

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Sistem Informasi

Suatu sistem dapat dirumuskan sebagai setiap kumpulan komponen atau subsistem yang dirancang untuk mencapai suatu tujuan. Norman L. Enger menyatakan bahwa suatu sistem dapat terdiri dari kegiatan-kegiatan yang berhubungan guna mencapai tujuan-tujuan perusahaan. Sedangkan Prof. Dr. Mr. S. Prajudi Atmosudirdjo menyatakan suatu sistem terdiri atas objek-objek, atau unsur-unsur, atau komponen-komponen yang berkaitan dan berhubungan satu sama lainnya sedemikian rupa sehingga unsur-unsur tersebut merupakan suatu kesatuan pemrosesan atau pengolahan tertentu (Tata Sutabri; 2004: 10).

II.1.1. Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem (Abdul Kadir; 2003: 54).

II.1.2. Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem yaitu :

a. Tujuan

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem lain berbeda-beda.

b. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tak tampak.

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna. Pada sistem informasi, proses dapat berupa suatu tindakan yang bermacam-macam. Meringkas data, melakukan penghitungan, dan mengurutkan data merupakan beberapa contoh proses.

d. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan, laporan dan sebagainya.

e. Mekanisme dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*) yang mencuplik keluaran.

Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan (Abdul Kadir; 2003: 54-57).

II.1.3. Karakteristik Sistem

Sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

a. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

b. Batasan Sistem

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lain atau sistem dengan lingkungan luarnya.

c. Lingkungan Luar Sistem

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem.

d. Penghubung Sistem

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain.

e. Masukan Sistem

Energi yang dimasukan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*).

f. Keluaran Sistem

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

g. Pengolah Sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

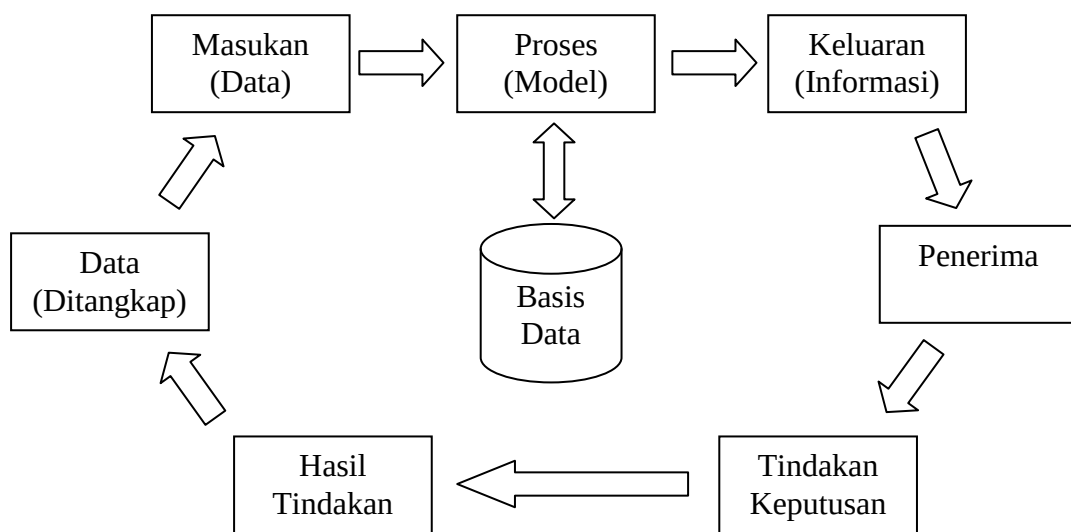
h. Sasaran Sistem

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dapat dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri; 2004: 12-13).

II.1.4. Pengertian Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya (Tata Sutabri; 2004 :18).

McFadden, dkk (1999) mendefenisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Menurut Davis (1999), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat sekarang (Abdul Kadir; 2003: 31).



Gambar II.1 Siklus informasi
(Sumber : Abdul Kadir; 2003: 32)

II.1.5. Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri; 2004: 36).

Menurut Bodnar dan Hopwood (1993), sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna. Menurut Turban, McLean dan Wetherbe (1999), sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik. Sedangkan menurut Hall (2001), sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai. Dari definisi tersebut dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan (Abdul Kadir; 2003: 10-11).

II.1.6. Sistem Informasi Akuntansi

Akuntansi adalah suatu proses yang meliputi; pencatatan, penggolongan, pengikhtisaran dan pelaporan transaksi keuangan perusahaan yang terjadi dalam suatu periode tertentu (Drs. Hendi Somantri; 2005: 19).

Sistem informasi akuntansi adalah sistem informasi yang paling tua dan paling banyak digunakan dalam bisnis. Bodnar dan Hopwood (1993) mendefinisikan sistem informasi akuntansi sebagai kumpulan sumber daya yang dirancang untuk mentransformasikan data keuangan menjadi informasi. Gelinas, Orams dan Wiggins (1997) mendefinisikannya sebagai subsistem khusus dari sistem informasi manajemen yang tujuannya adalah menghimpun, memproses,

dan melaporkan informasi yang berkaitan dengan transaksi keuangan (Abdul Kadir; 2003: 98-99).

