

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

Prihati, Mustafi, Suharto (2011), pada penelitian yang berjudul “Penerapan Model *Human Computer Interaction* (HCI) Dalam Analisis Sistem Informasi (Studi Kasus SAS Dikmenti DKI Jakarta)”. Tujuan penelitian ini merancang dan menerapkan model analisa sistem informasi dengan konsep *Human Computer Interaction* (HCI) dengan kriteria *usability* yang digunakan berdasarkan Jacob Neilsen yaitu *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*. Penelitian ini menghasikan kesimpulan dengan menerapkan HCI dapat menghasilkan sistem yang berguna, aman, produktif, efektif, efisien dan fungsional.

Santi, I, H (2015), pada penelitian yang berjudul “Tinjauan *Human Computer Interaction* (HCI) Terhadap Aplikasi Penjadwalan Seklolah”. Tujuan penelitian ini adalah untuk meninjau tampilan (*user Interface*) pada aplikasi penjadwalan guru, untuk mengetahui tampilan pada aplikasi penjadwalan guru sudah sesuai dengan tampilan yang baik atau belum. Penulis mengungkapkan bahwa komponen terpenting dalam mendesain suatu perangkat lunak yaitu *interface* karena *interface* merupakan jembatan antara pengembang dan pengguna. Dengan melihat beberapa aspek yang memenuhi kriteria tampilan yang baik, sehingga *interface* yang dihasilkan dapat dengan mudah dipelajari.

Yamani, H, Z (2017), pada penelitian yang berjudul “Perancangan Desain *User Interface* Pada Aplikasi *Job Matching Profile* menggunakan Prinsip 8 *Golden Rules Of Interface Design*”. Tujuan penelitian ini adalah untuk merancang interface pada aplikasi “*Job Matching Profile*” agar dapat digunakan dengan mudah oleh penggunanya. Desain dan *interface* yang sesuai menjadi penentu suatu aplikasi untuk mudah digunakan, hal ini disebabkan jika

tampilan tidak efektif maka pengguna akan kesulitan menggunakan aplikasi tersebut. Aplikasi “*Job Matching Profile*” didesign dengan menerapkan aturan 8 *golden rules of interface design* untuk menghasilkan tampilan aplikasi yang mudah digunakan.

Berdasarkan Penelitian diatas, bahwa dengan menerapkan HCI menggunakan metode 8 *golden rules* pada sebuah aplikasi atau *website* akan menghasilkan tampilan yang mudah dipelajari serta mudah digunakan. Karena saat mendesain sebuah perangkat lunak dibutuhkan pertimbangan agar tampilan yang dihasilkan mudah untuk di operasikan oleh penggunanya.

## **2.2 Landasan Teori**

### **2.2.1 Human Computer Interaction (HCI)**

HCI (*Human Computer Interaction*) adalah disiplin ilmu yang mengkaji tentang komunikasi atau interaksi antar pengguna dan sistem. Perannya yaitu untuk menghasilkan sistem yang berguna, aman, produktif, efektif, efisien dan fungsional, (Prihati, 2011).

Adapun pendapat lainnya, menurut Hewett (1992), HCI merupakan disiplin ilmu yang terkait dengan desain, evaluasi dan implementasi dari sistem komputer interaktif untuk manusia dengan mempelajari kejadian utama yang ada disekelilingnya

Salah satu bahasan terpenting dalam HCI adalah *user interface* (antarmuka pengguna) merupakan bagian dari sistem yang dikendalikan oleh pengguna untuk melaksanakan fungsi dari suatu sistem pada *user interface*. Peran antar suatu sistem amatlah penting karena pengguna yang melaksanakan fungsi berhubungan dengan sistem melalui *user interface*. Oleh karena itu yang harus diperhatikan dalam proses pembangunan suatu sistem yaitu bentuk keseluruhan dari sistem tersebut, (Sudarmawan,2007).

