PERTEMUAN 4

Desain UI/UX Konsisten Lintas Platform

Rekayasa Lintas Platform (S1 Teknik Informatika)

Membangun Ekosistem Digital yang Seamless

Pendahuluan & Konteks

Memahami tantangan dan urgensi konsistensi antarmuka pengguna dalam ekosistem perangkat yang beragam.

Learning Objectives

- **Menjelaskan & Merancang Design System:** Mampu mendokumentasikan dan menerapkan komponen standar untuk menjamin konsistensi antarmuka di berbagai platform.
- **Responsivitas & Adaptabilitas:** Mampu membedakan serta menerapkan prinsip tata letak dinamis dan penyesuaian antarmuka berdasarkan metode input pengguna.
- **Standar Aksesibilitas (A11y):** Mengimplementasikan prinsip desain inklusif agar sistem dapat digunakan oleh semua demografi, termasuk difabel.
- **Evaluasi Heuristik:** Mampu melakukan inspeksi kegunaan (usability) sistem menggunakan metode evaluasi pakar berdasarkan prinsip desain yang teruji.

Real-World Context

"Dalam lingkungan multi-device masa kini, pengguna sering berpindah platform (web desktop ke aplikasi mobile). Jika antarmuka terasa sangat berbeda, pengguna akan mengalami kebingungan (cognitive load tinggi) dan harus belajar ulang."

— Urgensi Konsistensi UI/UX

Key Terms (Terminologi Kunci)



Design System

"Bahasa tunggal" berupa dokumentasi dan pustaka komponen standar yang menghubungkan visi tim desain dengan implementasi tim pengembang.



Design Tokens

Variabel teknis (JSON/XML) penyimpan atribut desain dasar seperti warna, jarak, dan tipografi untuk menjamin konsistensi lintas platform (Web, iOS, Android).



Mental Model

Ekspektasi atau pola pikir bawaan pengguna tentang bagaimana sebuah sistem atau elemen digital seharusnya bekerja berdasarkan pengalaman mereka.

Bagian 1: Design System

Sumber Kebenaran Tunggal (Single Source of Truth) dalam Rekayasa Perangkat Lunak

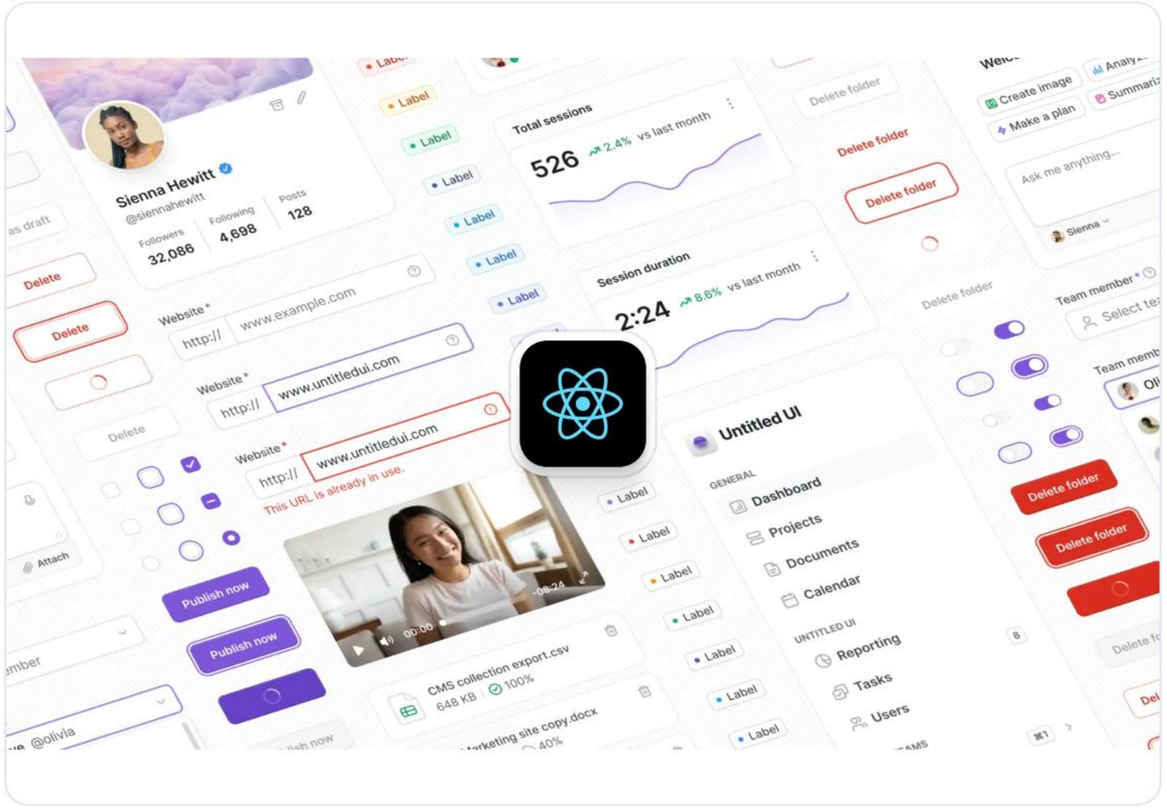
Apa itu Design System?

