

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Perancangan Sistem**

Perancangan sistem merupakan tahapan selanjutnya setelah analisa sistem, mendapatkan gambaran dengan jelas tentang apa yang dikerjakan pada analisa sistem, maka dilanjutkan dengan memikirkan bagaimana membentuk sistem tersebut

Perancangan sistem adalah suatu fase dimana diperlukan suatu keahlian perancangan untuk elemen-elemen komputer yang akan menggunakan sistem yaitu pemilihan peralatan dan program komputer untuk sistem yang baru. (Andri Kristanto, 2008)

#### **2.2. Konsep Sistem**

Istilah sistem bukanlah hal yang asing bagi kebanyakan orang. Suatu sistem terdiri dari sistem-sistem bagian (*subsistem*) dan masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil lagi. Secara umum, sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan sasaran tertentu. ( Agus Mulyono, 2009).

##### **2.2.1 Definisi Sistem**

Untuk dapat memahami atau dapat mendefinisikan sebuah sistem terdapat pendekatan yang dapat digunakan untuk menerangkan yaitu dengan pendekatan.

## 1. Prosedur

Yaitu : “suatu jaringan kerja dari prosedur-prosedur yang berupa urutan kegiatan yang saling berhubungan, berkumpul bersama-sama untuk mencapai tujuan tertentu.

Prosedur adalah rangkaian operasi klerika (tulis menulis), yang melibatkan beberapa orang di dalam suatu/lebih departemen yang digunakan untuk menjamin penanganan yang seragam dari tranaksi-transaksi yang terjadi serta untuk menyelesaikan suatu kegiatan tertentu.

Urutan kegiatan digunakan untuk menjelaskan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya.

## 2. Komponen/elemen

Yaitu kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu.

Yang dimaksud dengan sistem adalah sebagai sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan *input* serta menghasilkan *output* dalam proses transformasi teratur. (Agus Mulyono,2009)

### 2.3 Aplikasi Web

Aplikasi Web adalah aplikasi yang disimpan dan dieksekusi di lingkungan *web server*. Setiap permintaan yang dilakukan oleh user melalui aplikasi klien (*web browser*) akan direspon oleh aplikasi web dan hasilnya akan dikembalikan lagi ke hadapan user. Dengan aplikasi web, halaman yang ditampilkan di layar

*web browser* dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dimasukan oleh user. (Budi Raharjo, 2010).

Unsur-unsur dalam web adalah sebagai berikut :

### 1. Internet

Internet merupakan kepanjangan dari *Interconnection Networking*, internet merupakan rangkaian jaringan terbesar di dunia dimana semua jaringan berada pada semua organisasi dihubungkan dengan suatu jaringan terbesar melalui telphone, stelit dan sistem-sistem komunikasi yang lain sehingga dapat saling berkomunikasi (Agus Mulyono, 2009)

Untuk dapat bertukar informasi, digunakan protocol standar yaitu Transmision Control Protocol dan Internet Protocol yang lebih dikenal sebagai TCP/IP. Sedangkan internet merupakan jaringan komputer didalam suatu organisasi yang menggunakan teknologi internet sehingga memungkinkan saling berbagi informasi, komunikasi, kerja sama, dan dukungan bagi proses bisnis.

### 2. Nama domain/URL

Nama domain atau URL adalah alamat untuk di dunia internet yang digunakan untuk mengidentifikasi sebuah website. Nama domain memudahkan user dalam mengingat alamat IP. Layanan yang bertugas menerjemahkan dalam IP ke sebuah nama domain adalah DNS (Domain Name Service).

### 3. *Web browser*

*Web browser* merupakan aplikasi di pihak client yang berfungsi menerjemahkan dan menampilkan informasi dari server secara grafis kepada client.

#### 4. *Web server*

Sebuah komputer (*server*) dan *software* yang menyimpan dan mendistribusikan data komputer lainnya melalui jaringan internet.

#### 5. *Web hosting*

*Web hosting* yaitu sebagai ruangan yang terdapat dalam *hardisk* tempat menyimpan berbagai data, file-file, gambar dan lain-lain yang akan ditampilkan ke website.

