pada 0,5: transfer (dengan QRIS dan transfer terjadwal); tabungan dan pinjaman; bantuan; **tagihan — bersama angsuran pinjaman** (0,51).

Angsuran dikelompokkan dengan tagihan air oleh 55% dan dengan saldo tabungan oleh 45%: angsuran adalah sesuatu yang Anda *bayar*.

Keputusan: empat tab dengan kata-kata nasabah — *Kirim uang*, *Bayar*, *Tabungan & pinjaman*, *Bantuan*; angsuran di bawah *Bayar* **dan** ditautkan dari halaman pinjaman; “Mutasi” menjadi “Riwayat transaksi”, tetap sebagai kata kunci pencarian.

Tree testing: 18 dari 20 menemukan angsuran pada percobaan pertama.

> **WAWASAN KUNCI:** *Card sorting* memperlihatkan cara nasabah berpikir: kelompok mengusulkan pengelompokan, nama tumpukan mengusulkan label, butir yang terbelah meminta tautan silang. Perancang memutuskan; *tree testing* memeriksa.

Buku: Subbab 8.9.4.

Menguji Program

Kontras: 21:1 untuk hitam di atas putih dalam urutan mana pun, 1:1 untuk warna identik; pembaca token; *navy* di atas putih 12,46; perbaikan mencapai targetnya.

Card sorting: empat belas penggabungan untuk lima belas kartu; empat kelompok pada 0,5 dengan angsuran di antara tagihan; dua kelompok pada 0,2.

```
10 checks passed
```

Buku: Subbab 8.10.

Soal Terselesaikan

Soal 8.3 — ikon *amber* di atas bilah *navy*: $(0.4134+0.05)/(0.0343+0.05)=5.50$ → lulus. Kontras milik *pasangan*, tidak pernah milik satu warna.

Soal 8.5 — potongan 0,6: tujuh kelompok (transfer terjadwal, angsuran, dan Ganti PIN keluar); 0,7: delapan (QRIS juga keluar). Keempat kartu itu calon tautan silang dan label yang lebih jelas.

Soal 8.6 — *single linkage* menggabungkan kelompok transfer dan tagihan pada 0,55 alih-alih 0,22: *chaining* melalui transfer terjadwal.

Buku: Subbab 8.16, Soal 8.3, 8.5, dan 8.6.

Menerapkannya dalam Kenyataan

Visual: tiga hal terpenting per layar, dalam urutan visual; kedekatan dan wilayah bersama; satu warna untuk “dapat diketuk”; skala modular; skala jarak; uji menyipit.

Warna: token; setiap pasangan yang dipakai; `contrast.py` pada setiap perubahan; jangan warna saja; uji latar tembus pandang dan bergambar dengan isi sungguhan.

Sistem: kecil, berbasis platform, punya pemilik.

IA: *card sorting* terbuka dengan 15–30 pengguna; Persamaan (8.5) dan Persamaan (8.6); label dari nama tumpukan; cantumkan silang butir yang terbelah; *tree testing* sebelum layar.

Contoh: situs puskesmas (“imunisasi”, “posyandu”, “KIA” bersama), papan menu warung, pengaturan ojek yang diuji bersama pengemudi.

Buku: Subbab 8.11.

IMK dalam Praktik: Profitina

Dasbor sebagai kokpit: satu permintaan mengumpulkan notifikasi, kanal yang diikuti, peringkat, dan aktivitas, masing-masing dapat diklik ke bagiannya — diorganisasi menurut *apa yang membutuhkan saya sekarang*.

Penghitung yang bermakna: "N baru" hanya menghitung kiriman tingkat atas; menghitung balasan akan membesar-besarkan janji.

Arsitektur untuk mesin juga: situs dwibahasa yang menerbitkan `/llms.txt` untuk mesin jawaban AI, di samping data terstruktur schema.org.

Organisasi yang jelas dan label yang jujur melayani manusia dan sistem AI yang membaca atas nama mereka. Buku: Subbab 8.12.

Kesalahan yang Sering Terjadi

Semuanya sama keras; warna ikon acak; celah yang tidak mengelompokkan; terlalu banyak ukuran huruf.

Kontras dinilai dengan mata; warna sebagai satu-satunya tanda; gaya platform tanpa pengaturan aksesibilitasnya.

Design system tanpa token atau pemilik.

Mengorganisasi menurut bagan organisasi; label internal; memaksa butir yang terbelah ke satu tempat.

Buku: Subbab 8.13.

Ringkasan

1. Pengelompokan Gestalt dan hierarki membuat struktur terlihat (Gambar 8.1).

2. Huruf dari skala modular (Persamaan (8.1)); jarak dari skala 4 atau 8 px dan sebuah grid.

3. WCAG 2.2 (ISO/IEC 40500:2025): 4,5:1 teks, 3:1 teks besar dan UI; kontras dari Persamaan (8.2), Persamaan (8.3), Persamaan (8.4); WCAG 3.0 masih draf.

4. Pemeriksa membaca token W3C: enam dari sepuluh pasangan gagal (Gambar 8.2).

5. Token, komponen, pedoman (Gambar 8.3); Material 3 Expressive, Liquid Glass, Fluent 2.

6. *Card sorting* dengan Persamaan (8.5) dan Persamaan (8.6): empat kelompok, angsuran di bawah tagihan (Gambar 8.4).

Buku: Subbab 8.14.

Pertemuan Berikutnya

Bab 9 — Gaya Interaksi: GUI, Sentuh, Suara, Gestur.

Sebelum itu: kerjakan Soal Latihan 8.1, 8.3, 8.5, dan 8.7; jalankan `contrast` pada warna aplikasi Anda sendiri dan lakukan *card sorting* kecil bersama lima teman.

Aplikasi bank empat tab akan kembali ketika kita merancang untuk sentuhan dan ibu jari. Buku: Subbab 8.15 dan 8.16.

Kuliah 9 — Gaya Interaksi: GUI, Sentuh, Suara, Gestur

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 9

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita membahas cara-cara seseorang memberi tahu komputer apa yang harus dilakukan — perintah, menu, formulir, manipulasi langsung, bahasa alami — beserta saluran yang membawanya: penunjuk, sentuhan, gestur, suara, dan tatapan. Kita membangun satu inti pesanan warung dengan tiga antarmuka dan mengukur sebuah kios di peramban sungguhan. Buku: Bab 9, Subbab 9.1–9.14.

Yang Dibahas dalam Pertemuan Ini

1. **Meja kasir warung** — lima cara memesan segelas teh, dan apa yang dituntut masing-masing dari pembeli.
2. **Teorinya** — dua jurang, manipulasi langsung, peristiwa, dan kecepatan umpan balik.
3. **Sentuhan** — hukum Fitts, hukum FFitts, ukuran target di WCAG 2.2 serta di Apple dan Android.
4. **Gestur dan suara** — perintah yang tak terlihat, pendengaran yang tak pasti, dan kapan harus bertanya.
5. **Membangun dan mengukur** — satu inti pesanan, baris perintah, baris suara yang diketik, kios sentuh dalam Flask dan Java, serta audit ukuran target dengan Playwright.

Setiap program ada dalam Java dan Python, dan setiap keluaran yang ditampilkan adalah keluaran sungguhan. Buku: Bab 9.

Meja Kasir Bu Tini

Warung dekat kampus di Surabaya: lima menu, kerumunan tengah hari berisi mahasiswa, pengemudi, dan pegawai kantor.

"*Es teh dua, Bu!*" dari pintu — suara, dan "*Dua, ya?*" dari Bu Tini adalah konfirmasi lisan.

Pendatang baru menunjuk kartu menu — pemilihan menu lewat gestur; pegawai kantor menulis slip pesanan — formulir.

Keponakannya, Dimas, ingin membuat kios tablet. Seberapa besar tombolnya? Apa yang dilakukan kios bila tidak paham? Bagaimana dengan pelanggan lansia?

81,72% penduduk Indonesia — sekitar 235 juta orang — memakai internet pada 2026 (APJII), dan setiap orang menjumpai banyak gaya interaksi setiap hari. Buku: Subbab 9.1.

Apa Itu Gaya Interaksi

Bagaimana saya mengatakannya? Mengetik kata, memilih dari daftar, mengisi bagian kosong, bertindak atas objek, berbicara.

Bagaimana saya melihat hasilnya? Teks, gambar yang berubah, bunyi, getaran.

Lima gaya utama (Shneiderman): **bahasa perintah, pemilihan menu, pengisian formulir, manipulasi langsung, bahasa alami**.

Sentuhan, gestur, suara, dan tatapan adalah **modalitas** — saluran yang membawa gaya-gaya itu.

Aplikasi ponsel sebagian besar adalah manipulasi langsung dan menu lewat sentuhan; voice assistant adalah bahasa alami lewat ucapan; Vision Pro adalah manipulasi langsung lewat mata dan cubitan. Buku: Subbab 9.2.1.

Peta Gaya-Gaya Interaksi

Tabel 9.1 Lima gaya interaksi utama serta di mana masing-masing unggul dan lemah.

Gaya	Pengguna…	Unggul ketika	Lemah ketika
Bahasa perintah	mengetik sintaksis tetap	pengguna ahli, pengulangan	pemula, salah sintaksis
Pemilihan menu	memilih dari daftar	pengguna sesekali	pilihan banyak
Pengisian formulir	mengisi bagian kosong berlabel	data terstruktur	ruas tak jelas
Manipulasi langsung	bertindak atas objek terlihat	melihat dan membatalkan	tindakan abstrak
Bahasa alami	berbicara atau menulis bebas	tangan dan mata sibuk	ambiguitas, bising

Tabel 9.1: pertanyaan penentunya adalah apa yang harus diingat dan apa yang harus dilihat pengguna. Mengenali (menu) lebih mudah daripada mengingat (perintah); mengingat lebih cepat bagi ahli. Buku: Subbab 9.2.2 dan Tabel 9.1.

Segelas Teh, Lima Cara

Perintah: `add esteh 2` — 13 ketukan tombol; harus tahu kode, kata kerja, dan urutannya.

Menu: tekan 1, 1, 2 — tiga tekanan; membaca, tetapi tidak mengingat apa pun.

Formulir: tulis 2 pada baris “Es teh” — satu ruas, dan seluruh menu terlihat.

Manipulasi langsung: ketuk “Es teh” dua kali, lihat “2 x Es teh” muncul; minus membatalkan.

Bahasa alami: ucapkan “*es teh dua*” — tiga kata, tetapi kini sistem harus *memahami*.

Gaya yang sedikit menuntut ingatan pengguna banyak menuntut layar; gaya yang paling sedikit menuntut pengguna paling banyak menuntut sistem. Buku: Subbab 9.2.3.

Dua Jurang, dan Manipulasi Langsung

Jurang eksekusi: dari “saya mau dua teh” ke apa yang boleh saya lakukan di sistem.

Jurang evaluasi: dari apa yang ditampilkan sistem ke “pesanan saya sudah masuk?”

Manipulasi langsung mempersempit keduanya: objek tetap terlihat; tindakan fisik, cepat, bertahap, dan dapat dibatalkan; akibatnya langsung tampak.

Harganya: ruang layar, dan tindakan abstrak (“setiap Jumat, sepuluh teh”) canggung.

Jurang-jurang Norman dari Bab 1 menjelaskan mengapa perintah terasa sulit bagi pemula dan mengapa lembar kerja serta peta ojek daring terasa mudah. Buku: Subbab 9.3.1.

Peristiwa, Keadaan, dan Kecepatan Umpan Balik

Setiap antarmuka GUI, sentuh, atau suara menunggu **peristiwa**, menjalankan **penangan**, mengubah **keadaan**, menggambar ulang.

Kios: keadaan = keranjang; peristiwa = tambah, kurangi, bayar. Baris suara menambah satu keadaan: pesanan yang *menunggu konfirmasi*.

0,1 detik terasa seketika; **1 detik** menjaga alur pikiran; **10 detik** batas perhatian.

Lewat 1 detik tunjukkan aktivitas; lewat 10 detik tunjukkan kemajuan dan biarkan pengguna mengerjakan hal lain.

Ketiga batas itu berasal dari Miller (1968) dan dirangkum Nielsen; itulah sebabnya aplikasi yang baik langsung bereaksi dan mengerjakan pekerjaan lambat belakangan. Buku: Subbab 9.3.2.

Hukum Fitts

$$MT = a + b \log_2\left(\frac{D}{W} + 1\right) \tag{9.1}$$

MT: waktu mengenai target selebar W pada jarak D; a, b diukur per perangkat dan orang.

Logaritmanya adalah **indeks kesulitan** ID, dalam bit.

Memperkecil lebar menjadi setengahnya kira-kira sama mahalnya dengan menggandakan jarak: **target kecil adalah target lambat**.

Persamaan (9.1) adalah formulasi Shannon yang dipakai di IMK; Bab 2 menurunkannya untuk tetikus. Buku: Subbab 9.3.3.

Jari Bukan Penunjuk Tetikus: Hukum FFitts

$$ID_f = \log_2\left(\frac{D}{\sqrt{2\pi e\left(\sigma^2 - \sigma_a^2\right)}} + 1\right) \tag{9.2}$$

Jari menyentuh bidang selebar beberapa milimeter dan menutupi targetnya sendiri.

Bi, Li, dan Zhai: sentuhan memiliki **ketidaktepatan mutlak** σ_a yang tidak dapat dihapus kehati-hatian.

Di bawah ukuran tertentu, target yang lebih kecil tidak sekadar memperlambat — ia menimbulkan **kesalahan**.

Persamaan (9.2) hanya menyimpan sebaran yang terkait kecepatan di dalam indeks; sebaran jari sendirilah alasan setiap platform sentuh menetapkan ukuran target minimum. Buku: Subbab 9.3.3.

Contoh: Tautan Bahasa

Versi pertama kios menggambar tautan bahasanya setinggi 13 px; tombol menu 64 px. Jarak sekitar 300 px.

Tautan: $ID = \log_2(300/13 + 1) = \log_2(24.08) = 4.59$ bit.

Tombol: $ID = \log_2(300/64 + 1) = \log_2(5.69) = 2.51$ bit.

Tautan itu hampir **dua bit lebih sulit**, dan dekat dengan sebaran jari sendiri.

Dihitung dengan tangan memakai Persamaan (9.1); audit kemudian menemukan tautan yang sama secara otomatis. Buku: Subbab 9.3.3, Contoh.

Ukuran Target Minimum, September 2026

Tabel 9.2 Ukuran minimum target sentuh dan penunjuk per September 2026.

Sumber	Target minimum	Tingkat
WCAG 2.2, KK 2.5.8	24 × 24 px CSS, atau pengecualian jarak	AA
WCAG 2.2, KK 2.5.5	44 × 44 px CSS	AAA
Apple HIG	44 × 44 pt	pedoman
Material Design 3	48 × 48 dp (≈ 9 mm)	pedoman

Tabel 9.2: WCAG 2.2 menjadi ISO/IEC 40500:2025 pada Oktober 2025. Pengecualian jarak 2.5.8 meloloskan target kecil bila lingkaran 24 px padanya tidak menyentuh target lain. Buku: Subbab 9.3.3 dan Tabel 9.2.

Gestur: Ampuh tetapi Tak Terlihat

Usap untuk menghapus, cubit untuk memperbesar, tekan lama untuk menu — cepat bagi yang tahu, tak terlihat bagi yang tidak.

Tidak ada **penanda** (*signifier*) yang berkata “Anda bisa mengusap di sini”.

Aturan: gestur adalah **pintasan, bukan satu-satunya jalan**; ikuti konvensi platform; jangan menyingkirkan orang.

WCAG 2.2: **2.5.1 Pointer Gestures** (alternatif satu penunjuk), **2.5.7 Dragging Movements** (baru: alternatif ketukan untuk menyeret).

Peta yang hanya dapat dicubit atau daftar yang hanya dapat diseret gagal; tombol plus/minus dan naik/turun memperbaikinya. Buku: Subbab 9.3.4.

Suara: Alami, tetapi Ambigu

Cocok untuk tangan dan mata yang sibuk, serta bagi banyak pengguna lansia dan berpenglihatan rendah.

Sejak 4 September 2026 Gemini secara bertahap menggantikan Google Assistant di ponsel Android, jam, dan Android Auto.

Tiga masalah: **tanpa menu yang terlihat**, **ketidakpastian** (“*es teh dua*”: dua teh, atau teh untuk meja dua?), ucapan yang **sekejap**.

Jawabannya: konfirmasi sebanding dengan risiko dan keraguan; katakan apa yang bisa diucapkan; sediakan jalan kembali.

Riset CHI 2026 menjajaki voice assistant yang membantu orang dewasa dengan disabilitas intelektual bertindak mandiri. Buku: Subbab 9.3.5.

Konfirmasi Implisit atau Eksplisit

$$c = \frac{k}{m} \tag{9.3}$$

Implisit: ulangi dan lanjutkan — “Dua es teh — ada lagi?”

Eksplisit: berhenti dan tanya — “Maksudnya satu kopi? Katakan ya untuk memastikan.”

Keyakinan kita c: kata yang dipahami k dibagi kata yang terdengar m; di bawah 0,6, bertanya.

Tindakan yang mahal (membayar, menghapus): **selalu** eksplisit.

Persamaan (9.3) adalah pengganti yang dapat dijelaskan bagi keyakinan pengenal ucapan. Antarmuka suara yang tak pernah bertanya sedang menebak; yang selalu bertanya melelahkan. Buku: Subbab 9.3.5.

Bahasa Desain Masa Kini dan Ruang

Liquid Glass (iOS 26, 15 September 2025): kontrol tembus pandang yang membiaskan konten di belakangnya.

Material 3 Expressive (Mei 2025): 46 studi, 18.000+ partisipan; elemen kunci ditemukan hingga 4× lebih cepat.

Fluent 2: sistem bersama Microsoft untuk Windows dan Microsoft 365.

visionOS: tatap untuk membidik, cubit untuk bertindak — hukum Fitts dengan mata sebagai penunjuk.

Ikuti kontrol dan gestur platform; periksa keterbacaan kaca dan warna (Bab 11 mengukur kontras). Buku: Subbab 9.3.6.

Lembar Kerja Memilih Gaya

1 Pengguna · 2 Konteks · 3 Bentuk tugas — siapa, di mana, berapa pilihan, konkret atau abstrak.

4 Gaya utama · 5 Jalur kedua — gaya yang cocok, dan alternatif bagi yang tidak terlayani.

6 Umpan balik · 7 Kesalahan — apa yang terlihat dalam 0,1/1/10 detik; bagaimana kekeliruan terlihat dan dibatalkan.

8 Periksa — ukur ukuran target, alternatif gestur, dan kontras.

Diisi untuk kios: kios sentuh bertombol besar, pesanan lisan atau ketikan dengan konfirmasi, dan pembantu di meja kasir sebagai jalur kedua. Buku: Subbab 9.4 dan Tabel 9.3.

Satu Inti, Tiga Antarmuka



Gambar 9.1 Satu inti pesanan dengan tiga antarmuka yang berbagi menu, keranjang, dan total.

Gambar 9.1: antarmuka hanya menerjemahkan antara manusia dan inti; harga, aturan, dan total sama di setiap gaya, dan gaya baru dapat ditambahkan tanpa menyentuh inti. Buku: Subbab 9.5.

Pengurai Suara dalam Tiga Langkah Pikiran

Pindai kata dari kiri ke kanan; pada setiap posisi ambil **alias terpanjang** (“nasi goreng” mengalahkan yang lebih pendek).

Kata bilangan tepat sebelum atau tepat sesudah menu adalah jumlahnya.

Kata niat (“*batal*”, cancel) mengubah menjadi hapus; kata pengisi (“*tolong*”, “*ya*”) dipahami tetapi kosong makna.

Kode 9.1 Dari `order_core.py`: keyakinan menurut Persamaan (9.3) adalah bagian kata yang dipahami.

```
    conf = sum(known) / len(known) if known else 0.0
    return intent, items, round(conf, 2)
```

Kode 9.1 mengakhiri pengurai; versi Java di `OrderCore.java` membulatkan dengan cara yang sama. Buku: Subbab 9.5.1.

Bertanya Sebelum Bertindak

Kode 9.2 Dari `cli.py`: di bawah ambang, antarmuka menyimpan pesanan dan bertanya.

```
            if conf < CONFIRM_BELOW:
                # Keyakinan rendah: tanya dulu sebelum bertindak.
                pending = (intent, items)
                print(f"  -> Did you mean {heard}? Say 'yes' to confirm.")
                continue
```

Pesanan yang disimpan menunggu di `pending`; "*ya*", "*iya*", atau "yes" menuntaskannya; ucapan lain membuangnya.

Tanpa menu sama sekali → "Sorry, which menu item? Please say its name."

Kode 9.2 mengubah "Dua, ya?" Bu Tini menjadi kode. Buku: Subbab 9.5.2.

Menjalankan Sesi

```
> add esteh 2
  total Rp10.000
> add nasgor
  total Rp25.000
> add kopi x
  error: quantity must be a number: x
> say tolong nasi goreng dua sama es teh satu ya
  heard: add [2 x Fried rice, 1 x Iced tea] confidence 1.0
  -> OK. Total Rp60.000
> say minta yang anget anget dong
  heard: add [nothing] confidence 0.6
  -> Sorry, which menu item? Please say its name.
> say kopi item yang kentel banget
  heard: add [1 x Coffee] confidence 0.4
  -> Did you mean 1 x Coffee? Say 'yes' to confirm.
> say ya
  -> OK. Total Rp66.000
> remove esteh 1
  total Rp61.000
> show
   2 x Iced tea        Rp10.000
   3 x Fried rice      Rp45.000
   1 x Coffee           Rp6.000
  total Rp61.000
```

Perintah, salah sintaksis, ucapan yang dipahami penuh (keyakinan 1,0), ucapan tanpa menu (ditanya namanya), dan “kopi item yang kentel banget” dengan $c = 2/5 = 0.4$*, dikonfirmasi dengan “ya”. Java mencetak baris yang sama. Buku: Subbab 9.6.*

Kios dan Auditnya

Formulir HTML biasa, pola *post/redirect/get*; berjalan di ponsel termurah, dengan papan ketik, dan dengan pembaca layar.

Flask pada porta 4090, `HttpServer` JDK pada 4091 — **halaman identik bita demi bita**.

`audit_targets.py` membuka kios di Chromium pada 360 px, mengetuk seperti pembeli, mengukur setiap target.

```
button Iced tea Rp5.000        159 x 64  ok
button Coffee Rp6.000          159 x 64  ok
button Fried rice Rp15.000     159 x 64  ok
button Fried noodles Rp13.000  159 x 64  ok
button Fried tempeh Rp2.000    159 x 64  ok
button Remove one: Iced tea     48 x 48  ok
button Remove one: Fried rice   48 x 48  ok
button Pay                     328 x 56  ok
a      Bahasa Indonesia         99 x 13  below WCAG 2.5.8 (AA)
9 targets; smaller than 24 px: 1, 44 px: 1, 48 px: 1
```

Tautan bahasa 99 × 13 px berada di bawah ketiga ukuran; mungkin lulus 2.5.8 berkat jarak, tetapi gagal pada 44/48 dan menjadi target yang meleset bagi pelanggan lansia. Buku: Subbab 9.5.3, 9.5.4, dan 9.6.

