

BAB 2

LANDASAN TEORI

2.1 Teori yang Relevan

2.1.1 Statistik Deskriptif

Menurut Sugiyono (2004: 142), "Statistik deskriptif adalah statistik yang digunakan untuk menganalisa data dengan cara mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya tanpa bermaksud membuat kesimpulan yang berlaku untuk umum atau generalisasi".

Suprayogi dalam STATISTIKA DESKRIPTIF, mengatakan "Statistika deskriptif (descriptive statistics) berkaitan dengan penerapan metode statistik untuk mengumpulkan, mengolah, menyajikan, dan menganalisis data kuantitatif secara deskriptif".

Jadi, statistik deskriptif merupakan salah satu metode statistika yang memberikan gambaran umum dari data yang telah dikumpulkan dan diolah. Tujuan utama dari statistik deskriptif adalah untuk menyajikan informasi dari data yang didapat dari lapangan. Informasi tersebut bisa berupa tabel, grafik, maupun diagram. Biasanya informasi tersebut akan digunakan untuk dianalisis lebih lanjut dengan metode lainnya.

2.1.2 CHAID

Fielding (dalam O'Muirjeartaigh dan Payne, 1977) mengatakan bahwa CHAID (*Chi-square Automatic Interaction Detection*) adalah salah satu tipe dari metode AID (*Automatic Interaction Detection*) yang digunakan untuk menelusuri struktur keterkaitan antara peubah respon dan peubah penjelas yang masing – masing bertipe kategorik.

Metode ini terutama dikembangkan untuk menelusuri keterkaitan struktural dalam data survei.

Berbeda dengan Fielding, Du toit, et al (1986) berpendapat bahwa metode CHAID merupakan tehnik eksplorasi nonparametrik untuk menganalisis sekumpulan data yang berukuran besar dan cukup efisien untuk menduga peubah – peubah penjelas yang paling signifikan terhadap peubah respon. Interaksi antar peubah juga dapat dideteksi melalui metode ini.

Metode CHAID yang merupakan proses iteratif, terdiri dari dua variabel, yaitu variabel penjelas (variabel independen) dan variabel respon (variabel dependen). Variabel respon merupakan variabel yang menjadi reaksi dari variabel penjelas. Variabel respon dan variabel penjelas boleh lebih dari satu.

Menurut Gallagher (2000), CHAID akan membedakan variabel – variabel penjelasnya menjadi 3 bentuk yang berbeda, yaitu:

1. **Monotonik**, kategori – kategori pada variabel ini dapat dikombinasikan atau digabungkan oleh CHAID hanya jika keduanya berdekatan satu sama lain, yaitu variabel – variabel yang kategorinya mengikuti urutan aslinya (data ordinal)
2. **Bebas**, kategori – kategori pada variabel ini dapat dikombinasikan atau digabungkan walaupun keduanya berdekatan atau tidak satu sama lain (data nominal)
3. **Mengambang (*floating*)**, kategori – kategori pada variabel ini akan diperlakukan seperti monotonik kecuali untuk kategori terakhir (yaitu *missing value*), yang dapat dikombinasikan dengan kategori manapun

Metode CHAID berbeda dengan metode yang lain karena metode CHAID khusus menganalisis data yang berjenis nominal dan atau ordinal. Sesuai dengan namanya, metode CHAID menggunakan statistik uji khi-kuadrat.

Metode CHAID akan membagi data ke dalam kelompok – kelompok melalui beberapa tahapan. Tahapan pertama adalah membagi data menjadi beberapa kelompok berdasarkan satu variabel penjelas yang pengaruhnya paling signifikan terhadap variabel respon. Variabel penjelas yang signifikan ditentukan dengan uji khi-kuadrat. Setelah kita mendapatkan pembagian kelompok – kelompok tersebut, kita periksa kelompok – kelompok tersebut secara terpisah untuk membagi lagi menjadi beberapa kelompok yang lebih kecil berdasarkan variabel penjelas yang lain. Hal tersebut dilakukan sampai tidak ditemukan lagi variabel – variabel penjelas yang signifikan secara statistik.