Aplikasi berbasis *web* memiliki beberapa kelebihan sebagai berikut :

- a. *Platform Independent* yaitu aplikasi dapat dijalankan di berbagai sistem operasi seperti *Windows*, *Linux* dan *Mac OS*.
- b. Untuk menjalankan aplikasi di banyak komputer tidak perlu instal aplikasi di setiap komputer, cukup copy script programnya ke server atau salah satu komputer. Untuk komputer lain yang ingin menjalankan program cukup membuka alamat host server dimana program disimpan melalui browser.
- c. Aplikasi dapat dijalankan dari jarak jauh dengan menggunakan koneksi internet.

## 2.4 Konsep Database

### 2.4.1 Definisi Database

Secara umum, database berarti koleksi data yang saling terkait. Secara praktis, basis data dapat dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat. (Abdul Kadir, 2008)

Sesungguhnya ada beberapa macam database, antara lain yaitu database hirarki, database jaringan, dan databases relasional. Databases realisonal merupakan database yang populer saat ini dan telah diterapkan pada beberapa platform, dari PC hingga minikomputer.

Sebuah databse relasional tersusun atas jumlah tabel. Sebagai contoh, database akademis mencakup tabel-tabel seperti dosen, mahasiswa, KRS, nilai, dan lain-lain.

#### 2.4.2 DBMS

Sistem manajemen basis data ( Bahasa Inggris : *database management system*, DBMS), adalah perangkat lunak yang digunakan untuk mengelola dan mengontrol peangaksesan database. Contoh DBMS adalah Oracle, SQL Server 2000/2003. MS Acces, MySQL dan sebagainya. (Abdul Kadir,2010)

DBMS merupakan perangkat lunak untuk mendefinisikan, meciptakan, mengelola dan mengendalikan pengaksesan basisdata. DBMS juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. Penyimpanan data dalam bentuk DBMS mempunyai banyak manfaat dan kelebihan dibandingkan dengan penyimpanan dalam bentuk file-file atau *spread sheet*, diantaranya :

1. Performa yang dapat dengan penyimpanan dalam bentuk DBMS cukup besar, sangat jauh berdeda dengan tampilan data yang disimpan dalam bentuk flat file. Disamping memiliki unjuk kerja yang lebih baik, juga akan didapatkan efisensi penggunaan media penyimpanan dan memori.
2. Integritas data lebih terjamin dengan penggunaan DBMS. Masalah redudansi sering terjadi dalam flat file. Redudansi adalah kejadian berulangnya data atau

kumpulan data yang sama dalam sebuah database yang mengakibatkan pemborosan media penyimpanan.

3. Independensi, perubahan struktur database dimungkinkan terjadi tanpa harus mengubah aplikasi yang mengaksesnya sehingga pembuatan antarmuka ke dalam data akan lebih mudah dengan penggunaan DBMS.
4. Sentralisasi, data yang terpusat akan mempermudah pengelolaan database. Kemudahan di dalam melakukan bagi pakai dengan DBMS dan juga kekonsistenan data yang diakses secara bersama-sama akan lebih terjamin dari pada data disimpan dalam bentuk file atau worksheet yang tersebar.
5. Keamanan, DBMS memiliki sistem keamanan yang lebih fleksibel daripada pengamanan pada file sistem operasi. Keamanan dalam DBMS akan memberikan keluwesan dalam pemberian hak akses kepada pengguna.

### 2.4.3 MySQL

*MySQL* (baca:mai-se-kyu-el) merupakan software yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *Open Source* menyatakan bahwa software ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara men-download (mengunduh) di Internet secara gratis. (Abdul Kadir,2008)

MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL :

1. *Portabilitas*

MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan lainnya.

2. Perangkat lunak sumber terbuka

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user*

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah 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. Ragam tipe data

MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned, 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 system perizinan yang mendetail serta terenkripsi.

## 8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu table 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 protocol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).

## 10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari puluhan bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

## 11. Antar Muka

MySQL memiliki antar muka (*interface*) 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 table

MySQL memiliki struktur table yang lebih fleksibel dengan menangani ALTER TABLE, dibandingkan data semacam PostgreSQL ataupun Oracle.



