

Nama: Gunawan

Nim: H0724505

1. PERBEDAAN UX dan UI

- UX (User Experience)

UX adalah pengalaman pengguna saat menggunakan aplikasi. Fokusnya pada kenyamanan, kemudahan, dan kepuasan pengguna.

UX menjawab pertanyaan:

- Apakah Aplikasi mudah digunakan?
- Apakah Alurnya jelas?
- Apakah pengguna cepat mencapai tujuannya?

- UI (User Interface)

UI adalah Tampilan Visual dari aplikasi yang dilihat pengguna

UI menjawab pertanyaan:

- Apakah Tampilannya menarik?
- Apakah warna dan font enak dilihat?
- Apakah Tombol Terlihat Jelas?

Jadi perbedaannya UX lebih ke cara kerja dan pengalaman. Sedangkan UI lebih ketampilan dan desain visual.

2. penjelasan Gambar UX dan UI

Pada gambar terlihat perbandingan UX dan UI dengan ilustrasi otak kiri (UX) dan otak kanan (UI).

Bagian UX (User Experience)

a. Interaction Design → Mendesain

Bagaimana pengguna berinteraksi dengan aplikasi.

Contoh: Tombol diklik → pindah Halaman.

b. User Research → melakukan riset tentang kebutuhan pengguna

Contoh: SURVEY, wawancara user

c. Scenarios → Membuat skenario penggunaan aplikasi

Contoh: jika user ingin beli barang,

Langkahnya Bagaimana?"

d. Information Architect → menyusun

Struktur informasi dalam aplikasi

Contoh: Menu, kategori, navigasi

e. Wireframes dan prototype → membuat gambaran awal tampilan aplikasi sebelum jadi.

Wireframe = sketsa kasar

Prototype = versi percobaan yang bisa diuji.

Bagian UI (user interface)

- a. Visual Design → Desain tampilan keseluruhan agar menarik.
- B. Colors → pemilihan warna agar nyaman dilihat dan sesuai branding
- c. Graphic design → ikon, ilustrasi, gambar dalam aplikasi.
- d. Typography → pemilihan jenis Huruf (font), ukuran, dan jarak tulisan
- e. Layouts → Tata letak elemen dalam layar (posisi tombol gambar, teks).

3. perbedaan Native dan Cross-platform

✓ Native

Native adalah aplikasi yang dibuat khusus untuk satu sistem operasi

Contoh:

Android → Kotlin/Java iOS → Swift

Kelebihan:

✓ Performa lebih cepat

✓ Lebih stabil

✓ Akses fitur perangkat lebih maksimal

Kekurangan:

X Harus buat 2 aplikasi jika ingin Android & iOS

X Biaya dan waktu lebih banyak

✓ Cross-Platform

Cross platform adalah aplikasi yang bisa berjalan di banyak sistem operasi dengan satu kode

Contoh:

Flutter

React Native

Xamarin

Kelebihan:

✓ Satu kode untuk Android dan iOS

✓ Hemat waktu dan biaya

Kekurangan:

X. Kadang performa tdk secepat Native

X. Beberapa fitur device terbatas.

4. Kolaborasi Tim (Git) dan Problem Solving

Pertanyaan 1:

Mengapa Git penting dalam kolaborasi tim?

Jawaban:

Karena Git membantu tim mengelola versi kode, membagi tugas menggunakan branch dan menggabungkan hasil kerja tanpa saling menimpa file.

Pertanyaan 2:

Apa langkah dasar dalam menyelesaikan masalah (problem solving) saat terjadi bug?

Jawaban

identifikasi masalah, cari penyebabnya, perbaiki kode, lakukan testing ulang utk memastikan bug sudah selesai.