Bukan Sekadar UI Kit: Design System lebih dari sekadar file desain di Figma. Ia adalah ekosistem panduan, prinsip, dan kode yang hidup.

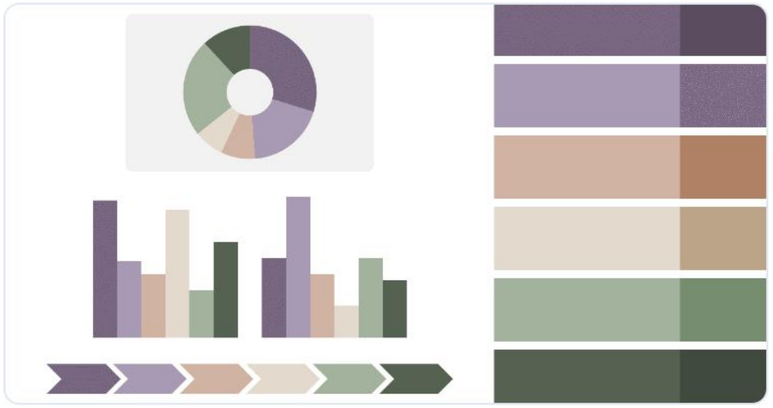
Sumber Kebenaran Tunggal: Bertindak sebagai panduan final bagi Designer dan Developer untuk mencegah inkonsistensi visual.

Komponen Reusable: Kumpulan elemen UI (tombol, navigasi, form) yang sudah disepakati fungsi, perilaku, dan tampilannya.

Efisiensi Skalabilitas: Memungkinkan tim untuk merakit halaman baru (web atau mobile) layaknya menyusun balok LEGO, menghemat waktu *development*.

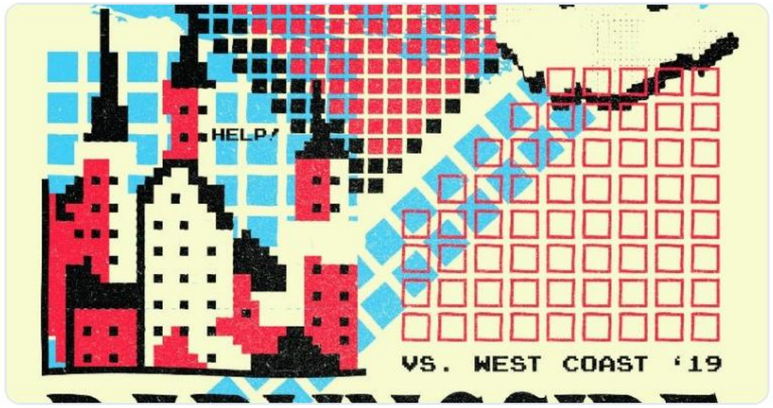


Komponen Visual Language



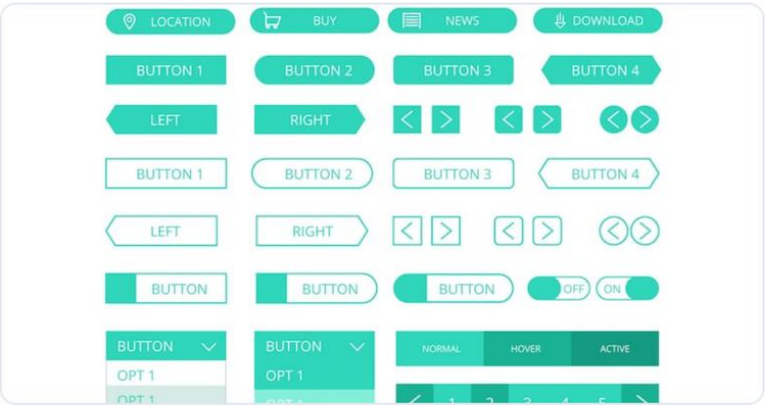
Palet Warna

Mendefinisikan warna utama (primary), sekunder, sukses, error, dan background untuk identitas produk.



Hierarki Tipografi

Menetapkan jenis font, ukuran (Heading 1 hingga Paragraph), dan ketebalan untuk keterbacaan optimal.



Koleksi Komponen

Bentuk baku dari tombol, input teks, *dropdown*, dan kartu yang memiliki *state* (hover, active, disabled).

Mengapa Membutuhkan Design System?

1

Single Source of Truth

Menghilangkan Redundansi & Kesalahan

Tanpa sistem desain, developer web dan mobile seringkali mengimplementasikan komponen yang sama dengan cara yang berbeda. Hal ini menyebabkan "utang teknis" (technical debt) dan ketidakkonsistenan antarmuka yang membingungkan pengguna lintas platform.

Design system memaksa pembaruan terpusat: jika warna *brand* berubah, cukup ubah satu variabel, dan seluruh platform akan menyesuaikan saat di-kompilasi ulang.

Visual Language vs Design Tokens

Visual Language (Ranah Desain)

Definisi konseptual tentang bagaimana merek terlihat dan terasa. Dibuat oleh desainer menggunakan *tools* seperti Figma atau Sketch.

- "Tombol utama kita menggunakan warna biru korporat."
- "Jarak antar elemen adalah 16 pixel."
- Fokus pada estetika dan konsistensi visual di atas kanvas desain.

