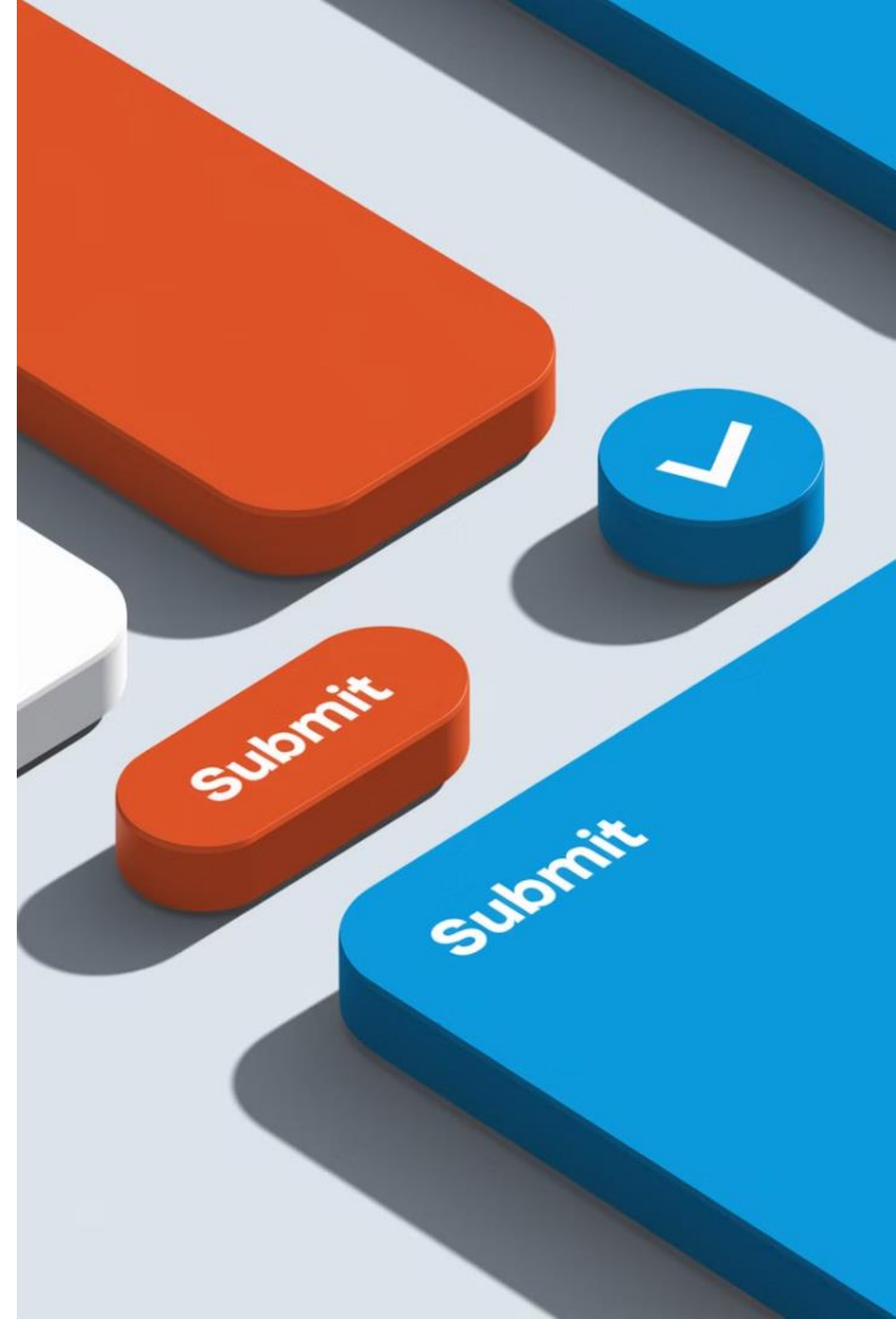


Prinsip dan Standar Desain UI

Dalam presentasi ini, kita akan membahas berbagai aspek penting dalam desain UI, mulai dari prinsip-prinsip dasar hingga evaluasi desain berdasarkan standar yang telah ditetapkan.



Agenda Presentasi

1

Prinsip Desain UI

Konsistensi, hierarki visual, dan keterbacaan dalam desain antarmuka pengguna

2

Standar Desain UI

Analisis standar desain UI pada berbagai platform

3

Responsive vs Adaptive Design

Perbedaan dan penerapan desain responsif dan adaptif

4

Prinsip Usability

Penerapan prinsip kegunaan dalam perancangan UI

5

Evaluasi Desain UI

Metode evaluasi desain berdasarkan standar yang telah ditetapkan

Prinsip Desain UI: Konsistensi

Definisi Konsistensi

Konsistensi dalam desain UI mengacu pada penggunaan elemen, pola, dan perilaku yang serupa di seluruh antarmuka. Hal ini menciptakan pengalaman yang dapat diprediksi dan memudahkan pengguna untuk belajar menggunakan sistem.

Jenis Konsistensi

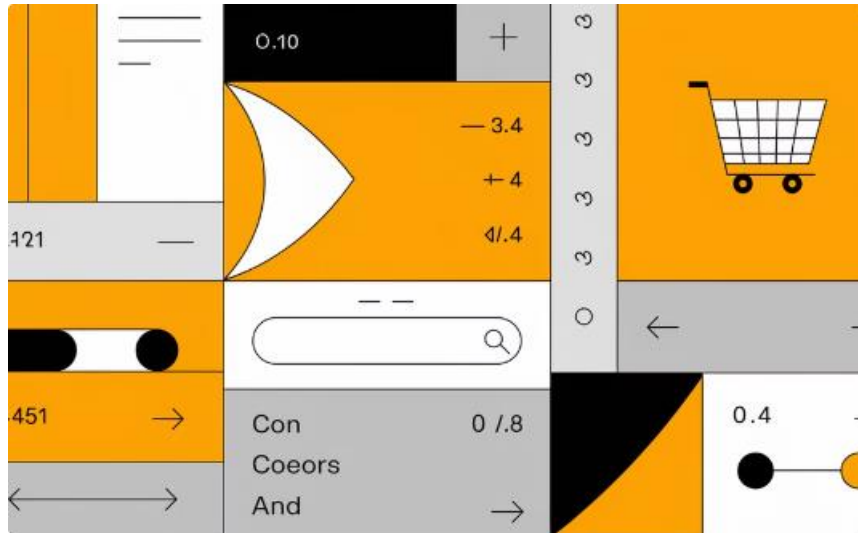
Konsistensi visual (warna, tipografi, ikon), konsistensi fungsional (perilaku interaksi), dan konsistensi eksternal (keselarasan dengan konvensi platform) adalah tiga jenis utama yang perlu diperhatikan dalam desain.

Manfaat Konsistensi

Meningkatkan efisiensi pembelajaran, mengurangi beban kognitif, meningkatkan kepercayaan pengguna, dan memperkuat identitas merek adalah beberapa manfaat utama dari konsistensi dalam desain UI.



Contoh Konsistensi dalam Desain UI



Konsistensi Visual

Penggunaan skema warna, tipografi, dan gaya visual yang konsisten di seluruh aplikasi menciptakan identitas yang kuat dan pengalaman yang menyatu. Perhatikan bagaimana elemen visual tetap konsisten di berbagai halaman.



Konsistensi Fungsional

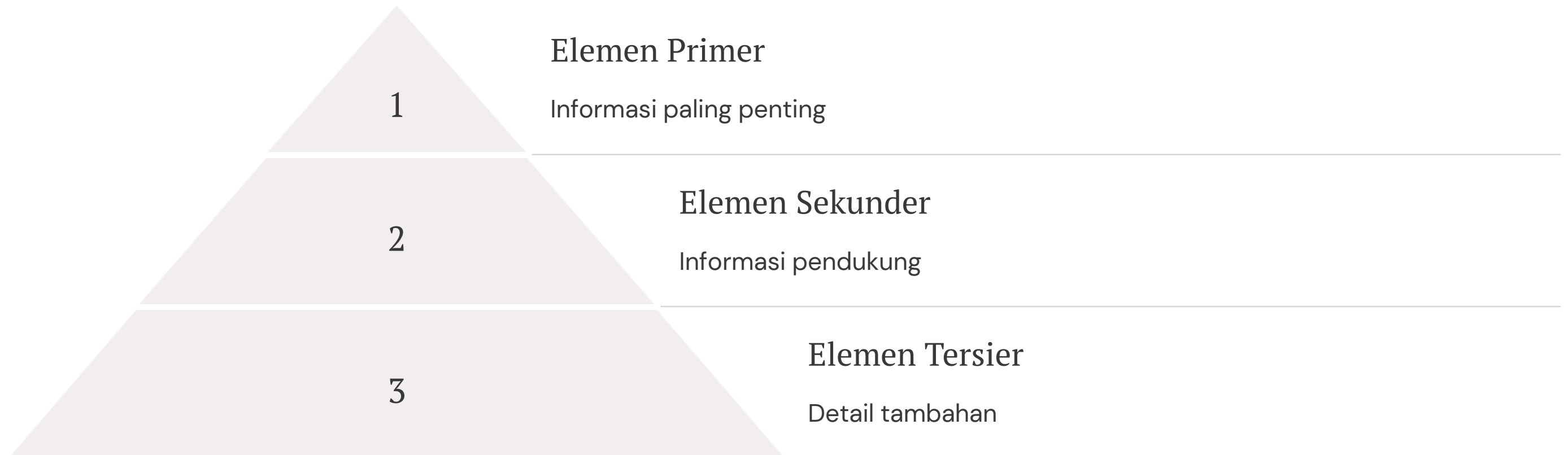
Tombol dan elemen interaktif yang berperilaku sama di seluruh aplikasi membantu pengguna memahami cara berinteraksi dengan sistem tanpa harus belajar ulang pada setiap halaman.



