

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Interaksi Manusia dan Komputer

Interaksi manusia dan komputer atau *Human Computer Interaction* (HCI) adalah suatu disiplin ilmu yang mengkaji komunikasi atau interaksi diantara pengguna dengan sistem komputer maupun sistem yang banyak digunakan dalam kehidupan sehari-hari, misalnya kendaraan, peralatan rumah tangga dan sebagainya. Peran utama dari HCI adalah untuk menghasilkan sebuah sistem yang mudah digunakan, aman, efektif, dan efisien. Model interaksi antara manusia dengan sistem melibatkan tiga komponen, yaitu pengguna, interaksi, dan sistem itu sendiri. Adapun perbedaan antara manusia dan komputer dapat dilihat pada Tabel 2.1 sebagai berikut :

**Tabel 2.1 Perbedaan Manusia dan Komputer**

Manusia	Komputer
Fleksibel dan mampu beradaptasi	Tidak fleksibel/tidak mampu beradaptasi
Dapat belajar bagaimana bekerja di lingkungan yang baru	Input harus dalam format yang jelas dan output harus didefinisikan sebelumnya
Estimasi, intuisi, kreativitas	Tidak dapat belajar
Pengenalan pola	Dapat didesain ulang
Pengetahuan dunia	Kalkulasi akurat, konsisten
Kesalahan manusiawi	Pengolahan data
Sering melakukan kesalahan	Bebas dari kesalahan

### 2.2 *User Interface* (UI)

Menurut Wilbert O. Galitz, *user interface* adalah bagian dari komputer dan perangkat lunak yang dapat dilihat, didengar, disentuh, diajak bicara, dan yang dapat dimengerti secara langsung oleh manusia [2]. Dapat dikatakan *user interface*

itu sebagai teknik dan mekanisme dari tampilan antarmuka untuk berinteraksi dengan pengguna. Berdasarkan pernyataan tersebut, maka dapat dikatakan bahwa *user interface* adalah bagian dari komputer dan perangkat lunak yang mengatur tampilan antarmuka untuk pengguna dan memfasilitasi interaksi yang menyenangkan antara pengguna dengan sistem. *User interface* (UI) juga bisa diartikan sebagai hasil akhir dari *user experience* (UX) yang dapat dilihat.

Perancangan antarmuka memiliki prinsip-prinsip yang harus dipenuhi dalam pembangunannya, terdapat 17 prinsip umum yang melatarbelakangi perancangan *user interface* [4], yaitu sebagai berikut :

1. Kompatibilitas Pengguna (*User Compatibility*)

Antarmuka pengguna yang akan dibangun harus memenuhi prinsip *user compability*, karena semua pengguna adalah tidak sama dan semua pengguna tidak seperti pengembang. Para *designer* harus paham tentang psikologi dasar pengguna tersebut.

2. Kompatibilitas Produk (*Product Compatibility*)

Kompabilitas antar produk harus diperhatikan dan dipertahankan.

3. Kompatibilitas Tugas (*Task Compability*)

Struktur dan alur sistem harus sesuai dan mendukung tugas pengguna.

4. Kompatibilitas Alur Kerja (*Workflow Compatibility*)

Sistem harus diorganisasikan dengan baik sehingga dapat memfasilitasi transisi antar tugas pengguna.

5. Konsistensi (*Consistency*)

Konsistensi membuat pengguna berfikir dengan menganalogikan dan memprediksi bagaimana melakukan sesuatu yang belum pernah dilakukan sebelumnya.

6. Keakraban (*Familiarity*)

Konsep, terminologi, pengaturannya di antarmuka harus yang dipahami pengguna dengan baik.

7. Kesederhanaan (*Simplicity*)

Kesalahan umum yang terjadi pada perancangan antarmuka adalah berusaha untuk menyediakan semua fungsionalitas.

8. Manipulasi langsung (*Direct Manipulation*)

Pengguna secara langsung dapat melihat aksinya pada objek yang terlihat.

