

BAB II

TINJAUAN PUSTAKA

2.1 Career centre

Career centre merupakan pusat yang menyimpan berbagai informasi mengenai pekerjaan, terutama bagi alumni dari sebuah perguruan tinggi yang sedang mencari pekerjaan. Di dalam *career centre* terdapat informasi mengenai *job placement centre* dan *tracer study*.

2.1.1 Job placement centre

Job placement centre merupakan informasi mengenai data-data pekerjaan serta pelacakan mengenai penempatan kerja dari sejumlah alumni sebuah perguruan tinggi.

2.1.2 Tracer study

Tracer study merupakan Pencarian data alumni dari sebuah perguruan tinggi untuk Penyimpanan informasi data-data yang dibutuhkan serta untuk membantu komunikasi antar alumni guna menunjang karir pekerjaan dengan tujuan untuk perbaikan kurikulum dan proses pendidikan.

2.1.2.1 Tujuan kegiatan *tracer study*

Tujuan utama kegiatan Tracer Study adalah mengidentifikasi kualitas lulusan dari suatu perguruan tinggi . Secara khusus, kegiatan ini bertujuan untuk:

1. Mengidentifikasi profil kompetensi alumni.
2. Mengetahui relevansi kurikulum yang diterapkan setiap Jurusan perguruan tinggi dengan kebutuhan pasar kerja.
3. Memberi masukan pada pengembangan kurikulum untuk meningkatkan akreditasi.
4. Memperoleh gambaran kompetensi yang dibutuhkan pengguna alumni.

2.1.2.2 Informasi yang harus diperoleh pada *tracer study*

Untuk keperluan tracer study bagi alumni , diperlukan beberapa informasi sebagai berikut:

1. Data individu yang meliputi data pribadi alumni.
2. Informasi dari alumni ke alumni tentang informasi kerja, jenis pekerjaan, kondisi pekerjaan.
3. Media yang digunakan dalam berkomunikasi antar alumni dan alumni, antara alumni-almamater, alumni-*stakeholder* dll.
4. Jaringan komunikasi antara alumni di lingkungan perguruan tinggi maupun luar lingkungan.

2.1.3 *Stakeholder*

Stakeholder merupakan seseorang atau sekumpulan orang yang secara langsung atau tidak langsung yang menopang sebuah organisasi karena hal ini mempengaruhi ataupun dipengaruhi oleh kegiatan organisasi tersebut. [Ringgo, 2008]

Beberapa kunci dari *stakeholder* di antaranya adalah sebagai berikut:

1) *Owners*

Owner merupakan seorang entrepreneur atau lebih yang menghasilkan suatu ide tentang suatu produk atau layanan. Seorang entrepreneur selalu bersikap kritis dalam membangun suatu bisnis baru karena mereka membuat produk baru yang berdasarkan selera konsumen. Selain itu, *owner* juga memiliki definisi sebagai seseorang atau sekumpulan orang yang memiliki wewenang khusus untuk memegang, menggunakan, menikmati, menyampaikan, mengirim dan mengatur suatu *asset* atau *property*.

Banyak orang yang ingin membuat suatu bisnis hanya karena mereka berharap agar mereka dihargai karena hasil jerih payah mereka sendiri. Selain itu, tidak sedikit pula mereka yang disebabkan karena mereka terantang atau bergengsi tinggi untuk membangun suatu bisnis sendiri. Kebanyakan dari *owner* dari suatu bisnis yang setuju bahwa

semua hal itu merupakan karakteristik dari orang-orang untuk temotivasi dalam memulai karirnya sendiri.

2) Creditors

Creditor adalah suatu institusi yang menyediakan dana untuk kemudiakan dipinjamkan kepada suatu perusahaan yang membutuhkan. Suatu perusahaan yang meminjam dari suatu *creditor* harus membayar bunga dari pinjaman mereka. Suatu *creditor* tidak sembarangan dalam meminjakan bantuan finansial kepada suatu perusahaan. Mereka hanya meminjamkan bantuan finansial tersebut kepada suatu perusahaan yang diyakininya sanggup untuk mengembalikannya beserta dengan bunganya.

