

BAB II

LANDASAN TEORI

II.1. Perancangan

Perancangan adalah suatu proses yang bertujuan untuk menganalisis, menilai memperbaiki dan menyusun suatu sistem, baik sistem fisik maupun non fisik yang optimum untuk waktu yang akan datang dengan memanfaatkan informasi yang ada. Perancangan suatu alat termasuk dalam metode teknik, dengan demikian langkah-langkah pembuatan perancangan akan mengikuti metode teknik. *Merris Asimov* menerangkan bahwa perancangan teknik adalah suatu aktivitas dengan maksud tertentu menuju kearah tujuan dari pemenuhan kebutuhan manusia, terutama yang dapat diterima oleh faktor teknologi peradaban kita. Dari definisi tersebut terdapat tiga hal yang harus diperhatikan dalam perancangan yaitu aktifitas dengan maksud tertentu, sasaran pada pemenuhan kebutuhan manusia dan berdasarkan pada pertimbangan teknologi [6].

II.2. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama

perangkat lunak aplikasi adalah pengolah kata, lembar kerja dan pemutar media [6].

II.3. Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Informasi yang berkualitas memiliki 3 kriteria, yaitu : [7].

1. Akurat (*accurate*)

Informasi harus bebas dari kesalahan, tidak bias ataupun menyesatkan, akurat juga berarti bahwa informasi itu harus dapat dengan jelas mencerminkan maksudnya.

2. Tepat pada waktunya (*timelines*)

Informasi yang datang pada penerima tidak boleh terlambat. Didalam pengambilan keputusan, informasi yang sudah usang tidak lagi bernilai.

3. Relevan (*relevance*)

Informasi yang disampaikan harus mempunyai kegiatan dengan masalah yang dibahas dengan informasi. Informasi harus bermanfaat bagi pemakainya.

II.4. Transaksi Penjualan

Transaksi dibagi menjadi dua kelas yaitu transaksi keuangan dan transaksi non-keuangan. Transaksi keuangan adalah sebuah peristiwa ekonomi yang mempengaruhi aktiva dan ekuitas suatu organisasi, direfleksikan dalam akun –

akunnya dan diukur dalam satuan moneter. Transaksi non-keuangan termasuk dalam sebuah peristiwa yang diproses sistem informasi organisasi yang tidak memenuhi defenisi sempit dari transaksi keuangan. Jadi dalam sistem informasi akuntansi tidak hanya mengolah data keuangan saja, data non-keuangan juga diikutsertakan karena dapat pengambilan keputusan tidak hanya informasi keuangan saja yang diperlukan, informasi non-keuangan tentang suatu kondisi dan keadaan juga dapat dipergunakan sebagai pertimbangan dalam pengambilan keputusan [7].

II.5. Laporan Keuangan

Laporan keuangan merupakan alat yang sangat penting untuk memperoleh informasi sehubungan dengan posisi keuangan dan hasil-hasil yang telah dicapai oleh perusahaan yang bersangkutan [7].

Laporan keuangan merupakan hasil akhir dari proses akuntansi yang meliputi dua laporan utama yakni, Neraca dan Laporan Laba Rugi [7].

Berdasarkan pengertian-pengertian laporan keuangan tersebut di atas, maka yang dimaksud dengan laporan keuangan perusahaan adalah bentuk pertanggung jawaban keuangan dari perusahaan, pada umumnya terdiri dari Neraca, Laporan Laba Rugi dan Laporan Perubahan Modal atau Laporan Laba yang ditahan [7].

II.6. Toko *Eiger Adventure*

PT. Eigerindo Multi Produk Industri atau dikenal sebagai Eiger adalah perusahaan yang paling terkenal di Indonesia khusus di bidang manufaktur dan ritel peralatan petualangan. Perusahaan ini didirikan pada tahun 1993 oleh Bapak Ronny Lukito di Bandung. Nama perusahaan, Eiger, terinspirasi oleh *mount Eiger*, sebuah gunung terkenal di *Alpen Bernese, Swiss*, naik ke ketinggian 3.970 meter di atas permukaan laut. Dimulai dengan fasilitas yang sangat terbatas, kami meluncurkan keluar untuk memproduksi tas dengan hanya dua mesin sederhana di Jalan Cihampelas No. 22 Bandung. Dan di tempat yang sama, kami membuka sebuah toko kecil untuk menjual tas kami.

PT. Eigerindo Multi Produk Industri. Produk kami ini dalam tiga merek utama, Eiger dengan posisi gaya hidup petualangan, *Bodypack* dengan posisi gaya hidup dan *Nordwand* dengan posisi hidup *outdoor*. Merek kami banyak diakui sebagai merek lokal terkemuka di Indonesia. Pada tahun 2009, kami terdaftar sebagai salah satu dari 250 Merek Top Indonesia Asli oleh majalah bisnis SWA dibedakan. Ini menunjukkan kerja keras kami, tekad dan hari demi hari komitmen untuk mencapai keunggulan kualitas dan reputasi yang luar biasa.

