BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Penelitian yang di laksanakan untuk pembuatan skripsi ini dilakukan di CV Gunung Agung , Jl. Ters Burujul No 37 Kabupaten Bandung. Telp (022) 5415620 FAX (022)5405437.

2.1.1 Sejarah Perusahaan

CV Gunung Agung adalah perusahaan swasta yang bergerak dalam kegiatan pelayanan jasa kontrakting (Mekanikal dan Elektrikal Sistem), penginstalasian serta perawatan jaringan listrik, penjualan dan *Service Showcase* (*local partner* PT Sanden Intercool Indonesia), sejak tahun 2004 di bawah pimpinan Bpk Hadian selaku pemimpin perusahaan.

Sejak Perusahaan ini berdiri, mereka menekankan untuk memberikan pelayanan yang memuaskan dan hasil kerja yang terbaik. CV Gunung Agung siap bekerja sama untuk melakukan pekerjaan sesuai dengan spesifikasi dan kondisi anda serta menantikan kesempatan untuk menunjukan kepada anda bahwa mereka adalah salah satu yang terbaik di bidangnya. Oleh karena itu, CV Gunung Agung memiliki tenaga – tenaga ahli dalam bidangnya.

CV Gunung Agung juga berusaha untuk melayani pelanggan secara menyeluruh dan berintegrasi. Hal ini merupakan salah satu usaha yang di lakukan perusahaan untuk memudahkan para pelanggan dalam memenuhi kebutuhan akan layanan jasa. Oleh karena itu, CV Gunung Agung memberikan pelayanan antara lain pleyanan jasa *kontrakting* , penginstalasian dan perawatan jaringan listrik,penjualan dan *Service Showcase*.

2.1.2 Visi dan Misi Perusahaan

Setiap instansi perusahaan tentunya memiliki Visi dan Misi agar perusahaan tersebut mempunyai tujuan yang baik. Adapun Visi dan Misi perusahaan CV Gunung Agung adalah :

Visi

“ Perusahaan penyedia jasa yang memberikan pelayanan terbaik dan terpercaya “

Misi

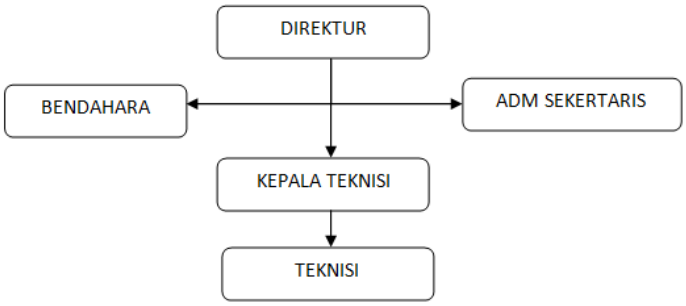
a. Bergerak dalam kegiatan pelayanan jasa *kontrakting* , penginstalasian serta perawatan jaringan listrik, penjualan dan *Service Showcase*.
b. Memperoleh kepercayaan dari konsumen.
c. Siap bekerja sama untuk melakukan pekerjaan sesuai dengan spesifikasi dan kondisi.

2.1.3 Badan Hukum

CV Gunung Agung adalah perusahaan swasta nasional, Berdasarkan Undang-Undang No. 3 Tahun 1982 tentang wajib Daftar Perusahaan dan Peraturan Daerah No. 12 dan 13 Tahun 2002 maka sejak tanggal 9 Oktober 2004 , CV Gunung Agung telah terdaftar pada Dinas Perindustrian dan Perdagangan Kabupaten Bandung sebagai perusahaan Persekutuan Komanditer (CV) dengan Nomor Tanda Daftar Perusahaan (TDP) 102435103458 dengan status pusat atas nama pemilik Bpk Hadian dan telah memiliki Surat Ijin Usaha Perdagangan (SIUP) untuk operasional kegiatan usaha adalah berdasarkan Nomor : 01303/10-12/PK/XI/04 serta CV Gunung Agung telah terdaftar di Direktorat Jendral Pajak dengan Nomor Pokok Wajib Pajak (NPWP) 09.337.419.7445.000.

2.1.4 Sturktur Organisasi Perusahaan

Struktur Organisasi CV Gunung Agung Bisa Di lihat pada gambar di 2.1:



Gambar 2.1 Struktur Organisasi CV Gunung Agung

2.1.5 Deskripsi Tugas

Secara garis besar setiap kepala bagian ataupun divisi bagian, memiliki tugas masing – masing dan harus bertanggung jawab dalam tugas jabatannya agar bisa bekerja secara maksimal , berikut ini adalah penjelasan masing – masing bagian yang ada di CV Gunung Agung.

1. **Direktur**
   a. Bertanggung Jawab atas seluruh kegiatan perusahaan baik kedalam maupun keluar.
   b. Membuat Kebijakan – kebijakan yang bersifat umum.
   c. Membuat dan menetapkan peraturan – peraturan perusahaan sesuai dengan undang – undang yang telah di tetapkan.
2. **ADM Sekertaris**
   a. Membuat Laporan – laporan dan segala kegiatan dalam kinerja CV Gunung Agung.
   b. Membuat Dokumentasi kegiataan kegiatan yang di laksanakan oleh CV Gunung Agung.
3. **Bendahara**
   a. Mengelola Keuangan dari CV Gunung Agung.
   b. Membuat Laporan – Laporan Keuangan baik pemasukan dan keluaran keungan dari CV Gunung Agung.
4. **Kepala Teknisi**
   a. Mengelola Pekerjaan yang di laksakan oleh CV Gunung Agung.

b. Membuat Laporan – Laporan Mengenai Pekerjaan yang di laksanakan oleh CV Gunung Agung.
c. Membagi tugas setiap teknisi lapangan.
d. Memantau kinerja setiap teknisi lapangan.

5. Teknisi Lapangan

a. Melaksakan Pekerjaan di lapangan yang di laksakan oleh CV Gunung Agung.
b. Membuat laporan – laporan pekerjaan apabila pekerjaan tersebut sudah selesai di laksanakan.

2.2 Landasan Teori

Landasan teori menjelaskan beberapa teori yang berkaitan dengan permasalahan yang dibahas sebagai dasar pemahaman dalam sebuah sistem.

2.2.1 Konsep Dasar Sistem

Secara umum sistem sendiri dapat diartikan sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu untuk mencapai suatu maksud .Menurut Al-Bahra Bin Ladjamudin (2005 : 1) dalam mendefinisikan sistem terdapat dua kelompok pendekatan sistem, yaitu sistem yang menekankan pada prosedur dan elemennya.

1) Pemahaman sistem dengan pendekatan prosedur, yaitu :
   Suatu urutan kegiatan yang saling berhubungan, berkumpul bersamasama untuk mencapai tujuan tertentu.
2) Pemahaman sistem dengan pendekatan komponen/elemen, yaitu :
   Kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu. Suatu sistem dapat terdiri dari beberapa subsistem.

Dari pemahaman diatas penulis dapat menyimpulkan bahwa sistem adalah kumpulan dari komponen atau elemen-elemen atau sub-subsistem yang saling berkaitan dan bekerja sama melakukan suatu urutan kegiatan yang saling berhubungan untuk mencapai suatu tujuan tertentu.