Menurut Gallagher, CHAID merupakan proses iteratif yang terdiri dari 4 tahap:

- Pemeriksaan tiap variabel penjelas menggunakan uji khi-kuadrat untuk menentukan kategori mana yang nantinya signifikan untuk menunjukkan perbedaan dalam variabel respon; dan mengumpulkan semua kategori yang tidak signifikan
- Penentuan variabel penjelas mana yang paling signifikan, yang terbaik untuk digunakan dalam membedakan variabel respon berdasarkan nilai kesignifikanan hasil uji yang dilakukan
- Pembagian data menggunakan kategori variabel penjelas tersebut dengan peringkat yang paling signifikan

- Untuk tiap tingkatan selanjutnya:
 - Pemeriksaan kategori variabel – variabel penjelas yang tersisa untuk menentukan peringkat yang paling signifikan dalam penentuan perbedaan variabel respon selanjutnya, dan memisahkannya dengan yang tidak signifikan
 - Penentuan variabel penjelas mana yang paling signifikan dan kemudian diteruskan lagi dengan pembagian datanya menggunakan variabel ini
 - Pengulangan tahap ke-4 untuk semua subgrup sampai teridentifikasi semua pembagian yang secara statistik telah signifikan

Berbeda dengan Gallagher, Magidson dalam Bagozzi (1994), mengatakan bahwa langkah – langkah analisis CHAID dibagi menjadi 3 tahap, yaitu:

1. Tahap 1: Penggabungan

Untuk setiap variabel penjelas, X_1, X_2, \dots, X_k

- Bentuk tabel kontingensi dua arah dengan variabel responnya
- Hitung statistik uji khi-kuadrat untuk setiap pasang kategori yang dapat dipilih untuk digabung menjadi satu, untuk menguji kebebasannya dalam sebuah subtabel kontingensi $2 \times J$ yang dibentuk oleh sepasang kategori tersebut dengan variabel responnya yang mempunyai J kategori.
- Untuk masing – masing nilai khi-kuadrat berpasangan, hitung nilai p berpasangan bersamaan. Di antara pasangan – pasangan yang tidak signifikan, gabungkan satu pasangan kategori yang paling mirip (pasangan yang memiliki nilai khi-kuadrat berpasangan terkecil) menjadi sebuah

kategori tunggal. Kemudian dilanjutkan ke langkah 4, tetapi apabila semua pasangan kategori yang tersisa signifikan, lanjutkan ke langkah 5.

- Untuk suatu kategori gabungan yang terdiri dari 3 kategori atau lebih, lakukan uji signifikansi antar kategori tersebut dengan kategori yang lain dalam satu kategori gabungan untuk melihat apakah suatu kategori variabel penjelas seharusnya dipisah. Jika hasil khi-kuadrat signifikan, pisahkan kategori tersebut dengan kategori yang lain. Jika terdapat lebih dari satu kategori yang bisa dipisah, pisahkan salah satu yang memiliki nilai khi-kuadrat yang terbesar. Kemudian kembali ke langkah 3.
- Gabungkan suatu kategori yang memiliki sedikit pengamatan yang tidak sesuai dengan kategori lain yang paling mirip, seperti yang diukur nilai khi-kuadrat berpasangan yang terkecil.
- Hitung nilai p terkoreksi Bonferroni berdasarkan tabel yang telah digabung.

2. Tahap 2: Pemisahan

- Pilih variabel penjelas terbaik, yaitu yang memiliki nilai p terendah. Kemudian lakukan pembagian kelompok dengan variabel penjelas tersebut dengan menggunakan masing – masing kategori variabel penjelas tersebut, yang telah digabung secara optimal, untuk menentukan sub pembagian dari kelompok awal menjadi sub kelompok yang baru). Jika tidak ada variabel penjelas dengan nilai p yang signifikan, jangan melakukan pembagian kelompok tersebut.

3. Tahap 3: Penghentian

- Kembali ke langkah 1 untuk menganalisis sub kelompok berikutnya. Hentikan ketika semua sub kelompok telah dianalisis dan juga telah berisi pengamatan – pengamatan dengan jumlah yang terlalu sedikit.