PT. Eigerindo Multi Produk dulu sebagai sebuah toko kecil dan produksi dengan hanya dua mesin jahit di Jalan Cihampelas No. 22 enam belas tahun yang lalu, telah berkembang menjadi pemimpin pasar dalam industri tas dan petualangan di Indonesia. Sampai sekarang, kami telah memiliki 6 Toko Petualangan Eiger, 18 ruang pameran, dan 88 *counter* yang tersebar di seluruh Indonesia [8].



Gambar II.1. Eiger Store Adventure Jl. Mustafa. Medan

II.7. Komputer

Komputer dalam berbagai skala dan ukuran telah banyak mempercepat aktivitas berbasis sehingga perannya dalam kegiatan bisnis begitu sulit dipisahkan. Peranan komputer dari yang hanya sekedar alat untuk menghitung, membuat lembar kerja keuangan, sistem kantor otomatis hingga dalam wujud memberikan dukungan pengambilan keputusan. Tuntutan efektifitas, efisiensi dan produktivitas kerja menyebabkan peranan komputer semakin vital [9].

II.8. Pengenalan Jaringan Komputer

Jaringan komputer adalah sistem yang terdiri dari komputer-komputer, serta piranti-piranti yang saling terhubung sebagai satu kesatuan. Dengan dihubungkan piranti-piranti tersebut, alhasil dapat saling berbagi sumber daya antar satu piranti dengan piranti lainnya [9].

II.8.1. Jenis Jaringan Komputer

Berdasarkan jangkauan area atau lokasi, jaringan dibedakan menjadi 5 jenis yaitu :

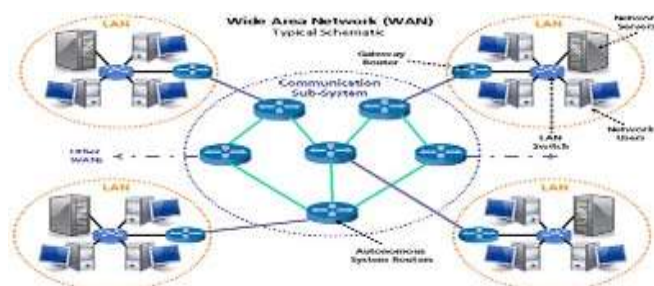
1. *Local Area Network* (LAN) merupakan jaringan yang menghubungkan sejumlah komputer yang ada dalam suatu lokasi dengan area yang terbatas seperti ruang atau gedung. LAN dapat menggunakan media komunikasi seperti kabel dan *wireless*.



Gambar II.2. Tampilan Jaringan LAN

(Sumber : wahana komputer ; 2012)

2. *Wide Area Network* (WAN) merupakan jaringan antara LAN satu dengan LAN lain yang dipisahkan oleh lokasi yang cukup jauh. Contoh penggunaan WAN adalah hubungan antara kantor pusat dengan kantor cabang yang ada di daerah-daerah.



Gambar II.3. Tampilan Jaringan WAN

(Sumber : wahana komputer ; 2012)

3. *Metropolitan Area Network* (MAN) merupakan jaringan yang lebih besar dari jaringan LAN tetapi lebih kecil dari jaringan WAN. Jaringan MAN dan jaringan WAN sama-sama menghubungkan beberapa LAN yang membedakan hanya lingkup areanya yang berbeda.



Gambar II.4. Tampilan Jaringan MAN

(Sumber : wahana komputer ; 2012)

4. *Wireless LAN* (WLAN) adalah singkatan dari *Wireless Local Area Network* yaitu suatu jenis jaringan komputer yang menggunakan gelombang radio sebagai alat atau media transmisi data. Informasi atau data ditransfer dari satu komputer ke komputer yang lainnya menggunakan gelombang radio. WLAN juga sering disebut dengan Jaringan Nirkabel atau jaringan *wireless*.



Gambar II.5. Tampilan Jaringan WLAN

(Sumber : wahana komputer ; 2012)

5. *Personal Area Network* (PAN) adalah hubungan antara dua atau lebih sistem komputer yang berjarak tidak terlalu jauh. Biasanya Jenis jaringan yang satu ini hanya berjarak 4 sampai 6 meter saja. Jenis jaringan ini sangat sering kita gunakan. contohnya menghubungkan *hp* dengan komputer [9].



