

## **BAB II**

### **LANDASAN TEORI**

#### **II.1 Tinjauan Perusahaan**

Pada tahap ini merupakan tinjauan perusahaan tempat penelitian yaitu PT. TIKI JNE, menampilkan profil, sejarah, logo instansi, struktur organisasi serta deskripsi pekerjaan., visi dan misi perusahaan dan deskripsi lokasi.

##### **II.1.1 Profil dan Sejarah PT. TIKI JNE**

PT. TIKI JNE (Jalur Nugraha Ekakurir) didirikan pada tanggal 26 November 1990 dengan memulai usahanya yang terpusat pada penanganan kegiatan Kepabeanan/Impor kiriman barang/dokumen dari luar negeri sampai pengantarannya ke seluruh wilayah Indonesia.

Pada tahun 1991, PT. TIKI JNE memperluas jaringan Internasionalnya dengan bergabung sebagai anggota asosiasi perusahaan-perusahaan kurir beberapa negara Asia (*Assosiated Courier Conference of Asia*) yang bermarkas di Hongkong yang kemudian memberi kesempatan kepada PT. TIKI JNE untuk mengembangkan wilayah antaran sampai ke seluruh dunia.

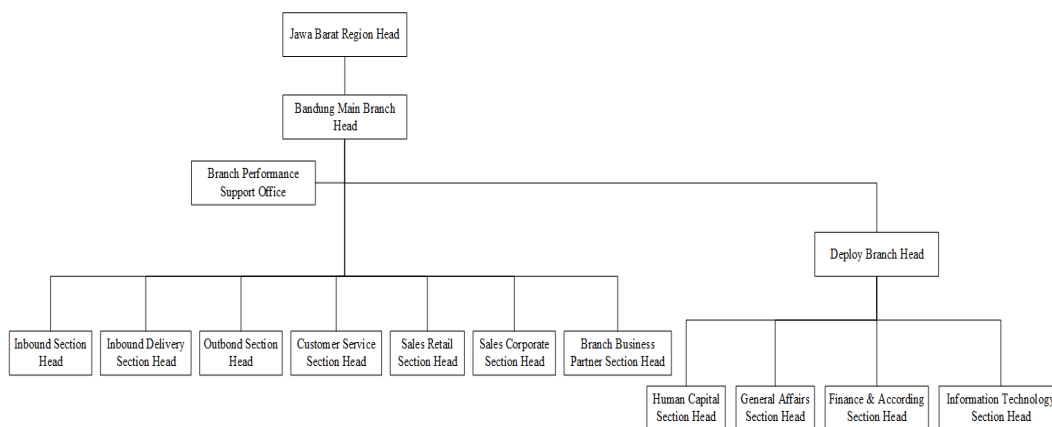
## II.1.2 Logo Instansi

Berikut Merupakan Logo Instansi dari PT. TIKI JNE, dapat dilihat pada Gambar II.1.



Gambar II.1 Logo PT. TIKI JNE

## II.1.3 Struktur Organisasi dan Deskripsi Pekerjaan



Gambar II.2 Struktur Organisasi PT TIKI JNE Bandung

Berdasarkan Skema organisasi dari Gambar II.2 diatas bisa disimpulkan job description dan jabatan serta tugasnya masing-masing, orang-orang tersebut yang terlibat di PT TIKI JNE Bandung sekarang ini, job description nya sebagai berikut :

1. Jawa Barat *Region Head* : Memonitoring/mengelola seluruh cabang regional Jawa Barat.
2. Bandung *Main Branch Head* : Kepala cabang utama, mengelola/memonitoring cabang utama Bandung.

3. *Branch Performance Support Office* : Mengelola data operasional JNE Bandung.
4. *Deploy Branch Head* : Mengelola/memonitoring *department support*.
5. *Inbound Section Head* : Mengelola barang kiriman yang masuk ke PT. TIKI JNE Bandung.
6. *Inbound Delivery Section Head* : Mengelola barang kiriman masuk ke Bandung dan melakukan kiriman barang ke penerima (*customer*).
7. *Outbond Section Head* : Mengelola barang yang keluar dari PT. TIKI JNE Bandung.
8. *Customer Service Section Head* : Melayani *customer* yang datang ke *counter* seputar produk WIC (*Walk In Customer*), dan via telepon (*call center*).
9. *Sales Retail Section Head* : Melakukan/mengelola penjualan *retail (counter)*, pemasaran.
10. *Sales Corporate Section Head* : Melakukan/mengelola penjualan *corporate* (lingkup antar perusahaan) serta *maintenance customer exiting*.
11. *Branch Business Partner Section Head* : Mengelola/memonitoring cabang dibawah cabang utama Bandung.
12. *Human Capital Section Head* : Mengelola SDM dari mulai *recruitment* hingga *training, development, ingagement*, dan lain-lain.
13. *General Affairs Section Head* : Mengelola asset perusahaan (Bangunan, surat legalitas perusahaan, ATK, Keamanan, Kebersihan).
14. *Finance & According Section Head* : Mengelola keuangan perusahaan, baik pengeluaran atau pemasukan.
15. *Information Technology Section Head* : Mengelola teknologi baik berupa *software, hardware*, jaringan perusahaan.

#### **II.1.4 Visi dan Misi TIKI JNE**

Sebuah perusahaan selalu menggagas impian atau tujuan yang ingin dicapai. Selain tujuan utama, visi dan misi masuk dalam bentuk-bentuk gagasan atau pedoman tertulis. Adapun visi dan misi yang sudah ditetapkan perusahaan PT. TIKI JNE Bandung sampai sekarang:

##### **II.1.4.1 Visi**

Visi merupakan tujuan masa depan sebuah instansi, organisasi, atau perusahaan. berikut adalah Visi dari perusahaan PT. TIKI JNE Bandung:

“Sebagai perusahaan rantai pasok global terdepan di dunia.”

##### **II.1.4.2 Misi**

Misi Adalah tahapan-tahapan yang harus dilalui untuk mencapai visi tersebut. berikut adalah Visi dari perusahaan PT. TIKI JNE Bandung:

“Memberi pengalaman terbaik kepada pelanggan secara konsisten.”

## II.2 Landasan Teori

Landasan teori merupakan bagian yang akan membahas tentang uraian pemecahan masalah yang akan ditemukan pemecahannya melalui pembahasan-pembahasan secara teoritis. Teori-teori yang akan dikemukakan merupakan dasar-dasar teori sistem informasi sebagai bahan acuan.

### II.2.1 Sistem Informasi

Sistem informasi berasal dari kata “Sistem” dan “Informasi”. Menurut James A. O’Brien & George Marakas (O’Brien & M. Marakas, 2010, p. 26), sistem adalah sekelompok komponen yang saling bekerja sama menuju tujuan bersama dengan menerima input dan output dalam suatu proses transformasi yang terorganisir. Sedangkan informasi adalah data yang telah diubah menjadi sesuatu yang berarti dan pernyataan berguna bagi pengguna akhir.