2.2.2 Konsep Dasar Informasi

Diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen didalam mengambil keputusan. Informasi mempunyai manfaat dan peranan yang sangat dominan dalam suatu organisasi/perusahaan. Informasi adalah hasil pengelolaan data yang memberikan arti dan manfaat. Informasi tidak dapat dipisahkan dari pengertian data.sumber dari informasi adalah data. Menurut **Jogiyanto(2001:8)** informasi dapat di definisikan sebagai berikut: "Data yang di olah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya" .

2.2.3 Sistem Informasi

Sistem informasi Menurut *Laudon* (2005:9) adalah satuan komponen yang saling berahubungan yang mengumpulkan, memproses, menyimpan dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi.

Menurut *Kennet hc. Laundon*(2007:15) Sistem informasi adalah berisi informasi tentang orang-orang, tempat, dan hal-lah penting di dalam organisasi atau di lingkungan sekelilingnya.

Berdasarkan pengertian diatas Sistem Informasi yang dikemukakan oleh para ahli dapat disimpulkan bahwa pengertian sistem informasi adalah satuan komponen yang saling berkaitan antara satu dan lainnya melalui proses, dimana proses tersebut menghasilkan data yang memiliki nilai.

2.2.4 Komponen Sistem Informasi

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), yaitu :

1. Blok Masukan (*Input Block*)

   Blok masukan ini mewakili data yang masuk kedalam sistem informasi.*Input* disini termasuk Metode-metode dan media untuk menangkap data yang akan dimasukan , yang dapat berupa dokumen - dokumen dasar

2. Blok Model (*Model Block*)

   Kombinasi prosedur, logika, dan model matemetik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diingiinkan

3. Blok Keluaran (*Output Block*)

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

   Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

   Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control block*)

   Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

2.2.5 Konsep Dasar Basis Data

Data merupakan respresntasi fakta dunia nyata yang mewakili suatu obyek (seperti : manusia: dosen, mhs, pelanggan ; barang = buku , meja) yang direkam baik dalam bentuk angka, huruf, teks, gambar atau suara dan memiliki arti secara jelas.

Informasi adalah hasil analisis dan sintesis terhadap data. Dengan kata lain informasi dapat dikatakan sebagai data yang diorganisasikan ke dalam bentuk

yang sesuai dengan kebutuhan seseorang didalam suatu organisasi atau perusahaan.

Basis Data *(database)* adalah metode untuk mengelola sistem berkas (*file*) terpadu yang menampung sekumpulan informasi dan diorganisir sedemikian rupa, sehingga informasi tersebut dapat dimanfaatkan kembali dengan cepat, serta infromasi yang saling berhubungan dapat disimpan secara bersamaan tanpa adanya pengulangan.

2.2.6 Pengertian Basis Data

Basis data dapat di artikan sebagai " suatu markas atau gudang, tempat bersarang atau berkumpul ", Menurut Yakub (2008 : 1). Menurut Andri Kristianto (2008 : 79) mendefinisikan basis data sebagai " Kumpulan data yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi".

Menurut Al-Bahra Bin Ladjamudin (2005 : 129) terdapat beberapa definisi basis data dari beberapa orang ahli basis data adalah sebagai berikut :

1) *Database* adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam magnetic disk, optical disk, magnetic drum atau media penyimpanan sekunder lainnya.
2) *Database* adalah sekumpulan program – program aplikasi umum yang bersifat "batch" yang mengeksekusi dan memproses data secara umum (seperti pencarian, peremajaan, penambahan, dan penghapusan terhadap data).
3) *Database* adalah koneksi terpadu dari data – data yang saling berkaitan dari suatu enterprise (perusahaan, instansi pemerintah atau swasta).
4) *Database* terdiri dari data yang akan digunakan atau diperuntukan terhadap banyak *user*, dimana masing – masing *user* (baik menggunakan teknik pemrosesan yang bersifat batch atau on-line) akan menggunakan data tersebut sesuai dengan tugas dan fungsinya, dan *user* lain dapat juga menggunakan data tersebut dalam waktu yang bersamaan.

Dalam basis data terdapat dua komponen bahasa basis data diantaranya adalah sebagai berikut :

1) *Data Definision Language* (DDL)

   Digunakan Untuk menspesifikan struktur / skema basis data yang menggambarkan desain basis data secara keseluruhan. Hasil kompilasi perintah DDL adalah kamus data yang berisi *file* data yang mendeskripsikan data sesungguhnya. Metode akses yang di gunakan oleh sistem basis data disebut dengan data *storage* dan *definition Language*

2) *Data Manipulation Language* (DML)

   Data Manipulation *Language* digunakan untuk memanipulasi data , adapun bentuk manipulasinya adalah sebagai : pencarian data yang sudah lama disimpan, penyimpanan data baru, penghapusan data, serta pengubahan data. *Data Manipulation Language* memiliki 2 tipe diantaranya adalah sebagai berikut :

   1. Prosedural

      Membutuhkan *user* untuk menspesifikasikan data apa saja yang dibutuhkannya dan bagaimana cara mendapatkannya. Contoh paket bahasanya procedural seperti : Base III dan Fox Base

   2. Non Prosedural

      Mebutuhkan *user* untuk menspesifikasikan data apa saja yang di butuhkannya tanpa tahu bagaimana cara mendapatkannya. Contoh Paket Bahasanya adalah seperti : SQL (*Structure Query Language*) dan QBE (*Query By Example*)

2.2.7 *Database Manajemen Sistem* (DBMS)

Manajemen sistem basis data atau *database* manajemen sistem dapat didefinisikan sebagai kumpulan program aplikasi yang digunakan untuk membuat dan mengelola basis data, Menurut Yakub (2008 : 14). Perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan basis data serta utilitas kumpulan data dalam jumlah, Menurut Andri Kristanto (2008 : 79).

DBMS dapat menjadi alternatif pengguna secara khusus untuk aplikasi, misalnya penyimpanan data dalam *file* dan menulis kode aplikasi yang spesifik untuk pengaturannya.

2.2.8 *Object Oriented Programing* (OOP)

OOP (*Object Oriented Programming*) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Pemograman Berorientasi *Object* menekankan konsep seperti berikut :

1. ***Objek***

   Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

2. ***Class***

   Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *'class of lion'* adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari singa.

3. ***Enkapsulasi***

   Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut

4. ***Inheritance***

   Mengatur *polimorfisme* dan *enskapsulasi* dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah

ada, objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut.

5. Polimorfisme

Melalui pengiriman pesan. tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut *polimorfisme* karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama.

2.2.9 Konfigurasi Jaringan

Konfigurasi jaringan terdiri dari pengertian jaringan komputer, arsitektur jaringan komputer, serta topologi jaringan komputer. Berikut adalah definisi definisi dari konfigurasi jaringan tersebut :

2.2.9.1 Pengertian Jaringan Komputer

Iwan Sofana (2008) menyatakan bahwa jaringan komputer (*computer network*), adalah satu himpunan interkoneksi sejumlah komputer *autonomous* dalam suatu himpunan interkoneksi sejumlah komputer *autonomous*.

Tujuan Jaringan komputer adalah berbagi sumber daya (data, printer, harddisk), berkomunikasi (email, chatting), dan untuk akses informasi (*Web* browsing).