Sistem Informasi Akuntansi suatu organisasi memainkan peranan penting dalam membantu organisasi mengadopsi dan mempertahankan posisi strategis. Mencapai kesesuaian yang baik antar aktivitas membutuhkan pengumpulan data tiap aktivitas. Hal lain yang juga penting adalah sistem informasi harus mengumpulkan dan mengintegrasikan baik data keuangan, maupun non-keuangan dan aktivitas-aktivitas organisasi (Marshall B. Romney, dkk; 2004; 17)

Sistem Informasi Akuntansi terdiri dari lima komponen:

- a. Orang-orang yang mengoperasikan sistem tersebut dan melaksanakan berbagai fungsi.
- b. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam mengumpulkan, memproses dan menyimpan data tentang aktivitas-aktivitas organisasi.
- c. Data tentang proses-proses bisnis organisasi.
- d. Software yang dipakai untuk memproses data organisasi.
- e. Infrastruktur teknologi informasi, termasuk komputer, peralatan pendukung (*peripheral device*) dan peralatan untuk komunikasi jaringan.

Kelima komponen ini secara bersama-sama memungkinkan suatu SIA memenuhi tiga fungsi pentingnya dalam organisasi, yaitu:

- a. Mengumpulkan dan menyimpan data tentang aktivitas-aktivitas yang dilaksanakan oleh organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut, dan para pelaku yang terlibat dalam berbagai aktivitas tersebut, agar pihak manajemen, para pegawai dan pihak-pihak luar yang berkepentingan dapat meninjau ulang (*review*) hal-hal yang telah terjadi.
- b. Mengubah data menjadi informasi yang berguna bagi pihak manajemen untuk membuat keputusan dalam aktivitas perencanaan, pelaksanaan dan pengawasan.
- c. Menyediakan pengendalian yang memadai untuk menjaga aset-aset organisasi, termasuk data organisasi, untuk memastikan bahwa data tersebut tersedia saat dibutuhkan, akurat, dan andal (Marshall B. Romney, dkk; 2004: 3)

II.2. *Entity Relationship Diagram-ERD*

Entity adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi (Al-Bahra bin Ladjamuddin; 2004: 124).

Entity Relationship (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antarobjek. Entitas adalah sesuatu atas objek dalam dunia nyata yang dapat dibedakan dari objek lain.

Entitas digambarkan dalam basisdata dengan kumpulan atribut. Misalnya atribut nim, nama, alamat dan kota bisa digambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

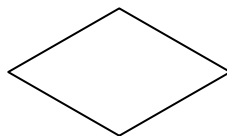
Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terdapat dua mahasiswa dengan nama, alamat dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa (Janner Simarmata,dkk; 2006: 59).

Entiti dilambangkan dengan bentuk persegi panjang, seperti tampak pada gambar berikut :



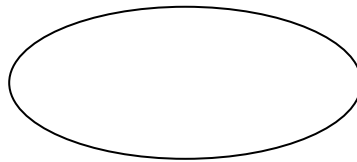
Gambar II.2 Lambang Entity Set
(Sumber : Kusrini; 2007: 21)

Relationship adalah hubungan diantara beberapa entiti. *Relationship set* adalah sekumpulan relasi yang mempunyai tipe yang sama. *Relationship set* digambarkan dengan diamond seperti tampak pada gambar di bawah ini.



Gambar II.3 Lambang Relationship Set
(Sumber : Kusrini; 2007: 21)

Atribut merupakan sebutan untuk mewakili suatu entity. Atribut dalam ERD dilambangkan dengan bentuk elips, seperti tampak pada gambar di bawah ini.



Gambar II.4 Lambang Atribut
(Sumber : Kusrini; 2007: 22)

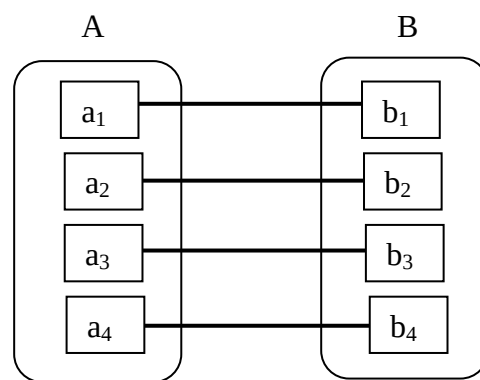
II.2.1. Pemetaan Kardinalitas

Pemetaan kardinalitas menyatakan jumlah entitas dimana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi.

Pemetaan kardinalitas sangat berguna dalam menentukan himpunan relasi biner meskipun pemetaan dapat berperan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

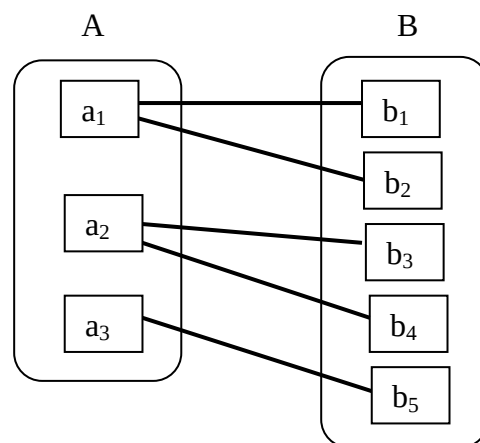
Untuk suatu himpunan relasi biner R antara himpunan entitas A dan B , pemetaan kardinalitas harus salah satu dari berikut:

- a. One-to-One, sebuah entitas A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A .



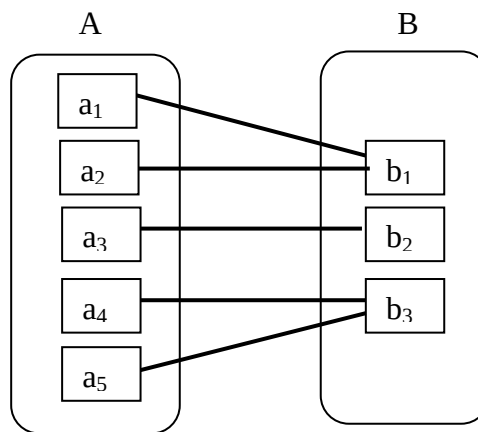
Gambar II.5 Hubungan one-to-one
(Sumber : Janner Simarmata,dkk; 2006: 64)

- b. One-to Many, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A.