3) Employee

Employee merupakan sebagian orang yang mengatur secara langsung dari suatu perusahaan. Suatu *employee* biasanya terikat kontrak kerja dari perusahaan dimana ia bekerja. Seorang *employee* yang bertanggung jawab untuk mengatur tugas-tugas dari *employee* lainnya dan membuat keputusan dalam bisnis biasa disebut sebagai seorang manajer. Maju mundurnya suatu perusahaan sangat dipengaruhi terhadap keputusan dari seorang menajer.

4) Suppliers

Supplier merupakan salah satu stakeholder yang cukup penting peranannya. Mereka menyediakan bahan-bahan di mana bahan tersebut sangat diperlukan oleh suatu perusahaan untuk menghasilkan produk mereka.

5) Customers

Suatu perusahaan tidak akan bertahan lama tanpa ada seorang *customer*. *Customer* merupakan target dari suatu perusahaan untuk menjualkan hasil produksinya. Untuk menarik seorang *customer*, suatu perusahaan harus menyediakan produk dan layanan yang terbaik serta harga yang bersahabat.

2.2 Algoritma pencarian (*searching algorithm*)

2.2.1 Pengertian Algoritma Pencarian (Searching Algorithm)

algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).^[9]

2.2.2 Pencarian berurutan (*sequential searching*)

Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut :

Data yang ada dibandingkan satu per satu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan. Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke- i dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali juga.

Algoritma pencarian berurutan dapat dituliskan sebagai berikut :

1. $i \leftarrow 0$
2. ketemu $\leftarrow$ false
3. Selama (tidak ketemu) dan ($i \leq N$) kerjakan baris 4
4. Jika ($\text{Data}[i] = x$) maka ketemu $\leftarrow$ true
5. Jika tidak $i \leftarrow i + 1$
6. Jika (ketemu) maka i adalah indeks dari data yang dicari
7. jika tidak data tidak Ditemukan

Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian sekuensial .

Listing Program :

```
int SequentialSearch(int x)
{
    int i = 0;
    bool ketemu = false;
    while ((!ketemu) && (i < Max)){
81
        if(Data[i] == x)
            ketemu = true;
        else
            i++;
    }
    if(ketemu)
        return i;
    else
        return -1;
}
```

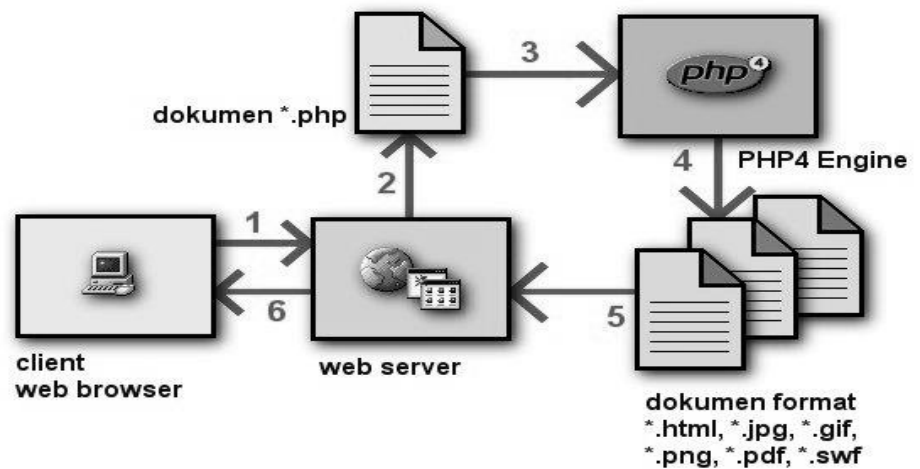
Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai -1.

2.3 Bahasa Pemrograman Yang Digunakan

2.3.1 PHP (*Profesional Home Page Hypertext Preprocessor*)

PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Bahasa pemrograman PHP bisa dikategorikan sebagai bahasa pemrograman web yang berjalan pada bagian server (*server side*), yaitu bagian yang bertanggung jawab atas cara kerja aplikasi. Dengan PHP ini pula seorang *programmer* web dapat membuat *back end storage* suatu *website* yang berguna untuk mengatur cara penyimpanan data. Penyimpanan data merupakan materi yang cukup kompleks dalam pembangunan aplikasi. Karena kecepatan, kebutuhan, dan keamanan data merupakan faktor kritis dalam aplikasi. Ada banyak solusi database yang tersedia di pasaran. Pada umumnya, database yang digunakan bertipe relasional (*Relational Database Management System* - RDBMS). Manajemen data dilakukan dengan bahan SQL (*Standart Query Language*).