Sistem (kadang-kadang disebut sistem dinamis) memiliki tiga komponen dasar yang berinteraksi atau fungsi:

#### 1. *Input*

*Input* melibatkan menangkap dan perakitan elemen yang memasuki sistem untuk diproses. Misalnya, bahan baku, energi, data, dan usaha manusia harus dijamin dan diatur untuk diproses

#### 2. *Processing*

*Processing* melibatkan proses transformasi yang mengkonversi input ke output. Contoh adalah proses manufaktur, proses pernapasan manusia, atau perhitungan matematis.

#### 3. *Output*

*Output* melibatkan mentransfer elemen yang telah diproduksi oleh proses transformasi ke tujuan akhir. Sebagai contoh, produk jadi, pelayanan manusia, dan informasi manajemen harus dikirimkan ke pengguna.

Selain tiga fungsi dasar tersebut, sebuah sistem akan lebih berguna dengan menambahkan dua elemen lain (O'Brien & M.Marakas, 2010, p. 29), yaitu:

### 1. *Feedback*

*Feedback* adalah data mengenai kinerja dari sebuah sistem. *Feedback* dapat dipahami juga sebagai respon dari sistem atas tindakan yang dilakukan. Contohnya adalah apabila ada debu yang masuk ke dalam sistem respirasi manusia, maka akan muncul reaksi untuk mengeluarkan debu dengan cara bersin.

### 2. *Control*

*Control* adalah fungsi yang mencakup pemantauan dan evaluasi *feedback* untuk menentukan apakah sistem berjalan mencapai sarannya, kemudian fungsi ini akan membuat penyesuaian yang dibutuhkan terhadap komponen *input* sistem dan pemrosesan untuk memastikan sistem dapat menghasilkan output yang sesuai. Contohnya adalah di lingkungan yang berdebu, manusia akan menggunakan masker penutup hidung untuk melindungi agar tidak ada debu yang masuk ke dalam hidung.

Gabungan dari kata “Sistem” dan “Informasi” membentuk sistem informasi. Menurut James A. O'Brien & George Marakas (O'Brien & M.Marakas, 2010, p. 4), suatu sistem informasi dapat berupa kombinasi terorganisir dari orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan informasi yang disebarkan dalam proses organisasi.

Menurut Kenneth C.Laudon & Jane P.Laudon (Laudon & P.Laudon, 2006, p. 13), sebuah sistem informasi dapat didefinisikan secara teknis sebagai seperangkat komponen yang saling terkait yang mengumpulkan (atau mengambil), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan, koordinasi, dan pengendalian, sistem informasi dapat juga membantu para manajer dan pekerja menganalisis masalah, memvisualisasikan subyek yang kompleks dan menciptakan produk baru. Dari beberapa pengertian sistem informasi, dapat diambil kesimpulan bahwa sistem informasi adalah sekumpulan komponen yang melakukan pengumpulan data untuk menghasilkan suatu informasi yang dapat berguna bagi pengguna akhir.

Dalam sistem informasi terdiri dari beberapa komponen. Menurut James A. O'Brien, George Marakas (O'Brien & M.Marakas, 2010, pp. 31-34), ada 5 komponen sistem informasi :

1. *People Resource*

Manusia adalah unsur penting untuk keberhasilan operasi dari semua sistem informasi. Sumber daya manusia termasuk *end user* dan *IS.spesialis*.

2. *Hardware Resources*

Sumber daya perangkat keras termasuk mesin yaitu sistem komputer dan peralatan-peralatan lain dan media.

3. *Software Resource*

Sumber daya perangkat lunak termasuk sistem perangkat lunak seperti sistem operasi program, aplikasi software yang langsung diproses untuk penggunaan khusus bagi end-users dan prosedur yang memberitahukan pengguna bagaimana menggunakan sistem informasi.

4. *Data Resources*

Data merupakan sumber daya organisasi yang berharga. Dengan demikian, sumber daya data harus dikelola secara efektif untuk menguntungkan semua pengguna akhir dalam suatu organisasi.

5. *Network Resources*

Teknologi telekomunikasi dan jaringan seperti internet, intranet, dan extranet telah menjadi bisnis elektronik penting dan operasi perdagangan semua jenis organisasi dan komputer berbasis sistem informasi.

## **II.2.2 Customer Relationship Management (CRM)**

Menurut O'Brien & Marakas (O'Brien & M.Marakas, 2010, p. 271), *Enterprise Resources Planning* (ERP) adalah sistem lintas fungsional perusahaan yang didorong oleh modul perangkat lunak terintegrasi yang mendukung dasar proses bisnis internal perusahaan. *Enterprise resources planning* (ERP) berkonsentrasi pada efisiensi produksi perusahaan internal, distribusi, dan proses keuangan. *Customer relationship management* (CRM) berfokus pada mendapatkan dan mempertahankan pelanggan yang menguntungkan melalui

pemasaran, penjualan, dan proses layanan. *Partner relationship management* (PRM) bertujuan mendapatkan dan mempertahankan mitra yang dapat meningkatkan penjualan dan distribusi produk dan jasa perusahaan. *Supply chain management* (SCM) berfokus pada pengembangan sumber dan proses pengadaan yang paling efisien dan efektif dengan pemasok untuk produk dan layanan yang dibutuhkan oleh bisnis. *Knowledge management* (KM) fokus pada penyediaan karyawan perusahaan dengan alat-alat yang mendukung kolaborasi kelompok dan mendukung keputusan.

Menurut O'Hara & Kumar dalam jurnal (Zani, Salim, Djumahir, & Solimun, 2013), CRM dalam konteks pendidikan adalah salah satu kunci sukses bagi lembaga pendidikan jangka panjang. Lembaga pendidikan harus berpikir bahwa tidak semua kegiatan akan sepenuhnya berjalan dengan baik tanpa mempertimbangkan apa yang diinginkan oleh pelanggan (dalam kasus ini orang tua murid). Lembaga-lembaga pendidikan harus memperhatikan dan meningkatkan kualitas pelayanan kepada orang tua siswa sehingga kepuasan kebutuhan dapat dipenuhi. CRM memiliki beberapa jenis serta tingkatan yaitu :

1. Strategis

Pengembangan strategis perusahaan untuk keputusan manajemen puncak dan menjadikan pedoman untuk tingkat CRM lainnya pada hubungan perusahaan dengan pelanggan.

2. Operasional

Berfokus pada proyek-proyek otomatisasi seperti otomatisasi layanan, armada, penjualan dan pemasaran.

3. Analitis

Berfokus pada pengolahan basis data dan disajikan bagi pengambil keputusan perusahaan untuk kepentingan strategis dan operasional.