2.2.9.2 Arsitektur Jaringan Komputer

Dalam sebuah jaringan komputer yang umum digunakan memiliki beberapa arsitektur jaringan , adapun arsitektur jaringannya adalah sebagai berikut:

1. **Jaringan Berbasis Server**

Jaringan berbasis server (central node) ini disebut juga client-server. Semua *workstation* (node) yang berhubungan dengan jaringan ini dikelola oleh pengontrol domain.Pada pengontrol domain ini semua account *workstation* atau *user* dikumpulkan dan disimpan pada *database*nya. Tipe inilah yang digunakan oleh jaringan komputer dilingkungan Pondok Pesantren Sukamiskin. Sebagai contoh sistem operasi yang digunakan pada tipe jaringan ini adalah Windows 2000 server, Windows NT Server, Linux Server, atau Novell Netware. Sedangkan untuk *workstation*-nya bisa menggunakan windows 2000 Profesional, windows NT *Workstation*, Windows 98, Linux *Workstation*.

2. **Jaringan Peer to Peer**

Pada jenis ini setiap komputer yang terhubung dapat membuat Account *user* serta berbagi sumber (*sharing*), sehingga setiap komputer dapat bersifat sebagai server dan dapat juga bersifat sebagai *workstation*. Jadi pada peer to peer tidak ada perbedaan sistem operasi antara komputer yang difungsikan sebagai server dengan komputer yang difungsikan sebagai *workstation*. Jaringan dengan tipe ini diorganisasikan kedalam *Workgroup*. Sebagai contoh sistem operasi produk Microsoft pada tipe ini adalah Windows 3.11, Windows 95/98, Windows NT *workstation* dan Windows Profesional.

3. **Jaringan Hybrid**

Jaringan Hybrid merupakan gabungan dari sifat pada jaringan peer to peer dan server. *Workgroup* yang terdiri dari beberapa komputer yang saling terhubung dapat mengelola sumber daya tanpa membutuhkan otorisasi dari administrator jaringan atau server. Pada jenis jaringan ini, terdapat pula sifat dari jaringan server sedemikian sehingga tingkat keamanan dapat lebih terjaga dan

adanya server yang mempunyai suatu fungsi layanan tertentu, seperti sebagai *file* server, print server, *database* server, mail server, dan lainnya.

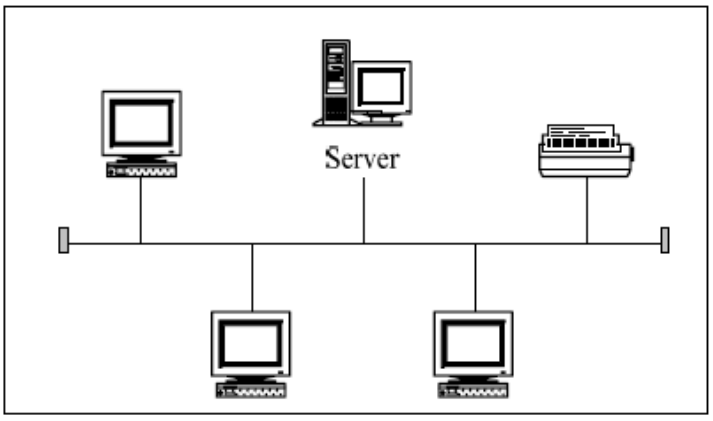
2.2.9.3 Topologi Jaringan

Topologi jaringan adalah bagian yang menjelaskan hubungan antar komputer yang di bangun berdasarkan kegunaan, keterbatasan *resource* dan keterbatasan biaya, berarti topologi-topologi jaringan yang ada bisa disesuaikan dengan keadaan di lapangan. Topologi jaringan ada beberapa sebagai berikut:

Menurut Budhi Irawan (2005:25). Topologi secara fisik dari suatu jaringan lokal adalah merujuk kepada konfigurasi kabel, komputer, dan perangkat lainnya. Berikut adalah macam-macam topologi fisik yang digunakan di dalam jaringan lokal dintaranya :

1. *Linear Bus* (Garis Lurus)

Topologi *Linear* Bus (Garis Lurus) terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah *terminator*. Semua *nodes* pada jaringan (*file server, workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama *(backbone)* jaringan-jaringan *Ethernet* dan *local talk* menggunakan topologi *linear* ini.

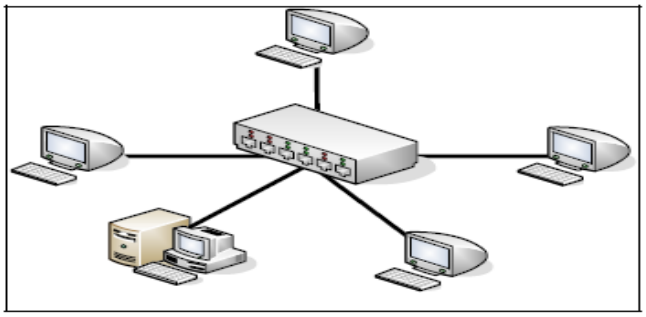


Gambar 2.2 Topologi *Linear Bus*

[Sumber: Budhi Irawan, Jaringan Komputer, 2005]

2. *Star* (Bintang)

Topologi model ini dirancang, yang mana setiap *nodes (file server, workstation,* dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator.* Data yang dikirim ke jaringan lokal dan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagai *repeater* (penguat aliran data). Konfigurasi pada jaringan model ini menggunakan kabel *Twisted Pair*, dan dapat digunakan pula kabel *coaxial* atau kabel *Fibre Optic*

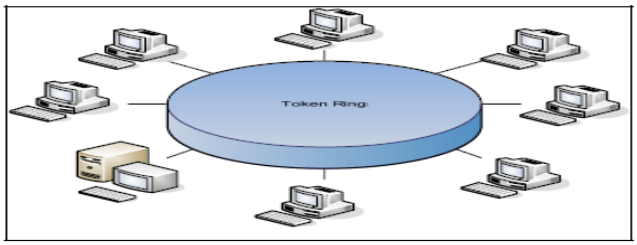


Gambar 2.3 Topologi *Star*

[Sumber: Budhi Irawan, Jaringan Komputer, 2005]

3. *Ring* (Cincin)

Topologi Ring (Cincin) menggunakan teknik konfigurasi yang sama dengan topologi star tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring.*

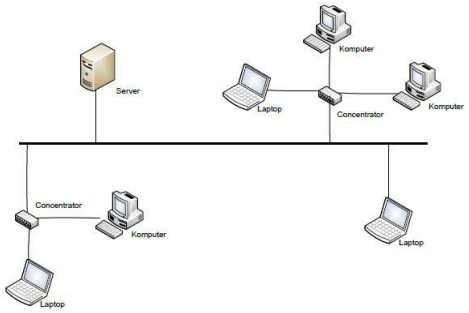


Gambar 2.4 Topologi *Ring*

[Sumber: Budhi Irawan, Jaringan Komputer, 2005]

4. *Tree* (Pohon)

Topologi model ini merupakan perpaduan antara topologi Linear Bus dan Star, yang mana terdiri dari kelompok-kelompok dari *workstation* dengan konfigurasi star yang terkoneksi ke kabel utama yang menggunakan topologi Linear Bus. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2.5 Topologi *Tree*

[Sumber: Budhi Irawan, Jaringan Komputer, 2005]

2.2.10 Web Service

Web Service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web Service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu *Web* site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*Service*) yang disediakan oleh suatu sistem yang menyediakan *Web Service*. *Web Service* menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda *platform*, sistem operasi, maupun bahasa compiler.

