BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Organisasi

Tinjauan organisasi adalah untuk mengetahui keadaan diorganisasi diantaranya adalah sejarah berdirinya organisasi, struktur organisasi serta visi dan misi.

2.1.1. Sejarah Organisasi

Saung Angklung Udjo (disingkat SAU), didirikan oleh Udjo Ngalagena (alm) yang akrab dengan panggilan Mang Udjo dan isterinya, Uum Sumiati. Mang Udjo dikenal sebagai pembuat angklung sejak tahun 1966, yang didasarkan atas hobi. SAU merupakan sanggar seni sebagai tempat pertunjukkan seni, laboratorium pendidikan sekaligus sebagai obyek wisata budaya khas Jawa Barat, dengan mengandalkan semangat gotong royong antar sesama warga.

SAU terletak di timur Bandung, tepatnya di Jalan Padasuka 118 Bandung 40192.

2.1.2. Visi Dan Misi Organisasi

Visi dari Saung Angklung Udjo adalah untuk mengembangkan budaya Sunda, khususnya budaya musik bambu dan menjadi tujuan wisata utama di Indonesia.

Adapun misi dari Saung Angklung Udjo adalah sebagai berikut :

1. Mengelola organisasi secara sehat dan didukung oleh teknologi tepat guna serta sumber daya manusia yang profesional Mengembangkan organisasi agar memiliki daya saing yang kuat.
2. Mengelola organisasi secara sehat dan didukung oleh teknologi tepat guna serta sumber daya manusia yang profesional.
3. Mengembangkan organisasi agar memiliki daya saing yang kuat.

2.2. Sejarah Dan Perkembangan Angklung

Angklung adalah alat musik tradisional Indonesia yang terbuat dari bambu. Sebuah angklung terdiri beberapa tabung bambu (tergantung fungsinya) yang berbeda ketinggian dan diameternya untuk mencapai harmoni nada yang diinginkan. Sebuah angklung melodi biasanya terdiri dari dua tabung yang menghasilkan nada terpaut satu oktaf,sementara angklung pengiring (accompagnement)terdiri daritiga atau bahkan empat tabung tergantung accord yang dimainkan. Tabung-tabung tersebut kemudian diikatkan pada rangka batang bambu untuk membentuk alat musik angklung yang lengkap.

Sebuah angklung hanya menghasilkan satu nada, jadi untuk memainkan sebuah lagu dibutuhkan beberapa set angklung yang dimainkan oleh banyak orang. Kurang lebih seperti kelompok paduan suara dalam membawakan sebuah lagu. Untuk memainkannya, kita cukup menggoyangkan atau menggetarkannya. Tangga nada yang digunakan pada angklung adalah tangga nada mayor, yang memiliki pola 1-1-1/2-1-1-1-1/2.

Angklung dipercayai berasal dari pulau Jawa, khususnya tanah Sunda. Beberapa catatan dari orang Eropa yang melakukan perjalanan ke tanah Sunda pada abad 19 mengatakan bahwa di daerah ini sering terlihat "permainan" angklung oleh orang-orang setempat. Angklung memang juga dikenal di daerah-daerah lain di pulau Jawa, tetapi di tanah Sunda alat musik ini lebih popular. Pada awalnya, angklung tradisional digunakan oleh orang-orang desa pada masa itu sebagai bagian dari ritual kepada Dewi Sri untuk meminta panen melimpah. Umumnya dibawakan dalam tangga nada pentatonis (terdiri dari lima nada) dan memainkan melodi yang berulang. Acara seperti ini biasanya dilakukan di ruang terbuka, sambil menari-nari dengan dengan diiringi alat musik tradisional lain seperti goong, kendang, dan tarompet. Kesenian semacam ini masih dilestarikan di beberapa tempat di Jawa Barat.

Awal abad 20, angklung tradisional mulai menghilang. Pada tahun 1938, Daeng Sutigna, seorang guru berpendidikan Belanda di Bandung, menciptakan angklung dalam tangga nada diatonis yang terdiri dari tujuh nada. Hal ini menandai lahirnya angklung modern. Kelebihan angklung ini adalah ia dapat membawakan lagu-lagu Barat klasik dan popular yang rata-rata bernada diatonis, sehingga dapat menjangkau selera musik masyarakat yang lebih luas. Kini lagu yang dimainkan tidak lagi berkisar pada lagu-lagu tradisional, tetapi juga lagu-lagu klasik, lagu pop, new age, bahkan lagu rock. Dengan angklung modern, lagu rock melodius seperti We Are the Champion dan Bohemian Rapshody dari Queen dapat dibawakan oleh alat musik angklung.

Angklung "jenis baru" ini pertama kali diperkenalkan pa Daeng kepada sekelompok anak-anak pramuka. Setelah dipertunjukkan oleh murid-murid sekolah pada acara Konferensi Asia Afrika tahun 1955 di Bandung, angklung diatonis atau angklung modern ini semakin dikenal masyarakat hingga saat ini menjadi kegiatan ekstrakurikuler di berbagai sekolah. Memang pada awalnya pak Daeng menginginkan angklung sebagai alat pendidikan. Mottonya adalah 5 M :

Murah, Mudah, Menarik, Massal, dan Mendidik. Murah, karena bahan-bahan untuk membuat alat musik ini murah dan mudah didapat di Indonesia. Mudah, karena untuk memainkan angklung seseorang tidak perlu memiliki keterampilan khusus. Menarik, dilihat dari keunikannya bentuknya dan cara memainkannya. Massal karena untuk memainkannya melibatkan banyak orang. Dan Mendidik dalam arti alat musik ini memiliki unsur pendidikan selain musik.

2.3. Landasan Teori

2.3.1. Konsep Dasar Sistem

Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling tergantung dan berintegrasi dalam kesatuan variabel atau komponen (Riyanto, Putra dan Indelarko, 2009). Terdapat dua kelompok pendekatan, yaitu menekankan pada prosedur dan komponen atau elemennya.

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkelompok dan bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur sendiri, yaitu urutan yang tepat

dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya.

Sedangkan pendekatan yang menekankan pada komponen, mendefinisikan "sistem" sebagai berikut,

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Beberapa penulis mendefinisikan "sistem" yang menekankan pada elemen atau komponennya.

2.3.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu di dalamnya, karakteristik sistem tersebut yaitu :

1. Komponen sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk suatu kesatuan.

2. Batas sistem

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Maka suatu suatu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk suatu satu kesatuan.

2.3.3. Informasi

Informasi adalah arti dari hubungan dan penafsiran data yang mengijinkan seseorang membuat keputusan. Informasi dikatakan berharga jika informasi itu mempengaruhi proses pengambilan keputusan lebih baik. Sasaran utama dari sistem informasi adalah menyediakan informasi yang akurat dan penting. Informasi juga dapat berarti beberapa kesatuan yang tak terukur yang dapat mengurangi ketidakpastian tentang suatu peristiwa atau langkah.