Perbaikannya, Terukur

Satu aturan CSS: `.lang { display: inline-block; min-height: 48px; padding: 12px 16px; }`.

```
button Iced tea Rp5.000        159 x 64  ok
button Coffee Rp6.000          159 x 64  ok
button Fried rice Rp15.000     159 x 64  ok
button Fried noodles Rp13.000  159 x 64  ok
button Fried tempeh Rp2.000    159 x 64  ok
button Remove one: Iced tea     48 x 48  ok
button Remove one: Fried rice   48 x 48  ok
button Pay                     328 x 56  ok
a      Bahasa Indonesia        173 x 48  ok
9 targets; smaller than 24 px: 0, 44 px: 0, 48 px: 0
```



Gambar 9.2 Kios akhir pada 360 px setelah tiga ketukan, dengan bidang sentuh 48 px di mana-mana.

Gambar 9.2 adalah tangkapan layar yang diambil audit dari kios kita sendiri yang sedang berjalan. Buku: Subbab 9.8.1.

Pengujian

`pytest`: total, pengurangan berlebih, bilangan sebelum dan sesudah menu, niat membatalkan, keyakinan rendah, jumlah salah — 4 pengujian lulus.

`OrderCoreTest.java`: tujuh pemeriksaan yang sama, “all passed”.

Dua pemeriksaan dengan tangan: kedua peladen memberi MD5 yang sama untuk halaman kosong; audit mencetak sembilan baris yang sama terhadap keduanya.

Buku: Subbab 9.7.

Di Mana Pengurai Masih Gagal

"*es teh manis dua gelas*" → satu es teh dengan keyakinan 0,6: dijalankan, dan **salah**.

"*berapa totalnya*" → akhiran *-nya* menyembunyikan kata *total*.

Keyakinan mengukur seberapa banyak yang dipahami, bukan apakah benar.

Pelanggan Indonesia mencampur imbuhan, bahasa Jawa, dan Inggris — kumpulkan ucapan nyata.

Soal Latihan 9.4 dan 9.5 memperbaiki keduanya; pembayaran selalu harus dikonfirmasi eksplisit. Buku: Subbab 9.8.2.

Memakainya Sungguhan

Ganti `menu.csv` dengan menu Anda **dan kata-kata yang diucapkan pelanggan Anda**.

Bot WhatsApp atau Telegram adalah satu antarmuka lagi bagi `parse_utterance`.

Pertahankan pembantu di meja kasir: kios hendaknya memperpendek antrean, bukan menggantikan manusia.

Audit pada 320 dan 360 px di ponsel termurah, di bawah matahari, dengan ibu jari.

Buku: Subbab 9.9.

IMK dalam Praktik: Profitina

Dasbor sebagai kokpit: satu permintaan mengumpulkan notifikasi, kanal, peringkat, dan aktivitas; setiap butir satu ketukan dari bagiannya — manipulasi langsung atas aliran peristiwa.

Pengiriman optimistis: kotak tulis langsung kosong dan kiriman muncul; jika gagal, teks dipulihkan — umpan balik dalam 0,1 detik, kegagalan dapat dipulihkan.

Share to X manual, per butir, dengan antrean dan kuota harian — tindakan publik tetap di bawah kendali eksplisit.

Mesin konten mengusulkan draf (bahasa alami) dan pengguna menyunting serta memutuskan (manipulasi langsung). Buku: Subbab 9.10.

Kesalahan yang Sering Terjadi

Satu gaya untuk semua orang; gestur tanpa alternatif.

Mengukur target dengan mata; menyamakan pengecualian jarak dengan kenyamanan.

Menebak alih-alih bertanya; menganggap keyakinan sebagai kebenaran.

Logika diduplikasi di antarmuka; umpan balik lebih lambat dari 0,1 detik.

Buku: Subbab 9.11.

Ringkasan

Lima gaya utama, dibawa oleh sentuhan, gestur, suara, dan tatapan (Tabel 9.1).

Manipulasi langsung mempersempit kedua jurang; batas umpan balik 0,1 / 1 / 10 detik.

Fitts (Persamaan (9.1)) dan FFitts (Persamaan (9.2)); target 24 px AA, 44 AAA/Apple, 48 Material (Tabel 9.2).

Suara: konfirmasi ketika ragu (Persamaan (9.3)) atau ketika mahal.

Buku: Subbab 9.12.

Soal untuk Dicoba

9.1 Indeks kesulitan pada 24, 44, dan 48 px untuk $D = 300$ px.

9.3 Implisit atau eksplisit: menambah teh, membayar Rp61.000, membatalkan semuanya.

9.4 Buat "*es teh manis dua gelas*" menghasilkan dua teh.

9.7 Kriteria WCAG 2.2 mana yang dilanggar aplikasi yang hanya bisa dicubit dan diseret?

Penyelesaian lengkapnya ada di buku. Buku: Subbab 9.13 dan 9.14.

Pertemuan Berikutnya

Bab 10 — UX Seluler dan Responsif. Halaman pemesanan sebuah pangkalan ojek, dirancang di desktop dan nyaris tak dapat dipakai di ponsel.

Viewport dan piksel CSS, tata letak *mobile-first*, ibu jari, formulir ponsel, Core Web Vitals, dan anggaran kinerja.

Sebelum itu: jalankan kios, audit pada 320 px, dan coba Soal Latihan 9.1 dan 9.4. Buku: Bab 10.

Kuliah 10 — UX Seluler dan Responsif

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 10

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita merancang untuk satu tangan, layar sempit, dan jaringan yang tak menentu. Kita membaca statistik seluler Indonesia dengan cermat, mempelajari mengapa halaman desktop menciut menjadi sepertiganya di ponsel, dan membangun ulang halaman pemesanan sebuah pangkalan ojek secara mobile-first, lalu mengukurnya pada empat lebar dan tiga jaringan. Buku: Bab 10, Subbab 10.1–10.14.

Yang Dibahas dalam Pertemuan Ini

1. **Ponsel Mas Yoga** — halaman pemesanan yang berfungsi di monitor dan gagal di bawah pohon.
2. **Membaca angka** — APJII, StatCounter, dan Ookla mengukur hal yang berbeda.
3. **Piksel dan *viewport*** — mengapa halaman tanpa satu baris menjadi 37% ukurannya.
4. **Tata letak, ibu jari, dan formulir** — kisi cair, titik henti pilihan konten, jangkauan ibu jari, jenis masukan, dan *autocomplete*.
5. **Kecepatan** — Core Web Vitals, perjalanan pulang-pergi, lebar pita, dan anggaran kinerja, diukur dengan program kita sendiri.

Buku: Bab 10.

Ponsel Mas Yoga

Pangkalan ojek Pak Darto di Jalan Kenari: bangku, pohon, empat pengemudi; pelanggan ingin memesan lebih dahulu.

Halaman buatan sang keponakan tampak rapi di desktop: tabel pengemudi, formulir empat ruas, tautan Telepon dan WhatsApp.

Di ponsel Mas Yoga: kartu pos kelabu mungil, gulir ke samping, papan ketik huruf untuk nomor telepon, tautan “Telepon” terlalu kecil bagi ibu jari.

Bu Ani menyerah dan menelepon — halaman itu menjadikan panggilan telepon satu-satunya antarmuka yang andal.

Tak ada yang salah menurut ukuran desktop; halaman itu mengandaikan layar lebar, tetikus, meja, dan kabel cepat. Buku: Subbab 10.1.

Responsif dan *Mobile-First*

Responsif: satu halaman yang menyesuaikan diri — kisi cair, gambar lentur, *media query* (Marcotte, 2010).

***Mobile-first*:** rancang kasus terkecil dan terlemah lebih dahulu, lalu *tambahkan* tata letak untuk layar lebih besar (Wroblewski).

Ponsel lebih dari sekadar layar kecil: satu ibu jari, interupsi, sinar matahari, jaringan berubah-ubah, prosesor lambat.

Halaman yang jelas pada 360 px juga jelas pada 1280 px.

Buku: Subbab 10.2.

Tiga Sumber, Tiga Hitungan Berbeda

APJII (Mei 2026): 81,72% penduduk, sekitar 235 juta, memakai internet — menghitung **orang**.

StatCounter (Agustus 2026): 38,98% tayangan halaman dari seluler, 60,59% desktop, 0,43% tablet; Android 77,07%, iOS 22,88% dari seluler — menghitung **tayangan halaman di situs yang memasang pelacaknya**.

Ookla lewat Bloomberg Technoz (Februari 2026): median unduh seluler 59,18 Mbit/s, unggah 17,97, latensi 21 ms — sebuah **median**, bukan ekor sebaran.

Selalu tanyakan: *apa* yang dihitung, *di mana*, *bagaimana*?

Tayangan halaman desktop dapat mendominasi padahal sebagian besar orang memakai ponsel di aplikasi; median menyembunyikan sepersepuluh yang paling lambat. Buku: Subbab 10.3.1.

Piksel CSS dan Piksel Perangkat

$$w_{css} = \frac{w_{dev}}{r} \tag{10.1}$$

r adalah rasio piksel perangkat (2, 3, atau lebih pada ponsel masa kini).

Ponsel selebar 1080 piksel perangkat dengan $r = 3$ memberi 360 piksel CSS.

Perancang bekerja dalam piksel CSS — “ponsel 360 piksel”.

Persamaan (10.1): peramban memetakan satu piksel CSS ke $r \times r$ piksel perangkat. Buku: Subbab 10.3.2.

Viewport, dan Halaman yang Menciut

$$s = \frac{w_{vp}}{w_{layout}} \tag{10.2}$$

$$h_{eff} = s \cdot h_{css} \tag{10.3}$$

Tanpa `<meta name="viewport" content="width=device-width, initial-scale=1">` ponsel menata halaman pada **980 px** lalu memperkecilnya.

Setiap ukuran dikalikan s pada Persamaan (10.2); yang dijumpai pengguna adalah h_{eff} pada Persamaan (10.3).

Satu baris di kepala halaman menentukan apakah sisa kerja seluler terlihat sama sekali. Buku: Subbab 10.3.2.

Contoh: Halaman Pertama Pak Darto

Ponsel 360 px, tata letak 980 px: $s = 360/980 = 0.367$.

Tulisan kecil 11 px → $0.367 \times 11 = 4.04$ px di layar.

Tautan “Telepon” 12 px → $0.367 \times 12 = 4.4$ px sebagai target.

Ponsel 320 px: $s = 0.327$, dan tulisan itu menciut menjadi 3,6 px.

Kira-kira setinggi tanda koma. Buku: Subbab 10.3.2, Contoh.

Tata Letak Cair dan Titik Henti

Mulai dengan satu kolom, satuan relatif (`%`, `rem`, `fr`), `max-width` untuk baris panjang.

Media query menambahkan tata letak bila ada ruang: `@media (min-width: 700px) { … }`.

Container query membuat komponen menyesuaikan diri dengan wadahnya — kartu di bilah samping dan di kolom utama.

Pilih titik henti di tempat **konten** membutuhkannya, bukan pada lebar ponsel tahun lalu.

Untuk halaman pemesanan, satu kolom enak dibaca hingga sekitar 700 px; di atasnya, kartu berdampingan lebih hemat ruang. Buku: Subbab 10.3.3.

Tangan dan Ibu Jari

Hoober mengamati 1.333 orang: **49%** satu tangan dan ibu jari, **36%** ditopang dengan jari tangan lain, **15%** dua tangan.

Orang sering berganti pegangan, bahkan dalam satu tugas.

Tindakan yang sering dan penting **dalam jangkauan ibu jari** — layar bagian bawah dan tengah; yang jarang atau merusak lebih jauh.

Halaman responsif meletakkan Telepon dan WhatsApp di bilah bawah, masing-masing separuh lebar, setinggi 52 px.

Ukuran target Bab 9 berlaku tanpa perubahan: sekurang-kurangnya 24 px CSS (AA), 44–48 agar nyaman. Buku: Subbab 10.3.4.

Formulir di Ponsel

Jenis masukan memilih papan ketik: `type="tel"` → papan angka; `type="email"` → tombol @.

`autocomplete` menyatakan arti ruas: `name`, `tel`, `street-address` → pengisian satu ketukan.

`inputmode` memilih papan ketik tanpa mengubah ruas, misalnya `numeric`.

`<label>` sungguhan di atas setiap ruas, dihubungkan dengan `for`/`id`; *placeholder* lenyap saat mengetik.

Kriteria WCAG 2.2 1.3.5 Identify Input Purpose (AA) mensyaratkan tujuan ruas data pribadi dapat dibaca mesin. Buku: Subbab 10.3.5.

Core Web Vitals

LCP (Largest Contentful Paint) ≤ **2,5 detik**.

INP (Interaction to Next Paint) ≤ **200 ms** — menggantikan First Input Delay pada Maret 2024.

CLS (Cumulative Layout Shift) ≤ **0,1**.

Dinilai “baik” pada **persentil ke-75** kunjungan nyata.

Buku: Subbab 10.3.6.

Perjalanan Pulang-Pergi, Bita, dan Anggaran

$$t \approx k \cdot RTT + \frac{8S}{B} \tag{10.4}$$

k perjalanan pulang-pergi (DNS, TCP, TLS, permintaan) masing-masing RTT detik, lalu S kB pada B kbit/s.

Profil seluler Lighthouse: RTT 150 ms, 1.638,4 kbit/s, CPU 4× lebih lambat — jauh lebih lambat daripada median, dengan sengaja.

Anggaran kinerja: tetapkan bobot maksimum agar Persamaan (10.4) tetap dalam 2,5 detik pada profil yang dipilih.

Dua hal yang dikendalikan perancang: berapa perjalanan pulang-pergi dan berapa bita. Buku: Subbab 10.3.6.

Kriteria WCAG yang Terasa di Ponsel

1.4.10 Reflow (AA): dapat dipakai pada 320 px CSS tanpa gulir dua arah.

1.4.4 Resize Text (AA): teks hingga 200% tanpa kehilangan.

1.3.4 Orientation (AA): tidak dikunci tegak atau mendatar kecuali esensial.

1.3.5 Identify Input Purpose (AA) dan **2.5.8 Target Size (AA)**.

320 px setara jendela desktop 1280 px pada perbesaran 400%, sehingga reflow juga melayani pengguna desktop berpenglihatan rendah. Buku: Subbab 10.3.7.

Daftar Periksa *Mobile-First*

Tabel 10.1 Daftar periksa *mobile-first* dengan hasil halaman pemesanan pertama.

Pemeriksaan	Halaman pemesanan pertama
Baris *viewport*	Tidak ada: halaman diperkecil ke 37%
Reflow pada 320 px	Diperkecil alih-alih ditata ulang
Teks isi ≥ 16 px	Efektif 3,6–4,0 px
Target ≥ 24 px (44–48)	Efektif 3,9–4,4 px
Tindakan utama dalam jangkauan ibu jari	Tautan mungil di atas
Papan ketik dan *autocomplete*	Ruas telepon berupa teks biasa
Bobot dalam anggaran	2.492 kB: 12,8 dtk pada profil uji

Tabel 10.1: sebagian besar baris dapat diotomatiskan; jangkauan ibu jari dan kebermaknaan memerlukan manusia (Bab 5 dan 7). Buku: Subbab 10.4 dan Tabel 10.1.

Dua Templat, Satu Perangkat Kata

`naive.html`: `body { width: 960px }`, fon 13 px, tabel, teks lepas sebelum masukan, ruas telepon berjenis teks, tautan 11 px, tanpa baris *viewport*.

`responsive.html`: gaya dasar *mobile-first*, 16 px, kartu pengemudi, label di atas masukan 48 px, bilah bawah yang tetap, satu *media query* pada 700 px.

`strings.csv` memuat kata bahasa Inggris dan Indonesia; `pages.py` / `Pages.java` mengisi templat mana pun.

Flask pada porta 4100, peladen JDK pada 4101 — bita yang identik.

Buku: Subbab 10.5.1 dan 10.5.2.

Peramban yang Berpura-pura Menjadi Ponsel

`check_responsive.py` membuka setiap halaman pada 320, 360, 768, dan 1280 px; di bawah 700 px sebagai konteks **seluler** dengan sentuhan.

Ia membaca `innerWidth`, `scrollWidth`, fon terkecil, target terkecil, dan jenis ruas telepon.

Kode 10.1 Dari `check_responsive.py`: ukuran diskalakan dengan Persamaan (10.2) sebelum dilaporkan.

```
            scale = w / vw            # < 1 bila peramban memperkecil halaman
            over = "yes" if sw > vw else "no"
```

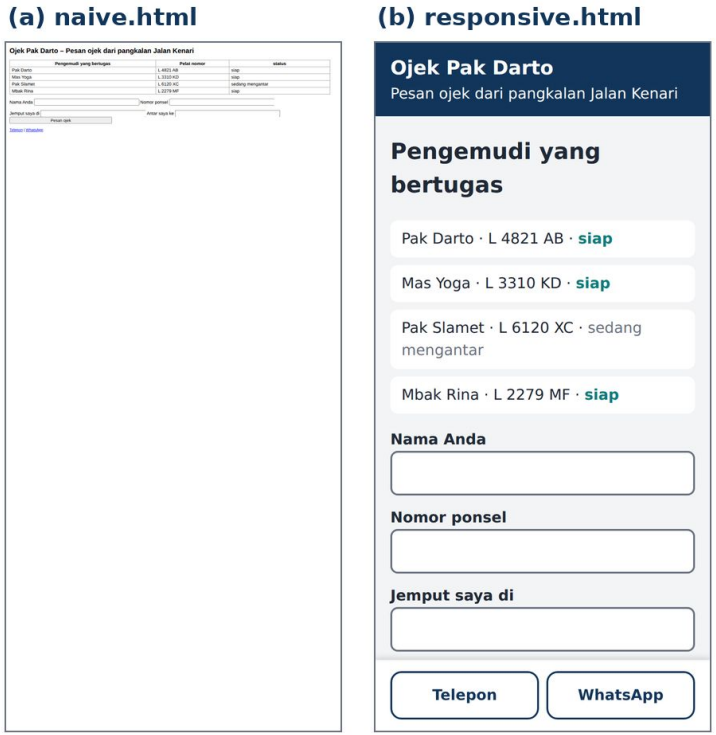
Kode 10.1: laporan menampilkan apa yang dijumpai mata dan ibu jari, seperti pada Persamaan (10.3). Buku: Subbab 10.5.3.

Yang Ditemukan Pemeriksaan

```
naive: width  layout  overflow  text px  target px  phone input
         320     980        no      3.6        3.9  type=text
         360     980        no      4.0        4.4  type=text
         768     768       yes     11.0       12.0  type=text
        1280    1280        no     11.0       12.0  type=text
responsive: width  layout  overflow  text px  target px  phone input
         320     320        no     16.0       48.0  type=tel
         360     360        no     16.0       48.0  type=tel
         768     768        no     16.0       48.0  type=tel
        1280    1280        no     16.0       48.0  type=tel
```

Halaman pertama ditata pada 980 px di ponsel (teks 3,6–4,0 px, target 3,9–4,4 px), meluap ke samping pada 768 px, dan ruas teleponnya berupa teks biasa. Halaman responsif mempertahankan teks 16 px dan target 48 px pada setiap lebar. Buku: Subbab 10.6.

Kedua Halaman di Ponsel



Gambar 10.1 Kedua halaman pemesanan di ponsel 360 piksel: halaman desktop yang menciut dan halaman *mobile-first*.

Gambar 10.1: (a) ditata pada 980 px lalu diciutkan; (b) teks terbaca, ruas berlabel 48 px, dan bilah Telepon/WhatsApp dalam jangkauan ibu jari. Buku: Subbab 10.6.

Perkiraan Waktu Muat

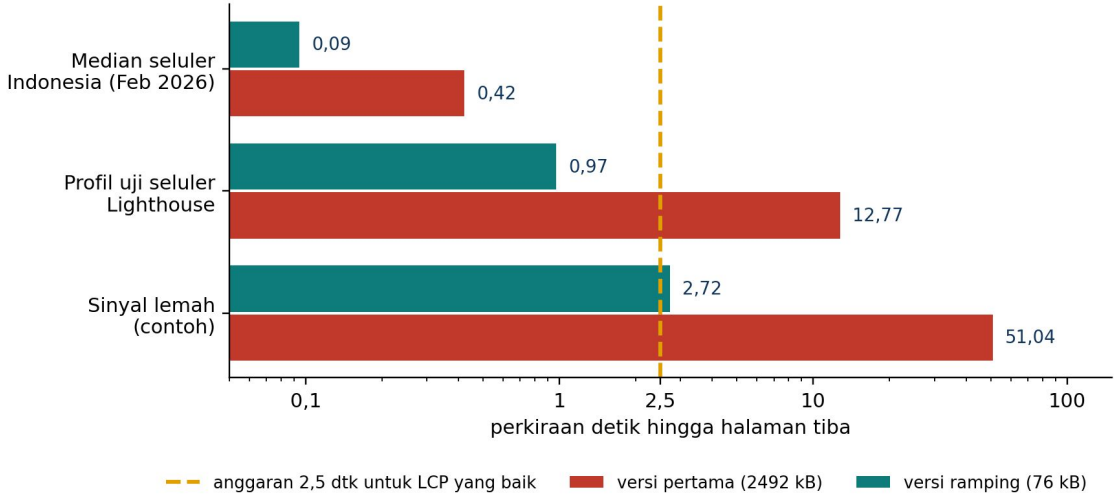
Kode 10.2 `budget.py`: Persamaan (10.4) dengan $k = 4$ perjalanan pulang-pergi.

```python
def seconds(kb, mbps, rtt_ms):
    return ROUND_TRIPS * rtt_ms / 1000 + kb * 8 / (mbps * 1000)
```

```
resources_v1.csv: 2492 kB
  Indonesia median mobile (Ookla Feb 2026)    0.42 s  within 2.5 s
  Lighthouse mobile test profile             12.77 s  OVER 2.5 s
  Weak signal (example: 0.4 Mbit/s)          51.04 s  OVER 2.5 s
resources_v2.csv: 76 kB
  Indonesia median mobile (Ookla Feb 2026)    0.09 s  within 2.5 s
  Lighthouse mobile test profile              0.97 s  within 2.5 s
  Weak signal (example: 0.4 Mbit/s)           2.72 s  OVER 2.5 s
```

Kode 10.2. Periksa dengan tangan: $4 \times 0.150 + 8 \times 2492/1638.4 = 0.6 + 12.17 = 12.77$ *detik. Pada median, halaman berat tiba dalam 0,42 detik — sebabnya tampak baik-baik saja di kota. Buku: Subbab 10.6.*

Enam Perkiraan dalam Satu Bagan



Gambar 10.2 Perkiraan waktu tiba halaman 2.492 kB dan 76 kB pada tiga jaringan; garis putus-putus adalah anggaran 2,5 detik.

Gambar 10.2 memakai sumbu waktu logaritmis. Ujilah di ponsel paling lambat milik pengguna Anda, di luar ruangan, dengan koneksi lemah. Buku: Subbab 10.6.

Membaca Keluaran Baris demi Baris

Tata letak 980 pada 320 dan 360: baris *viewport* hilang — peramban berpura-pura desktop lalu menciutkan.

Luapan “no” di ponsel: bukan keberhasilan — halaman muat hanya karena diperkecil.

Luapan “yes” pada 768: badan tetap 960 px lebih lebar daripada jendela tablet: bibit kegagalan WCAG 1.4.10.

