BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah kumpulan elemen yang saling berkaitan dan bekerja sama dalam melakukan kegiatan untuk mencapai suatu tujuan. Pengertian sistem dilihat dari masukan dan keluarannya. Sistem adalah suatu rangkaian yang berfungsi menerima *input* (masukan), mengolah *input* dan menghasilkan *output* (keluaran). Sistem yang baik akan mampu bertahan dalam lingkungannya. Pengertian sistem dilihat dari prosedur/kegiatannya. Sistem adalah suatu rangkaian prosedur/kegiatan yang dilihat untuk melaksanakan program perusahaan (V.Wiratna Sujarweni ; 2015. 1-2).

Sistem ialah serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu (Anastasia Diana & Lilis Setiawati ; 2011. 3).

Dari kesimpulan diatas, Sistem adalah entitas atau satuan yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang saling terhubung dan terkait untuk mencapai suatu tujuan.

II.2. Informasi

Informasi adalah data yang berguna yang telah diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat. Informasi sangat penting bagi organisasi. Pada dasarnya informasi adalah penting seperti sumber daya yang lain, misalnya peralatan, bahan, tenaga, dsb.

Informasi yang berkualitas dapat mendukung keunggulan kompetitif suatu organisasi. Dalam sistem informasi akuntansi, kualitas dari informasi yang disediakan merupakan hal penting dalam kesuksesan sistem.

Secara konseptual seluruh sistem organisasional mencapai tujuannya melalui proses alokasi sumber daya, yang diwujudkan melalui proses pengambilan keputusan manajerial. Informasi memiliki nilai ekonomik pada saat ia mendukung keputusan alokasi sumber daya, sehingga dengan demikian mendukung sistem untuk mencapai tujuan.

Pemakai informasi akuntansi dapat dibagi dalam dua kelompok besar: ekstern dan intern. Pemakai ekstern mencakup pemegang saham, investor, kreditor, pemerintah, pelanggan, pemasok, pesaing, serikat pekerja, dan masyarakat. Pemakai intern terutama para manajer, kebutuhannya bervariasi tergantung pada tingkatannya (Agustinus ; 2012 : 1).

II.3. Sistem Informasi

Menurut Gelinas, Oram dan Wiggins (1990) Sistem Informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan dan mengelola data serta menyediakan informasi keluaran kepada para pemakai (Tata Sutabri:2010)

II.4. Suku Bunga

Bunga juga dapat diartikan sebagai harga yang harus dibayar kepada nasabah (yang memiliki simpanan) dengan yang harus dibayar oleh nasabah kepada bank (nasabah yang memperoleh pinjaman). Bunga bank dapat diartikan sebagai balas jasa yang diberikan oleh bank yang berdasarkan prinsip konvensional kepada nasabah yang membeli atau menjual produknya.

Ada dua macam bunga yang diberikan kepada nasabahnya yaitu :

1. Bunga Simpanan

   Bunga yang diberikan sebagai rangsangan atau balas jasa bagi nasabah yang menyimpn uangnya dibank. Contohnya adalah Jasa giro, Bunga Tabungan, Bunga Deposito

2. Bunga Pinjaman

   Bunga yang diberikan kepada para peminjam atau harga yang harus dibayar oleh nasabah peminjam kepada bank. Contohnya adalah Bunga kredit. (Ismail:2010)

II.4.1. Faktor Yang Mempengaruhi Suku Bunga

Faktor-faktor utama yang mempengaruhi besar kecilnya penetapan suku bunga adalah :

1. Kebutuhan Dana : Jika bank kekurangan dana, sementara permohonan pinjaman meningkat, maka yang dilakukan bank agar dana tersebut terpenuhi dengan meningkatkan suku bunga simpanan.

2. Persaingan : Dalam memperebutkan dana simpanan, maka disamping promosi yang paling utama pihak perbankan harus memperhatikan pesaing
3. Kebijaksanaan Pemerintah : Dalam arti bunga simpanan dan bunga pinjaman kita tidak boleh melebihi bunga yang sudah ditetapkan pemerintah.
4. Target Laba Yang diinginkan Jika ingin laba besar maka bunga pinjaman ikut besar dan sebaliknya.
5. Jangka Waktu Semakin panjang jangka waktu pinjaman akan semakin tinggi bunganya.
6. Kualitas Jaminan Semakin Likuit jaminan yang diberikan, semakin rendah bunga kredit yang dibebankan dan sebaliknya
7. Reputasi Perusahaan Perusahaan yang bonafid sangat menentukan suku bunga karena perusahaan tersebut resiko kredit macetnya kecil.
8. Produk yang kompetitif Maksudnya produk yang dibiayai laku dipasaran
9. Hubungan baik : Biasanya bank menggolongkan nasabahnya antara nasabah utama dan biasa. Ini didasarkan keaktifan serta loyalitas nasabah yg bersangkutan terhadap bank
10. Jaminan Pihak Ketiga : Dalam hal ini pihak yang memberikan jaminan kepada penerima kredit, biasanya pihak yang memberikan jaminan bonafid. (Ismail:2010)

