

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Prihati, Mustafi, Suharto (2011) membuat sebuah jurnal yang berjudul “Penerapan Model *Human Computer Interaction* (HCI) dalam Analisis Sistem Informasi”. Dalam jurnal tersebut dijelaskan bahwa sering terjadi salah pemahaman pengguna terhadap *software* yang ada, sehingga meskipun pengguna cukup puas dengan hasil yang didapat dari *software* tersebut pengguna tetap merasa bahwa fasilitas yang diberikan oleh *software* dirasa belum cukup. Dari persoalan yang terjadi dapat dihindari dengan menerapkan konsep HCI yang baik. Dengan menerapkan konsep HCI yang baik maka dapat terciptanya sistem yang memberikan kenyamanan dan kemudahan dalam penggunaan, dengan demikian tujuan penerapan sistem akan dapat dicapai dan tidak akan mengalami kegagalan.

Santi, I, H (2015) melakukan penelitian dengan judul “Tinjauan *Human Computer Interaction* (HCI) Terhadap Aplikasi Penjadwalan Sekolah” yang mengtakan bahwa Aplikasi perangkat lunak dibuat bertujuan untuk membantu memudahkan pengguna dalam menjawab permasalahan yang sedang terjadi. Permasalahan tersebut salah satu nya dapat diselesaikan dengan cara membuat tampilan (*User Interface*) yang baik sehingga *interface* yang dihasilkan dapat dengan mudah diipelajari dan nyaman untuk digunakan (*User Friendly*).

Wijaya, R, P (2015) melakukan penelitian dengan judul “Perancangan Website Berpedoman kepada 8 *Golden Rules of Interface Design*”. Dimana pada penelitian ini latar belakang masalah yang dihadapi website yang ada di Universitas Muhammadiyah Yogyakarta (UMY) masih belum menggunakan aturan khusus antarmuka, sehingga menyebabkan pengunjung website susah mencari informasi yang dicari, pengunjung tidak betah dan tidak mengerti mengenai cara penggunaannya. Dengan latar belakang masalah peneliti mengimplementasikan pengaturan interaksi dan komputer 8 *golden rules of interface design*. Hasil dari implementasi dengan menggunakan aturan 8 *golden rules of interface design* menghasilkan *website* yang baik tampilannya, penempatan artikel dan isi *website*

yang baik, serta pengunjung yang memiliki rasa puas dan nyaman ketika mengakses *website*.

Berdasarkan penelitian-penelitian tersebut, dapat dilihat bahwa penerapan HCI menggunakan metode 8 *golden rules* dalam sebuah aplikasi sangat dibutuhkan agar tampilan yang di buat dapat digunakan dengan mudah dan nyaman (*user friendly*). Dengan dilakukannya pengujian *user friendly* pada pengguna, penulis dapat mengetahui kelemahan yang ada pada sistem.

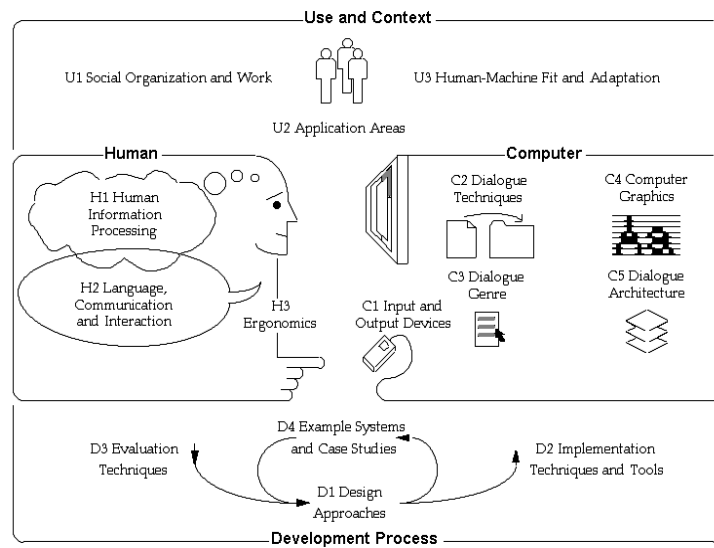
2.2 Landasan Teori

2.2.1 Human Computer Interaction (HCI)

HCI yaitu studi yang mempelajari hubungan interaksi antara manusia, komputer dan penugasan. Prinsipnya bagaimana manusia dan komputer dapat secara interaktif menyelesaikan penugasan dan bagaimana sistem yang interaktif tersebut dapat dibuat. (Saifulloh, 2015).

Adapun pemahan lain terhadap HCI dimana ketika membangun sebuah sistem informasi, seorang desainer atau pengembang sistem harus “memperhatikan faktor interaksi manusia dan komputer karena sistem informasi yang dibuat oleh manusia dan tujuannya untuk manusia”. (Prihati, 2011)

Salah satu bahasan terpenting dalam bidang HCI adalah antar muka pengguna (*user interace*), yang merupakan bagian sistem yang dikendalikan oleh pengguna untuk mencapai dan melaksanakan fungsi-fungsi suatu sistem. Pengguna berhubungan dengan sistem melalui antar muka pengguna. Peran antar suatu sistem sangatlah penting. Oleh karenanya bentuk dan pembangunan antar muka pengguna perlu dilihat sebagai salah satu proses utama dalam keseluruhan pembangunan suatu sistem. (Sudarmawan ,2007). Interaksi manusia dan komputer digambarkan dalam gambar berikut:



Gambar 2.1 Ruang Lingkup Human Computer Interaction (HCI)

Menurut Sudarmawan (2007) pada gambar 2.1 menunjukkan ruang lingkup *Human Computer Interaction* (HCI) sebagai berikut:

- a. **Manusia**
Ruang lingkup manusia meliputi ergonomi, antropologi, psikologi, latar belakang, kemampuan mengelola informasi, dan lain-lain.
- b. **Komputer**
Ruang lingkup komputer meliputi hardware, software, rekayasa perangkat lunak, sistem cerdas, sistem informasi, dan lain-lain.
- c. **Interaksi**
Jembatan interaksi manusia dan komputer adalah *user interface*. *User interface* berkaitan erat dengan desain tampilan, navigasi, pelabelan, menu, dan lain-lain.
- d. **Aktivitas**
Aktivitas menunjukkan bagaimana *user* mengerjakan tugas, apa *goal* atau tujuan yang ingin dicapai dari suatu aktivitas, kemudahan melakukan aktivitas, dan lain-lain.
- e. **Lingkungan Kerja**

2.2.2 Prinsip 8 *Golden Rules of Interface Design*

Dalam bukunya yang berjudul "*Designing The User Interface*", Ben Shneiderman (1986) mengemukakan bahwa *8 golden rules principle* merupakan sebuah aturan yang menetapkan *interface* agar dapat menyempurnakan implementasi dalam hal berinteraksi pada manusia. Beberapa poin-poin yang terdapat dalam *8 golden rules* diantaranya sebagai berikut:

1. *Strive for consistency*

Konsisten terhadap jenis, ukuran, dan spasi *font* merupakan hal yang terpenting untuk membuat sebuah aplikasi yang mempunyai nilai yang baik. Tampilan aplikasi tidak seharusnya rumit, mempunyai warna yang tidak terlalu banyak. *User* biasanya tidak ingin mempelajari hal baru yang membuat waktu *user* terbuang karena menggunakan aplikasi.

2. *Cater to universal usability*

Mengenali kebutuhan *user* yang beragam dan memudahkan dalam melakukan perubahan konten. Mencari perbedaan user pemula, user ahli, rentang usia, keterbatasan, kemampuan fisik, dan perbedaan teknologi yang merupakan panduan dalam merancang *interface*. Menambahkan fitur-fitur untuk *user* ahli seperti *shortcut* untuk *intermediate user* dan untuk *user* pemula membuat tampilan yang kompleks dapat meningkatkan kualitas dari desain *interface*.

3. *Offer informative feedback*

Aplikasi yang baik selalu memberikan timbal balik ketika terjadi sesuatu di dalam aplikasi itu sendiri. Sehingga memudahkan *user* untuk bertindak seperti yang sistem inginkan.

4. *Design dialogs to yield closure*

Aplikasi yang baik juga harus mempunyai suatu perbedaan tampilan yang bisa memberi tahu *user* bahwa kegiatan yang dilakukan sudah selesai. Seperti, ketika *user* sudah melakukan *login* maka akan muncul menu utama, menandakan bahwa proses *login* yang *user* lakukan sudah selesai.

5. *Prevent errors*

Aplikasi yang baik adalah dengan adanya validasi yang dapat mengurangi kesalahan yang akan terjadi pada *user*. Apabila *user* melakukan kesalahan, sistem harus mampu mendeteksi kesalahan yang dilakukan oleh *user*, sehingga mengurangi peluang *user* untuk melakukan kesalahan yang sama.

6. *Permit easy reversal of actions*

Di dalam sebuah aplikasi harus adanya fasilitas untuk *user* bisa mengedit atau menghapus tindakan yang sudah dilakukan sebelumnya. Sehingga mengurangi kekhawatiran *user* untuk melakukan kegiatan pada aplikasi.

7. *Support internal locus of control*

Aplikasi perlu memiliki fasilitas untuk *user* agar dapat dengan bebas bernavigasi dan mengubah informasi akun yang dimilikinya sesuai dengan yang dikehendaki.

8. *Reduce short term memory load*

Keterbatasan manusia dalam mengolah informasi dalam memori jangka pendek membutuhkan antarmuka (*interface*) yang sederhana dan mengurangi pergerakan *window*. Yang dimaksud pergerakan *window* seperti ketika *user* harus mengingat suatu informasi pada satu layar dan informasi tersebut digunakan pada layar lain.

2.2.3 Unified Modelling Language (UML)

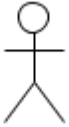


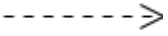
Unified Modelling Language (UML) adalah alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Selain itu merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem.

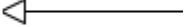





Software yang terkait dengan objek. Penggunaan UML bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem didalam

aplikasi, mendokumentasikan hasil analisa dan desain serta untuk menggambarkan sebuah sistem *software*. Model UML yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case Diagram* dan *Activity Diagram*.

- a. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tabel 2.1 Use Case Diagram




Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .


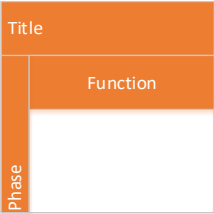
Gambar	Nama	Keterangan
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

- a. *Activity Diagram* adalah menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision

yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses *parallel* yang mungkin terjadi beberapa eksekusi, *Activity diagram* merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *Activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antara *subsistem*) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level secara umum.

Tabel 2.2 *Activity Diagram*

No	Gambar	Nama	Keterangan
1		Initial Node	Bagaimana objek dibentuk atau diawali
2		<i>Activity</i> Final Node	Bagaimana objek dibentuk dan diakhiri
3		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain

No	Gambar	Nama	Keterangan
4		Decision	Digunakan untuk menggambarkan suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu
5		Swimlane	Digunakan untuk pembagian <i>Activity</i> diagram yang menunjukkan siapa yang melakukan aktivitas