Informasi merupakan pemrosesan data yang tampak dalam konteks untuk menyampaikan arti kepada orang lain. Definisi lain dari Informasi, yaitu (Riyanto, Putra dan Indelarko, 2009).

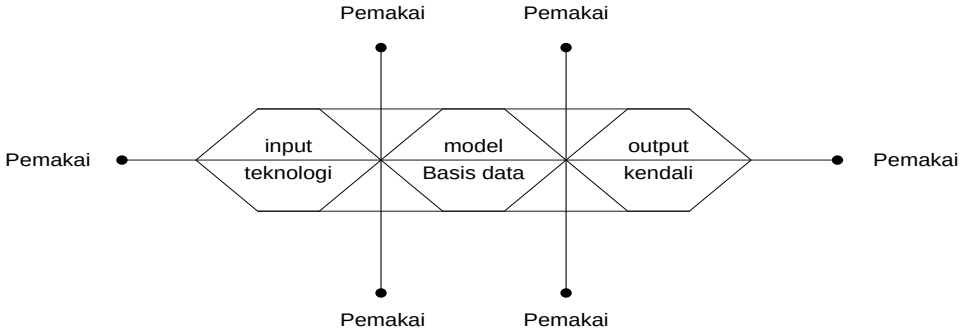
> Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.3.4. Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Riyanto, Putra dan Indelarko, 2009).

2.3.5. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan, yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*controls block*). Sebagai suatu sistem keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.



Gambar II.1 Blok Sistem Informasi yang berinteraksi. (Sumber: Riyanto, Putra dan Indelarko, 2009

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi input dan data yang tersimpan di *databasei* dengan cara yang tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

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

4. Blok Teknologi

   Teknologi merupakan *toll 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 keseluruhan. Teknologi terdiri dari tiga bagian utama, yaiut teknisi (*humanware* dan *brainware*), perangkat lunak (*software*) dan perangkat keras (hardware). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi.

5. Blok Basis Data

   Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam *database* untuk keperluan penyediaan informasi lebih lanjut. Data di dalam *database* perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

   Organisasi *database* yang baik juga berguna untuk efisiensi kapasitas penyimpannya. *Database* diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management System*).

6. Blok Kendali

   Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sitem itu sendiri, kesalahan-kesalahan, ketidakefisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.3.6. Internet

2.3.6.1. Pengertian Internet

Internet merupakan jaringan global komputer dunia, besar dan sangat luas sekali dimana setiap komputer saling terhubung satu sama lainnya dari negara ke negara lainnya di seluruh dunia dan berisi berbagai macam informasi, mulai dari text, gambar, audio, video, dan lainnya.

Internet itu sendiri berasal dari kata Interconnection Networking, yang berarti hubungan dari banyak jaringan komputer dengan berbagai tipe dan jenis, dengan menggunakan tipe komunikasi seperti telepon, salelit, dan lainnya.

Dalam mengatur integrasi dan komunikasi jaringan komputer ini menggunakan protokol yaitu TCP/IP. TCP (*Transmission Control Protocol*) bertugas untuk memastikan bahwa semua hubungan bekerja dengan benar, sedangkan IP (*Internet Protocol*) yang mentransmisikan data dari satu komputer ke komputer lain. TPC/IP secara umum berfungsi memilih rute terbaik transmisi data, memilih rute alternatif jika suatu rute tidak dapat di gunakan, mengatur

dan mengirimkan paket-paket pengiriman data. Untuk dapat ikut serta menggunakan fasilitas Internet, Anda harus berlangganan ke salah satu ISP (*Internet Service Provider*) yang ada dan melayani daerah Anda. ISP ini biasanya disebut penyelenggara jasa internet. Anda bisa menggunakan fasilitas dari Telkom seperti Telkomnet Instan, speedy dan juga layanan ISP lain seperti first media, netzip dan sebagainya.

Internet memberikan banyak sekali manfaat, ada yang bisa memberikan manfaat baik dan buruk. Baik bila digunakan untuk pembelajaran informasi dan buruk bila digunakan untuk hal yang berbau pornografi, informasi kekerasan, dan lain-lainnya yang negatif.

Internet ini memungkinkan pengguna komputer di seluruh dunia untuk saling berkomunikasi dan berbagi informasi dengan cara saling mengirimkan email, menghubungkan komputer satu ke komputer yang lain, mengirim dan menerima file dalam bentuk text, audio, video, membahas topik tertentu pada *newsgroup, website social networking* dan lain-lain.

2.3.6.2. Layanan Aplikasi Di Internet

Aplikasi Internet yang tersedia saat ini sudah banyak dan terus bertambah seiring dengan kemajuan teknologi infomasi. Aplikasi-aplikasi internet ini kemudian digunakan dalam berbagai bidang seperti bidang akademis, militer, medis, media massa, dan berbagai sektor industri lainnya. Dari sekian banyak aplikasi internet yang ada, yang banyak dikenal dan digunakan antara lain : *Word Wide Web* (www), *Electronik mail, Mailing List (milis),*

Newsgroup, Internet Relay Chat (IRC), File Transfer Protocol (FTP), Telnet, Gopher, dan Ping.

1. *Word Wide Web* (WWW)

   World Wide Web (WWW) adalah aplikasi yang paling menarik di internet dan seperti e-mail, aplikasi ini sangat penting dan banyak digunakan. Dokumen Web tersebut dibuat dengan format *hypertext* dan *hypermedia* dengan menggunakan *Hypertext Markup Language* (HTML). HTML mempunyai kemampuan untuk menghubungkan (link) sebuah dokumen dengan dokumen yang lain. Dokumen yang dibuat dengan HTML tersebut dapat memuat teks, gambar, animasi, audio dan video. Aplikasi ini kadang disebut *The Killer Application atau the world is at your fingertip* karena anda bisa mendapatkan informasi dengan mudah sekali bukan sekadar teks bahkan gambar (images), maupun multimedia. Dalam aplikasi ini banyak kemudahan yang dapat dilakukan antara lain memesan atau membeli suatu barang secara online, mendaftar secara online, mencapai multimedia. Informasi yang diletakkan di WWW disebut *homepage* dan setiap *homepage* mempunyai alamatnya masing-masing. Untuk dapat menarik perhatian pengguna sehingga *homepage* dapat sering dikunjungi, maka anda harus dibangun semenarik mungkin dan kaya akan informasi yang jelas. Dalam hal ini bidang seni sangat diperlukan, sehingga dunia periklanan dan dunia perniagaan semakin sinergi.