Gambar II.6. Tampilan Jaringan PAN

(Sumber : wahana komputer ; 2012)

II.8.2. Jaringan *Client Server*

Jaringan *client server* menghubungkan komputer *server* dengan komputer *client/workstation*. Komputer *Server* adalah komputer yang menyediakan fasilitas bagi komputer-komputer *client/workstation* yang terhubung dalam jaringan. Sedangkan komputer adalah komputer yang menggunakan fasilitas yang disediakan oleh komputer *server*. Komputer *server* pada sebuah jaringan tipe *client server* disebut dengan *dedicated server*, karena komputer yang digunakan hanya sebagai penyedia fasilitas untuk komputer *client/workstation*. Komputer *server* tidak dapat berperan sebagai komputer *client/workstation*. Keunggulan tipe jaringan *client server* adalah [9] :

1. Terdapat Administrator jaringan yang mengelola sistem keamanan, sehingga sistem keamanan dan administrasi jaringan akan lebih terkontrol.

2. Komputer *server* difungsikan sebagai pusat data, komputer *client* dapat mengakses data yang ada dari komputer *client* manapun. Apabila terdapat komputer *client* yang rusak, pengguna masih dapat mengakses data dari komputer *client* yang lain.
3. Pengaksesan data lebih tinggi karena penyediaan dan pengelolaan fasilitas jaringan dilakukan oleh komputer *server*. Dan komputer *server* tidak terbebani dengan tugas lain sebagai *workstation*.
4. Pada tipe jaringan *client server*, sistem *backup* data lebih baik, karena *backup* data dapat dilakukan terpusat di komputer *server*. Apabila data pada komputer *client/workstation* mengalami masalah atau kerusakan masih tersedia *backup* pada komputer *server*.

Kelemahan tipe jaringan *client server* adalah :

1. Biaya mahal, karena membutuhkan komputer yang memiliki kemampuan tinggi yang difungsikan sebagai komputer *server*.
2. Kelancaran jaringan tergantung pada komputer *server*. Bila komputer *server* mengalami gangguan maka jaringan akan terganggu [9].

II.9. IP Address

IP (*Internet Protocol*) address merupakan alamat yang diberikan kepada komputer-komputer yang terhubung dalam suatu jaringan. *IP Address* terdiri dari dua bagian, yaitu : *Network ID* dan *Host ID*. *Network ID* menentukan alamat dalam jaringan (*network address*), sedangkan *Host ID* menentukan alamat dari peralatan jaringan yang sifatnya unik untuk membedakan antara satu mesin

dengan mesin yang lain. Ibarat sebuah alamat rumah, *network ID* seperti alamat rumah dan *Host ID* seperti nomor rumah [9].

IP Address berdasarkan perkembangan dibagi menjadi dua jenis :

1. IPv4 (*internet protocol* versi 4), merupakan *IP Address* yang terdiri dari 32 bit yang dibagi menjadi 4 segmen berukuran 8 *bit*.
2. IPv6 (*internet protocol* versi 6), merupakan *IP Address* yang terdiri dari 128 bit yang digunakan untuk mengatasi permintaan *IP Address* yang semakin meningkat.

IP Address terdiri dari 32 *bit* angka *biner* yang dituliskan dalam bentuk empat kelompok dan masing-masing kelompok terdiri dari delapan (*oktat*) *bit* yang dipisahkan oleh tanda titik.

Contoh : 11000000.101010000.00000000.01100100.

IP Address dapat juga ditulis dalam bentuk angka decimal dalam empat kelompok, dari angka 0-255. *IP Address* dibedakan menjadi 3 kelas, yaitu kelas A, kelas B dan kelas C. Tujuan membedakan kelas IP adalah untuk menentukan jumlah komputer yang bisa terhubung dalam sebuah jaringan [9].

1. IP kelas A terdiri dari 8 *bit* pertama digunakan untuk *network ID* dan 24 *bit* berikutnya merupakan *host id*. IP kelas A memiliki 126 *network*, yakni dari nomor 1.xxx.xxx.xxx sampai dengan 126.xxx.xxx.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP kelas A :

ONNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

- a. *Bit* pertama nilainya 0
- b. *Network ID* adalah 8 *bit* dan *Host ID* adalah 24 *bit*
- c. *Bit* pertama diisi antara 0 sampai dengan 127
- d. Range IP antara 1.xxx.xxx.xxx – 126.xxx.xxx.xxx
- e. Jumlah *Network* adalah 126 (untuk 0 dan 127 dicadangkan)
- f. Jumlah *Host* adalah 16.777.214
- g. Sebagai contoh *IP address* 10.11.22.33, maka *Network ID* adalah 10 dan *Host ID* adalah 11.22.33.