16.0 dan 48.0 di mana-mana: fon dasar dan tinggi ruas halaman responsif bertahan pada setiap lebar.

Angka yang bersih dapat menyembunyikan alasan yang buruk; selalu tanyakan mengapa sebuah pemeriksaan lulus. Buku: Subbab 10.6.

Hormati Kuota Data

Banyak pengguna membeli data per kuota; setiap megabita yang tak perlu adalah uang **pengguna**.

Halaman 2,5 MB yang dibuka dua kali sehari memakai sekitar 150 MB sebulan — untuk satu halaman pemesanan.

Halaman yang tetap berfungsi tanpa gambar melayani pengguna mode hemat data.

Anggaran kinerja adalah aturan kecepatan **sekaligus** aturan keadilan.

Buku: Subbab 10.3.6 dan 10.9.

Pengujian

`pytest` (3 pengujian): rumus terhadap nilai tangan, $0.6+1.0=1.6$ detik; kedua templat dalam kedua bahasa tanpa tempat isian tersisa; *viewport* dan `tel` hanya pada halaman responsif.

`Ch10Test.java`: tujuh pemeriksaan yang sama, “all 7 checks passed”.

Buku: Subbab 10.7.

Membuatnya Lebih Baik

Sinyal lemah: halaman ramping butuh 2,72 detik; fotonya 70 dari 76 kB. Muat belakangan, pakai `srcset`, atau hapus: tanpa foto $t \approx 1.2 + 0.12 = 1.32$ detik.

Kunjungan kedua: 1,2 dari 2,72 detik adalah perjalanan pulang-pergi; *caching* HTTP dan *service worker* kecil melewatinya.

Yang tak terlihat pemeriksaan: urutan ruas, kata-kata, posisi mendatar di dudukan dasbor — bawa ponselnya ke pangkalan dan amati.

Buku: Subbab 10.8.

Memakainya Sungguhan

Rancang pada 360 px, lalu 320, lalu tambahkan titik henti di tempat konten meregang.

Selalu sertakan baris *viewport*.

Tuliskan anggaran: pada profil Lighthouse, paling banyak sekitar **389 kB** untuk 2,5 detik.

Jalankan `check_responsive.py` pada halaman Anda sendiri; pakai jenis masukan dan *autocomplete* di mana-mana.

Hormati kuota: halaman 2,5 MB yang dibuka dua kali sehari menghabiskan sekitar 150 MB sebulan.

Buku: Subbab 10.9.

IMK dalam Praktik: Profitina

Kokpit dasbor muat di ponsel: satu permintaan, setiap butir satu ketukan dari bagiannya.

***Widget* komunitas** berada di bilah samping sempit di profitina.com dan selebar penuh di aplikasi — persoalan *container query*.

Server-Sent Events menjaga satu koneksi panjang: $k \cdot RTT$ pada Persamaan (10.4) dibayar sekali, bukan pada setiap *polling*.

Pengiriman optimistis menjaga interaksi tetap tanggap dalam arti INP di jaringan lambat.

Buku: Subbab 10.10.

Kesalahan yang Sering Terjadi

Tanpa baris *viewport*; lebar tetap; titik henti menurut perangkat.

Placeholder sebagai label; papan ketik yang salah.

Hanya menguji pada median; interaksi yang hanya lewat *hover*.

Mengutip statistik tanpa bertanya apa yang dihitung.

Buku: Subbab 10.11.

Ringkasan

Desain responsif dan *mobile-first*; baca APJII, StatCounter, dan Ookla menurut apa yang dihitungnya.

$w_{css} = w_{dev} / r$ (Persamaan (10.1)); tanpa baris *viewport* ukuran menciut dengan s (Persamaan (10.2), Persamaan (10.3)).

Ibu jari, formulir, Core Web Vitals; $t \approx k \cdot RTT + 8S/B$ (Persamaan (10.4)) dan anggaran.

Terukur: teks 4 px dan 2,5 MB lawan teks 16 px, target 48 px, dan 76 kB (Tabel 10.1).

Buku: Subbab 10.12.

Soal untuk Dicoba

10.1 Ukuran efektif teks 13 px di ponsel 320 px dan 412 px.

10.2 Halaman terbesar dalam 2,5 detik pada profil Lighthouse dan pada sinyal lemah.

10.3 Jawab rekan yang ingin “desktop lebih dahulu, karena 60% adalah desktop”.

10.5 `type`, `autocomplete`, dan `inputmode` terbaik untuk keempat ruas pemesanan.

Penyelesaian lengkapnya ada di buku. Buku: Subbab 10.13 dan 10.14.

Pertemuan Berikutnya

Bab 11 — Aksesibilitas dan Desain Inklusif (WCAG 2.2). Kios antrean puskesmas yang menyingkirkan empat pasien.

Hukum di Indonesia dan Uni Eropa, WCAG 2.2 dan 3.0, menghitung kontras, pembaca layar, dan audit dalam Java dan Python.

Sebelum itu: jalankan pemeriksaan terhadap kedua peladen dan coba Soal Latihan 10.1 dan 10.2. Buku: Bab 11.

Kuliah 11 — Aksesibilitas dan Desain Inklusif (WCAG 2.2)

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 11

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita membuat kios antrean puskesmas dapat dipakai oleh setiap orang yang melewati pintunya. Kita memandang disabilitas sebagai ketidakcocokan antara orang dan desain, mempelajari apa yang diwajibkan hukum dan WCAG 2.2, menghitung kontras dengan tangan, melihat cara pembaca layar membaca halaman, dan membangun pemeriksa kontras serta audit dalam Java dan Python. Buku: Bab 11, Subbab 11.1–11.14.

Yang Dibahas dalam Pertemuan Ini

1. **Empat pasien** — dan halaman yang menyingkirkan mereka.
2. **Gagasan** — model sosial, desain inklusif, spektrum persona.
3. **Hukum dan standar** — UU 8/2016, PP 42/2020, European Accessibility Act, WCAG 2.2 = ISO/IEC 40500:2025, kedudukan WCAG 3.0.
4. **Kontras dan teknologi bantu** — tiga persamaan; peran, nama, dan keadaan; aturan pertama ARIA.
5. **Pengujian** — WebAIM Million 2026, empat lapis pengujian, dan program kita sendiri.

Buku: Bab 11.

Suatu Pagi di Puskesmas Sukamaju

Pak Harto, 71 tahun, penglihatannya melemah karena usia: tidak melihat petunjuk kelabu pucat; tidak tahu kotak mana yang diisi.

Mbak Laras, tunanetra, memakai TalkBack: mendengar “gambar, gambar, tombol, tautan” — tidak lebih.

Mas Dimas, lengan kanan patah, memakai papan ketik: Tab melompati ketiga kotak layanan.

Bu Wulan, menggendong bayi: ibu jarinya meleset dari ikon bantuan yang mungil.

Tak satu pun luar biasa; kegagalannya adalah yang paling umum di web, dan masing-masing mudah diperbaiki. Buku: Subbab 11.1.

Disabilitas adalah Ketidakcocokan

Model sosial: orang berpenglihatan rendah dibuat tak berdaya *oleh* teks pucat; orang yang sama tidak terhalang teks besar.

Aksesibilitas: penyandang disabilitas dapat memakai desain secara efektif.

Desain inklusif: merancang untuk seluruh keragaman manusia sejak awal.

Spektrum persona: permanen (satu lengan), sementara (lengan patah), situasional (menggendong bayi).

Desain dapat diubah — itulah sebabnya model sosial memberdayakan perancang. Buku: Subbab 11.2.

Siapa yang Terdampak

WHO: sekitar **1,3 miliar orang — 16%, satu dari enam** — mengalami disabilitas yang berarti, dan jumlahnya bertambah seiring menuanya penduduk.

Tambahkan keterbatasan sementara dan situasional, maka hampir semua orang terlayani buruk pada suatu saat dalam harinya.

Usia mengubah penglihatan, pendengaran, kendali motorik halus, dan kecepatan mempelajari antarmuka baru.

Karena itulah pasien lansia yang pertama menyadari kios yang tidak aksesibel. Buku: Subbab 11.3.1.

Apa yang Diwajibkan Hukum

UU 8/2016: penyandang disabilitas berhak atas aksesibilitas serta informasi dan komunikasi, termasuk dalam pelayanan publik.

PP 42/2020: aksesibilitas terhadap permukiman, **pelayanan publik**, dan pelindungan dari bencana; penyelenggara wajib menyediakan layanan yang aksesibel.

European Accessibility Act: berlaku sejak **28 Juni 2025** untuk perdagangan elektronik, perbankan, buku elektronik, komunikasi, dan lainnya; standar EN 301 549.

WCAG 2.2 (W3C Recommendation, diperbarui 12 Desember 2024) = **ISO/IEC 40500:2025** (Oktober 2025). **WCAG 3.0**: *Working Draft* 10 September 2026, masih bertahun-tahun lagi.

Pada September 2026, bangun dan uji terhadap WCAG 2.2 tingkat AA. Buku: Subbab 11.3.2.

WCAG 2.2 dalam Satu Halaman

POUR: Perceivable, Operable, Understandable, Robust.

Tingkat **A** (minimum), **AA** (sasaran hukum yang lazim), **AAA** (tertinggi).

86 kriteria keberhasilan: sembilan ditambahkan pada 2.1, dan 4.1.1 Parsing dihapus.

Di balik kegagalan umum: 1.1.1, 1.3.1, 1.4.3, 1.4.11, 2.1.1, 2.4.4, 2.4.7, 3.1.1, 4.1.2.

Bab 9 dan 10 sudah memakai 2.5.8, 2.5.1, 2.5.7, 1.4.10, dan 1.3.5. Buku: Subbab 11.3.3.

Sembilan Kriteria Baru

Tabel 11.1 Sembilan kriteria keberhasilan yang ditambahkan pada WCAG 2.2.

Kriteria	Tingkat
2.4.11 Focus Not Obscured (Minimum)	AA
2.4.12 Focus Not Obscured (Enhanced)	AAA
2.4.13 Focus Appearance	AAA
2.5.7 Dragging Movements	AA
2.5.8 Target Size (Minimum)	AA
3.2.6 Consistent Help	A
3.3.7 Redundant Entry	A
3.3.8 Accessible Authentication (Minimum)	AA
3.3.9 Accessible Authentication (Enhanced)	AAA

Tabel 11.1: sebagian besar menolong orang dengan disabilitas kognitif, motorik, atau penglihatan rendah. Buku: Subbab 11.3.3 dan Tabel 11.1.

Kontras, Langkah 1: Pelinearan

$$c_{lin} = \begin{cases} c/12.92 & \text{jika } c \leq 0.04045 \\ \left(\frac{c+0.055}{1.055}\right)^{2.4} & \text{jika tidak} \end{cases} \tag{11.1}$$

Setiap kanal 0–255 dibagi 255 menjadi c.

Persamaan (11.1) membatalkan kurva gamma yang dipakai layar untuk menyandikan kecerahan.

WCAG 2.2 memakai **0,04045**; tutorial lama menulis 0,03928 (tanpa beda praktis).

Buku: Subbab 11.3.4.

Kontras, Langkah 2 dan 3

$$L = 0.2126\, R_{lin} + 0.7152\, G_{lin} + 0.0722\, B_{lin} \tag{11.2}$$

$$CR = \frac{L_1 + 0.05}{L_2 + 0.05} \tag{11.3}$$

Persamaan (11.2): luminans relatif — hijau paling berbobot, biru paling ringan.

Persamaan (11.3): yang lebih terang di atas yang lebih gelap, masing-masing ditambah 0,05 untuk silau layar; dari 1 sampai 21.

AA: **4,5:1** teks biasa, **3:1** teks besar (18 pt atau 14 pt tebal) dan bagian kontrol (1.4.11); AAA: 7:1.

Buku: Subbab 11.3.4.

Contoh: Petunjuk yang Pucat

#9CA3AF di atas putih. Merah 156: $c = 0.6118$, $R_{lin} = 0.3325$; $G_{lin} = 0.3663$; $B_{lin} = 0.4287$.

$L = 0.2126 \times 0.3325 + 0.7152 \times 0.3663 + 0.0722 \times 0.4287 = 0.3636$.

$CR = 1.05 / 0.4136 = 2.54$ — nyaris hanya separuh dari 4,5.

Itulah sebabnya Pak Harto tidak dapat melihatnya. Buku: Subbab 11.3.4, Contoh.

Cara Pembaca Layar Membaca Halaman

TalkBack, VoiceOver, NVDA, JAWS membaca **pohon aksesibilitas**: peran, nama, keadaan.

`<button>`: peran, Tab, Enter/Spasi. `<div onclick>`: tanpa peran, dilewati Tab (2.1.1).

`<img>` memerlukan `alt` (atau `alt=""` bila hiasan); tautan dan tombol yang hanya gambar tak bernama (2.4.4, 4.1.2).

`<label>` menamai ruas (1.3.1); judul membentuk kerangka; `lang` memilih lafal (3.1.1).

Aturan pertama ARIA: *bila elemen bawaan sudah cukup, pakailah. Buku: Subbab 11.3.5.*

Wajah Web pada 2026

WebAIM Million (Februari 2026): kegagalan pada **95,9%** halaman beranda; rata-rata **56,1** galat per halaman.

Enam kegagalan = **96%** galat: kontras rendah 83,9%, tanpa alt 53,1%, tanpa label 51,0%, tautan kosong 46,3%, tombol kosong 30,6%, tanpa bahasa 13,5%.

Halaman dengan ARIA rata-rata memiliki **lebih banyak** galat (59,1 berbanding 42).

Kegagalan umum itu sederhana; pengujian otomatis hanya melihat sebagian masalah. Buku: Subbab 11.3.6.

Empat Lapis Pengujian

1. Pemeriksaan otomatis — program kita, DevTools, axe, WAVE.

2. Penelusuran papan ketik — Tab, Shift+Tab, Enter, Spasi, panah; fokus terlihat dan tak pernah tertutup.

3. Pembaca layar dan perbesaran — TalkBack/VoiceOver/NVDA; 200% dan 400%.

4. Penyandang disabilitas — tugas nyata, metode Bab 7.

Lapis 1 dan 2 pada setiap perubahan; keempatnya sebelum layanan publik ditayangkan. Buku: Subbab 11.4.

Kegagalan, Kriteria, dan Pemeriksaan Kita

Tabel 11.2 Kegagalan paling umum, kriteria di baliknya, dan pemeriksaan bab ini.

Kegagalan (WebAIM 2026)	WCAG 2.2	Pemeriksaan kita
Teks berkontras rendah	1.4.3, 1.4.11	`contrast`
Gambar tanpa alt	1.1.1	`a11y_audit`
Masukan tanpa label	1.3.1, 4.1.2	`a11y_audit`
Tautan / tombol kosong	2.4.4, 4.1.2	`a11y_audit`
Bahasa halaman tidak dinyatakan	3.1.1	`a11y_audit`
Non-kontrol yang dapat diklik, lompatan judul, id ganda	2.1.1, 1.3.1, 4.1.2	`a11y_audit`

Tabel 11.2. Buku: Subbab 11.4 dan Tabel 11.2.

Pemeriksa Kontras

Kode 11.1 Dari `contrast.py`: Persamaan (11.1), Persamaan (11.2), dan Persamaan (11.3) dalam tiga fungsi kecil.

```
def channel(c):
    c = c / 255
    return c / 12.92 if c <= 0.04045 else ((c + 0.055) / 1.055) ** 2.4


def luminance(hexcolor):
    r, g, b = (int(hexcolor[i:i + 2], 16) for i in (1, 3, 5))
    return 0.2126 * channel(r) + 0.7152 * channel(g) + 0.0722 * channel(b)


def ratio(fg, bg):
    hi, lo = sorted((luminance(fg), luminance(bg)), reverse=True)
    return (hi + 0.05) / (lo + 0.05)
```

*Kode 11.1. Untuk pasangan yang gagal, alat ini juga **menyarankan** warna lebih gelap yang lulus, dengan rona yang tetap. `Contrast.java` sama. Buku: Subbab 11.5.2.*

Warna-Warna Kios

use	fg	bg	ratio	need	result
body text	#1F2937	#FFFFFF	14.68	4.5	pass
hint text (first version)	#9CA3AF	#FFFFFF	2.54	4.5	FAIL -> try #6D727A
button label	#FFFFFF	#0F7C7C	5.01	4.5	pass
amber number on white	#E0A100	#FFFFFF	2.27	3.0	FAIL -> try #BE8900
field border	#D1D5DB	#FFFFFF	1.47	3.0	FAIL -> try #929599
field border (second version)	#6B7280	#FFFFFF	4.83	3.0	pass
now serving on navy	#FFFFFF	#12355B	12.46	3.0	pass



Gambar 11.1 Tujuh pasangan warna kios beserta rasionya; tiga gagal AA.

Gambar 11.1: petunjuk pucat 2,54, nomor amber 2,27, bingkai terang 1,47. Halaman yang diperbaiki memakai #4B5563, teks gelap untuk nomor, dan bingkai #6B7280. Buku: Subbab 11.6.

Audit

Dua ekspresi reguler — `TAG` dan `PAIR` — agar Java (JDK tanpa pengurai HTML) dapat identik.

Berlabel = `<label for>`, dibungkus `<label>`, `aria-label`, atau `aria-labelledby`.

Bernama = teks, gambar dengan `alt` tidak kosong, atau `aria-label`.

`onclick` pada apa pun selain tautan atau tombol = tak terjangkau papan ketik.

```
../data/queue_v1.html: 12 problem(s)
  line  2  SC 3.1.1  page has no lang attribute
  line 13  SC 1.1.1  img logo.png has no alt
  line 15  SC 1.3.1  heading jumps from h1 to h4
  line 16  SC 2.1.1  clickable <div> is not reachable by keyboard
  line 17  SC 2.1.1  clickable <div> is not reachable by keyboard
  line 18  SC 2.1.1  clickable <div> is not reachable by keyboard
  line 20  SC 1.3.1  input card has no label
  line 21  SC 1.1.1  img print.png has no alt
  line 21  SC 4.1.2  button has no text a screen reader can read
  line 22  SC 1.1.1  img help.png has no alt
  line 22  SC 2.4.4  a has no text a screen reader can read
  line 23  SC 4.1.2  id "card" is used twice
../data/queue_v2.html: 0 problem(s)
```

Dua belas masalah pada halaman pertama, nol pada yang diperbaiki; setiap baris berpadanan dengan salah satu dari empat pasien. Buku: Subbab 11.5.3 dan 11.6.

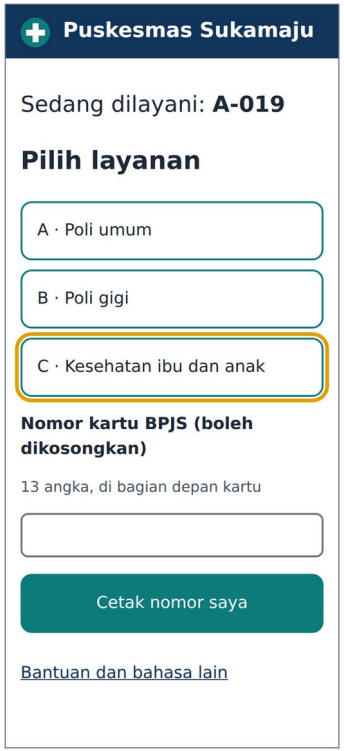
Halaman yang Diperbaiki

`lang`, logo hiasan dengan `alt=""`, judul h1 → h2.

`<button>` sungguhan setinggi 64 px; ruas kartu berlabel dengan petunjuk lewat `aria-describedby`, `inputmode="numeric"`, bingkai 3:1.

`aria-live="polite"` mengumumkan “Sedang dilayani: A-020”.

Cincin amber `:focus-visible` 4 px memperlihatkan posisi papan ketik.



Gambar 11.2 Kios yang diperbaiki setelah tiga kali Tab, dengan cincin fokus pada layanan ketiga.

Gambar 11.2. Buku: Subbab 11.5.4 dan 11.6.

Menulis Teks Alternatif yang Baik

Uraikan **tujuannya**, bukan pikselnya: ikon cetak di tombol adalah “Cetak nomor saya”, bukan “ikon printer”.

Gambar hiasan diberi `alt=""` agar dilewati pembaca layar; logo di samping nama yang terlihat adalah hiasan.

Jangan nama berkas (“print.png”) atau “gambar dari…”; pembaca layar sudah mengucapkan “gambar”.

Bagan memerlukan pesannya dalam teks, atau tabel: “Waktu tunggu antrean turun dari 40 ke 25 menit pada Agustus”.

Soal Latihan 11.4 mengubah aturan nama berkas menjadi kode. Buku: Subbab 11.3.5 dan 11.11.

Uji Papan Ketik Sepuluh Menit

Singkirkan tetikus. Tekan Tab dari atas: apakah setiap kontrol mendapat fokus, dalam urutan baca?

Apakah fokus selalu terlihat, dan tak pernah tertutup *header* lengket atau jendela sembul (2.4.7, 2.4.11)?

Dapatkah setiap tindakan dilakukan dengan Enter, Spasi, dan panah — tanpa jebakan yang tak dapat ditinggalkan dengan Tab?

Perbesar ke 200% dan 400%: apakah halaman tertata ulang tanpa gulir ke samping (1.4.4, 1.4.10)?

Sepuluh menit seperti ini menemukan masalah yang tidak dilaporkan alat otomatis mana pun. Buku: Subbab 11.4 dan 11.8.

Pengujian

`pytest` (4 pengujian): hitam di atas putih 21:1, warna di atas dirinya 1:1, #767676 di atas putih **4,54**; saran perbaikan lulus; 12 masalah lawan 0; label pembungkus dihitung.

`Ch11Test.java`: enam pemeriksaan yang sama, “all 6 checks passed”.

Buku: Subbab 11.7.

Melampaui Lapis Otomatis

Papan ketik: enam perhentian — tiga layanan, ruas kartu, Cetak, Bantuan — sesuai urutan baca; cincin selalu terlihat; tak ada yang tertutup (2.4.11).

Perbaikan: setelah memilih layanan, umumkan “Nomor Anda B-024”.

Pembaca layar: “Pilih layanan, judul tingkat 2; A, Poli umum, tombol; …”.

Manusia: apakah pasien menyebut “Kesehatan ibu dan anak” atau “KIA”? Pertahankan petugas yang mengambilkan nomor bagi yang meminta.

Buku: Subbab 11.8.

Memakainya Sungguhan

Tuliskan “WCAG 2.2 tingkat AA” dalam setiap pesanan layanan publik (UU 8/2016, PP 42/2020).

Jalankan audit dan pemeriksaan kontras pada setiap rilis, seperti pengujian unit.

Perbaiki dengan HTML bawaan dahulu; uji palet *design system* sekali, sejak awal.

Terbitkan pernyataan aksesibilitas beserta cara lain untuk memperoleh bantuan.

Buku: Subbab 11.9.

IMK dalam Praktik: Profitina

Skor kesiapan tidak boleh hanya warna (1.4.1): angka, kata tingkat, dan penjelasan sebagai teks.

Bagan tren mingguan memerlukan alternatif teks atau tabel dan ekspor CSV.

Swalayan UU PDP (persetujuan, ekspor, penghapusan, halaman hukum HTML) juga persoalan aksesibilitas: persetujuan yang tak dapat ditangkap pengguna pembaca layar tidaklah bermakna.

