

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu (Jogianto;2005:2).

II.1.1. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*) (Jogianto;2011:3).

II.1.2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*) (Jogianto;2005:6)

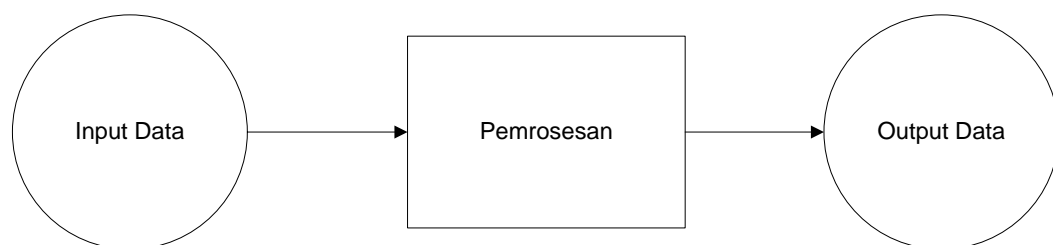
II.2. Informasi

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. (Hanif Al Fatta; 2007: 9)

II.3. Sistem Informasi

Sistem Informasi didefinisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan, dan menyajikan sinergi organisasi pada proses. (Hanif Al Fatta; 2007: 9)

Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output - IPO*) dapat dilihat pada gambar berikut :



Gambar II.1. *Input-Proses-Output*
Sumber : Hanif Al Fatta (2007: 9)

II.3.1. Tipe-Tipe Sistem Informasi

Sistem informasi dibedakan menjadi beberapa tipe aplikasi, yaitu :

a. *Transaction Processing System (TPS)*

Transaction Processing System (TPS) atau Sistem Pemrosesan Transaksi adalah sistem informasi terkomputerisasi yang dikembangkan untuk memproses sejumlah besar data untuk transaksi bisnis rutin.

b. *Management Information System (MIS)*

Management Information System (MIS) atau Sistem Informasi Manajemen (SIM) adalah sebuah sistem informasi pada level manajemen yang berfungsi untuk membantu perencanaan, pengendalian, dan pengambilan keputusan dengan menyediakan resume rutin dan laporan-laporan tertentu. SIM mengambil data mentah dari TPS dan mengubahnya menjadi kumpulan data yang lebih berarti yang dibutuhkan manajer untuk menjalankan tanggung jawabnya. Untuk mengembangkan suatu SIM, diperlukan pemahaman yang baik tentang informasi apa saja yang dibutuhkan manajer dan bagaimana mereka menggunakan informasi tersebut.

c. *Decision Support System (DSS)*

Decision Support System (DSS) atau Sistem Pendukung Keputusan (SPK) merupakan sistem informasi pada level manajemen dari suatu organisasi yang mengombinasikan data dan model analisis canggih atau peralatan data analisis untuk mendukung pengambilan yang semi terstruktur dan tidak terstruktur. DSS dirancang untuk membantu pengambilan keputusan organisasional.

d. *Expert System (ES)*

Expert System (ES) merupakan representasi pengetahuan yang menggambarkan cara seorang ahli dalam mendekati suatu masalah. *ES* lebih berpusat pada bagaimana mengodekan dan memanipulasi pengetahuan dari informasi (misalnya aturan *if..then*).

II.4. Akuntansi

Akuntansi adalah suatu aktivitas jasa, fungsinya menyediakan informasi kuantitatif, terutama yang bersifat keuangan, mengenai satuan-satuan ekonomi, dan dimaksudkan untuk dipergunakan dalam pengambilan keputusan ekonomi yaitu dalam menetapkan pilihan yang tepat diantara beberapa alternatif tindakan. Tujuan umum akuntansi keuangan dan laporan keuangan adalah menyediakan informasi keuangan yang handal mengenai sumber daya dan kewajiban-kewajiban ekonomi dari suatu badan usaha (Teguh Wahyono;2004:9).