Design Tokens (Ranah Teknis)

Penerjemahan nilai desain konseptual ke dalam format data terstruktur (variabel) yang dipahami oleh mesin dan berbagai platform (CSS, XML, Swift).

- `$color-brand-primary = #0056b3;`
- `$spacing-md = 16px;`
- Fokus pada skalabilitas, *maintenance* kode, dan *cross-platform deployment*.

Design Tokens dalam Praktik

Design tokens mengekstraksi nilai desain keras (hardcoded) menjadi variabel semantik. File JSON bertindak sebagai distributor nilai ke berbagai *platform*.

Desainer: Mengubah warna di Figma.

Ekspor: Diubah menjadi file JSON/YAML.

Web: JSON dikonversi menjadi CSS Variables (`var(--color-primary)`).

Android: JSON dikonversi menjadi XML resources (`@color/primary`).

iOS: JSON dikonversi menjadi Swift constants.



Anatomi Komponen Reusable

- **Enkapsulasi Logika:** Komponen UI (misal:) tidak hanya menyimpan gaya visual, tetapi juga logika dasar seperti *loading state* atau aksi ketika di-klik.
- **Properti (Props) yang Fleksibel:** Komponen dirancang menerima parameter agar bisa digunakan di berbagai konteks tanpa mengubah kode dasar komponen itu sendiri.
- **Sentralisasi Pembaruan:** Jika terdapat perbaikan *bug* atau peningkatan aksesibilitas pada tombol, pengembang cukup memperbaiki komponen sumber, dan seluruh halaman yang memakainya otomatis terbarui.
- **Cross-Platform Ready:** Dalam kerangka kerja seperti Flutter atau React Native, komponen *reusable* ini ditulis sekali dan dikompilasi ke format *native* iOS dan Android.

Bagian 2: Familiaritas Antarmuka

Mempertahankan Mental Model Pengguna Lintas Perangkat

Menjaga Mental Model Pengguna



Pola Navigasi

Jika fitur "Pengaturan" berbentuk ikon gerigi di pojok kanan atas versi Web, usahakan letak dan ikon yang ekuivalen di aplikasi Mobile untuk mempermudah pencarian.



Kosakata (Microcopy)

Gunakan label dan pesan *error* yang konsisten. Jangan gunakan label "Keranjang" di Web tetapi berubah menjadi "Tas Belanja" di aplikasi Mobile.



Ekspektasi Interaksi

Tombol aksi utama (Primary Button) harus selalu menonjol dengan warna *brand* yang sama, mengindikasikan fungsi "Submit" atau "Lanjut" di semua perangkat.

Dampak Familiaritas Desain



Beban Kognitif (Cognitive Load)

Kurva Pembelajaran yang Singkat

Konsistensi UI/UX secara drastis mengurangi beban kognitif pengguna. Mereka tidak perlu mempelajari cara menggunakan aplikasi dua kali (sekali untuk Web, sekali untuk Mobile).

Familiaritas membangun kepercayaan. Saat sistem berperilaku persis seperti ekspektasi pengguna berdasarkan interaksi sebelumnya, rasio penyelesaian tugas (task completion rate) akan meningkat signifikan.

Konsistensi dalam Praktik Skalabilitas

Familiaritas tidak berarti UI Web harus identik 100% dengan Mobile (pixel-perfect replication). Ini tentang menerjemahkan pengalaman ke batasan perangkat keras yang berbeda secara harmonis.

> Tabel kompleks di Web diubah menjadi daftar kartu (Card List) di Mobile.
>
> Konten tetap sama, fungsi tetap sama, *branding* konsisten.
>
> Pengguna menyadari bahwa ini adalah aplikasi dari "pabrik" yang sama.



Bagian 3: Responsivitas & Adaptabilitas

Merespons Layar Fisik vs Mengadaptasi Metode Interaksi

Konsep Fundamental: Perbedaan Utama

Desain Responsif (Responsive Design)

Pendekatan desain yang bereaksi terhadap ukuran dan orientasi layar. Fokus utama adalah pada **Tata Letak (Layout)** visual.

- Grid cairan (Fluid Grids) dan CSS Media Queries.
- Satu basis kode HTML, namun CSS mengatur elemen bertumpuk di layar kecil (Mobile) dan berjejer di layar besar (Desktop).
- *Analogi:* Air yang menyesuaikan bentuk wadahnya.

Desain Adaptif (Adaptive Design)

Pendekatan yang menyesuaikan fitur dan fungsionalitas UI berdasarkan **Kapabilitas Input & Konteks** perangkat pengakses.

- Mendeteksi *touch screen* vs kursor *mouse*, atau mendeteksi ketersediaan GPS/Kamera.
- Sistem menyajikan elemen UI yang spesifik untuk medium tersebut.
- *Analogi:* Kendaraan yang mengganti roda menjadi pelampung saat masuk air.