PHP bersifat *open source*, karenanya gratis dan bebas. Sedangkan untuk masalah *database*, biasanya adalah MySQL yang dijalankan *webserver apache* diatas sistem operasi linux. Semuanya gratis dan bebas



Gambar 2.1. Alur kerja PHP

2.3.2 HTML (*HyperText Markup Language*)

HyperText Markup Language (HTML) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan format *hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format ASCII normal sehingga menjadi *homepage* dengan perintah-perintah HTML.

2.3.3 CSS (*Cascading Style Sheet*)

Merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Sama halnya *styles* dalam

aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa file. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna body teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, margin kanan, margin atas, margin bawah, dan parameter lainnya.

CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

2.3.4 jQuery

jQuery adalah pustaka Javascript yang bersifat terbuka (*opensource*) dan menekankan pada interaksi antara *javascript* dan HTML. Pustaka ini dirilis pada Januari 2006 di BarCamp NYC oleh John Resig dan berlisensi ganda di bawah Lisensi MIT dan GPL. Situs resmi jQuery yaitu di <http://jquery.com>.

Untuk menggunakan pustaka jQuery tersebut, pengguna harus mendownload file javascript dari jQuery yang berfungsi sebagai *prototype* fungsi-fungsi yang akan digunakan pada web.

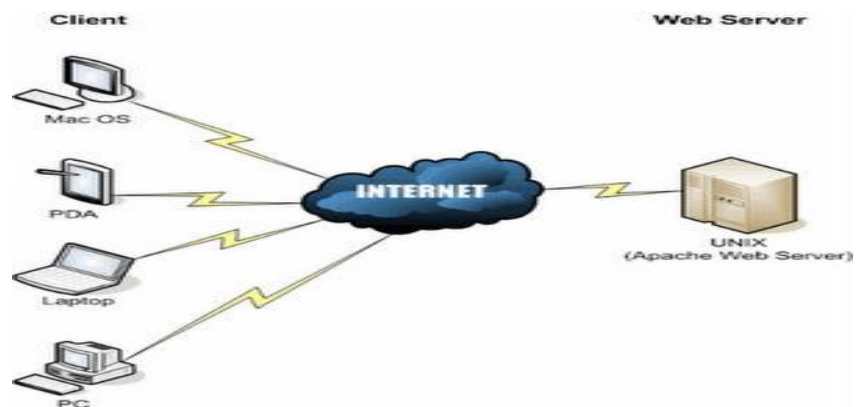
2.4 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

2.4.1 Webserver Apache

Web server atau server web merupakan perangkat lunak yang ditempatkan pada komputer jenis apapun yang sesuai dengan spesifikasi teknis minimal yang dianjurkan oleh perangkat lunak tersebut yang mampu menerima permintaan HTTP/HTTPS dari klien melalui media browser (IE , Firefox , Chrome, dll) dan mengirimkan kembali dalam bentuk halaman-halaman website yang umumnya secara standar adalah *Hypertext Markup Language* (.html).

Tugas utama *webserver* adalah menghasilkan halaman web yang benar kepada pengguna berdasarkan kode PHP yang dituliskan oleh pembuat halaman web.



Gambar 2.2 Hubungan antara *device*, internet, dan *webserver*

Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunaanya

2.4.2 Pengertian MySQL

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured 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 oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

2.4.3 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

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

6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.4.4 Kesenambungan antara PHP dan MySQL

Penggunaan PHP dan MySQL dapat menjadikan dan memudahkan untuk pembuatan aplikasi secara gratis dan stabil (dikarenakan banyak komunitas developer PHP dan MySQL yang ber-kontribusi terhadap bugs).

2.5 Alat Perancangan Yang Digunakan

2.5.1 Unified Modelling Language (UML)

¹UML adalah metode untuk mendeskripsikan arsitektur desain suatu sistem secara detail dan terperinci. Menggunakan *blueprint* dapat memudahkan pengembang aplikasi untuk membuat atau merawat sistem, dan memastikan sistem agar tetap sesuai dengan perubahan dari persyaratan sistem yang berubah.