### 2.2.2 Prinsip 8 Golden Rules Of Interface Design

Pada buku Ben Shneiderman (1986), yang berjudul *Designing the User Interface*, Ben Shneiderman mengemukakan bahwa kumpulan dari prinsip dari 8 *golden rules* diturunkan berdasarkan pengalaman dari sebagian besar penyempurnaan implementasi sistem dan interaksi pada manusia. Aturan 8 *golden rules* merupakan dasar dari medesign sebuah *website* atau aplikasi. Berikut aturan dari 8 *golden rules* oleh Ben Shneiderman:

1. ***Strive for consistency.***

Mendesain sebuah aplikasi atau *website* perlu memperhatikan konsistensi dari tampilan *interface*. Mendesain tampilan *website* perlu memperhatikan bagian-bagian dari tampilan tersebut seperti *font*, warna, ukuran, tata letak, dan lainnya perlu konsisten. Jika seorang pengguna sedang berada di suatu halaman dan *pengguna* tersebut akan ganti ke halaman selanjutnya maka tampilan dari halaman awal dan setelahnya sama atau masih berhubungan, agar *pengguna* merasa nyaman berada di halaman *website* tersebut.

2. ***Cater Universal Usability.***

Mendesain aplikasi atau *website* juga perlu memperhatikan kegunaan secara universalnya seperti kegunaan pada kebutuhan pengguna dan juga perlu memperhatikan pengguna yang sudah bisa menggunakan *website* dan yang belum terbiasa. Seperti menambahkan penjelasan pada tampilan untuk *pengguna* pemula atau menambahkan *shortcut* dengan fungsi tertentu untuk yang pengguna yang sudah biasa menggunakan *website* karena preferensi antarmuka pengguna yang sudah biasa menggunakan *website* dengan yang belum terbiasa berbeda dan juga perlu memperhatikan factor seperti umur, kebudayaan, bahasa dan lainnya.

### 3. ***Offer Informative Feedback.***

Pada sebuah aplikasi atau *website* perlu memiliki respon dari *event* yang dilakukan pengguna saat menjalankan aplikasi atau berada dalam halaman *website*. Sebagai contoh jika pengguna menyimpan data ke *database*, maka akan muncul informasi pada tampilan aplikasi atau *website* bahwa ada pemberitahuan yang pengguna lakukan agar pengguna tau jika datanya sudah disimpan.

### 4. ***Design Dialogs to Yield Closure.***

Aplikasi atau *website* memiliki urutan aksi seperti awal, tengah, dan akhir. Seperti rangkaian urutan saat pengguna ingin melakukan suatu aksi pada aplikasi atau *website* dengan munculnya informasi bahwa urutan pengguna dalam menyelesaikan aksinya tersebut sudah selesai. Seperti saat pengguna akan menginput data dan melihat inputan data.

### 5. ***Prevent Errors.***

Untuk mendesain tampilan yang baik, sebisa mungkin membuat tampilan antar muka yang dapat menghindarkan pengguna kesalahan saat menjalankan proses pada aplikasi atau *website* dengan memberikan informasi atau validasi pada aplikasi atau *website* kepada pengguna agar pengguna dapat memahami cara menjalankan suatu proses atau aksi dengan memberikan petunjuk yang disesuaikan dengan format aplikasi atau halaman *website*. Seperti pada halaman *website* ada data inputan dan memberikan informasi jika ada data yang belum diisi, bisa juga di fokuskan pada *caption feedback* untuk memberi pemberitahuan.

### 6. ***Permit Easy Reversal of Action.***

Saat menjalankan proses pada sebuah aplikasi atau *website*, aksi yang sudah dilakukan sebisa mungkin dapat diubah atau di manipulasi. Seperti saat pengguna akan memasukkan data pada halaman formulir dan pengguna ingin membatalkan pengisian formulir, maka harus ada aksi

yang memiliki fungsi untuk membatalkan penyimpanan data atau mengedit data.

7. ***Support Internal Locus of Control.***

Suatu aplikasi atau *website* yang baik perlu memberikan kemudahan untuk pengguna dalam menavigasikan aplikasi atau *website* sesuai dengan keinginan mereka, seperti mengganti informasi sesuai dengan keinginan pengguna.

8. ***Reduce Short-Term Memory Load.***

Aplikasi atau *website* yang baik perlu menghindari tampilan yang membuat pengguna harus mengingat informasi yang diinputkan sehingga harus mengingat kembali aksi yang dilakukan sebelumnya karena keterbatasan manusia dalam ingatan jaka pendek (*short term memory*). Tampilan yang baik perlu menampilkan data yang sudah diinputkan sebelumnya oleh pengguna, sehingga pengguna tidak perlu mengingat data yang dimasukan.