Web Service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam *Web Service* dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detil pemrograman yang

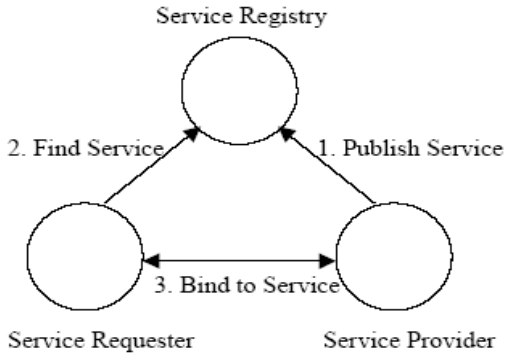
terdapat di dalamnya. Ada beberapa alasan mengapa digunakannya *Web Service* adalah sebagai berikut:

1. *Web Service* dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. *Web Service* memiliki kemudahan dalam proses *deployment-nya* , karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web Service* cukup di-*upload* ke *Web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. *Web Service* berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian *Web Service* tidak memerlukan konfigurasi khusus di sisi firewall.

2.2.10.1 Arsitektur *Web Service*

Dalam sebuah arsitektur Komputer, terdapat tiga buah entitas yang saling berhubungan, ke tiga entitas tersebut adalah sebagai berikut :

1. *Service Requester* (peminta layanan)
2. *Service Provider* (penyedia layanan)
3. *Service Registry* (daftar layanan)



Gambar 2.6 Arsitektur *Web Service*

Fungsi dari setiap entitas yang berada pada arsitektur *Web Service* adalah sebagai berikut :

- ***Service Provider* :** Berfungsi untuk menyediakan layanan/*Service* dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.
- ***Service Registry* :** Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/*Service* yang telah di-register.
- ***Service Requestor*** : Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.

Setiap entitas dalam sebuah arsitektur komputer memiliki tiga operasi yang terlibat di dalamnya, diantaranya adalah sebagai berikut :

1. ***Publish/Unpublish*:** Menerbitkan/menghapus layanan ke dalam atau dari registry.
2. ***Find:*** *Service requestor* mencari dan menemukan layanan yang dibutuhkan.
3. ***Bind:*** *Service requestor* setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke *Service* provider untuk melakukan interaksi dan mengakses layanan/*Service* yang disediakan oleh *Service* provider.

2.2.10.2 Jenis – Jenis *Web Service*

Jenis – jenis *Web Service* di bagi menjadi dua bagian , diantaranya adalah REST dan SOAP yang akan di jelaskan di bawah ini :

1. ***Representational State Transfer* (REST)**

REST adalah salah satu jenis *Web Service* yang menerapkan konsep perpindahan antar *State*. *State* disini dapat digambarkan seperti jika browser meminta suatu halaman *Web*, maka server akan mengirimkan *State* halaman *Web* yang sekarang ke browser. Ber*navigasi* melalui link-link yang disediakan sama halnya dengan mengganti *State* dari halaman *Web*. Begitu pula REST bekerja, dengan ber*navigasi* melalui link-link HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *State* satu sama lain. Perintah HTTP yang bisa

digunakan adalah fungsi GET, POST, PUT atau DELETE. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi client. Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *Web Service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi pada *resource*nya. sedangkan aplikasi *reporting Service* berbasis android membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource* data kerusakan. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *Web Service* yang menggunakan REST adalah: Flickr API(*Application Program Interface),* YouTube API, Amazon API.

2. *Simple Objek Akses Protocol* (SOAP)

SOAP merupakan protokol standar yang ringan dan ditujukan untuk pertukaran informasi dalam desentralisasi sistem. Protokol ini melakukan pemanggilan metode dalam bahasa XML (*Extensible Markup Language*). Dengan demikian, SOAP adalah suatu mekanisme yang sederhana untuk melakukan pertukaran struktur dan tipe informasi dalam lingkungan yang tersebar dan terdistribusi menggunakan XML. Secara garis besar juga protocol SOAP dapat di bagi menjadi beberapa bagian diantaranya adalah sebagai berikut :

I. SOAP *Envelope*

Untuk mendefinisikan apa yang terdapat dalam pesan (message) yang dikirim, dan kepada siapa pesan tersebut akan digunakan. Elemen ini merupakan *root* dari pesan dan bersifat wajib.

II. SOAP *Header*

Elemen ini berisi informasi *header* dan bersifat opsional. Informasi yang ada pada *header* tidak ditampilkan di aplikasi.

III. SOAP *Body*

Elemen ini berisi panggilan dan merespon informasi. Elemen ini wajib.

IV. SOAP *Encoding Rules*

Mendefinisikan mekanisme serialisasi yang dapat digunakan untuk pertukaran tipe data yang didefinisikan dari tiap aplikasi.

V. SOAP RPC *Representation*

Menyediakan suatu standar yang dapat digunakan untuk mewakili remote procedure call (RPC) dan data hasil kembaliannya.

VI. SOAP *Binding*

Mendefinisikan standar yang digunakan utnuk pertukaran SOAP Envelope antara peers dengan protokol yang umum untuk transportasi.

2.2.11 Android

Android adalah sistem operasi berbasis *Linux* yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk *ponsel*, *smartphone* dan juga *PC tablet*. Secara umum Android adalah *platform* yang terbuka *(Open Source)* bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

Semenjak kehadirannya pada 9 Maret 2009, Android telah hadir dengan versi 1.1, yaitu sistem operasi yang sudah dilengkapi dengan pembaruan estetis pada apalikasinya, seperti jam alrem, *voice search*, pengiriman pesan dengan

Gmail, dan pemberitahuan *email*. Hingga tahun 2012, Android telah berkembang dengan pesat.versi terakhir yang diproduksi disebut sebagai Android versi 4.4 atau Android Kit Kat. Berikut ini merupakan Perkembangan Android dari tahun 2009 - 2013

1) Android versi 1.1

   Google merilis Android versi 1.1.pada 9 Maret 2009 Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2) Android versi 1.5 (*Cupcake*)

   Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*) pada pertengahan Mei 2009. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke headset *Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3) Android versi 1.6 (*Donut*)

   Donut (versi 1.6) dirilis pada September 2009 dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, 16 *Gestures*, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

4) Android versi 2.0/2.1 (*Eclair*)

   Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah

pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan *user interface* dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

5) Android versi 2.2 (*Froyo: Frozen Yoghurt*)

   Android Froyo yaitu android versi 2.2 diluncurkan pada tanggal 20 Mei 2010, dengan perubahan terhadap versi android sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 *JavaScript engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam *SD Card*, kemampuan *WiFi Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi Android Market.

6) Android versi 2.3 (*Gingerbread*)

   Android *Gingerbread* diluncurkan pada tanggal 6 Desember 2010. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan *Web*M, efek audio baru (*reverb, equalization, headphone virtualization dan bass boost*), dukungan kemampuan *Near Field Communication (NFC)*, dan dukungan jumlah kamera yang lebih dari satu.