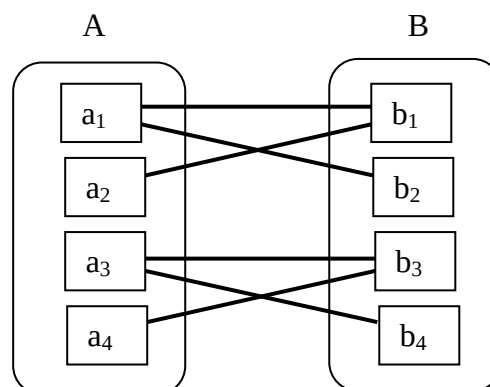
Gambar II.6 Hubungan one-to-many
(Sumber : Janner Simarmata,dkk; 2006: 64)

- c. Many-to-One, sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



Gambar II.7 Hubungan many-to-one
(Sumber : Janner Simarmata,dkk; 2006: 64)

- d. Many-to-Many, sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.



Gambar II.8 Hubungan many-to-many
(Sumber : Janner Simarmata,dkk; 2006: 64)

II.3. Kamus Data

Kamus data (KD) atau data dictionary (DD) atau disebut juga dengan istilah systems data dictionary adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefenisikan data yang mengalir disistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir disistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. KD digunakan untuk merancang input, laporan-laporan dan database. KD dibuat berdasarkan arus data yang ada di DAD (Jogiyanto; 2005: 725).

II.3.1. Isi Kamus Data

KD harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya. Untuk maksud keperluan ini, maka KD harus memuat hal-hal berikut ini:

- a. Nama arus data

Karena KD dibuat berdasarkan arus data yang mengalir di DAD, maka nama dari arus data juga harus dicatat di KD.

b. Alias

Alias atau nama lain dari data dapat dituliskan bila nama lain ini ada.

Alias perlu ditulis karena data yang sama mempunyai nama yang berbeda untuk orang atau departemen satu dengan yang lainnya.

c. Bentuk data

Bentuk dari data ini perlu dicatat di KD, karena dapat digunakan untuk mengelompokkan KD ke dalam kegunaannya sewaktu perancangan sistem. KD yang mencatat data yang mengalir dalam bentuk dokumen dasar atau formulir akan digunakan untuk merancang bentuk input sistem.

d. Arus data

Arus data menunjukkan dari mana data mengalir kemana data akan menuju. Keterangan arus data ini perlu dicatat di KD supaya memudahkan mencari arus data ini di DAD.

e. Penjelasan

Untuk lebih memperjelas lagi tentang makna dari arus data yang dicatat di KD, maka bagian penjelasan dapat diisi dengan keterangan-keterangan tentang arus data tersebut.

f. Periode

Periode ini menunjukkan kapan terjadinya arus data ini. Periode perlu dicatat di KD karena dapat digunakan untuk mengidentifikasi kapan input data harus dimasukkan ke sistem, kapan proses dari program harus dilakukan dan kapan laporan-laporan harus dihasilkan.

g. Volume

Volume yang perlu dicatat di KD adalah tentang volume rata-rata dan volume puncak dari arus data. Volume rata-rata menunjukkan banyaknya rata-rata arus data yang mengalir dalam satu periode tertentu dan volume puncak menunjukkan volume yang terbanyak.

h. Struktur data

Struktur data menunjukkan arus data yang dicatat di KD terdiri dari item-item data apa saja (Jogiyanto; 2005: 726-728).

II.3.2. Mendefenisikan Struktur Data

Struktur dari data terdiri dari elemen-elemen data yang disebut dengan item data, sehingga secara prinsip struktur dari data ini dapat digambarkan dengan menyebutkan nama dari item-item datanya. Nama dari item data saja yang dicatat di KD tidaklah cukup, masih diperlukan informasi lainnya dari struktur data tersebut, seperti misalnya informasi tentang apakah item data tersebut pasti ada atau hanya bersifat optional (dapat ada dan dapat tidak ada). Biasanya untuk menunjukkan informasi-informasi tambahan ini di KD dipergunakan notasi-notasi sebagai berikut ini (Jogiyanto; 2005: 730).

Tabel II.1 Notasi Struktur Data

Notasi	Keterangan
=	Terdiri dari
+	AND (dan)
()	Pilihan (boleh Ya atau Tidak)
{ }	Iterasi / pengulangan proses

[]	Pilih salah satu pilihan
I	Pemisah pilihan didalam tanda []
*	Keterangan atau catatan
@	Petunjuk (key field)

(Sumber : Tata Sutabri; 2004: 172)

II.4. Normalisasi

Normalisasi adalah bagian perancangan basis data. Tanpa normalisasi, sistem basisdata menjadi tidak akurat, lambat, tidak efisien, serta tidak membreikan data yang diharapkan.

Pada waktu menomalisasi basisdata, ada empat tujuan yang harus dicapai, yaitu:

- Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
- Meminimalkan jumlah data berulang dalam basisdata.
- Membuat basisdata yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integritas data.
- Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda akan hanya mengubah pada satu tempat (Janner Simarmata,dkk; 2006: 77).