Algoritma CHAID menurut Kass (1980) adalah sebagai berikut:

1. Untuk masing – masing variabel penjelas, dibuat tabulasi silang antara kategori – kategori variabel penjelas dengan kategori – kategori variabel respon
2. Dari setiap tabulasi yang diperoleh, disusun semua subtabel berukuran $2 \times d$ yang mungkin. d adalah banyaknya kategori variabel respon. Carilah nilai χ^2_{hitung} semua subtabel tersebut. Dari seluruh χ^2_{hitung} yang diperoleh, cari yang terkecil dan katakan $\chi^2_{terkecil}$. Jika $\chi^2_{hitung} < \chi^2_{\alpha}$ maka kedua kategori variabel penjelas yang memiliki $\chi^2_{terkecil}$ digabungkan menjadi satu kategori campuran atau gabungan.
3. Pada setiap kategori gabungan yang terdiri atas tiga atau lebih kategori asal, dicari pembagian biner yang paling signifikan. Dari pembagian ini dicari χ^2_{hitung} terbesar. Jika χ^2_{hitung} terbesar $> \chi^2_{\alpha}$ maka pembagian biner berlaku. Kembali ke tahap 2.
4. Setelah diperoleh penggabungan optimal untuk setiap variabel penjelas, cari nilai p yang terkecil dari masing – masing sub tabel tersebut. Jika nilai p terkecil $< \alpha$ yang telah ditetapkan, maka variabel penjelas pada nilai p tersebut adalah variabel penjelas yang paling signifikan terhadap respon.

5. Jika pada tahap 4 diperoleh variabel yang pengaruhnya paling signifikan, kembali ke tahap 1 untuk setiap data hasil pemisahan.

Struktur Data uji khi-kuadrat

Baris / Kolom	1	2	...	c	Total
1	n_{11}	n_{12}	...	n_{1c}	$n_{1.}$
2	n_{21}	n_{22}	...	n_{2c}	$n_{2.}$
.
.
.
r	n_{r1}	n_{r2}	...	n_{rc}	$n_{r.}$
Total	$n_{.1}$	$n_{.2}$		$n_{.c}$	n

Probabilitas Sel

Baris / Kolom	1	2	...	c	Total
1	p_{11}	p_{12}	...	p_{1c}	$p_{1.}$
2	p_{21}	p_{22}	...	p_{2c}	$p_{2.}$
.
.
.
r	p_{r1}	p_{r2}	...	p_{rc}	$n_{r.}$
Total	$p_{.1}$	$p_{.2}$		$p_{.c}$	p

Dimana:

p_{ij} = Probabilitas kejadian irisan antara baris i dan kolom j

$p_{i.}$ = Probabilitas total pada baris ke-i

$p_{.j}$ = Probabilitas total kolom ke-j

Hipotesis pada pengujian khi-kuadrat adalah:

H_0 : $p_{ij} = p_{i.}p_{.j}$ (tidak terdapat hubungan antara baris dan kolom (bebas))

H_a : $p_{ij} \neq p_{i.}p_{.j}$ (terdapat hubungan antara baris dan kolom (tidak bebas))

Keputusan:

H_0 ditolak jika nilai $\chi^2_{\text{hitung}} \geq \chi^2_{\alpha (c-1)(r-1)}$ atau nilai $p < \alpha$.

Pada CHAID, khi-kuadrat digunakan dalam dua hal. Yang pertama, statistik khi-kuadrat digunakan untuk menentukan apakah kategori – kategori dalam sebuah variabel penjelas bersifat seragam dan bisa digabungkan menjadi satu. Yang kedua, statistik khi-kuadrat digunakan untuk menentukan variabel penjelas mana yang paling signifikan untuk membagi atau membedakan kategori – kategori dalam variabel respon ketika semua variabel penjelas sudah diringkas menjadi bentuk yang signifikan dan tidak mungkin digabung lagi, (Gallagher, 2000).

Koreksi Bonferroni

Koreksi Bonferroni adalah suatu proses koreksi yang digunakan ketika beberapa uji statistik untuk kebebasan dilakukan secara bersamaan yang biasanya digunakan dalam perbandingan berganda.

2.1.3 Populasi dan Sampel

Sugiyono berpendapat bahwa “Populasi adalah wilayah generalisasi yang terdiri atas : objek/subyek yang mempunyai kualitas dan karakteristik tertentu yang ditetapkan”.

Sugiyono mendefinisikan “Sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut”.