9. Kontrol (*Control*)

Dapat membuat frustrasi dan demolarisasi bagi pengguna, jika merasa dikontrol oleh mesin. Contohnya penggunaan tombol next yang terlalu banyak.

10. *What You See Is What You Get (WYSIWYG)*

Adanya korespondensi satu ke satu antara informasi di layar dengan informasi di *printerd-output* atau *file*.

11. Keluwesan (*Flexibility*)

Mengijinkan semakin banyak kontrol pengguna dan mengakomodir keterampilan pengguna yang bervariasi.

12. Tanggap (*Responsiveness*)

Komputer harus selalu merespon dengan segera setiap *input* dari pengguna.

13. Teknologi tak terlihat (*Invisible Technology*)

Pengguna sebaiknya mengetahui sedikit mungkin detil teknis bagaimana sistem diimplementasikan. Contohnya pengguna tidak perlu tau atau bingung mengenai proses *download*.

14. Kekokohan (*Robustness*)

Sistem sebaiknya mentolelir kesalahan manusia yang umum dan tidak dapat dihindari, seperti lupa. *Crash system* harus diminimalisir, menyediakan *recovery* yang mudah dipahami jika terjadi *crash*.

15. Perlindungan (*Protection*)

Pengguna seharusnya memproteksi dari hasil-hasil yang menyebabkan “bencana” karena kesalahan umum manusia (pelupa). Contohnya pengguna lupa *password* maka seharusnya sistem memproteksinya dengan fasilitas *forgot password* sehingga kemungkinan kecil orang lain tidak dapat mengakses.

16. Mudah dipelajari (*Easy of Learning*)

Sistem mudah dipelajari bagi pengguna yang masih pemula.

17. Mudah digunakan (*Easy of Use*)

Sistem mudah digunakan bagi para pengguna yang sudah mahir.

### 2.3 *User Experience* (UX)

Menurut definisi dari ISO 9241-210, *user experience* adalah persepsi atau pengalaman seseorang dan responnya dari penggunaan sebuah produk, sistem, atau jasa. *User experience* menilai seberapa kepuasan dan kenyamanan seseorang terhadap sebuah produk, sistem, dan jasa. Menurut Jakob Nielsen, *user experience* mencakup seluruh aspek interaksi terhadap pengguna dengan perusahaan, layanan, dan produk-produknya [5].

Berdasarkan kedua pernyataan diatas, dapat disimpulkan bahwa *user experience* adalah pengalaman yang dirasakan seseorang terhadap penggunaan produk yang menilai tentang tingkat kemudahan dan kenyamanan terhadap fungsionalitas dari sebuah perangkat lunak.

#### 2.3.1 Model Jesse James Garrett

Jesse James Garrett dalam bukunya “*The Elements of User Experience: User-Centered Design for the Web and Beyond*” menciptakan sebuah model untuk menjelaskan elemen dari *User Experience*. Garrett membaginya menjadi 5 elemen, yaitu *strategy plane*, *scope plane*, *structure plane*, *skeleton plane*, dan *surface plane*.

*Strategy plane* adalah elemen yang berada pada level pertama. Pada elemen ini untuk mengetahui apa kebutuhan dari *user* (*user needs*) dan tujuan dari dibuatnya produk (*product objectives*). Intinya pada tahap ini akan dilakukan penelitian untuk menyelaraskan *user goals* dan *business goals*. *User needs* adalah *goals* dari produk yang berasal dari luar organisasi atau perusahaan, yaitu *user* yang akan menggunakan produk kita. Kita harus mengetahui apa yang *user* inginkan

terhadap produk kita. *Product objectives* adalah *goals* yang berasal dari dalam organisasi atau perusahaan, yaitu berupa *business goals* seperti target revenue perusahaan, target conversion rate, dan lain-lain.