Normalisasi adalah suatu file yang terdiri dari beberapa grup elemen yang berulang-ulang perlu diorganisasikan kembali. Proses untuk mengorganisasikan file untuk menghilangkan grup elemen yang berulang-ulang

ini disebut dengan normalisasi (*normalization*). Normalisasi juga banyak dilakukan dalam merubah bentuk database dari struktur pohon atau struktur jaringan menjadi struktur hubungan. Konsep dan teknik normalisasi ini dikenalkan oleh Dr. E.F Codd di papernya pada tahun 1970 dan 1972. E.F Codd dalam papernya ini mendefinisikan struktur data yang baru, yaitu yang disebut dengan struktur data hubungan (*relational data structure*). Istilah data hubungan menunjukkan suatu struktur data yang mempunyai hubungan dengan elemen-elemen data lainnya, baik dalam satu file atau dalam file yang lain (Jogiyanto; 2005: 403-404).

Normalisasi adalah suatu teknik yang menstrukturkan data dalam cara tertentu untuk membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam *database*. Proses normalisasi menghasilkan struktur *record* yang konsisten secara logis yang mudah dimengerti dan sederhana dalam pemeliharaannya. Pada proses normalisasi ini perlu dikenal dulu definisi dari tahap normalisasi, yaitu :

a. Bentuk Tidak Normal (*Unnormalized Form*)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak diharuskan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaannya.

b. Bentuk Normal Pertama (1NF)

Bentuk normal ke satu mempunyai ciri yaitu bahwa setiap data dibentuk dalam flat file (file datar/rata), data dibentuk dalam satu *record* demi *record* dan nilai dari field berupa *atomic value*.

c. Bentuk Normal Kedua (2NF)

Bentuk normal kedua mempunyai syarat, yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukanlah kunci haruslah bergantung secara fungsi pada kunci utama/*primary key*.

d. Bentuk Normal Ketiga (3NF)

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif.

e. Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribut *superkey* (Tata Sutabri; 2004: 207-208).

II.5. Database

Basis data adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya.

Sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekan serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan (Linda Marlinda; 2004: 1).

II.5.1. Komponen Lingkungan Basis Data

Komponen-komponen yang menyusun lingkungan basis DBMS terdiri atas:

a. Perangkat Keras

Perangkat keras digunakan untuk menjalankan DBMS beserta aplikasi-aplikasinya. Perangkat keras berupa komputer dan periferal pendukungnya. Komputer dapat berupa PC, minikomputer, *mainframe*, dll.

b. Perangkat Lunak

Komponen perangkat lunak mencakup DBMS itu sendiri, program aplikasi, serta perangkat lunak pendukung untuk komputer dan jaringan. Program aplikasi dapat dibangun dengan menggunakan bahasa pemrograman C++, Pascal, Delphi atau Visual Basic.

c. Data

Bagi si pemakai, komponen terpenting dalam DBMS adalah data karena dari data inilah pemakai dapat memperoleh informasi yang sesuai dengan kebutuhan masing-masing.

d. Prosedur

Prosedur adalah petunjuk tertulis yang berisi cara merancang hingga menggunakan basis data. Dalam hal ini, prosedur terdiri dari:

a) Cara masuk ke DBMS (*login*)

- b) Cara memakai fasilitas-fasilitas tertentu dalam DBMS maupun cara menggunakan aplikasi
- c) Cara mengaktifkan dan menghentikan DBMS
- d) Cara membuat cadangan basis data dan cara mengembalikan cadangan ke DBMS
- e. Orang
 - Komponen orang dapat dibagi menjadi tiga kelompok, antara lain:
 - a) Pemakai akhir (*end-user*)
 - b) Pemrogram aplikasi
 - c) Administrator basis data (Abdul Kadir; 2003: 258-260).

II.5.2. Tujuan Basis Data

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, kecepatan dalam pengambilan kembali. Untuk mencapai tujuannya, syarat sebuah basis data adalah sebagai berikut :

- a. Tidak adanya redudansi dan inkosistensi data

Redudansi terjadi jika suatu informasi disimpan di beberapa tempat. Akibat dari redudansi adalah inkosistensi data atau data yang tidak konsisten.
- b. Kesulitan Pengaksesan Data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *Query* ataupun dari tool untuk melihat tabelnya.
- c. *Multiple User*

Basis data memungkinkan penggunaan data bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda (Kusrini; 2007: 2-5)

II.5.3. Manfaat/Kelebihan Basis Data

Banyak manfaat yang dapat kita peroleh dengan menggunakan basis data.

Manfaat/kelebihan basis data diantaranya adalah :

a. Kecepatan dan Kemudahan (*speed*)

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika.

b. Kebersamaan Pemakai (*shareability*)

Sebuah basis data dapat digunakan oleh banyak *user* dan banyak aplikasi. Untuk data-data yang diperlukan oleh banyak bagian/orang, tidak perlu dilakukan pencatatan dimasing-masing bagian, tetapi cukup dengan satu basis data untuk dipakai bersama.

c. Pemusatan Kontrol Data

Karena cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan disatu tempat saja.

d. Efisiensi Ruang Penyimpanan (*space*)

Dengan pemakaian bersama, kita tidak perlu menyediakan tempat penyimpanan diberbagai tempat, tetapi cukup satu saja sehingga ini akan menghemat ruang penyimpanan yang dimiliki oleh sebuah organisasi.

e. Keakuratan (*accuracy*)

Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antar data dan lain-lain dapat menekankan ketidakakuratan dalam pemasukkan/penyimpanan data.

f. Ketersediaan (*availability*)

Dengan basis data kita dapat mem-*backup* data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu kita simpan ditempat lain.

g. Kelengkapan (*completeness*)

h. Keamanan (*security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna. Basis data bisa diberikan *password* untuk membatasi orang yang mengaksesnya.

i. Kemudahan dalam pembuatan program aplikasi baru

Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan DBMS, sehingga pembuat aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

j. Pemakaian secara langsung

Basis data memiliki fasilitas untuk melihat datanya secara langsung dengan *tool* yang disediakan oleh DBMS. Untuk melihat data, langsung ke tabel ataupun dengan menggunakan *query*.

k. Kebebasan Data (*data independence*)

Jika sebuah program telah selesai dibuat dan ternyata ada perubahan isi/struktur data. Maka dengan basis data, perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

l. *User view*

Basis data menyediakan pandangan yang berbeda-beda untuk tiap-tiap pengguna. Tidak semua data boleh diakses oleh semua pengguna (Kusrini; 2007: 5-9).