Jadi IP di atas mempunyai *host* dengan nomor 11.22.33 pada jaringan 10

2. IP kelas B terdiri dari 16 *bit* pertama digunakan untuk *Network ID* dan 16 *bit* berikutnya merupakan *Host ID*. IP kelas B memiliki 16.384 *Network*, yakni dari nomor 128.0.xxx.xxx sampai dengan 191.255.xxx.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP Kelas B

10NNNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

- a. *Bit* pertama nilainya 10
- b. *Network ID* adalah 16 *bit* dan *Host ID* adalah 16 *bit*
- c. *Bit* pertama diisi antara 128 sampai dengan 191
- d. Range IP antara 128.0.xxx.xxx – 191.255.xxx.xxx
- e. Jumlah *Network* adalah 16.384 (64 x 256)
- f. Jumlah *Host* adalah 65.532

- g. Sebagai contoh *IP address* 130.1.2.3, maka *Network ID* adalah 130.1 dan *Host ID* adalah 2,3

Jadi IP di atas mempunyai *host* dengan nomor 2,3 pada jaringan 130.1

3. IP kelas C terdiri dari 24 *bit* pertama digunakan untuk *Network ID*, dan 8 *bit* berikutnya merupakan *Host ID*. IP kelas C memiliki 2.097.152 *Network*, yakni dari nomor 192.0.0.xxx sampai dengan 223.255.255.xxx (xxx merupakan variabel yang nilainya dari 0 sampai dengan 255).

Format IP kelas C

110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH

(N = *Network ID*, H = *Host ID*)

- a. *Bit* pertama nilainya 110
- b. *Network ID* adalah 24 *bit* dan *Host ID* adalah 8 *bit*
- c. *Bit* pertama diisi antara 192 sampai dengan 223
- d. Range IP antara 192.0.0.xxx – 233.255.255.xxx
- e. Jumlah *Network* adalah 2.097.152 (32 x 256 x 256)
- f. Jumlah *Host* adalah 254

Sebagai contoh *IP address* 192.168.0.100, maka *Network ID* adalah 192.168.0 dan *Host ID* adalah 100. Jadi IP di atas mempunyai *host* dengan nomor 100 pada jaringan 192.168.0 [9].

1. *IP Address Private*, merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan lokal (LAN). *IP Address Private* digunakan untuk komunikasi pada jaringan yang tidak terhubung langsung dengan Internet. *IP*

Address Private hanya dapat dipakai untuk komunikasi pada jaringan *intranet* dan tidak dapat digunakan pada jaringan *internet*.

- a. *IP Address Private* kelas A, memiliki range *IP Address* antara 10.0.0.1 – 10.255.255.254
 - b. *IP Address Private* kelas B, memiliki range *IP Address* antara 172.16.0.1 – 172.31.255.254
 - c. *IP Address Private* kelas C, memiliki range *IP Address* antara 192.168.0.1 – 192.168.255.254
2. *IP Address Public*, merupakan alamat-alamat IP yang disediakan untuk digunakan pada jaringan *Internet*.
3. *Subnet Mask*, merupakan angka biner 32 bit yang digunakan untuk membedakan *Network ID* dan *Host ID*. *Subnet Mask* menunjukkan letak suatu *host*, apakah berada di jaringan lokal atau berada di jaringan luar. Pada *subnet mask*, bit yang berhubungan dengan *Network ID* diset 1 sedangkan bit yang berhubungan dengan *Host ID* diset 0
- a. *IP Address* kelas A menggunakan *subnet mask*
 $11111111.00000000.00000000.00000000 = 255.0.0.0$
 - b. *IP Address* kelas B menggunakan *subnet mask*
 $11111111.11111111.00000000.00000000 = 255.255.0.0$
 - c. *IP Address* kelas C menggunakan *subnet mask*
 $11111111.11111111.11111111.00000000 = 255.255.255.0$ [9].

II.10. *Smartphone*

Smartphone adalah telepon pintar yang memiliki kemampuan seperti komputer. *Smartphone* diklasifikasikan sebagai *high end mobile phone* yang dilengkapi dengan kemampuan *mobile computing*. Dengan kemampuan *mobile computing* tersebut, *Smartphone* memiliki kemampuan yang tak bisa dibandingkan dengan ponsel biasa. *Smartphone* yang pertama kali muncul merupakan kombinasi dari fungsi suatu personal digital assistant (PDA) dengan telepon genggam ataupun telepon dengan kamera. Seiring dengan perkembangannya, kini *Smartphone* juga mempunyai fungsi sebagai *media player portable*, *low end digital compact camera*, *pocket video camera* dan GPS. *Smartphone modern* juga dilengkapi dengan layar *touchscreen* resolusi tinggi, *browser* yang mampu menampilkan *full web* seperti pada PC, serta akses data *WiFi* dan *internet broadband* [10].