Peringatan **Link Safety** harus diumumkan sebagai teks, bukan hanya ikon berwarna.

Buku: Subbab 11.10.

Kesalahan yang Sering Terjadi

`div` yang dapat diklik; gambar tanpa (atau dengan buruknya) `alt`; *placeholder* sebagai label.

Teks dan bingkai pucat; `outline: none`; warna sebagai satu-satunya sinyal.

ARIA sebagai hiasan; mempercayai alat begitu saja; aksesibilitas sebagai langkah terakhir.

Buku: Subbab 11.11.

Ringkasan

Disabilitas adalah ketidakcocokan; spektrum persona memperluas siapa yang diuntungkan. Satu dari enam orang (WHO).

UU 8/2016, PP 42/2020, EAA (28 Juni 2025), ISO/IEC 40500:2025; WCAG 3.0 masih draf.

POUR, A/AA/AAA, 86 kriteria, sembilan baru (Tabel 11.1); kontras dengan Persamaan (11.1), Persamaan (11.2), Persamaan (11.3).

Empat lapis pengujian (Tabel 11.2); 12 masalah ditemukan dan diperbaiki (Gambar 11.2).

Buku: Subbab 11.12.

Soal untuk Dicoba

11.1 Kontras #767676 di atas putih dengan tangan.

11.2 Amber di atas navy, putih di atas amber — mana untuk “sedang dilayani”?

11.4 Laporkan `alt` yang hanya berupa nama berkas.

11.6 Masuk log dengan CAPTCHA dan WCAG 3.3.8 — mengapa *passkey* menolong (90% mengetahui, 75% mengaktifkan pada sekurang-kurangnya satu akun, FIDO 2026).

Penyelesaian lengkapnya ada di buku. Buku: Subbab 11.13 dan 11.14.

Pertemuan Berikutnya

Bab 12 — Desain Emosional, Kepercayaan, dan UX yang Etis. Keanggotaan sebuah laundry yang meniru trik aplikasi besar.

Tiga tingkat Norman, kepercayaan, *dark pattern*, DSA, FTC, UU PDP dan PP 33/2026, serta *linter* dan buku besar persetujuan.

Sebelum itu: telusuri `queue_v2.html` *dengan Tab dan pembaca layar, dan coba Soal Latihan 11.1 dan 11.3. Buku: Bab 12.*

Kuliah 12 — Desain Emosional, Kepercayaan, dan UX yang Etis

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 12

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita merancang untuk perasaan secara jujur. Sebuah laundry meniru trik persuasi aplikasi besar dan menghabiskan kepercayaan bertahun-tahun dalam beberapa minggu. Kita menamai trik-trik itu, melihat apa kata Uni Eropa, FTC, serta UU PDP dan PP 33/2026 tentangnya, dan membangun linter persetujuan serta buku besar persetujuan dalam Java dan Python. Buku: Bab 12, Subbab 12.1–12.14.

Yang Dibahas dalam Pertemuan Ini

1. **Keanggotaan Bu Sri** — bagaimana trik “pertumbuhan” mengubah laundry tepercaya menjadi “aplikasi pinjol”.
2. **Desain emosional** — tiga tingkat Norman dan efek estetika–kegunaan.
3. **Kepercayaan** — kecakapan, itikad baik, kejujuran; *passkey*.
4. ***Dark pattern* dan hukum** — katalog, dua ukuran, DSA, FTC, UU PDP, dan PP 33/2026.
5. **Alat** — tinjauan alur yang adil, *linter* untuk alur yang diuraikan, dan buku besar peristiwa persetujuan.

Buku: Bab 12.

Keanggotaan Bu Sri

Laundry Wangi, laundry kiloan; bertahun-tahun pesanan lewat WhatsApp, balasan jujur, dompet dikembalikan lengkap dengan uang yang dihitung.

Member Hemat di aplikasi baru: Rp25.000 sebulan — dengan spanduk merah “Diskon 50%!”, hitung mundur yang **mulai lagi dari 09:59**, “Tinggal 3 kuota hari ini!” setiap hari.

Satu kotak **sudah tercentang** untuk syarat, privasi, promosi, dan pembagian ke toko mitra; tombol besar “Saya setuju semuanya”; tautan mungil “Tidak, saya suka bayar mahal”.

Bergabung 2 ketukan; keluar 6 langkah lewat WhatsApp.

Sebulan pendaftaran naik, lalu nomor muncul di siaran bengkel, keluhan di grup obrolan, dan “Kok sekarang Ibu kayak aplikasi pinjol?” Buku: Subbab 12.1.

Gagasannya dalam Bahasa Sehari-hari

Desain emosional: merancang dengan sengaja bagaimana antarmuka membuat orang merasa — terbaiknya, *delight* ("sedang disetrika — siap pukul 16.00").

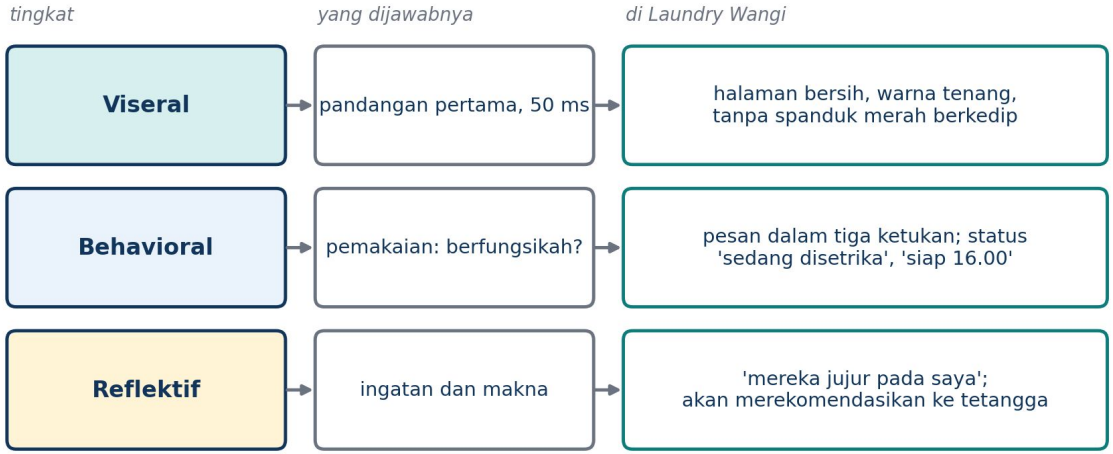
Kepercayaan: kesediaan mengandalkan sistem yang tak dapat Anda periksa sepenuhnya; lambat dibangun, cepat hilang.

Dark pattern (pola menipu, Brignull 2010): desain yang membuat pengguna melakukan apa yang tidak dimaksudkannya, demi keuntungan bisnis.

Persetujuan: diberikan bebas, spesifik, berdasarkan informasi, tidak ambigu — dan dapat ditarik.

Buku: Subbab 12.2.

Tiga Tingkat Norman



Gambar 12.1 Tiga tingkat desain emosional Norman yang diterapkan pada Laundry Wangi.

*Gambar 12.1: **viseral** (kesan pertama dalam sekitar 50 ms, Lindgaard dkk.), **behavioral** (berfungsikah?), **reflektif** (percayakah saya, akankah saya merekomendasikan?). Buku: Subbab 12.3.1.*

Efek Estetika–Kegunaan

Kurosu dan Kashimura (1995): antarmuka yang menarik **dinilai** lebih mudah dipakai, meskipun sebenarnya tidak.

Tingkat viseral mewarnai penilaian behavioral.

Alasan untuk memedulikan rupa — sekaligus peringatan: keindahan dapat menyembunyikan masalah kegunaan dari uji yang hanya menanyakan pendapat.

Dark pattern bekerja pada tingkat viseral dan behavioral dan **merusak tingkat reflektif**, tempat kepercayaan tinggal.

Pelanggan Bu Sri berbicara dari tingkat reflektif. Buku: Subbab 12.3.1.

Bagaimana Kepercayaan Dibangun dan Hilang

Kecakapan (berfungsi), **itikad baik** (bertindak untuk saya), **kejujuran** (berkata benar, menepati janji).

Dibangun oleh: status akurat, harga jelas, jalan batal dan keluar yang terlihat, kesalahan diakui, hanya meminta data yang perlu.

Hilang oleh: kebohongan yang ketahuan, biaya tersembunyi, data dipakai tak terduga, jalan keluar jauh lebih sulit daripada jalan masuk.

FIDO 2026: 47% cenderung membatalkan pembelian atau masuk log karena lupa kata sandi; 90% mengetahui ***passkey***, 75% telah mengaktifkannya.

Passkey memperbaiki tingkat behavioral dan reflektif sekaligus. Buku: Subbab 12.3.2.

Temuan Riset

Mathur dkk. (2019): menjelajahi sekitar 11.000 situs belanja; **1.818** kemunculan *dark pattern* pada **1.254** situs (sekitar 11%) — penghitung waktu palsu, stok menipis palsu, *confirmshaming*.

Gray dkk. (CHI 2024): ontologi yang menyatukan taksonomi — *sneaking, obstruction, interface interference, forced action, social engineering*.

Ahuja, Gunawan, Bielova, dan Santos (CHI 2026): 59 pola dipetakan ke pelanggaran otonomi menurut DSA; delapan faktor desain.

Riset IMK dan regulasi kini menggarap pertanyaan yang sama. Buku: Subbab 12.3.3 dan 12.3.4.

Katalog di Laundry Wangi

Tabel 12.1 *Dark pattern* alur pertama Laundry Wangi dan alternatifnya yang adil.

Pola	Yang dilakukan	Alternatif yang adil
Desakan waktu palsu	Hitung mundur mulai lagi dari 09:59	Tenggat nyata, atau tidak ada
Kelangkaan palsu	“Tinggal 3 kuota” setiap hari	Hanya batas yang nyata
Kotak sudah tercentang	Tujuan opsional menyala	Tujuan opsional mati
Persetujuan dibundel	Satu centang untuk empat hal	Satu pilihan per tujuan
Campur tangan antarmuka	“Setuju” besar, “tidak” kelabu mungil	Tombol setara
Confirmshaming	“Saya suka bayar mahal”	Netral: “Nanti saja”
Sulit berhenti	Gabung 2 langkah, keluar 6	Keluar di tempat bergabung

Tabel 12.1. Buku: Subbab 12.3.3 dan Tabel 12.1.

Dua Ukuran

$$A = \frac{s_{out}}{s_{in}} \tag{12.1}$$

$$P = \frac{w_a h_a}{w_r h_r} \tag{12.2}$$

Persamaan (12.1) **asimetri hambatan:** langkah keluar dibagi langkah bergabung; adil berarti $A \leq 1$.

Persamaan (12.2) **rasio penonjolan:** luas tombol setuju dibagi luas tombol tolak; patokan buku ini $P \leq 2$ dan gaya yang sama (bukan ambang hukum).

Laundry Wangi: $A = 6/2 = 3$; $P = (320 \times 56)/(90 \times 14) = 17920/1260 = 14.2$.

Buku: Subbab 12.3.3.

Hukum: Uni Eropa dan Amerika Serikat

DSA Pasal 25 (berlaku bagi semua platform daring sejak 17 Februari 2024): dilarang antarmuka yang menipu, memanipulasi, atau secara material mendistorsi keputusan yang bebas dan berdasarkan informasi.

Digital Fairness Act: telah diumumkan; diperkirakan diajukan pada kuartal keempat 2026 — *dark pattern*, desain adiktif, personalisasi yang tidak adil.

FTC: *Bringing Dark Patterns to Light* (2022); aturan *click-to-cancel* **dibatalkan** Sirkuit Kedelapan pada Juli 2025; pemberitahuan awal penyusunan aturan baru dikirim ke OMB pada 30 Januari 2026.

Prinsip Persamaan (12.1) tetap menjadi arah kebijakan di Amerika Serikat. Buku: Subbab 12.3.4.

Hukum: Indonesia

UU PDP 27/2022: persetujuan yang sah secara eksplisit sebagai dasar pemrosesan (Pasal 20); tertulis atau terekam, dapat dibedakan, bahasa yang jelas (Pasal 22); hak menarik kembali (Pasal 9); bukti persetujuan (Pasal 24); persetujuan orang tua untuk anak (Pasal 25).

PP 33/2026: peraturan pelaksanaan, ditetapkan **16 Juli 2026**, berlaku **16 Januari 2027**; persetujuan eksplisit dan sah, untuk tujuan tertentu yang disampaikan.

Kotak yang sudah tercentang dan dibundel serta penarikan yang hanya lewat WhatsApp tidak sejalan dengan semua itu.

Merancang layar persetujuan adalah cara persyaratan hukum sampai kepada pengguna. Buku: Subbab 12.3.4.

Mengukur Perasaan

Kuesioner: **UEQ** (daya tarik, kejelasan, efisiensi, keterandalan, stimulasi, kebaruan) dan **UEQ-S** 8 butir; SUS untuk kegunaan yang dirasakan (Bab 7).

Untuk kepercayaan, perilaku berbicara lebih keras: pembatalan, penarikan persetujuan, keluhan, rekomendasi.

Desain yang menaikkan pendaftaran **sekaligus** penarikan dan keluhan bukanlah keberhasilan.

Buku: Subbab 12.3.5.

Persuasi yang Menghormati Pengguna

Keterbukaan: tetap berhasilkah bila pengguna memahaminya sepenuhnya? Penghematan yang benar ya; hitung mundur palsu tidak.

Keterbalikan: dapatkah pengguna dengan mudah membatalkan apa yang didorong desain?

Keselarasan: apakah ia melayani tujuan pengguna sebaik tujuan bisnis?

Lulus ketiganya, maka boleh sepersuasif yang Anda mau.

Buku: Subbab 12.3.6.

Tinjauan Alur yang Adil

Semua klaim benar? Tujuan opsional mati? Satu pilihan per tujuan, kata-kata lugas?
Pilihan setara tampak setara ($P \leq 2$)? Penolakan netral? Keluar semudah bergabung ($A \leq 1$)?
Penarikan di tempat persetujuan diberikan? Persetujuan dicatat sebagai bukti (siapa, tujuan, kapan, versi teks)?
Total biaya jelas sebelum membayar — tanggal perpanjangan, harga, cara berhenti, pengingat?

Tabel 12.2 Tinjauan alur yang adil, versi ringkas.

Pertanyaan	Periksa
Klaim benar?	Hitung mundur, stok, harga
Opsional mati?	Promosi, pembagian, pemrofilan
Pilihan setara setara?	$P \leq 2$ menurut Persamaan (12.2)
Keluar semudah masuk?	$A \leq 1$ menurut Persamaan (12.1)
Persetujuan sebagai bukti?	Peristiwa dengan versi teks

Tabel 12.2: setiap “tidak” adalah cacat yang diperbaiki sebelum rilis; tabel di buku berisi sembilan baris. Buku: Subbab 12.4 dan Tabel 12.2.

Menguraikan Sebuah Alur

Kode 12.1 `flow_v1.txt`: alur pertama, satu baris per tujuan, tombol, hitung mundur, langkah, dan penarikan.

```
# Laundry Wangi sign-up and membership, first version
purpose|orders|Process my laundry orders|required|on
purpose|promo|Send me promotions by WhatsApp|optional|on
purpose|partners|Share my contact with partner shops|optional|on
bundled|yes
button|accept|I agree to everything|320|56|filled
button|reject|No thanks, I like paying more|90|14|text
countdown|resets
steps|subscribe|2
steps|cancel|6
withdraw|WhatsApp to customer service only
```

Kode 12.1: program hanya dapat memeriksa apa yang diuraikan. Buku: Subbab 12.5.1.

Dua Ukuran dalam *Linter*

Kode 12.2 Dari `consent_lint.py`: Persamaan (12.2) dan Persamaan (12.1) sebagai aturan.

```
    prominence = (aw * ah) / (rw * rh)
    if prominence > MAX_PROMINENCE or rstyle != "filled":
        found.append(("interface interference",
                      f"accept has {prominence:.1f}x the area of reject, "
                      f"reject is '{rstyle}'"))
    if any(p in rlabel.lower() for p in SHAMING):
        found.append(("confirmshaming", f"reject label: '{rlabel}'"))
    if flow.get("countdown") == "resets":
        found.append(("false urgency", "countdown restarts on every visit"))
    asym = flow["steps"]["cancel"] / flow["steps"]["subscribe"]
    if asym > 1:
        found.append(("hard to cancel",
                      f"cancel takes {asym:.1f}x the steps of subscribing"))
```

Kode 12.2. Aturan lainnya: opsional-dan-menyala, `bundled|yes`, frasa mempermalukan (Inggris dan Indonesia), hitung mundur yang mulai lagi, penarikan yang "hanya" di tempat lain. `ConsentLint.java` sama. Buku: Subbab 12.5.2.

Menjalankan *Linter*

```
../data/flow_v1.txt: prominence 14.2, cancel/subscribe 3.0, 8 finding(s)
  - pre-ticked box         optional purpose 'promo' is on
  - pre-ticked box         optional purpose 'partners' is on
  - bundled consent        one tick covers all purposes
  - interface interference accept has 14.2x the area of reject, reject is 'text'
  - confirmshaming         reject label: 'No thanks, I like paying more'
  - false urgency          countdown restarts on every visit
  - hard to cancel         cancel takes 3.0x the steps of subscribing
  - hard to withdraw       withdraw: WhatsApp to customer service only
../data/flow_v2.txt: prominence 1.0, cancel/subscribe 1.0, 0 finding(s)
```

Delapan temuan pada alur pertama — satu untuk setiap hal yang dialami pelanggan Bu Sri — dan nol pada alur kedua, yang tujuannya mati, tombolnya setara, hitung mundurnya hilang, dan berhentinya dua langkah. Buku: Subbab 12.5.3 dan 12.6.

Kedua Layar Persetujuan



Gambar 12.2 Layar persetujuan pertama dan kedua di ponsel.

Gambar 12.2: (a) spanduk merah, hitung mundur yang mulai lagi, satu kotak tercentang, tombol setuju besar, dan tautan yang mempermalukan; (b) satu pilihan per tujuan, penjelasan lugas, dua tombol setara. Buku: Subbab 12.6.

Buku Besar, Bukan Penanda

Persetujuan adalah **riwayat**: memberi, menarik, memberi lagi untuk teks yang lebih baru.

Simpan setiap peristiwa: waktu, pengguna, tujuan, tindakan, versi teks; turunkan keadaan kini.

Satu pertanyaan bagi bagian lain sistem: *bolehkah kita memakai data ini untuk tujuan ini sekarang?*

```
U17 orders   give     2026-09-01T08:02  text v2  -> may use
U17 promo    withdraw 2026-09-03T19:40  text v2  -> must NOT use
U23 orders   give     2026-09-04T10:15  text v2  -> may use
U23 partners give     2026-09-04T10:15  text v2  -> may use
U23 promo    give     2026-09-20T07:30  text v3  -> may use
U17 promo allowed? False
U23 promo allowed? True
```

Penarikan U17 yang lebih akhir menang; persetujuan U23 untuk teks v3 — v4 memerlukan pertanyaan baru (Soal Latihan 12.5). Inilah bukti yang diminta Pasal 24 UU PDP. Buku: Subbab 12.5.4 dan 12.6.

Mengapa Usaha Kecil Lebih Berisiko

Pelanggan laundry kiloan tinggal di lingkungan yang sama dan berbagi satu grup obrolan.

Satu pengalaman buruk sampai ke puluhan orang yang belum pernah memakai aplikasinya — dalam hitungan menit.

Satu pengalaman baik beredar dengan cara yang sama: kemeja tertukar yang dikembalikan dengan permintaan maaf; "hujan akan menunda pengeringan" yang jujur.

Bagi Bu Sri, desain yang etis bukan kemewahan moral, melainkan strategi usaha yang paling masuk akal.

Buku: Subbab 12.2 dan 12.3.2.

Kata-Kata yang Menenangkan

Bandingkan “Pembayaran gagal!” berhuruf merah dengan “Pembayaran belum masuk. Saldo Anda tidak terpotong; silakan coba lagi atau bayar di kasir.”

Kalimat kedua menjawab tiga pertanyaan sekaligus: **apa yang terjadi, amankah uang saya, apa yang bisa saya lakukan sekarang?**

Aturan yang sama untuk keterlambatan, penolakan, dan kesalahan: katakan yang benar, katakan artinya bagi pengguna, tawarkan langkah berikutnya.

Bagi pelanggan yang pernah menjadi korban penipuan daring, kalimat semacam itu membedakan layanan tepercaya dari yang patut dicurigai.

Buku: Subbab 12.8.

Pengujian

`pytest` (3 pengujian): delapan temuan dengan 14.2 dan 3.0 yang diperiksa dengan tangan; alur kedua bersih; penarikan yang lebih akhir menang dan tujuan yang tak pernah ditanyakan tidak diizinkan.

`Ch12Test.java`: enam pemeriksaan yang sama, “all 6 checks passed”.

Buku: Subbab 12.7.

Membuatnya Lebih Baik

Rasa senang yang benar: status per pesanan (“dikeringkan — hujan bisa menambah setengah hari”), dan “Anda hemat Rp18.000 pada September”.

Pengingat sebelum menagih: tanggal perpanjangan di halaman Akun, pengingat sehari sebelumnya, berhenti dalam satu langkah.

***Microcopy* yang menenangkan:** “Pembayaran belum masuk. Saldo Anda tidak terpotong” menjawab apa yang terjadi, apakah uang aman, dan apa langkah berikutnya.

Melampaui *linter*: tujuan yang kabur (“untuk meningkatkan layanan kami”) lolos pemeriksaan uraian; bacakan kepada orang luar.

Buku: Subbab 12.8.

Memakainya Sungguhan

Jalankan tinjauan alur yang adil pada setiap alur yang meminta uang, data, atau perhatian; periksa dengan *linter* pada setiap rilis.

Simpan buku besar persetujuan dengan versi teks; tampilkan di halaman Akun dengan kendali penarikan per tujuan.

Tulis tujuan dalam bahasa pengguna dengan akibat yang konkret.

Hitung ongkos trik di samping pendaftaran.

Bila Anda tak akan mengucapkannya di meja kasir, aplikasi Anda pun jangan. Buku: Subbab 12.9.

IMK dalam Praktik: Profitina

Swalayan UU PDP: persetujuan eksplisit, Argon2id, pembatasan laju masuk log, ekspor data, penggantian surel/kata sandi, penghapusan akun penuh, halaman hukum HTML — jalan keluar setersedia jalan masuk.

Langganan Pro (Rp50.000, 30 hari): perpanjangan otomatis dari saldo dengan pengingat dalam 24 jam sebelumnya.

Kanal: syarat saldo 50 PROFIT yang diperiksa, **tidak dipotong** — dan dinyatakan demikian; batasnya disebutkan.

Mesin konten: menyajikan draf AI hanya bila lebih baik, dan mengatakannya bila tak ada model. *Aplikasi menyarankan; pengguna memutuskan.*

Buku: Subbab 12.10.

Kesalahan yang Sering Terjadi

Meniru trik pertumbuhan; desakan waktu dan kelangkaan palsu.

Persetujuan yang sudah tercentang dan dibundel; tombol tak setara; kata-kata yang membuat merasa bersalah.