Konsistensi Eksternal

Mengikuti konvensi platform (seperti iOS atau Android) membantu pengguna merasa familiar dengan aplikasi baru karena sudah terbiasa dengan pola interaksi yang serupa di aplikasi lain.

Prinsip Desain UI: Hierarki Visual



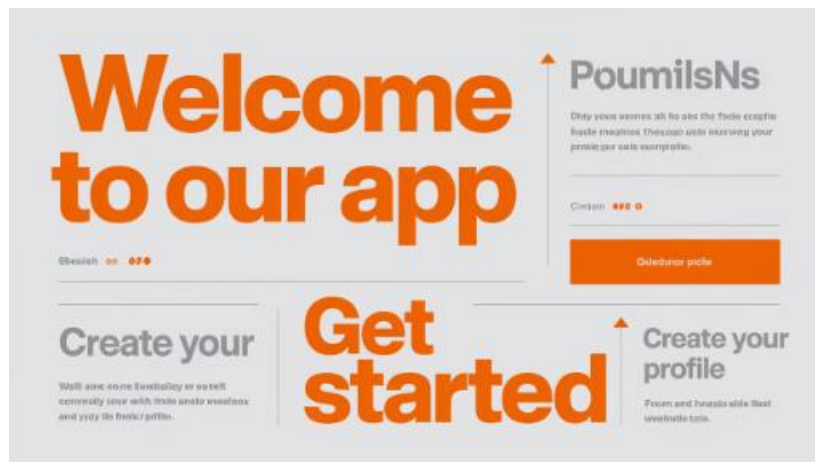
Hierarki visual adalah prinsip pengorganisasian elemen desain berdasarkan kepentingannya. Dengan menerapkan hierarki yang jelas, desainer dapat mengarahkan perhatian pengguna ke informasi yang paling penting terlebih dahulu, kemudian ke informasi pendukung, dan akhirnya ke detail tambahan.

Hierarki visual dapat dicapai melalui penggunaan ukuran, warna, kontras, penempatan, dan ruang kosong. Elemen yang lebih besar, lebih kontras, atau ditempatkan di posisi strategis akan menarik perhatian lebih banyak dan dipersepsikan sebagai lebih penting.

Teknik Menciptakan Hierarki Visual

Ukuran dan Skala

Elemen yang lebih besar secara alami akan menarik perhatian lebih banyak. Gunakan variasi ukuran untuk menunjukkan tingkat kepentingan relatif antar elemen. Judul utama biasanya lebih besar dari subjudul, yang lebih besar dari teks isi.



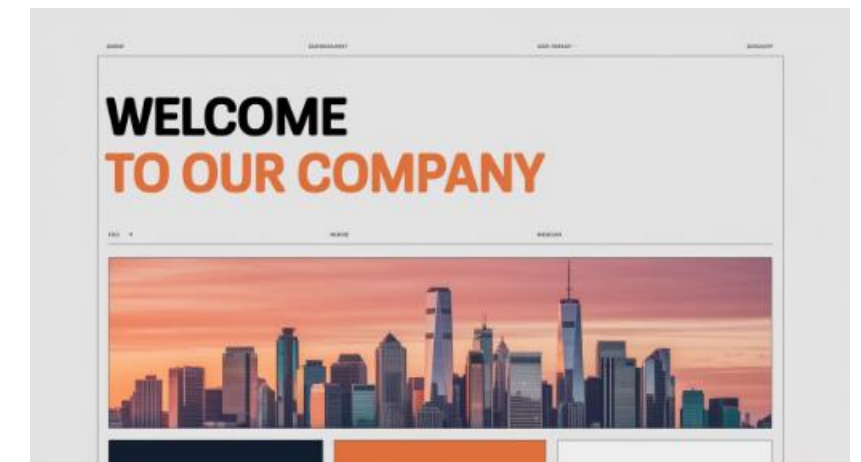
Warna dan Kontras

Warna yang lebih cerah atau kontras yang lebih tinggi akan menarik perhatian. Gunakan warna aksen untuk elemen penting dan warna netral untuk elemen sekunder. Pastikan kontras cukup untuk keterbacaan yang baik.



Penempatan dan Ruang

Elemen di bagian atas biasanya dianggap lebih penting. Gunakan ruang kosong (white space) untuk menciptakan pemisahan dan penekanan. Pengelompokan elemen terkait juga membantu menciptakan hierarki informasi.



Prinsip Desain UI: Keterbacaan

1 Tipografi yang Tepat

Pilih jenis huruf yang mudah dibaca pada berbagai ukuran dan perangkat. Pertimbangkan penggunaan font sans-serif untuk teks digital. Pastikan ukuran font cukup besar untuk dibaca dengan nyaman (minimal 16px untuk teks utama pada web).

3 Panjang Baris yang Optimal

Jaga panjang baris teks antara 50–75 karakter untuk keterbacaan optimal. Baris yang terlalu panjang menyulitkan mata untuk melacak dari akhir satu baris ke awal baris berikutnya.

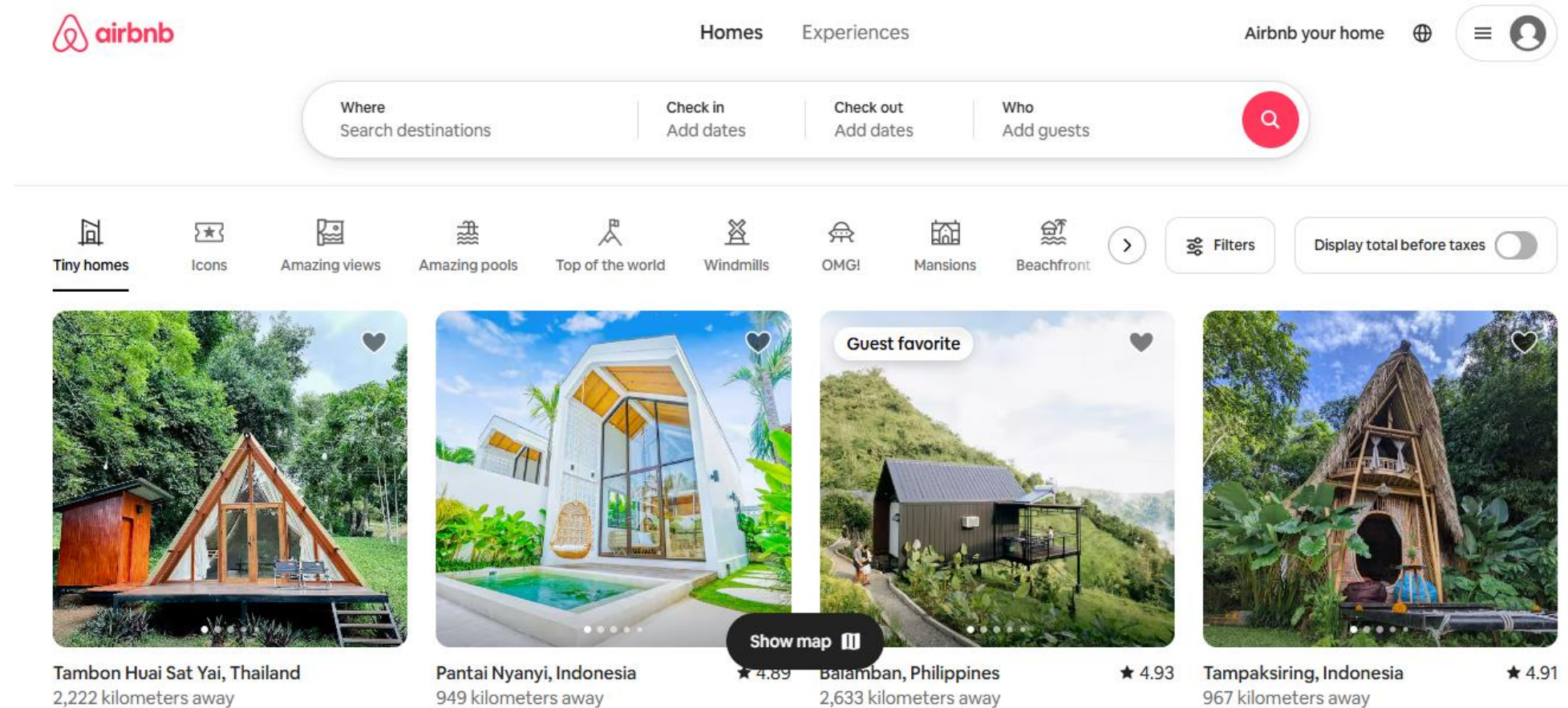
2 Kontras yang Memadai

Pastikan ada kontras yang cukup antara teks dan latar belakang. Rasio kontras minimal 4.5:1 untuk teks reguler dan 3:1 untuk teks besar sesuai standar WCAG 2.0 AA. Hindari kombinasi warna yang sulit dibaca seperti merah pada hijau.