### 2.2.3 Unified Modelling Language (UML)

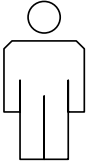
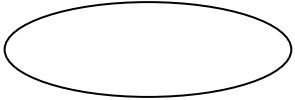

UML atau *Unified Modelling Language* merupakan konsep visualisasi dan desain hasil analisa dalam memodelkan sistem. Terdapat banyak diagram yang dapat digunakan untuk menganalisa dan merancang sistem pada UML. Diantaranya yaitu struktur diagram, *behavior* diagram, dan *interaction* diagram. Masing-masing kategori menjelaskan arsitektur sistem dan Saling terintergrasi, (Gushelmi, Deded Ramad Kamda, 2012).





Dalam merancang sebuah *software* perlu mengidentifikasi bagian-bagian dan lingkup dari sistem dengan mendokumentasikan hasil analisa melalui gambaran sebuah diagram. *Behavior* diagram merupakan diagram yang menggambarkan fungsi metode pada sebuah sistem atau *business* proses.


Pada penelitian ini penulis akan menggunakan UML *behavior* diagram yaitu *Use Case* dan *Activity Diagram*.

- a. *Use Case* diagram medeskripsikan tentang suatu sistem dan interaksinya dengan pengguna, dimana pada *Use Case* diagram divisualisasikan terhadap tindakan aktor yang dilakukan untuk menjalankan sistem, seperti aktor siapa saja yang berhak untuk menggunakan fungsi tersebut dan apa saja yang dapat aktor lakukan pada sistem dilihat pada tabel 2.1.

**Tabel 2.1 Use Case Diagram**





Gambar	Nama	Keterangan
	Aktor	Peran spesifik dari pengguna saat berinteraksi dengan <i>use case</i> .
	<i>Use Case</i>	Urutan aksi dari sistem yang menunjukkan suatu hasil yang terstruktur bagi pengguna.
	<i>Subsystem</i>	Menspesifikan bagian <i>use case</i> sebagai sistem secara terbatas.

Gambar	Nama	Keterangan
	<i>Association</i>	Penghubung antar objek.
	<i>Dependency</i>	Penghubung yang terjadi pada elemen mandiri yang bergantung pada elemen tidak mandiri.
	<i>Generalization</i>	Hubungan anak sebagai perilaku dan struktur data object sebagai induk.
<b>&lt;&lt;include&gt;&gt;→</b>	<i>Include</i>	Hubungan antara yang harus dilakukan aktor pada <i>use case</i> bersifat pilihan yang pada sistem pengguna tidak harus memilih aksi tersebut.
<b>&lt;&lt;extend&gt;&gt;→</b>	<i>Extend</i>	Hubungan antara <i>use case</i> yang mengahuskan pengguna untuk melakukan aksi yang disesuaikan pada perluasan perilaku pada <i>use case</i> .
	<i>Collaboration</i>	Sinergi antar elemen yang terjadi Karen interaksi antar elemen lain.

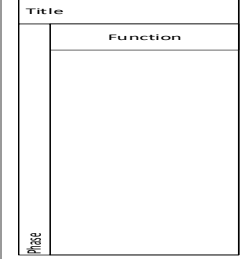
	<i>Note</i>	Elemen yang tersedia sebagai pelengkap.
-----------------------------------------------------------------------------------	-------------	-----------------------------------------

b. *Activity Diagram* merupakan penjabaran dari alur sebuah sistem. Bagaimana masing-masing sistem berawal, pengambilan keputusan pada sistem dan berakhirnya sistem atau dapat dikatakan *Activity diagram* memvisualisasikan proses-proses dan jalur-jalur aktivitas dari level secara umum penjelasan dapat dilihat pada tabel 2.2.

**Tabel 2.2** *Activity Diagram*

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Action</i>	Memvisualisasikan interaksi masing kelas antar muka.
	<i>Decision</i>	Diigunakan untuk pengambilan keputusan yang terjadi pada suatu kelas antarmuka.
	<i>Initial Node</i>	Digunakan untuk memulai suatu objek.
	<i>Final Node</i>	Digunakan untuk mengakhiri objek.