Dari pengertian tersebut dapat di tinjau dua sisi kegiatannya. Dari sisi ini akuntansi berarti suatu seni mencatat, mengklarifikasi dan meringkas data keuangan dengan cara yang berarti. Yang kedua ditinjau dari segi informasi yang dihasilkan. Disini akuntansi didefinisikan sebagai suatu sistem keuangan yang memberikan informasi penting yang dibutuhkan suatu organisasi agar dapat beroperasi secara efisien, mengetahui hasil kerja selama ini dan merencanakan kegiatan kedepan agar lebih baik (Teguh Wahyono;2004:10).

II.5. Sistem Informasi Akuntansi

Sistem Informasi Akuntansi adalah suatu komponen organisasi yang mengumpulkan, mengolah, menganalisis, dan mengkomunikasikan informasi

finansial dan pengambilan keputusan yang relevan kepada pihak diluar dan di dalam perusahaan (Teguh Wahyono;2004:13).

Pada umumnya perusahaan dagang didirikan untuk mendapatkan keuntungan yang sebesar-besarnya. Guna mendapatkan keuntungan tersebut perusahaan perlu melakukan siklus pendapatan yang terdiri dari pesanan penjualan, pengiriman barang, penagihan dan piutang usaha, serta penagihan kas (Rudianto;2009:110).

II.6. Penggajian

Gaji dan upah merupakan bagian dari kompensasi-kompensasi yang paling besar yang diberikan perusahaan sebagai balas jasa kepada karyawannya. Dan bagi karyawan ini merupakan nilai hak dari prestasi mereka, juga sebagai motivator dalam bekerja(Mulyadi;2005:373)

Istilah gaji (*salary*) biasanya digunakan untuk pembayaran atas jasa material, administrative dan jasa-jasa yang sama. Tarif gaji biasanya diekspresikan dalam periode bulanan. Istilah upah (*wages*) biasanya digunakan untuk pembayaran kepada karyawan lapangan (pekerja kasar) baik yang terdidik maupun tidak terdidik (Mulyadi;2005:374).

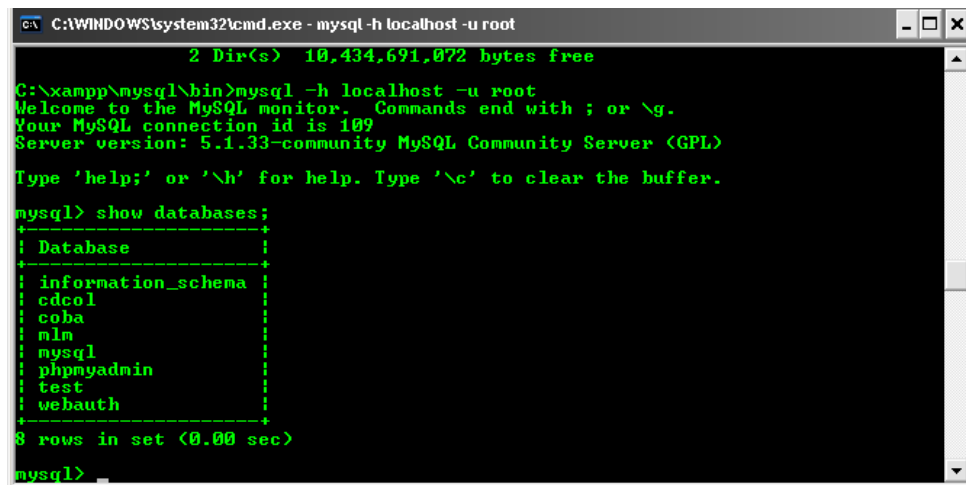
Gaji umumnya merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan yang mempunyai jenjang jabatan manajer, sedangkan upah umumnya merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksana (buruh). Umumnya gaji dibayarkan secara tetap perbulan, sedangkan upah dibayarkan berdasarkan hari kerja, jam kerja atau jumlah satuan produk yang dihasilkan.