Perbandingan Komprehensif

Aspek Evaluasi	Responsif (Fluid Layout)	Adaptabilitas (Input & Context)
Fokus Utama	Ukuran & Resolusi Layar (Dimensi Fisik)	Metode Input, Sensor, Lingkungan Pengguna
Mekanisme Teknis	CSS Flexbox, CSS Grid, Media Queries	JavaScript Feature Detection, Platform APIs
Contoh Implementasi	Gambar mengecil otomatis; Menu berubah menjadi *Hamburger icon*.	Area klik (Touch Target) membesar; Menu tersembunyi merespons gestur *Swipe*.
Kasus Lintas Platform	Menjamin web terbaca di *Smartphone* tanpa harus melakukan *Zoom-in*.	Aplikasi Mobile menggunakan biometrik (Fingerprint), Web menggunakan kata sandi.

Implementasi Tata Letak Cair (Fluid)

Responsivitas sangat mengandalkan sistem grid yang menggunakan nilai persentase (%) alih-alih pixel tetap (px).

Grid 12-Kolom: Standar industri untuk tata letak web modern.

Desktop: Konten membentang menggunakan 4 hingga 8 kolom.

Tablet: Konten menyempit, menggunakan 6 hingga 12 kolom.

Mobile: Konten bertumpuk linear (100% lebar, 1 kolom penuh) demi kemudahan pergerakan jempol (*thumb scrolling*).



Adaptabilitas Input: Mouse vs Touch

Mouse / Kursor (Web)

Memiliki tingkat presisi tinggi hingga skala pixel. Desain dapat mengandalkan elemen UI kecil, *tooltip* penjelasan saat *hover*, dan navigasi dropdown kompleks bertingkat.

Layar Sentuh (Mobile)

Presisi rendah. Elemen UI tidak memiliki status *hover*. Menu navigasi mengandalkan laci samping (*drawer*) atau navigasi bilah bawah (*bottom tab bar*).

Touch Target Standar

Untuk layar sentuh, ukuran minimal tombol (Touch Target) harus **44px hingga 48px** agar tidak terjadi salah klik (*fat-finger error*) sesuai pedoman Apple/Google.

Bagian 4: Aksesibilitas (A11y)

Mendesain Sistem Inklusif untuk Semua Demografi Pengguna

Definisi Inklusivitas Digital

"Aksesibilitas web/sistem berarti orang dengan keterbatasan (disabilitas) dapat mempersepsikan, memahami, menavigasi, dan berinteraksi secara penuh dengan sistem digital Anda."

— World Wide Web Consortium (W3C)

Standar Aksesibilitas Dasar (WCAG)

- **Keterbatasan Visual:** Membutuhkan rasio kontras warna yang memadai untuk kemudahan membaca, dan *tag* semantik ARIA bagi pengguna tunanetra yang mengandalkan alat *Screen Reader*.
- **Keterbatasan Pendengaran:** Sistem yang memuat konten video atau audio harus selalu menyediakan teks tertutup (*Closed Captions* / CC) atau transkrip teks.
- **Keterbatasan Motorik:** Antarmuka (khususnya Web) harus dapat dinavigasikan sepenuhnya menggunakan tombol Tab pada *Keyboard*, tanpa membutuhkan *mouse* sama sekali. Fokus visual (*focus indicator*) saat navigasi keyboard harus terlihat jelas.
- **Keterbatasan Kognitif:** Gunakan bahasa yang sederhana (*plain language*), hindari animasi berkedip berlebihan, dan berikan waktu yang cukup bagi pengguna untuk membaca instruksi *form*.

Elemen Kunci Aksesibilitas UI

Rasio Kontras

Berdasarkan standar WCAG AA, teks reguler dengan latar belakangnya harus memiliki rasio kontras minimal **4.5:1**. (Misal: Jangan gunakan teks abu-abu muda di atas latar belakang putih).

Atribut Alt Text

Setiap gambar yang memiliki nilai informasi (bukan sekadar dekorasi) harus memiliki atribut `alt=""`. Jika tombol hanya berupa ikon, tambahkan `aria-label="Simpan Data"`.

Navigasi Keyboard

Pastikan urutan DOM elemen interaktif (tab order) logis. Pengguna harus dapat berpindah dari input form 1 ke form 2, dan mengeksekusi tombol "Kirim" hanya dengan tombol keyboard.

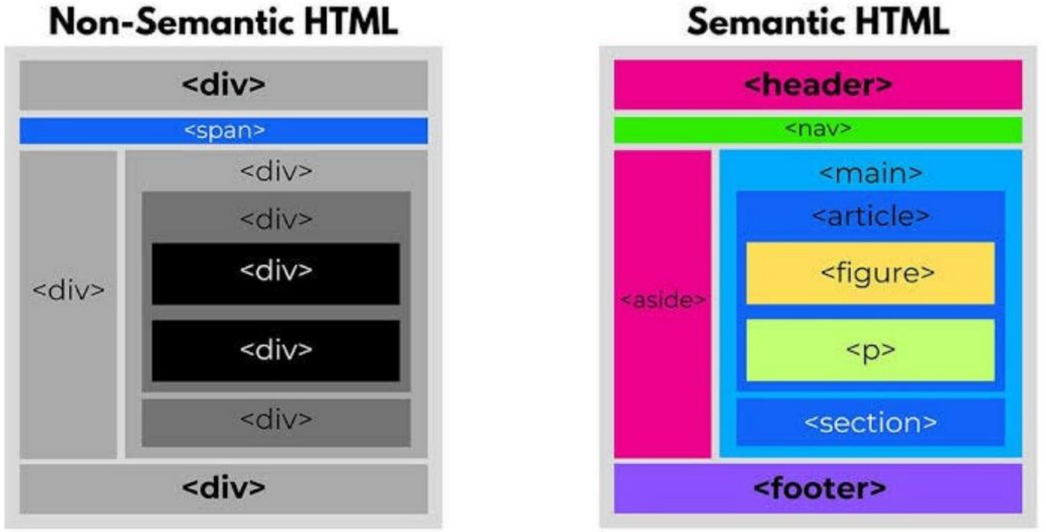
Pentingnya HTML Semantik

Menulis kode web lintas platform bukan sekadar menggunakan elemen secara serampangan. HTML semantik adalah fondasi dari *Screen Reader* (pembaca layar).