II.5. Metode Flat Rate

Flat rate merupakan metode pembelajaran suku bunga kredit yang rata setiap kali angsuran, atau total angsuran pokok maupun angsuran bunga sama setiap kali angsuran atau setiap bulan. Metode falt rate sering digunakan bank Perkreditan

Rakyat/ atau beberapa lembaga pembiayaan. Kelebihan dari metode pembebanan bunga flat rate adalah cara perhitungan angsuran perbulan sangat sederhana dan mudah dimengerti, sehingga nasabah juga dapat melakukan perhitungan sendiri. Bunga Flat adalah sistem perhitungan suku bunga yang besarannya mengacu pada pokok hutang awal. Biasanya diterapkan untuk kredit barang konsumsi seperti handphone, home appliances, mobil atau kredit tanpa agunan (KTA). Dengan menggunakan sistem bunga flat ini maka porsi bunga dan pokok dalam angsuran bulanan akan tetap sama. Misalnya besarnya angsuran adalah satu juta rupiah dengan komposisi porsi pokok 750 ribu dan bunga 250 ribu. Maka, sejak angsuran pertama hingga terakhir porsinya akan tetap sama. (Ismail:2010)

II.5.1. Studi Kasus Metode Flat Rate

Pada tanggal 25 Maret 2006 PT. Andika Karya Tuan Andi mendapat persetujuan pinjaman investasi dari Bank ABC senilai Rp. 90.000.000,- untuk jangka waktu 1 tahun. Bunga yang dibebankan sebesar 24% pa.

Hitunglah cicilan setiap bulannya jika di hitung dengan metode Flat dan Sliding Rate ?

Jawaban Pembebanan Bunga dengan metode Flat Rate

a. Menghitung pokok pinjaman (pj) per bulan sebagai berikut :

Pokok Pinjaman yang harus dibayar setiap bulan adalah

PJ = jumlah pinjaman / jangka waktu

PJ = Rp. 90.000.000 / 12 bulan = Rp. 7.500.000

b. Selanjutnya menghitung bunga (BG) per tahun adalah

BG = bunga x nominal pinjaman x 1 / 12 bulan

BG = 24% x Rp. 90.000.000 x 1 / 12 bulan = Rp. 1.800.000

Jadi jumlah angsuran setiap bulan adalah

Pokok pinjaman Rp. 7.500.000

Bunga Rp. 1.800.000

Jumlah Angsuran Rp. 9.300.000.

Jumlah angsuran ini setiap bulan sama sampai dengan 12 bulan dan jika kita uraikan dalam bentuk tabel sebagai berikut :

Tabel II.1. Perhitungan Kredit Dengan Flat Rate

Bulan	Tgl. Angsuran	Angsuran	Pokok	Bunga	Saldo Akhir
1	07-05-2014	9,300,000	7,500,000	1,800,000	82,500,000
2	07-06-2014	9,300,000	7,500,000	1,800,000	75,000,000
3	07-07-2014	9,300,000	7,500,000	1,800,000	67,500,000
4	07-08-2014	9,300,000	7,500,000	1,800,000	60,000,000
5	07-09-2014	9,300,000	7,500,000	1,800,000	52,500,000
6	07-10-2014	9,300,000	7,500,000	1,800,000	45,000,000
7	07-11-2014	9,300,000	7,500,000	1,800,000	37,500,000
8	07-12-2014	9,300,000	7,500,000	1,800,000	30,000,000
9	07-01-2015	9,300,000	7,500,000	1,800,000	22,500,000
10	07-02-2015	9,300,000	7,500,000	1,800,000	15,000,000
11	07-03-2015	9,300,000	7,500,000	1,800,000	7,500,000
12	07-04-2015	9,300,000	7,500,000	1,800,000	0
Jumlah		**111,600,000**	**90,000,000**	**21,600,000**	

II.6. Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data 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 (*www. utexas. edu*).

1. Bentuk Nornal Pertama (1 NF)

Contoh yang kita gunakan di sini adalah sebuah perancangan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat dituliskan sebagai berikut :

PEMASOK (P#, Status, Kota, b#, qty) di mana

p# : kode pemasok (kunci utama)

status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok.

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomi. Ini berarti kolom-kolom tidak mempunyai nilai berulang. Tabel II.2. menunjukkan tabel pemasok dalam 1 NF.