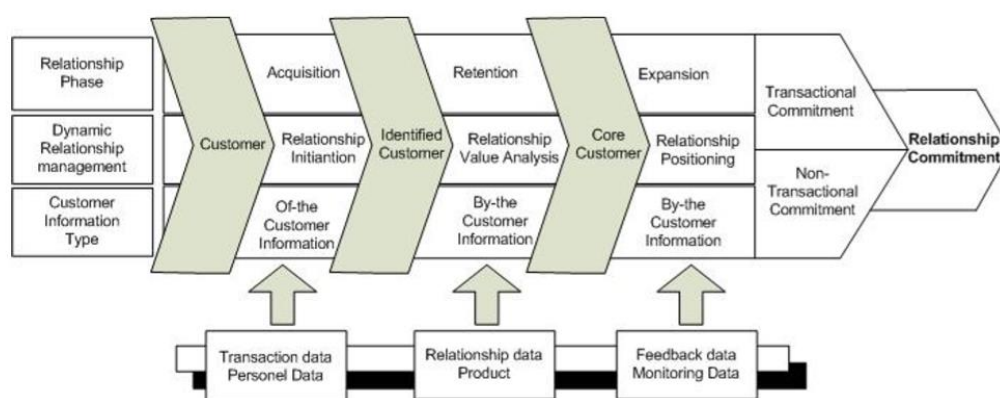
4. Kolaboratif

Hubungan dengan mitra kerja di luar perusahaan.



### II.2.3 Framework of Dynamic CRM

Untuk menetapkan fitur CRM secara tepat diperlukan *framework* sebagai acuan. CH Park & YG Kim mengusulkan sebuah CRM *framework* yang dinamakan “A *framework of Dynamic CRM*”. *Framework* ini menjelaskan serangkaian tahapan pada pembangunan atau penerapan CRM. Substantif terpentingnya adalah informasi yang didapat dari customer sehingga diperoleh outputnya yang berupa *Relationship Commitment*.



**Gambar II.3 Framework of Dynamic CRM**

Berikut Pendekatan Dalam Menetapkan Fitur CRM :

Siklus CRM yang terdiri dari proses *Acquire–Retain–Expansion*, masing masing fase dalam siklus tersebut dapat dijelaskan sebagai berikut:

1. *Acquire*, proses acquiring a new relationship adalah untuk mendapatkan pelanggan baru yang dapat diwujudkan dalam berbagai strategi. Kata kunci yang dikaitkan dengan fase ini adalah diferensiasi, inovasi, dan kenyamanan (*convenience*).
2. *Retain*, *retaining customer relationship* adalah fase dimana organisasi melakukan strategi untuk mempertahankan pelanggan dengan cara peningkatan pelayanan. *Customer retention* menjadi strategi yang penting bagi perusahaan karena pilihan bagi pelanggan semakin banyak yang diciptakan oleh banyak organisasi penyedia layanan (*competitor*). Kata kunci

yang diasosiasikan dengan fase ini adalah *adaptability*, *listening*, dan *Responship*.

3. *Expansion*, adalah masa untuk membina hubungan baik yang telah ada dengan pelanggan, dengan selalu mendengar keinginan pelanggan dan melayaninya dengan baik, terciptanya pelanggan yang loyal terhadap produk/layanan organisasi. Beberapa kata kunci yang di asosiasikan dengan fase ini adalah *loyalty*, *reduce cost*, dan *customer service*.

Dari beberapa definisi tentang CRM dapat disimpulkan bahwa *Customer Relationship Management (CRM)* terdiri beberapa pihak yang terlibat dalam penerapannya secara langsung mulai dari customer hingga organisasi penyedia layanan (*service*) dan menjadi sebuah siklus yang dapat di management sesuai dengan proses bisnis organisasi tersebut. Untuk menetapkan fitur CRM secara tepat, sebagai landasan untuk menyusun *Software Requirement Specification (SRS)*, diperlukan framework sebagai acuan. CH Park & YG Kim mengusulkan sebuah *CRM Framework* yang dinamakan “*A framework of Dynamic CRM*”. Substantif terpentingnya adalah informasi yang didapat dari customer sehingga diperoleh outputnya yang berupa *Relationship Commitment*, model tersebut dapat dilihat pada Informasi yang menjadi fokus perhatian adalah sebagai berikut:

- a. Informasi mengenai pelanggan
- b. Informasi untuk pelanggan
- c. Informasi oleh pelanggan

Didasari dari ketiganya di atas jelas informasi amat penting untuk mewujudkan *Customer Relation Management (CRM)* Organisasi untuk melakukan persaingan, dengan melakukan pengolahan informasi tersebut akan didapat *customer behavior*, karena *customer behavior* pada saat ini lebih cenderung mengingkan bagaimana mereka merasa diperhatikan, dilayani serta ketanggapan dari organisasi sebagai penyedia layanan kepada mereka dan biasanya kecenderungan tersebut tidak melihat jumlah harga yang mereka keluarkan lagi. Dengan semakin meningkatnya pelayanan terhadap customer suatu

organisasi seharusnya harus sudah dapat memisahkan seperti yang penulis bahas di awal yang mana fasilitas dan yang mana suatu *service*.

#### II.2.4 Metode QFD

*Quality Function Deployment* (QFD) dikenalkan dan dikembangkan oleh Yoji Akao dari tahun 1965 hingga 1967 bersama Katsuyo Ishihara dari *Matsushita Electric*. QFD diterapkan pertama kali pada tahun 1972 oleh *Mitsubishi Heavy Industries Limited* di Kobe Shipyard, Jepang. Kemudian pada akhir 1970, konsep proses detail QFD dikembangkan lebih lanjut oleh Toyota yang selanjutnya banyak digunakan oleh industri manufaktur dunia.

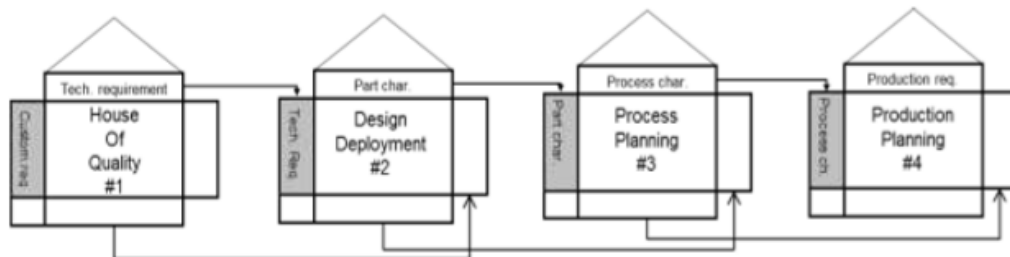
QFD merupakan sebuah manajemen kualitas yang berfokus pada konsumen dan metodologi pengembangan produk yang bersifat *tangible*. Namun, secara perlahan QFD mulai diterapkan pada sector jasa untuk mendesain dan mengembangkan kualitas pelayanan.

*Quality Function Deployment* (QFD) adalah suatu sistem untuk mengubah keinginan pelanggan menjadi karakteristik kualitas dan mengembangkan suatu desain kualitas untuk menghasilkan produk yang secara sistematis menyebarkan (deploying) hubungan antara keinginan dan karakteristik tersebut. (Akao, 1990).