Gunakan untuk tombol interaktif, bukan .

Gunakan , , untuk membagi zona halaman.

Struktur semantik memberi konteks pada perangkat disabilitas tentang peran (*role*) sebuah elemen UI di dalam sistem.



Bagian 5: Evaluasi Heuristik

Metode Inspeksi Usability Berbasis Pakar Desain

Apa itu Evaluasi Heuristik?

Metode Inspeksi

Metode evaluasi *usability* di mana sejumlah kecil pakar UX menilai antarmuka sistem dengan membandingkannya terhadap daftar prinsip desain yang telah diakui secara luas (Heuristik).

Tujuan Strategis

Mengidentifikasi masalah pada desain antarmuka sejak dini, sebelum tahap pengembangan massal. Hal ini meminimalkan biaya perbaikan (*re-work*) kode di masa depan.

10 Prinsip Nielsen

Standar industri yang paling sering digunakan adalah **10 Usability Heuristics** yang dirumuskan oleh Jakob Nielsen (Nielsen Norman Group).

Top 4 Heuristik Nielsen untuk Cross-Platform

1. **Visibility of System Status:** Sistem harus selalu memberi tahu pengguna apa yang sedang terjadi (contoh: *loading bar*, notifikasi proses). Jangan biarkan pengguna menebak-nebak kondisi sistem.
2. **Match Between System and Real World:** Sistem berbicara menggunakan bahasa pengguna, dengan kata-kata dan konsep yang familier, bukan menggunakan *jargon* internal teknis sistem (database error codes).
3. **Consistency and Standards:** Pengguna tidak perlu bertanya-tanya apakah kata-kata, situasi, atau tindakan yang berbeda memiliki arti yang sama. Ikuti konvensi *platform* (iOS guidelines, Material Design).
4. **Error Prevention:** Merancang antarmuka yang mencegah terjadinya kesalahan sejak awal (contoh: *input constraint* format email, dialog peringatan sebelum hapus data).

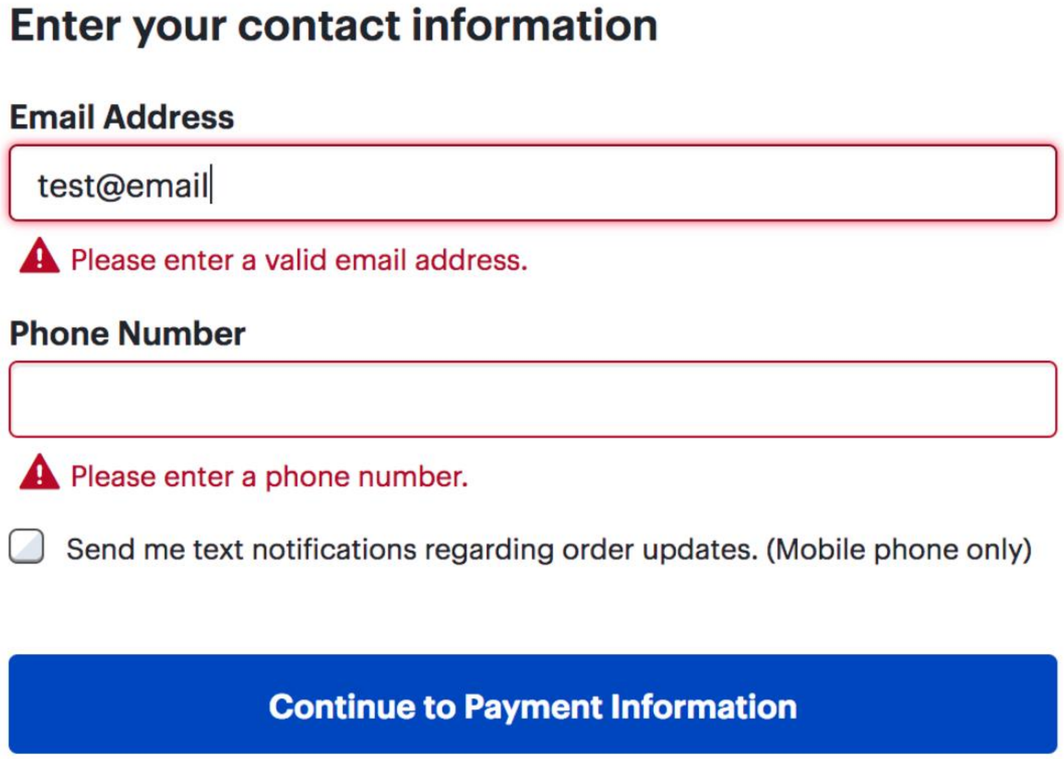
1. Visibility of System Status

Visibilitas sangat penting dalam rekayasa lintas platform (Web, Mobile, IoT) mengingat koneksi jaringan tidak selalu stabil.