Tabel II.2. Normalisasi Pertama Pemasok

P#	Status	Kota	B#	Qty
P1	20	Yogyakarta	B1	300
P1	20	Yogyakarta	B2	200
P1	20	Yogyakarta	B3	400
P1	20	Yogyakarta	B4	200
P1	20	Yogyakarta	B5	100
P1	20	Yogyakarta	B6	100

P2	10	Medan	B1	300
P2	10	Medan	B2	400
P3	10	Medan	B2	200
P4	20	Yogyakarta	B2	200
P4	20	Yogyakarta	B4	300
P4	20	Yogyakarta	B5	400

(Sumber : Janner Simarmata, dkk, 2010).

2. Bentuk Normal Kedua (2 NF).

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1 NF, tetapi tidak pada 2 NF, sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1 NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Tabel pemasok berada pada 1 NF, tetapi tidak pada 2 NF karena status dan kota tergantung secara fungsional hanya pada kolom p# dari kunci gabungan (p#, b#). Ini dapat digambarkan dengan membuat daftar ketergantungan fungsional.

P# ⟶ Kota, Status

Kota ⟶ Status

(P#, B#) ⟶ qty

Proses mengubah tabel 1 NF ke 2 NF adalah :

a. Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukan.

c. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru penentu akan menjadi kunci utama pada tabel baru.

d. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.

e. Tabel asal bisa diberi nama baru.

Pada contoh, kita memindahkan kolom p#, status, dan kota ke tabel baru yang disebut pemasok2. Kolom p# menjadi kunci utama tabel ini. Tabel II.3. menunjukkan hasilnya.

Tabel II.3. Tabel Bentuk Normal Kedua (2NF).

Pemasok2

P#	Status	Kota
P1	20	Yogyakarta
P2	10	Medan
P3	10	Medan
P4	20	Yogyakarta
P5	30	Bandung

Barang

P#	B#	Qty
P1	B1	300
P1	B2	200
P1	B3	400
P1	B4	200
P1	B5	100
P1	B6	100
P2	B1	300
P2	B2	400
P3	B2	200
P4	B2	200
P4	B4	300
P4	B5	400

(Sumber :Janner Simarmata, dkk, 2010).

3. Bentuk Normal Ketiga (3 NF).

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3 NF) jika tabel sudah berada pada 2 NF dan setiap kolom yang bukan kunci tidak tergantung secara transistif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

Tabel barang sudah dalam bentuk normal ketiga. Kolom bukan kunci, qty, tergantung sepenuhnya pada kunci utama (p#, b#). Pemasok masih berada pada 2 NF, tetapi belum berada pada 3 NF karena dia mengandung ketergantungan transitif. Ketergantungan transitif terjadi ketika sebuah kolom bukan kunci, yang ditentukan oleh kunci utama, menentukan kolom lainnya. Konsep ketergantungan transistif daopat digambarkan dengan menunjukkan ketergantungan fungsional pada pemasok2, yaitu :

Pemasok2. p# ⟶ Pemasok2, status

Pemasok2. p# ⟶ Pemasok2, kota

Pemasok2. kota ⟶ Pemasok2, status

Perlu dicatat bahwa pemasok2, status ditentukan, baik oleh kunci utama p#, maupun kolom bukan kunci, kota

Proses mengubah tabel menjadi 3 NF adalah :

a. Tentukan semua penentu selain kunci utama dan kolom yang ditentukannya.
b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom yang ditentukannya.
c. Pindahkan kolom yang ditentukan dari tabel asal ke tabel baru. Penentu menjadi kunci utama tabel baru.
d. Hapus kolom yang baru saja dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
e. Tabel asal bisa diberi nama baru.

Untuk mengubah PEMASOK2 menjadi 3 NF, kita membuat tabel baru yang disebut KOTA_STATUS dan memindahkan kolom kota dan status ke tabel baru. Status dihapus dari tabel diberi nama baru PEMASOK_KOTA. Tabel II.4 menunjukkan hasilnya.

Tabel II.4. Tabel Bentuk Normal Ketiga (3 NF)

PEMASOK_KOTA

P#	Kota
P1	Yogyakarta
P2	Medan
P3	Medan
P4	Yogyakarta
P5	Bandung

KOTA_STATUS

Kota	Status
Yogyakarta	20
Medan	10
Bandung	30
Semarang	40

(Sumber : Janner Simarmata, dkk, 2010).

4. Bentuk Normal Boyce Code (BCNF)

Setelah 3 NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3 NF sudah cukup karena sangat jarang entitas yang berada pada 3 NF bukan merupakan 4 NF dan 5 NF. Lebih lanjut, mereka berpendapat bahwa keuntungan yang didapat mengubah entitas ke 4 NF dan 5 NF sangat kecil sehingga tidak perlu dikerjakan. Bentuk Normal Boyce- Code (BCNF) adalah versi 3 NF lebih teliti dan berhubungan dengan tabel relasional yang mempunyai (a) banyak kunci kandidat (b) kunci kandidat gabungan, dan (c) kunci kandidat yang saling tumpang tindih.

BCNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom di mana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel

relasional berada pada BCNF jika dan hanya setiap penentu adalah kunci kandidat.

5. Bentuk Normal Keempat (4 NF)

Sebuah tabel relasional berada pada bentuk normal keempat (4 NF) jika dia dalam BCNF dan semua ketergantungan multivalue merupakan ketergantungan fungsional.

Bentuk normal keempat (4 NF) didasarkan pada konsep ketergantungan multivalue (MVD). Sebuah ketergantungan multivalue terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berebeda.

Defenisi secara formal diberikan oleh CJ. Date, yaitu :

Misalnya, ada sebuah tabel relasional R dengan kolom A, B dan C, Maka R.A R.B $\longrightarrow$ (kolom A menentukan kolom B).

Adalah benar jika dan hanya jika himpunan nilai B yang cocok dengan pasangan nilai A dan nilai C pada R hanya tergantung pada nilai A dan tidak tergantung pada nilai C.

MVD selalu terjadi dalam pasangan, yaitu R.A $\longrightarrow$ R.B dipenuhi jika dan hanya jika R.A $\longrightarrow$ R.C dipenuhi pula.

6. Bentuk Normal Kelima (5 NF).

Sebuah tabel berada pada bentuk normal kelima jika dia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil.

Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep

ketergantungan gabungan (*join dependence*). Ketergantungan gabungan berarti sebuah tabel, setelah deskomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain 5 NF menunjukkan ketika sebuah tabel tidak dapat dideskomposisi lagi (Janner Simarmata, 2012).

II.7. Basis Data (*Database*)

Secara sederhana database (basis data/ pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat (Kadir, 2004). Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *hard disk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk database.

Pengaplikasian database dapat kita lihat dan rasakan dalam keseharian kita. Database ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (anjungan tunai mandiri/ *automatic teller machine*) bank karena bank telah mempunyai database tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks database sebenarnya kita sudah melakukan perubahan (*update*) data pada database di bank.

Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep database. Data yang kita simpan di HP juga mempunyai struktur yang diisi melalui formulir (*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya.

Pemahaman tentang database ini dapat didekatkan pada konsep akuntansi. Kita bisa umpamakan bahwa ketika kita melakukan proses akuntansi secara manual, kita menuliskan suatu catatan ke dalam lajur dan kolom buku. Mulai dari jurnal, buku besar, buku pembantu kita memasukkan catatan satu demi satu. Melihat buku akuntansi tersebut, sebenarnya kita sudah melihat konsep database, yang jika dikelola dengan komputer masih diperlukan penyesuaian dalam membentuk kolom-kolomnya. (Mujilan, 2012)

II.7.1. Model Database

Model database yang saat ini banyak digunakan adalah model database relational. Imam (2008) menyebutkan "Model database ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*). Tabel-tabel yang ada dihubungkan (*relationship*) sedemikian rupa menggunakan *field-field* kunci (*key field*) sehingga dapat meminimalkan duplikasi data."

Model database relational ini dapat kita kenal konsepnya mulai dari yang paling sederhana misalnya dengan penerapan program aplikasi *excel*. Meskipun untuk pengelolaan database secara luas *excel* jarang digunakan dan kurang mencukupi, namun untuk melihat konsep database dan konsep membangunnya

program ini dapat dimanfaatkan. *Excel* mempunyai baris yang disebut *raw* dan mempunyai kolom. Kemudian item data merupakan sel atau

pertemuan antara baris dan kolom. Tabel-tabel dapat diumpakan apabila kita menggunakan tabel dalam suatu *sheet* tertentu. Data dari berbagai tabel dapat diambil dari tabel lain menggunakan perintah *look up* yang berdasarkan kode kunci tertentu. Kode kunci tersebut berada pada suatu kolom tertentu, yang dalam konsep database relational disebut sebagai *key field* tadi.