4 Spasi yang Cukup

Gunakan spasi antar baris (line height) sekitar 1.5 kali ukuran font. Berikan juga ruang yang cukup antara paragraf dan elemen lainnya untuk menciptakan pemisahan visual yang jelas.

Contoh Penerapan Keterbacaan dalam UI



Keterbacaan yang baik sangat penting dalam desain UI karena jika pengguna tidak dapat membaca konten dengan mudah, mereka tidak akan dapat menggunakan produk secara efektif. Contoh di atas menunjukkan berbagai penerapan prinsip keterbacaan dalam desain antarmuka modern.

Perhatikan bagaimana penggunaan kontras yang tepat, pemilihan tipografi yang sesuai, dan pengaturan spasi yang baik secara signifikan meningkatkan keterbacaan dan pengalaman pengguna secara keseluruhan.

Standar Desain UI pada Berbagai Platform



Apple Human Interface Guidelines

Standar desain untuk ekosistem Apple (iOS, macOS, watchOS, tvOS). Menekankan pada kesederhanaan, kejelasan, dan kedalaman melalui hierarki visual. Menggunakan sistem desain yang terpadu dengan komponen dan pola interaksi yang konsisten.



Material Design (Google)

Sistem desain untuk Android dan web. Terinspirasi oleh dunia fisik dan tekstur, dengan prinsip material sebagai metafora. Menggunakan animasi, kedalaman, dan bayangan untuk menciptakan hierarki dan makna.



Microsoft Fluent Design

Standar desain untuk Windows dan ekosistem Microsoft. Berfokus pada lima elemen kunci: cahaya, kedalaman, gerakan, material, dan skala. Dirancang untuk bekerja dengan baik di berbagai perangkat dan modalitas input.



Web Content Accessibility Guidelines

Standar internasional untuk aksesibilitas web. Memberikan panduan untuk membuat konten web yang dapat diakses oleh semua orang, termasuk penyandang disabilitas. Mencakup prinsip-prinsip dapat dirasakan, dapat dioperasikan, dapat dipahami, dan kuat.

Perbandingan Standar Desain UI

Aspek	Apple HIG	Material Design	Fluent Design
Filosofi Desain	Kesederhanaan dan kejelasan	Material sebagai metafora	Harmonis, intuitif, dan inklusif
Navigasi	Tab bar, navigation bar	Bottom navigation, navigation drawer	Navigation view, command bar
Tipografi	San Francisco (sistem font)	Roboto, Product Sans	Segoe UI
Elemen Visual	Flat design dengan kedalaman minimal	Kartu, bayangan, dan elevasi	Acrylic material, parallax, reveal
Animasi	Halus dan fungsional	Bermakna dan responsif	Mengalir dan terhubung

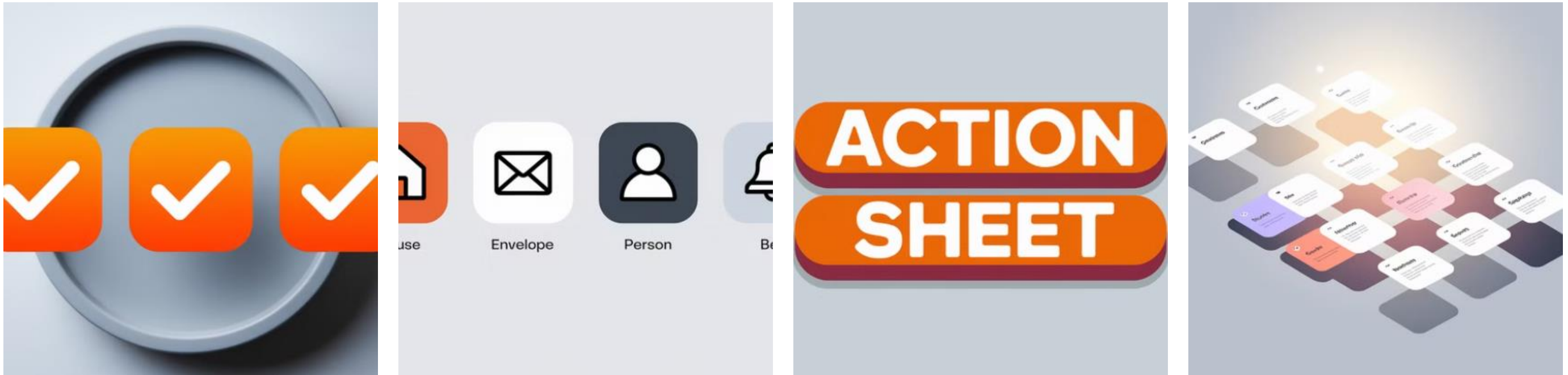


Material



Fluent

Komponen UI dalam Standar Apple HIG



Apple Human Interface Guidelines (HIG) menyediakan berbagai komponen UI standar yang dirancang untuk memberikan pengalaman pengguna yang konsisten di seluruh ekosistem Apple. Komponen-komponen ini telah dioptimalkan untuk berbagai ukuran layar dan modalitas input.

Beberapa komponen utama dalam HIG termasuk navigation bar untuk navigasi hierarkis, tab bar untuk navigasi antar kategori utama, action sheet untuk menampilkan pilihan tindakan, dan table view untuk menampilkan daftar informasi. Penggunaan komponen standar ini membantu pengguna merasa familiar dengan aplikasi baru.

Komponen UI dalam Material Design

Bottom Navigation

Komponen navigasi utama yang terletak di bagian bawah layar, memungkinkan perpindahan antar tampilan utama dengan sekali sentuh. Biasanya menampilkan 3–5 tujuan dengan ikon dan label teks.

Cards

Permukaan yang berisi konten dan tindakan terkait dengan satu topik. Kartu dapat berisi teks, gambar, dan kontrol seperti tombol. Kartu memberikan cara konsisten untuk menyajikan informasi serupa.

Floating Action Button (FAB)

Tombol lingkaran yang mengambang di atas antarmuka, mewakili tindakan utama dalam aplikasi. FAB harus digunakan untuk tindakan positif seperti membuat, berbagi, atau menjelajah.

Snackbars

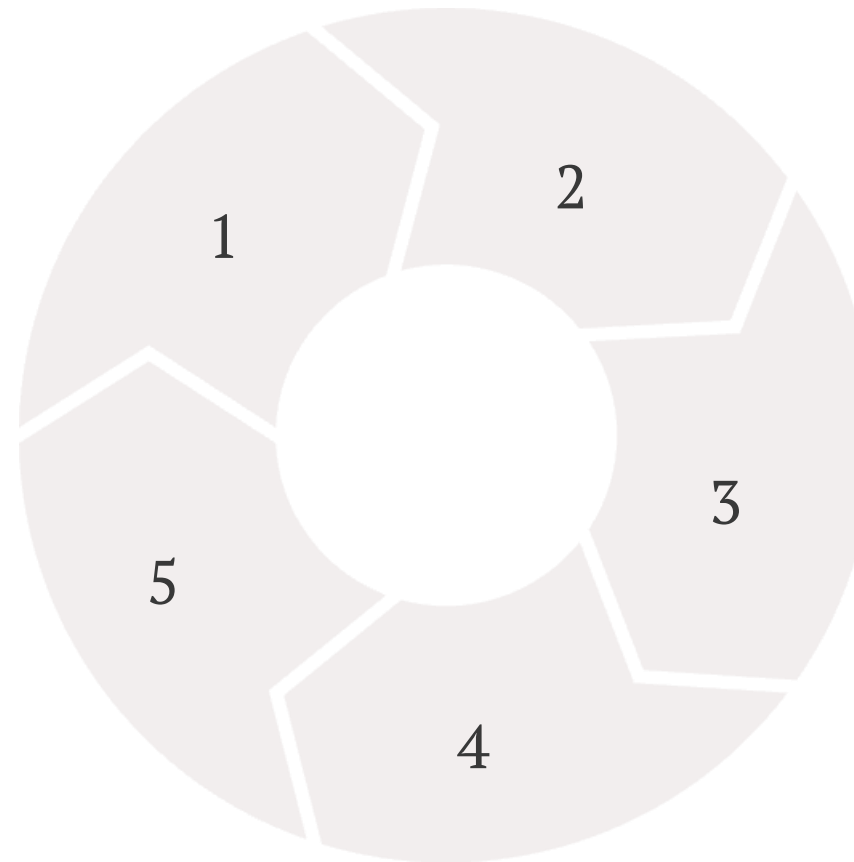
Pesan singkat yang muncul di bagian bawah layar untuk memberikan umpan balik tentang operasi. Dapat berisi tindakan tunggal dan secara otomatis menghilang setelah beberapa saat.