Gunakan animasi *Spinner* (Indikator Loading) saat mengambil data dari API backend.

Sediakan *Toast/Snackbar* notifikasi (misal: "Data berhasil disimpan") agar pengguna mendapat kepastian fungsional.

Indikator status ini mengurangi kecemasan pengguna saat menunggu *response* sistem.

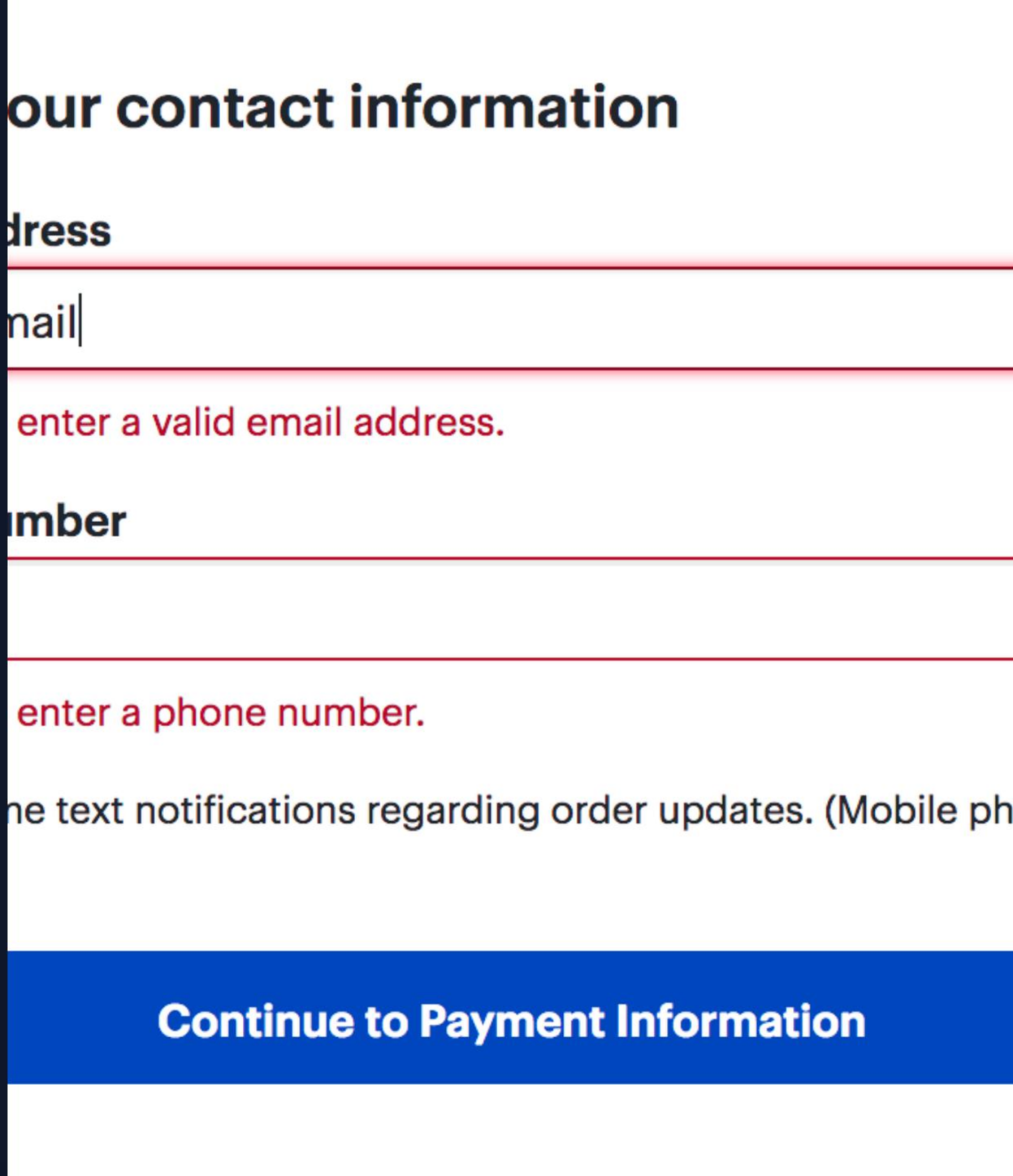


4. Error Prevention

Mencegah kesalahan (*Error Prevention*) selalu lebih baik daripada merancang pesan perbaikan kesalahan yang bagus.

Desain lintas platform yang andal menggunakan mekanisme validasi waktu nyata (real-time validation) dan menonaktifkan tombol kirim (disabled states) sebelum semua kriteria input penting terpenuhi.

- Panduan format kata sandi yang proaktif.
- Kotak dialog peringatan destruktif.