2. *Electronic Mail* (E-mail)

   Electronic mail (E-mail) adalah salah satu kemudahan atau aplikasi yang paling banyak digunakan di internet. Hal ini karena *e-mail* merupakan alat komunikasi yang paling murah dan cepat. Melalui *e-mail* anda dapat berhubungan dengan siapa saja yang terhubung ke internet di seluruh dunia dengan biaya relatif terjangkau. Konsep *e-mail* adalah seperti anda mengirim surat dengan pos biasa, dengan beralamatkan tempat yang anda tuju. Dari pos tersebut akan disampaikan ke pos yang terdekat dengan alamat yang dituju dan akhirnya sampai ke alamat tersebut. Penerima hanya membuka kotak pos saja yang ada di depan rumah, sedangkan pengirim tidak tahu apakah orang yang dituju tersebut sudah menerima surat tersebut, sampai surat itu dibalas. Dengan *e-mail* data dikirim secara elektronik sehingga sampai di tujuan sangat cepat. Anda juga dapat mengirim file-file seperti program, gambar, grafik dan sebagainya, dan dapat mengirim ke lebih dari satu orang sekaligus dalam satu waktu.

3. *Mailing List* (Milis)

   Mailing list atau sering juga disebut milis adalah aplikasi internet yang digunakan sebagai sarana diskusi atau bertukar informasi dalam satu kelompok melalui e-mail. Setiap e-mail yang dikirim ke alamat milis akan dikirim ke seluruh alamat e-mail yang terdaftar sebagai anggota milis tersebut. Milis sebenarnya bentuk lain dari *e-mail*. Fasilitas ini digunakan oleh kelompok-kelompok untuk bertukar informasi dan berdiskusi sesama anggota kelompok. Milis mempunyai sifat yang sama dengan e-*mail*.

4. *Newsgroup*

   Newsgroup adalah forum perbincangan, atau boleh dibayangkan sebagai suatu tempat di mana terdapat ruangan-ruangan perbincangan yang unik, dan tiap-tiap ruangan mempunyai topik perbincangan yang berbeda. Di setiap ruangan itu biasanya terdapat lebih dari satu orang yang saling bertukar pendapat atau pikiran. Jadi Anda memberikan pendapat Anda ke semua orang yang ada di ruangan tersebut. *Newsgroup* juga dianggap seperti "Bulletin *Board*" yang ada di sekolah atau kantor, di mana setiap orang boleh meletakkan artikel-artikel atau pendapat-pendapatnya dan boleh dilihat dan dibaca oleh semua orang. Dengan aplikasi *newsgroup* artikel atau surat yang Anda kirim dapat dengan cepat terletak di grup yang Anda inginkan.

5. *Internet Relay Chat* (IRC)

   Internet Relay Chat (IRC) adalah aplikasi internet yang digunakan untuk bercakap-cakap di internet . Bercakap-cakap di internet dikenal dengan istilah *chatting*. *Chatting* dilakukan dengan cara mengetik apa yang ingin anda katakan kepada teman *chatting* anda, kemudian mengirimkannya dalam bentuk teks. Melalui *chating* Anda dapat berkenalan dan bercerita dengan orang lain di belahan dunia lain.

6. *File Transfer Protocol* (FTP)

   File transfer protocol yang disingkat dengan FTP adalah aplikasi internet yang di gunakan untuk mengirimkan atau mengambil file ke atau dari sebuah komputer lain. FTP memungkinkan transfer data lebih cepat. FTP

sering digunakan untuk mencari dan mengambil (*download*) arsip file di suatu server di internet yang mempunyai alamat tertentu yang menyediakan berbagai arsip (file). File tersebut memang diizinkan untuk diambil oleh penguna internet yang membutuhkan. File ini dapat berupa hasil penelitian, artikel-artikel, jurnal dan lain-lain. FTP juga dipergunakan untuk meng-upload file materi situs (*homepage*) sehingga dapat diakses oleh penguna dari pelosok dunia.

7. Telnet

   Telnet adalah aplikasi internet yang digunakan untuk mengakses komputer yang letaknya jauh. Telnet dapat Anda gunakan apabila Anda mempunyai alamat IP (IP Address) dari komputer yang akan diakses dan Anda juga harus mempunyai hak akses (User ID dan password).

8. Gopher

   Gopher adalah aplikasi yang digunakan untuk mencari informasi yang ada diinternet. Namun informasi yang dicari hanya terbatas pada teks saja. Untuk mendapatkan informasi melalui gopher, Anda harus menghubungkan diri dengan gopher server yang ada di internet .

9. Ping

   Ping singkatan dari paket internet gopher. Ping digunakan untuk mengetahui apakah komputer yang Anda gunakan mempunyai sambungan (terkoneksi) dengan komputer lain di internet. Pengecekan hubungan ini dilakukan dengan cara mengirimkan sejumlah paket data.

Melalui aplikasi yang ada di internet, keuntungan internet yang dapat Anda rasakan antara lain sebagai berikut :

a. Sarana untuk mendapatkan dan menyampaikan informasi yang cepat dan murah. Hal ini diperoleh dengan menggunakan aplikasi. *e-mail*, WWW, *Newsgroup*, FTP dan Gopher.
b. Mengurangi biaya kertas dan biaya distribusi. Misalnya Koran nasional dapat di cetak di daerah, koran masuk internet, majalah, brosur, dan sebagainya dapat di tampilkan lewat internet .
c. Sebagai media promosi. Internet dimanfaatkan sebagai sarana untuk beriklan dan menyampaikan profil perusahaan dan produk-produknya.
d. Sarana komunikasi interaktif. Komunikasi via internet dapat dilakukan dengan menggunakan fasilitas *e-mail*, WWW, *Video Conferencing*, *Internet Relay Chat*, dan Internet Phone.
e. Sebagai alat *Research and Development*.
f. Sarana untuk bertukar data.

2.3.7. Basis Data

2.3.7.1. Pengertian Basis Data

Sistem Basis Data adalah sistem yang terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan (Rosa dan Shalahuddin, 2011). Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah. Sistem Informasi tidak dapat dipisahkan dengan kebutuhan akan basis data

apapun bentuknya, baik berupa *file* teks ataupun *Database Management System* (DBMS)**.**

2.3.7.2. DBMS

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai SIstem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelolah, dan menampilkan data (Rosa dan Shalahuddin, 2011).

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational DBMS* yaitu DBMS berbasis relasioanl model. Pada tahun 1980an mulai berkembang *Object Oriented DBMS* (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berbasis objek.

Berikut ini 4 macam DBMS versi komersial yang paling banyak digunakan di dunia saat ini, yaitu :

1) Oracle
2) Microsoft SQL Server
3) IBM DB_2
4) Microsoft Access

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut:

1) MySQL
2) PostgreSQL
3) Firebird

4) SQLite

Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelolah data pada DBMS.