Menurut Cohen, metode QFD memiliki beberapa tahap perencanaan dan pengembangan melalui matriks, yaitu:

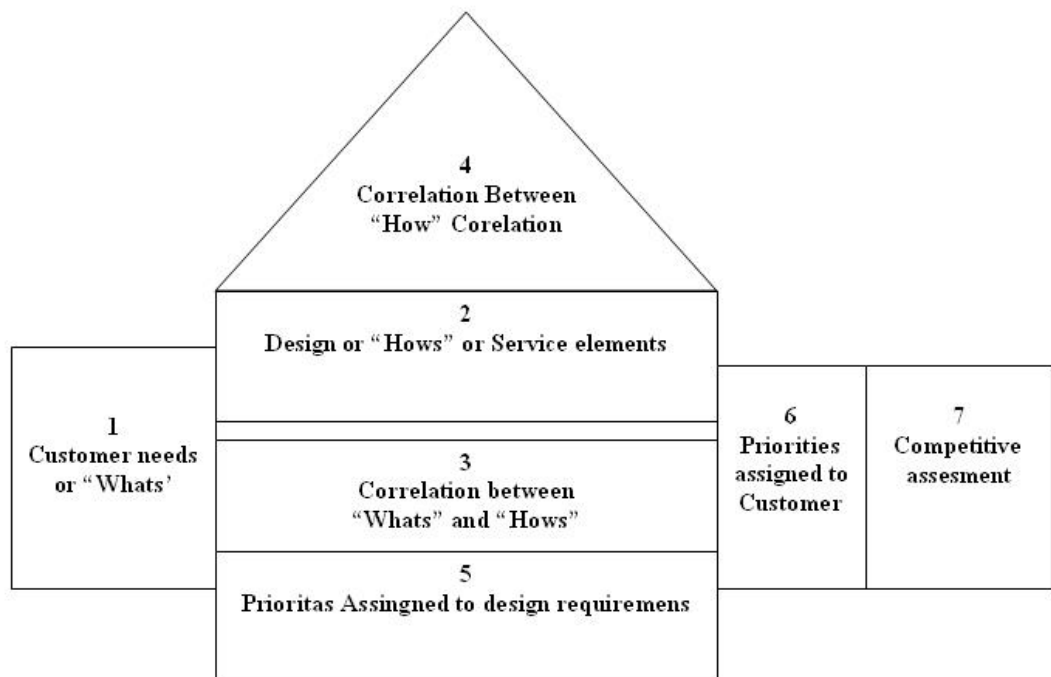
1. Matriks perencanaan (*house of quality* - HOQ) merupakan utama dari QFD yang menjelaskan tentang *customer needs, technical requirements, co-relationship, relationship, customer competitive evaluation, competitive technical-assesment*, dan target.
2. Matriks perencanaan desain (*desain deployment*) adalah matriks untuk mengidentifikasi desain yang kritis terhadap pengembangan produk.
3. Matriks perencanaan proses (*process planning*) merupakan matriks untuk mengidentifikasi pengembangan proses pembuatan suatu produk.
4. Matriks perencanaan produksi (*production planing*) memeparkan tindakan yang perlu diambil dalam perbaikan produksi suatu produk.

Keempat tahap dalam analisis penyusunan matriks QFD seperti tampak pada Gambar II.4.



**Gambar II.4 Model QFD**

Penerapan metode *Quality Function Deployment* dalam proses perancangan produk dan jasa diawali dengan pembentukan matriks perencanaan produk atau sering disebut sebagai *House of Quality* (rumah kualitas) seperti pada Gambar II.5.



**Gambar II.5 House of Quality**

Berikut penjelasan dan urutan *House of Quality* yang terdapat pada Gambar II.5:

1. *Customer Needs or Whats*, merupakan daftar keinginan dan kebutuhan pelanggan, yang secara kualitatif diperoleh melalui pendekatan deskriptif atau wawancara terhadap pelanggan.
2. *Design requirement or Hows*, merupakan bahasa teknis dari perusahaan untuk menyusun desain perbaikan produk atau jasa yang umumnya merupakan gambaran teknis dari kebutuhan dan keinginan konsumen pada bagian *Customer Needs or Whats*.
3. *Correlation matrixs between whats and hows*, bagian ini mencakup informasi yang menunjukkan :
  - a. Tingkat kepentingan dari keinginan dan kebutuhan pelanggan serta tingkat kepuasan pelanggan terhadap pelayanan yang diberikan oleh perusahaan.

- b. Tingkat hubungan antara keinginan dan kebutuhan pelanggan dengan kemampuan perusahaan untuk memenuhinya dengan menggunakan simbol yang menunjukkan korelasi kuat, sedang dan lemah.
4. *Correlation between hows*, merupakan korelasi antara karakteristik kemampuan teknis dari perusahaan yang dibedakan menjadi positif dan negatif.
  5. *Priorities assigned to design requirement*, merupakan prioritas yang ditetapkan perusahaan untuk melaksanakan setiap target sesuai dengan tingkat kesulitannya.
  6. *Priorities assigned to Customer*, merupakan bagian yang menunjukkan tingkat kepentingan dari keinginan dan kebutuhan pelanggan berdasarkan informasi langsung dari konsumen.
  7. *Competitive assessment*, menunjukkan penilaian pelanggan terhadap produk atau jasa yang dibandingkan dengan pesaing.

### II.2.5 Skala Likert

Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan Skala *Likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan. Jawaban setiap item instrumen yang menggunakan Skala *Likert* mempunyai sifat dari sangat positif sampai sangat negatif, yang dapat berupa kata-kata antara lain: Sangat Penting (SP), Penting (P), Ragu-ragu (R), Tidak Penting (TP), Sangat Tidak Penting (STP). Untuk penilaian ekspektasi, maka jawaban itu dapat diberi skor, misalnya:

- Sangat Penting (SP) = 5,
- Penting (P) = 4,
- Ragu-ragu (R) = 3,
- Tidak Penting (TP) = 2 ,
- Sangat Tidak Penting (STP) = 1.

Sedangkan untuk penilaian persepsi, maka jawaban itu dapat diberi skor, misalnya:

- Sangat Baik (SB) = 5,

- Baik (B) = 4,
- Cukup Baik (CB) = 3,
- Tidak Baik (TB) = 2
- Sangat Tidak Baik (STB) = 1

Instrumen penelitian yang menggunakan skala Likert dapat dibuat dalam bentuk checklist ataupun pilihan ganda.

## **II.2.6 Basis Data**

Basis Data terdiri atas 2 kata, yaitu Bisnis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek. Basis Data sendiri dapat didefinisikan yaitu Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

### **II.2.6.1 DBMS**

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah Perangkat Lunak (Sistem) yang khusus. Perangkat inilah (disebut DBMS/*Database Management System*) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan/konsistensi data, dan sebagainya.