Menurut Sugiyono, “Jumlah sampel yang 100% mewakili populasi adalah sama dengan jumlah anggota populasi itu sendiri. Makin besar jumlah sampel mendekati

populasi, maka peluang kesalahan generalisasi semakin kecil dan sebaliknya makin kecil jumlah sampel menjauhi populasi, maka makin besar kesalahan generalisasi (diberlakukan umum)".

2.1.4 Teknik Sampling

Teknik sampling merupakan teknik pengambilan sampel. Pada penelitian ini, teknik sampling yang digunakan adalah *purposive sampling*. Menurut Sugiyono, *purposive sampling* adalah teknik penentuan sampel dengan pertimbangan tertentu.

2.1.5 Kepuasan

Kepuasan pelanggan atau sering disebut juga dengan *Total Customer Satisfaction*, seperti yang dikutip oleh Ni Nyoman dan Putu, "menurut Barkelay dan Saylor (1994:82) merupakan fokus dari proses *Customer-Driven Project Management* (CDPM), bahkan dinyatakan pula bahwa kepuasan pelanggan adalah kualitas. Begitu juga definisi singkat tentang kualitas yang dinyatakan oleh Juran (1993:3) bahwa kualitas adalah kepuasan pelanggan. Menurut Kotler yang dikutip Tjiptono (1996:146) bahwa kepuasan pelanggan adalah tingkat perasaan seseorang setelah membandingkan kinerja (atau hasil) yang dirasakan dengan harapannya". Jadi, tingkat kepuasan adalah tingkat keberhasilan suatu produk, baik dalam bentuk kualitas pelayanan maupun kualitas non-pelayanan yang dirasakan oleh pelanggan yang bisa sesuai atau tidak dengan harapan.

Seperti yang dikutip dari Ni Nyoman dan Puru Riyasa, "Tjiptono (1996:159) mengatakan bahwa ketidakpuasan pelanggan disebabkan oleh faktor internal dan faktor eksternal. Faktor internal yang relatif dapat dikendalikan perusahaan, misalnya

karyawan yang kasar, jam karet, kesalahan pencatatan transaksi. Sebaliknya, faktor eksternal yang di luar kendali perusahaan, seperti cuaca, gangguan pada infrastruktur umum, aktivitas kriminal, dan masalah pribadi pelanggan. Lebih lanjut dikatakan bahwa dalam hal terjadi ketidakpuasan, ada beberapa kemungkinan yang bisa dilakukan pelanggan, yaitu tidak melakukan apa-apa, pelanggan yang tidak puas tidak melakukan komplain, tetapi mereka praktis tidak akan membeli atau menggunakan jasa perusahaan yang bersangkutan lagi; ada beberapa faktor yang mempengaruhi apakah seorang pelanggan yang tidak puas akan melakukan komplain atau tidak, yaitu (a) derajat kepentingan konsumsi yang dilakukan, (b) tingkat ketidakpuasan pelanggan, (c) manfaat yang diperoleh, (d) pengetahuan dan pengalaman, (e) sikap pelanggan terhadap keluhan, (f) tingkat kesulitan dalam mendapatkan ganti rugi, (g) peluang keberhasilan dalam melakukan komplain. Sehubungan dengan hal tersebut di atas, kepuasan pelanggan juga sangat dipengaruhi oleh tingkat pelayanan. Menurut Moenir (1998:197), agar layanan dapat memuaskan orang atau sekelompok orang yang dilayani, ada empat persyaratan pokok, yaitu (1) tingkah laku yang sopan, (2) cara menyampaikan sesuatu yang berkaitan dengan apa yang seharusnya diterima oleh orang yang bersangkutan, (3) waktu penyampaian yang tepat, dan (4) keramahtamahan. Faktor pendukung yang tidak kalah pentingnya dengan kepuasan di antaranya faktor kesadaran para pejabat atau petugas yang berkecimpung dalam pelayanan umum, faktor aturan yang menjadi landasan kerja pelayanan, faktor organisasi yang merupakan alat serta sistem yang memungkinkan berjalannya mekanisme kegiatan pelayanan, faktor pendapatan yang dapat memenuhi kebutuhan hidup minimum, faktor keterampilan petugas, dan faktor sarana dalam pelaksanaan tugas pelayanan”.