Komponen UI dalam Fluent Design

Navigation View
Panel navigasi yang dapat diciutkan

Reveal Highlight
Efek pencahayaan interaktif



Command Bar

Bilah alat untuk tindakan umum

Pivot

Kontrol tab untuk navigasi

Acrylic Material

Efek transparan dengan blur

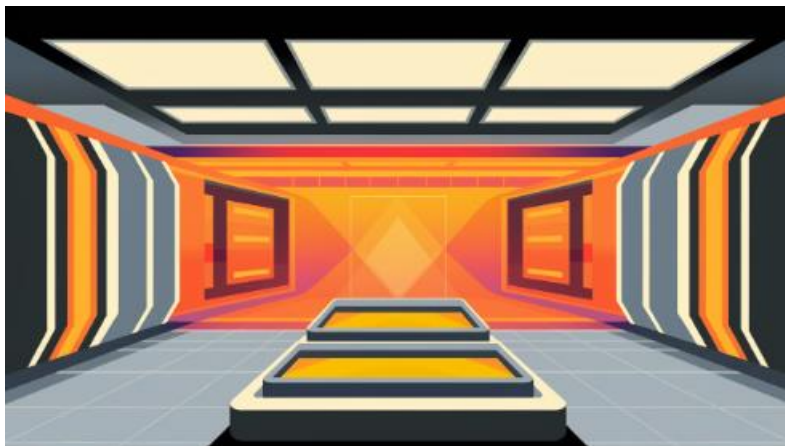
Microsoft Fluent Design System menawarkan berbagai komponen UI yang dirancang untuk bekerja secara harmonis di seluruh perangkat Windows. Sistem ini menggabungkan elemen visual yang kaya dengan interaksi yang intuitif untuk menciptakan pengalaman yang menarik dan inklusif.

Komponen-komponen ini dirancang dengan mempertimbangkan berbagai modalitas input, termasuk sentuh, pena, keyboard, mouse, dan bahkan interaksi suara dan pandangan, menjadikannya sangat serbaguna untuk berbagai jenis aplikasi.

Analisis Standar Desain UI: Kelebihan dan Kekurangan

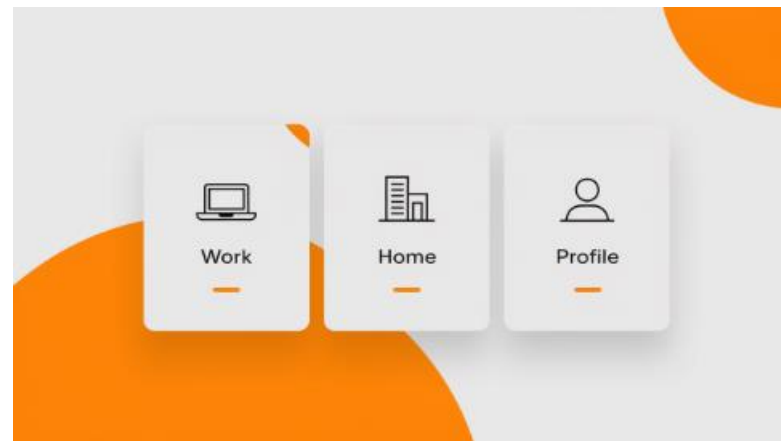
Apple HIG

- Kelebihan: Konsistensi tinggi, pengalaman pengguna yang terpadu, estetika yang bersih
- Kekurangan: Kurang fleksibel, terlalu ketat, terbatas pada ekosistem Apple



Material Design

- Kelebihan: Sistem yang komprehensif, adaptabilitas tinggi, dokumentasi yang kaya
- Kekurangan: Kadang terasa terlalu generik, implementasi yang tidak konsisten



Fluent Design

- Kelebihan: Inklusif untuk berbagai modalitas input, efek visual yang menarik
- Kekurangan: Adopsi yang lebih lambat, kurang konsisten di seluruh produk Microsoft



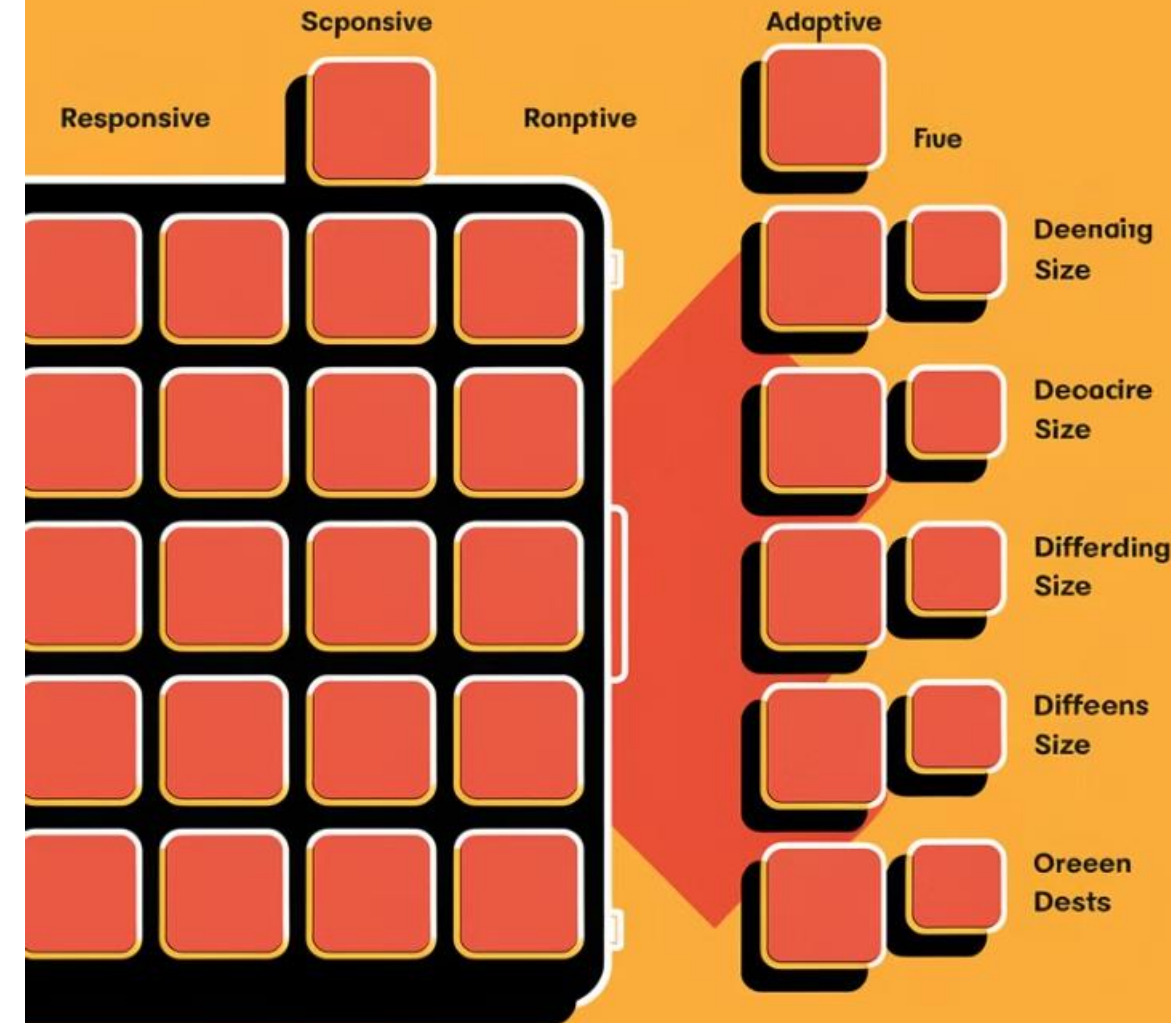
Responsive Design vs Adaptive Design

Responsive Design

Pendekatan desain yang membuat tampilan website menyesuaikan diri secara mulus dengan ukuran layar perangkat. Menggunakan CSS media queries dan unit relatif (% , em, rem) untuk mengubah tata letak secara dinamis. Satu layout yang "mengalir" dan menyesuaikan diri dengan berbagai ukuran viewport.

Adaptive Design

Pendekatan desain yang menyediakan beberapa tata letak tetap yang dirancang untuk ukuran layar tertentu. Server mendeteksi perangkat dan memuat layout yang sesuai. Biasanya menggunakan beberapa breakpoint tetap (misalnya: 320px, 768px, 1024px) dengan layout yang dioptimalkan untuk masing-masing.



Cara Kerja Responsive Design

Fluid Grids

1

Menggunakan unit relatif seperti persentase daripada piksel tetap untuk ukuran elemen. Ini memungkinkan elemen untuk memperbesar atau memperkecil secara proporsional dengan ukuran viewport. Misalnya, kolom mungkin didefinisikan sebagai 50% dari lebar kontainer, bukan 500px tetap.

Media Queries

3

Blok CSS yang hanya diterapkan pada kondisi tertentu, seperti lebar viewport minimum atau maksimum. Memungkinkan penerapan gaya yang berbeda berdasarkan karakteristik perangkat, seperti mengubah tata letak dari tiga kolom menjadi satu kolom pada layar kecil.

4

Flexible Images

Gambar yang dapat menyesuaikan ukuran dengan menggunakan properti CSS seperti `max-width: 100%`. Ini memastikan gambar tidak pernah lebih besar dari elemen induknya dan dapat menyusut saat diperlukan tanpa kehilangan proporsi aslinya.