II.7. Visual Basic. Net 2008

Pemrograman *Microsoft Visual Studio .NET* 2008 adalah sebuah *platform* untuk membangun, menjalankan dan meningkatkan generasi lanjut dari aplikasi terdistribusi. *.Net platform* merupakan *platform* terbaru untuk pemrograman aplikasi *window* dari *microsoft* dalam upaya meningkatkan produktivitas pembuatan sebuah program aplikasi dan memungkinkan terbukanya peluang untuk menjalankan program pada *multi sistem operasi* serta dapat memperluas pengembangan aplikasi *Client-Server*.

II.8. MySQL

MySQL (My Structure Query Language) atau yang biasa dibaca “mai-se-kuel” adalah sebuah program pembuatan *database* yang bersifat *open-source*, artinya siapa saja boleh menggunakannya dan tidak dicekal. *MySQL* sebenarnya produk yang berjalan pada *platform Linux*. Karena sifatnya yang *open source*, *MySQL* dapat dijalankan pada semua *platform* baik *Windows* maupun *Linux*. Selain itu, *MySQL* juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (Banyak Pengguna). Tampilan *MySQL* yang berjalan secara command dapat dilihat pada gambar II.2



```

C:\WINDOWS\system32\cmd.exe - mysql -h localhost -u root
2 Dir(s) 10,434,691,072 bytes free

C:\xampp\mysql\bin>mysql -h localhost -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 109
Server version: 5.1.33-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| coba      |
| mim       |
| mysql     |
| phpmyadmin |
| test      |
| webauth   |
+-----+
8 rows in set (0.00 sec)

mysql>

```

Gambar II.2. Tampilan MySQL Yang Berjalan Secara Command
Sumber : Rahmat Priyanto (2008 : 4)

Kelebihan lain dari MySQL adalah menggunakan bahasa *Query* standar yang dimiliki *SQL* (*Structure Query Language*). *SQL* adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti *Oracle*, *Progres SQL*, *SQL Server*, dan lain-lain. Sebagai sebuah program penghasil *database*, MySQL tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interface*). MySQL dapat didukung oleh hampir semua program aplikasi baik yang *open source* seperti *PHP* maupun yang tidak, yang ada pada *platform windows* seperti *Visual Basic*, *Delphi*, dan lainnya, Popularitas sebagai aplikasi *web* dikarenakan kedekatannya dengan popularitas *PHP*, sehingga seringkali disebut sebagai *Dynamic Duo*.

Untuk melakukan administrasi dalam data base MySQL, dapat menggunakan modul yang sudah termasuk yaitu *command-line* (Perintah: *MySQL* dan *MySQLAdmin*). Juga dapat diubah dari situs MySQL yaitu menjadi sebuah modul berbasis grafik (GUI): *MySQL Administrator* dan *MySQL Query Browser*. Selain itu terdapat juga sebuah perangkat lunak gratis administrasi *database*

MySQL berbasis *web* yang sangat populer yaitu *phpmyadmin*. Untuk perangkat lunak untuk administrasi *database MySQL* yang dijual secara komersial antara lain: *MySQL front*, *Navicat* dan *EMS SQL Manager for MySQL*. (Rahmat Priyanto ; 2008 : 4).

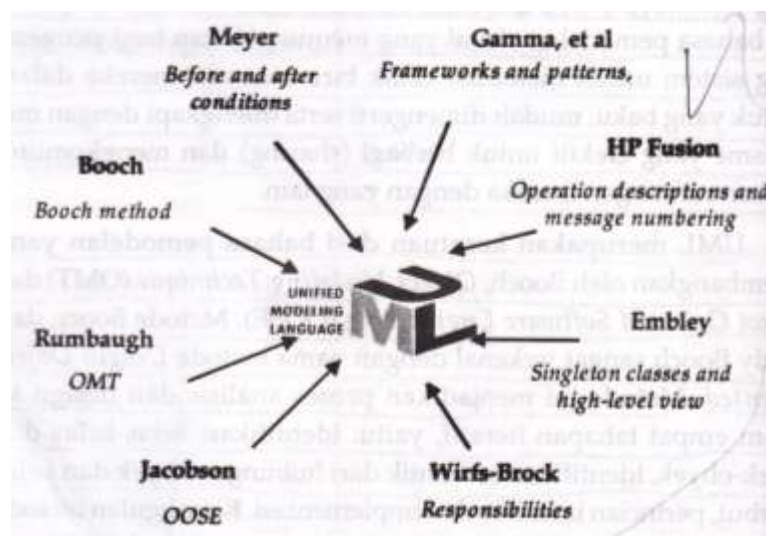
II.9. UML (*Unified Modelling Language*)

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 (Munawar;2005:17).