## **II.2.7 Object Oriented Analysis Design**

Menurut Satzinger, Jackson & Burd (Satzinger, Jackson, & Burd, 2010, p. 60), *Object-Oriented Analysis (OOA)*, mendefinisikan semua jenis objek yang melakukan pekerjaan dalam suatu sistem dan menunjukkan interaksi dari pengguna yang dibutuhkan untuk menyelesaikan tugas.

Menurut Satzinger, Jackson & Burd (Satzinger, Jackson, & Burd, 2010, p. 60), *Object-Oriented Design (OOD)*, mendefinisikan semua jenis objek dari objek yang diperlukan untuk dapat berkomunikasi dengan orang dan perangkat yang ada didalam sistem, menggambarkan bagaimana objek dapat menyelesaikan tugas,

dan menyesuaikan dan menyempurnakan masing-masing jenis objek sehingga dapat diimplementasikan dengan bahasa tertentu atau lingkungan.

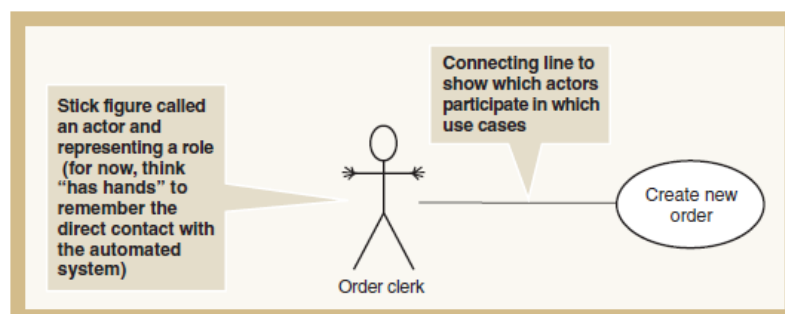
Menurut Whitten et al. (Whitten, Bentley, & Dittman, 2009, p. 25), *object oriented analysis and design* (OOAD) merupakan suatu kumpulan alat dan teknik untuk mengembangkan suatu sistem yang akan menggunakan teknologi objek untuk membangun sebuah sistem dan piranti lunak.

### II.2.7.1 Unified Modeling Language (UML)

Menurut Satzinger, Jackson, & Burd (Satzinger, Jackson, & Burd, 2005, p. 48), *Unified Modeling Language* (UML) adalah serangkaian standar konstruksi model dan notasi yang dikembangkan secara khusus untuk pengembangan *object-oriented*.

### II.2.7.2 Use Case Diagram

Menurut Satzinger, Jackson, & Burd (Satzinger, Jackson, & Burd, 2010, p. 242), *use case diagram* merupakan sebuah diagram yang menunjukkan bermacam-macam peran pengguna dan cara mereka berinteraksi dengan sistem.



Sumber : (Satzinger, Jackson, & Burd, 2010, p. 243)

**Gambar II.6 Contoh *Use Case Diagram***

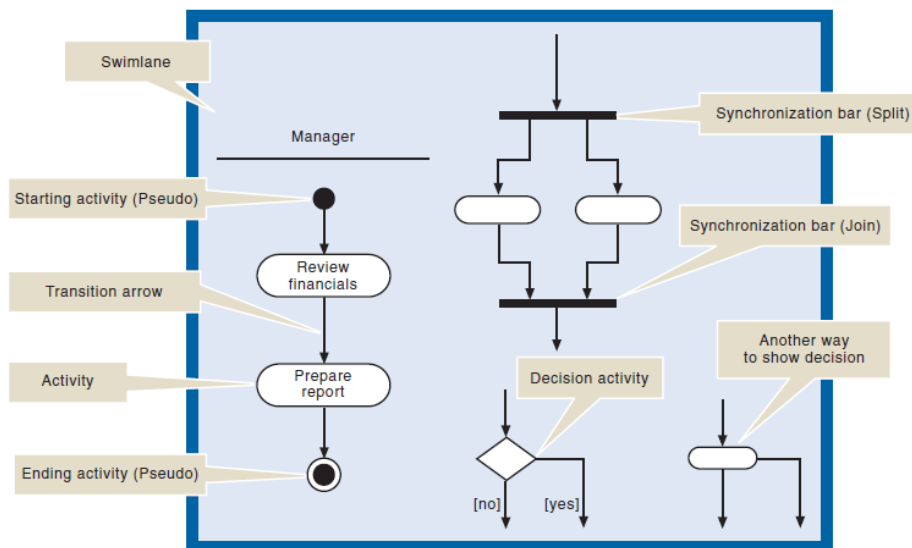


Menurut Satzinger, kita dapat mengembangkan *use case diagram* dalam dua langkah (Satzinger, Jackson, & Burd, 2010, p. 248) :

- a. Mengidentifikasi aktor-aktor dari sistem. Aktor adalah peran yang dimainkan oleh pengguna. Harus mengidentifikasi peran spesifik pengguna, orang yang sama dapat memainkan berbagai peran karena ia menggunakan sistem. Hal ini penting untuk mengidentifikasi setiap kemungkinan peran dalam menggunakan sistem.
- b. Setelah peran aktor telah diidentifikasi, langkah berikutnya adalah untuk mengembangkan daftar tujuan peran-peran yang ada dalam penggunaan sistem otomatis. Tujuan adalah tugas yang dilakukan oleh seorang aktor untuk mencapai beberapa fungsi bisnis yang memberikan nilai tambah dalam bisnis. Tujuan adalah tugas-tugas seperti " proses penjualan," "menerima retur," atau "mengirimkan *order*". Pada penyelesaian tujuan, data dari sistem harus stabil untuk beberapa waktu.

### **II.2.7.3 Activity Diagram**

Menurut Satzinger, Jackson, & Burd (Satzinger, Jackson, & Burd, 2010, p. 141) , *activity diagram* adalah jenis diagram alur kerja yang menggambarkan kegiatan pengguna dan aliran sekuensial mereka.



Sumber : (Satzinger, Jackson, & Burd, 2010, p. 142)

**Gambar II.7 Notasi Activity Diagram**

Dalam activity diagram terdapat beberapa simbol yang digunakan, yaitu (Satzinger, Jackson, & Burd, 2010, p. 141):

- Synchronization bar*, Merupakan notasi yang digunakan untuk mengontrol pemisahan atau panyatuan dari jalur yang berurutan.
- Swimlane*, Merupakan suatu daerah persegi dalam activity diagram yang mewakili aktivitas-aktivitas yang diselesaikan oleh agen tunggal.
- Starting activity (pseudo)*, Merupakan notasi yang menandakan dimulainya sebuah aktivitas
- Transition arrow*, Merupakan garis penunjuk arah yang menggambarkan transisi dari suatu aktivitas dan arah dari suatu aktivitas.
- Activity*, Merupakan notasi yang menggambarkan suatu aktivitas
- Ending activity (pseudo)*, Merupakan notasi yang mendakan berakhirnya suatu aktivitas.