Jadi, banyak faktor yang mempengaruhi kepuasan pelanggan dan semuanya harus diperhatikan dan dipertimbangkan dengan baik agar pelanggan merasa puas dan tidak pindah ke pesaing.

2.1.6 Bahasa Pemrograman Java

Java adalah bahasa pemrograman berorientasi objek yang dibuat berdasarkan bahasa pemrograman objek sebelumnya (C++, Ada, Simula). Java diciptakan oleh James Gosling, pengembang dari Sun Microsystems pada tahun 1991.

Berdasarkan *white paper* resmi dari SUN, Java memiliki karakteristik berikut:

1. Sederhana

Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*.

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata ke dalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Dapat didistribusi dengan mudah

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada Java.

4. *Interpreter*

Program Java dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada *platform* yang berbeda-beda.

5. *Robust*

Java mempunyai reliabilitas yang tinggi, *compiler* pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi error pada pemrograman.

6. Aman

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

7. *Architecture Neutral*

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada *platform* yang berbeda dengan *Java Virtual Machine*.

8. *Portabel*

Source code maupun program Java dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus dikompilasi ulang.

9. *Performance*

Performance pada Java sering dikatakan kurang tinggi. Namun *performance* Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan Just In Time Compilers (JIT).

10. *Multithreaded*

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

2.1.7 Rekayasa Perangkat Lunak (RPL)

Istilah RPL digunakan pertama kali pada akhir 1950-an dan awal 1960-an. Saat itu, masih terdapat perdebatan tajam mengenai aspek *engineering* dari pengembangan perangkat lunak. Pada tahun 1968 dan 1969, komite sains NATO mensponsori dua konferensi tentang RPL, yang memberikan dampak kuat terhadap pengembangan rekayasa perangkat lunak. Banyak yang menganggap dua konferensi inilah yang menandai awal resmi profesi rekayasa perangkat lunak.

Menurut Roger Pressman, RPL adalah:

- Pembentukan dan penggunaan prinsip rekayasa (*engineering*) untuk mendapatkan perangkat lunak secara ekonomis namun andal dan dapat bekerja secara efisien pada komputer (Fritz Bauer, 1968).
- Suatu disiplin, kaidah yang mengintegrasikan proses, metode, dan alat bantu (*tools*) untuk pembangunan perangkat lunak komputer.

Menurut *Institute of Electrical and Electronics Engineers (IEEE)*, RPL adalah

- *The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*
- *The study of approaches as in (1).*

Model Proses

Menurut Pressman (2005, 78) model proses merupakan gambaran dari suatu proses rekayasa perangkat lunak yang terdiri dari aktifitas-aktifitas, tindakan-tindakan, tugas-tugas, tujuan-tujuan dan hasil kerja yang dibutuhkan untuk menghasilkan perangkat lunak yang berkualitas.

Model proses yang digunakan dalam tugas akhir ini adalah model *Waterfall*. Model ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial.

Tahap-tahap model proses *waterfall* adalah:

1. *Requirements Analysis and Definition*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

2. *System and Software Design*

Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

3. *Implementation and Unit Testing*

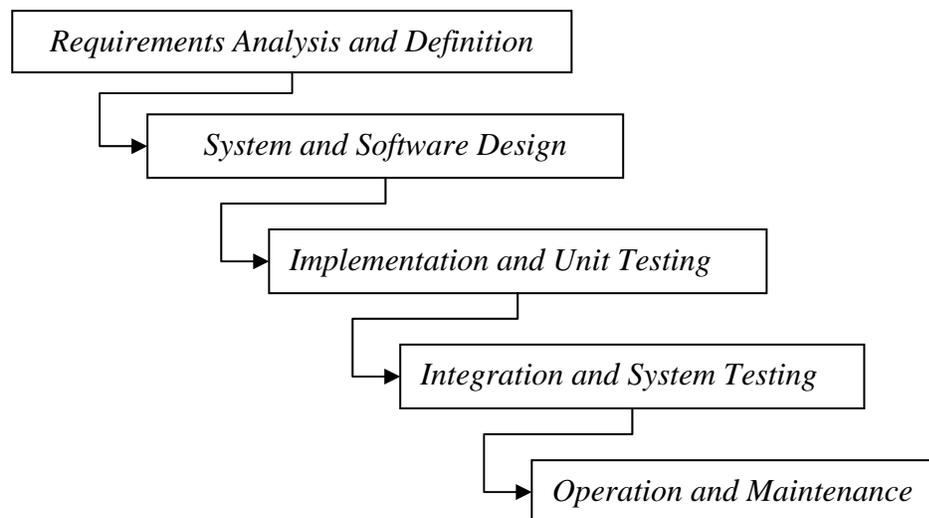
Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

4. *Integration and System Testing*

Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

5. *Operation and maintenance*

Mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.