Gambar II.7. Smartphone

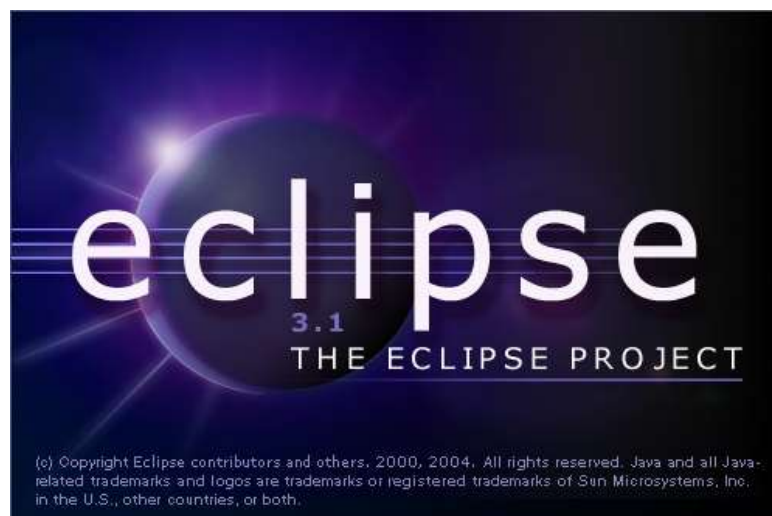
II.11. Android

Android adalah sebuah sistem operasi pada *handphone* yang bersifat terbuka dan berbasis pada sistem operasi *Linux*. Android bisa digunakan oleh setiap orang yang ingin menggunakannya pada perangkat mereka. Android

menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak [10].

II.12. Eclipse

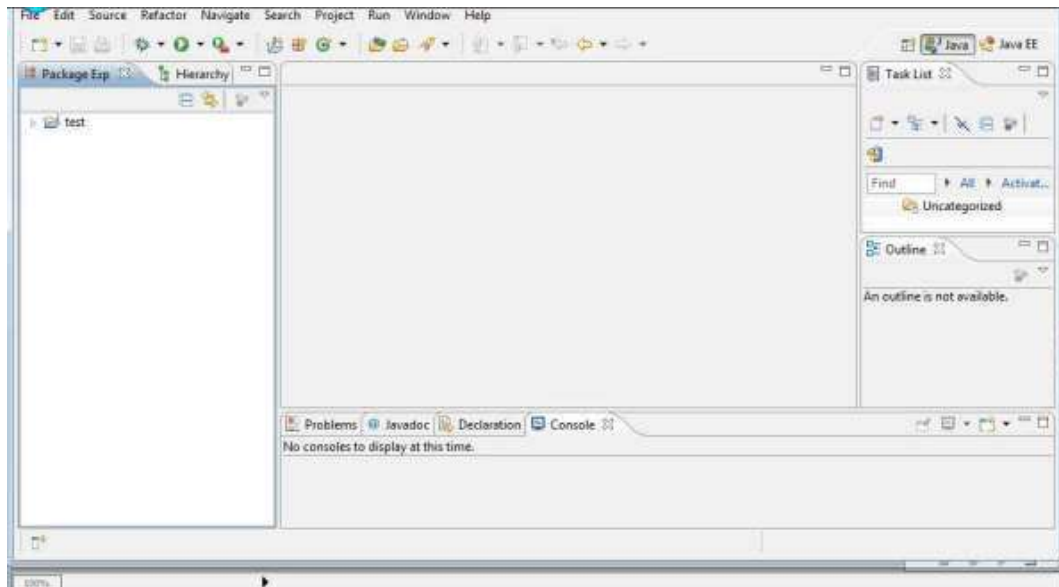
Eclipse merupakan komunitas *open source* yang bertujuan menghasilkan *platform* pemrograman terbuka. Eclipse terdiri dari *framework* yang dapat dikembangkan lebih lanjut, peralatan bantu untuk membuat dan mengatur *software* sejak awal hingga diluncurkan. *Platform Eclipse* didukung oleh ekosistem besar yang terdiri dari vendor teknologi, *start-up inovatif*, *universitas*, riset institusi serta individu. Banyak orang mengenal Eclipse sebagai IDE (*integrated development environment*) untuk bahasa *Java*, tapi Eclipse lebih dari sekedar IDE untuk *Java*. Secara umum Eclipse digunakan untuk membangun *software inovatif* berstandar industri dan alat bantu beserta *framework*-nya membantu pekerjaan menjadi lebih mudah [10].



Gambar II.8. Eclipse

II.12.1. Area Kerja Eclipse

Saat pertama kali menjalankan program Eclipse, maka kita akan mendapati tampilan halaman pembuka Eclipse seperti yang terlihat pada gambar berikut ini.



Gambar II.9. Area Kerja Eclipse

II.13. JDK (Java Development Kit)

JDK adalah *Sun Microsystem* produk ditujukan untuk pengembangan *Java*. Sejak diperkenalkannya *Java*, telah jauh SDK *Java* yang paling banyak digunakan. Pada Tanggal 17 November 2006, *Sun* mengumumkan bahwa akan dirilis dibawah GNU *General Public License* (GPL), sehingga membuat perangkat lunak bebas [10].

II.14. Java

Java merupakan sebuah *platform* sekaligus bahasa pemrograman tingkat tinggi yang mempunyai kriteria sederhana, berorientasi objek, terdistribusi,

dinamis, aman dan lainnya. Bahasa ini dikembangkan dengan model yang mirip seperti bahasa *C++* dan *smalltalk* namun lebih mudah dipakai dan juga memiliki *platform* independen yang dapat dijalankan pada sistem operasi apapun [10].