Jalan keluar lebih sulit daripada jalan masuk; satu penanda persetujuan yang ditimpa; tujuan yang kabur.

Mengira lonjakan pendaftaran sebagai keberhasilan.

Buku: Subbab 12.11.

Ringkasan

Viseral, behavioral, reflektif (Gambar 12.1); kepercayaan = kecakapan, itikad baik, kejujuran.

Dark pattern (Tabel 12.1) pada sekitar 11% dari 11.000 situs belanja; A (Persamaan (12.1)) dan P (Persamaan (12.2)) membuat dua di antaranya terukur.

DSA Pasal 25; DFA diperkirakan akhir 2026; aturan FTC dibatalkan, penyusunan dimulai lagi; UU PDP dan PP 33/2026 (mulai 16 Januari 2027).

Tinjauan (Tabel 12.2), *linter*, dan buku besar mengubah prinsip menjadi pemeriksaan (Gambar 12.2).

Buku: Subbab 12.12.

Soal untuk Dicoba

12.1 Hitung A untuk langganan satu ketukan dengan pembatalan tujuh langkah, lalu rancang $A \leq 1$.

12.2 P untuk "Terima semua" 300 × 48 lawan tautan "Pengaturan" 70 × 16.

12.5 Mengapa persetujuan v3 milik U23 tidak mencakup v4, dan `may_use` yang memeriksa versi.

12.6 Kapan pesan kelangkaan adalah informasi, dan kapan ia *dark pattern*?

Penyelesaian lengkapnya ada di buku. Buku: Subbab 12.13 dan 12.14.

Pertemuan Berikutnya

Bab 13 — Merancang Antarmuka AI: Chat, Kopilot, dan Agen.

Ketidakpastian, penjelasan, manusia dalam lingkaran keputusan, dan kepercayaan ketika sistemnya sendiri adalah sebuah model.

Sebelum itu: periksa alur dari aplikasi yang Anda pakai dengan linter, dan coba Soal Latihan 12.1 dan 12.5. Buku: Bab 13.

Kuliah 13 — Merancang Antarmuka AI: Chat, Kopilot, dan Agen

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 13

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita merancang percakapan antara manusia dan sistem yang menebak: apa yang harus ditampilkannya, kapan ia harus bertanya, apa yang boleh dikerjakannya sendiri, dan bagaimana kita mengukur apakah orang bersandar padanya dengan bijak. Buku: Bab 13, Subbab 13.1–13.12.

Yang Dibahas Pertemuan Ini

1. **Masalahnya** — pesanan WhatsApp tengah malam sebuah laundry, dan tiga pertanyaan pemiliknya.
2. **Chat, kopilot, agen** — tiga jenis antarmuka AI dan garis akibat di antara ketiganya.
3. **Teorinya** — pedoman interaksi manusia-AI, tingkat otomatisasi, kepercayaan, ketergantungan, dan kalibrasi, agen pada 2025–2026, aturan 2026.
4. **Metodenya** — kanvas interaksi AI dan daftar periksa rilis.
5. **Programnya** — asisten pesanan dan analisis ketergantungan dalam Java dan Python, dengan keluaran nyata.

Setiap program hari ini ada dalam Java dan Python, dan setiap keluaran yang ditampilkan adalah keluaran yang benar-benar dicetak program. Buku: Bab 13.

Pesan Tengah Malam Bu Wati

Profitina Laundry menerima dua puluh sampai tiga puluh pesanan WhatsApp antara pukul sepuluh malam dan dua dini hari.

Pesannya pendek dan tidak lengkap: “ada sekantong cucian, tolong jemput jam 16” tidak menyebut berat, layanan, maupun hari.

Ketika Bu Wati menjawab pagi harinya, sebagian pelanggan sudah pindah ke toko yang menjawab tengah malam.

Keponakannya mengusulkan asisten AI. Tiga pertanyaan Bu Wati: *Bagaimana pelanggan tahu itu mesin? Apa yang terjadi bila ia salah paham? Siapa yang bertanggung jawab bila ia memesan hal yang keliru?*

Ini pertanyaan tentang antarmuka di sekitar model, bukan tentang modelnya. Dua asisten di atas model yang sama bisa tepercaya atau berbahaya, bergantung pada keputusan desain ini. Buku: Subbab 13.1.

Chat, Kopilot, dan Agen

> **DEFINISI.** *Antarmuka chat* menjawab dengan kata-kata dan orang yang memutuskan. *Kopilot* mengusulkan perubahan di dalam alat lain yang diterima atau ditolak orang. *Agen* mengambil tindakan di dunia atas nama orang: memesan, membayar, mengirim, menghapus.

Balasan chat yang salah membuang satu menit; saran salah yang diterima menjadi kesalahan orang itu; tindakan agen yang salah *sudah telanjur terjadi*.

Makin banyak yang dapat dilakukan AI, makin besar kewajiban antarmukanya menunjukkan apa yang akan dilakukan dan membiarkan orang menghentikannya.

Pada Juli 2025 ChatGPT agent mulai menjelajah, mengisi formulir, dan membeli, serta dilatih untuk meminta izin sebelum tindakan yang berakibat nyata. Asisten Bu Wati adalah agen kecil: ia mengobrol, mengusulkan, dan memesan. Buku: Subbab 13.1 dan 13.2.1.

Empat Pertanyaan yang Harus Dijawab Setiap Antarmuka AI

Antarmuka klasik deterministik: tombol yang sama melakukan hal yang sama. Sistem AI menebak, dan tebakan dapat keliru dengan cara baru setiap kali. Tabel 13.1 memuat apa yang perlu diketahui orang.

Tabel 13.1 Empat pertanyaan pengguna tentang sistem AI mana pun.

Pertanyaan pengguna	Yang harus ditampilkan antarmuka
Apa yang bisa dilakukannya, dan seberapa baik?	Cakupan dan batasnya, sebelum pemakaian pertama
Seberapa yakin ia pada jawaban ini?	Ketidakpastian dalam bentuk yang dipahami orang
Apa yang akan dilakukannya berikutnya?	Tindakan yang diusulkan, sebelum terjadi
Bagaimana saya mengoreksi, menghentikan, atau membatalkannya?	Jalan kembali, dan jalan ke manusia

Uji cepat bagi produk AI mana pun: bila salah satu pertanyaan tak dapat dijawab hanya dari layar, desainnya berlubang. Buku: Subbab 13.2.2 dan Tabel 13.1.

Contoh yang Dikerjakan dengan Tangan

"I have a bag of laundry, pick up at 16 please": layanan — tidak disebut, *asumsikan* cuci lipat dan katakan demikian; berat — tidak disebut, *tanyakan*; kecepatan — asumsikan reguler; jemput — "at 16", besok hanya *mungkin*.

Pelanggan menjawab "about 2.5 kg". Tarif minimum 3 kg, jadi asisten harus menjelaskan *mengapa* ia menagih 3.

$$P = r_s \cdot \max(w, w_{\min}) \cdot e \tag{13.1}$$

Menurut Persamaan (13.1): $P = 8000 \cdot \max(2.5, 3) \cdot 1 = 24000$ rupiah. Baru setelah "yes" ia memesan.

r_s *tarif per kilogram layanan* s, w *beratnya,* $w_{\min} = 3$ *tarif minimum, dan* $e = 1.5$ *untuk kilat, selain itu 1. Menebak berat tidak dapat diterima karena harga bergantung padanya. Buku: Subbab 13.2.3.*

Pedoman Interaksi Manusia-AI

Amershi dkk. (CHI 2019) meringkas 150+ rekomendasi menjadi **18 pedoman**, diuji bersama 49 desainer pada 20 produk AI; kini dipelihara sebagai Microsoft HAX Toolkit.

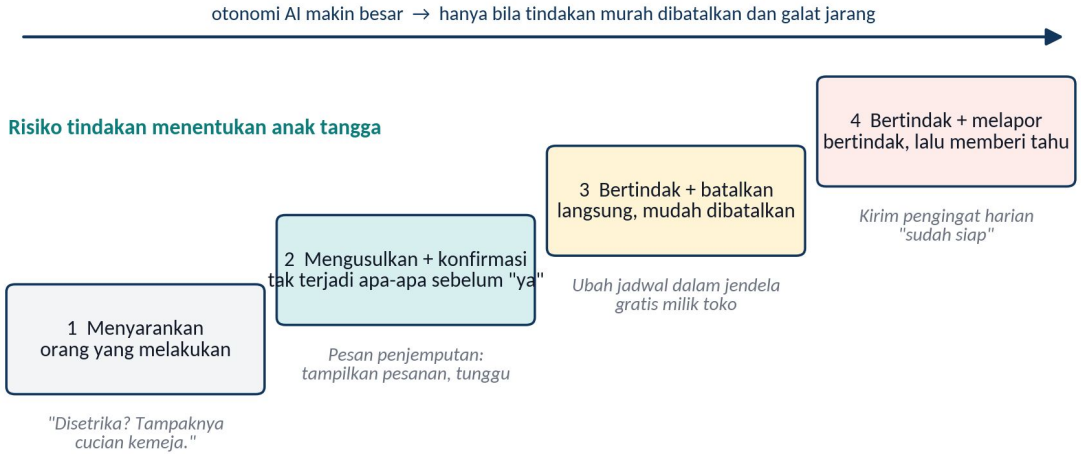
Di awal: G1 jelaskan apa yang dapat dilakukannya; G2 jelaskan seberapa baik.

Ketika keliru: G8 penolakan efisien; G9 koreksi efisien; G10 batasi layanan saat ragu; G11 jelaskan mengapa ia berbuat begitu.

Seiring waktu: G13 belajar dari perilaku; G17 kendali global (menjangkau manusia kapan saja).

People + AI Guidebook dari Google membahas wilayah yang sama dari sudut pandang tim produk; nasihatnya tentang kepercayaan: penjelasan harus membantu orang menakar ketergantungan, bukan menambahnya. Keduanya sepakat: rancanglah di sekitar apa yang terjadi ketika AI keliru. Buku: Subbab 13.3.1 dan Tabel 13.2.

Tangga Otonomi



Gambar 13.1 Tangga otonomi: risiko dan keterbalikan setiap tindakan menentukan seberapa banyak AI boleh berbuat tanpa bertanya.

Gambar 13.1 mengubah tingkat otomatisasi Parasuraman, Sheridan, dan Wickens serta prinsip inisiatif campuran Horvitz menjadi empat anak tangga: menyarankan, mengusulkan dan mengonfirmasi, bertindak dengan pembatalan, bertindak lalu melapor. Buku: Subbab 13.3.2 dan Gambar 13.1.

Otonomi Dipilih per Tindakan

> **WAWASAN KUNCI:** Otonomi bukan sifat AI; ia sifat setiap tindakan. Pembayaran, penghapusan, atau pesan atas nama pelanggan berada di bagian bawah tangga, sebaik apa pun modelnya.

Menyarankan layanan (anak tangga 1); mengusulkan penjemputan dan menunggu “ya” (anak tangga 2); mengirim sendiri pengingat harian “sudah siap” (anak tangga 4).

Horvitz (1999): bertindak sendiri hanya bila manfaat harapan melebihi biaya harapan dari keliru; bertanyalah saat ragu, karena bertanya lebih murah daripada memperbaiki.

Pemesanan mengirim seseorang bersepeda motor melintasi kota. Bahkan dengan keyakinan tinggi, tindakan itu tetap di anak tangga kedua, tempat satu kata “ya” hanya memakan satu detik. Buku: Subbab 13.3.2 dan 13.8.

Kepercayaan dan Ketergantungan

Lee dan See: tujuannya **ketergantungan yang tepat** — bersandar ketika AI benar, tidak bersandar ketika ia salah.

$$\text{ketergantungan berlebih} = \frac{\text{diterima dan AI salah}}{\text{AI salah}} \tag{13.2}$$

$$\text{ketergantungan kurang} = \frac{\text{ditolak dan AI benar}}{\text{AI benar}} \tag{13.3}$$

$$\text{tim benar} = \frac{\text{diterima dan AI benar} + \text{ditolak dan AI salah}}{\text{semua kasus}} \tag{13.4}$$

Persamaan (13.2) menghitung bias otomatisasi, Persamaan (13.3) menghitung saran baik yang terbuang, dan Persamaan (13.4) adalah yang dipedulikan toko: proporsi keputusan akhir yang benar. Buku: Subbab 13.3.3.

Temuan CHI 2026

Raees, Khan, Lykourentzou, dan Papangelis meninjau studi pengambilan keputusan manusia-AI: orang **terlalu bersandar pada saran AI tanpa menelaahnya secara analitis**.

Intervensi yang membuat orang menelaah (memutuskan lebih dulu, menunjukkan alasan) memperbaiki ketergantungan tetapi menuntut waktu dan tenaga.

Setiap desain yang menurunkan ketergantungan berlebih menambah gesekan; perancang memutuskan di mana gesekan itu diletakkan.

Tak ada makan siang gratis dalam desain ketergantungan. Ukur kedua jenis galat, bukan hanya satu. Buku: Subbab 13.3.3.

Kalibrasi: Jujurkah Keyakinannya?

Sistem *terkalibrasi* bila, di antara kasus yang ia katakan 80%, ia benar kira-kira 80% dari waktu. Jaringan saraf modern sering terlalu yakin (Guo dkk., 2017).

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left| \mathrm{acc}(b) - \mathrm{conf}(b) \right| \tag{13.5}$$

Persamaan (13.5) merata-ratakan selisih antara keyakinan yang dinyatakan dan akurasi sebenarnya atas B kelompok, dibobot ukuran kelompok.

ECE 0 berarti keyakinan yang jujur; 0,10 berarti rata-rata meleset sepuluh poin. Bila ECE tinggi, persentase menyesatkan; kata-kata dengan ambang terukur lebih melayani orang. Buku: Subbab 13.3.3.

Agen dan Alat pada 2025–2026

Agen yang mengoperasikan perangkat lunak: ChatGPT agent menarasikan di layar, bertanya sebelum membeli, mewajibkan pengawasan untuk mengirim surel, dan membiarkan orang menyela, mengambil alih, atau menghentikan.

Protokol alat bersama: Model Context Protocol menghubungkan aplikasi AI dengan alat; pada Desember 2025 ia bergabung dengan Agentic AI Foundation di bawah Linux Foundation.

Akibat desainnya: agen harus menunjukkan alat *mana* yang akan dipakai dengan data *apa*, dan membiarkan orang menyetujuinya.

Inilah anak tangga kedua dan ketiga ditambah kendali global pedoman G17. Buku: Subbab 13.3.4.

Aturan pada 2026

Pasal 50 EU AI Act (sejak 2 Agustus 2026): beri tahu orang bahwa mereka berinteraksi dengan AI; tandai konten buatan AI. Omnibus 2026 menunda kewajiban risiko tinggi ke 2027–2028, tetapi mempertahankan Pasal 50.

Indonesia: UU PDP 27/2022 dan PP 33/2026 — persetujuan bebas, spesifik, tidak ambigu, tidak digabung, dapat ditarik; tanpa permintaan persetujuan yang menipu.

Dua peraturan presiden tentang peta jalan AI (2026–2029) dan etika AI masih menunggu tanda tangan per September 2026.

Aturan praktisnya berlaku di mana saja: katakan bahwa itu mesin, katakan data apa yang dipakainya, biarkan orang menjangkau manusia. Buku: Subbab 13.3.5.

Kanvas Interaksi AI

Tabel 13.2 Sebagian baris kanvas interaksi AI untuk asisten pesanan.

Baris	Asisten pesanan
1. Tugas	Menerima pesanan jemput; bukan keluhan, bukan pengembalian uang
2. Keterbukaan	Baris pertama setiap chat
4. Ketidakpastian	Kata: yakin ≥ 0.9, mungkin ≥ 0.6, selebihnya diasumsikan
5. Tanya atau asumsikan	Menanyakan berat dan waktu; boleh mengasumsikan layanan dan kecepatan
6. Tindakan dan anak tangga	Harga: 1; pemesanan: 2; pengingat: 4
7. Batal dan henti	“no” sebelum pemesanan; UNDO selama satu pesan sesudahnya
8. Alih ke manusia	STAFF; seluruh chat diteruskan
10. Kegagalan	Katakan demikian dan alihkan ke manusia

Tabel 13.2 memuat delapan dari sepuluh baris; buku menambahkan masukan dan data (baris 3) serta ukuran (baris 9). Isilah bersama orang yang menanggung risikonya: pemrogram sendirian membiarkan baris 7, 8, dan 10 kosong. Buku: Subbab 13.4 dan Tabel 13.3.

Daftar Periksa Rilis

Seseorang yang tidak ikut membangun asisten menjawab “ya” untuk semuanya, hanya dari layar:

Apakah saya tahu ini mesin? Dapatkah saya melihat apa yang dipahaminya? Apakah saya tahu bagian mana yang ditebak?

Dapatkah saya menolak? Dapatkah saya membatalkan? Dapatkah saya menjangkau manusia?

Apakah ia bertanya sebelum membelanjakan uang saya atau bertindak atas nama saya?

Tujuh pertanyaan, satu untuk setiap kegagalan yang sungguh dialami pengguna. Buku: Subbab 13.4.

Asisten Pesanan: Cara Kerjanya

Aturan kata kunci, bukan model bahasa: pokok bahasannya antarmuka di sekitar tebakan. Mengganti aturan dengan model hanya mengubah `parse`.

`parse` mengubah pesan menjadi empat **slot** dengan keyakinannya: layanan, berat, kecepatan, jemput.

`label` mengubah keyakinan menjadi kata; `Assistant` menyimpan slot yang diingat, usulan yang menunggu, dan pemesanan terakhir dengan jendela pembatalan satu pesan.

Java memakai `record Slot` dan `LinkedHashMap`; aritmetika bilangan bulat dalam persepuluhan kilogram menjaga harga identik di kedua bahasa.

Kode 13.1 dan 13.2 di buku memuat program lengkap `laundry_assistant.py` dan `LaundryAssistant.java`. Buku: Subbab 13.5.1 dan 13.5.2.

Empat Keputusan Desain di Kode

Keterbukaan lebih dulu: “[Automated assistant. Type STAFF at any time to reach a person.]”

Bertanya, bukan menebak (G10): berat atau waktu jemput yang kosong menghasilkan pertanyaan, dengan jalan keluar bagi yang tidak tahu.

Tandai asumsi (G11): “assumed – please check” muncul tepat di tempat nilai bawaan dipakai.

Ingat percakapan (G13): slot diganti hanya bila pesan baru menyatakannya lebih jelas.

Setiap keputusan terpetakan ke satu baris Tabel 13.2 di buku. Buku: Subbab 13.5.1.

Menjalankannya: Percakapan 1

```
=== Conversation 1 ===
Bot : [Automated assistant. Type STAFF at any time to reach a person.]
You : Hi, can you wash 4 kg of clothes and iron them? Pick up tomorrow at 9.
Bot : Here is what I understood:
        service  wash + iron      (sure)
        kg       4 kg            (sure)
        speed    regular         (assumed - please check)
        pickup   tomorrow 09:00   (sure)
        price    4 kg x Rp10,000 = Rp40,000
      Book this pickup? (yes / no)
You : yes
Bot : Booked: pickup tomorrow 09:00.
      Type UNDO in your next message to cancel.
You : undo
Bot : Cancelled. Nothing will be picked up.
```

Semua diberikan kecuali kecepatan; asisten mengusulkan, menandai asumsinya, memesan setelah “yes”, dan membatalkan setelah “undo”. Buku: Subbab 13.5.3.

Menjalankannya: Percakapan 2

```
You : I have a bag of laundry, pick up at 16 please
Bot : How many kilograms, roughly? If you are not sure,
      we weigh it at pickup and send the exact price
      before washing.
You : about 2.5 kg
Bot : Here is what I understood:
        service  wash + fold      (assumed - please check)
        kg       2.5 kg           (sure)
        speed    regular          (assumed - please check)
        pickup   tomorrow 16:00   (likely - please check)
        price    3 kg x Rp8,000 = Rp24,000
        (minimum charge is 3 kg)
      Book this pickup? (yes / no)
```

Asisten menanyakan berat yang kosong, mengingat jam dari pesan pertama, menandai harinya sebagai sekadar mungkin, dan menjelaskan tarif minimum. “undo” kedua di chat ini datang terlambat dan ditolak dengan sopan. Buku: Subbab 13.5.3.

Mengukur Ketergantungan: Studinya

Fitur berikutnya: saran kepada staf apakah pakaian aman dicuci mesin, dari foto label perawatannya.

Versi A hanya menampilkan saran; **versi B** menambahkan keyakinan dalam kata dan satu baris alasan.

Sebelum studi nyata, log *simulasi* 200 kasus per versi menguji analisisnya. Kebenaran yang ditanamkan: model terlalu yakin 8 poin; staf di B lebih jarang menerima bila keyakinannya rendah atau alasannya tampak salah.

`make_log.py` *menulis log dengan seed;* `reliance.py` *dan* `Reliance.java` *menghitung Persamaan (13.2), Persamaan (13.3), Persamaan (13.4), dan Persamaan (13.5). Buku: Subbab 13.6.*

Hasilnya

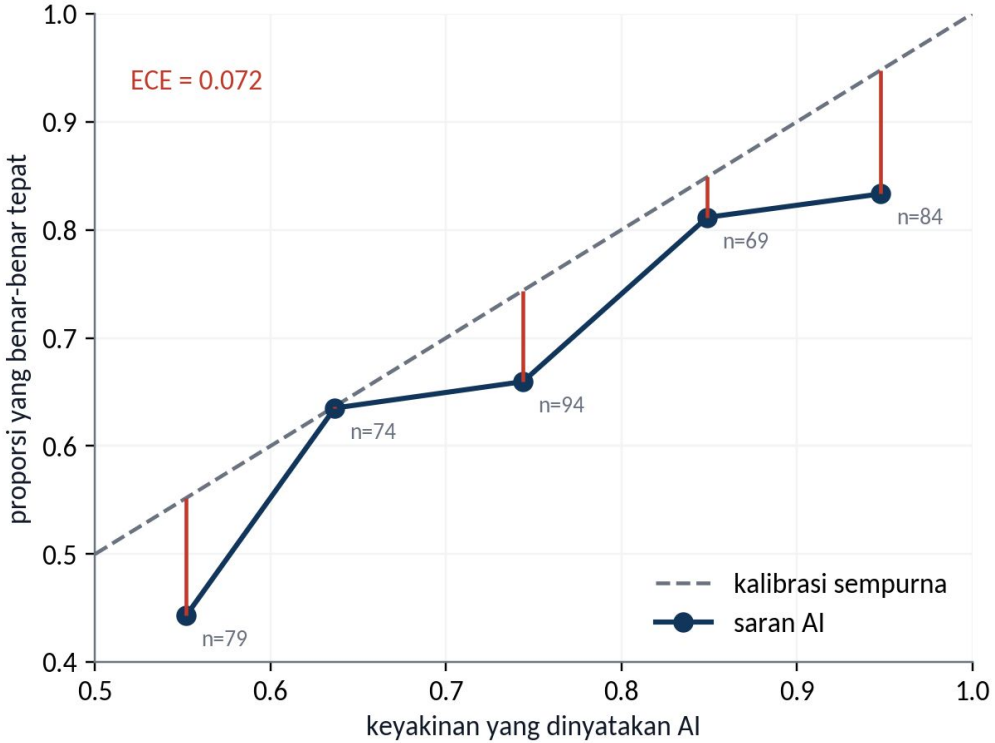
version	cases	AI right	over-reliance	under-reliance	team right
A	200	66.0%	91.2%	6.1%	65.0%
B	200	69.0%	48.4%	26.1%	67.0%

confidence bin	cases	mean confidence	actually right
0.5 - 0.6	79	0.552	0.443
0.6 - 0.7	74	0.637	0.635
0.7 - 0.8	94	0.744	0.660
0.8 - 0.9	69	0.849	0.812
0.9 - 1.0	84	0.948	0.833

expected calibration error (ECE) = 0.072

B memangkas ketergantungan berlebih dari 91,2% menjadi 48,4% tetapi menaikkan ketergantungan kurang dari 6,1% menjadi 26,1%; keputusan akhir yang benar hanya bergerak dari 65,0% ke 67,0%. Menampilkan keyakinan lebih mengubah cara orang keliru daripada seberapa sering. Buku: Subbab 13.6.