7) Android versi 3.0/3.1 (*Honeycomb*)

   Android *HoneyComb* dirancang khusus untuk *tablet*, disini anda dapat melihat *tablet* yang memakai android untuk os beserta harga *tablet*. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras *(hardware)* untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah *Motorola Xoom*. Perangkat *tablet* dengan *platform* Android 3.0 akan segera hadir di

Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

8) Android versi 4.0 (*ICS :Ice Cream Sandwich*)

   Android *Ice Cream Sandwich* diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline*, dan berbagi informasi dengan menggunakan *NFC.*

9) Android versi 4.1 (*Jelly Bean*)

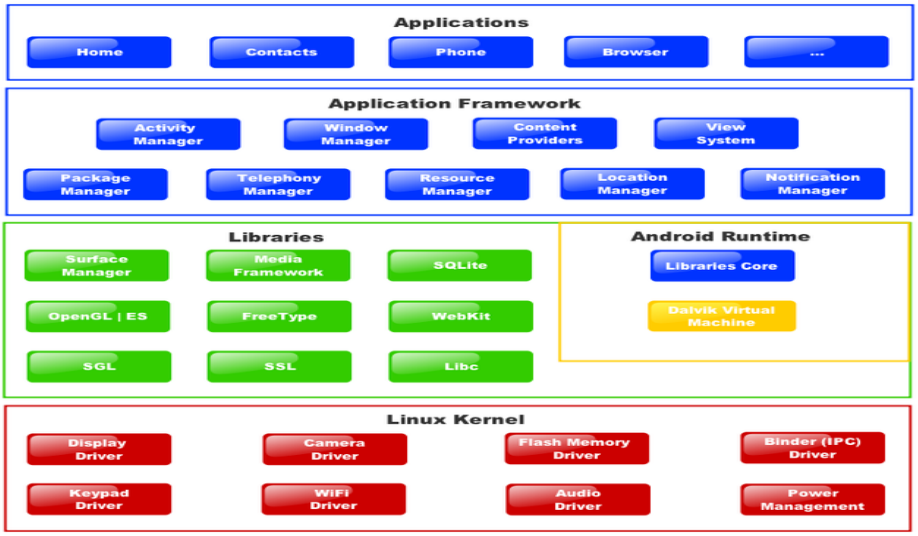
   Android *Jelly Bean* yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan *input keyboard,* desain baru fitur pencarian, UI yang baru dan pencarian melalui *Voice Search* yang lebih cepat.

10) Android versi 4.4 (Kit Kat)

   Android Kit Kat yang di luncurkan pada pertengahan agustus 2013 lalu membawa fitur unggulan seperti *Firmware* yang bisa digunakan untuk *smartphone* lama ,Pergantian *Fragmentation* atau ukuran tampilan layar , *Miracast updates Gallery* baru, Visualisasi *API* baru *Boot* animasi baru, pergantian warna logo Android berwarna biru, *Notifikasi widget* baru

2.2.12 Arsitektur Android

Android terdiri dari beberapa stack *software* yang terdiri dari: *Applications, Application Framework, Libraries, Android Runtime dan Kernel Linux*. Pada gambar 2.7 merupakan gambar arsitekturnya :



Gambar 2.7 Arsitektur Android

Secara garis besar arsitektur komputer dapat di jelaskan sebagai berikut :

2.2.12.1 Application Widgets

Application dan Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan *instalasi* dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk *klien* email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *java.*

2.2.12.2 Application Framework

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Arsitektur aplikasi dirancang supaya pengembang dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (reuse). Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

1. *View* yang kaya dan *extensible* yang dapat digunakan untuk membangun aplikasi, termasuk *list, grid*, kotak teks, tombol, dan bahkan sebuah *embeddable Web.*

2. *Content provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak *help*) atau membagi data tersebut.
3. *Resource Manager* yang menyediakan akses ke kode non sumber lokal seperti *string,* gambar, dan tata letak *file.*
4. *Notification Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dokostumisasi didalam status bar.
5. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan *navigasi* umum *backstack.*

2.2.12.3 Libraries

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel. Layer ini meliputi berbagai Library C/C++ inti seperti Libe dan SSL, serta:

1. *Libraries* media untuk pemutaran media audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries* grafik mencakup *SGL* dan *OpenGL* untuk grafis *2D* dan *3D.*
4. *Libraries SQLite* untuk dukungan *database*.
5. *Libraries SSL* dan *Webkit* terintegrasi dengan *Web browser* dan keamanan.
6. *Libraries LiveWebcore* mencakup *modern Web browser* dengan *engine embedded Web view*.
7. *Libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API"s.*

2.2.12.4 Android Run Time

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam *Android Run Time* dibagi menjadi dua bagian yaitu:

a. *Core Libraries*: Aplikasi Android dibangun dalama bahasa *java,* sementara *Dalvik* sebagai *virtual* mesinnya buka *Virtual Machine Java,*

sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa *java/c* yang ditangani oleh *Core Libraries*.

b. *Dalvik Virtual Machine*: *Virtual* mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara *efisien*, dimana merupakan pengembangan yang mapu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

2.2.12.5 Linux Kernel

Linux Kernel adalah layer dimana inti dari operating sistem Android itu berada. Berisi *file-file* sistem yang mengatur sistem *processing, memory, resource, drivers,* dan sistem-sistem operasi Android lainnya. *Linux Kernel* yang digunakan Android adalah *linux kernel* versi 2.6.

2.2.13 UML (Unified Modelling Language)

UML adalah bahasa untuk mendokumentasikan, memodelkan, dan membangun sistem perangkat lunak dalam bentuk diagram visual. *UML* dicetuskan oleh Three Amigos (tiga sekawan), Grady Booch, James Rumbaugh, dan Ivar Jacobson.

Adapun Versi-versi dari *UML*:

a. *UML* 1.0 diresmikan pada Januari 1997

b. *UML* 1.1 diresmikan pada Juli 1997

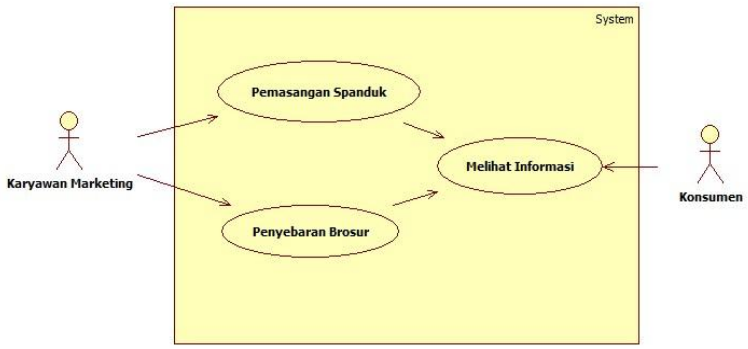
c. *UML* 1.2 diresmikan pada Juni 1998

d. *UML* 1.3 diresmikan pada tahun 1999

Berikut ini adalah jenis – jenis diagram yang biasa di gunakan pada UML :

a. Use Case diagram

Mendeskripsikan apa yang sistem lakukan tanpa mendeskripsikan bagaimana sistem menyelesaikannya. Dibuat berdasarkan interaksi dan relasi dari individual *Use Case* yang berisi *aktor, event,* dan *Use Case.* Berikut ini contoh *Use Case diagram* :