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 iteratif, 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.

Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya. Unsur-unsur yang membentuk UML ditunjukkan dalam Gambar II.3



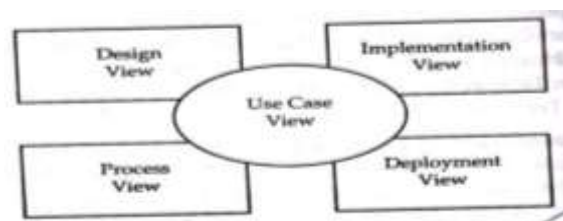
Gambar II.3 Unsur-unsur yang membentuk UML

Sumber: Pemodelan Visual dengan UML, Munawar, 2005:18

UML adalah hasil kerja dari konsorsium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam OOAD (*Object Oriented Analysis dan Design*). UML tidak hanya dominan dalam penotasian di lingkungan OO tetapi

juga populer di luar lingkungan OO. Ada tiga karakter penting yang melekat di UML yaitu sketsa, cetak biru dan bahasa pemrograman. Sebagai sebuah sketsa UML bisa berfungsi sebagai sebuah cetak biru karena sangat lengkap dan detil. Dengan cetak biru ini maka akan bisa diketahui informasi detil tentang coding program (*Forward engineering*) atau bahkan membaca program dan menginterpretasikannya kembali ke dalam diagram (*reverse engineering*). *Reverse engineering* sangat berguna pada situasi dimana kode program yang tidak terdokumentasi asli hilang atau bahkan belum dibuat sama sekali. Sebagai bahasa pemrograman, UML dapat menterjemahkan diagram yang ada di UML menjadi kode program siap untuk dijalankan.

UML dibangun atas model 4+1 *view*. Model ini didasarkan pada fakta bahwa struktur sebuah sistem dideskripsikan dalam *view* dimana salah satu diantaranya *use case view*. *Use case view* ini memegang peran khusus untuk mengintegrasikan *content* ke *view* yang lain. Model 4+1 *view* ditunjukkan pada gambar II.4



Gambar II.4 Model 4+1 View

Sumber: Pemodelan Visual dengan UML, Munawar, 2005:20

Kelima *view* tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap *view* berhubungan dengan perspektif tertentu dimana sistem akan diuji. *View* yang berbeda akan menekankan pada aspek yang

berbeda dari sistem yang mewakili tentang sistem bisa dibentuk dengan menggabungkan informasi-informasi yang ada pada kelima *view* tersebut.

Use case view mendefinisikan perilaku eksternal sistem. Hal ini menjadi daya tarik bagi *end user*, analis dan tester. Pandangan ini mendefinisikan kebutuhan sistem karena mengandung semua *view* yang lain yang mendeskripsikan aspek-aspek tertentu dari peran dan sering dikatakan yang mendrive proses pengembangan perangkat lunak.

Design view mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan di *use case*. *Design view* ini berisi definisi komponen program, class-class utama bersama-sama dengan spesifikasi data, perilaku dan interaksinya. Informasi yang terkandung di *view* ini menjadi perhatian para programmer karena menjelaskan secara detail bagaimana fungsionalitas sistem akan diimplementasikan.

Implementasi *view* menjelaskan komponen-komponen fisik dari sistem yang akan dibangun. Hal ini berbeda dengan komponen logic yang dideskripsikan pada *design view*. Termasuk disini diantaranya *file exe*, *library* dan *database*. Informasi yang ada di *view* dan integrasi sistem.

Proses *view* berhubungan dengan hal-hal yang berkaitan dengan *concurrency* dan dalam sistem. Sedangkan *deployment view* menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan dijalankan. Kedua *view* ini menunjukkan kebutuhan non fungsional dari sistem seperti toleransi kesalahan dan hal-hal yang berhubungan dengan kinerja (Munawar;2005:17-21).