2.3.8. Web Service

Web Services adalah sebuah layanan yang disediakan sebuah sistem untuk meningkatkan kolaborasi antar sistem. Webservice disediakan oleh suatu sistem, misalkan sebuah website yang menyediakan layanan kepada sistem yang lain.

Komunikasi antar sistem tersebut menggunakan sebuah format yang bersifat universal, sehingga walaupun sistem yang berhubungan adalah berbeda platform, sistem operasi maupun berbeda bahasa pemrograman akan tetap bisa saling komunikasi.

Isi webservice berupa data atau informasi yang diminta oleh client dalam bentuk sebuah format yang bersifat universal (JSON atau XML). Sedangkan Yang dibutuhkan Untuk membuat sebuah webservice adalah sebuah webserver seperti Apache atau lightpd, apabila ada pemrosesan data maka diperlukan sebuah bahasa pemrograman atau server side scripting yang kemudian mengubah hasil pengelolahan data tersebut menjadi format JSON atau XML.

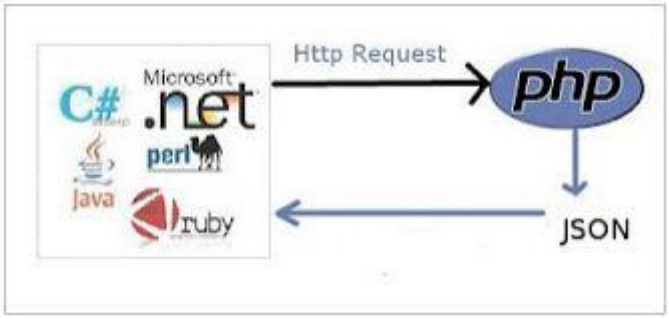
Sebuah sistem dapat menyediakan Webservice dengan menggunakan bahasa pemrograman apa saja seperti PHP/Java/Ruby/ C/Phtyon/Basic yang penting webservice tersebut dapat diakses melalui protokol HTTP.

Sedangkan client atau pemakai webservice tersebut dapat menggunakan bahasa pemrograman apa saja seperti PHP/Javascript/ Java/Ruby/C/Phtyon atau

apa saja dan tidak harus sama dengan bahasa pemrograman yang dipakai oleh penyedia Webservice. Yang penting client tersebut dapat melakukan request HTTP dan dapat melakukan parsing format JSON atau XML.

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999.

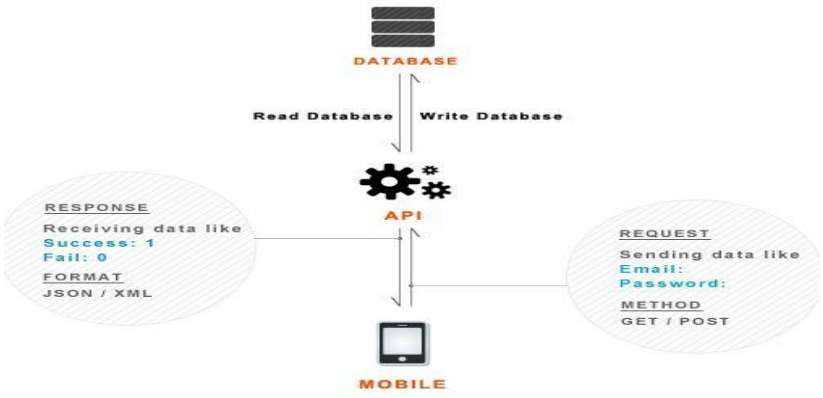
Workflow Webservice dengan menggunakan PHP adalah seperti gambar dibawah ini :



Gambar II. 2 Komunikasi Webservice menggunakan PHP

2.3.8.1. Pembentukan API

Secara umum, pembentukan API dapat dilihat pada gambar II.3



Gambar II.3 Pembentukan API

Pada gambar diatas, ada tiga komponen yang membuat API berjalan. Ketiga komponen itu adalah:

1. **API**, merupakan penyedia *service* yang dibutuhkan oleh aplikasi dalam *mobile* untuk mengakses *database*.
2. ***Database***, merupakan penyedia data yang akan diparsing oleh API melalui PHP dan ditampilkan ke aplikasi dalam *mobile*.
3. ***Mobile***, merupakan tempat dimana berawalnya *Request* dan berakhirnya *Respon* dari *mobile* kepada API yang menjalankan *method* dalam PHP.

2.3.9. Android

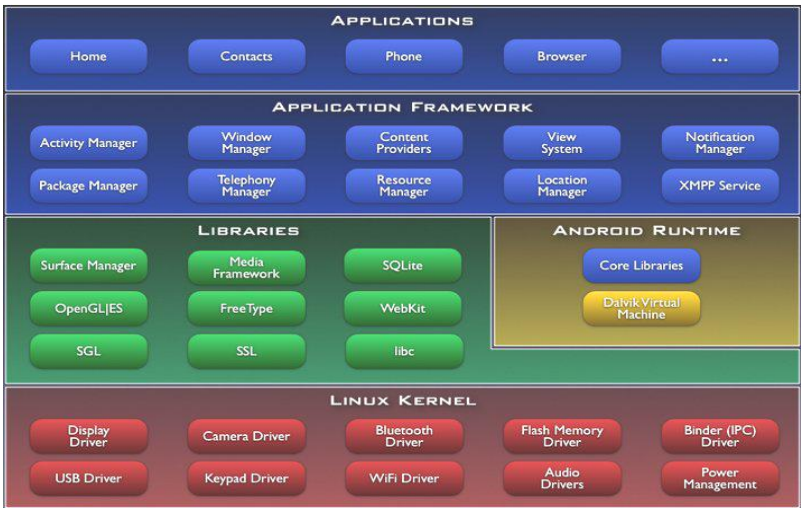
Android merupakan suatu *software stack* untuk *mobile device*. Di dalamnya terdapat sistem operasi, *middleware*, dan *key application*. Aplikasi pada platform ini dikembangkan dalam bahasa pemrograman java. Android sendiri memiliki banyak fitur diantaranya adalah:

1. Merupakan sebuah **A*pplication Framework*** sehingga *programmer* dapat menggunakan beberapa fungsi yang telah disediakan.
2. **Dalvik *virtual machine***. Tiap aplikasi dalam Android memiliki *instance virtual machine* yang dapat bekerja secara efisien dalam lingkungan memori yang terbatas.
3. ***Integrated browser*.** Web browser berbasis WebKit *engine* terdapat pada browser *default* Android atau pun dapat diintegrasikan dengan aplikasi lain.
4. ***Optimized graphics***. *Library* grafis 2D yang kaya dan 3D berbasis OpenGL ES 1.0 yang mendukung akselerasi hardware.