Gambar 2.8 Contoh *Use Case diagram*

b. Use Case Skenario

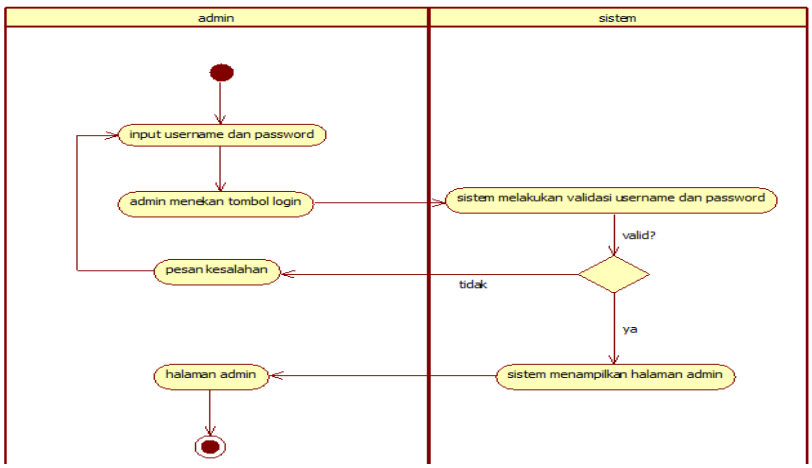
Merupakan hasil penjelasan dari setiap *Use Case*. Terbagi menjadi tiga bagian, yaitu: identifikasi dan inisiasi, *step perfomed*, kondisi akhir. Berikut ini contoh *Use Case Skenario* pada gambar 2.9 :

Identifikasi			
Nomor *Use Case*	7		
Nama *Use Case*	Mengisi *Testimonials* di Halaman *Testimonials*		
Deskripsi :	Pengunjung Mengisi *testimonials*		
Jenis :	Primer		
Aktor :	Pengunjung *(User)*		
Skenario Utama			
Kondisi Awal			
No	Aksi Aktor	No	Respon Sistem
1	Mengunjungi Website Sangkanpark.com		
		2	Tampil Halaman Utama
3	Pilih Menu *Testimonials*		
		4	Tampil Halaman *Testimonials*
5	Mengisi *form* (Isi pesan, Input Nama, Pilih Kategori, Upload Foto, input pekerjaan)		
		6	Klik Tombol Send

Gambar 2.9 Contoh *Use Case Skenario*

c. Actifity Diagram

Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses yang dipakai untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur.

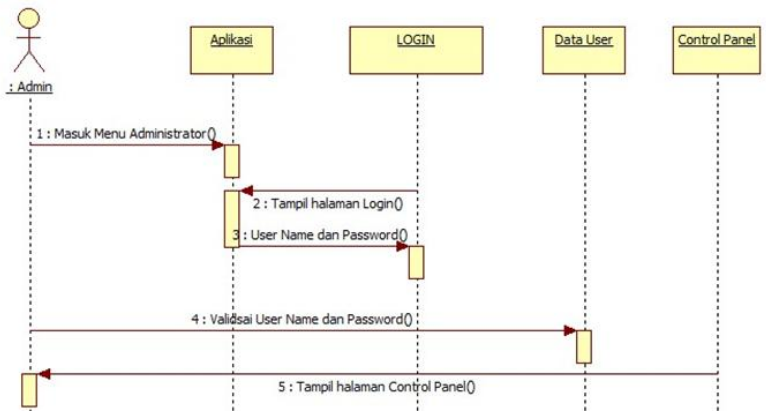


Gambar 2.10 Contoh *Actifity diagram*

d. Sequence diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

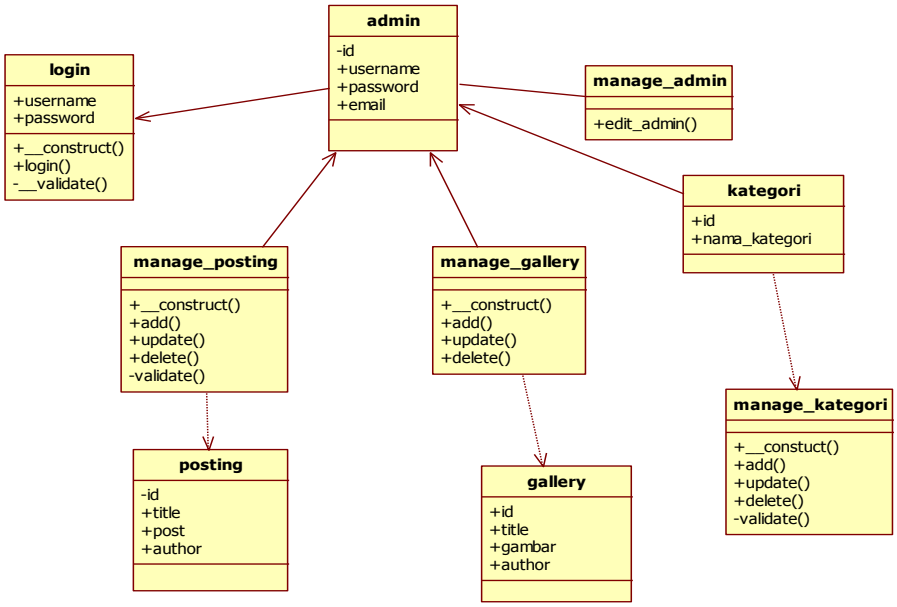
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan. Berikut adalah contoh *Sequence diagram pada gambar 2.11* :



Gambar 2.11 Contoh *Sequence diagram*

e. *Class diagram*

Class diagram menggambarkan keadan suatu sistem dengan menjelaskan keterhubungan antara suatu *class* dengan *class* yang lain yang terdapat pada sistem tersebut. *Class diagram* bersifat statis, di dalam *class diagram* digambarkan *relasi* dari masing - masing *class* tetapi tidak menggambarkan apa yang terjadi ketika *class* tersebut *berelasi*. Berikut pada gambar 2.12 contoh *class diagram*:



Gambar 2.12 Contoh *Class diagram*

2.3 *JAVA*

*Jav*a adalah bahasa pemograman berorientasi objek dengan unsur bahasa *C++* yang cocok dengan lingkungan internet. Penguasaan bahasa pemograman merupakan entry point bagi mereka yang ingin menekuni bidang IT. Paradigma *Object Oriented Programing, multi platform,* dan sistem distribusi telah menjadi trend pengembangan *software* saat ini. Hal tersebut menurut para profesional IT selalu mengupdate skill yang dimilikinya.

Netbeans merupakan sebuah aplikasi *Integrated Development Environment (IDE)* yang berbasiskan *Java* dari *Sun Micro*sistems yang berjalan di atas swing. *Swing* merupakan sebuah teknologi *Java* untuk pengembangan aplikasi desktop

yang dapat berjalan pada berbagai macam *platform* seperti *windows, linux, Mac OS X dan Solaris*. Sebuah *IDE* merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface (GU*I), suatu kode *editor atau text, suatu compiler dan suatu debugger.*

2.4 *PHP* (*Hypertext Preprocessor*)

PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *Web* yang dinamis. Ketika dipanggil dari *Web browser*, program yang ditulis dengan *PHP* akan di-parsing di dalam *Web server* oleh *interpreter PHP* dan diterjemahkan ke dalam dokumen *HTML*, yang selanjutnya akan ditampilkan kembali ke *Web browser*. Karena pemrosesan program *PHP* dilakukan di lingkungan *Web server*, *PHP* dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, PHP tidak akan terlihat pada saat *user* memilih perintah "*View Source*" pada *Web browser* yang digunakan.

PHP diciptakan oleh *Rasmus Lerdorf*, seorang pemrogram *C* yang andal. Semula *PHP* hanya digunakan untuk mencatat jumlah pengunjung pada homepagenya.