Sebabnya: Model yang Terlalu Yakin



Gambar 13.2 Kalibrasi model label perawatan hasil simulasi: titik di bawah diagonal berarti model mengklaim melebihi yang pantas.

Gambar 13.2: di kelompok teratas model mengklaim 0,948 tetapi benar 0,833; ECE-nya 0,072. Pelajarannya: kalibrasikan dulu, baru tampilkan keyakinan, lalu ukur ulang ketergantungan. Buku: Subbab 13.6 dan Gambar 13.2.

Menguji Janji

```
ok   service
ok   weight
ok   pickup
ok   price 40,000
ok   minimum 3 kg, express
ok   confirm books
ok   undo expires after one message
all checks passed
```

“Undo expires after one message” bukan rincian implementasi; itu janji yang dibuat antarmuka. Uji yang mengunci janji itu mencegah perubahan kelak merusaknya diam-diam. Buku: Subbab 13.7.

Membuatnya Lebih Baik

Jelaskan tebakannya, singkat: “besok, karena Anda menulis malam hari dan tidak menyebut hari”.

Tetapkan ambang dari data: “sure” harus benar setidaknya sembilan dari sepuluh kali.

Tanyakan menurut urutan biaya: slot yang galatnya paling mahal didahulukan.

Tinggalkan catatan yang terlihat: ringkasan yang dapat diteruskan atau ditangkap layar pelanggan.

Menggoda tetapi keliru: memesan tanpa konfirmasi bila semua slot “sure”.

Setiap perbaikan mengalir dari sebuah pedoman atau sebuah pengukuran. Buku: Subbab 13.8.

Memakainya Sungguhan

Katering Pak Joko: menu, jumlah kotak, waktu, alamat — dan **kebutuhan diet**, yang *tidak pernah* boleh diasumsikan. Setiap penyebutan alergi diteruskan ke manusia.

Catat usulan, keyakinan, dan hasilnya, dengan persetujuan tersendiri yang sederhana sesuai PP 33/2026.

Putuskan sejak awal apa yang terjadi bila model atau jaringan mati; baca chat nyata setiap minggu.

Aturannya: untuk slot yang nilai bawaannya bisa mencelakakan orang bila salah, tidak ada nilai bawaan. Buku: Subbab 13.9.

IMK dalam Praktik: Profitina

Kegagalan yang jujur: bila tak ada model lokal, aplikasi mengatakannya alih-alih diam-diam menyajikan templat.

Gerbang mutu: draf AI disajikan hanya bila mengalahkan templat bawaan; setelah revisi September 2026 draf AI menang pada 5 dari 6 kasus tolok ukur.

Aplikasi menyarankan, pengguna memutuskan: sinyal di balik setiap skor; tak ada yang diterbitkan tanpa tindakan eksplisit.

AI Profitina (Qwen2.5) berjalan lokal, sehingga data pengguna tak pernah meninggalkan server untuk pemrosesan AI; alur umpan baliknya membuat "belajar seiring waktu" terlihat. Buku: Subbab 13.10.

Kesalahan yang Sering Terjadi

Menyembunyikan bahwa ia mesin; menebak yang harus ditanyakan; persentase yang tak terkalibrasi.

Hanya mengukur ketergantungan berlebih — laporkan Persamaan (13.2), Persamaan (13.3), dan Persamaan (13.4) bersama-sama.

Satu tingkat otonomi untuk seluruh produk; tanpa jalan keluar; melupakan percakapan; pembatalan yang bukan pembatalan.

Setiap kesalahan di buku disertai cara memperbaikinya. Buku: Subbab 13.11.

Ringkasan: Lima Hal untuk Diingat

1. Chat menjawab, kopilot mengusulkan, agen bertindak; makin banyak yang dapat dilakukan AI, makin banyak yang harus ditunjukkan antarmukanya dan makin mudah dihentikan.

2. Jawab empat pertanyaan Tabel 13.1 hanya dari layar.

3. Pilih otonomi per tindakan dari risiko dan keterbalikannya (Gambar 13.1).

4. Ukur ketergantungan berlebih, ketergantungan kurang, dan keputusan akhir yang benar; kalibrasikan sebelum menampilkan keyakinan (Persamaan (13.5)).

5. Terbuka, minimalkan data, dan selalu sediakan manusia — hukum 2026 dan desain yang baik sepakat.

Buku: Subbab 13.12.

Pertemuan Berikutnya

Bab 14 — UX Berbasis Data. Peristiwa, corong, metrik HEART, dan eksperimen A/B yang dapat dipercaya.

Log ketergantungan hari ini adalah jenis bukti yang diajarkan Bab 14 untuk dikumpulkan dalam skala besar.

Sebelum itu: kerjakan Soal 13.1, 13.3, 13.4, dan 13.6.

Buku: Subbab 13.13 dan 13.14.

Kuliah 14 — UX Berbasis Data: Analitik, Eksperimen, dan Metrik

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 14

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini kita mengganti pendapat dengan apa yang benar-benar dilakukan orang: peristiwa, corong, metrik yang diturunkan dari tujuan, dan eksperimen A/B yang kita periksa sebelum kita percayai, semuanya dikumpulkan dengan persetujuan. Buku: Bab 14, Subbab 14.1–14.12.

Yang Dibahas Pertemuan Ini

1. **Masalahnya** — situs pemesanan warung yang ramai pengunjung tetapi sepi pesanan, dan tiga pendapat yang bersaing.
2. **Peristiwa, corong, metrik** — konversi langkah dan keseluruhan, serta di mana perbaikan berbuah.
3. **Teorinya** — HEART, angka Indonesia 2026 yang dibaca kritis, eksperimen terkendali, empat jebakan, metrik fitur AI, persetujuan.
4. **Metodenya** — rencana eksperimen satu halaman.
5. **Programnya** — corong dan pemeriksaan A/B dalam Java dan Python, pada log yang sehat dan yang rusak.

Setiap keluaran yang ditampilkan adalah keluaran nyata program-program buku. Buku: Bab 14.

Checkout Bu Tini yang Sepi

Warung Bu Tini punya situs pemesanan: menu → keranjang → alamat → pembayaran → selesai. Ratusan orang membuka menu setiap minggu; sedikit pesanan yang masuk.

Tiga penjelasan: fotonya, menu yang panjang, warna yang kuno. Masing-masing tebakan dan seminggu kerja.

Yang tak diketahui siapa pun adalah *di mana* orang berhenti. Beberapa peristiwa yang dipilih dengan baik menjawabnya.

Penurunan tercuram ternyata di formulir alamat. Raka membuat versi B: alamat diingat, QRIS satu ketukan. Benarkah B lebih baik?

Menjawab “benar-benar lebih baik, atau sekadar minggu yang bagus?” dengan jujur adalah tugas eksperimen terkendali. Buku: Subbab 14.1.

Apa yang Dikatakan Orang dan Apa yang Dilakukannya

Wawancara dan *think-aloud* (Bab 5 dan 7) menjelaskan **mengapa**, dengan segelintir orang.

Analitik mencatat **apa** yang dilakukan ribuan orang, tanpa bertanya, dan tak dapat mengatakan mengapa.

Corong menunjukkan 44% meninggalkan checkout; hanya percakapan yang mengungkap bahwa mengetik “Jl. Kenanga Blok C-12 No. 7, RT 03/RW 05” di ponsel kecil adalah penyebabnya.

Data-informed, bukan data-driven: angka memberi masukan bagi keputusan yang diambil orang yang memahami konteksnya. Buku: Subbab 14.2.1.

Peristiwa, Corong, dan Metrik

DEFINISI. *Peristiwa* mencatat bahwa sesuatu terjadi (siapa, apa, kapan). *Corong* adalah daftar langkah berurutan beserta pengguna yang mencapai setiap langkah. *Metrik* adalah angka dari peristiwa yang memantau suatu tujuan. *Laju konversi* adalah “pengguna yang melakukan X dibagi pengguna yang dapat melakukannya”.

$$s_k = \frac{n_k}{n_{k-1}}, k = 2, \ldots, K \tag{14.1}$$

$$c = \frac{n_K}{n_1} = s_2 \cdot s_3 \cdots s_K \tag{14.2}$$

Persamaan (14.1) adalah konversi langkah; Persamaan (14.2) konversi keseluruhan. Karena langkah-langkah saling mengalikan, memperbaiki langkah terlemah paling banyak menaikkan konversi keseluruhan. Buku: Subbab 14.2.2.

Contoh yang Dikerjakan dengan Tangan

1.000 membuka menu → 610 keranjang → 430 checkout → 245 pembayaran → 228 selesai.

Langkah: 0,61; 0,705; **0,570**; 0,931. Keseluruhan: $228/1000 = 0.228$.

Naikkan langkah terlemah menjadi 0,66: keseluruhan $0.61 \cdot 0.705 \cdot 0.66 \cdot 0.931 \approx 0.264$, sekitar **36 pesanan lebih** per seribu.

Sempurnakan langkah pembayaran yang sudah kuat: paling banyak $0.228 \cdot (1/0.931 - 1) \approx 0.017$, **17 pesanan**.

Corong tidak memberi tahu cara memperbaiki checkout; ia memberi tahu di mana perbaikan paling berbuah. Buku: Subbab 14.2.3.

Memilih Metrik: HEART

Rodden, Hutchinson, dan Fu (Google, CHI 2010): **H**appiness, **E**ngagement, **A**doption, **R**etention, **T**ask success — dan proses **Goals–Signals–Metrics**.

Tabel 14.1 Proses Goals–Signals–Metrics dari HEART diterapkan pada warung; keterlibatan sengaja ditinggalkan.

Kategori	Tujuan	Metrik
Happiness	Memesan terasa mudah	Proporsi yang menjawab 6–7 untuk “memesan itu mudah”
Adoption	Pelanggan langsung mencoba memesan daring	Pelanggan daring baru per minggu
Retention	Pelanggan daring memesan lagi	Laju pesan ulang 30 hari
Task success	Orang yang ingin memesan dapat menyelesaikannya	Konversi keseluruhan; waktu memesan
Task success	Pesanan tiba dengan benar	Pesanan salah alamat per 100

Tabel 14.1: bagi situs pemesanan, waktu yang lebih lama di halaman berarti kesulitan, bukan keterlibatan. Metrik yang naik ketika pengalaman memburuk lebih buruk daripada tanpa metrik. Buku: Subbab 14.3.1.

Indonesia pada 2026, Dibaca Kritis

APJII 2026: 81,7% penduduk daring (sekitar 235 juta), naik dari 80,66%; Jawa 85,95%, Maluku–Papua 69,74%; kota 84,75%, desa 78,18%.

StatCounter, Agustus 2026: 60,59% *tampilan halaman* dari desktop, 38,98% seluler, 0,43% tablet.

Proporsi tampilan halaman menghitung tampilan di situs yang dilacak — bukan orang. Komputer kantor, penempatan kode pelacak, dan bot menggesernya.

Selalu tanyakan apa yang dihitung sebuah angka sebelum memutuskan maknanya. Bagi Bu Tini, komposisi perangkat yang penting hanya yang ada di lognya sendiri. Buku: Subbab 14.3.2.

Eksperimen Terkendali

Perbandingan antarminggu mencampurkan desain dengan gajian, cuaca, dan hari libur. **Acaklah** pengunjung antara A dan B pada waktu yang sama.

$$z = \frac{p_B - p_A}{\sqrt{\acute{p}(1-\acute{p})\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}} \tag{14.3}$$

$$d \pm 1.96\sqrt{\frac{p_A(1-p_A)}{n_A} + \frac{p_B(1-p_B)}{n_B}} \tag{14.4}$$

Persamaan (14.3) menguji apakah selisihnya melampaui kebetulan ($|z| > 1.96$ berarti $p < 0.05$); Persamaan (14.4) memberi besar efek beserta ketidakpastiannya, yang dibutuhkan pemilik warung. Buku Kohavi, Tang, dan Xu adalah panduan bakunya. Buku: Subbab 14.3.3.

Berapa Pengguna yang Diperlukan?

$$n = \frac{(z_{\alpha/2} + z_{\beta})^2 [p_1(1-p_1) + p_2(1-p_2)]}{(p_2 - p_1)^2} \tag{14.5}$$

Dengan $z_{\alpha/2} = 1.96$ (5%, dua sisi) dan $z_{\beta} = 0.8416$ (daya 80%), Persamaan (14.5) memberi pengguna per varian.

Memperkecil setengah efek yang ingin dideteksi **melipatempatkan** sampelnya.

Contoh: 20% → 22% memerlukan 6.507 per varian.

Hitung n sebelum eksperimen; ia menentukan berapa lama eksperimen berjalan. Buku: Subbab 14.3.3.

Empat Jebakan

Ketidaksesuaian rasio sampel: kelompok yang timpang berarti pengguna yang hilang. Periksa lebih dulu, berhenti bila $p<0.001$ (Fabijan dkk., KDD 2019).

$$\chi^2 = \frac{(n_A - e)^2}{e} + \frac{(n_B - e)^2}{e} \tag{14.6}$$

Mengintip: berhenti ketika p pertama kali turun di bawah 0,05 menggelembungkan positif palsu.

Kebaruan dan belajar: jalankan setidaknya satu, lebih baik dua, siklus mingguan penuh.

Hukum Goodhart: ukuran yang menjadi sasaran berhenti menjadi ukuran yang baik — lindungi **metrik pagar**.

Pada Persamaan (14.6), $e = (n_A + n_B)/2$ untuk rencana 50/50, dengan satu derajat bebas. Menyembunyikan ongkos antar akan menaikkan konversi sekaligus menjadi pola gelap; metrik pagar menangkapnya. Buku: Subbab 14.3.4.

Metrik untuk Fitur AI

Laju penerimaan naik ketika saran membaik — dan ketika orang berhenti membacanya.

Tambahkan **laju suntingan**, pasangan **penolakan-dan-hasil** (log Bab 13), dan **metrik pagar bahaya**.

Metrik utama: hasil usaha (pesanan selesai dengan benar), dengan penerimaan hanya sebagai konteks.

Versi yang menaikkan penerimaan sementara keputusan akhir yang benar tetap datar telah membuat orang lebih percaya, bukan sistem lebih berguna. Buku: Subbab 14.3.6.

Data dengan Persetujuan

PP 33/2026 (pelaksana UU PDP 27/2022): persetujuan bebas, spesifik, tidak ambigu, tidak digabung; dapat ditarik; pemrosesan berhenti dalam 3 × 24 jam.

Alamat antar: diperlukan layanan. Peristiwa analitik: memerlukan persetujuan **tersendiri**.

Consent mode GA4: sejak 15 Juni 2026, `analytics_storage` dan `ad_storage` diatur terpisah.

Kepatuhan paling sederhana: peristiwa pihak pertama yang anonim di server sendiri, tanpa pernah memasukkan alamat atau nomor telepon ke analitik. Buku: Subbab 14.3.5.

Rencana Eksperimen

Tabel 14.2 Rencana eksperimen checkout warung, diputuskan sebelum pengunjung pertama.

Butir	Eksperimen checkout warung
Hipotesis	43% berhenti di langkah alamat; mengingatnya dan QRIS satu ketukan akan menaikkan pesanan
Varian	A: sekarang; B: alamat diingat + QRIS
Metrik utama	`order_done` / `view_menu`
Metrik pagar	Salah alamat; keluhan; pembayaran gagal
Efek minimum terdeteksi	+2 poin persentase
Ukuran sampel	Sekitar 7.000 per varian; setidaknya dua minggu penuh
Pemeriksaan lebih dulu	Rasio sampel ($p \geq 0.001$)
Aturan keputusan	Luncurkan B bila selang 95% di atas 0 dan metrik pagar aman

Tabel 14.2 mencegah tim memilih metrik yang menyanjung belakangan dan mengubah perbedaan pendapat menjadi eksperimen tersendiri. Buku: Subbab 14.4 dan Tabel 14.2.

Log Peristiwa dan Simulasinya

Satu baris per peristiwa: `user_id,variant,step`.

`make_events.py`: 5.000 pengunjung simulasi, penempatan dengan lemparan koin, peluang melanjutkan per langkah; hanya checkout → bayar yang berbeda (A 0,58, B 0,66).

Log kedua, `events_srm.csv`, diam-diam kehilangan 15% pengunjung B — bug pencatatan di ponsel lama.

Menguji analisis terhadap kebenaran yang Anda tetapkan sendiri adalah cara memercayainya sebelum lalu lintas nyata datang. Buku: Subbab 14.5.1.

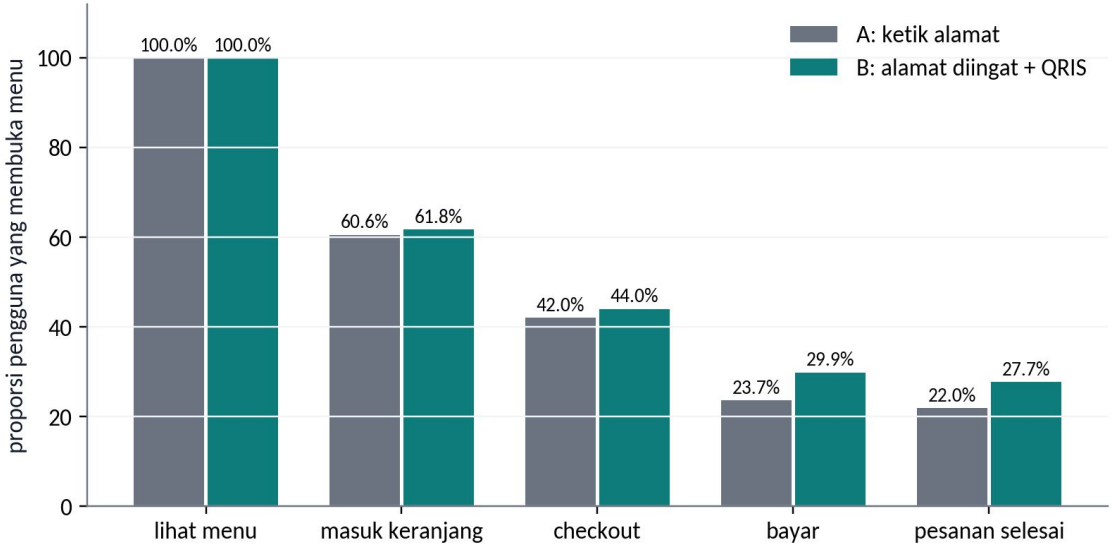
Program Corong

`funnel.py` / `Funnel.java` membaca log menjadi himpunan id pengguna per varian dan langkah: hitung **pengguna yang berbeda**, bukan peristiwa.

```
Variant A
  step          users  from previous  from start
  view_menu      2481          100.0%      100.0%
  add_to_cart    1503           60.6%       60.6%
  checkout       1043           69.4%       42.0%
  pay             588           56.4%       23.7%
  order_done      545           92.7%       22.0%
Variant B
  step          users  from previous  from start
  view_menu      2519          100.0%      100.0%
  add_to_cart    1556           61.8%       61.8%
  checkout       1109           71.3%       44.0%
  pay             753           67.9%       29.9%
  order_done      698           92.7%       27.7%
```

`TreeMap` dan `LinkedHashMap` di Java menjaga keluarannya identik dengan Python. Buku: Subbab 14.5.2 dan 14.5.3.

Corong dalam Gambar



Gambar 14.1 Corong checkout: proporsi pembuka menu yang mencapai setiap langkah, untuk A dan B.

Gambar 14.1: identik sampai checkout; B mempertahankan 67,9% di pembayaran sedangkan A 56,4%, dan jaraknya melebar menjadi 27,7% lawan 22,0%. Hanya deskriptif: ia tak dapat menyingkirkan kebetulan atau pembagian yang rusak. Buku: Subbab 14.5.3 dan Gambar 14.1.

Pemeriksaan A/B, Berurutan

`abtest.py` / `AbTest.java`: pertama rasio sampel (Persamaan (14.6)); hanya bila lolos, selisih dan selangnya (Persamaan (14.4)), kenaikan, z (Persamaan (14.3)), dan ukuran sampel (Persamaan (14.5)).

Kedua nilai-p berasal dari satu fungsi: $\operatorname{erfc}\left(|z|/\sqrt{2}\right)$ dan $\operatorname{erfc}\left(\sqrt{\chi^2/2}\right)$.

Java tak punya `erfc`, jadi kedua bahasa memakai hampiran terbitan yang sama (galat $< 1.2 \times 10^{-7}$) dan mencetak digit yang sama.

Urutan pemeriksaannya adalah urutan rencana. Buku: Subbab 14.6.1 dan 14.6.2.

Menjalankannya pada Log yang Sehat

```
users: A = 2481, B = 2519
sample ratio check: chi2 = 0.29, p = 0.5910
orders:  A = 545 (21.97%), B = 698 (27.71%)
difference = +5.74 points, 95% CI [+3.35, +8.13]
relative lift = +26.1%, z = 4.70, p < 0.0001
users per variant needed to detect +2 points from 22.0%: 6940
```

Pembagiannya lolos; seluruh selang di atas nol dan di atas minimum dua poin. Namun mendeteksi perubahan dua poin memerlukan 6.940 per varian: dengan efek sebenarnya yang lebih kecil, "tidak signifikan" berarti "kami tak dapat membedakan". Buku: Subbab 14.6.3.

Menjalankannya pada Log yang Rusak

```
users: A = 2472, B = 2155
sample ratio check: chi2 = 21.72, p < 0.0001
STOP: sample ratio mismatch - find the bug before reading on
```

Tak seorang pun dapat mengatakan apakah 15% pengguna B yang hilang akan memesan atau pergi. Hasilnya tak dapat diperbaiki belakangan.

Program menolak melanjutkan: hasil dengan ketidaksesuaian rasio sampel tidak tepercaya sampai penyebabnya ditemukan. Buku: Subbab 14.6.4.

Menguji Kode Statistik

```
ok   erfc(0) = 1
ok   |z| = 1.96 gives p = 0.05
ok   a perfect split has chi2 = 0
ok   equal rates give z = 0
ok   sample size 20% -> 22%
all checks passed
```

Pemeriksaan bahwa $|z| = 1.96$ *memberi* $p = 0.05$ *menambatkan kode pada angka yang disepakati setiap tabel statistik; lupa* $\sqrt{2}$ *langsung tertangkap. Buku: Subbab 14.7.*

Membuatnya Lebih Baik

Buat segmen, tetapi rencanakan — dua puluh segmen tak terencana menemukan satu yang “signifikan” karena kebetulan.