II.7.2. Struktur Database

Untuk memahami konteks database kita perlu memahami istilah dan hal-hal yang terkait dengan database. Dalam berbagai program aplikasi database terdapat kesamaan ataupun sedikit perbedaan di dalamnya. Seseorang yang mempelajari database dengan program aplikasi tertentu harus memperhatikan struktur dan karakteristik sesuai dengan bahasa dalam aplikasi tersebut. Namun demikian, secara umum terdapat karakteristik sebagai berikut:

1. Nama *file*

Nama *file* adalah nama yang digunakan untuk mengidentifikasi adanya data yang disimpan dalam komputer dan digunakan untuk pemanggilan data. *File* yang dikelola akan muncul dalam komputer dengan ekstensi sesuai dengan program aplikasinya. *File* tersebut dapat digunakan untuk menandakan adanya *file* database, ataupun *file table*. Database dan table akan saling terkait, meskipun cara menyimpan dalam komputer akan mengalami sedikit perbedaan pada beberapa aplikasi. Misalnya Ms. *Access* akan menyimpan dengan *file* yang dapat kita lihat adalah *file* databasenya. Di program *MySql* nama database ini akan menjadi

folder. Sementara di *FoxPro* nama database dapat menjadi *file* tersendiri. *Table* cara menyimpannya juga berbeda, dalam *Ms. Access* mungkin kita tidak melihat nama *table* secara kasat mata karena akan dikelola di dalam *file* database. Di dalam *MySql* kita bisa melihat beberapa nama *file* terkait dengan pengelolaan *table*. Dan di dalam *FoxPro table* ini dapat menjadi nama *file* terpisah dan dapat dikenali pula sebagai *free table*.

2. Database

Database sebenarnya merupakan nama untuk menampung berbagai *table* di dalamnya. Konsep ini akan sama dalam berbagai program aplikasi. Misalnya kita membangun database akuntansi dengan nama database "*akun_base*". Di dalam *akun_base* akan diorganisasi berbagai table yang terkait dengan kegiatan akuntansi misalnya tabel: rekening, pelanggan, jurnal, buku induk, dan administrator program, dan sebagainya. Setiap data yang masuk tidaklah dicatat dalam database, namun di dalam masing-masing *table* yang sesuai.

3. *Table*

Table merupakan tempat untuk menyimpan data sesuai dengan kelompok data. Setiap isi table mengandung data yang mempunyai karakteristik dalam penggunaannya. Untuk mempermudah pengolahan biasanya pembangun database mengkategorikan *table* sesuai dengan data isinya sebagai berikut :

a. *Master table*

Master table berisi data tentang hal-hal utama dalam kegiatan database. Table ini berisi record yang relatif permanen atau seringkali menjadi acuan ketika mengoperasikan transaksi. Dalam master tabel identitas record menjadi

penting dan diusahakan merupakan data atau kode yang bersifat unik. Unik dapat diartikan bahwa tidak ada dalam satu table berisi kode yang sama. Disain kode menjadi penting di sini. Misalnya dalam mendisain nama akun dalam database akuntansi, maka kode akun menjadi sangat penting artinya. Dalam *table* berisi nama barang, maka kode barang menjadi hal penting. Contoh lain dalam database akademik, tabel *master* dapat berupa : mahasiswa, daftar dosen, daftar kurikulum.

b. *Transaction table*

Tabel transaksi digunakan untuk menyimpan data dalam menjalankan suatu kegiatan atau bisnis. Data ini seringkali akan bertambah dalam kesehariannya ketika terjadi transaksi yang sesuai dengannya. Secara lebih mudah dapat dipahami dalam akuntansi seringkali mencatat transaksi dalam jurnal. Terkait hal tersebut, transaksi ini dicatat dalam tabel jurnal. Dalam mencatat transaksi ini, kita harus menyesuaikan kode data tertentu dengan kode yang terdapat dalam *master table.*

c. *Tabulation table*

Tabulasi data dapat digunakan untuk menyimpan data seperti halnya master data namun bersifat sebagai data pembantu ketika menginput formulir baik untuk data master maupun transaksi. Misalnya untuk memetakan keterangan hobi, jenis kelamin, nama golongan, nama level manajemen, dan sebagainya. Dengan konsep penamaan field yang baik mungkin saja table tabulasi ini dapat digunakan untuk memuat berbagai kelompok data. Misalnya fieldnya berupa kode dan keterangan. Contoh kelompok gender

dengan L = laki-laki; P = perempuan. Kelompok level dengan M = Manajer, O = operator, S = seller

d. *Temporary table*

Temporary adalah data sementara yang digunakan untuk membantu ketika terjadi proses transaksi. Data ini dapat saja langsung dihapus ketika transaksi selesai terproses. Misalnya digunakan untuk mempermudah perhitungan, penyimpanan data sementara sebelum diproses setuju ke database. Misalnya: ketika terjadi transaksi di depan kasir, data-data pertama akan ditangkap dan dimasukan dalam file temporary sebelum akhirnya kasir melakukan perintah "ok" yang menandakan data transaksi siap untuk disimpan atau diproses dalam komputer. Ketika masa tunggu ini, data masih dapat diedit, dibatalkan, ataupun ditambah. Sementara ketika sudah masuk ke sistem, edit atau penambahan akan membutuhkan prosedur tertentu. Seandainya dianalogikan dengan sistem akuntansi maka proses edit data yang telah masuk ke sistem dapat digunakan prosedur seperti halnya melakukan jurnal koreksi.