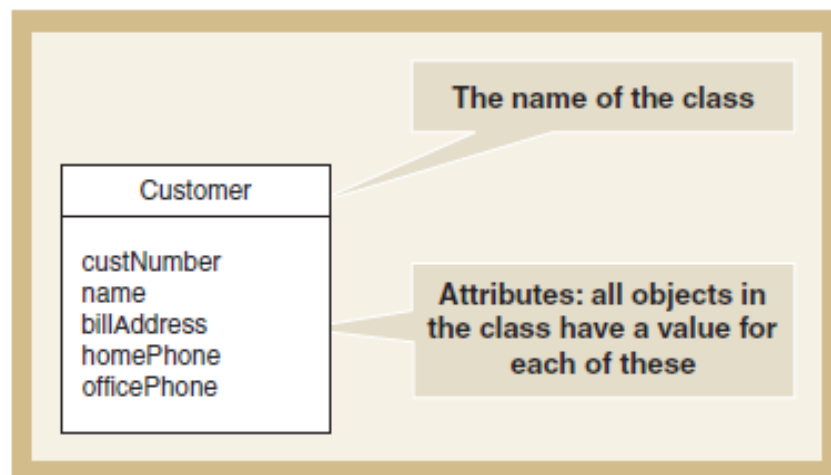
#### II.2.7.4 Class Diagram

Menurut Satzinger, Jackson, & Burd (Satzinger, Jackson, & Burd, 2010, p. 60), class diagram adalah model grafis yang digunakan dalam pendekatan object-oriented untuk menunjukkan kelas objek dalam sistem.

Menurut Satzinger, Jackson & Burd ada 3 tahapan dalam class diagram :

a. *Domain Class Diagram*

Dalam *domain class* memiliki 2 bagian. Bagian atas berisi nama *class*, dan bagian bawah berisi atribut dari kelas. Nama *Class* selalu dimulai dengan huruf kapital, dan nama atribut selalu diawali dengan huruf kecil (Satzinger, Jackson, & Burd, 2010, p. 187).



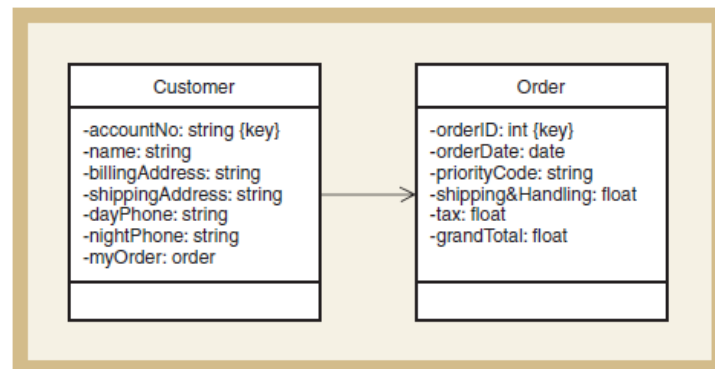
Sumber : (Satzinger, Jackson, & Burd, 2010, p. 187)

**Gambar II.8 Notasi Domain Class Diagram**

b. *First Cut Design Class Diagram*

First-cut design class diagram adalah pengembangan dari domain class diagram yang membutuhkan 2 tahap (Satzinger, Jackson, & Burd, 2010, p. 413):

- a. menentukan tipe data dari atribut kelas dan informasi initial value
- b. menambahkan panah navigation visibility. Navigasi visibilitas adalah prinsip desain di mana satu objek memiliki referensi ke obyek lain dan dengan demikian dapat berinteraksi dengan objek.



Sumber : (Satzinger, Jackson, & Burd, 2010, p. 414)

**Gambar II.9** Contoh *First Cut Design Class Diagram*

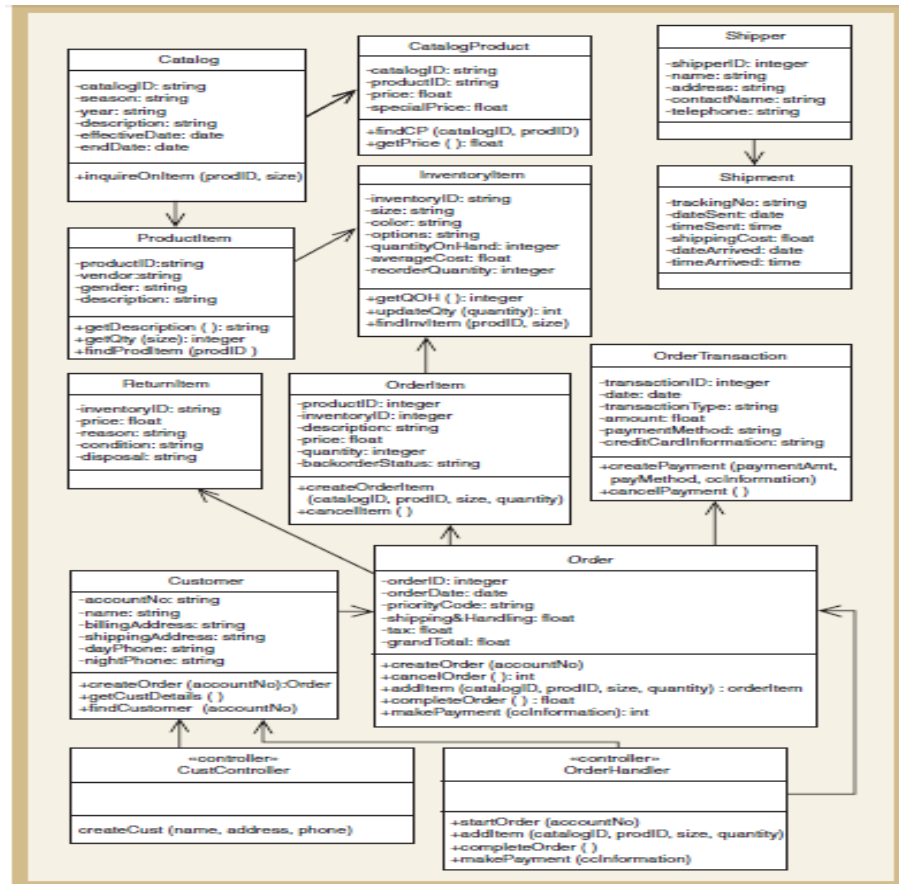
Berikut adalah beberapa panduan umum untuk menentukan panah *navigation visibility* (Satzinger, Jackson, & Burd, 2010, p. 415) :

1. One-to-many relationships yang menunjukkan hubungan superior/subordinate biasanya arah panahnya dari superior ke subordinate, misalnya, dari Orde ke OrderItem.
2. Mandatory relationships, di mana objek dalam satu kelas tidak dapat eksis tanpa objek dari kelas lain, arah panahnya dari class independen ke class dependen, misalnya, dari Customer kepada Order.
3. Ketika sebuah objek membutuhkan informasi dari objek lain, panah navigasi diperlukan, baik untuk menunjuk obyek itu sendiri atau induknya dalam suatu hirarki.
4. Panah Navigation mungkin juga dua arah.