CUPED — sesuaikan dengan perilaku sebelum eksperimen untuk menyempitkan selang (Deng dkk., 2013).

Metrik pagar di samping metrik utama; peta panas ketukan; pantau metrik setelah peluncuran.

Pasangkan angka dengan manusia — lima sesi berpikir keras menyumbang hipotesis berikutnya.

Peta panas ketukan adalah kerabat murah pelacakan mata yang tidak mencatat apa pun tentang orangnya. Buku: Subbab 14.8.

Memakainya Sungguhan

Kumpulkan peristiwa di server sendiri, setelah persetujuan analitik tersendiri; jangan pernah nama, telepon, atau alamat.

Tempatkan varian secara deterministik dari *hash* id anonim.

Simpan catatan keputusan: rencana, keluaran, keputusan — untuk setiap eksperimen.

Laundry Pak Made di Bali memakai program yang sama dengan langkah berbeda (`open_form`, `choose_service`, `choose_time`, `confirm`). Buku: Subbab 14.9.

IMK dalam Praktik: Profitina

Skor visibilitas AI mingguan per merek, dengan tren, ekspor CSV, pindai ulang terjadwal, dan peringatan.

Tujuan: dikutip mesin jawaban AI; sinyal: sifat situs yang terukur; metrik: skor — maka setiap sinyal dijelaskan, untuk menghindari Goodhart.

Penghitung “N baru” hanya menghitung unggahan tingkat atas; satu permintaan menghimpun dasbor; pengguna dapat mengekspor atau menghapus datanya.

Definisi metrik adalah keputusan antarmuka. Buku: Subbab 14.10.

Kesalahan yang Sering Terjadi

Metrik yang mudah; peristiwa alih-alih pengguna; tampilan halaman dibaca sebagai orang; perbandingan sebelum-sesudah.

Tanpa pemeriksaan rasio sampel; mengintip; p tanpa besar efek; “tidak signifikan” dibaca “tidak ada efek”.

Tanpa metrik pagar; data pribadi di dalam analitik.

Setiap kesalahan di buku disertai cara memperbaikinya. Buku: Subbab 14.11.

Ringkasan: Lima Hal untuk Diingat

1. Corong menghitung pengguna berbeda per langkah; perbaiki langkah terlemah lebih dulu (Persamaan (14.2)).

2. Turunkan metrik dari tujuan dengan HEART (Tabel 14.1); buang metrik yang naik ketika keadaan memburuk.

3. Acak, tentukan ukuran di muka (Persamaan (14.5)), dan laporkan selangnya (Persamaan (14.4)), bukan hanya p.

4. Periksa rasio sampel lebih dulu (Persamaan (14.6)); lindungi metrik pagar.

5. Kumpulkan peristiwa anonim, dengan persetujuan tersendiri, sesuai PP 33/2026.

Buku: Subbab 14.12.

Pertemuan Berikutnya

Bab 15 — Antarmuka Masa Depan. Komputasi spasial, wearable, dan ambient; ukuran sebagai sudut pandang; pandangan mata dan sentuhan Midas.

Sebelum itu: kerjakan Soal 14.1, 14.2, 14.3, dan 14.6.

Buku: Subbab 14.13 dan 14.14.

Kuliah 15 — Antarmuka Masa Depan: Spasial, Wearable, dan Ambient

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 15

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini antarmuka meninggalkan persegi panjang: melayang di headset, duduk di kacamata, mengetuk pergelangan, berpendar di dinding. Kita belajar mengukur apa pun dengan sudut yang ditutupnya, mencegah memandang menjadi memilih, dan merancang bagi orang-orang yang tak akan pernah memiliki perangkatnya. Buku: Bab 15, Subbab 15.1–15.12.

Yang Dibahas Pertemuan Ini

1. **Masalahnya** — Mbah Sumi melewatkan gilirannya di ruang tunggu puskesmas.
2. **Gagasannya** — spasial, *wearable*, ambient; ukuran adalah sudut.
3. **Teorinya** — perangkat 2025–2026, pandangan dan sentuhan Midas, *wearable* yang dibaca sekilas, teknologi yang tenang, kenyamanan, keselamatan, privasi, aksesibilitas, gagasan spasial di layar datar.
4. **Metodenya** — daftar periksa konteks pemakaian.
5. **Programnya** — kalkulator ukuran target dan analisis waktu tatap dalam Java dan Python.

Setiap keluaran yang ditampilkan adalah keluaran nyata program-program buku. Buku: Bab 15.

Mbah Sumi dan Layar Antrean

Mbah Sumi, 72 tahun, menunggu pemeriksaan tekanan darah. Nomornya, L-017, akan tampil di televisi lima meter jauhnya dan dipanggil lewat pengeras suara yang bergema.

Matanya tak lagi tajam dan pendengarannya lebih buruk. Dua kali ia melewatkan giliran dan menunggu satu jam lagi.

Kacamata, jam tangan, dan *headset* menjanjikan bantuan — tetapi tak satu pun pasien punya kacamata pintar, dan sedikit yang punya jam tangan pintar.

Perbaikan pertama adalah nomor yang lebih besar di televisi, diukur dari bangku, tanpa biaya dan menolong semua orang. Geometri yang sama mengukur tombol melayang di headset. Buku: Subbab 15.1.

Melampaui Persegi Panjang

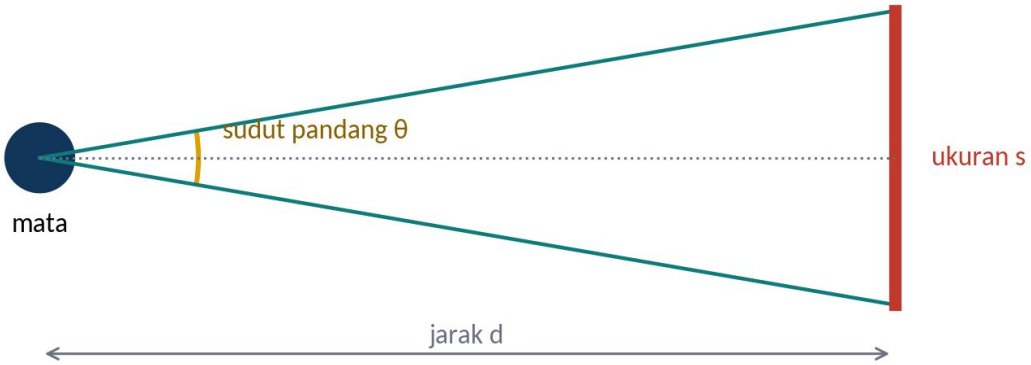
> **DEFINISI.** Antarmuka *spasial* menempatkan konten di ruang tiga dimensi di sekitar orang. Antarmuka *wearable* dibawa di tubuh dan dipakai dalam lirikan singkat. Antarmuka *ambient* tertanam di lingkungan dan memberi tahu di tepi perhatian.

Pada ketiganya, perhatian biasanya berada di tempat lain: berjalan, memasak, bercakap, terkantuk di bangku.

Dua tuntutan mengikutinya: **mudah ditangkap sekilas** dan **sulit terpicu tanpa sengaja**.

Sebagian besar isi bab mengalir dari dua tuntutan itu. Buku: Subbab 15.2.1.

Ukuran adalah Sudut



Gambar 15.1 Sudut pandang benda berukuran s pada jarak d.

$$\theta = 2\arctan\left(\frac{s}{2d}\right) s = 2d\tan\left(\frac{\theta}{2}\right) \theta_{\text{deg}} \approx 57.3\frac{s}{d} \tag{15.1}$$

Gambar 15.1 dan Persamaan (15.1): sudut yang sama berasal dari benda kecil yang dekat atau benda besar yang jauh. Kuku ibu jari pada jarak sepanjang lengan menutup kira-kira satu sampai dua derajat. Buku: Subbab 15.2.2.

Contoh yang Dikerjakan dengan Tangan

Nomor 4 cm dilihat dari 5 m: $\theta = 2\arctan(4/1000) \approx 0.46^\circ$, sekitar 27,5 menit busur.

Baris 20/20 bagan Snellen berukuran 5 menit; orang yang hanya dapat membaca baris 20/100 memerlukan 25 — nomor itu di batas keterbacaan.

Untuk 1,5 derajat dari 5 m: $s = 1000 \cdot \tan(0.75^\circ) = 13.1$ cm — sekitar tiga kali lebih tinggi.

Pada televisi 43 inci (tinggi layar sekitar 53 cm), nomor 13 cm masih menyisakan tempat untuk nama layanan dan nomor berikutnya. Buku: Subbab 15.2.3.

Perangkat 2025–2026

Tabel 15.1 Perangkat yang membentuk desain spasial dan wearable pada 2026, dengan masukan dan keluarannya.

Perangkat	Masukan utama	Keluaran utama
Apple Vision Pro dengan M5 (Okt. 2025)	Mata, tangan, suara	Micro-OLED hingga 120 Hz, audio spasial
Samsung Galaxy XR, Android XR (Okt. 2025)	Mata, tangan, suara dengan Gemini	Layar dengan *passthrough*
Meta Ray-Ban Display + Neural Band (Sep. 2025)	Gestur gelang EMG, suara	Layar kecil di satu sisi
Ray-Ban Meta Audio (Sep. 2026)	Suara, sentuhan	Hanya audio, tanpa kamera
Kacamata audio Android XR (musim gugur 2026)	Suara dengan Gemini, ketuk bingkai	Audio
Apple Watch, watchOS 26	Sentuhan, *double tap*, *wrist flick*	Layar, getaran

Tabel 15.1: masukan menjauh dari permukaan; keluaran menyusut menjadi lirikan dan bunyi; asisten AI ada di kebanyakan perangkat. Pada April 2026 MacRumors melaporkan penjualan Vision Pro yang lemah meski ada pembaruan M5: headset mahal tetap pasar yang kecil. Buku: Subbab 15.3.1.

Memandang Bukan Memilih

Mata bergerak tiga atau empat kali sedetik, entah orangnya menginginkan sesuatu atau tidak.

Sentuhan Midas (Jacob, 1990): bila semua yang dipandang teraktifkan, orang tak dapat memandang tanpa memicu.

Jalan keluarnya: **sinyal kedua** (pandang + cubit di Vision Pro dan Galaxy XR; gestur EMG di Neural Band) atau **waktu tatap**.

Pandangan juga tidak presisi dan berbeda antarorang (Feit dkk., CHI 2017); Apple meminta target spasial setidaknya 60 *point*, dibandingkan 44 di iOS.

Waktu tatap penting ketika tangan tak dapat dipakai: disabilitas motorik, tangan steril atau kotor, kios umum dengan pelacak mata. Buku: Subbab 15.3.2.

Wearable: Lirik, Ketuk, Lanjut

Tampilkan satu hal: nomor dan berapa orang di depan, bukan menu.

Utamakan getaran: ketukan di pergelangan bersifat pribadi dan tak perlu dibaca.

Sediakan gestur tanpa bidik: *double tap* (2023) dan *wrist flick* (watchOS 26) menolong siapa pun yang tangannya sibuk.

Sedikit pola yang berbeda jelas: ketukan ganda lembut = “bersiaplah”; getaran panjang kuat = “maju sekarang”.

Dua pola dapat dipelajari; delapan akan tertukar. Setiap pola mendapat satu makna. Buku: Subbab 15.3.3.

Suara di Tubuh

Kacamata audio hampir sepenuhnya bergantung pada suara. Keluaran lisan linear dan cepat berlalu: **kata terpenting di depan** — “L-017, giliran Anda, pintu tiga”.

Ulangi sebelum bertindak, seperti asisten laundry di Bab 13 mengulang pesanan.

Berbicara itu terdengar orang lain: pasien mungkin tak ingin menyebut keluhannya keras-keras; sediakan alternatif tanpa suara.

Rencanakan untuk bahasa sesungguhnya: pengenal suara yang dilatih dengan bahasa Indonesia baku kurang andal untuk tuturan Jawa atau Madura — ujilah dengan suara pengguna nyata.

Suara dan gestur pergelangan adalah dua masukan yang tak perlu dibidik, dan keduanya menolong orang yang tangan atau matanya sibuk. Buku: Subbab 15.3.3.

Teknologi yang Tenang

Weiser (1991): komputer menghilang ke dalam benda sehari-hari. Weiser dan Brown (1996): teknologi harus **tenang**, memindahkan informasi antara pusat dan pinggiran perhatian.

Lampu di atas setiap pintu yang makin terang ketika antrean bergerak; pita berwarna di televisi.

Sinyal ambient membawa sedikit informasi: pasangkan dengan saluran presisi untuk saat bertindak.

Ambient berkata “perhatikan”; layar, suara, atau pergelangan berkata “maju sekarang”. Buku: Subbab 15.3.4.

Kenyamanan, Keselamatan, Privasi, Aksesibilitas

Kenyamanan: *cybersickness* ketika mata dan keseimbangan tak sepakat — dunia yang stabil, sesi singkat, tidak pernah wajib.

Keselamatan: orang yang memakai layar di wajah kurang memperhatikan dunia.

Privasi: kamera merekam orang di sekitar; gambar dan suara adalah data pribadi menurut UU PDP 27/2022.

Aksesibilitas: setiap mode masukan menyingkirkan seseorang; XR Accessibility User Requirements dari W3C meminta alternatif, teks keterangan, dan penyesuaian.

Aturannya: setidaknya dua jalan masuk dan dua jalan keluar, salah satunya bekerja tanpa perangkat baru. Buku: Subbab 15.3.5.

Gagasan Spasial di Layar Datar

Liquid Glass (Juni 2025, iOS 26): kontrol tembus pandang yang membiaskan konten di belakangnya, terilhami visionOS.

Teks di atas kaca tak punya latar tetap: label yang lolos 4,5:1 di atas putih bisa gagal di atas foto.

Uji keterbacaan di atas latar terburuk; hormati pengaturan “kurangi transparansi” dan “naikkan kontras”.

Kedalaman, gerak, dan transparansi dapat menuntun perhatian sekaligus mengorbankan keterbacaan; sediakan versi yang tenang dan pejal sebagai pengaturan. Buku: Subbab 15.3.6.

Daftar Periksa Konteks Pemakaian

Tabel 15.2 Daftar periksa konteks pemakaian yang dijawab untuk ruang tunggu puskesmas.

Pertanyaan	Ruang tunggu puskesmas
Seberapa jauh orang dari informasinya?	Bangku ke dinding: 3–5 m
Apakah tangannya bebas?	Sering tidak: anak, tas, tongkat
Di mana perhatiannya?	Pada pendamping, ponsel, terkantuk
Siapa lagi yang hadir?	Pasien lain; tanpa nama di layar
Perangkat apa yang dimiliki orang?	Ponsel sederhana; sedikit jam tangan; tanpa kacamata
Kemampuan apa yang beragam?	Low vision, gangguan dengar, literasi rendah, banyak bahasa
Apa yang terjadi bila gagal?	Petugas memanggil nomor dengan suara

Tabel 15.2 biasanya menunjukkan bahwa perangkat baru adalah langkah terakhir: nomor yang lebih besar dulu, lalu lampu yang tenang, lalu SMS dan ketukan pergelangan opsional; kacamata paling akhir, kalaupun ada. Buku: Subbab 15.4.

Kalkulator Ukuran Target

`visual_angle.py` / `VisualAngle.java` menerapkan Persamaan (15.1) pada empat jarak: ponsel, kios, panel virtual, televisi dinding.

```
place                   distance    1 deg    2 deg    3 deg
phone in the hand          30 cm   0.52 cm  1.05 cm  1.57 cm
kiosk touchscreen          50 cm   0.87 cm  1.75 cm  2.62 cm
virtual panel (headset)   150 cm   2.62 cm  5.24 cm  7.86 cm
queue TV on the wall      300 cm   5.24 cm 10.47 cm 15.71 cm

A 1.0 cm button seen from each place:
  phone in the hand         1.91 deg
  kiosk touchscreen         1.15 deg
  virtual panel (headset)   0.38 deg
  queue TV on the wall      0.19 deg
```

Tombol 1 cm, cukup dengan 1,91 derajat di tangan, menyusut menjadi 0,19 derajat di seberang ruangan. Buku: Subbab 15.5.

Sudut Apa Pun dari Baris Perintah

```
place                   distance  0.5 deg  1.5 deg
phone in the hand          30 cm   0.26 cm  0.79 cm
kiosk touchscreen          50 cm   0.44 cm  1.31 cm
virtual panel (headset)   150 cm   1.31 cm  3.93 cm
queue TV on the wall      300 cm   2.62 cm  7.85 cm
```

`python visual_angle.py 0.5 1.5`: pada 3 m nomor 1,5 derajat memerlukan 7,85 cm; pada 5 m, 13,1 cm, sama dengan hitungan tangan.

Fungsi bantu `g` di Java mencetak 1.0 sebagai “1”, seperti `%g` Python, sehingga judul kolomnya sama. Buku: Subbab 15.5.

Memilih Waktu Tatap: Datanya

Kios yang dikendalikan mata bagi pasien yang tak dapat memakai tangannya. Berapa lama pandangan harus bertahan?

`make_gaze.py`: 1.000 lirikan simulasi; kira-kira satu dari lima disengaja (median kesabaran 1.200 ms), sisanya lirikan membaca (median 260 ms, dekat fiksasi 225–250 ms saat membaca dalam hati).

$$J(T) = 3\,F(T) + M(T) \tag{15.2}$$

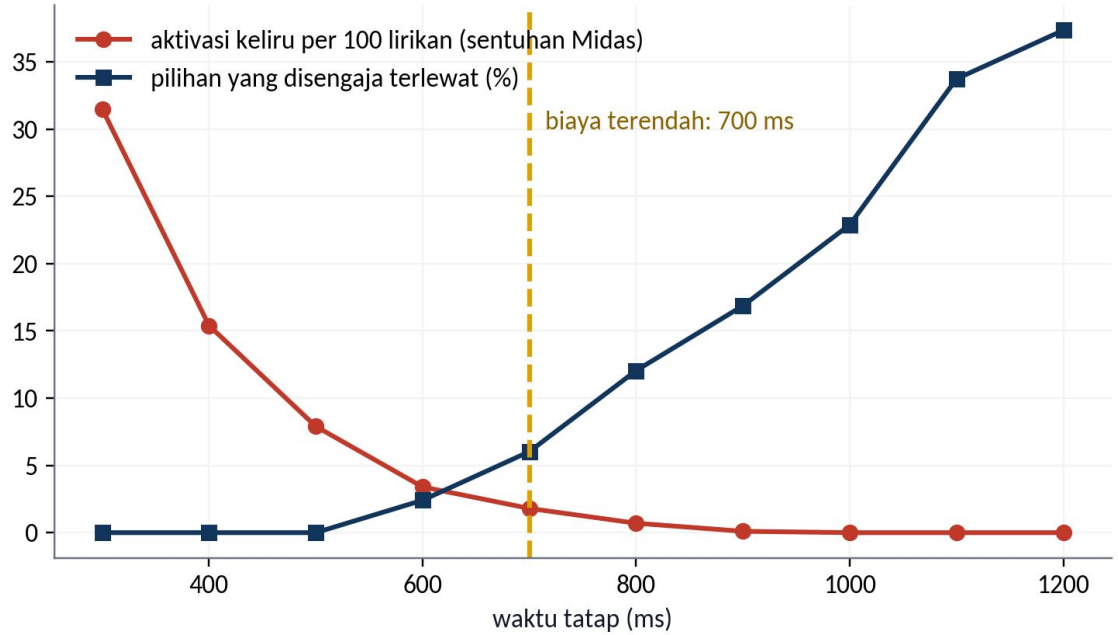
Pada Persamaan (15.2), F adalah aktivasi keliru per 100 lirikan dan M persentase pilihan disengaja yang terlewat; satu aktivasi keliru dihitung setara tiga pilihan terlewat — keputusan desain, bukan hukum. Buku: Subbab 15.6.1 dan 15.6.2.

Hasilnya

```
glances: 1000, of which intended selections: 166
dwell (ms)  false activations /100  intended missed
       300                    31.5             0.0%
       400                    15.4             0.0%
       500                     7.9             0.0%
       600                     3.4             2.4%
       700                     1.8             6.0%
       800                     0.7            12.0%
       900                     0.1            16.9%
      1000                     0.0            22.9%
      1100                     0.0            33.7%
      1200                     0.0            37.3%
lowest cost (3 x false + missed) at dwell = 700 ms
```

Pada 300 ms hampir sepertiga dari semua lirikan akan memicu tombol. Biaya terendah ada pada 700 ms. Buku: Subbab 15.6.3.

Tawar-Menawarnya dalam Gambar



Gambar 15.2 Aktivasi keliru turun dan pilihan terlewat naik seiring bertambahnya waktu tatap.

Gambar 15.2: tak satu pun galat lenyap pada waktu tatap berapa pun. Satu-satunya jalan keluar adalah sinyal kedua — karena itu pandang + cubit, dan langkah konfirmasi pada kios pandangan umum. Buku: Subbab 15.6.3 dan Gambar 15.2.

Menguji Sifat, Bukan Hanya Angka

```
ok   round trip
ok   1 degree at 3 m is about 5.24 cm
ok   longer dwell, fewer false activations
ok   longer dwell, more missed selections
all checks passed
```

Uji waktu tatap memeriksa sifat yang tetap benar ketika log simulasi diganti data nyata. Buku: Subbab 15.7.

Membuatnya Lebih Baik

Konfirmasikan, jangan hanya menatap: pilih pada 700 ms, pastikan dengan YA tersendiri.

Sesuaikan waktu tatap dengan orangnya; tampilkan cincin yang terisi selama menatap.

Ukur dari kasus terburuk yang wajar: kursi terjauh, penglihatan pasien, galat pelacak yang terukur.

Rancang saluran ambient dan presisi bersamaan; lampu yang tak sepakat dengan layar mengajari orang mengabaikannya.

Buku: Subbab 15.8.

Memakainya Sungguhan

Bangku terjauh 6 m → `python visual_angle.py 1.5` → nomor 15,7 cm, tebal dan berkontras tinggi.

Strip berwarna di atas setiap pintu: abu-abu → kuning tua saat tersisa lima nomor → hijau toska saat dipanggil.

SMS opsional ketika tiga nomor lagi, setelah persetujuan tersendiri; jam tangan menampilkannya di pergelangan.

Urutan biaya = urutan jangkauan: pengaturan huruf, lampu, SMS, kios pandangan.

Metode yang sama mengukur papan keberangkatan terminal bus, lampu peringatan pabrik, dan papan harga kios pasar. Buku: Subbab 15.9.

IMK dalam Praktik: Profitina

Lewat kacamata audio atau asisten suara, orang mendengar **satu** jawaban lisan, barangkali menyebut satu sumber.

Menjadi sumber yang dikutip makin penting ketika tak ada halaman tautan — argumen bagi Generative Engine Optimization.

Antarmuka Profitina sendiri: penghitung “N baru” berbasis SSE sebagai sinyal pinggiran yang tenang; notifikasi dihimpun di satu kokpit; AI berjalan lokal.

Buku: Subbab 15.10.

Kesalahan yang Sering Terjadi

Ukuran tanpa jarak; merancang dari kursi terdekat; membiarkan pandangan mengaktifkan.