5. **SQLite**. Basis data relasional yang ringan namun sangat *powerful*.
6. ***Media Support***. Mendukung berbagai format audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
7. **GSM *Telephony***. Mendukung fungsi komunikasi GSM.
8. **Bluetooth, EDGE, 3G** dan **WiFi**. Mendukung komunikasi pada jaringan (tergantung hardware).
9. **Kamera**, **GPS**, **kompas** dan ***accelerometer***. Mendukung berbagai fitur yang disediakan oleh *hardware*.
10. ***Tools Developer*** yang lengkap. Termasuk *device emulator*, *tools* untuk *debugging*, *profiling* memori dan performa, *plugin* untuk Eclipse IDE.

2.3.9.1. Arsitektur Android

Android terdiri dari beberapa *stack software* yang terdiri dari: *Applications, Application Framework, Libraries, Android Runtime* dan *Kernel Linux.* Arsitektur lengkap platform ini dapat dilihat pada Gambar II.4.



Gambar II.4 Arsitektur platform android

2.3.9.2. Application

Application merupakan program yang langsung berhubungan dengan *user*. Baik program yang merupakan bawaan dari Android sendiri maupun program yang dibuat oleh *developer* menggunakan bahasa pemrograman java. Contoh program bawaan dari platform Android sendiri adalah *email client*, program SMS, *calendar*, *maps*, *web browser*, *contact* dan sebagainya.

2.3.9.2.1. *Aplication Framework*

Lapisan ini berisi sekumpulan API yang dapat digunakan oleh *programmer* maupun *core application* dari Android. Lapisan ini dirancang untuk memudahkan penggunaan komponen dari Android sendiri. Aplikasi manapun dalam Android dapat berbagi fungsi sehingga aplikasi lain dapat memanfaatkannya.

Aplikasi pada Android disusun atas beberapa komponen:

1. Sekumpulan *Views*. Digunakan untuk mengatur tampilan pada aplikasi. Contohnya adalah *lists, grids, text box, button*, bahkan *embeddable web browser.*
2. *Content providers.* Komponen yang mengatur agar aplikasi dapat mengakses *resources* dari aplikasi lain (seperti *Contacts*), atau berbagi data dengan aplikasi lain.
3. *Resource Manager*. Menyediakan akses ke pada *resource non-code* seperti *localized string*, grafik dan *file layout*.
4. *Notification Manager*. Memungkinkan agar suatu aplikasi dapat menampilkan peringatan yang dapat di kostumasi pada *status bar*.

5. *Activity Manager*. Mengatur siklus aplikasi dan navigasi antar aplikasi yang sedang berjalan.

2.3.9.2.2. *Libraries*

Android mendukung beberapa *library* C/C++ yang digunakan pada berbagai komponen Android. Kemampuan ini dapat diakses oleh *developer* melalui Android *application framework.* Beberapa *library* diantaranya adalah:

1. *System* C *library*. Implementasi *library* C standar (libc).
2. *Media Libraries*. Mendukung berbagai format multimedia (termasuk MPEG4,H.264, MP3, AAC, AMR, JPG, PNG).
3. *Surface Manager*. Mengatur akses ke subsistem *display*.
4. LibWebCore. *Engine web browser* modern.
5. SGL. *Engine* grafis 2D.
6. 3D *Library*. Implementasi OpenGL ES 1.0 yang mendukung akselerasi hardware.
7. FreeType. *Rendering* untuk *bitmap* dan *vector font.*
8. SQLite. Basis data relasional yang kecil namun sangat ampuh.

2.3.9.2.3. *Android Runtime*

Tiap aplikasi pada Android memiliki proses-nya masing-masing. Tiap aplikasi tersebut memiliki *instans* dari Dalvik *virtual machine* (VM). Dalvik *virtual machine* dirancang agar suatu *device* dapat menjalankan beberapa VM secara efisien. Dalvik VM *mengeksekusi* file dengan format Dalvik *Executable format* (.dex) yang dirancang untuk meminimalkan *memory footprint* . Dalvik VM berbasis *register*, dan dapat menjalankan kelas-kelas yang dikompilasi dengan

bahasa pemrograman java dan ditransformasikan menjadi format .dex. Dalvik VM sendiri bergantung pada Kernel Linux untuk fungsi dasarnya, seperti *threading* dan manajemen memori secara *low-level*.

2.3.9.2.4. *Linux Kernel*

Android menggunakan Kernel Linux versi 2.6 sebagai sistem utama. Fungsi kernel yang digunakan antara lain untuk keamanan, manajemen memori, manajemen proses, manajemen jaringan dan *driver model*. Kernel juga berfungsi sebagai *layer* abstrak antara *hardware* dan lapisan lainnya pada *software stack*.

2.3.9.2.5. *Anatomi Aplikasi Android*

Terdapat empat *building blocks* pada aplikasi Android yaitu:

- *Activity*
- *Intent Receiver*
- *Service*
- *Content Provider*

Tidak semua aplikasi membutuhkan keempat blok ini, tetapi suatu aplikasi dibuat menggunakan kombinasi beberapa blok ini. Setelah memutuskan blok komponen yang akan digunakan, blok tersebut didaftarkan pada suatu *file* yang disebut dengan AndroidManifest.xml. *File* XML ini digunakan untuk menyatakan komponen apa saja yang dibutuhkan oleh sebuah aplikasi dan kemampuan serta kebutuhan aplikasi tersebut.

2.3.9.2.5.1. *Activity*

Activity paling umum diantara keempat *building block*. *Activity* biasanya suatu tampilan (*screen*) dalam suatu aplikasi. Tiap *activity* diimplementasikan

sebagai satu kelas yang meng-*extend* base class *Activity*. Kelas yang dibuat akan menampilkan *user interface* yang terdiri dari beberapa *views* dan akan merespon suatu *event*. Kebanyakan, aplikasi terdiri dari beberapa *screen*. Contohnya aplikasi pengiriman pesan akan memiliki *screen* untuk mendaftarkan pesan yang masuk, menulis pesan dan *screen* lainnya untuk melakukan pengaturan. Tiap *screen* pada contoh ini akan diimplementasi sebagai suatu *activity*. Aplikasi dapat berpindah ke *screen* lain dengan cara memulai suatu *activity* baru. Pada suatu kasus tertentu, *activity* akan mengembalikan nilai kepada *activity* sebelumnya. Contohnya *activity* untuk memilih foto akan mengembalikan foto yang dipilih pada *activity* yang memanggilnya. Pada saat *screen* baru terbuka, screen sebelumnya akan dihentikan dan *state* nya akan disimpan pada *history stack*. *User* dapat melakukan navigasi ke *activity* sebelumnya melalui *history* tersebut. *Screen* akan di hapus dari *history stack* apabila sistem memutuskan bahwa *screen* tersebut tidak layak untuk berada pada memori. Android menyimpan *history stack* untuk setiap aplikasi yang dijalankan dari tampilan awal (*home screen*).