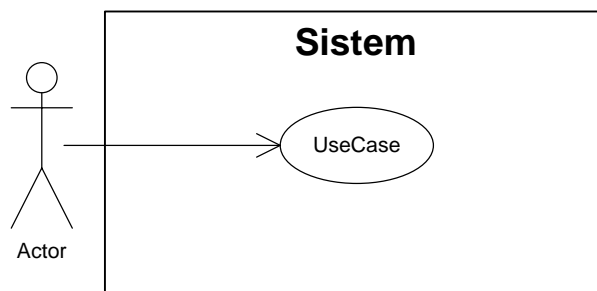
II.9.1. Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara deskripsikan 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.

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model *use case* adalah bagian dari model *requirement*. Termasuk disini adalah problem domain object dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari *perspektif user*.

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 II.5



Gambar II.5 Usecase Diagram

Sumber: Pemodelan Visual dengan UML, Munawar, 2005:64

Untuk mengidentifikasi *actor*, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. *Actor* adalah *abstraction* dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*.

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. Oleh karena itu sangat penting untuk memilih abstraksi yang cocok. *Use case* dibuat berdasarkan keperluan *actor*. *Use case* harus merupakan ‘apa’ yang dikerjakan *software* aplikasi, bukan ‘bagaimana’ *software* aplikasinya mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*. Namun *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama (Munawar;2005:63-66).

II.9.2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi

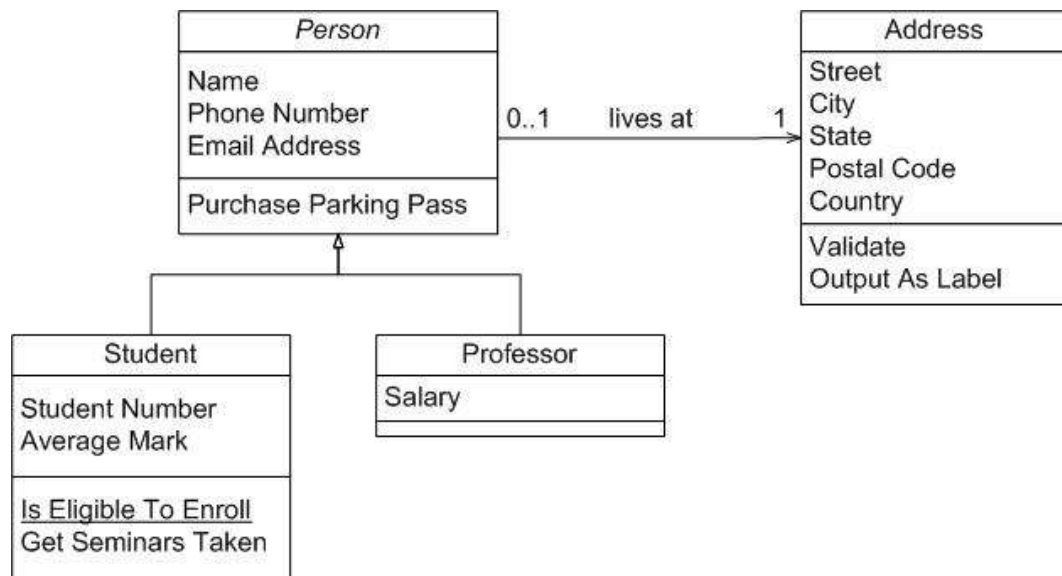
objek. *Class* menggambarkan keadaan (*atribut*/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda*/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok :

1. Nama kelas
2. Atribut
3. 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 II.6 dibawah ini:



Gambar II.6 Class Diagram


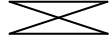
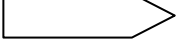
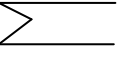

Sumber: Pemodelan Visual dengan UML, Munawar, 2005:220

II.9.3. 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 II.1 :