II.5.4. Istilah-istilah yang Dipergunakan di dalam Sistem Basis Data

- a. *Enterprise*, suatu bentuk organisasi, seperti: bank, universitas, pabrik, dan lain-lain. Data yang disimpan di dalam basisdata merupakan data operasional suatu enterprise.
- b. Entitas, suatu obyek yang dapat dibedakan dengan objek lainnya yang dapat diwujudkan didalam basisdata.
- c. *Attribute/field*, karakteristik entitas tertentu.
- d. Data value (nilai atau isi data) merupakan data actual atau informasi yang disimpan ditiap data elemen atau atribut. Isi atribut disebut nilai data.

- e. Record, kumpulan isi elemen data (*atriute*) yang saling berhubungan menginformasikan tentang suatu entiti secara lengkap.
- f. File, kumpulan record sejenis yang mempunyai panjang elemen dan atribut yang sama, namun berbeda-beda data valuenya.
- g. Kunci elemen data, sebagai tanda pengenalan yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas.
- h. Database Management System (DBMS), kemudian file yang saling berkaitan bersama dengan program untuk pengelolaannya (Linda Marlinda; 2004: 4-5).

II.5.5. MySQL

MySQL database server adalah RDBMS (Relasional Database Management System) yang dapat menangani data yang bervolume besar. Meskipun begitu, tidak menuntut resource yang besar. MySQL adalah database yang paling populer diantara database-database yang lain.

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. Keunggulan MySQL adalah kemampuannya dalam menyediakan berbagai fasilitas atau fitur-fitur yang dapat digunakan oleh bermacam-macam user. User ini sendiri termasuk administrator database, programmer aplikasi, manager, sampai end user (pemakai akhir).

MySQL adalah sebuah manajemen sistem database server yang mampu menangani beberapa user, yaitu mampu menangani beberapa instruksi sekaligus

dari beberapa user dalam satu waktu. Dan, MySQL merekam semua data user didalam sistemnya dalam tabel user.

MySQL memiliki beberapa kelebihan dan keuntungan dibanding database lain, diantaranya adalah :

- a. Banyak ahli berpendapat MySQL merupakan server tercepat.
- b. MySQL merupakan sistem manajemen database yang OpenSource (kode sumbernya terbuka), yaitu software ini bersifat free atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuaynya.
- c. MySQL mempunyai performa yang tinggi tapi simpel.
- d. Database MySQL mengerti bahasa SQL (Structured Query Language).
- e. Semua klien dapat mengakses server dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses database.
- f. Database MySQL dapat diakses dari semua tempat diinternet dengan hak akses tertentu.
- g. MySQL dapat berjalan diberbagai operating sistem seperti Linux, Windows, Solaris dan lain-lain (Wahana Komputer; 2010: 5-9)

II.6. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. 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 mereka dengan yang lain (Munawar; 2005: 17).

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Karena ini merupakan sebuah bahasa, UML mempunyai sejumlah aturan untuk menggabungkan/mengkombinasikan elemen-elemen tersebut (Munawar; 2005: 22).

Meskipun UML sudah banyak menyediakan diagram yang bisa membantu mendefinisikan sebuah aplikasi, tidak berarti bahwa semua diagram tersebut akan bisa menjawab persoalan yang ada. Dalam banyak kasus, diagram lain selain UML sangat banyak membantu (Munawar; 2005: 23).

Tabel II.2 Tipe Diagram UML

Diagram	Tujuan
Activity	Perilaku prosedural & paralel
Class	Class, fitur, dan relasinya
Communication	Interaksi diantara obyek. Lebih menekankan ke Link
Component	Struktur dan koneksi dari komponen
Deployment	Penyebaran / instalasi ke klien
Package	Struktur hierarki saat kompilasi
Sequence	Interaksi antar obyek. Lebih menekankan pada urutan
State Machine	Bagaimana event mengubah sebuah obyek
Use Case	Bagaimana user berinteraksi dengan sebuah sistem

(Sumber : Munawar; 2005 : 23)

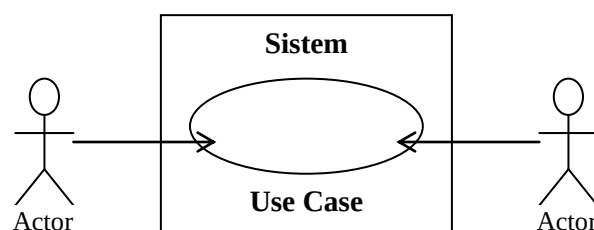
II.6.1. Notasi *Use Case*

II.6.1.1 *Use Case*

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antar user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langka-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. *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*.

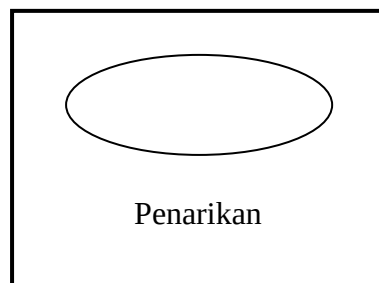
Diagram use case menunjukkan 3 aspek dari sistem yaitu : *actor*, *use case* dan sistem/sub sistem *boundary*. *Actor* mewakili peran orang, sistem yang lain atau alat berkomunikasi dengan *use case* (Munawar; 2005: 63-65).