## **2.5 Peralatan Pendukung (Tools System)**

### **2.5.1 Pemodelan UML (Unified Modeling Language)**

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. (Chonoles,2003:bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintak dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi erro yang terjadi? Bagaimana keamanan terhadap sistem yang kita buat? Dan sebagainya dapat dijawab dengan UML. (Menggunakan UML, Prabowo Pudjo Widodo, 2011).“

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak
2. Sarana komunikasi anatar perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

#### **2.5.1.1 Bangun dasar Metodologi UML**

Bangunan dasar metodologi UML menggunakan tiga bangunan dasar yang digunakan untuk mendeskripsikan system/perangkat lunak yang akan dikembangkan yaitu :

## 1. Sesuatu (things)

Sesuatu (*things*) adalah hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen-elemen lainnya dari sebuah konsep dan atau fisik. Bentuk dari beberapa benda (*things*) adalah sebagai berikut :

- a. Kelas, adalah himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. Kelas mengimplementasikan satu atau lebih antarmuka (*interface*). Secara grafis, kelas digambarkan dengan empat-persegi-panjang yang memuat nama, atribut serta operasi yang dimilikinya.
- b. *Use Case*, adalah deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*. *Use case* digunakan untuk menstrukturkan perilaku pada suatu model. Secara grafis, *use case* digambarkan dengan elips tegas yang berisi namanya.

## 2. Relasi (Relationship)

*Relationship* adalah hubungan-hubungan yang terjadi antar-elemen dalam UML. Hubungan-hubungan ini penting sekali dalam UML. Ada empat macam *relationship* dalam UML, yaitu :

- a) *Dependency* (Kebergantungan). Hubungan dimana perubahan yang terjadi pada suatu elemen *independen* (mandiri) akan mempengaruhi elemen yang bergantung padanya (elemen yang tidak mandiri-independen).
- b) Asosiasi. Adalah apa yang menghubungkan antara objek satu dengan objek yang lainnya; bagaimana hubungan suatu objek dengan objek lainnya.

- c) Generalisasi. Hubungan di mana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk-*ancestor*). Arah dari atas ke bawah (dari objek induk ke objek anak) dinamakan *spesialisasi*, sedangkan arah berlawanan sebaliknya (bawah ke atas) dinamakan *generalisasi*.
- d) Realisasi. Operasi yang benar-benar dilakukan oleh suatu objek.

### 3. Diagram

UML menyediakan beberapa jenis diagram, di antaranya: *Class Diagram*, *Use Case Diagram*, *Sequence Diagram*, *Collaboration Diagram*, *Statechart Diagram*, *Activity Diagram*, *Component Diagram* dan *Deployment Diagram*. Diagram yang akan dibahas di sini hanya empat diagram, yaitu :



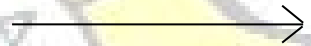
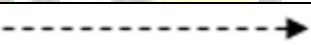
#### 1. Use Case Diagram

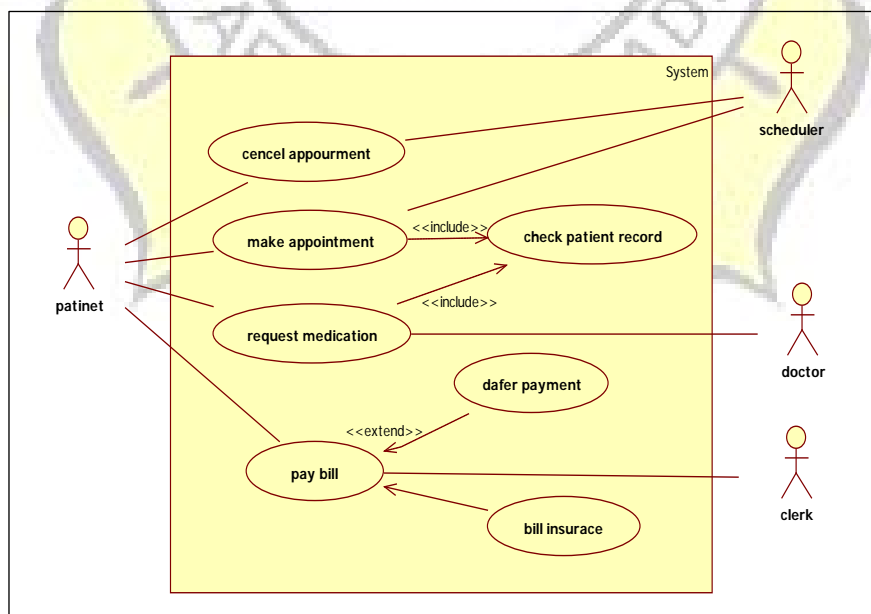
*Use case* diagram menggambarkan fungsionalitas yang di harapkan dari sebuah sistem. sebuah *usecase* mempresentasikan sebuah *interaksi* antara aktor dengan sistem. Sebuah *use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, membuat daftar belanja, dan sebagainya. Sorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa bahwa *use case* yang di *include* akan di panggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di *include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar yang common. Sebuah *use case* juga dapat meng-*extend* use case

lain dengan dengan *behavior*-nya sendiri. Sementara hubungan *generalisasi* antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Berikut adalah elemen dalam use case:

Tabel 2.1 Simbol *Class Diagram* (Munawar,2005)

Penjelasan	Notasi UML
Aktor : mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.	 pelanggan
Use case : abstraksi dari interaksi Antara sistem dan actor	
Association : adalah abstraksi dari penghubung antara aktor dan use case	
Generalisasi : menunjukkan spesialisasi aktor untuk dapat berpartisipasi dalam use case	



Gambar 2.1 *Use Case Diagram* (Munawar,2005)

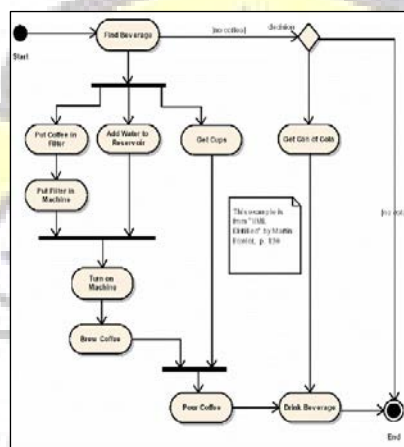
## 2. Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagai mana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

*Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar mentransisi di trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggunakan behavior internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh suatu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standart UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behavior pada kondisi tertentu. Untuk mengilustrasikan proses-proses parallel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau *vertical*. Activity diagram dapat di bagi menjadi beberapa object *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Simbol-simbol yang terdapat dalam *activity diagram*, sebagai berikut:

Table 2.2 Simbol *activity diagram* (Munawar,2005)

Keterangan	Simbol
Titik awal atau permulaan.	●
Titik akhir atau akhir dari beraktivitas.	⦿
Aktiviti,atau aktivitas yang di lakukan oleh actor	⬭
Decision, atau pilihan untuk mengambil keputusan.	◇
Arah tanda panah alur proses	→

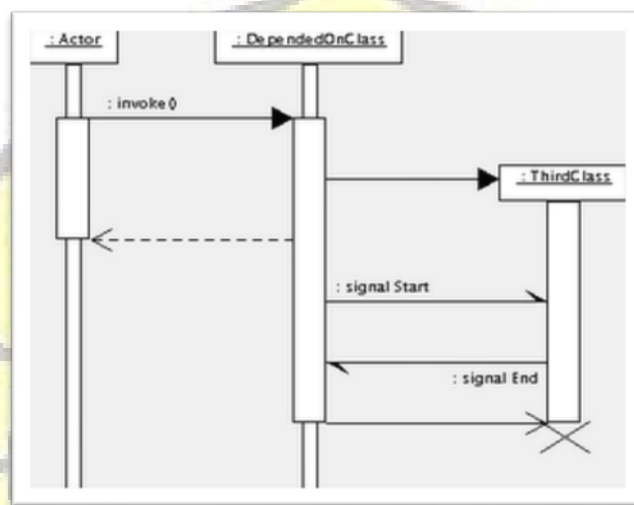


Gambar 2.2 *Activity Diagram* (Munawar,2005)

### 3. Sequence Diagram


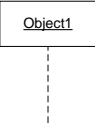
*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem termasuk (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri antar *vertikal*(waktu) dan dimensi *horizontal* (objek-objek yang terkait). Sequence diagram bisa di gunakan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang mentrigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang di hasilkan masing-



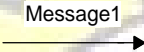
masing objek, termasuk aktor, memiliki *lifeline vertical*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lain. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/media dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawalidengannya diterimanya sebuah message. Untuk objek-objek yang memiliki sifat khusus, standart UML mendefinisikan icon khusus untuk objek *boundary*, *controller* dan *persistent entity*.



Gambar 2.3 *Sequence Diagram* (Munawar,2005)

Tabel 2.3 Komponen – Komponen Sequence Diagram

Nama	Bentuk	Keterangan
Actor		Seorang <i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> . Simbol <i>Actor</i> sama dengan dengan simbol pada <i>Actor Use Case Diagram</i> .
Object		<i>Object</i> merupakan <i>instance</i> dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek di dalamnya.