¹ Boogs, Wendy and Boogs, Michael, (1999) Mastering UML with Rational Rose, Alameda:SYBEX. (9)

UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang kita sebut sebuah model.

Diagram-diagram yang tersedia pada UML versi 1.4 ada 9 diagram yang terbagi atas 2 kategori, yaitu :

1. *Behavioral diagram* seperti *use case diagram*, *sequence diagram*, dan *activity diagram*.
2. *Structural diagram* seperti *class diagram*, dan *deployment diagram*.

2.5.2 Use Case Diagram

Use case diagram adalah deskripsi fungsi dari sebuah sistem berdasarkan sudut pandang dari para pengguna sistem. *Use case* mendefinisikan apa yang dilakukan oleh sistem dan elemen – elemennya. Komponen notasi dasar yang dipunyai oleh *use case diagram* adalah *actor*, *use case*, dan *association*. Usecase diagram mempunyai stereotype, yaitu sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan stereotype digunakan simbol “<<” diawalnya dan ditutup dengan “>>” diakhirnya. Terdapat 2 *stereotype* yang paling sering digunakan dalam *use case diagram* yaitu <<extend>> dan <<include>>. <<extend>> digunakan untuk

menunjukkan bahwa satu *usecase* merupakan tambahan fungsional dari *usecase* yang lain jika syarat atau kondisi tertentu dipenuhi.

Tabel 2.1 notasi pada *usecase diagram*

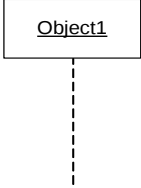
actor	Aktor adalah pengguna sistem. Aktor tidak terbatas hanya manusia saja, sebuah sistem yang berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, dapat disebut aktor.	
se Case	Use case digambarkan sebagai lingkaran elips dengan nama use case dituliskan didalam elips tersebut	
ssociation	Association digunakan untuk menghubungkan aktor dengan usecase. Association digambarkan dengan sebuah garis yang menghubungkan antara actor dengan use case.	

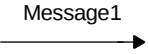
2.5.3 *Sequence Diagram*

Sequence diagram mendokumentasikan komunikasi atau interaksi antar *class – class*. Diagram ini menunjukkan sejumlah

objek dan *message* (pesan) yang diletakkan diantara objek-objek didalam *use case*. Berikut adalah notasi-notasi nya :

Tabel 2.2 Notasi pada *sequence diagram*

Object	Objek merupakan instance dari sebuah class dan ditulis tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma	
Actor	Aktor juga dapat berkomunikasi dengan objek, maka aktor juga dapat diurutkan sebagai kolom. Simbol actor sama dengan simbol pada aactor use case diagram	
Lifeline	Lifeline mengindikasikan keberadaan sebuah objek dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah objek.	
Activation	Activation dinotasikan sebagai sebuah kotak segi empat yang digambarkan pada sebuah lifeline. Activation mengindikasikan sebuah objek yang akan melakukan sebuah aksi.	

Message	Message, digambarkan dengan anak panah horizontal antara activation. Message mengindikasikan komunikasi antara objek-objek	
----------------	--	---

2.5.4 Activity Diagram

Activity Diagram secara grafis menggambarkan rangkaian aliran aktivitas baik proses bisnis atau *use case*. Diagram ini juga dapat digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi di eksekusi, dan memodelkan hasil dari *action* tersebut. Berikut adalah sebuah contoh notasi *activity diagram* :

Tabel 2.3 Notasi pada activity diagram

Keterangan	Simbol
Titik awal	
Titik akhir	
Activity	
Pilihan untuk mengambil keputusan	
Fork, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua atau lebih kegiatan menjadi satu	

2.5.5 *Class Diagram*

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan dan interaksi diantara mereka. Sebuah *class* memiliki tiga area pokok :

1. Nama, merupakan nama dari sebuah kelas
2. Atribut, merupakan properti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada objek dari *class*
3. Operasi, adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang dapat dilakukan oleh *class* lain terhadap sebuah *class*.