4. *Field*

Field adalah penanda untuk kolom data. Jika dalam *excel* penanda tersebut adalah kolom A, B, dan seterusnya, sementara dalam konsep *table* dalam database maka nama *field* memegang peranan penting. Dalam konsep table dalam database, ketika memanggil dengan nama *field* tertentu maka data-data di dalamnya akan muncul. Pengolahan dapat dilakukan dengan membuat *filter*, misalnya berdasarkan kode tertentu, berdasarkan *record* tertentu. Misalnya kita ingin

memanggil record terkait nama karyawan Andi. Dalam database Andi ini diberi ID: 11001. Sehingga kita bisa menggunakan konsep filtrasi untuk memanggil personalia dengan kode ID 11001. Apabila data ketemu, maka kita dapat menggunakan data berdasarkan *field-field* yang ada, misalnya nama, tempat lahir, tanggal lahir, alamat, dan sebagainya yang mengacu pada ID 11001.

Dalam mengatur setting *field,* biasanya akan terkait hal-hal sebagai berikut:

a. *Field type* : tipe *field* ini dapat terkait apakah *field* tersebut akan berisi data berupa *key field* (*primary key*, *secondary key*), atau *descriptor*. *Primary key* akan berisi ID atau kode pokok yang akan digunakan dalam mengidentifikasi *record*, sehingga data di dalam *field* tersebut tidak diijinkan untuk memiliki lebih dari satu data yang sama. *Secondary* adalah subset dari *key* utama. Misalnya saja kode mata kuliah dalam satu semester tidak boleh terdapat lebih dari satu pada ID atau NIM yang sama. *Descriptor* adalah *field* berisi data yang akan merupakan satu kesatuan dengan yang lainnya sebagai penjelasan akan adanya *record* atau ID tertentu.

b. Data *type*: tipe data merupakan jenis data yang dapat dimasukkan dalam *field*. Hal ini dapat dibagi secara umum sebagai karakter/ *text*, numerik, tanggal dan sebagainya.

c. *Field Size*. Penting untuk memahami ukuran field yang akan digunakan dalam menampung data. Dalam pengembangan sistem harus dapat memperkirakan berapa lebar ukuran *field* yang efektif. Apabila terlalu lebar akan terjadi banyak spasi kosong dan berpengaruh pada ukuran *file*

yang disimpan. Sementara apabila terlalu sempit akan terdapat data yang tidak tersimpan. Misalnya ketika kita menghitung bahwa nama

menggunakan ukuran 40 karakter sudah memenuhi untuk *field* data kita. Konsekuensinya, apabila terdapat nama di atas 40 karakter maka akan terpotong menjadi 40 karakter. Konsekuensi lain adalah nama tersebut diinput hingga memuat maksimal 40 karakter yaitu dengan mengadakan singkatan nama.

5. *Records*

Records merupakan baris data. Karena satu baris data biasanya mengindikasikan satu kesatuan data tertentu, maka satu *record* ada yang menyebut satu data. Misalnya keterangan mengenai biodata Andi disimpan dalam satu record beridentitas ID 11001, maka untuk menyebutkan satu kesatuan data seputar Andi dalam baris tertentu ada yang menyebutkan sebagai *record* data Andi. Dalam konsep database masing-masing *record* memiliki nomor identitas tersendiri baik itu identitas yang diberikan komputer

ataupun yang diinputkan secara manual. Sehingga dalam konteks tertentu dapat digunakan konsep nomor *record*, ID otomatis, ID *primary key*. (Mujilan, 2012)

II.8. *UML* (*Unified Modelling Language*)

Unified Modelling Language (UML) menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifak dalam UML adalah informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasikan oleh

pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan desain adalah bahwa setiap perubahan terjadi pada suatu artifak yang harus juga dilakukan pada artifak lainnya. (Sumber: Evi Triandini Dan I gede Suardika, 2012)

II.8.1. Diagram-*Diagram UML*

1. *Diagram Use Case (Use Case Diagram)*

Pengertian use case menurut Jhon Satzinger, 2010 dalam buku system analysis and design in a changing world menyatakan bahwa "*use case* adalah sebuah kegiatan yang dilakukan oleh system, biasanya dalam menanggapi permintaan dari pengguna system. (Sumber: Evi Triandini Dan I gede Suardika, 2012)

Notasi Use Case Diagram dapat dilihat pada gambar berikut ini :

Gambar II.1. Notasi Use Case Diagram
(sumber : Evi Triandini dan I Gede Suardika:2012)