*c. Update design class Diagram*

Update design class diagram dibuat dengan menambahkan method kedalam class. Tiga jenis method yang ditemukan di sebagian besar kelas: (1) constructor methods, (2) data get and set methods, dan (3) use case specific methods. Constructor methods menciptakan instances baru dari objek. Get and set methods mengambil dan memperbarui nilai atribut. Method yang ketiga adalah

use case specific methods - harus dimasukkan dalam the design class diagram (Satzinger, Jackson, & Burd, 2010, p. 457).



Sumber : (Satzinger, Jackson, & Burd, 2010, p. 457)

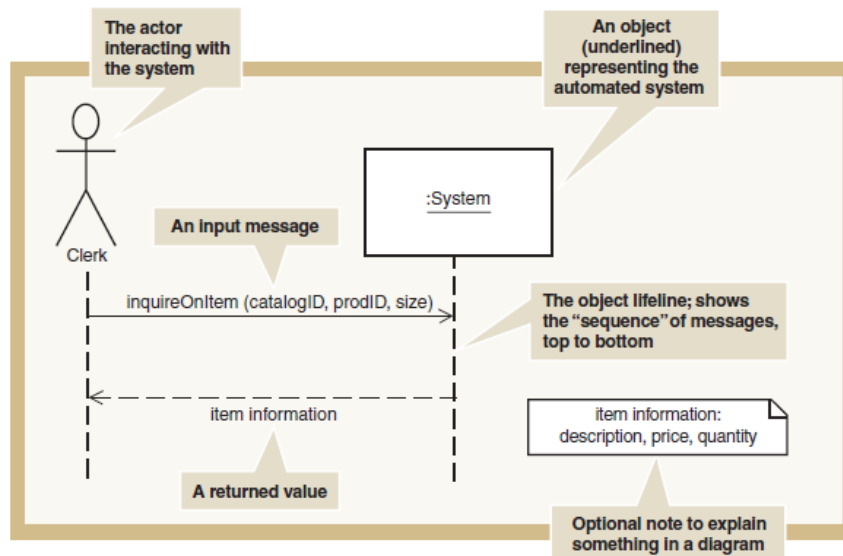
**Gambar II.10** Contoh *Updated Design Class Diagram*

## II.2.7.5 Sequence Diagram

Menurut Satzinger Jackson & Burd , ada 4 tahapan *dalam sequence diagram*:

### a. *System Sequence Diagram (SSD)*

SSD adalah diagram yang menunjukkan urutan pesan antara aktor eksternal dan sistem selama *use case* atau skenario (Satzinger, Jackson, & Burd, 2010, p. 242).

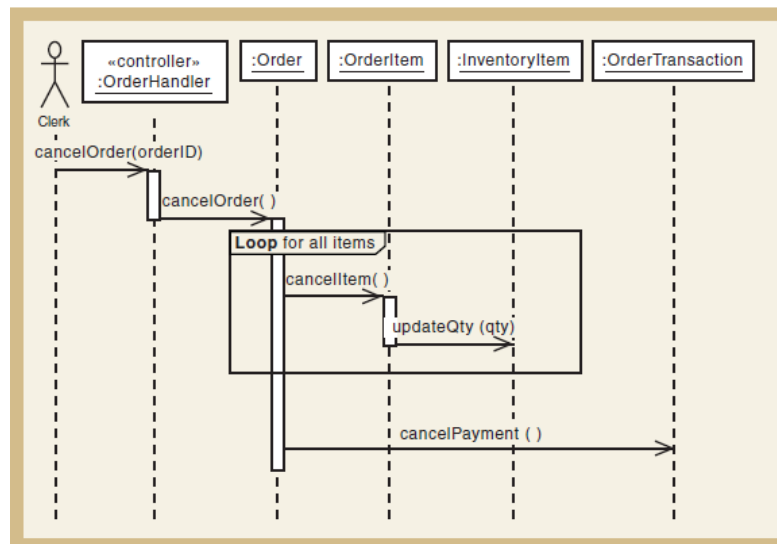


Sumber : (Satzinger, Jackson, & Burd, 2010, p. 253)

**Gambar II.11 Notasi *System Sequence Diagram* (SSD)**

*b. First Cut Sequence Diagram*

Membangun *first cut sequence diagram* dengan elemen-elemen dari System Sequence Diagram (SSD). Mengganti objek *:System* dengan *use case controller*. Kemudian menambahkan objek lain yang diperlukan *use case*. Langkah selanjutnya adalah menentukan *message*, termasuk objek yang harus menjadi sumber dan tujuan dari setiap *message* untuk mengumpulkan semua informasi yang diperlukan (Satzinger, Jackson, & Burd, 2005, p. 316).



Sumber (Satzinger, Jackson, & Burd, 2010, p. 439)

**Gambar II.12 Contoh First Cust Sequence Diagram**

c. *Three Layer Design Sequence Diagram*

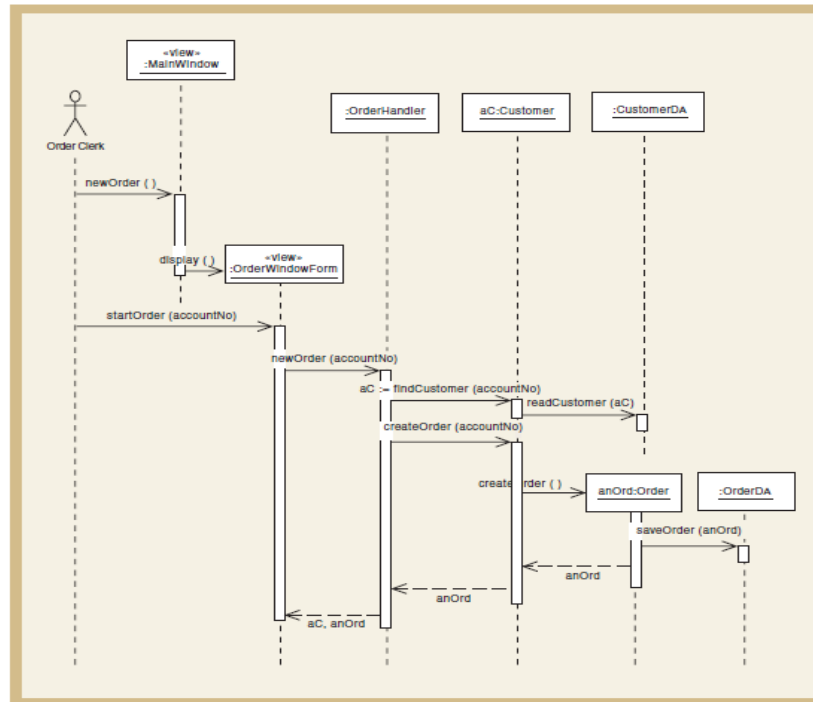
Three layer design sequence diagram merupakan gambaran lengkap dari sequence diagram dan juga pengembangan dari first cust sequence diagram dengan menambahkan layer-layer berikut (Satzinger, Jackson, & Burd, 2005, p. 320):

1. *View layer*

*View Layer* melibatkan interaksi manusia-komputer dan membutuhkan merancang *user interface* untuk setiap *use case*. Desain *user interface* adalah salah satu aktivitas utama dari disiplin desain UP (Satzinger, Jackson, & Burd, 2005, p. 320).