Terlalu banyak pola getaran; sinyal ambient dipaksa membawa informasi presisi.

Mewajibkan perangkat baru; melupakan orang di sekitar; mengabaikan kenyamanan.

Setiap kesalahan di buku disertai cara memperbaikinya. Buku: Subbab 15.11.

Ringkasan: Lima Hal untuk Diingat

1. Antarmuka spasial, *wearable*, dan ambient harus mudah ditangkap sekilas dan sulit terpicu tanpa sengaja.

2. Ukuran adalah sudut pandang; ubah dengan Persamaan (15.1) untuk jarak realistis terjauh.

3. Memandang bukan memilih: tambahkan sinyal kedua, atau pilih waktu tatap dengan tawar-menawar Gambar 15.2.

4. Ambient berkata “perhatikan”; saluran presisi berkata “maju sekarang”.

5. Jawab Tabel 15.2 lebih dulu: perbaikan termurah bagi semua orang datang sebelum perangkat baru.

Buku: Subbab 15.12.

Pertemuan Berikutnya

Bab 16 — Proyek Akhir. Seluruh buku pada satu masalah: kios antrean untuk puskesmas Mbah Sumi, diriset, dibangun, diuji, dan diperbaiki.

Sebelum itu: kerjakan Soal 15.1, 15.3, 15.4, dan 15.6.

Buku: Subbab 15.13 dan 15.14.

Kuliah 16 — Proyek Akhir: Merancang dan Mengevaluasi Sistem Interaktif Utuh

Mata kuliah: Interaksi Manusia dan Komputer (IMK)

Buku: Irfan Subakti, *Interaksi Manusia dan Komputer*, Bab 16

Dosen: Irfan Subakti — yifana@gmail.com

Hari ini seluruh buku bertemu satu masalah nyata: kios antrean untuk puskesmas, diriset, dirumuskan, dibangun dalam Python dan Java, diuji bersama pengguna termasuk lansia, dan diperbaiki berdasarkan bukti. Buku: Bab 16, Subbab 16.1–16.14.

Yang Dibahas Pertemuan Ini

1. **Penugasannya** — Puskesmas Sehat Selalu dan aturan “tidak boleh ada yang lebih dirugikan daripada hari ini”.
2. **Jalurnya** — empat fase, dan setiap bab dari 2 sampai 15 sebagai sebuah keputusan.
3. **Menemukan dan merumuskan** — riset di loket, persona, kebutuhan, anggaran faktor manusia, aksesibilitas dan hukum.
4. **Mengembangkan** — prototipe kertas, keputusan desain, kios yang berjalan.
5. **Menyerahkan dan iterasi** — uji rintis, SUS, UMUX-Lite, penyelesaian, kontras; temuan, keparahan, uji ulang; menyajikan keputusan.

Setiap tangkapan layar berasal dari kios kita sendiri yang sedang berjalan, dan setiap angka adalah keluaran program yang nyata. Buku: Bab 16.

Penugasannya

Sekitar dua ratus pasien pada pagi yang sibuk; satu loket pendaftaran; antrean sampai ke luar pintu; empat layanan: umum, KIA, gigi, lansia.

Kepala puskesmas menginginkan kios layar sentuh bagi pasien yang tahu layanannya. Tim tiga mahasiswa, satu tablet sumbangan, satu semester.

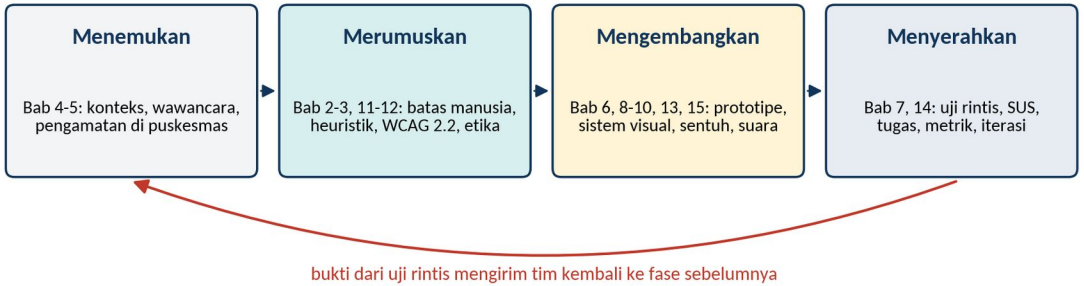
Satu aturan: **tidak boleh ada yang lebih dirugikan daripada hari ini.**

Kios yang cepat bagi anak muda tetapi membingungkan Mbah Sumi akan memperpendek antrean sekaligus membuat puskesmas lebih buruk bagi yang paling membutuhkan. Buku: Subbab 16.1.

Sistem Interaktif yang Utuh

Lebih dari layar: tempatnya, kertas nomor yang digantikannya, loket yang diringankannya, perawat, pasien yang tak mau memakainya, dan hukumnya.

ISO 9241-210: pahami konteks, rinci kebutuhan, hasilkan desain, evaluasi — lalu ulangi.



Gambar 16.1 Jalur proyek akhir dalam empat fase, dengan bab yang diterapkan setiap fase dan putaran baliknya.

Gambar 16.1 memakai Double Diamond dari Design Council. Panah di bawah paling penting: bukti mengirim tim kembali. Buku: Subbab 16.2.1.

Setiap Bab sebagai Keputusan

Tabel 16.1 Bab-bab terpilih dan keputusan yang disumbangkan masing-masing bagi kios antrean.

Bab	Keputusannya bagi kios
2	Empat pilihan; target besar dekat tangan; tak ada yang perlu diingat
3	Status terlihat; kata-kata loket sendiri; pencegahan galat
5	Dua pagi pengamatan; delapan wawancara
6	Prototipe kertas, lalu HTML yang berjalan
7	Uji rintis dengan delapan orang: tugas, SUS, UMUX-Lite
11	Kontras, ukuran target, tautan lompat, bahasa, pesan status, huruf besar
12	Tanpa data pribadi; tanpa tekanan; bantuan darurat dari manusia
13	**Tanpa AI**: tak ada yang perlu ditebak, dan tebakan bisa mencelakakan
14	Metrik HEART; hanya hitungan anonim
15	Ukuran dari jarak pandang

Tabel 16.1 memuat sepuluh dari empat belas baris; buku menambahkan Bab 4, 8, 9, dan 10. Memilih untuk tidak memakai suatu teknologi juga keputusan desain. Buku: Subbab 16.2.2 dan Tabel 16.1.

Menemukan: Riset di Loket

Temuan 1: sekitar dua pertiga adalah pasien lama yang tahu layanannya dan mengantre hanya untuk kertas nomor.

Temuan 2: sebagian pasien lansia “tidak berani” menyentuh layar umum.

Temuan 3: petugas mendahulukan pasien yang sakit berat; kios bisa menyembunyikan mereka.

Temuan 4: bahasa Indonesia, Jawa, dan Madura; permintaan “bahasa Indonesia yang sederhana, bukan bahasa kantor”.

Pengamatan dengan izin, tanpa merekam nama atau wajah; enam pasien dan dua petugas diwawancarai. Buku: Subbab 16.3.

Persona dan Skenario Kunci

Bu Lestari, 34: KIA setiap bulan, bayi di satu lengan, ingin nomor dalam sepuluh detik.

Mbah Sumi, 72: sendirian setiap Selasa kedua, hanya membaca huruf besar, belum pernah memakai kios, tak akan memakai kios yang membuatnya merasa lamban.

Kalimat skenario yang menjadi kebutuhan 1: *ia harus dapat mengambil nomor dengan satu sentuhan, atau mengabaikan kios sama sekali dan pergi ke loket seperti biasa.*

Rancang bagi persona yang paling tidak beruntung; yang lain ikut diuntungkan. Buku: Subbab 16.3.

Merumuskan: Kebutuhan yang Dapat Diuji

1. Satu sentuhan bagi pasien lama. **2.** Tanpa data pribadi sama sekali.

3. Gejala gawat → langsung ke manusia. **4.** Bahasa Indonesia dan Inggris, kata-kata loket, huruf besar.

5. Tak seorang pun dipaksa: loket tetap buka, kertas dan nomor kios setara. **6.** Nomor dan orang di depan dengan huruf besar.

Setiap kebutuhan ditelusuri ke sebuah temuan dan sebuah bab. Buku: Subbab 16.4.1.

Anggaran Faktor Manusia

$$T_{\text{pilih}} = a + b \log_2(n+1) \tag{16.1}$$

$$I\,D = \log_2\left(\frac{D}{W} + 1\right) \tag{16.2}$$

Hick (Persamaan (16.1)): empat layanan $= \log_2 5 = 2.32$ bit; delapan menjadi $\log_2 9 = 3.17$ bit.

Fitts (Persamaan (16.2)): tombol 64 piksel pada jarak hingga 300 piksel memberi 2.51 bit; tautan 32 piksel memberi 3.37 bit.

Empat pilihan dan tombol tinggi selebar layar: murah bagi tim, baik bagi semua, terbaik bagi tangan yang gemetar. Buku: Subbab 16.4.2.

Aksesibilitas sebagai Kebutuhan

Tabel 16.2 Kriteria sukses WCAG 2.2 terpilih dan cara kios memenuhinya.

Kriteria WCAG 2.2	Cara kios memenuhinya
1.4.3 Contrast (Minimum)	Pasangan teks terendah 7,56:1
1.4.11 Non-text Contrast	Cincin fokus diubah dari 2,27:1 ke 12,46:1
2.4.1 Bypass Blocks	Tautan lompat, tampak saat difokus
2.5.8 Target Size (Minimum)	Tombol setinggi 64 px, selebar layar
3.1.2 Language of Parts	`lang` pada tautan ke bahasa lain
4.1.3 Status Messages	Nomor dalam wilayah `role="status"`

Tabel 16.2: WCAG 2.2 menjadi ISO/IEC 40500:2025 pada Oktober 2025. Buku: Subbab 16.4.3 dan Tabel 16.2.

Hukum dan Etika sebagai Batasan

UU 8/2016 dan **PP 42/2020**: pelayanan publik yang aksesibel adalah hak; puskesmas adalah pelayanan publik.

European Accessibility Act (sejak 28 Juni 2025) mencakup terminal layan mandiri — tanda ke mana dunia bergerak.

UU PDP 27/2022 dan PP 33/2026: kios tidak mengumpulkan data pribadi apa pun, jadi tak perlu layar persetujuan.

Bab 12: tanpa penghitung mundur, tanpa iklan, tanpa tekanan; bantuan darurat mengarah ke manusia, bukan diagnosis.

Cara termurah mematuhi undang-undang pelindungan data adalah tidak mengumpulkan apa pun yang dilindunginya. Buku: Subbab 16.4.3.

Mengembangkan: Prototipe Kertas

Sketsa 1 dibuka dengan layar sambutan dan “Mulai”: keempat pasien menyentuh logonya; seorang bertanya “rusak, ya?”.

Sketsa 2 menaruh keempat layanan lebih dulu: keempatnya memilih dengan benar tanpa ragu.

Mbah Sumi: tulisannya “terlalu kecil untuk mata tua” — mode huruf besar bermula di sini.

Prototipe sepuluh menit menghapus satu layar utuh dari desain. Buku: Subbab 16.5.1.

Kios yang Berjalan



Gambar 16.2 Kios yang berjalan: 1 bahasa dan ukuran huruf, 2 catatan bantuan darurat, 3 keempat layanan, 4 nomor dan jumlah orang di depan.

Gambar 16.2 ditangkap dengan Playwright dari server kita sendiri. Yang sengaja tidak ada: perkiraan waktu tunggu (tebakan) dan JavaScript (jalan untuk rusak). Buku: Subbab 16.5.2.

Membangunnya dalam Python dan Java

`kiosk_app.py` (Flask, porta 4160): `TEXT` memuat kedua bahasa; `page` menambahkan `lang`, tautan lompat, alat, dan gaya; rute `/`, `/take`, `/status.json`.

`KioskServer.java` (`com.sun.net.httpserver`, porta 4161) menyajikan **HTML yang identik bita demi bita**, diperiksa dengan `curl`.

Pelajaran dari Java: `Map.of` menolak kunci `null` — versi pertama mogok tanpa `lang`. Kini: nilai bawaan, dan penangan yang mencatat dan menjawab alih-alih macet.

Kios tidak boleh macet diam-diam: pasien di depan layar yang membeku tidak tahu harus menunggu atau pergi. Buku: Subbab 16.6.1 dan 16.6.2.

Mengendalikannya dari Terminal

```
$ curl -s -X POST -d "lang=en&service=L" http://127.0.0.1:4160/take
$ curl -s -X POST -d "lang=en&service=L" http://127.0.0.1:4160/take
```

```
L-001
ahead of you: 0
L-002
ahead of you: 1
```

Baris-baris kunci dari dua halaman yang dikembalikan; `/status.json` lalu melaporkan dua nomor diberikan untuk L. Buku: Subbab 16.6.3.

Uji Rintis

Delapan partisipan: tiga 18–39, dua 40–59, tiga 60+. Cukup untuk menemukan masalah serius (Nielsen dan Landauer), jauh terlalu sedikit untuk rerata yang presisi.

T1 ambil nomor poli umum. **T2** ubah ke bahasa Inggris dan ambil nomor KIA. **T3** perbesar huruf dan ambil nomor poli lansia.

Lalu sepuluh pernyataan SUS dan dua pernyataan UMUX-Lite; data di `sus.csv`, `umux_lite.csv`, `tasks.csv` (set data latihan).

Catatan berpikir keras menjelaskan angka-angkanya. Buku: Subbab 16.7.1.

Ukuran-Ukurannya

$$\text{SUS} = 2.5\left[\sum_{i\in[1,3,5,7,9]}(a_i - 1) + \sum_{i\in[2,4,6,8,10]}(5 - a_i)\right] \tag{16.3}$$

$$\hat{p} = \frac{x + z^2/2}{n + z^2},\ \hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n + z^2}} \tag{16.4}$$

Persamaan (16.3): 0 sampai 100, bukan persentase; sekitar 68 adalah rerata. UMUX-Lite: dua butir, setara SUS $= 0.65 \cdot \text{UMUX-Lite} + 22.9$.

Persamaan (16.4): selang Wald yang disesuaikan tetap masuk akal pada 8 dari 8. Waktu: rerata geometrik.

Sauro dan Lewis menganjurkan selang Wald yang disesuaikan untuk sampel kecil. Buku: Subbab 16.7.2.

Kontras Warna, Dihitung

$$L = 0.2126\,R_{\text{lin}} + 0.7152\,G_{\text{lin}} + 0.0722\,B_{\text{lin}}, \text{rasio} = \frac{L_1 + 0.05}{L_2 + 0.05} \tag{16.5}$$

Setiap kanal sRGB dilinearkan lebih dulu; Persamaan (16.5) lalu memberi rasio warna yang lebih terang terhadap yang lebih gelap.

Teks memerlukan 4,5:1; cincin fokus dan elemen nonteks lain 3:1.

`evaluate.py` dan `Evaluate.java` *memeriksa setiap pasangan warna di lembar gaya, termasuk cincin fokus versi draf. Buku: Subbab 16.7.2 dan 16.7.3.*

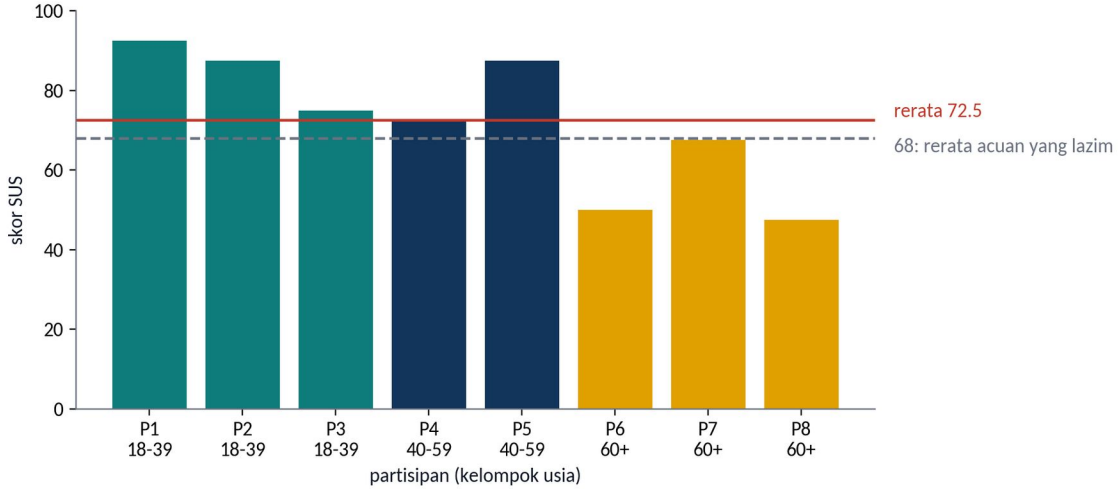
Hasilnya

```
SUS per participant
  P1  18-39   92.5
  P2  18-39   87.5
  P3  18-39   75.0
  P4  40-59   72.5
  P5  40-59   87.5
  P6  60+     50.0
  P7  60+     67.5
  P8  60+     47.5
SUS mean 72.5 (SD 17.0), 95% CI [58.3, 86.7]
UMUX-Lite mean 79.2, SUS-comparable 74.4

task  done  completion  95% CI (adj. Wald)  geo. mean time
T1    8/8        100%  [ 63%, 100%]            12.7 s
T2    6/8         75%  [ 40%,  94%]            26.0 s
T3    7/8         88%  [ 51%, 100%]            22.8 s
```

Bagian kontras dari keluaran menunjukkan semua pasangan lolos kecuali cincin fokus versi draf (2,27:1), yang sudah diganti biru tua (12,46:1). Buku: Subbab 16.7.4.

Membaca Hasilnya



Gambar 16.3 SUS per partisipan dan kelompok usia; ketiga partisipan berusia 60+ di bawah 68.

Gambar 16.3: rerata 72,5 tampak baik, selang [58.3, 86.7] mengatakan delapan orang tak dapat memastikan, dan polanya mengatakan kios belum cukup baik bagi lansia. T2 gagal dua kali ("tidak melihat" tautan bahasa); T3 sekali ("tidak terjadi apa-apa"). Buku: Subbab 16.7.4.

Iterasi: dari Temuan ke Perubahan

Tabel 16.3 Temuan uji rintis dengan penilaian keparahan Nielsen dan perubahan yang direncanakan.

Temuan	Keparahan	Perubahan untuk iterasi 2
Tautan bahasa tak terlihat oleh lansia	3	Dua tombol bahasa besar di atas layanan
Perubahan huruf besar tak disadari	2	Huruf besar dengan sekali sentuh; baris konfirmasi
SUS lansia di bawah 68	3	Uji ulang dengan setidaknya lima lansia
Cincin fokus 2,27:1	2	Sudah diperbaiki: cincin biru tua, 12,46:1
Takut menyentuh kios	2	Pendamping di dekat kios bulan pertama; “Loket tetap buka seperti biasa”

Tabel 16.3 mengikat setiap perubahan pada bukti dan diakhiri uji ulang yang dinilai dari T2 dan T3 bagi lansia, bukan dari rerata keseluruhan. Buku: Subbab 16.8 dan Tabel 16.4.

Menguji Kode

```
ok   best answers give SUS 100
ok   neutral answers give SUS 50
ok   black on white is 21:1
ok   8 of 8 still has a lower bound
all checks passed
```

Server: HTML identik dari keduanya; kode layanan tak sah dialihkan (302 Flask, 303 Java); Playwright gagal bila tombol yang diharapkan tidak ada.

Buku: Subbab 16.9.

Menyajikan Keputusan Desain

1. Masalah dan orang-orangnya. **2.** Kebutuhan yang ditelusuri ke temuan dan bab.

3. Desainnya, setiap keputusan terikat pada kebutuhan. **4.** Buktinya, termasuk hasil yang mengecewakan.

5. Apa yang berubah berikutnya, dan cara menilainya. **6.** Apa yang sengaja *tidak* dilakukan, dan mengapa.

Butir 6 sering paling meyakinkan pendengar yang skeptis: ia menunjukkan pilihan yang sedang tren telah ditimbang dan ditolak dengan alasan yang terikat pada pengguna. Buku: Subbab 16.10.

Memakainya Sungguhan

Peramban dalam mode kios; dipasang setinggi jangkauan kursi roda; printer termal untuk nomor kertas.

Hitungan di basis data kecil, diatur ulang setiap pagi; endpoint `/call` untuk loket (Soal 16.6).

Endpoint status mengisi televisi ruang tunggu, diukur seperti di Bab 15; server produksi dengan jalan ulang otomatis.

Kantor desa, kantor cabang bank, kantor pos, dan kantor Samsat memakai proses yang sama — dan aturan yang sama. Buku: Subbab 16.11.

IMK dalam Praktik: Profitina

Pertanyaan riset: bagaimana orang yang bukan insinyur memahami mengapa mesin jawaban AI mengutip, atau mengabaikan, sebuah situs? Jawabannya: setiap sinyal dijelaskan dengan kata sederhana.

Kepatuhan UU PDP swalayan: persetujuan, ekspor, ganti surel dan kata sandi, penghapusan penuh — di dalam produk.

Evaluasi berkelanjutan: jauh lebih dari seribu uji otomatis menjadi gerbang setiap rilis; rilis bertahap yang diverifikasi kesehatannya; alur umpan balik yang terlihat.

Siklus yang sama dengan kios, pada skala yang lebih besar. Buku: Subbab 16.12.

Kesalahan yang Sering Terjadi

Memulai dari solusi; merancang untuk pengguna rata-rata; menganggap uji kecil itu presisi.

Memeriksa aksesibilitas di akhir; menambah teknologi karena tersedia; mengumpulkan data yang tak diperlukan.

Berhenti setelah satu iterasi; hanya menyajikan kabar baik.

Setiap kesalahan di buku disertai cara memperbaikinya. Buku: Subbab 16.13.

Ringkasan: Lima Hal untuk Diingat

1. Sistem yang utuh mencakup layanan di sekitar layar; jalankan siklus Gambar 16.1 dan kembali lagi.

2. Setiap metode dipakai di tempat yang layak — atau sengaja ditinggalkan (Tabel 16.1).

3. Hukum faktor manusia (Persamaan (16.1), Persamaan (16.2)) dan WCAG 2.2 (Tabel 16.2) menjadi kebutuhan sejak hari pertama.

4. Ukur dengan SUS (Persamaan (16.3)), selang (Persamaan (16.4)), dan kontras (Persamaan (16.5)); laporkan menurut kelompok yang penting.

5. Ubah temuan menjadi perubahan berprioritas yang terikat bukti, beserta uji ulang (Tabel 16.3).

Buku: Subbab 16.14.

Setelah Mata Kuliah Ini

Pilih tempat nyata di dekat Anda tempat orang kesulitan dengan sebuah antarmuka: warung, puskesmas, kantor desa, laundry.

Jalankan seluruh siklus sekali: amati, rumuskan, buat prototipe, bangun, uji dengan lima orang, perbaiki, uji lagi.

Kerjakan Soal 16.1–16.6; semuanya melatih setiap perhitungan dan keputusan hari ini.

Aturan Puskesmas Sehat Selalu berlaku di mana saja: tidak boleh ada yang lebih dirugikan daripada sebelumnya. Buku: Subbab 16.15 dan 16.16.