Tabel II.1. Simbol Activity Diagram

Notasi	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk

	menggabungkan dua kegiatan paralel menjadi satu
	<i>Gate</i> menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir (<i>Flow Final</i>)

Sumber : *Pemodelan Visual dengan UML, Munawar, 2005:110*

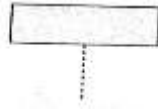
II.9.4. Sequence Diagram

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 II.7



Gambar II.7 Bentuk Participant

Sumber: Pemodelan Visual dengan UML, Munawar, 2005:88

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 II.8



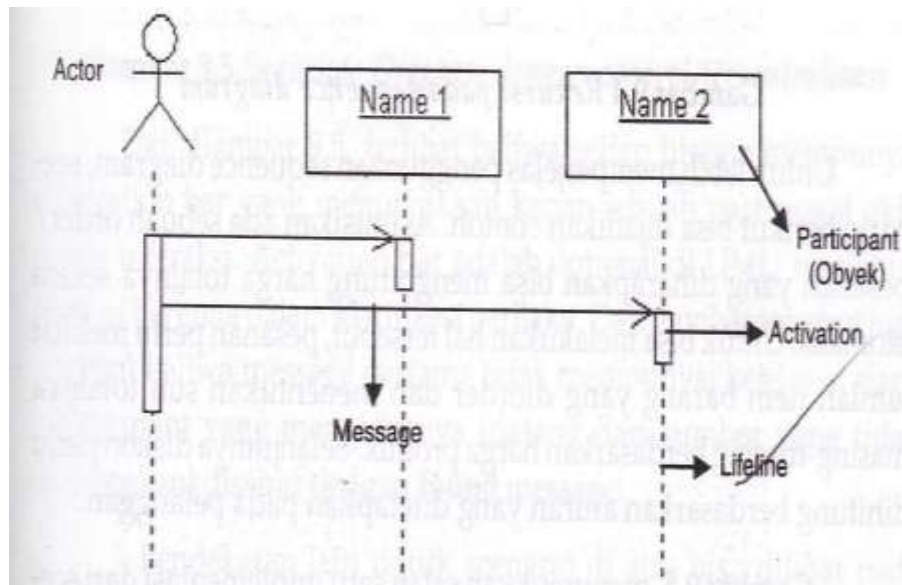
Gambar II.8 Bentuk Message

Sumber: Pemodelan Visual dengan UML, Munawar, 2005:88

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 II.9



Gambar II.9 Bentuk Messege

Sumber: Pemodelan Visual dengan UML, Munawar,2005:89

II.10. Sejarah Singkat Perusahaan

Arwana mas didirikan sejak tahun 2005 dengan memproduksi perhiasan pengganti emas dengan nama perusahaan PT. Arwana Mas Medan. Dengan misi membuat semua masyarakat dapat memakai perhiasan emas dengan harga yang sangat terjangkau dan sebagai tabungan karena dapat di jual kembali.

Produk yang diproduksi dengan merek Arwana ini adalah sebuah perhiasan yang menggantikan emas. Adapun perbedaan Arwana ini dengan emas adalah hanya bagian dalamnya saja. Bagian dalam terbuat dari logam alloy yang merupakan logam pencampuran emas. Dikarenakan bagian dalam tidak perlu menggunakan emas lagi karena tidak terlihat dan tersentuh.

Bagian luar dilapis dengan Emas asli yang hanya bisa dilakukan oleh Arwana karena menggunakan teknologi canggih dari Jerman.

Dengan adanya perhiasan pengganti emas ini maka banyak orang telah dapat memiliki perhiasan emas dengan harga yang sangat terjangkau dan banyak model untuk mempercantik diri.

Karena banyaknya permintaan dan respon dari masyarakat akan perhiasan pengganti emas Arwana ini maka Arwana terus memperluas jaringan pelayanan sampai ke seluruh pelosok Indonesia dan di provinsi Sumatera utara telah dikembangkan hampir 100 cabang di Medan dan sekitarnya.