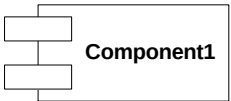
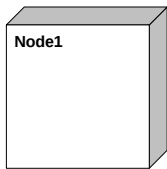
Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, *class* yang tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, *class* yang dapat dipanggil oleh seluruh *class* yang ada.
4. *Package*, *class* yang hanya dapat dipanggil oleh *instance* sebuah *class*

2.5.6 Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik serta merupakan bagian-bagian *software* yang berjalan pada *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment diagram* dapat digunakan pada bagian-bagian awal maupun akhir proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. Berikut adalah notasi - notasi yang digunakan pada *deployment diagram* :

Tabel 2.4 notasi pada *deployment diagram*

Component	Pada deployment diagram, komponen - komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka	
Node	Node menggambarkan bagian-bagian perangkat keras dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.	
Association	Sebuah association digambarkan sebaga sebuah garis ang menghubungkan dua node yang	

	mengindikasikan jalur komunikasi elemen-elemen perangkat keras.	
--	---	--

2.6 Model Pengembangan Perangkat Lunak

2.6.1 Pengertian Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem ^[2]

Pada definisi ini, ada dua istilah kunci yaitu:

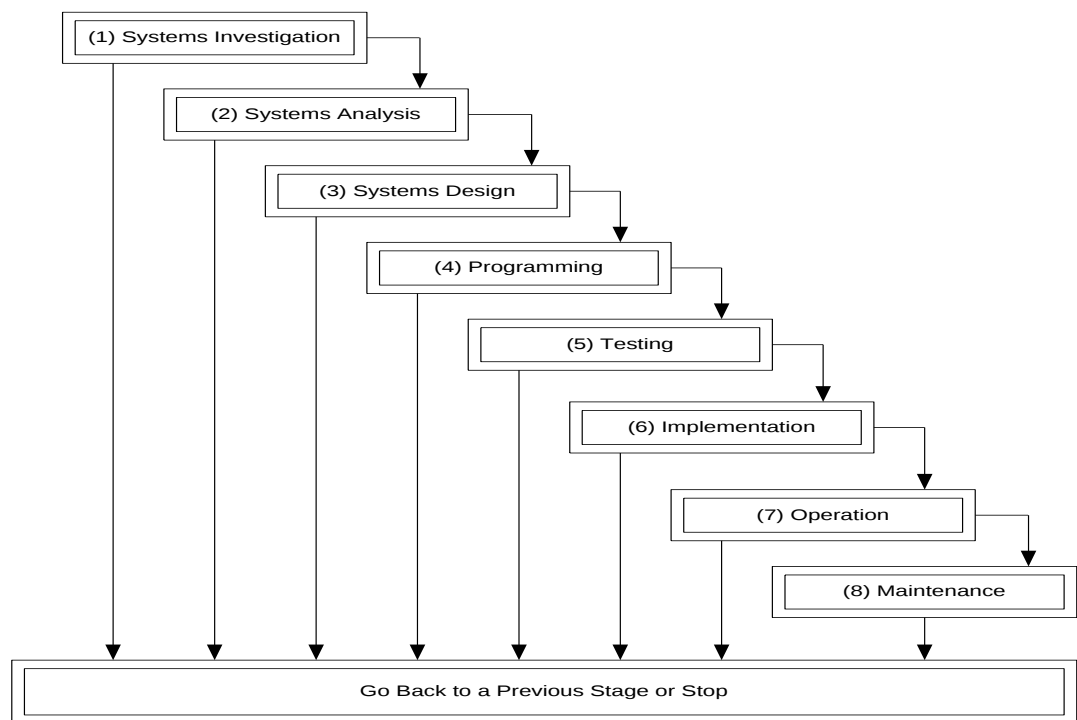
- a. Disiplin rekayasa, perekayasa membuat suatu alat bekerja. Menerapkan teori, metode, dan alat bantu yang sesuai, selain itu mereka menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan-batasan ini.
- b. Semua aspek produksi perangkat lunak, rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode dan teori untuk mendukung produksi perangkat lunak.

Secara umum, rekayasa perangkat lunak memakai pendekatan sistematis dan terorganisasi terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Namun demikian, rekayasa ini sebenarnya mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan. Pengembangan informal sangat cocok untuk pengembangan sistem *e-commerce web* membutuhkan gabungan keahlian perangkat lunak dan perancangan grafis.