II.15. Android SDK (System Development Kit)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman *Java*. Android SDK menyediakan *tools* dan API untuk pengembangan *platform* aplikasi android dengan menggunakan bahasa pemrograman *java* [10].

II.16. PHP

PHP singkatan dari PHP (*Hypertext Preprocessor*) yaitu bahasa pemrograman web *server side* yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada *server (server side HTML embended scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamsi. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua script PHP dieksekusi pada *server* di mana script tersebut dijalankan [11].

II.17. MySQL

MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang multithread, multi user yang bersifat gratis yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan Swedia, yaitu MySQL AB. MySQL AB memegang hak cipta kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larson, dan Michael Monty Widenius. Seperti yang telah disebutkan sebelumnya, MySQL bersifat gratis atau *open source* sehingga bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung atau support secara gratis. Pemrograman PHP juga sangat mendukung / *support* dengan *database* MySQL sehingga apabila mempelajarinya dengan sungguh-sungguh dapat mengaplikasikan PHP dan MySQL dalam membuat aplikasi *website* maupun dalam membuat *website* [11].

II.18. XAMPP

XAMPP adalah sebuah *software web apache* yang di dalamnya sudah tersedia *database server* MySQL dan *support php programming*. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *linux* dan *windows*. Keuntungan lainnya adalah hanya menginstal sekali sudah

tersedia *apache web server*, *MySQL database server*, *PHP support* (PHP 4 dan PHP 5) serta beberapa modul lainnya. Hanya bedanya versi untuk *windows* sudah dalam bentuk instalasi grafis dan yang *linux* dalam bentuk *file* terkompresi *tar*. kelebihan lain yang berbeda dari versi *windows* adalah memiliki fitur untuk mengaktifkan sebuah *server* secara grafis, sedangkan *linux* masih berupa perintah di dalam *console* [11].

II.19. UML (*Unified Modelling Language*)

UML(*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan baik [6].

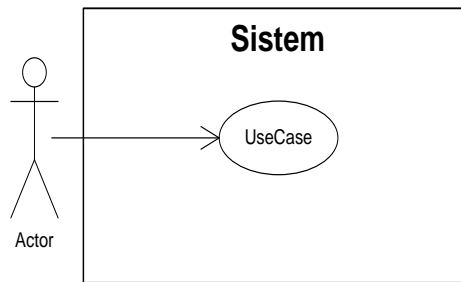
UML merupakan kesatuan bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *object Oriented Engineering* (OOSE). Metode *Booch* dari *Grady Booch* sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan *design* ke dalam empat tahapan interaktif, yaitu: identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan objek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode *Booch* adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan *OMT* yang dikembangkan oleh *Rumbaugh* didasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Tahapan

utama dalam metodologi ini adalah analisis, desain sistem, desain objek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih memberi penekanan dan *use case*. OOSE memiliki tiga tahapan yaitu membuat model *requirement* dan analisis, desain dan implementasi dan model pengujian (*test Model*). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak [6].

II.19.1. Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna.

Notasi *use case* menunjukkan 3 aspek dari sistem yaitu *actor use case* dan *system/sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Ilustrasi *actor*, *usecase* dan *system* ditunjukkan pada gambar di bawah ini.



Gambar II.10. Usecase Diagram

(Sumber : Munawar ; 2012 : 64)

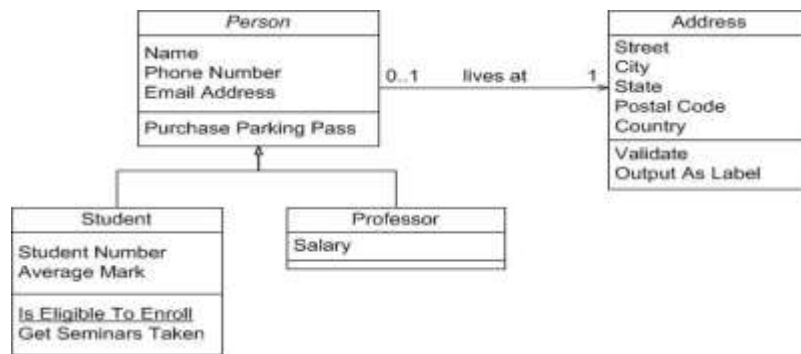
II.19.2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda/fungsi*). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* serta objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain. *Class* memiliki tiga area pokok : nama kelas, atribut dan metode.

Atribut dan metode dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan.
3. *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Contoh diagram *class* dapat dilihat pada gambar berikut ini.



Gambar II.11. Class Diagram


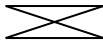
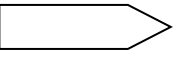
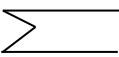

(Sumber : Munawar ; 2012 : 220)

II.19.3. Activity Diagram

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. Adapun simbol *activity diagram* dapat dilihat pada table berikut ini.