Scope plane adalah elemen yang berada pada level kedua. Elemen ini terdiri dari *functional specifications* dan *content requirements*. *Functional specifications* adalah deskripsi dari sekumpulan fitur apa saja yang akan ada di dalam produk, misalkan fitur cart, fitur pemesanan, fitur pembayaran, dan lain-lain. *Content requirements* adalah deskripsi dari sekumpulan elemen konten apa saja yang akan ada di dalam produk, misalkan gambar, video, ilustrasi, maps, button, icon, dan lain-lain.

*Structure plane* adalah elemen yang berada pada level ketiga. Elemen ini terdiri dari *interaction design* dan *information architecture*. Pada *structure plane* ini, elemen konten dan fitur yang sudah ditentukan pada *scope plane* akan diatur lebih lanjut. *Interaction design* adalah kita mendefinisikan bagaimana sistem merespon terhadap apa yang *user* lakukan. Pada intinya kita akan membuat sebuah *user flow/task flow* yang akan dilalui *user* dari awal menggunakan produk sampai akhir (tujuan pengguna terpenuhi), misalkan pada *e-commerce*: alur yang dilalui oleh *user* mulai dari home sampai proses checkout dan *user* berhasil melakukan pembayaran. Selain itu juga kita membuat bagaimana interaksi pada setiap *flow*, apakah *user* harus melakukan *swipe*, *scroll*, dan lain-lain. *Information architecture* adalah pengaturan konten elemen untuk membantu *user* dalam memahami informasi yang ditampilkan pada produk. Konten-konten tersebut akan dikelompokkan menjadi beberapa kategori atau menu.

*Skeleton plane* adalah elemen yang berada pada level keempat. Elemen ini terdiri dari *information design*, *interface design*, dan *navigation design*. *Information design* adalah penyajian dari informasi untuk memudahkan pemahaman. Informasi yang sudah dikelompokkan dalam *information architecture* akan dituangkan ke dalam sebuah *screen*. *Interface design* adalah pengaturan elemen *interface* sehingga memungkinkan *user* untuk mulai bisa berinteraksi dengan fungsi atau fitur dari produk. *Interface design* ini berbentuk *wireframe*. *Navigation design* adalah

sekumpulan elemen *screen* yang memungkinkan *user* berpindah melalui *information architecture*. Implementasi dari *skeleton plane* adalah pembuatan *wireframe* yang akan memberikan gambaran secara kasar dari antarmuka dari produk yang nantinya akan digunakan oleh *user*.

*Surface plane* adalah elemen yang berada pada level kelima. Elemen ini terdiri dari *sensory experience* yang diwujudkan dalam bentuk *high-fidelity design*. *High-fidelity design* merupakan *design* yang sudah sangat detail dari aspek warna, font, resolusi, dan lain-lain.

## 2.4 Usability Testing

*Usability* menurut *International Standards Organization (ISO)*, yaitu ISO 9241-11 yang mengatur mengenai standar ergonomi untuk pekerjaan kantor dengan *visual display terminal*, *usability testing* adalah sejauh mana sebuah produk bisa digunakan oleh pengguna tertentu untuk tujuan tertentu dengan efektif, efisien, dan kepuasan dalam konteks penggunaan tertentu. Efektivitas mengacu pada keakuratan dan kelengkapan pengguna untuk mencapai tujuan tertentu. Efisiensi berkaitan dengan sumber daya yang dikeluarkan dalam hubungannya dengan akurasi dan kelengkapan pengguna untuk mencapai tujuan. Kepuasan mengacu pada kenyamanan dan penerimaan penggunaan.

Sedangkan menurut Jakob Nielsen, *usability* adalah set atribut kualitas yang menilai seberapa mudah antarmuka pengguna digunakan. Ada lima syarat yang harus dipenuhi agar suatu *website* mencapai tingkat *usability* yang ideal [6], yaitu :

1. *Learnability*

Menjelaskan tingkat kemudahan pengguna dalam memahami *website* untuk memenuhi tugas-tugas dasar saat pertama kali menggunakannya.