Viewport Meta Tag

Tag HTML yang memberi tahu browser cara menyesuaikan dimensi dan skala halaman web untuk berbagai perangkat. Tag viewport yang umum adalah: `<meta name="viewport" content="width=device-width, initial-scale=1">`.



Cara Kerja Adaptive Design

1

Deteksi Perangkat

Server mendeteksi jenis perangkat atau ukuran layar pengguna melalui user agent atau teknik deteksi lainnya. Informasi ini digunakan untuk menentukan versi layout mana yang harus disajikan.

2

Breakpoint Tetap

Desainer membuat beberapa versi layout yang berbeda untuk ukuran layar tertentu (misalnya: mobile, tablet, desktop). Setiap layout dioptimalkan secara khusus untuk ukuran layar tersebut.

3

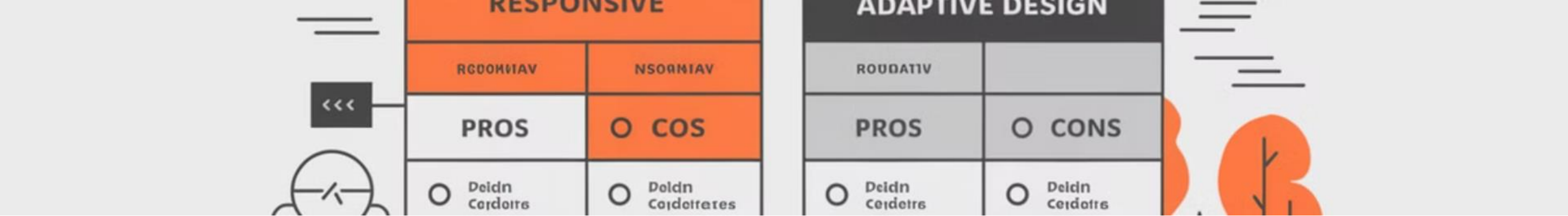
Pengiriman Layout

Server mengirimkan versi layout yang paling sesuai dengan perangkat pengguna. Layout ini biasanya memiliki ukuran tetap dan tidak banyak berubah setelah dimuat di perangkat.

4

Pengoptimalan Konten

Selain layout, konten juga dapat dioptimalkan untuk perangkat tertentu, seperti mengirimkan gambar resolusi rendah ke perangkat mobile untuk menghemat bandwidth.



Perbandingan Responsive vs Adaptive Design

Aspek	Responsive Design	Adaptive Design
Fleksibilitas	Sangat fleksibel, menyesuaikan dengan semua ukuran layar	Kurang fleksibel, hanya optimal pada breakpoint yang ditentukan
Pengembangan	Satu codebase untuk semua perangkat	Multiple layouts yang harus dikelola secara terpisah
Performa	Dapat lebih berat karena harus memuat semua aset	Dapat lebih ringan karena hanya memuat aset yang diperlukan
Kontrol Desain	Kontrol lebih sedikit pada layout spesifik	Kontrol lebih besar pada layout untuk ukuran tertentu
Pemeliharaan	Lebih mudah dikelola dan diperbarui	Lebih kompleks untuk dikelola dan diperbarui
Pengalaman Pengguna	Konsisten di semua perangkat, tapi mungkin tidak optimal	Dioptimalkan untuk perangkat tertentu, tapi mungkin tidak konsisten

Kapan Menggunakan Responsive Design

1 Proyek Baru

Untuk proyek yang dimulai dari awal, responsive design sering menjadi pilihan yang lebih baik karena memungkinkan satu codebase yang dapat beradaptasi dengan perangkat baru di masa depan tanpa perlu pengembangan tambahan.

3 Sumber Daya Terbatas

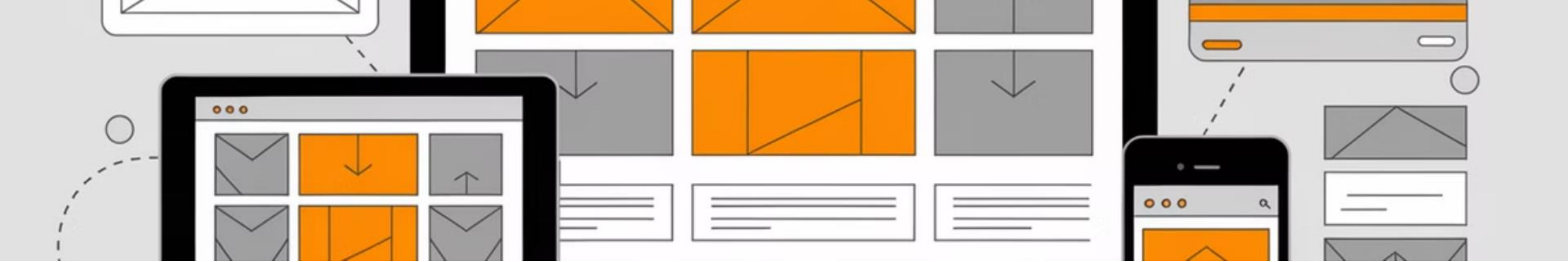
Dengan sumber daya pengembangan yang terbatas, responsive design memungkinkan Anda untuk memelihara satu versi situs daripada beberapa versi terpisah, menghemat waktu dan upaya dalam jangka panjang.

2 Konten yang Konsisten

Jika Anda ingin menyajikan konten yang sama ke semua pengguna terlepas dari perangkat mereka, responsive design memungkinkan konsistensi konten dengan penyesuaian layout yang mulus.

4 SEO yang Lebih Baik

Google secara eksplisit merekomendasikan responsive design dan dapat memberikan peringkat lebih tinggi untuk situs yang mobile-friendly. Satu URL untuk semua perangkat juga menyederhanakan strategi SEO.



Kapan Menggunakan Adaptive Design



Performa Optimal

Ketika performa adalah prioritas utama, adaptive design memungkinkan pengiriman konten yang dioptimalkan khusus untuk setiap perangkat, seperti gambar dengan resolusi yang sesuai atau skrip yang lebih ringan untuk perangkat mobile.



Situs Lama

Untuk situs yang sudah ada dengan basis pengguna yang besar, adaptive design memungkinkan penambahan dukungan mobile tanpa perlu merombak seluruh situs, meminimalkan risiko dan gangguan.



Pengalaman yang Sangat Berbeda

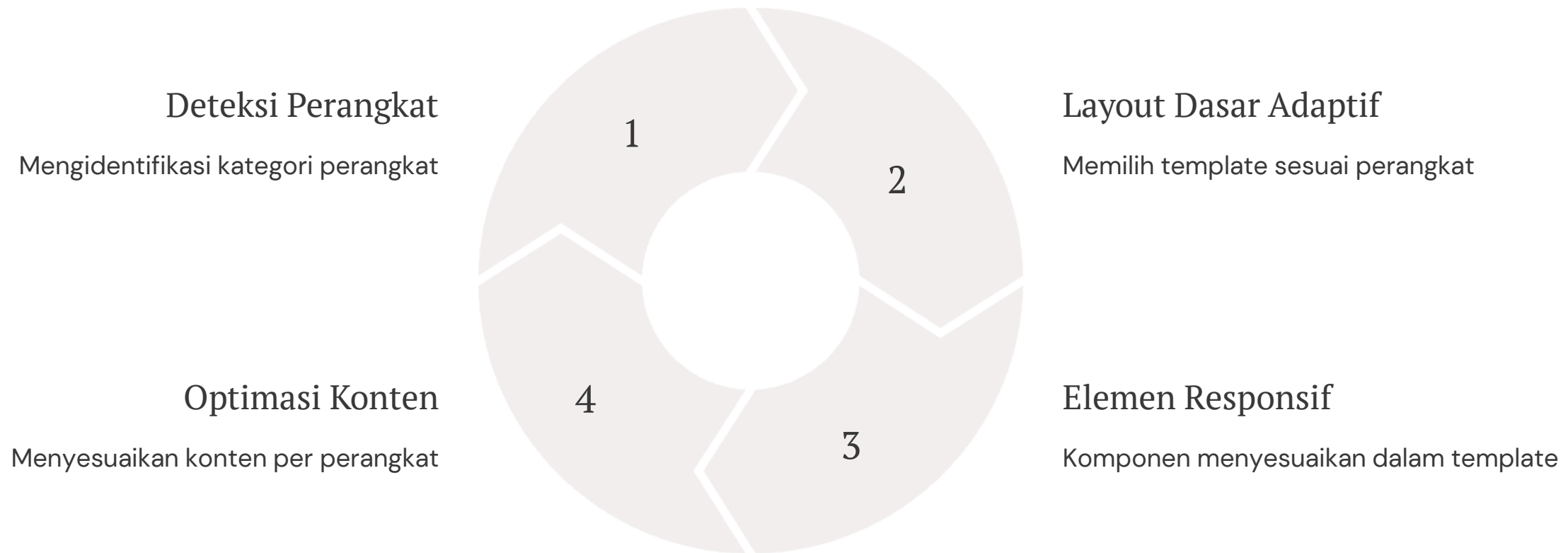
Jika Anda ingin menyediakan pengalaman yang sangat berbeda berdasarkan perangkat (misalnya, aplikasi desktop yang kaya fitur vs. versi mobile yang disederhanakan), adaptive design memberikan kontrol lebih besar.