Gambar II.9. Use Case Model

Sumber : “ Pemodelan Visual Dengan UML (Munawar; 2005: 64)”

Menurut (Pilone, 2005; bab7.1) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas. Sedangkan (Whitten, 2004; 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait, baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval* (Prabowo Pudjo Widodo; 2011: 21-22).



Gambar II.10 Simbol Use case

(Sumber : Prabowo Pudjo Widodo; 2011: 22)

II.6.2. Class Diagram

Class dalam notasi UML digambarkan dengan kotak. Nama class menggunakan huruf besar diawal kalimatnya dan diletakkan di atas kotak. Bila *class* mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku

kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar (Munawar; 2003: 35).



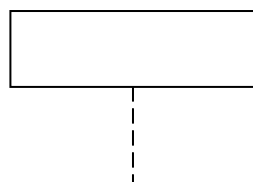
Gambar II.11. Notasi class di UML
(Sumber : Munawar; 2005: 64)

Atribut adalah properti dari sebuah *class*. Atribut ini melukiskan batas nilai yang mungkin ada pada *obyek* dari *class*. Sebuah *class* mungkin mempunyai nol atau lebih atribut. Secara konvensi, jika nama atribut terdiri dari atas satu suku kata, maka ditulis dengan huruf kecil. Akan tetapi jika nama atribut mengandung lebih dari satu suku kata maka semua kata digabungkan dengan suku kata pertama menggunakan huruf kecil dan awal suku kata berikutnya menggunakan huruf besar (Munawar; 2005: 35).

II.6.3. Sequence Diagram

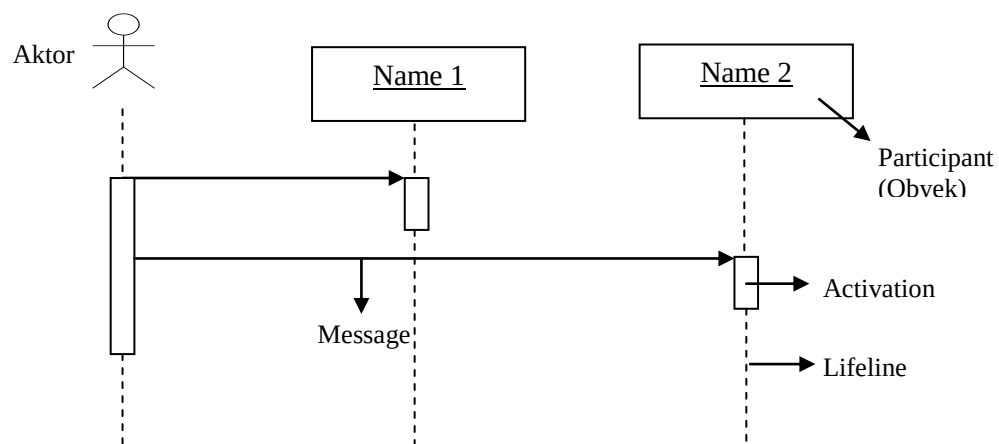
Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh *obyek* dan *message* (pesan) yang diletakkan diantara *obyek-obyek* ini di dalam *use case*.

Komponen utama *sequence* diagram terdiri atas *obyek* yang dituliskan dengan kotak segiempat bernama, *message* diwakili oleh garis dengan tanda panah (Munawar; 2005: 87).



Gambar II.12. Participant pada sebuah sequence diagram
(Sumber : Munawar; 2005: 88)

Gambar II.12. menunjukkan esensi simbol dari *sequence diagram* dan simbol kerjanya secara bersama-sama. *Participant* terletak disebelah atas. Setiap *lifeline* menggunakan garis putus-putus yang menurun dari *participant*. Garis yang solid dengan tanda panah menghubungkan antara satu *lifeline* dengan *lifeline* yang lain dan mewakili sebuah *message* dari satu *participant* ke *participant* yang lain. Dari gambar tersebut terlihat seorang *actor* menginisialisasi *sequence diagram* meskipun *actor* bukan bagian dari *sequence diagram*.



Gambar II.13. Simbol-simbol yang ada pada sequence diagram
(Sumber : Munawar; 2005: 89)

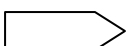
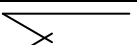
II.6.4. Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan 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 (Munawar; 2005: 109).

Berikut adalah simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram*.

Tabel II.3 Simbol-simbol yang sering dipakai pada *activity diagram*

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Rake; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (Flow Final)

(Sumber : Munawar; 2005: 110)

II.7. Java

Java dibuat pada tahun 1995 di Sun Microsystem. *Java* merupakan bahasa berorientasi objek dan serbaguna. Kode *Java* dikompilasi dalam format yang disebut *bytecode*. *Bytecode* ini dapat dijalankan di semua komputer yang telah dilengkapi dengan program *Java Interpreter* dan *JavaVirtual Machine*.

Java sangat populer karena pada masa awal internet menjadi populer. *Java* telah menyediakan sarana untuk membuat program (yang disebut *applet*) yang berjalan pada *web browser*. Bahasa ini juga mendukung koneksi ke *database*, menyediakan sarana untuk membuat aplikasi berbasis windows dan juga dapat dipakai untuk pemrograman jaringan (Abdul Kadir : 2003 : 231-232).