2.3.9.2.5.2. *Intent and Intent Filter*

Android menggunakan kelas khusus bernama *intent* untuk berpindah dari *screen* ke *screen* lainnya. Sebuah *intent* mendeskripsikan apa yang ingin aplikasi penuhi. Bagian struktur data dari *intent* yang paling penting adalah *action* dan *data* dari aksi yang dilakukan. Nilai *action* yang biasanya adalah MAIN, VIEW, PICK, EDIT dan sebagainya. Data diekspresikan sebagai URI. Sebagai contoh, untuk melihat informasi suatu *contact* seseorang, kita membuat *intent* dengan action VIEW dan data sebagai URI yang merepresentasikan orang tersebut.

Terdapat suatu kelas yang dinamakan IntentFilter. Apabila *intent* adalah suatu permintaan untuk melakukan sesuatu, IntentFilter adalah deskripsi yang dapat dilakukan oleh *intent activity*. *Activity* yang dapat menampilkan informasi *contact* dari seseorang akan mempublikasikan IntentFilter yang menyatakan bahwa ia dapat menangani *action* VIEW apabila dipakai pada data yang merepresentasikan seseorang. Suatu aplikasi mempublikasikan IntentFilter pada *file* AndroidManifest.xml.

Navigasi dari suatu *screen* ke *screen* lainnya dipenuhi dengan cara me-*resolve intent*. Untuk melakukan navigasi ke depan, *activity* memanggil startActivity(myIntent). Sistem akan mencari IntentFilter untuk semua aplikasi dan memilih *activity* yang IntentFilter-nya yang paling mendekati dengan myIntent. *activity* baru akan mendapat informasi tentang *intent* yang menyebabkan *activity* tersebut dijalankan. Proses pencarian *intent* terjadi sewaktu runtime pada saat startActivity dipanggil. Hal tersebut membawa keuntungan sebagai berikut:

1. *Activity* dapat melakukan *reuse functionality* dari komponen lain dengan cara melakukan permintaan dalam bentuk *intent*.
2. *Activity* dapat di *replace* kapan pun dengan *activity* yang ekuivalen.

2.3.9.2.5.3. *Intent Recivier*

IntentReceiver dapat digunakan pada saat kita menginginkan kode kita dijalankan akibat reaksi dari *external event* tertentu. Contohnya, pada saat telepon berbunyi, atau pada saat ada data pada jaringan, atau pada suatu waktu tertentu. IntentReceiver tidak menampilkan *user interface*, meskipun ia dapat

menggunakan NotificationManager untuk memperingatkan *user* apabila sesuatu terjadi. *IntentReceiver* didaftarkan pada *file* AndroidManifest.xml, tetapi secara manual juga bisa ditambahkan dengan fungsi Context.registerReceiver(). Suatu aplikasi tidak harus berjalan pada *IntentReceiver*nya agar bisa dipanggil, tetapi sistem yang akan menjalankan aplikasi tersebut. Apabila diperlukan, pada saat IntentReceiver mendapat *trigger*, aplikasi juga dapat mengirimkan *intent broadcast* kepada aplikasi lainnya dengan fungsi Context.broadcastIntent().

2.3.9.2.5.4. *Service*

Service adalah sebuah kode yang hidup terus-menerus dan berjalan tanpa memiliki tampilan. Contoh yang sederhana adalah suatu aplikasi *media player* yang sedang memainkan lagu dari *play list.* Pada aplikasi *media player*, kemungkinan terdapat satu atau lebih activity yang memperbolehkan *user* memilih lagu dan memainkannya. Tetapi musik yang sedang diputar tidak boleh ditangani oleh activity karena *user* akan menginginkan musik tetap diputar meskipun ia berpindah ke *screen* baru. Pada kasus ini activity *media player* dapat menjalankan *service* menggunakan fungsi Context.startService() untuk menjalankan *service* di *background* yang memutar musik. Sistem akan tetap memutar musik sampai selesai. Selain itu *activity* dapat terhubung dengan *service* menggunakan fungsi Context.bindService(). Pada saat terhubung, aplikasi dapat berkomunikasi melalui *interface* yang diberikan oleh *service* yang bersangkutan. Pada contoh *media player*, *interface service* akan memberikan fungsi *pause*, *rewind* dan sebagainya.

2.3.9.2.5.5. *Content Provider*

Content provider digunakan untuk pertukaran data antara aplikasi. Data yang dipertukarkan bersifat *shared*. *Content provider* adalah sebuah kelas yang berisi *method* standar yang memungkinkan aplikasi lain dapat menyimpan dan mengambil suatu data yang ditangani oleh *content provider*.

2.3.9.2.5.6. *Siklus Hidup Aplikasi Android*

Pada banyak kasus, tiap aplikasi pada Android masing-masing memiliki Linux proses. Proses ini diciptakan untuk aplikasi tersebut pada saat kode program tersebut akan dieksekusi, akan terus berjalan sampai tidak dibutuhkan lagi dan sistem memerlukan memori untuk aplikasi lain. Konsep dasar program Android adalah suatu proses pada aplikasi tidak langsung di atur oleh aplikasi tersebut. Proses tersebut ditentukan oleh sistem melalui kombinasi : informasi aplikasi bagi sistem, seberapa pentingnya aplikasi tersebut bagi *user* dan ketersediaan memori. Contoh kasus *life-cycle bug* adalah *IntentReceiver* yang menciptakan *thread* pada saat menerima *intent* pada *method* onReceiveIntent() dan selesai dari fungsi tersebut. Setelah fungsi tersebut selesai, sistem menganggap bahwa *IntentReceiver* tersebut menjadi tidak aktif sehingga ia memiliki proses yang tidak dibutuhkan (kecuali terdapat komponen aplikasi lain di dalamnya). Oleh karena itu, sistem dapat menghapus proses tersebut kapanpun apabila dibutuhkan. Hal tersebut akan menghapus semua *thread* yang berjalan pada proses tersebut. Solusi masalah ini adalah memakai *Service* dari IntentReceiver, sehingga sistem mengetahui bahwa ada suatu yang aktif dalam proses tersebut.

Sebagai *programmer*, kita harus mengetahui bagaimana cara yang tepat untuk merancang aplikasi. Kesalahan pada perancangan akan membuat suatu aplikasi akan dihentikan oleh sistem pada saat yang tidak diinginkan. Untuk menentukan proses mana yang akan dihapus pada situasi *low memory*, Android memiliki "*importance hierarchy*" berdasarkan komponen dan *state* yang sedang berjalan. Urutan berdasarkan kepentingan tersebut adalah:

1. ***Foreground process,*** Adalah suatu proses yang dibutuhkan untuk berinteraksi dengan user saat ini. Berbagai komponen suatu aplikasi dapat mengakibatkan proses di dalamnya menjadi foreground proses. Suatu proses dikatakan foreground apabila memenuhi kondisi sebagai berikut:
    a. Menjalankan Activity pada *top screen* yang sedang berinteraksi dengan *user* (setelah *method* onResume() dipanggil).
    b. Mempunyai IntentReceiver yang sedang berjalan (method IntentReceiver.onReceiveIntent() sedang dieksekusi).
    c. Memilik service yang fungsi *callback*-nya (Service.onCreate(), Service.onStart() atau Service.onDestroy()) sedang dieksekusi.

