

BAB III

LANDASAN TEORI

3.1 Transfer Bank

Transfer adalah pemindahan dana antar rekening di suatu tempat ke tempat yang lain, baik untuk kepentingan nasabah atau untuk kepentingan bank itu sendiri.

Pihak-pihak yang terkait dengan transaksi transfer:

- a. *Remitter/Applicant*, yaitu pemilik dana (pengirim) yang akan memindahkan dananya melalui jasa pengiriman uang.
- b. *Beneficiary*, yaitu pihak akhir yang akan berhak menerima dana transfer dari bank.
- c. *Remitting Bank* merupakan bank pelaku transfer atau bank yang menerima amanat dari nasabah untuk di transfer kepada penerima dana.
- d. *Paying Bank* adalah bank yang menerima transfer masuk untuk diteruskan kepada penerima.

Jenis transfer ada dua, yaitu transfer masuk dan transfer keluar. Transfer masuk adalah pengiriman uang yang diterima dari cabang lain bank untuk keuntungan penerima dana. Sedangkan transfer keluar adalah pengiriman uang atas perintah nasabah untuk keuntungan pihak lain yang berupa bank atau nasabah lain.

Berdasarkan undang-undang dasar Republik Indonesia nomor 3 tahun 2011 tentang transfer dana terbagi menjadi dua cara, yaitu transfer dalam satu bank dan transfer antar bank.

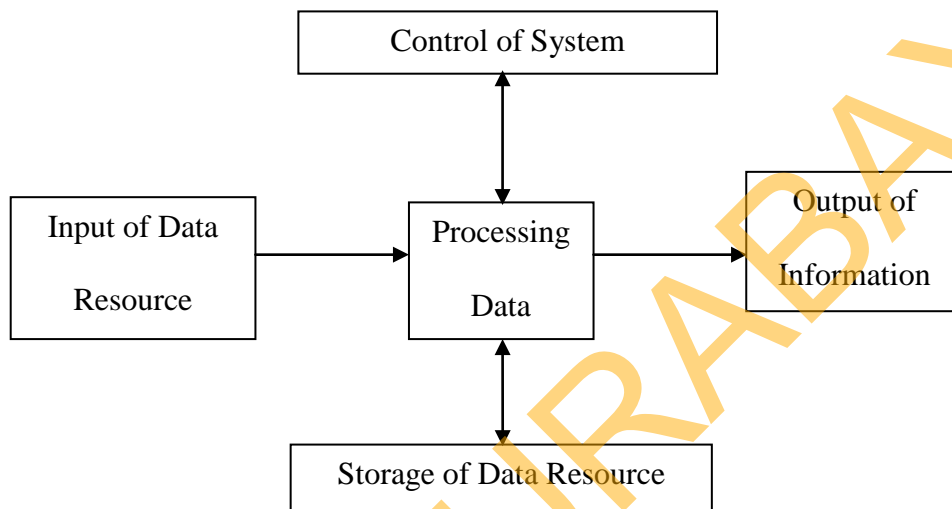
- a. Transfer dalam satu bank adalah transfer dana yang dikirim dan diterima dalam satu bank.
- b. Transfer antar bank merupakan transfer dana yang dilakukan dengan tujuan bank yang berbeda dengan bank asal pengiriman. Dalam melakukan transfer antar bank terdapat dua cara, yaitu :
 1. RTGS adalah kiriman/transfer ke Bank lain dari nominal Rp. 1,- s/d tak terbatas , yang waktu kirimnya sampai hari itu juga. Biayanya sekitar Rp. 20.000,- hingga Rp. 50.000,- tergantung Bank pengiriman. Waktu pengiriman dapat dilakukan mulai jam buka Bank hingga pukul 14.00 atau pukul 15.00.
 2. KLIRING LLG KREDIT adalah transfer dana ke Bank lain dari nominal Rp. 1,- s/d < Rp. 100.000.000,- , yang waktu kirimnya sekitar 1 hari hingga seminggu. Biaya pengiriman sekitar Rp. 5.000,- hingga Rp. 15.000,- tergantung Bank pengirim. Waktu pengiriman dapat dimulai jam buka Bank hingga tutup Layanan LLG.

3.2 Sistem Informasi

Suatu sistem mempunyai tujuan atau sasaran. Tujuan biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit. Sasaran menentukan masukan dan keluaran yang dihasilkan. Sistem dikatakan berhasil jika mencapai sasaran dan tujuan.

Sistem informasi terdiri dari *input*, proses, dan *output*. Pada proses terdapat hubungan timbal balik dengan 2 elemen, yaitu kontrol kinerja sistem dan sumber-sumber penyimpanan data, baik berupa karakter-karakter huruf maupun berupa numerik. Saat ini data bisa berupa suara atau *audio* maupun gambar atau *video*.

Data ini diproses dengan metode-metode tertentu dan akan menghasilkan *output* yang berupa informasi. Informasi yang dihasilkan dapat berupa laporan atau *report* maupun solusi dari proses yang telah dijalankan.



Gambar 3.1. Proses Sistem Informasi
(Sumber: Herlambang dan Tanuwijaya, 2005)

3.3 Perancangan Sistem

Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut (Lucas, 1993).

Tahapan perancangan memiliki tujuan mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik. Kegiatan yang dilakukan dalam tahap perancangan ini meliputi perancangan *input*, *output*, dan *file*.