Sebagai sebuah bahasa pemrograman, *Java* dapat membuat seluruh bentuk aplikasi, *desktop*, *web* dan lainnya. Sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan *Java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

OOP (*Object Oriented Programming* – Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di *Java* adalah Objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis *Java* secara cepat, tepat, mudah dan

terorganisir. Kelebihan ini menjadikan *Java* sebagai salah satu bahasa pemrograman termudah, bahkan untuk fungsi-fungsi yang *advance* seperti komunikasi antara komputer sekalipun (Hendra Kurniawan; 2011: 4-5).

II.7.1. Tipe Data *Java*

Seperti bahasa pemrograman lainnya, *Java* mempunyai 8 tipe data primitif, yang mana 4 bertipe *integer* (bilangan bulat), 2 bertipe *floating-point* (bilangan pecahan), dan 2 yang terakhir bertipe *boolean* dan *char*. Tipe data tersebut antara lain :

a. *Integer*

- a) *Byte*: memiliki nilai integer dari -128 sampai +127 dan menempati 1 byte (8 bits) di memori.
- b) *Short*: memiliki nilai integer dari -32768 sampai 3276 dan menempati 2 bytes (16 bits) di memori.
- c) *Int*: memiliki nilai integer dari -2147483648 sampai 2147483647 dan menempati 4 bytes (32 bits) di memori.
- d) *Long*: memiliki nilai dari -9223372036854775808 sampai 9223372036854775807 dan menempati 8 bytes (64 bits) di memori.

b. *Floating Point*

- a) *Float*: memiliki nilai -3.4×10^8 (pangkat '8') sampai $+3.4 \times 10^8$ (pangkat '8') dan menempati 4 byte di memori.

- b) *Double*: memiliki nilai -1.7×10^{308} sampai $+1.7 \times 10^{308}$ (pangkat 308).

c. ***Boolean dan Char***

- a) *Boolean*: dalam *Java* dikenal tipe data *boolean* yang terdiri dari dua nilai saja, yaitu *true* dan *false*. *Boolean* sangat penting dalam mengevaluasi suatu kondisi, dan sering digunakan untuk menentukan alur program.
- b) *Char*: *char* adalah karakter tunggal yang didefinisikan dengan diawali dan diakhiri dengan tanda ' (petik tunggal). *Char* berbeda dengan *String*, karena *String* bukan merupakan tipe data primitif tetapi sudah merupakan sebuah objek.

II.7.2. Pemodelan Objek dan Kelas

II.7.2.1. Obyek

Dalam Pemrograman Berorientasi Obyek (PBO) dengan *Java*, modul program dimodelkan, dirancang dan ditulis sebagai *obyek*. Kerangka berpikirnya berbeda dengan pemrograman prosedural. Pemrograman prosedural memisahkan antara data, dan prosedur yang mengolah data tersebut. Sedangkan di dalam *Java*, data dan prosedur ini dipadukan sebagai *obyek*. prosedur disini sering disebut dengan berbagai istilah seperti kapasitas, kemampuan, fungsi, metoda, perilaku dan sebagainya yang menggambarkan kemampuan kelas atau kelas untuk melakukan sesuatu tindakan.

Sebagaimana *obyek* di alam yang mempunyai ukuran dan tingkah laku, sebuah *obyek* didalam program juga bisa mempunyai data dan prosedur. Di dalam *Java*, rancangan sebuah *obyek* ditulis dalam sebuah *class*. Didalam *class* data diimplementasikan sebagai member *variable*, sedangkan prosedur yang mengolah data tersebut diimplementasikan dalam *method* (Hendra Kurniawa; 2011: 9-10).

II.7.2.2. Kelas

Dalam konsep PBO, kelas adalah merupakan wadah yang berisi abstraksi atau model dari suatu objek atau kelas dapat dikatakan sebagai kumpulan *obyek* yang mempunyai data/atribut dan metoda yang sama. Contoh yang menjelaskan hubungan antara kelas dan *obyek* secara sederhana adalah misal karyawan adalah suatu kelas, sedangkan Tono, Tini, Tina adalah nama-nama anggota atau *obyek* dari kelas karyawan (Hendra Kurniawa; 2011: 10).

GAMBARAN UMUM PERUSAHAAN

II.8. Sejarah Singkat BMT Darul Ihsan

BMT Darul Ihsan yang bertempat di Kecamatan Air Batu Kabupaten Asahan. Pada awal berdirinya BMT Darul Ihsan tahun 2006 sebagai Ketua BMT Darul Ihsan adalah Bapak Helmi, SE hingga sekarang.

BMT Darul Ihsan adalah sebuah lembaga keuangan mikro yang berbadan hukum koperasi, jadi BMT Darul Ihsan disebut juga dengan koperasi syariah. BMT Darul Ihsan melakukan kegiatannya dengan cara mudhorobah (bagi hasil), murobahah (saling menguntungkan), dan ijarah (kontrak).

Visi

Menjadi BMT yang mandiri, amanah, menentramkan, profesional dan unggul dalam memberdayakan ekonomi umat.