    Hanya akan terdapat beberapa proses seperti itu dalam sistem, proses ini hanya akan dihapus apabila sistem tidak memiliki pilihan lain. Contohnya apabila memori tinggal sedikit sehingga proses *foreground* tidak bisa lagi berjalan. Hal ini diperlukan untuk menjaga *user interface* tetap interaktif dengan *user*.
2. ***Visible process,*** Adalah proses yang memegang Activity yang dapat dilihat *user* pada layar tetapi tidak *foreground* (pada saat *method*

onPause() dipanggil). Hal ini dapat terjadi, sebagai contoh *activity foreground* muncul dengan dialog sehingga *activity* sebelumnya terlihat di belakangnya. Proses ini sangat penting dan tidak akan di hapus kecuali untuk menjaga aplikasi *foreground* tetap berjalan.

3. ***Service process,*** Adalah proses yang memegang Service yang dimulai dengan method StartService(). Meskipun proses ini secara langsung tidak dapat dilihat oleh *user*, biasanya proses ini menjalankan perintah dari *user* (seperti memutar lagu, men-*download* data dari jaringan dan lain-lain). Sistem akan tetap membiarkan proses ini berjalan selama proses *foreground* dan *visible* masih bisa berjalan.
4. ***Background process***, Adalah proses yang memegang Activity yang saat ini tidak dapat dilihat oleh *user* (method onStop() sedang dipanggil). Proses ini tidak secara langsung mempengaruhi *user*. Apabila *activity life cycle* telah dipenuhi dengan benar, sistem dapat menghapus proses ini sewaktu-waktu untuk memberikan memori tambahan. Biasanya banyak ditemukan proses semacam ini, sehingga proses tersebut disimpan dalam LRU (*Least Recently Used*) untuk menjaga agar proses yang baru saja dilihat oleh *user* akan dihapus paling akhir pada kondisi *low memory.*
5. ***Empty Process,*** Adalah proses yang tidak memegang suatu komponen yang sedang aktif. Alasan mengapa membiarkan proses semacam ini adalah untuk melakukan *cache* sehingga proses *startup* menjadi lebih cepat apabila aplikasi tersebut akan dijalankan kembali. Sistem seringkali

menghapus proses semacam ini untuk mencapai keseimbangan dari *system resource* antara cached processes dan kernel caches.

2.3.10. UML

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya.[1] UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut.[1] UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.[1] Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP.[1] UML merupakan dasar bagi perangkat (tool) desain berorientasi objek dari IBM.[1]

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.3.10.1. Sejarah UML

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch, metodologi coad, metodologi OOSE, metodologi OMT, metodologi shlaer-mellor, metodologi wirfs-brock, dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8) [3]. Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – *http://www.omg.org*).

2.3.10.2. Diagram UML

UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu:

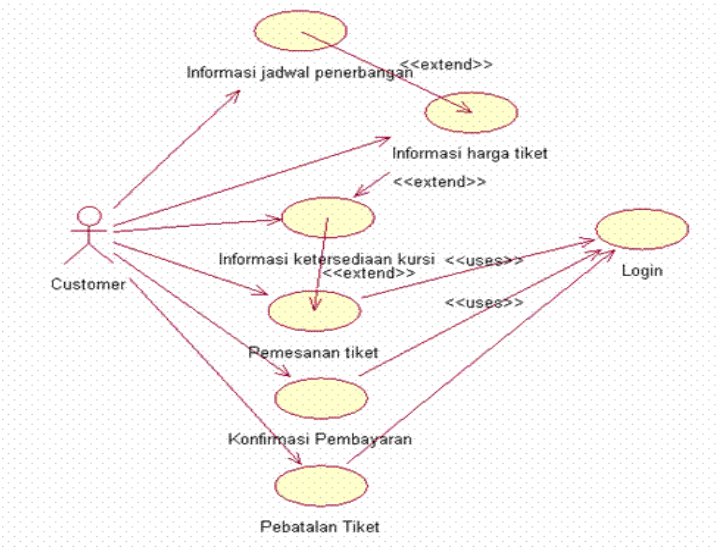
1. Use Case Diagram untuk memodelkan proses bisnis.
2. Conceptual Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. Sequence Diagram untuk memodelkan pengiriman pesan (message) antar objects.

4. Collaboration Diagram untuk memodelkan interaksi antar objects.
5. State Diagram untuk memodelkan perilaku objects di dalam sistem.
6. Activity Diagram untuk memodelkan perilaku Use Cases dan objects di dalam system.
7. Class Diagram untuk memodelkan struktur kelas.
8. Object Diagram untuk memodelkan struktur object.
9. Component Diagram untuk memodelkan komponen object.
10. Deployment Diagram untuk memodelkan distribusi aplikasi.

2.3.10.2.1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/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 *use case* yang di-*include* akan dipanggil 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 fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case*

yang satu merupakan spesialisasi dari yang lain. Penggambaran Use case diagram dapat di lihat dalam gambar II.5 di bawah ini.



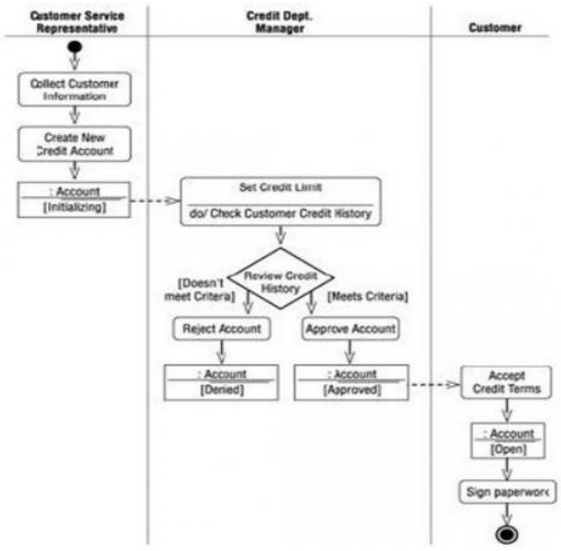
Gambar II.5 Contoh use case diagram

2.3.10.2.2. *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour 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 satu *use case* atau lebih.