Analitik Spesifik

Adaptive design memudahkan pelacakan dan analisis perilaku pengguna berdasarkan jenis perangkat, memungkinkan optimasi yang lebih tepat sasaran untuk segmen pengguna tertentu.

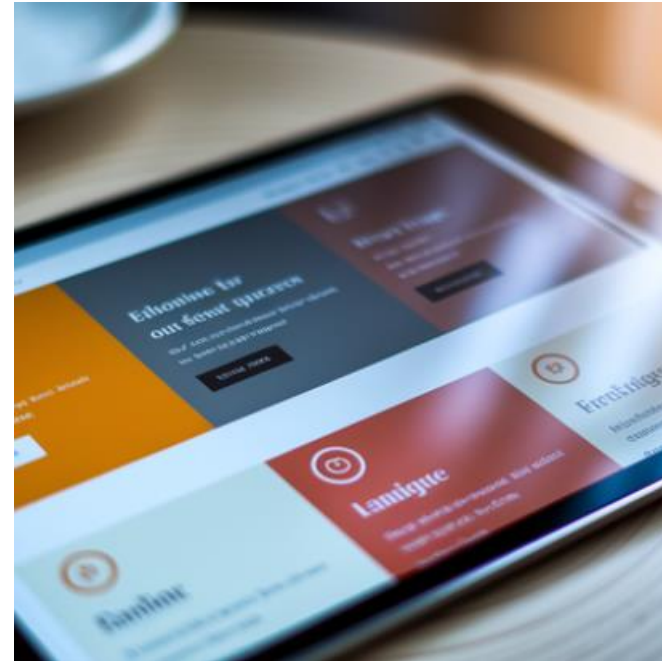
Pendekatan Hybrid: Menggabungkan Responsive dan Adaptive



Pendekatan hybrid menggabungkan kelebihan dari kedua metode desain. Ini melibatkan pembuatan beberapa template dasar untuk kategori perangkat utama (seperti pada adaptive design), tetapi dengan elemen-elemen di dalamnya yang responsif dan dapat menyesuaikan diri dengan variasi ukuran dalam kategori tersebut.

Dengan pendekatan ini, Anda bisa mendapatkan kontrol yang lebih baik atas pengalaman pengguna pada perangkat yang berbeda sambil tetap mempertahankan fleksibilitas untuk menangani berbagai ukuran layar dalam setiap kategori. Ini juga memungkinkan optimasi performa dengan hanya memuat aset yang diperlukan untuk perangkat tertentu.

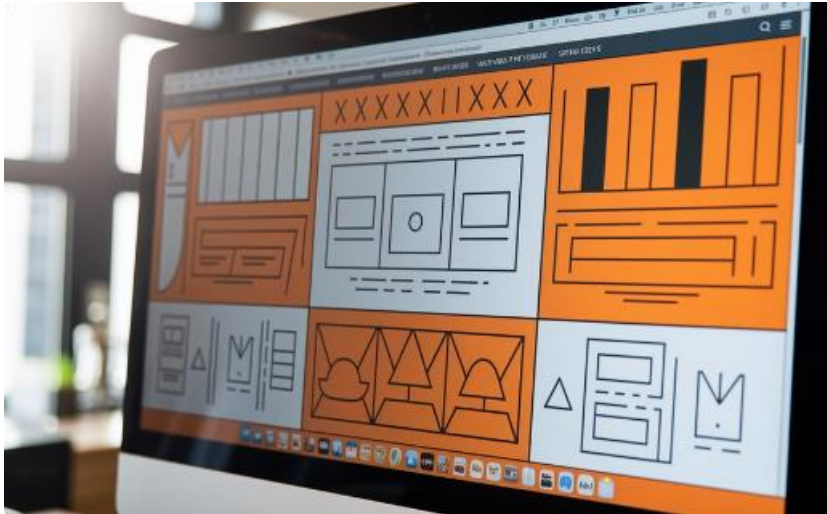
Contoh Implementasi Responsive Design



Implementasi responsive design biasanya melibatkan penggunaan CSS media queries untuk menerapkan gaya yang berbeda berdasarkan karakteristik perangkat. Dengan pendekatan "mobile-first", desainer mulai dengan layout dasar untuk perangkat mobile, kemudian secara progresif meningkatkan layout untuk layar yang lebih besar.

Perhatikan bagaimana layout berubah dari tata letak multi-kolom pada desktop menjadi tata letak kolom tunggal pada smartphone, dengan ukuran font, spasi, dan proporsi elemen yang disesuaikan untuk memberikan pengalaman optimal pada setiap perangkat. Semua ini dicapai dengan satu set HTML yang sama dan CSS yang responsif.

Contoh Implementasi Adaptive Design



Versi Desktop

Versi desktop dari situs adaptive menampilkan semua fitur dan konten, dengan tata letak yang dioptimalkan untuk layar besar dan interaksi mouse.

Perhatikan navigasi horizontal dan panel samping yang memanfaatkan ruang layar yang tersedia.



Versi Tablet

Versi tablet menyajikan layout yang dimodifikasi dengan elemen yang lebih besar untuk interaksi sentuh, tetapi masih mempertahankan sebagian besar fitur versi desktop. Navigasi mungkin diubah untuk mengakomodasi layar yang lebih kecil.



Versi Mobile

Versi mobile menawarkan pengalaman yang sangat disederhanakan, berfokus pada tugas-tugas utama dengan navigasi yang dioptimalkan untuk layar kecil, seperti menu hamburger. Beberapa fitur mungkin disembunyikan atau diubah untuk mengutamakan konten penting.

Prinsip Usability dalam UI Design

1

Efektivitas

Pengguna dapat mencapai tujuan

2

Efisiensi

Tugas diselesaikan dengan cepat

3

Kepuasan

Pengalaman yang menyenangkan

Usability atau kegunaan adalah ukuran seberapa mudah dan menyenangkan suatu antarmuka digunakan. Prinsip ini berfokus pada memastikan bahwa pengguna dapat mencapai tujuan mereka dengan efektif, efisien, dan memuaskan. Antarmuka dengan usability yang baik memungkinkan pengguna untuk belajar dengan cepat, menyelesaikan tugas dengan minimal kesalahan, dan mengingat cara menggunakan sistem bahkan setelah periode tidak menggunakannya.

Dalam desain UI, usability harus menjadi pertimbangan utama karena secara langsung memengaruhi pengalaman pengguna dan keberhasilan produk. Antarmuka yang sulit digunakan akan menyebabkan frustrasi, kesalahan, dan akhirnya penolakan oleh pengguna.

10 Prinsip Usability Nielsen

Visibilitas Status Sistem

Sistem harus selalu memberi tahu pengguna tentang apa yang sedang terjadi melalui umpan balik yang tepat dalam waktu yang wajar. Contohnya adalah indikator loading, konfirmasi tindakan, dan notifikasi status.



Kecocokan Sistem dengan Dunia Nyata

Sistem harus berbicara dalam bahasa pengguna, dengan kata, frasa, dan konsep yang familiar bagi pengguna, bukan istilah berorientasi sistem. Ikuti konvensi dunia nyata, membuat informasi muncul secara alami dan logis.



Kontrol dan Kebebasan Pengguna

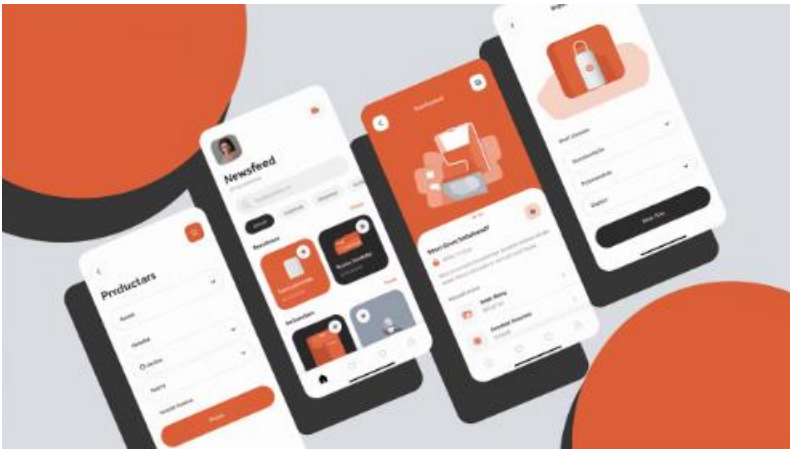
Pengguna sering memilih fungsi sistem secara tidak sengaja dan membutuhkan "pintu keluar darurat" yang jelas untuk meninggalkan keadaan yang tidak diinginkan tanpa harus melalui dialog yang panjang. Dukung undo dan redo.