Gambar 2.1 Model Proses *Waterfall*

2.1.8 Interaksi Manusia dan Komputer (IMK)

IMK menurut Galitz adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif.

Terdapat aturan – aturan dalam perancangan sebuah *interface*, yang dikenalkan oleh Shneiderman (2005, p74-75) dengan istilah *Eight Golden Rules of Interface Design*, yaitu:

1. *Strive for Consistency*

Konsistensi dilakukan pada urutan tindakan, perintah, dan istilah yang digunakan pada prompt, menu, serta layar bantuan. Selain itu, kita juga harus konsisten dalam penggunaan bentuk dan ukuran *font*, pemberian warna pada latar belakang dan tulisan, serta pembuatan *layout*.

2. *Enable frequent users to use shortcuts*

Ada kebutuhan dari pengguna yang sudah ahli untuk meningkatkan kecepatan interaksi, sehingga diperlukan singkatan, tombol fungsi, perintah tersembunyi, dan fasilitas makro.

3. *Offer informative feedback*

Untuk setiap tindakan operator, sebaiknya disertakan suatu sistem umpan balik. Untuk tindakan yang sering dilakukan dan tidak terlalu penting, dapat diberikan umpan balik yang sederhana. Tetapi ketika tindakan merupakan hal yang penting, maka umpan balik sebaiknya lebih substansial. Misalnya muncul suatu suara ketika salah menekan tombol pada waktu input data atau muncul pesan kesalahannya.

4. *Design dialogs to yields closure*

Urutan tindakan sebaiknya diorganisir dalam suatu kelompok dengan bagian awal, tengah, dan akhir. Umpan balik yang informatif akan memberikan indikasi bahwa cara yang dilakukan sudah benar dan dapat mempersiapkan kelompok tindakan berikutnya.

5. *Offer error prevention and simple error handling*

Sedapat mungkin sistem dirancang sehingga pengguna tidak dapat melakukan kesalahan fatal. Jika kesalahan terjadi, sistem dapat mendeteksi kesalahan dengan cepat dan memberikan mekanisme yang sederhana dan mudah dipahami untuk penanganan kesalahan.

6. *Permit easy reversal of actions*

Hal ini dapat mengurangi kekhawatiran pengguna karena pengguna mengetahui kesalahan yang dilakukan dapat dibatalkan; sehingga pengguna tidak takut untuk mengeksplorasi pilihan-pilihan lain yang belum biasa digunakan.

7. *Support internal locus of control*

Pengguna ingin menjadi pengontrol sistem dan sistem akan merespon tindakan yang dilakukan pengguna daripada pengguna merasa bahwa sistem mengontrol pengguna. Sebaiknya sistem dirancang sedemikian rupa sehingga pengguna menjadi inisiator daripada responden.

8. *Reduce short-term memory load*

Keterbatasan ingatan manusia membutuhkan tampilan yang sederhana atau banyak tampilan halaman yang sebaiknya disatukan, serta diberikan cukup waktu pelatihan untuk kode, mnemonic, dan urutan tindakan.

Suatu program yang interaktif dan baik harus bersifat *user friendly*. Menurut Shneiderman (2005, p15) ada 5 kriteria yang harus dipenuhi oleh suatu program agar bersifat *user friendly*, yaitu:

- Waktu belajar (*Time to learn*)

Berapa lama waktu yang dibutuhkan oleh *user* untuk mempelajari dan menggunakan perintah-perintah yang relevan untuk suatu tugas (*tasks*). Sebuah sistem yang baik, seharusnya mudah dipelajari dan digunakan bahkan oleh pengguna awam sekali pun.

- Kecepatan kinerja (*Speed of performance*)

Berapa lama waktu yang diperlukan untuk menyelesaikan tugas (*tasks*). Sebuah system yang baik, dapat menyelesaikan masalah dan melakukan pemrosesan data dengan cepat dan efisien.