2. *Data Access Layer*

Prinsip pemisahan tanggung jawab juga berlaku untuk *data access layer*. Desain Multilayer penting untuk mendukung jaringan multitier di mana database pada satu server, logika bisnis pada server lain, dan *user interface* pada beberapa klien desktop. Cara baru merancang sistem yang tidak hanya menciptakan sistem yang lebih kuat, tetapi juga sistem yang lebih fleksibel (Satzinger, Jackson, & Burd, 2005, p. 322).



Sumber (Satzinger, Jackson, & Burd, 2010, p. 454)

**Gambar II.13 Contoh *Three Layer Design Sequence Diagram***

## II.2.8 Metode Pengujian

Adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-compile ulang.

### II.2.8.1 Pengujian UAT (User Acceptance Testing)

*User Acceptance Testing* (UAT) adalah proses untuk mendapatkan konfirmasi bahwa sebuah sistem memenuhi yang disepakati persyaratan. Sebuah *Subject Matter Expert* (SME), lebih baik pemilik atau klien dari benda yang diuji, memberikan konfirmasi tersebut setelah pengadilan atau diperiksa. Dalam pengembangan perangkat lunak, UAT adalah salah satu tahap akhir proyek dan sering terjadi sebelum klien atau pelanggan menerima sistem baru. Pengguna



sistem melakukan tes ini, yang pengembang berasal dari kontrak klien atau pengguna spesifikasi kebutuhan.

## **II.2.9 Perangkat Lunak Pendukung**

Aplikasi ini dibuat dengan bahasa pemrograman PHP menggunakan *Bootstrap* dan menggunakan *framework Codeigniter* dengan dukungan dari *Web browser*, sedangkan untuk *database* menggunakan MySQL

### **II.2.9.1 HTML**

*HyperText Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman web. HTML merupakan pengembangan dari standar pemformatan dokumen text, yaitu *Standard Generalized Markup Language* (SGML). HTML pada dasarnya merupakan dokumen ASCII atau text biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

### **II.2.9.2 CSS**

CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu *website*, baik tata letaknya, jenis, huruf, warna dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk memformat halaman *web* yang ditulis dengan *HTML* atau *XHTML*.

### **II.2.9.3 Javascript**

*JavaScript* adalah Bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen *HTML*. Dimana objek tersebut dapat berupa suatu *window*, *frame*, *URL*, dokumen, *form*, *button*, atau item yang lain. Yang semua itu mempunyai properti yang saling berhubungan dengannya, dan masing-masing memiliki nama, lokasi, warna nilai, dan atribut lain.

### **II.2.9.4 MySQL**

*MySQL* merupakan *Database Management Sistem* (DBMS) tools open source yang mendukung *multiuser*, *multithreaded*, *populer* dan *free*. MySQL

dimiliki oleh perusahaan asal Swedia, MySQL AB, di mana perusahaan tersebut memegang seluruh hak cipta atas kode sumbernya. MySQL AB didirikan oleh David Axmark, Allan Larsson, dan Michael “Monty” Widenius.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*.

Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain:

1. *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source*

MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

#### 6. *Command and Functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

#### 7. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

#### 8. *Scalability and Limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

#### 9. *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix Socket (UNIX), atau *Named Pipes* (NT).

#### 10. *Localisation*

MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

#### 11. *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

#### 12. *Clients and Tools*

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi database, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

#### 13. *Structure Table*

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.

### II.2.9.5 PHP

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan *skript* yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilis kode sumber ini menjadi sumber terbuka, maka banyak programer yang tertarik untuk ikut mengembangkan PHP. Pada Nopember 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: *Hypertext Preprocessing*. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari *interpreter* PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek.

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.

2. *Web Server* yang mendukung PHP dapat ditemukan dimana – mana dari mulai *apache*, IIS, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis – milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman website oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. Selanjutnya *webserver* akan mencarikan berkas yang diminta dan menampilkan isinya di browser. Browser yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya.

Lalu bagaimana apabila yang dipanggil oleh user adalah halaman yang mengandung script PHP ? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa tipe file yang diminta user. Jika tipe file yang diminta adalah PHP, maka akan memeriksa isi script dari halaman PHP tersebut. Apabila dalam file tersebut tidak mengandung script PHP, permintaan user akan langsung ditampilkan ke browser, namun jika dalam file tersebut mengandung script PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan script-script PHP dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke browser user. selain itu dalam pembuatan website menggunakan PHP sekarang banyak sekali ditemui CMS :seperti *Jomla*, *wordpress*, Framework : seperti *laravel*, *Codeigniter*, *YII*, dan pasti semuanya menggunakan OOP. Jadi banyak hal yang bisa kita pelajari disini.

### II.2.9.6 Codeigniter

*Codeigniter* merupakan sebuah *framework* PHP dengan konsep MVC (*Model, View, Controller*) yang dapat memudahkan *developer* (pengembang) untuk membuat aplikasi *web* dengan cepat. Karena konsep MVC ini memisahkan antara *query* ke *database* (*Model*) dengan tampilan (*View*) serta logika pemrograman (*Controller*). Konsep MVC ini biasanya diterapkan pada bahasa pemrograman berorientasi objek (OOP). Oleh karena itu kode PHP nantinya akan ditulis dengan teknik OOP dan pattern MVC.

*Model View Controller* (MVC) merupakan salah satu contoh dari *Architectural Design Pattern*. Pada saat ini banyak *framework* yang didedikasikan pemrograman berorientasi objek.

### II.2.9.7 Bootstrap

*Bootstrap* merupakan *framework open-source* berupa *html, css* dan *javascript* yang dikembangkan oleh team di Twitter. *Bootstrap* dibuat untuk mendukung teknologi terbaru *web standard* yaitu *html5* dan *css3*. Tidak hanya untuk komputer, *Bootstrap* juga memiliki kemampuan untuk menampilkan halaman *web* versi *mobile* (*handphone, smartphone, tablet, dll*).

Dengan *bootstrap* kita bisa membuat *web* statis maupun dinamis lebih cepat karena sudah dimanjakan oleh *component* dan *framework* siap pakai yang tersedia dan tentunya mudah untuk dikembangkan kedepannya.