Tabel II.1. Simbol Activity Diagram

Notasi	Keterangan
●	Titik Awal
◎	Titik Akhir
▭	<i>Activity</i>
◇	Pilihan untuk pengambilan keputusan
▬	<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu

	<i>Rake</i> menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir (<i>Flow Final</i>)

(Sumber : Munawar ; 2012 : 110)

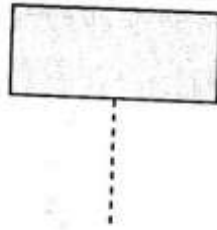
II.19.4. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan di antara objek-objek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

1. Objek /*participant*

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* dihubungkan dengan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. Bentuk *participant* dapat dilihat pada gambar berikut ini.



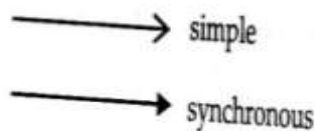
Gambar II.12. Bentuk *Participant*

(Sumber : Munawar ; 2012 : 88)

2. *Message*

Sebuah *message* bergerak dari satu *participant* ke *participant* yang lain dan dari satu *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri.

Sebuah *message* bisa jadi *simple*, *synchronous* atau *asynchronous*. *Message* yang *simple* adalah sebuah perpindahan (transfer), contoh dari satu *participant* ke *participant* yang lainnya. Jika sebuah *participant* mengirimkan sebuah *message* tersebut akan ditunggu sebelum diproses dengan urusannya. Namun jika *message asynchronous* yang dikirimkan, maka jawabannya atas *message* tersebut tidak perlu ditunggu. Simbol *message* pada *sequence diagram* dapat dilihat pada gambar berikut ini.



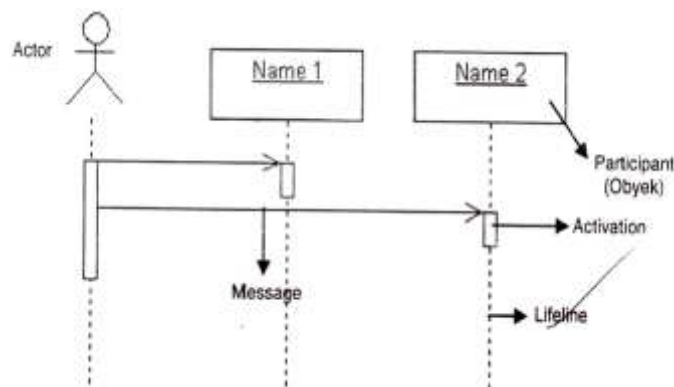
Gambar II.13. Bentuk *Message*

(Sumber : Munawar ; 2012 : 88)

3. *Time*

Time adalah diagram yang mewakili waktu pada arah vertikal. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat ke bawah.

Terdapat dua dimensi pada *sequence diagram* yaitu dimensi dari kiri ke kanan menunjukkan tata letak *participant* dan dimensi dari atas ke bawah menunjukkan lintasan waktu. Simbol-simbol yang ada pada *sequence diagram* ditunjukkan pada gambar berikut ini.



Gambar II.14. Bentuk *Time*

(Sumber : Munawar ; 2005 : 89)

II.19.5. ERD (*Entity Relationship Diagram*)

Entity relationship diagram adalah alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas – entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basis data yang baik sehingga data dapat disimpan dan diambil secara efisien. Elemen – elemen diagram hubungan entitas yaitu :

1. Entitas (*Entity*)

Entitas adalah sesuatu yang nyata atau *abstrak* dimana kita akan menyimpan data.

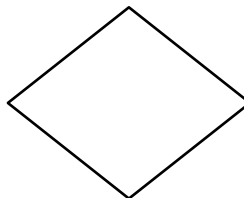


Gambar II.15. Simbol Entitas

(Sumber : Janner Simarmata. Dkk : 2013)

2. Relasi (*Relationship*)

Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya mahasiswa bisa mengambil banyak mata kuliah.



Gambar II.16. Simbol Relasi

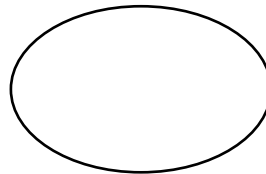
(Sumber : Janner Simarmata. Dkk : 2013)

3. Atribut (*Attribute*)

Atribut adalah ciri umum semua semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah *property*, elemen data dan *field*. Misalnya, nama, alamat, nomor pegawai, dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang mengidentifikasi satu

dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal.