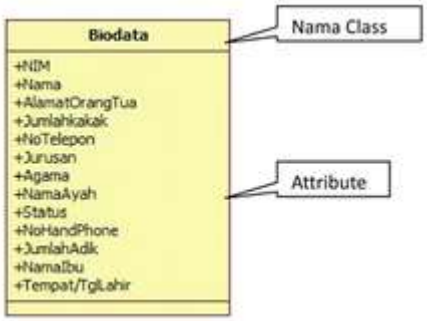
2. Diagram Kelas (*Class Diagram*)

Menurut John Satzinger, dalam buku system analysis and design in a changing world menyatakan bahwa "dalam uml ada dua jenis clas doiagram yaitu: domain class diagram dan design class diagram

1. Domain class diagram

Focus domain class diagram adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada class perangkat lunak yang nantinya akan dirancang.

Berikut uni adalah gambar dari domain class diagram :



Gambar II.2. domain class diagram
(Sumber : Evi Triandini dan I Gede Suardika:2012)

2. Design class diagram

Tujuan utamanya adalah untguk mendokumnetasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun. Design class diagram menggambarkan kelas berorientasi objek yang dibutuhkan dalam pemrograman, navigasi diantara kelas, atribut names dan propertinya serta method names dan propertynya.

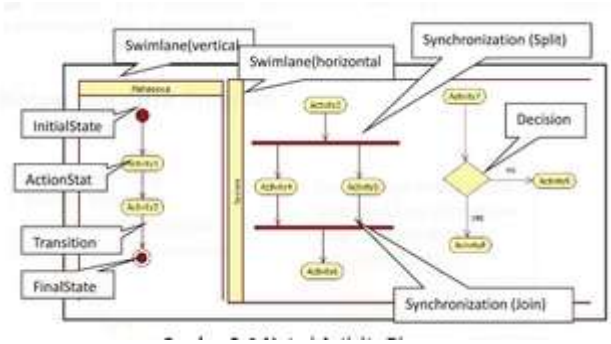
Gambar dari calss diagram dapat dilhat pada gambar di bawah ini :

Gambar II.3. calss diagram
(Sumber : Evi Triandini dan I Gede Suardika:2012)

3. Diagram Aktivitas (*Activity Diagram*)

Menurut John Satzinger, dalam buku system analysis and design in a changing world menyatakan bahwa" activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau pun sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

Notasi activity diagram adalah sebagai berikut :



Gambar II.4. Notasi activity diagram
(Sumber : Evi Triandini dan I Gede Suardika:2012)

Penjelasan dari gambar diatas adalah sebagai berikut :

Tabel II.6. Keterangan Notasi activity diagram

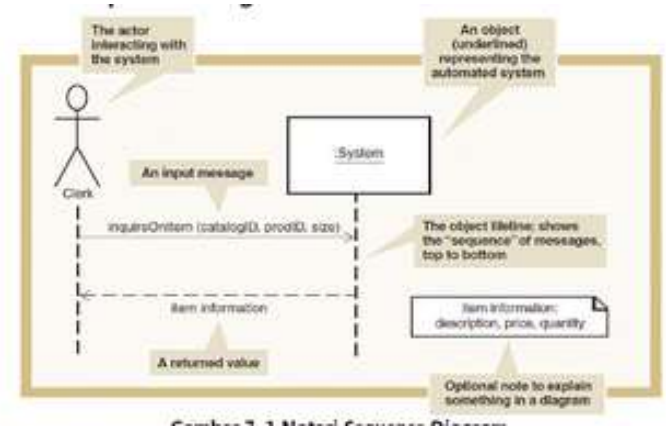
Notasi	Keterangan
Swimlane	Mewakili agen yang melakukan aktivitas karena dalam alur kerja umumnya mempunyai agen yang berbeda dengan yang melakukan langkah yang berbeda dari proses alur kerja.
Initialstate	Awal dari alur kerja
ActionState	Melambangkanaktivitas tersendiri dalam alur kerja
Transition	Melambangkan dari urutan diantara aktivitas
Final State	Akhir dari alur kerja
Sychronization	Membagi alur kerja menjadi bebrapa alur yang berbarengan ataupun menggabungkan lagi alur yang berbarengan
Decision	Titik pengambilan keputusan dimana aliran proses tersebut akan mengikuti satu jalur atau jalur lainnya

(Sumber : Evi Triandini dan I Gede Suardika)

4. *Sequence Diagram*

Menurut John Satzinger, dalam buku system analysis and design in a changing world menyatakan bahwa" System Sequence Digram (SSD) adalah diagram output serta urutan interaksi antara pengguna dan system untuk sebuah use case.