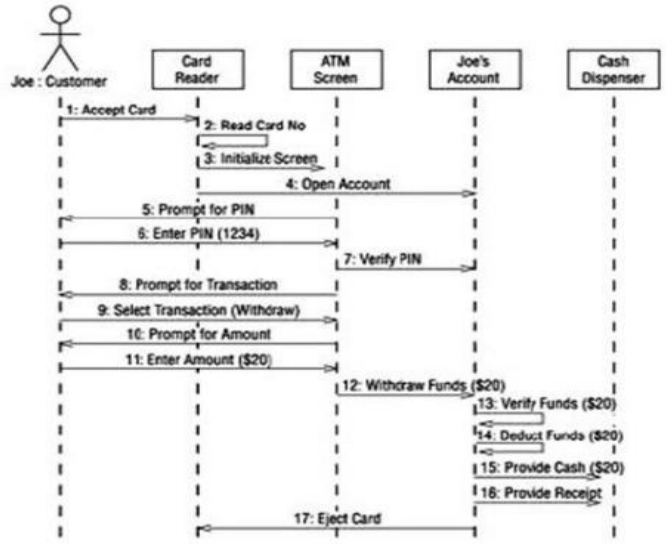
Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu. Contoh Activity diagram dapat di lihat pada gambar II.6 berikut ini.



Gambar II.6 Contoh activity diagram

2.3.10.2.3. *Sequence Diagram*

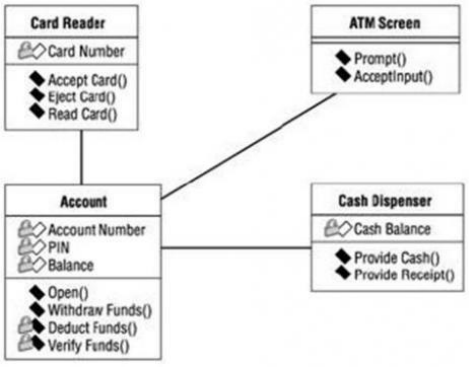
Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Contoh sequence diagram dapat di lihat pada gambar II.7 berikut ini.



Gambar II.7 Contoh Sequence Diagram

2.3.10.2.4. *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Penggambaran class diagram dapat di lihat dalam gambar II.8 dibawah ini.



Gambar II.8 Contoh Class Diagram

2.3.11. WEB

Layanan WWW (*World-Wide Web*) atau biasa juga disebut Web merupakan jenis layanan yang paling popular dikalangan pengguna internet. WWW tidak hanya berfungsi sebagai media untuk mencari informasi, tetapi web sudah banyak digunakan secara komersial oleh hampir semua perusahaan-perusahaan diseluruh dunia untuk mengiklankan usaha mereka. Setiap dokumen yang ditulis menggunakan suatu format standar yang disebut HTML (*HypertText Markup Language*).

2.3.12. PHP

PHP (*HypertText Preprocessor*) adalah jenis *middleware open source*, sehingga dapat digunakan oleh siapa saja dengan cuma-cuma. Selain gratis, PHP memiliki berbagai kelebihan, diantaranya untuk membuat *website* yang dinamis. PHP akan dijalankan pada sisi server sehingga *client* tidak bisa melihat kode aslinya dan dengan PHP, halaman web dapat dibuat lebih interaktif.

2.3.13. MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi *user* serta menggunakan perintah standar SQL (*Structured Query Language*) (Nugroho, 2005). MySQL merupakan sebuah *database server* yang *free*, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya.

MySQL pertama kali dirintis oleh seorang programer database bernama Michael Widenius. Selain MySQL ada beberapa program *database server* lain yang menggunakan standar *query* berupa SQL, antara lain adalah :

1. Oracle
2. PostgreSQL
3. MySQL front
4. MsQL
5. SQL Server 97/..
6. Dll

MySQL adalah sebuah *database server*, dapat juga berperan sebagai *client* sehingga sering disebit *database client/server*, yang pen *source* dengan kemampuan dapat berjalan baik di OS (Operating Sistem) manapun, dengan *Platform* Windows maupun LINUX.

2.3.14. Dreamweaver

Dreamweaver merupakan software aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain web secara visual. Aplikasi ini juga dikenal dengan istilah WYSIWYG (*What You See Is What You Get*) yaitu bahwa anda tidak harus berusaha dengan tag-tag HTML untuk membuat sebuah situs. Selain itu, Dreamweaver juga memberikan keleluasaan kepada pengguna untuk menggunakannya sebagai media penulisan bahasa pemrograman web. Dreamweaver memiliki kemampuan untuk dapat berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, JavaScript, dan yang lainnya juga

memberikan fasilitas maksimal kepada para desainer web yang menyertakan bahasa pemrograman web didalmnya.

2.3.15. Apache

Apache merupakan web server yang saat ini telah digunakan hampir dari 60% oleh server di dunia disebabkan karena sifat *software* ini sangat fleksibel dan dapat digunakan pada berbagai *platform* seperti Linux dan Windows**.**

2.3.16. PhpMyAdmin

PhpMyAdmin merupakan *tools open source* yang dibuat menggunakan program PHP untuk mengakses *database* MySQL via web. Dengan menggunakan program ini kita dapat mengelolah *database* MySQL dengan sangat mudah.

Hampir semua operasi terhadap *database* dan perintah yang ada dapat dijalankan pada program tersebut, termasuk diantaranya menggunakan perintah *Database Definition Language* (DDL), *Database Manipulation Language* (DML) serta *Data Control Language* (DCL).

2.3.17. Java

Java merupakan bahasa pemograman yang menjadi kelanjutan dari bahasa C++. Java dikembangkan oleh Sun Microsystem, Inc. pada tahun 1995 dan telah mengalami revolusi (perkembangan) yang sangat pesat sampai sekarang ini.

Bahasa pemograman ini memiliki beberapa keunggulan dibandingkan dengan bahasa pemograman lainnya yaitu ukuran filenya kecil, akses data cepat, bersifat kompiler-interpreter dan dapat dibaca oleh berbagai system operasi seperti windows ataupun linux.

Java memiliki sifat *platform independent* (terbuka) yang menjadi ciri khas bahasa pemograman tersebut. Hal ini menyebabkan bahasa pemograman ini dapat di baca seluruh komputer tanpa harus mengkonversinya sehingga ketika user sedang berkomunikasi dengan menggunakan fasilitas komputer, user tidak perlu mengkhawatirkan perbedaan software maupun hardware dari masing-masing komputer. Java juga bersifat *neutral architecture,* karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras.

Sebagai sebuah *platform,* Java terdiri dari dua bagian utama, yaitu :

1. *Java Virtual Machine*(JVM)
2. *Java Application Programming Interface*(Java API)

Sun membagi arsitektur Java menjadi tiga bagian, yaitu :

1. J2EE(*Java 2 Enterprise Edition*)
2. J2SE(*Java 2 Second Edition*)
3. J2ME(*Java 2 Micro Edition*)

2.3.18. Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse: Multi-platform:

Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.