- Tingkat kesalahan *user* (*Rate of errors by users*)

Berapa banyak kesalahan dan kesalahan apa saja yang dilakukan oleh pengguna dalam menyelesaikan tugas. Sebuah sistem yang baik meminimalkan jumlah dan tingkat kesalahan pengguna.

- Penghafalan dari waktu ke waktu (*Retention over time*)

Seberapa lama pengguna dapat mempertahankan pengetahuannya mengenai penggunaan sistem tersebut sehingga pengguna tidak perlu mempelajari ulang cara penggunaan sistem di masa yang akan datang. Penghafalan mungkin berhubungan erat dengan waktu untuk belajar dan frekuensi penggunaan.

- Kepuasan subjektif (*Subjective satisfaction*)

Berapa besar tingkat kepuasan pengguna terhadap sistem yang digunakan. Hal ini dapat diketahui melalui wawancara atau dengan survei tertulis yang mengandung tingkat kepuasan dan komentar dari pengguna.

2.1.9 *Unified Modeling Language (UML)*

Menurut Booch, et al (2000, p1) UML merupakan bahasa pemodelan, bukan sebuah metode. Sebagian besar metode itu sifatnya konsisten, setidaknya mempunyai dasar atau aturan, terdiri atas bahasa pemodelan dan proses. Bahasa pemodelan dipakai untuk mengekspresikan desain.

UML biasa digambarkan dengan diagram – diagram agar sistem mudah dimengerti.

Berikut ini adalah diagram – diagram UML yang digunakan dalam penelitian ini:

1. *Use Case Diagram*

Use Case Diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat user. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya.

Adapun komponen-komponen dalam *use case diagram* adalah sebagai berikut (Booch et al, 2000, p40-45):

- a. *Actor* berperan sebagai *user* terhadap system
- b. *Use case* merupakan bagian dari skenario yang terikat bersama-sama dengan tujuan umum *user*
- c. *Use case relationship* merupakan penghubung antara *actors* dan *use cases*
- d. *Include* dipakai untuk memisahkan *use cases* dan untuk menghindari perulangan
- e. *Generalization* dipakai untuk mendeskripsikan variasi dari tindakan dan merupakan keinginan *user*
- f. *Extend* digunakan untuk mendeskripsikan variasi dari tindakan

2. *Activity Diagram*

Activity diagram menggambarkan urutan kegiatan, dengan dukungan untuk suatu kondisi dan *parallel behavior*. (Booch et al, 2000, p129).

2.2 Hipotesis

Hipotesis merupakan jawaban sementara terhadap rumusan masalah penelitian.

Hipotesis dalam penelitian ini adalah:

1. H₀: Tidak ada perbedaan tingkat kepuasan terhadap fasilitas berdasarkan identitas peserta kursus.
H_a: Ada perbedaan tingkat kepuasan terhadap fasilitas berdasarkan identitas peserta kursus.
2. H₀: Tidak ada perbedaan tingkat kepuasan terhadap biaya kursus berdasarkan identitas peserta kursus.
H_a: Ada perbedaan tingkat kepuasan terhadap biaya kursus berdasarkan identitas peserta kursus.

3. H0: Tidak ada perbedaan tingkat kepuasan terhadap tenaga pengajar berdasarkan identitas peserta kursus.

Ha: Ada perbedaan tingkat kepuasan terhadap tenaga pengajar berdasarkan identitas peserta kursus.

4. H0: Tidak ada perbedaan tingkat kepuasan terhadap metode pengajaran berdasarkan identitas peserta kursus.

Ha: Ada perbedaan tingkat kepuasan terhadap metode pengajaran berdasarkan identitas peserta kursus.

5. H0: Tidak ada perbedaan tingkat kepuasan terhadap waktu kursus berdasarkan identitas peserta kursus.

Ha: Ada perbedaan tingkat kepuasan terhadap waktu kursus berdasarkan identitas peserta kursus.

6. H0: Tidak ada perbedaan tingkat kepuasan terhadap lokasi berdasarkan identitas peserta kursus.

Ha: Ada perbedaan tingkat kepuasan terhadap lokasi berdasarkan identitas peserta kursus.