	<i>Swimlane</i>	Digunakan untuk pembagian dari <i>activity diagram</i> .
-----------------------------------------------------------------------------------	-----------------	----------------------------------------------------------

## 2.2.4 Validitas dan Reliabilitas

### a. Validitas

Uji validitas digunakan untuk mengukur valid atau tidak pernyataan yang ada pada kuisioner. Kuisioner dikatakan valid jika pertanyaan diukur dengan alat ukur yang berbeda hasilnya konsisten tidak berubah dan apa yang diukur dapat mengungkapkan sesuatu yang diukur pada pertanyaan kuisioner. Penulis menggunakan validitas item, yaitu dengan melihat skor item dengan skor total dari item tersebut, pengujian ini menggunakan R-hitung atau *Pearson Correlation* dengan menghitung korelasi antara skor perbutir pertanyaan dengan total skor. Karl Pearson pertamakali menjelaskan tentang koefisien dari korelasi dalam pengembangan teori matematikanya mengenai koorelasi. Pearson menyebut teori ini sebagai *metode product moment* atau R-hitung yang merupakan ukuran kekuatan asosiasi liner pada variable, setelahnya metode ini menggunakan namanya, (Rebekic.A, 2015). Ada dua cara untuk melihat valid atau tidak validnya data yaitu dengan melihat nilai signifikannya dan membandingkan nilai R-tabel dan R-hitung, nilai signifikan yang diperoleh pada setiap indicator yaitu 0.05 atau 5%. Jika nilai signifikan lebih besar dari 0,05 maka data itu tidak valid, dan cara yang kedua yaitu dengan membandingkan R-tabel, jika R-hitung lebih besar daripada R-tabel maka data dapat dikatakan valid (Ghozali, 2011). Pengujian validitas item menggunakan perangkat lunak SPSS.

Rumus untuk menghitung *Pearson Correlation* atau R-hitung pada gambar 2.1.

$$r_{xy} = \frac{N \sum XY - \sum X \sum Y}{\sqrt{N \sum X^2 - (\sum X)^2} \cdot \sqrt{N \sum Y^2 - (\sum Y)^2}}$$

**Gambar 2.1** Rumus menghitung *Pearson Correlation*

Dengan keterangan sebagai berikut:

$r_{xy}$  = *Pearson Correlation*

x = Skor item

y = Skor total

N = jumlah responden

Untuk menghitung R-tabel yaitu dengan rumus dibawah ini :

$$r_{tabel} = \frac{t_{tabel}}{\sqrt{df + t_{tabel}^2}}$$

**Gambar 2.2** Rumus menghitung R-tabel

$T_{table} = \alpha, v$

Df = degree of freedom

Df dihitung dengan rumus

$$Df = n - 2$$

Dengan n merupakan jumlah responden.

#### b. Reliabilitas

Reliabilitas adalah suatu ukuran yang menunjukkan sejauh mana hasil pengukuran tetap konsisten jika pengukurannya diukur beberapa kali dalam waktu yang berbeda. Kuisioner dapat dikatakan reliabel jika skor yang didapat pada kuisioner tetap stabil, stabil maksudnya bila skor yang didapat pada suatu waktu dan pada waktu yang berbeda ada kesamaan atau konsisten pada hasilnya (Prasetyo B.W, 2006). Ada teori yang mengatakkn pengujian reliabilitas

diberi ketentuan  $\alpha$  lebih besar daripada 0,6. Nilai reliabilitas dapat dihitung dengan *Cronbach alpha*, rumusnya dapat dilihat pada gambar 2.3:

$$\alpha = \frac{k}{k-1} \left( \frac{s_x^2 - \sum_{j=1}^k s_j^2}{s_x^2} \right)$$

**Gambar 2.3** Rumus *Cronbach Alpha*

Keterangan dari rumus diatas :

$S^2_j$ = varian Skor item

K = banyaknya item yang diujikan

$S^2_x$ = varian skor total item

#### c. Skala Likert

Skala Likert digunakan untuk melihat sejauh mana tingkat kekuatan responden dengan menampilkan skala dari sangat setuju hingga sangat tidak setuju pada pertanyaan dalam lima skala jawaban, yaitu 1 poin untuk sangat tidak setuju, 2 untuk tidak setuju, 3 untuk netral, 4 untuk setuju dan 5 untuk sangat setuju. Penggunaan skalanya dapat di lihat di tabel 2.3, (Sugiyono, 2008)

**Tabel 2.3** Tabel Skala Likert

1	STS	Sangat Tidak Setuju
2	TS	Tidak Setuju
3	N	Netral
4	S	Setuju
5	ST	Sangat Setuju