2.5 *CSS* (*Cascade Style Sheet*)

sebuah *style sheet Language* yang digunakan untuk merubah atau meningkatkan tampilan design pada halaman *HTML* dengan cara yang terorganisir dan efisien. *CSS* juga dapat digunakan untuk mengontrol semua aspek visual dari sebuah halaman *Web*. Ini termasuk warna, huruf, *layout* dan masih banyak lagi yang lainnya. *CSS* sangat penting untuk *HTML* karena *CSS* merupakan penghubung antara konten dan *design* dalam sebuah halaman *Web*. *CSS* biasanya dipisahkan kedalam dua tipe :

a. Internal *CSS* : *CSS* yang terdapat dalam halaman *HTML*.
b. Eksternal *CSS* : *CSS* yang disisipkan/ditambahkan pada halaman *HTML*.

Pada umumnya, menggunakan Eksternal *CSS* lebih di anjurkan karena memberikan konsistensi untuk *site-wide* dan mengurangi kompleksitas atau pengulangan dari pada menggunakan Internal *CSS*. Adapun keuntungan apabila menggunakan *CSS* adalah :

1. Menambahkan tampilan baru pada halama *HTML*.
2. Meperbaharui seluruh tampilan *Website*/*blog* hanya dengan sedikit merubah/menambahkan kode *CSS*.
3. Mebuat setiap halaman *Web*/*blog* menjadi lebih interaktif dan menarik.

2.6 JSON (JavaScript Object Notation)

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman *JavaScript*, Standar *ECMA*-262 Edisi ke-3 - Desember 1999. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga *C* termasuk *C, C++, C#, Java, JavaScript, Perl, Python*. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. *JSON* terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*Object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data *universal*. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian

karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data.

2.7 *Apache Web server*

Web server Apache merupakan program aplikasi yang berjalan diserver berfungsi untuk menjalan aplikasi *Web* sehingga bisa akses oleh *client* baik melalui jaringan secara *intranet* maupun *internet* (R. Kresno Aji dan Agus Hartanto, 2003).

Apache merupakan turunan dari *Web server* yang dikeluarkan oleh *NCSA*, yaitu *NCSA HTTPd* sekitar tahun 1995-an. Pada dasarnya *Apache* adalah "*APatCHy*"

(*patch*) dan pengganti dari *NCSA HTTPd*. *Apache Web server* merupakan tulang punggung dari *World Wide Web* (*Open Source Research Group* Universitas Ahmad Dahlan, 2002).

Web server menunggu permintaan dari *client* yang menggunakan *browser*, seperti *Nestcape Navigator, Internet Explorer, Modzilla*, *Lynks*. *Web server* dalam berkomunikasi dengan client menggunakan *protocol HTTP* (*Hyper Text Transfer Protocol*).

Apache berada di bawah *GNU*, General Public Licence yang bersifat free sehingga Apache dapat di download gratis pada alamat *http://www.apache.org*, saat ini *Apache* banyak digunakan sebagai *Web server* untuk portal-portal besar dengan pertimbangan sebagai berikut :

1. *Apache* termasuk dalam kategori *free software* (*software* gratis)
2. Instalasi *Apache* sangat mudah
3. Mampu beroperasi pada banyak *platform* sistem operasi seperti *AUX* 3.1, *Free BSD, Linux, Solaris, Windows*.
4. Mudah dalam pengkonfigurasian karena *Apache* hanya mempunyai empat *file* konfigurasi.

Apache mudah dalam penambahan *peripheral* lainnya ke dalam *platform Web server*, misalkan menambah modul.

2.8 *GPS* (*Global Positioning* Sistem)

Global Positioning Sistem (*GPS*) adalah suatu sistem *navigasi* yang memanfaatkan *satelit*. Penerima *GPS* memperoleh sinyal dari beberapa *satelit* yang mengorbit bumi. *Satelit* yang mengitari bumi pada orbit pendek ini terdiri dari 24 susunan *satelit*, dengan 21 *satelit* aktif dan 3 buah *satelit* sebagai cadangan. Dengan susunan orbit tertentu, maka satelit *GPS* bisa diterima di seluruh permukaan bumi dengan penampakan antara 4 sampai 8 buah *satelit*. *GPS* dapat memberikan informasi posisi dan waktu dengan ketelitian sangat tinggi.

Nama lengkap *GPS* adalah *NAVSTAR* GPS (*Navigational satellite Timing and Ranging Global Positioning* Sistem), namun lebih sering dikenal sebagai *GPS*. *GPS* mulai diaktifkan untuk umum pada 17 Juli 1995. Sedangkan, *Assisted-Global Positioning* Sistem (*A-GPS*) merupakan penyempurnaan dari *GPS* sebagai satelit penentu posisi di belahan bumi.*Satelit GPS* yang dimiliki bumi mempunyai konstelasi 24 *satelit* dalam enam *orbit* yang mendekati lingkaran, setiap *orbit* ditempati oleh 4 buah *satelit* dengan *interval* antara yang tidak sama. *Orbit* satelit *GPS* berinklinasi *550°* terhadap bidang *equator* dengan ketinggian rata-rata dari permukaan bumi sekitar 20.200 *km*.

Metode *Advanced Positioning* yang terdapat pada *A-GPS* merupakan metode penentuan posisi yang paling tinggi akurasinya dibandingkan metode deteksi posisi lainnya seperti misalnya *Time Difference Of Arrival (TDOA),* maupun *Enhanced Observed Time Difference (E-OTD)* sehingga *A-GPS* jauh lebih efisien dan efektif dalam mengakses informasi dari satelit karena tidak perlu mencari data satu persatu dari ke-24 *satelit* yang ada, namun *A-GPS* telah mengetahui sasaran (*satelit*) mana yang dibutuhkan atau dituju.

Prinsip kerja dari *GPS* adalah *Satelit GPS* memberikan informasi kepada *receiver GPS* mengenai jarak/ posisi *satelit*. Sehingga kita tahu bahwa kita berada pada suatu *radius* tertentu dari *satelit*. Bila ada dua *satelit* maka kita tahu posisi kita, berada pada 2 lokasi, yaitu perpotongan dua *radius* tadi. *GPS receiver* mampu menghitung tempat yg paling mungkin. Semakin banyak sinyal *satelit* ditangkap semakin teliti *satelit* menghitung posisi.

2.9 *Google Maps*

Google Maps merupakan sebuah layanan peta dunia virtual berbasis *Web* yang disediakan Google. Layanan ini gratis dan bisa ditemukan serta diakses di *http://maps.google.com*. Google *Maps* menawarkan peta yang dapat digeser (*panned*), diperbesar (*zoom* i*n*), diperkecil (*zoom out*) dan dapat diganti dalam beberapa mode, fitur pencarian rute (*routing*), penunjuk arah dari satu objek peta ke objek yang lain (*direction*), dan juga pencari tempat (*place*).