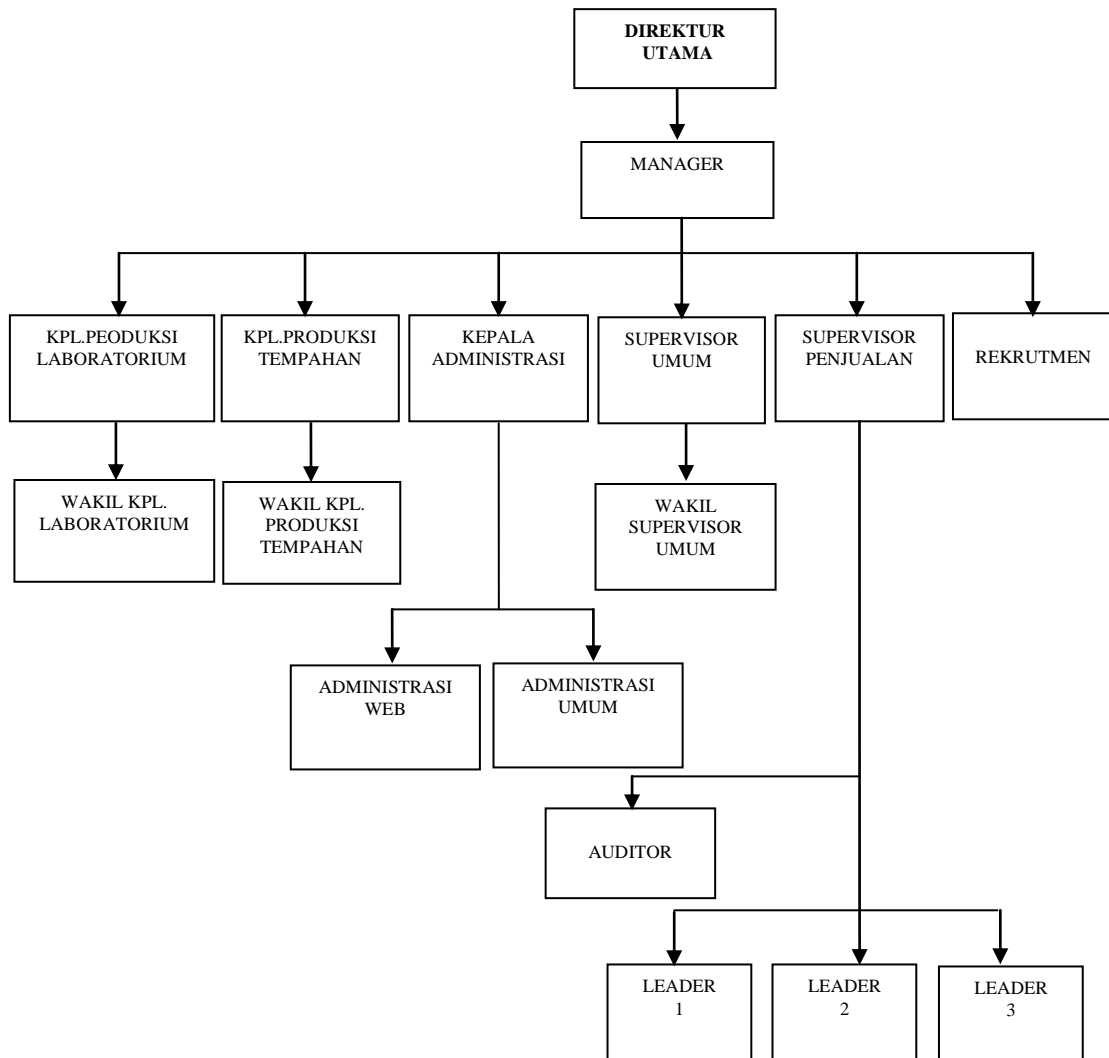
II.10.1. Struktur Organisasi

Organisasi pada dasarnya bertujuan untuk mengkoordinir suatu kelompok atau badan yang terdiri dari berbagai personalia serta mendayagunakan kemampuan yang ada secara keseluruhan untuk di arahkan kepada tujuan tertentu.