2. *Efficiency*

Menjelaskan tingkat kecepatan pengguna dalam menyelesaikan tugas-tugas setelah mempelajari *website*.

3. *Memorability*

Menjelaskan ukuran bagi pengguna, sehingga *website* akan mudah diingat. Bila *website* banyak dilakukan perubahan, maka pengunjung akan memerlukan waktu untuk menyesuaikan dan mempelajarinya kembali.

#### 4. *Errors*

Menjelaskan berapa jumlah kesalahan yang dibuat oleh pengguna, dan bagaimana cara pengguna memperbaiki kesalahan dengan mudah. Menghindari adanya *link* yang tidak berfungsi (*broken link*) atau halaman *website* yang masih dalam proses pembuatan (*under construction*).

#### 5. *Satisfaction*

Menjelaskan tingkat kepuasan pengguna dalam menggunakan *website*. Pengunjung menginginkan *website* dapat dengan mudah digunakan dan dipelajari. Selain itu mereka ingin bisa menemukan apa yang dicari dengan cepat, mengetahui di mana mereka berada dan bisa pergi ke mana saja dalam sebuah *website*.

Berdasarkan pernyataan di atas maka dapat disimpulkan bahwa pengertian *usability testing* adalah sebuah tolak ukur yang menilai tentang penggunaan sebuah produk oleh pengguna berdasarkan lima komponen kualitas, yaitu *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*. *Usability* merupakan unsur penting dalam *Human Computer Interaction* (HCI). *Usability* yang baik menggambarkan sistem tersebut mudah digunakan, tidak menimbulkan kekhawatiran bagi pengguna, serta mudah untuk dipelajari dan diingat. Untuk mengetahui kondisi produk saat ini, evaluasi produk dilakukan dengan menggunakan *usability testing*. *Usability testing* adalah teknik untuk mengetahui sejauh mana sebuah produk bisa digunakan dengan baik oleh pengguna, seberapa efisien dan efektif produk tersebut bisa membantu pengguna mencapai tujuan tertentu, dan apakah pengguna merasa puas dengan produk yang digunakan.

## 2.5 Metode *User Centered Design* (UCD)

*User Centered Design* (UCD) merupakan paradigma baru dalam pengembangan sistem berbasis *website*. *User Centered Design* (UCD) juga sering disebut sebagai *Human Centered Design* (HCD). Menurut ISO 13407 (1999), *Human Centered Design* adalah sebuah pendekatan pengembangan sistem interaktif yang secara khusus fokus untuk membuat sebuah sistem berguna. Definisi lainnya menurut [7], *User Centered Design* adalah sebuah proses desain *interface* (antarmuka) yang fokus terhadap tujuan kegunaan, karakteristik pengguna, lingkungan, tugas, dan alur kerja di dalam desainnya [8][9]. UCD adalah sebuah proses *iterative* (berulang-ulang), dimana desain dan evaluasi dibangun dari langkah awal hingga implementasi secara terus menerus.

### 2.5.1 Prinsip-prinsip dalam metode UCD

Berikut ini merupakan prinsip-prinsip yang harus diperhatikan dalam metode UCD :

1. Fokus pada pengguna

Perancangan harus melibatkan pengguna atau calon pengguna secara langsung melalui wawancara dan observasi. Tujuannya adalah untuk memahami kognisi, karakter, dan sikap pengguna serta karakteristik *anthropometric*. Aktivitas utamanya meliputi pengambilan data, analisis, dan integrasi ke dalam informasi perancangan dari pengguna tentang karakteristik tugas, lingkungan teknis, dan organisasi.

2. Perancangan terintegrasi

Perancangan harus mencakup antarmuka pengguna, sistem bantuan, dukungan teknis serta prosedur instalasi dan konfigurasinya.