Heuristik vs Usability Testing

Aspek Evaluasi UX	Evaluasi Heuristik (UX Audit)	Usability Testing (User Testing)
Evaluator (Pelaksana)	Pakar UX Profesional (Expert Review)	Pengguna Asli (End-users)
Metodologi Pendekatan	Menganalisis UI berdasarkan standar prinsip (*Best Practices*).	Mengamati pengguna saat menyelesaikan tugas/skenario interaktif.
Biaya & Waktu Fase	Cepat dan relatif murah. Ideal di tahap purwarupa (Prototyping) awal.	Butuh waktu rekrutmen & biaya. Ideal di tahap Beta menuju rilis produk.
Output Temuan Utama	Menemukan pelanggaran struktural UI lintas platform secara menyeluruh.	Menemukan area spesifik yang membingungkan alur pikir (*workflow*) pengguna asli.

Bagian 6: Implementasi Strategis

Siklus Kerja dan Sinkronisasi dalam Rekayasa Perangkat Lunak

Alur Kerja Sistem Lintas Platform

Tahap 1: Visual (Figma)

Desainer membangun konsep visual, warna, dan struktur tipografi sistem.

Tahap 2: Design Tokens

Ekstraksi elemen desain menjadi variabel JSON (Single Source of Truth).

Tahap 3: Reusable UI Code

Developer membangun komponen React/Flutter berbasis tokens tersebut.

Tahap 4: Cross-Platform

Aplikasi Web, iOS, dan Android memiliki UI Konsisten, Responsif & Aksesibel.

Sinkronisasi Backend & Konsistensi Data

UI yang konsisten harus didukung oleh integrasi backend yang solid. Data yang direpresentasikan UI (Web/Mobile) harus tersinkronisasi *real-time*.

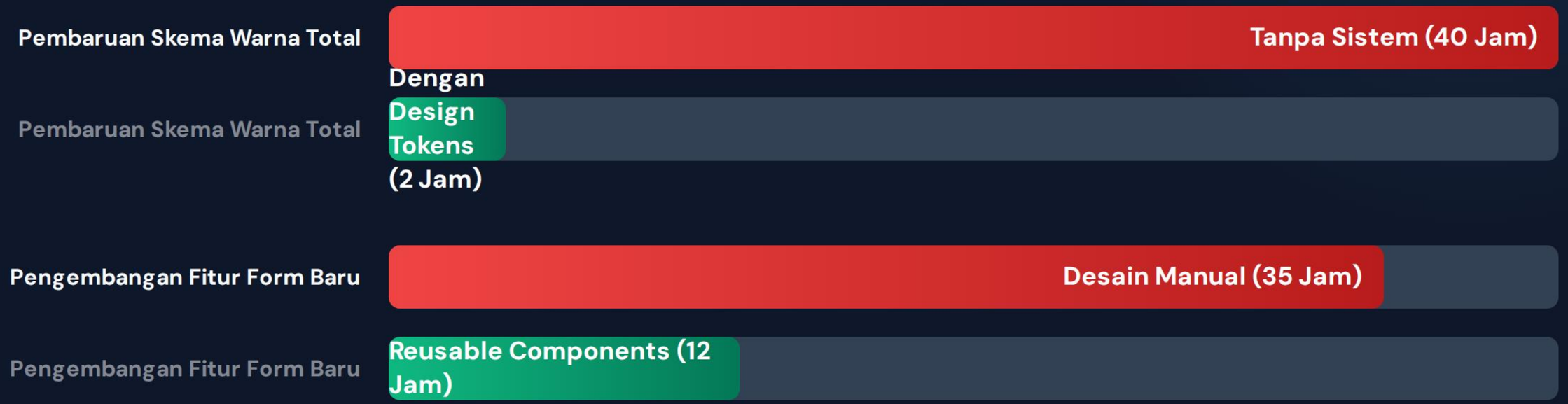
Integritas API: Data ditampilkan pada aplikasi web identik dengan aplikasi mobile karena berasal dari *service/backend* yang sama (REST/GraphQL).

Manajemen State: UI bereaksi terhadap perubahan data secara dinamis tanpa pengguna harus *refresh* halaman secara manual.

Ini memastikan *task continuity* (kontinuitas tugas) saat pengguna berpindah perangkat keras.