10 Prinsip Usability Nielsen (Lanjutan)

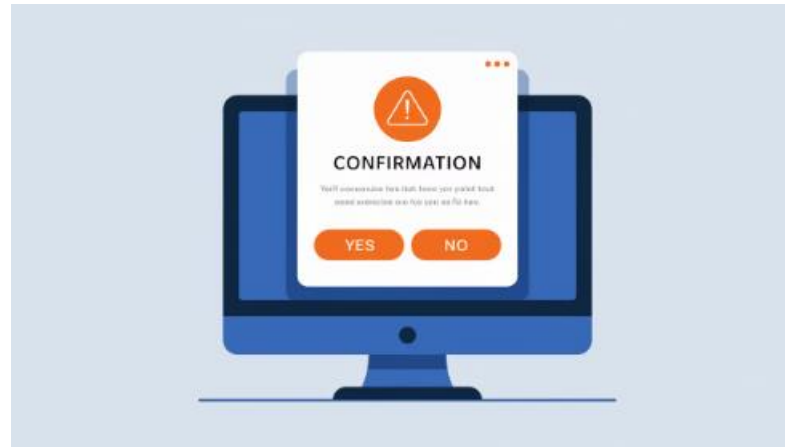
Konsistensi dan Standar

Pengguna tidak perlu bertanya-tanya apakah kata, situasi, atau tindakan yang berbeda berarti hal yang sama. Ikuti konvensi platform dan pastikan elemen serupa berperilaku dengan cara yang konsisten di seluruh aplikasi.



Pencegahan Kesalahan

Lebih baik dari pesan kesalahan yang bagus adalah desain yang mencegah masalah terjadi. Eliminasi kondisi rawan kesalahan atau periksa untuk mereka dan berikan pengguna opsi konfirmasi sebelum mereka melakukan tindakan.



Pengenalan daripada Mengingat

Minimalkan beban memori pengguna dengan membuat objek, tindakan, dan opsi terlihat. Pengguna tidak perlu mengingat informasi dari satu bagian dialog ke bagian lain. Instruksi penggunaan sistem harus terlihat atau mudah diambil.



10 Prinsip Usability Nielsen (Lanjutan)

Fleksibilitas dan Efisiensi

Akselerator—tidak terlihat oleh pengguna pemula—dapat mempercepat interaksi bagi pengguna ahli, sehingga sistem dapat melayani pengguna yang tidak berpengalaman dan berpengalaman. Izinkan pengguna untuk menyesuaikan tindakan yang sering dilakukan.



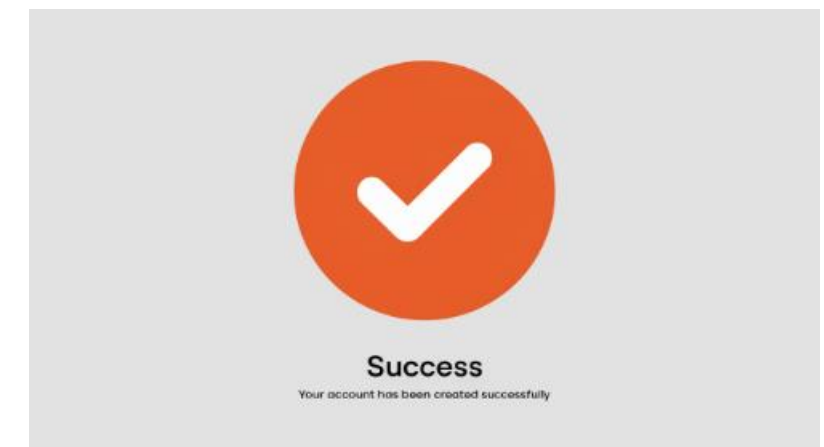
Estetika dan Desain Minimalis

Dialog tidak boleh berisi informasi yang tidak relevan atau jarang dibutuhkan. Setiap unit informasi ekstra dalam dialog bersaing dengan unit informasi yang relevan dan mengurangi visibilitas relatifnya.



Bantuan Mendiagnosis dan Memulihkan dari Kesalahan

Pesan kesalahan harus diekspresikan dalam bahasa biasa (tanpa kode), secara tepat menunjukkan masalah, dan secara konstruktif menyarankan solusi.



10 Prinsip Usability Nielsen (Lanjutan)

Bantuan dan Dokumentasi

Meskipun lebih baik jika sistem dapat digunakan tanpa dokumentasi, mungkin perlu untuk menyediakan bantuan dan dokumentasi. Informasi tersebut harus mudah dicari, berfokus pada tugas pengguna, mencantumkan langkah-langkah konkret yang harus dilakukan, dan tidak terlalu besar.

Prinsip-prinsip usability Nielsen ini, yang dikembangkan oleh Jakob Nielsen, adalah panduan yang sangat dihormati dalam desain antarmuka pengguna. Mereka menyediakan kerangka kerja untuk mengevaluasi dan meningkatkan kegunaan sistem interaktif.

Meskipun prinsip-prinsip ini dikembangkan pada tahun 1990-an, mereka tetap relevan hingga saat ini dan dapat diterapkan pada berbagai jenis antarmuka, dari aplikasi desktop tradisional hingga aplikasi mobile dan web modern. Memahami dan menerapkan prinsip-prinsip ini dapat secara signifikan meningkatkan pengalaman pengguna dan keberhasilan produk.

Penerapan Prinsip Usability dalam Desain Form

1 Label yang Jelas

Gunakan label yang deskriptif dan posisikan di atas atau di sebelah kiri field input. Hindari menggunakan placeholder sebagai pengganti label karena akan menghilang saat pengguna mulai mengetik, memaksa mereka untuk mengingat apa yang harus dimasukkan.

3 Grouping Logis

Kelompokkan field yang terkait secara logis dan gunakan judul bagian untuk membuat struktur form yang jelas. Ini membantu pengguna memahami hubungan antar field dan mempermudah pengisian form yang panjang.

2 Validasi Real-time

Berikan umpan balik langsung saat pengguna mengetik, bukan hanya setelah pengiriman form. Tunjukkan pesan kesalahan yang spesifik dan konstruktif di dekat field yang bermasalah, dengan petunjuk jelas tentang cara memperbaikinya.

4 Progres yang Jelas

Untuk form multi-langkah, tunjukkan indikator progres yang jelas sehingga pengguna tahu berapa banyak langkah yang tersisa. Berikan opsi untuk menyimpan progres dan melanjutkan nanti jika form sangat panjang.

Penerapan Prinsip Usability dalam Desain Navigasi

1

Struktur yang Jelas

Buat hierarki navigasi yang logis dan mudah dipahami. Pengguna harus selalu tahu di mana mereka berada dalam struktur situs atau aplikasi. Gunakan breadcrumbs, highlight pada item menu aktif, dan judul halaman yang jelas.

2

Konsistensi Penempatan

Tempatkan elemen navigasi utama secara konsisten di seluruh situs atau aplikasi. Navigasi utama harus mudah ditemukan dan tetap di lokasi yang sama di setiap halaman untuk menciptakan pengalaman yang dapat diprediksi.

3

Pengelompokan yang Bermakna

Kelompokkan item navigasi berdasarkan kategori yang bermakna bagi pengguna. Batasi jumlah item menu utama (idealnya 7 ± 2) untuk menghindari kelebihan pilihan. Gunakan dropdown atau mega menu untuk kategori dengan banyak sub-item.

4

Pencarian yang Efektif

Sediakan fitur pencarian yang mudah diakses untuk pengguna yang tahu apa yang mereka cari. Pencarian harus memberikan hasil yang relevan dan memungkinkan penyaringan atau penyortiran hasil untuk menemukan informasi dengan cepat.

Penerapan Prinsip Usability dalam Desain Mobile



Desain Ramah Jempol

Posisikan elemen interaktif penting di area yang mudah dijangkau jempol. Untuk sebagian besar pengguna smartphone, ini berarti bagian tengah hingga bawah layar. Hindari menempatkan tombol penting di sudut atas yang sulit dijangkau dengan satu tangan.



Target Sentuh yang Memadai

Buat target sentuh cukup besar (minimal 44x44 piksel) dan berikan ruang yang cukup di antara elemen interaktif untuk menghindari kesalahan sentuh. Ingat bahwa jari lebih besar dan kurang presisi dibandingkan pointer mouse.



Gestur yang Intuitif

Gunakan gestur yang sudah umum dan berikan petunjuk visual untuk gestur yang kurang familiar. Misalnya, pengguna sudah terbiasa dengan swipe untuk scroll, pinch untuk zoom, dan tap untuk memilih, tetapi gestur yang lebih kompleks mungkin memerlukan petunjuk.

Penerapan Prinsip Usability dalam Desain Konten

Scannable

Buat konten yang mudah dipindai dengan menggunakan judul yang informatif, paragraf pendek, daftar bullet, dan penekanan visual pada kata-kata kunci. Pengguna cenderung memindai daripada membaca kata per kata, terutama di web.

Hierarki Informasi

Sajikan informasi paling penting terlebih dahulu, diikuti oleh detail pendukung. Gunakan prinsip "piramida terbalik" dari jurnanisme, dengan informasi paling penting di awal dan detail yang semakin spesifik setelahnya.

Bahasa yang Jelas

Gunakan bahasa yang sederhana dan langsung. Hindari jargon teknis kecuali untuk audiens yang tepat. Tulis dalam kalimat aktif dan singkat. Gunakan terminologi yang konsisten di seluruh antarmuka.