Selain itu dapat menggambarkan suatu rangkaian atau proses yang harus dilalui dalam perumusan tujuan. Pengambilan keputusan dan di lain pihak struktur ini merupakan perincian kegiatan yang harus dikerjakan masing-masing pegawai sesuai dengan bidang dan wewenangnya.

Struktur organisasi ini dapat juga disebut dengan kerangka dasar yang menunjukkan orang-orang yang ada di dalamnya terdapat tujuan yang telah ditetapkan dan biasanya disebut oleh pimpinan sebelum dijalankan kegiatan usaha dengan maksud agar lebih jelas dan terlihat akan susunan atau posisi dari setiap karyawan yang ada dalam organisasi tersebut. Peran dari seseorang pemimpin di dalam suatu organisasi akan sangat menentukan keberhasilan dari tujuan yang akan dicapai dalam organisasi itu sendiri. Dengan adanya struktur organisasi ini maka akan jelas bagi setiap anggota tugas dan tanggung jawabnya.

Adapun gambar struktur organisasi pada PT.Arwan Medan adalah :



Gambar II.10 Struktur Organisasi

Sumber : PT.Arwana Mas Medan

II.10.2. Pembagian Tugas, Wewenang Dan Tanggung Jawab

Untuk menggerakkan suatu organisasi dibutuhkan orang-orang yang memegang jabatan tertentu, dimana masing-masing orang melaksanakan tugas, wewenang dan tanggung jawab sesuai dengan jabatannya dalam uraian tugas. Organisasi yang baik adalah organisasi yang jelas dan teratur sehingga dalam

melaksanakan tugas dan tanggung jawab setiap pemegang jabatan memiliki gambaran batasan batasan tugas serta tanggung jawab tugas yang diembankan.

Uraian tugas dan tanggung jawab masing-masing staff yang terdapat pada struktur organisasi PT.Arwana Mas Indonesia adalah sebagai berikut:

1. Direktur Utama Arwana Mas Indonesia

Direktur Utama Arwana Mas Indonesia merupakan pemilik perusahaan PT.

Arwana mas Indonesia yang mempunyai wewenang tertinggi pada PT.Arwana.

2. *Manager*

Manager merupakann pimpinan tinggi PT.Arwana mas Indonesia yang mempunyai tanggungjawab penuh terhadap segala aktivitas perusahaan, dan merumuskan strategi- setrategi perusahaan. Tugas manager di PT.Arwana Mas Indonesia lebih ditekankan pada penjualan produk.

3. *Supervisor Penjualan*

Supervisor Penjualan merupakan karyawan yang bertanggungjawab penuh terhadap penjualan untuk memberikan pelayanan kepada pelanggan agar dapat mencapai target omset yang maksimal.

4. *Supervisor Umum*

Supervisor Umum mempunyai tugas dan tanggungjawab terutama pada produk atau ATK pada PT.Arwana Mas Indonesia.

5. Rekrutmen

Rekrutmen mempunyai tanggungjawab penuh untuk menyeleksi setiap karyawan yang akan bekerja pada PT.Arwana Mas Idonesia.

6. Kepala Administrasi

Kepala Administrasi mempunyai tanggungjawab untuk mengatur segala keuangan yang ada pada PT.Arwana Mas Indonesia.

7. Kepala Produksi Tempahan.

Kepala Produksi Tempahan mempunyai tugas untuk memproduksi beberapa macam produk yang telah ditentukan oleh pelanggan yang telah melakukan pemesanan kepada PT.Arwana Mas Indonesia.

8. Kepala Produksi Laboratrium

Kepala Produksi Laboratorium mempunyai tanggung jawab penuh terhadap mutu produk yang dihasilkan dan pelaksanaan analisis terhadap setiap produk yang dihasilkan dan didalam pelaksanaan tugas- tugas kepala produksi laboratorium dibantu oleh wakil dan anggota laboratorium.

9. Auditor

Auditor merupakan bagian yang memiliki wewenang untuk mengecek atau memeriksa segala produk yang adapada agen yang ada dilapangan.

10 *Leader*

Leader mempunyai tugas dan wewenang untuk mengawasi hasil kerja agen yang ada di lapangan.