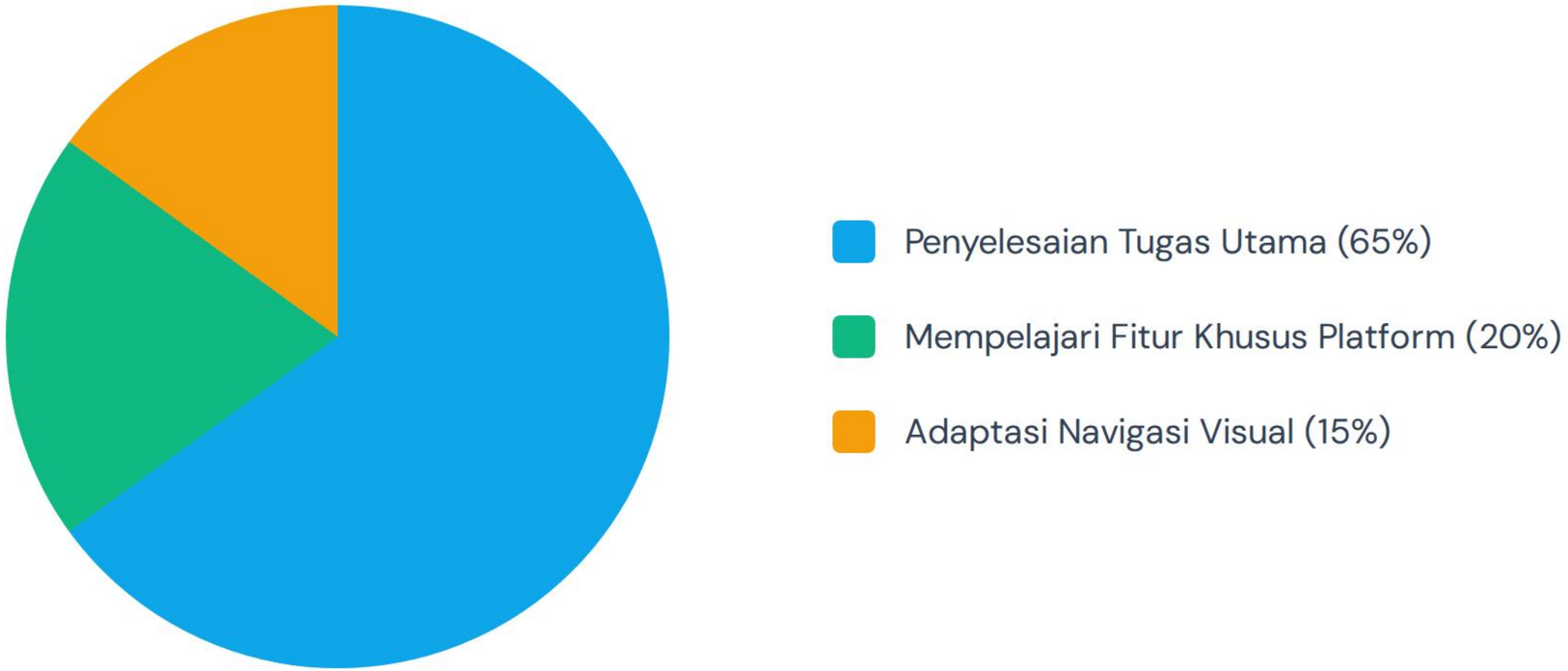


Dampak Implementasi Design System



Visualisasi estimasi waktu (Jam) pengembang aplikasi. Design System membutuhkan investasi waktu di awal untuk penyusunan (setup), namun secara drastis mempercepat proses skalabilitas dan pemeliharaan jangka panjang.

Reduksi Beban Kognitif Lintas Platform



Dengan UI/UX yang konsisten (Design System), sebagian besar energi mental (kapasitas kognitif) pengguna dialokasikan langsung untuk **menyelesaikan tujuan utama mereka**, bukan terbuang untuk mencoba memahami letak navigasi yang berbeda antar platform.

Bagian 7: Kesimpulan & Penutup

Merangkum Praktik Terbaik Desain Lintas Platform

Evaluasi Mandiri (Self-Assessment)

Uji pemahaman Anda terhadap materi Pertemuan 4 ini:

- Jelaskan perbedaan mendasar antara Desain Responsif (layout) dan Desain Adaptif (konteks input) dalam lintas platform!
- Mengapa penggunaan *Design Tokens* jauh lebih direkomendasikan ketimbang menuliskan nilai *hex color* secara langsung di dalam *codebase*?
- Sebutkan minimal dua implementasi prinsip aksesibilitas UI agar antarmuka ramah bagi tunanetra (screen reader) dan gangguan motorik (keyboard only).
- Dalam konteks Evaluasi Heuristik, berikan contoh penerapan prinsip *"Visibility of System Status"* pada aplikasi e-commerce!

Q & A

Terima kasih atas perhatiannya.

Mata Kuliah: Rekayasa Lintas Platform | Materi: Desain UI/UX Konsisten

Image Sources



https://cdn.prod.website-files.com/6365d860c7b7a7191055eb8a/68da31df35a45866e0463fde_best-react-component-libraries-cover.webp

Source: www.untitledui.com



https://images.presentationgo.com/2025/02/pastel-nature-color-palette-widescreen.png

Source: www.presentationgo.com



https://miro.medium.com/v2/resize:fit:1400/0*rOFaV0jCD0cTY_an

Source: medium.muz.li



https://img.magnific.com/free-vector/modern-web-design-button-collection-with-flat-design_23-2147938229.jpg?semt=ais_hybrid&w=740&q=80

Source: www.magnific.com



https://penpot.app/blog/content/images/thumbnail/DEV-GUIDE-TO-DT-AND-CSS.jpg

Source: penpot.app



https://storage.ghost.io/c/c8/0c/c80ce826-734b-44fa-90f7-43d0deb4f5bc/content/images/2024/05/Responsive---In-line-Image---02--1-.png

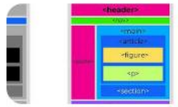
Source: bubble.io

Image Sources



https://cdn.prod.website-files.com/65130e79c72ae8812db3412e/657251be9b1da4fd89960241_Grid-System_Image-1.png

Source: designlab.com



https://miro.medium.com/1*Zdcpe1oOwoVUgC-oggCASw.jpeg

Source: medium.com



https://media.nngroup.com/media/editor/2018/11/23/best-buy-error.png

Source: www.nngroup.com