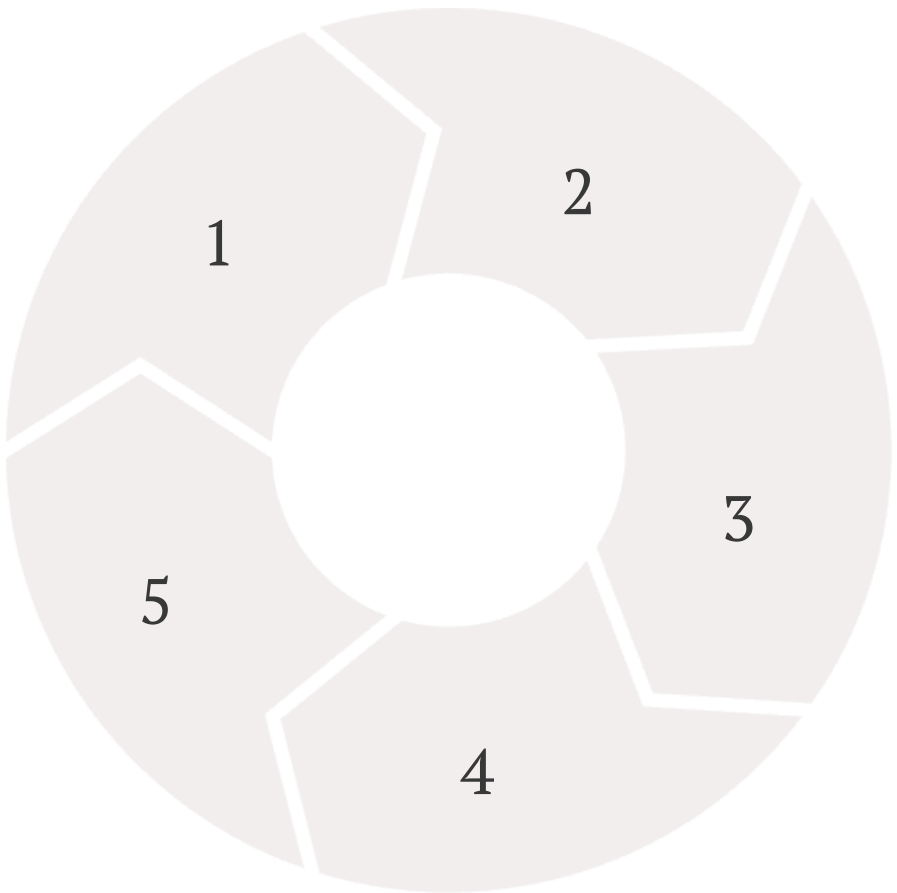
Multimedia yang Relevan

Gunakan gambar, video, atau infografis untuk menjelaskan konsep kompleks. Pastikan media mendukung konten dan tidak hanya dekoratif. Sediakan teks alternatif untuk aksesibilitas.

Evaluasi Desain UI: Metode dan Pendekatan

Pengujian Pengguna
Observasi langsung dengan pengguna nyata

A/B Testing
Perbandingan langsung dua desain



Evaluasi Heuristik
Penilaian ahli berdasarkan prinsip

Analisis Metrik
Pengukuran kuantitatif performa

Survei & Wawancara
Umpan balik langsung dari pengguna

Evaluasi desain UI adalah proses sistematis untuk menilai efektivitas, efisiensi, dan kepuasan pengguna terhadap antarmuka. Evaluasi dapat dilakukan pada berbagai tahap pengembangan, dari prototipe awal hingga produk yang sudah diluncurkan.

Pendekatan evaluasi yang komprehensif biasanya menggabungkan beberapa metode untuk mendapatkan pemahaman yang lebih lengkap tentang kekuatan dan kelemahan desain. Hasil evaluasi kemudian digunakan untuk memperbaiki dan mengoptimalkan desain UI.



Pengujian Pengguna (User Testing)

Perencanaan

Tentukan tujuan pengujian, rekrut peserta yang mewakili pengguna target, dan siapkan skenario tugas yang realistis. Pilih metode pengujian yang sesuai (moderasi langsung, jarak jauh, atau tidak dimoderasi) dan siapkan lingkungan pengujian.

Pelaksanaan

Jelaskan proses kepada peserta, tekankan bahwa Anda menguji produk bukan mereka. Minta peserta untuk berpikir keras saat menyelesaikan tugas. Catat interaksi, kesalahan, kebingungan, dan komentar. Hindari mengarahkan peserta.

Analisis

Identifikasi pola masalah yang muncul di beberapa peserta. Prioritaskan masalah berdasarkan frekuensi dan tingkat keparahan. Catat juga apa yang berhasil dengan baik, bukan hanya masalah.

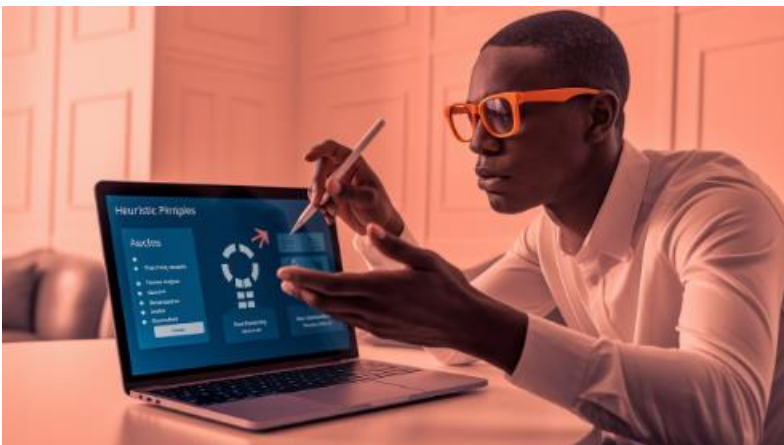
Rekomendasi

Buat rekomendasi spesifik untuk memperbaiki masalah yang ditemukan. Kaitkan rekomendasi dengan prinsip desain dan temuan pengujian. Prioritaskan perbaikan berdasarkan dampak pada pengalaman pengguna.

Evaluasi Heuristik

Definisi

Evaluasi heuristik adalah metode inspeksi usability di mana satu atau lebih ahli menilai antarmuka berdasarkan seperangkat prinsip usability (heuristik). Metode ini dikembangkan oleh Jakob Nielsen dan Rolf Molich sebagai cara yang cepat dan hemat biaya untuk menemukan masalah usability.

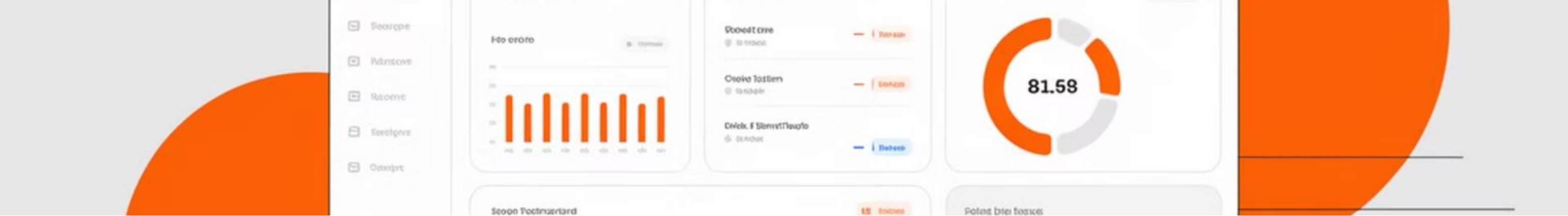


Proses

- Pilih 3–5 evaluator dengan pengetahuan tentang prinsip usability
- Tentukan heuristik yang akan digunakan (biasanya 10 prinsip Nielsen)
- Setiap evaluator memeriksa antarmuka secara independen
- Evaluator mencatat masalah dan mengaitkannya dengan heuristik
- Gabungkan temuan dan prioritaskan berdasarkan tingkat keparahan

Kelebihan dan Keterbatasan

- Kelebihan: Cepat, hemat biaya, dapat dilakukan pada tahap awal
- Keterbatasan: Tidak melibatkan pengguna nyata, bergantung pada keahlian evaluator, mungkin melewatkan masalah kontekstual



Analisis Metrik dalam Evaluasi UI

98%

Tingkat Penyelesaian Tugas

Persentase pengguna yang berhasil menyelesaikan tugas tertentu. Metrik dasar yang mengukur efektivitas antarmuka.

1:42

Waktu Penyelesaian

Waktu rata-rata yang dibutuhkan pengguna untuk menyelesaikan tugas. Mengukur efisiensi antarmuka.

2.3

Tingkat Kesalahan

Jumlah rata-rata kesalahan yang dibuat pengguna saat menyelesaikan tugas. Mengukur kejelasan dan prediktabilitas antarmuka.

87

SUS Score

System Usability Scale, skor 0-100 yang mengukur persepsi kegunaan secara keseluruhan berdasarkan kuesioner standar 10 pertanyaan.

Metrik kuantitatif memberikan data objektif tentang performa antarmuka dan dapat digunakan untuk mengukur perbaikan dari waktu ke waktu. Metrik ini paling efektif ketika dikombinasikan dengan data kualitatif dari wawancara dan observasi untuk memberikan konteks dan pemahaman yang lebih dalam.