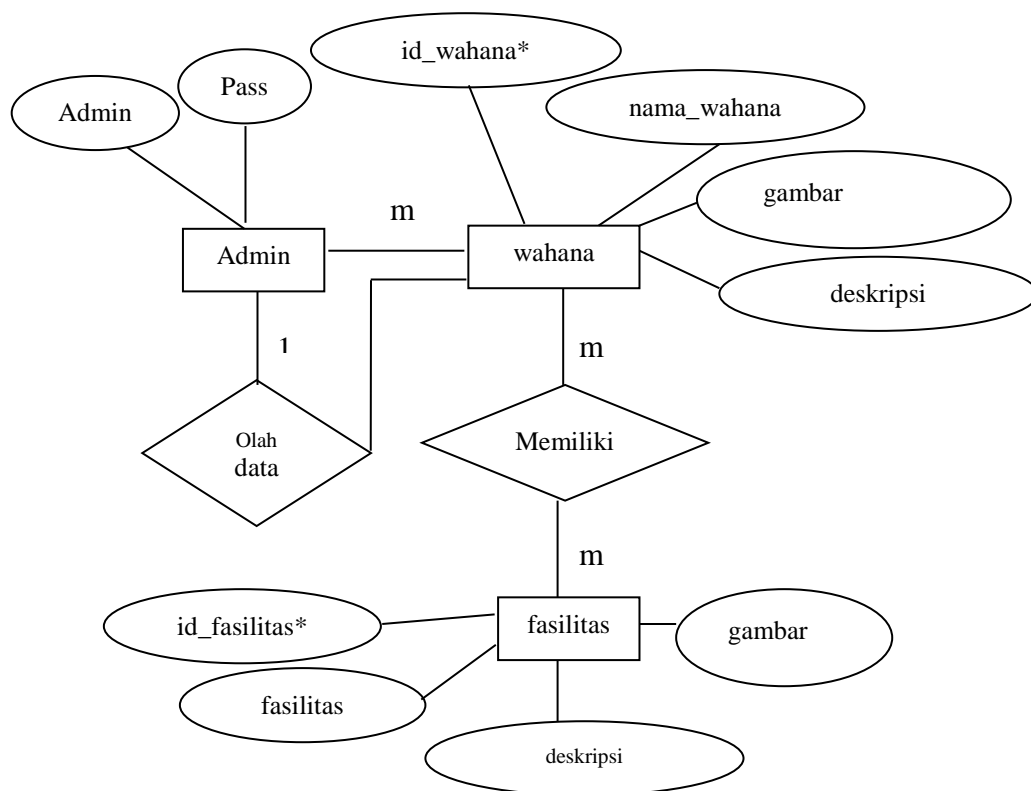
Misalnya, nomor pegawai adalah kunci utama untuk pegawai.



Gambar II.17. Simbol Atribut

(Sumber : Janner Simarmata. Dkk : 2013)

Berikut contoh ERD.



Gambar II.18. Contoh ERD

(Sumber : Janner Simarmata. Dkk : 2013)

II.20. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional [6].

Normalisasi adalah bagian perancangan basis data. Tanpa normalisasi, sistem basis data menjadi tidak akurat, lambat, tidak *efisien*, serta tidak memberikan data yang diharapkan. Pada waktu menormalisasikan basis data, ada empat tujuan yang harus dicapai, yaitu:

1. Mengatur data dalam kelompok – kelompok sehingga masing – masing kelompok hanya mengenai bagian kecil sistem.
2. Meminimalkan jumlah data berulang dalam basis data.
3. Membuat basis data yang datanya diakses dan dimanipulasi secara cepat dan *efisien* tanpa melupakan integritas data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada satu tempat.

II.20.1. Bentuk – Bentuk Normalisasi

1. Bentuk normal pertama (1NF)

Tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi di antara setiap baris pada suatu tabel, dan setiap *atribut* harus mempunyai nilai data yang *atomic* (bersifat *atomic value*).

2. Bentuk normal kedua (2NF)

Normal kedua didasari atas konsep *full functional dependency* (ketergantungan fungsional sepenuhnya) yang dapat didefinisikan sebagai berikut.

Jika A dan B adalah *atribut-atribut* dari suatu relasi, B dikatakan *full functional dependency* (miliki ketergantungan fungsional sepenuhnya) terhadap A, jika B adalah tergantung fungsional terhadap A, tetapi tidak secara tepat memiliki ketergantungan fungsional dari *subset* (himpunan bagian) dari A.

3. Bentuk normal ketiga (3NF)

Jika kita hanya mengupdate satu baris saja, sementara baris yang lainnya tidak, maka data di dalam *database* tersebut akan *inkonsisten*/tidak teratur. Anomali *update* ini disebabkan oleh suatu ketergantungan transitif (*transitive dependency*). Kita harus menghilangkan ketergantungan tersebut dengan melakukan normalisasi ketiga (3-NF).

4. Bentuk normal *boyce-code* (BCNF)

Suatu relasi dalam basis data harus dirancang sedemikian rupa sehingga mereka memiliki ketergantungan sebagian (*partial dependency*), maupun ketergantungan transitif [14].

II.21. Kamus Data

Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data biasanya berisi:

1. Nama - nama dari data
2. Digunakan pada – merupakan proses-proses yang terkait data
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data [6].