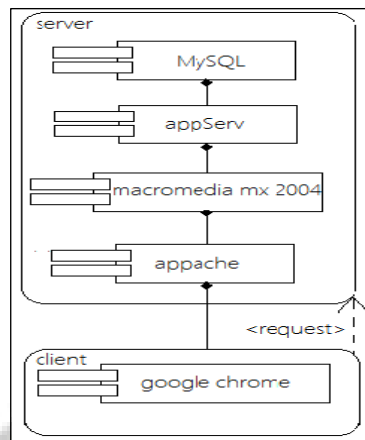
Nama	Bentuk	Keterangan
Activation		Dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan aksi.
Lifeline		Lifeline mengindikasikan keberadaan sebuah obyek dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.
Message		Message digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object.

Tabel 2.3 menunjukkan komponen atau bentuk-bentuk dasar dari *sequence diagram*. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

#### 4. Deployment Diagram

*Deployment Diagram* menjelaskan bahwa aplikasi website berbasis database data pegawai memiliki empat table di mana masing-masing table tersebut terhubung ke dalam database *MY SQL*.





Gambar 2.4 Deployment Diagram (Munawar,2005)

## 2.6 HTML

*HTML* adalah singkatan dari *Hyper-Text Markup Language*. *HTML* merupakan tulang punggung aplikasi web. Sekali pun sejumlah teknologi pendukung aplikasi web telah muncul, tetapi peran *HTML* tak tergantikan. Kehadiran *PHP* merupakan pelengkap yang memungkinkan aplikasi web tidak lagi bersifat statis, melainkan dinamis. Perubahan-perubahan pada sisi data tidak perlu membuat aplikasi diubah. Cara seperti itu terutama dilakukan pada sisi *server*. Artinya, kode seperti *PHP* berjalan di *server*. Selain cara seperti itu, penambahan kode pada *HTML* juga bisa dilakukan pada sisi klien. (Abdul Kadir, 2011).

## 2.7 CSS

*Cascading Style Sheet* (*CSS*) 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*,

*subbab, bodytext, footer, images, dan style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, 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.

Ada dua sifat CSS yaitu internal dan eksternal. Jika internal yang dipilih, maka skrip itu dimasukkan secara langsung ke halaman website yang akan didesain. Kalau halaman web yang lain akan didesain dengan model yang sama, maka skrip CSS itu harus dimasukkan lagi ke dalam halaman web yang lain itu.

Sifat yang kedua adalah eksternal di mana skrip CSS dipisahkan dan diletakkan dalam berkas khusus. Nanti, cukup gunakan semacam tautan menuju berkas CSS itu jika halaman web yang didesain akan dibuat seperti model yang ada di skrip tersebut. (Abdul Kadir, 2011).

## **2.8 JavaScript**

Javascript menurut (Sunyoto,2007:17) adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode Javascript dapat disisipkan dalam halaman web menggunakan tag SCRIPT

Beberapa hal tentang Javascript :

1. Javascript didesain untuk menambah interaktif suatu web
2. Javascript merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. Javascript berisi baris kode yang dijalankan di komputer (web browser).
5. Javascript biasanya disisipkan (*embedded*) dalam halaman HTML.
6. Javascript adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi).
7. Setiap orang dapat menggunakan Javascript tanpa membayar lisensi.

## 2.9 PHP

PHP adalah teknologi yang diperkenalkan tahun 1994 oleh Rasmus Lerdorf. Beberapa versi awal yang tidak dipublikasikan digunakan pada situs pribadinya untuk mencatat siapa saja yang mengakses daftar riwayat hidup onlinennya. Versi pertama digunakan oleh pihak lain pada awal tahun 1995 dan dikenal sebagai Personal Home Page Tools. Terkandung didalamnya sebuah parser engine (mesin pengurai) yang sangat disederhanakan, yang hanya mampu mengolah macro khusus dan beberapa utilitas yang sering digunakan pada pembuatan home page, seperti buku tamu, pencacah, dan hal semacamnya. Parser tersebut ditulis ulang pada pertengahan 1995 dan dinamakan PHP/FI Versiaon 2. FI(Form Interpreter) sendiri berasal dari kode lain yang ditulis juga oleh Rasmus, yang menterjemahkan HTML dari data. Ia menggabungkan script Personal Home Page Tools dengan Form Interpreter dan menambahkan dukungan terhadap *server* database yang menggunakan format mSQL sehingga lahirlah PHP/FI.