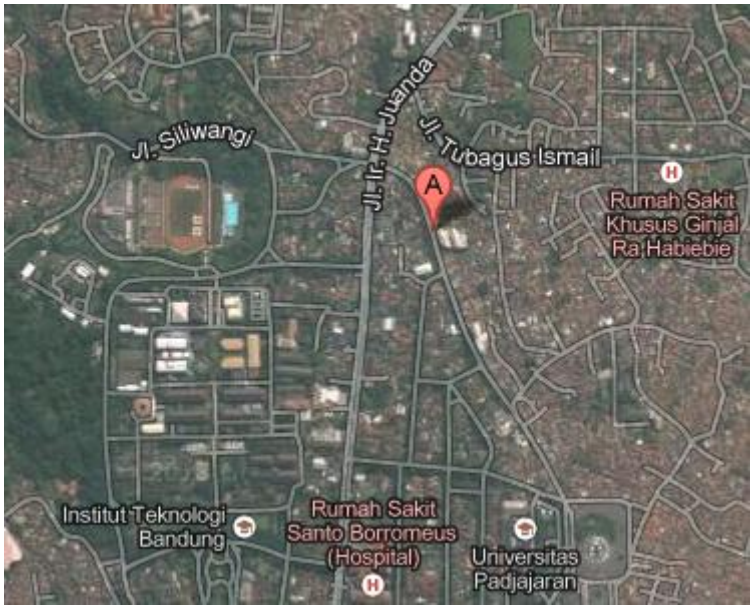
Sementara ini Google *maps* memiliki dua mede untuk menampilkannya diantaranya adalah sebagai berikut :

1. *Maps* menampilkan peta dalam bentuk garis contohnya adalah sebagai berikut pada gambar 2.13 :



Gambar 2.13 Contoh tampilan *Maps* Google

2. *Satelite* , Menampilkan peta dalam bentuk *citra satellite* seperti yang di tunjukan pada gambar 2.14 di bawah ini :



Gambar 2.14 Contoh Gambar *satelite* Google *Maps*

2.9.1 Google *Maps API*

Google Maps API adalah kumpulan *API* yang memungkinkan Anda menghamparkan data Anda di Google *Map* yang disesuaikan. Anda dapat membuat aplikasi *Web* dan seluler menarik dengan *platform* pemetaan canggih dari Google termasuk basis data *citra satelit*, pemandangan jalan, profil ketinggian, petunjuk arah mengemudi, peta dengan sentuhan gaya, demografi, analisis, dan tempat yang luas. Dengan cakupan *global* yang paling akurat di dunia dan komunitas pemetaan yang aktif dalam membuat pembaruan setiap harinya, pengguna Anda akan mendapatkan manfaat dari layanan yang terus-menerus ditingkatkan. Dalam pembuatan program yang mengimplementasikan Google *Maps API* ada beberapa urutan berikut ini adalah tingkatannya :

1. Memasukkan *Maps API JavaScript* ke dalam *HTML* kita.
2. Membuat *element div* dengan nama *map_canvas* untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property -properti pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body *HTML* dengan *event onload*.

2.10 Pembangunan Perangkat Lunak

Dalam sebuah pembuatan perangkat lunak di perlukan beberapa *tools* pendukung untuk merancang dan membangun sebuah perangkat lunak yang bisa di gunakan oleh *user*.

2.10.1 *ADCS3* (*Adobe Dreamweaver CS3*)

Menurut Alexander F.K. Sibero, 2011, dikutip dalam (Utomo & Bakara, 2013) Adobe Dreamwaver CS3 adalah suatu produk *Web Developer* yang dikembangkan oleh Adobe Sistem Inc. Sebelumnya produk Dreamweaver dikembangkan oleh Macromedia Inc. Yang kemudian sampai saat ini pengembangaannya diteruskan oleh Adobe Sistem Inc. Setelah diambil oleh Adobe Sistems Inc, dreamweaver dikembangkan dan dirilis dengan kode nama Creative Suit (CS). Ruang kerja atau *Workspace* adalah bagian keseluruhan tampilan Adobe dreamweaver yang terdiri dari *Welcome Screen*, *Menu*, *Insert bar*, *Document Window*, *CSS Panel*, *Application Panel*, *Tag Inspector Property Inspector*, *Result Panel*, dan *Files Panel*, masing–masing dari komponen tersebut memiliki fungsi dan aturan.

Adobe dreamweaver merupakan *Web* editor yaitu progam aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen *Web* baik *client slide* scripting maupun *server* side *scripting*. Saat ini banyak tersedia *Web* editor mulai dari yang paling sederhana hingga yang lebih *smart*. Mulai dari *Web* editor yang berbayar hingga yang gratis.

2.10.2 *MYSQL*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: *database management* sistem) atau *DBMS* yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQLAB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

MySQL adalah *Relational Database Management* Sistem (*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 *closed source* atau 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 optimizernya 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*. 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. ***OpenSource*** : *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 dan 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 dan limits*** : *MySQL* mampu menangani *database* dalam skala besar, dengan jumlah *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. ***Connectivity*** : *MySQL* dapat melakukan koneksi dengan client menggunakan protokol *TCP/IP*, *Unix soket* (*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 dan tools*** : *MySQL* dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.
13. **Struktur tabel** : *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan *database* lainnya semacam *PostgreSQL* ataupun *Oracle*.

Untuk melakukan administrasi dalam basis data *MySQL*, dapat menggunakan modul yang sudah termasuk yaitu *command-line* (perintah: *mysql* dan *mysqladmin*). Juga dapat diunduh dari situs *MySQL* yaitu sebuah modul berbasis grafik (*GUI*): *MySQL Administrator* dan *MySQLQuery Browser*. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data *MySQL* berbasis *Web* yang sangat populer yaitu *phpMyAdmin*. Untuk perangkat

lunak untuk administrasi basis data *MySQL* yang dijual secara komersial antara lain: *MySQL* front, *Navicat* dan *EMS SQL Manager for MySQL*.

2.10.3 *SDK* (*Software Development Kit*)

Android-*SDK* merupakan *tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis google android. Android *SDK* mencakup seperangkat alat pengembangan yang komprehensif. Android *SDK* terdiri dari *debugger*, *libraries*,*handset emulator*,dokumentasi, contoh kode, dan *tutorial*.

2.10.4 *JDK* (Java *Development Kit*)

JDK (Java Development Kit) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemrograman *Java*. *JDK* berjalan diatas sebuah *virtual machine* yang dinamakan *JVM (Java Virtual Machine).* Dokumentasi *JDK* berisi spesifikasi *API*, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas *JDK* dan *utilitas*, demo, dan *link* ke informasi terkait. Dokumentasi ini juga tersedia dalam *bundle* yand dapat diunduh dan diinstal pada computer.

2.10.5 ECLIPSE

Eclipse adalah sebuat komunitas proyek *open source* yang difokuskan pada membangun sebuah *platform* pengembangan yang dapat diperluas, *runtime* dan kerangka aplikasi untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh daur hidup perangkat lunak. Komunitas *Eclipse* memiliki lebih dari 60 proyek *open source*. Proyek-proyek ini secara konsep terbagi menjadi 7 kategori diantaranya adalah sebagai berikut :

1. *Enterprise Development*
2. *Embedded and Device Development*
3. *Rich Client Platform*
4. *Rich Internet Applications*
5. *Application Frameworks*

6. *Application Lifecycle Management (ALM)*

7. *Service Oriented Architecture (SOA)*

Komunitas *Eclipse* juga didukung oleh ekosistem besar dan dinamis dari penyedia solusi teknologi informasi utama, start-up inovatif, universitas dan lembaga penelitian, dan individu yang memperluas, mendukung, dan melengkapi *platform Eclipse*.

2.10.6 *ADT* (*Android Development Tools*)

Android Development Tools (ADT) adalah plugin yang didesain untuk *IDE Eclipse* yang memberikan kita kemudahan dalam mengembangakan aplikasi android dengan menggunakan *IDE Eclipse*. Dengan menggunakan *ADT* untuk *Eclipse* memudahkan dalam membuat aplikasi *project* android, membuat *GUI* aplikasi , dan menambahkan komponen-komponen yang lainnya, begitu juga dapat melakukan running aplikasi menggunakan Android *SDK* melalui *eclipse*.