3. Pengujian terhadap pengguna

Pendekatan yang baik dalam perancangan sistem yang berpusat pada pengguna adalah secara empiris dibutuhkan observasi tentang perilaku pengguna, evaluasi *feedback* yang cermat, wawasan pemecahan terhadap masalah yang ada, dan motivasi yang kuat untuk mengubah rancangan.



#### 4. Perancangan interaktif

Sistem yang sedang dikembangkan harus didefinisikan, dirancang, dan dilakukan *test* berulang kali. Berdasarkan hasil *test* kelakuan dari fungsi, antarmuka, sistem bantuan, dokumentasi pengguna, dan pendekatan pelatihannya.

### 2.5.2 Proses dalam metode UCD

Berikut ini adalah beberapa proses dalam metode UCD :

#### 1. Memahami dan Menentukan Konteks Pengguna

Mengidentifikasi orang yang akan menggunakan produk. Ini akan menjelaskan untuk apa dan dalam kondisi seperti apa mereka akan menggunakan produk.

#### 2. Menentukan Kebutuhan Pengguna

Mengidentifikasi kebutuhan pengguna.

#### 3. Menghasilkan Solusi Perancangan

Membangun desain sebagai solusi dari produk yang sedang dianalisis.

#### 4. Evaluasi Hasil Perancangan

Melakukan evaluasi terhadap desain yang dilakukan apakah tujuan pengguna telah tercapai.

## 2.6 HTML

Arief Ramadhan [10] mendefinisikan bahwa “HTML adalah singkatan dari Hypertext Markup Language. HTML digunakan untuk membuat halaman web. Sebuah file dokumen yang ditulis dalam format HTML akan dibaca dan diterjemahkan oleh web browser (misal Internet Explorer) untuk kemudian disajikan dalam bentuk web”. Selain itu, bahasa HTML menjadi inti bagi bahasa pemrograman CSS, JavaScript, dan PHP yang mendukung dalam pembangunan *website*. Berikut adalah contoh syntax pada HTML :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

## 2.7 CSS

Arief Ramadhan [10] mengatakan bahwa “Sebuah file CSS terdiri atas berbagai macam aturan style yang dapat digunakan untuk mengatur tampilan web Anda”. Mungkin beberapa orang sudah mengetahui bahwa HTML sebenarnya bisa untuk melakukan modifikasi tampilan tanpa menggunakan CSS, karena di HTML memiliki atribut tersendiri yang sudah disediakan untuk memodifikasi tampilan web. Namun, menggunakan HTML untuk memodifikasi tampilan *website* dirasa kurang tepat karena pada definisinya sangat berbeda antara HTML dan CSS. Maka dari itu, CSS menjadi yang diutamakan untuk membantu *website* dalam memodifikasi tampilannya. Berikut adalah contoh syntax pada CSS :

```
body{
```

```

background: #fff;
font-family: sans-serif;
}
.myclass{
margin: 10px;
padding: 10px;
float: left;
}

```

## 2.8 Javascript

Andi Suntoyo [11] mendefinisikan bahwa “Javascript adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla FireFox, Netscape, dan Opera”. Dalam membangun *website*, Javascript lebih difokuskan untuk membantu proses penanganan pada data yang sangat kompleks dengan tujuan data dapat diakses oleh *user* secara cepat. Berikut adalah contoh syntax pada Javascript :

```

<script>
var x = 5;
var y = 2;
var z = x + y;
document.getElementById(“demo”).innerHTML = z;
</script>

```

## 2.9 XAMPP

XAMPP adalah software web server apache yang di dalamnya tertanam server MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis. XAMPP sendiri mendukung dua sistem operasi, yaitu Windows dan Linux. Untuk Linux dalam proses installasinya menggunakan command line, sedangkan untuk Windows dalam proses installasinya

menggunakan *User Interface Graphic* (GUI) sehingga lebih mudah dalam penggunaan XAMPP pada Windows dibandingkan dengan Linux.