PHP/FI tumbuh dengan pesat, dan orang-orang mulai menyiapkan kode-kode programnya supaya bisa didukung oleh PHP. (Abdul Kadir, 2011).

## **2.10 Cuti**

Cuti adalah keadaan tidak masuk kerja yang diijinkan dalam jangka waktu tertentu. Cuti diberikan dalam rangka usaha menjamin kesegaran jasmani dan rohani, maka kepada Karyawan setelah bekerja selama jangka waktu tertentu perlu diberikan cuti.

### **2.10.1 Jenis-jenis Cuti**

Adapun jenis-jenis cuti berdasarkan Company Regulation milik PT. Summit Oto Finance dapat di golongkan menjadi beberapa jenis cuti, antara lain :

#### **1. Cuti Tahunan**

- a) Istirahat tahunan adalah hari-hari istirahat Karyawan setelah Karyawan menjalani masa kerja selama 12 (dua belas) bulan terus menerus dengan tetap menerima upah penuh.
- b) Lamanya istirahat tahunan ditetapkan 12 (dua belas) hari kerja Perusahaan.
- c) Perusahaan berhak mengatur hari-hari istirahat tahunan Karyawan dalam tahun takwim, untuk menjalin kelangsungan produktivitas kerja Perusahaan dengan memperhatikan kepentingan karyawan.
- d) Hari-hari resmi sebagaimana diatur dalam Keputusan Menteri Agama RI yang kebetulan istirahat tahunan tersebut timbul tidak dipergunakan oleh Karyawan bukan karena alasan/penundaan yang dibuat oleh Perusahaan.

## 2. Cuti Besar

- a) Karyawan yang telah bekerja selama 6 (enam) tahun secara terus menerus berhak atas cuti besar. Lamanya cuti besar dan hal-hal yang berkaitan dengan cuti besar akan diatur dalam Surat Keputusan (SK) tersendiri dengan berpedoman pada Peraturan Perundang-undangan yang berlaku tentang Ketenagakerjaan dan peraturan pelaksanaannya.
- b) Karyawan yang bermaksud menggunakan hak cuti besarnya supaya mengajukan permohonan kepada Atasan langsung dengan tembusan ke Departemen Personalia dan penggunaannya tidak boleh sekaligus.

## 3. Cuti Bersalin

Ketentuan istirahat melahirkan diatur berdasarkan Peraturan Perundang-undangan yang berlaku tentang ketenagakerjaan dan peraturan pelaksanaannya sebagai berikut :

- a) Istirahat melahirkan adalah membebaskan Karyawan wanita dari kewajiban untuk bekerja sehubungan dengan kelahiran anaknya dengan tetap menerima upah selama 3 bulan (1,5 bulan sebelum melahirkan dan 1,5 bulan setelah melahirkan).
- b) Untuk menjaga kesehatan dan keselamatan ibu dan anak, istirahat melahirkan diberikan satu setengah bulan menurut perhitungan saatnya ia akan melahirkan anak dan satu setengah bulan sesudah melahirkan anak atau gugur kandungan, hal tersebut harus dilampirkan surat keterangan dokter/bidan yang merawatnya.

#### 4. Cuti Khusus

a) Izin mengikuti wisuda, Untuk Karyawan yang akan mengikuti Wisuda Sarjana/Diploma III diberikan izin khusus selama 1 (satu) hari kerja dan berlaku hanya untuk 1 (satu) hari.

b) Izin menunaikan ibadah, Karyawan yang telah mencapai masa kerja 1 (satu) tahun berturut-turut dapat diberikan izin meninggalkan pekerjaan untuk melaksanakan kewajiban ibadah haji dalam musim haji (bukan umroh), dengan ketentuan :

- Pemberian izin hanya sekali dan lamanya harus sesuai dengan jadwal perjalanan yang ditentukan Pemerintah cq. Departemen Agama, dan izin khusus ini hanya berlaku bagi Karyawan Warga Negara Indonesia.
- Selama meninggalkan Pekerjaannya Karyawan berhak menerima upah sesuai dengan Perundang-undangan yang berlaku tentang Ketenagakerjaan dan peraturan pelaksanaannya.