Berikut ini adalah gambar dari sequence diagram :



Gambar II.5. sequence diagram
(Sumber : Evi Triandini dan I Gede Suardika:2012)

Penjelasan dari gambar diatas adalah sebagai berikut :

a. Actor, mewakili seprang aktor (orang atau peran yang berinteraksi dengan sistem)
b. Kontak berlabel, system adalah objek yang mwakili keseluruhan sistem yang terotomatisasi
c. Garis putus-putus vertikal (lifelines) adalah perpanjangan objek tersebut, baik aktor maupun objek, sepanjang durasi dari sequence diagram
d. Anak panah antara lifeline mewakili message yang dikirim atau diterima oleh aktor dari sistem
e. Message diberi label untuk menggambarkan maksud message dan input apa pun yang sedang dikirim. Message dipertimbangkan sebagai sebuah inti yang diminta pada tujuan objek, kebanyakan seperti perintah.

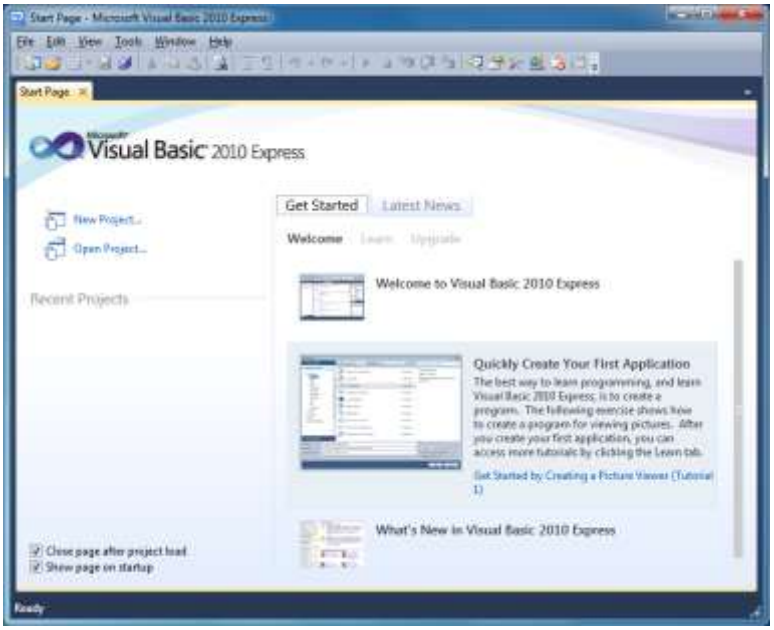
II.9. Bahasa Pemograman *Microsoft Visual Studio 2010*

Visual Basic 2010 adalah inkarnasi dari bahasa visual baasic yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat lainnya sseperti C++. Anda dapat menggunakan visual basic 2010 untuk membuat aplikasi windows, mobile, web, dan office atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke program lainnya. Visual basic menyediakan berbagai tools dan fitur canggih yang memungkinkan dapat menulis kode, menguji dan menjalankan program tunggal ataau terkadang serangkaian program yang terkait dengan satu aplikasi.

Tabel II.7. Jendela Aplikasi Visual Studio 2010

No	Bagian	Keterangan
1	Title Bar	Menampilkan nama aplikasi yang sedang terbuka.
2	Mneu Bar	Menampilkan daftar perintah yang memungkinkan anda dapat menulis, mengedit, menyimpan, mencetak, menguji dan menjalankan program visual basic.
3	Standard Toollbar	Berisi Tombol yang menjalankan perintah yang sering digunakan sepertin open project, new project, save, cut, copy, paste, dan undo
4	Toolbox	Berisi komponen NET yang dapat anda gunakan untuk mengembangkan antarmuka pengguna grafis untuk program visual studio 2010.
5	Area Kerja Utama	Menampilkan item yang sedang di kerjakan
6	Solution Explorer	Menampilkan elemen dari visual bassic solution, yaitu nama yang ddiberikan kepada program visual bassic dan item lainnya yang dihasilkan oleh visual basic 2010 sehingga program akan mengeksekusi dengan benar.
7	Properties Window	Setiap Objek dalm program visual basic memiliki seperangkat karakteristik yang disebut sifat-sifat objek.

Untuk melihat tampilan visual basic 2010 dapat dilihat pada gambar II.2. sebagai berikut :



Gambar II.6. Tampilan Utama Visual Basic 2010
Sumber : (Christopher Lee:2014)

II.10. *SQL Server 2008*

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang di-develop oleh Microsoft, yang digunakan untuk menyimpan dan mengolah data. Pada *SQL Server 2008*, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada *SQL Server 2008*, kita bisa membuat objek-objek yang sering digunakan pada aplikasi bisnis, seperti membuat database, *table*, *function*, *stored database*, *trigger* dan *view*. Selain objek, kita juga menjalankan perintah Sql (*Structured Query Language*) untuk mengambil data. (Elex Media Competindo, 2010,101)