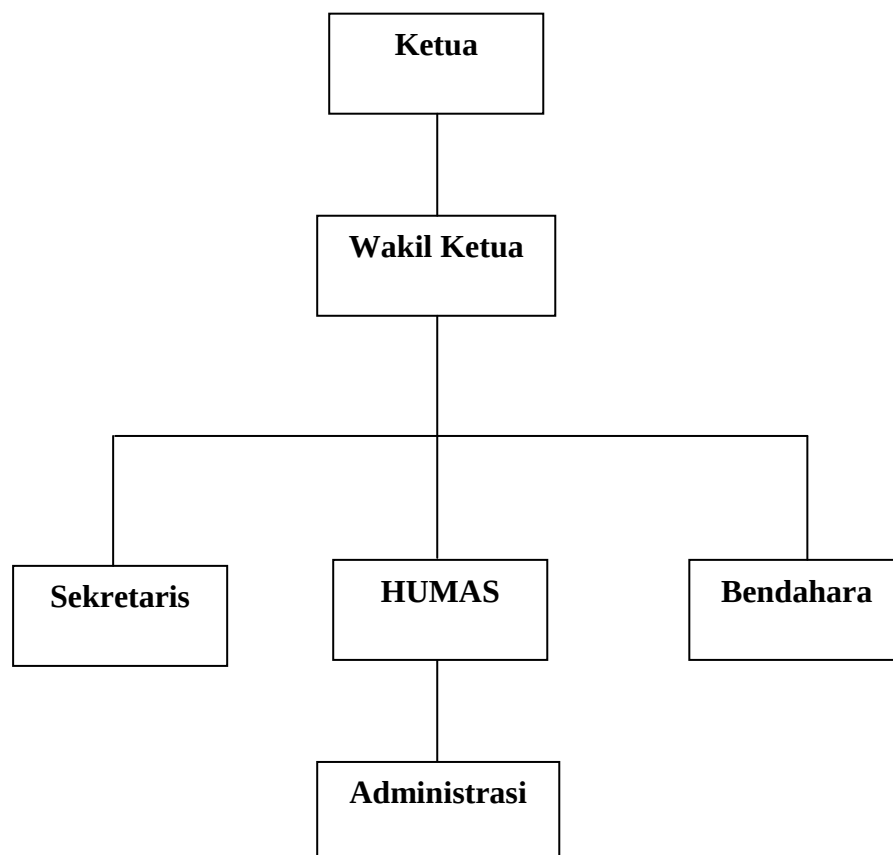
Misi

1. Menerapkan prinsip-prinsip syariah dalam kegiatan ekonomi, memberdayakan pengusaha kecil.
2. Menjalankan fungsi lembaga pembiayaan syariah yang profesional dan menguntungkan.
3. Berperan serta dalam gerakan merubah dari ekonomi ribawi ke ekonomi syariah.

II.8.1 Struktur Organisasi BMT Darul Ihsan

Organisasi adalah suatu badan terdiri dari sekelompok orang yang bekerja sama untuk mencapai tujuan yang telah digariskan sebelumnya. Struktur organisasi merupakan suatu gambaran secara sistematis tentang hubungan kerjasama dari orang-orang di dalamnya serta tanggungjawab personil.

Untuk lebih mengetahui struktur organisasi dari BMT Darul Ihsan Kabupaten Asahan, berikut ditunjukkan pada gambar II.14 :



Gambar II.14 : Struktur Organisasi BMT Darul Ihsan
(Sumber : BMT Darul Ihsan Kabupaten Asahan)

1. Ketua

Tugas dan tanggung jawabnya antara lain :

- 1) Mengendalikan seluruh kegiatan koperasi
- 2) Memimpin, mengkoordinir dan mengontrol jalannya aktivitas koperasi dan bagian-bagian yang ada didalamnya.
- 3) Menerima laporan atas kegiatan yang dikerjakan masing-masing bagian.
- 4) Mengambil keputusan atas hal-hal yang dianggap penting bagi kelancaran kegiatan koperasi.

2. Wakil Ketua

Tugas dan tanggung jawabnya antara lain :

- 1) Membantu Ketua dalam melaksanakan tugas sesuai dengan ketentuan yang telah ditetapkan.
- 2) Melaksanakn tugas-tugas lain yang diberikan oleh Ketua.
- 3) Mewakili Ketua apabila berhalangan dalam melaksanakan tugas sehari-hari.

3. Sekretaris

Tugas dan tanggung jawabnya antara lain :

- 1) Menyediakan fasilitas kelancaran pelaksanaan kegiatan sehari-hari.
- 2) Membantu Ketua dalam pelaksanaan kerja.
- 3) Menyelenggarakan kegiatan surat menyurat dan ketatausahaan koperasi.
- 4) Menyampaikan hal-hal yang penting kepada Ketua.

4. Bendahara

Tugas dan tanggung jawabnya antara lain :

- 1) Merencanakan anggaran belanja dan pendapatan koperasi.
- 2) Memelihara semua kekayaan koperasi.
- 3) Mempertanggung jawabkan dana dan hasil kegiatan lainnya kepada Ketua.

5. Humas

Tugas dan tanggung jawabnya antara lain :

- 1) Menyusun strategi dan kebijakan pengelolaan SDM di koperasi.
- 2) Mengkoordinasikan dan mengontrol pelaksanaan fungsi SDM di koperasi untuk memastikan semuanya sesuai dengan strategi kebijakan, sistem dan rencana kerja yang telah disusun.
- 3) Mengkoordinasikan dan mengontrol penyusunan dan pelaksanaan program pelatihan dan pengembangan untuk memastikan tercapainya target tingkat kemampuan dan kompetensi setiap karyawan.
- 4) Menyusun sistem manajemen kerja, serta mengkoordinasikan dan mengontrol pelaksanaan siklus manajemen kinerja.

6. Administrasi

Tugas dan tanggung jawabnya antara lain :

- 1) Mengatur surat menyurat yang ada dalam koperasi.
- 2) Mengarsipkan dokumen-dokumen penting koperasi.
- 3) Memonitor kebutuhan-kebutuhan ATK koperasi.
- 4) Mempersiapkan rapat-rapat di koperasi.
- 5) Menjadwalkan kegiatan-kegiatan dilakukan di koperasi.