2.6.2 Daur hidup pengembangan perangkat lunak

Systems development life cycle (SDLC) adalah metode pengembangan sistem tradisional yang digunakan oleh banyak organisasi sekarang ini. SDLC merupakan kerangka kerja terstruktur yang terdiri dari proses-proses sekuensial yang membangun sistem informasi.

Pada metode pengembangan sistem SDLC, tugas-tugas pada satu tahapan diselesaikan sebelum pengembangan dilanjutkan ke tahapan berikutnya. Apabila dibutuhkan, pengembang bisa bergerak maju mundur dari berbagai tahapan seperti ditunjukkan oleh gambar 2.3.



Gambar 2.3 Siklus Hidup Pengembangan Sistem (SDLC) 8 level

Tahapan-tahapan dalam SDLC adalah sebagai berikut :

1. Investigasi sistem (*systems investigation*)

Pada tahap ini, dilakukan studi kelayakan untuk menentukan kemungkinan suksesnya proyek pengembangan sistem yang diajukan dan menaksir kelayakan proyek dari segi teknis, ekonomis, dan perilaku.

2. Analisis sistem (*systems analysis*)

Tahapan ini mendefinisikan masalah yang dihadapi, mengenali penyebabnya, menentukan solusinya, dan mengenali kebutuhan informasi. Untuk memahami sifat program yang dibangun, perancang harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan.

3. Perancangan sistem (*systems design*)

Hasil dari fase perancangan sistem adalah perancangan teknis yang menentukan hal-hal berikut:

- a. Output, input, dan antar muka (*interface*) sistem
- b. Perangkat keras, piranti lunak, database, telekomunikasi, personel, dan prosedur.
- c. Bagaimana setiap komponen berintegrasi.

Perancangan sistem mencakup pembuatan model logikal dan fisik dari komponen-komponen data sistem.

4. Pemrograman (*programming*)

Spesifikasi desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca (bahasa pemrograman) untuk mengurangi kompleksitas dan meningkatkan kesinambungan (*modularity*). Pengecekan atas kesalahan (*error*) yang mungkin terjadi perlu dilakukan sebagai tindak lanjut.

5. Pengetesan (*testing*)

Testing dilakukan untuk memastikan apakah kode komputer yang ditulis memberikan hasil aktual yang diinginkan di bawah kondisi-kondisi yang diberikan. Proses pengujian juga berfokus pada logika internal perangkat lunak dan eksternal fungsional, yaitu untuk mendeteksi adanya *error* (“*bugs*”) dalam kode komputer.

6. Implementasi (*implementasi*)

Implementasi merupakan proses konversi (*conversion*) sistem lama menjadi sistem yang baru. Setelah sistem lama dibongkar dan sistem baru diinstal, perubahan terjadi dan fase operasi dimulai.

7. Operasi (*operation*)

Setelah konversi, sistem baru akan beroperasi dalam periode tertentu. Saat operasi sistem baru telah stabil, audit dilakukan guna menaksir kapabilitas sistem dan menentukan apakah penggunaannya tepat.

8. Pemeliharaan (*maintenance*)

Perangkat lunak akan mengalami perubahan setelah berada di tangan klien. Perubahan akan terjadi karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan dalam lingkungan eksternalnya atau karena pelanggan membutuhkan perkembangan fungsional unjuk kerja.

2.7 Interaksi Manusia dan Komputer

Interaksi manusia dan komputer adalah disiplin ilmu yang mempelajari hubungan antara manusia dan komputer yang meliputi perancangan, evaluasi, dan implementasi antarmuka pengguna komputer agar mudah digunakan oleh manusia. Sedangkan interaksi manusia dan komputer sendiri adalah serangkaian proses, dialog dan kegiatan yang dilakukan oleh manusia untuk berinteraksi

dengan komputer secara interaktif untuk melaksanakan dan menyelesaikan tugas yang diinginkan.

Tujuan dari IMK adalah untuk menghasilkan sistem yang bermanfaat (usable) dan aman (safe), artinya sistem tersebut dapat berfungsi dengan baik. Sistem tersebut bisa untuk mengembangkan dan meningkatkan keamanan (safety), utilitas (utility), ketergunaan (usability), efektifitas (effectiveness) dan efisiensinya (efficiency). Sistem yang dimaksud konteksnya tidak hanya pada perangkat keras dan perangkat lunak, tetapi juga mencakup lingkungan secara keseluruhan, baik itu lingkungan organisasi masyarakat kerja atau lingkungan keluarga. Sedangkan Ketergunaan (usability) disini dimaksudkan bahwa sistem yang dibuat tersebut mudah digunakan dan mudah dipelajari baik secara individu ataupun kelompok. Utilitas mengacu kepada fungsionalitas sistem atau sistem tersebut dapat meningkatkan efektifitas dan efesiensi kerjanya.

Dari suatu ilmu pengetahuan tentang perspektif dari komputer, Maksud dari interaksi dan secara rinci pada interaksi antara satu atau lebih manusia dan satu atau lebih komputasi mesin. Yang mengontrol sebuah mesin tersebut adalah seseorang, dengan menggunakan suatu program grafik interaktif yang ada pada suatu stasiun kerja. Disini bisa kita lihat bahwa ada bermacam-macam arti dengan apa yang dimaksud dengan interaksi, manusia, dan mesin, Makanya, kita tidak mungkin untuk meniadakan bagian dari Interaksi Manusia dan Komputer,

meskipun demikian kita ingin mengidentifikasi ruang lingkupnya, serta topik lain yang lebih umum tentang Interaksi Manusia dan Komputer.

Selanjutnya, dalam berinteraksi dengan komputer, para pemakai pertama kali akan berhadapan dengan perangkat keras komputer. Untuk sampai pada isi yang ingin disampaikan oleh perangkat lunak, pemakai dihadapkan terlebih dahulu dengan seperangkat alat seperti papan ketik (keyboard), monitor, mouse, joystick, dan lain-lain. Pemakai harus dapat mengoperasikan seperangkat alat tersebut. Selanjutnya, pemakai akan berhadapan dengan macam-macam tampilan menu, macam-macam perintah yang terdiri dari kata atau kata-kata yang harus diketikkannya, misalnya save, copy, delete, atau macam-macam ikon. Peralatan, perintah, ikon dan lain-lain yang disebutkan di atas dikenal dengan nama interface (antarmuka). Interface ini merupakan lapisan pertama yang langsung bertatap muka dengan pemakai.

Sebab Interaksi Manusia dan Komputer mempelajari antara suatu manusia dan suatu mesin didalam sebuah komunikasi, yang mendukung pengetahuan dari kedua-duanya antara sisi mesin dan sisi manusia. Pada sisi mesin, teknik dalam komputer grafik, sistem operasi, bahasa programan, dan lingkungan pengembangannya. Sedangkan pada sisi manusia antara lain, teori komunikasi, disain industri dan grafis, linguistik, ilmu-ilmu sosial, psikologi, dan tujuan manusia relevan. Dan tentu saja perancangan sistem dan metoda lain yang sesuai.

Komputer bisa mempengaruhi tingkah laku seseorang, baik itu disadari maupun tidak disadari oleh seseorang tersebut. Ia telah membuat informasi lebih mudah untuk diakses. Video game juga telah merubah perilaku seseorang lebih dari apa pun. Anak-anak yang bermain video game terlalu berlebihan mulai percaya bahwa apa yang ada didalam permainan itu adalah benar. Beberapa orang menjadi kecanduan komputer sehingga mereka harus menggunakannya setiap hari seperti obat. Komputer telah mengubah kehidupan mereka sehingga mereka dapat menggunakannya sebanyak mungkin.

Prinsip kerja dalam sebuah computer adalah masukan, proses, keluaran (input, process, output). Kepada computer kita memberikan data masukan, yang biasanya berupa angka maupun deretan karakter, yang kemudian akan diolah (diproses) oleh Komputer menjadi keluaran yang diinginkan atau diharapkan pengguna. Ketika seseorang bekerja dengan sebuah Komputer, ia akan melakukan interaksi dengan Komputer menggunakan cara-cara tertentu. Cara yang umum digunakan adalah bahwa pengguna memberikan suatu perintah kepada Komputer, dan computer menanggapi dengan mencetak atau menuliskan tanggapan itu pada layar tampilan. Dari “masukan, proses keluaran” diatas pengguna memang tidak tahu menahu (atau tidak ingin tahu) dengan apa (proses) yang sesungguhnya terjadi didalam sistem Komputer. Dengan kata lain, lewat masukan dan keluaranlah pengguna dan Komputer saling berinteraksi.