Berikut ini adalah proses dalam perancangan sistem yang terbagi menjadi dua, yaitu:

3.3.1 *Relationship Diagram*

Bagi perancang basis data, *Entity Relationship Diagram* (ERD) berguna untuk memodelkan sistem yang nantinya akan dikembangkan basis datanya. Model ini juga membantu perancangan basis data pada saat melakukan analisis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data di dalamnya. Bagi pemakai, model ini sangat membantu dalam hal pemahaman model sistem dan rancangan basis data yang akan dikembangkan oleh perancang basis data (Sutanta, 2004). Sebuah ERD tersusun atas 3 komponen yaitu:

a. Entitas (*entity*).

Entitas menunjukkan obyek-obyek dasar yang terkait di dalam sistem. Obyek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan di dalam basis data.

b. Atribut.

Atribut sering disebut sebagai properti yang merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut berfungsi sebagai penjelas sebuah entitas.

c. Kerelasian antar entitas.

Kerelasian antar entitas mendefinisikan hubungan antar dua buah entitas. Kerelasian antar entitas dapat dikelompokkan dalam tiga jenis yaitu kerelasian jenis satu ke satu, kerelasian jenis banyak ke satu dan kerelasian jenis banyak ke banyak.



3.3.2 Data Flow Diagram

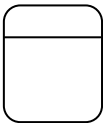
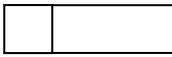
Data Flow Diagram (DFD) digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data itu mengalir (misalnya lewat telepon, surat) atau lingkungan fisik dimana data itu akan disimpan (misalnya file kartu, tape, disket). DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. DFD juga merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas (Kendall dan Kendall, 2005:243).

3.3.3 Simbol DFD

Ada empat simbol pokok di dalam menggambar suatu DFD dapat dilihat pada Tabel 3.1, dengan menggunakan simbol Gane & Sarson (Kendall dan Kendall, 2005:243).

Tabel 3.2 Simbol pokok DFD

No	Simbol	Nama	Keterangan
1		Entitas luar	Merupakan entitas di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan masukan atau menerima keluaran dari sistem.
2		Arus Data	Arus data mengalir di antara proses, simpanan data dan entitas luar. Arus data menunjukkan arus dari data yang dapat

No	Simbol	Nama	Keterangan
			berupa masukan untuk sistem atau hasil dari proses sistem.
3		Proses	Merupakan kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Nama suatu proses biasanya berbentuk suatu kalimat diawali dengan kata kerja.
4		Simpanan Data	Merupakan simpanan data yang dapat berupa suatu file atau basis data di sistem komputer, suatu arsip atau catatan manual, suatu agenda atau buku.

a. Bentuk DFD

Terdapat dua bentuk DFD yaitu *Physical Data flow Diagram* (PDFD) dan *Logical Data Flow Diagram* (LDFD) (Kendall dan Kendall, 2005:251). PDFD lebih menekankan pada bagaimana proses dari sistem diterapkan sedang LDFD lebih menekankan pada proses apa yang terdapat di sistem.

PDFD lebih tepat digunakan untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari PDFD adalah bagaimana proses–proses dari sistem diterapkan (dengan cara apa, oleh siapa dan di mana) termasuk proses manual.

LDFD lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). LDFD tidak menekankan pada bagaimana sistem diterapkan, tetapi penekanannya hanya pada logika dari kebutuhan sistem yaitu proses apa secara logika yang dibutuhkan oleh sistem yang biasanya proses yang digambarkan hanya merupakan proses secara komputer saja.

b. Pembuatan DFD

Untuk memulai membuat DFD dari suatu sistem daftarkan semua komponen yang terlibat (entitas luar, proses, arus data dan simpanan data). Setelah semua teridentifikasi maka dilanjutkan dengan melakukan langkah berikut (Kendall dan Kendall, 2005:245):

a. Pembuatan *context diagram*

Context diagram adalah level tertinggi dalam sebuah DFD dan hanya berisi satu proses yang merupakan representasi dari suatu sistem. Proses dimulai dengan penomoran ke-0 dan tidak berisi simpanan data.

b. Pembuatan diagram level 0

Diagram level 0 merupakan hasil pemecahan dari *Context diagram* menjadi bagian yang lebih terinci yang terdiri dari beberapa proses. Sebaiknya jumlah proses pada level ini maksimal 9 proses untuk menghindari diagram yang sulit untuk dimengerti. Setiap proses diberikan penomoran dengan sebuah bentuk *integer*. Simpanan data mulai ditampilkan pada level ini.

c. Pembuatan *child diagram*

Setiap proses pada diagram level 0 dipecah lagi agar didapat level yang lebih terinci lagi (*child diagram*). Proses pada level 0 yang dipecah lebih terinci lagi disebut *parent process*. *Child diagram* tidak menghasilkan keluaran atau

menerima masukan yang mana *parent process* juga tidak menghasilkan keluaran atau menerima masukan. Semua arus data yang menuju ke atau keluar dari *parent process* harus ditampilkan lagi pada *child diagram*.

d. Pengecekan kesalahan

Pengecekan kesalahan pada diagram digunakan untuk melihat kesalahan yang terdapat pada sebuah DFD. Kesalahan yang umum terjadi dalam pembuatan DFD yaitu:

1. Sebuah proses tidak mempunyai masukan atau keluaran.
2. Simpanan data dengan entitas luar dihubungkan secara langsung tanpa melalui suatu proses.
3. Kesalahan dalam penamaan pada proses atau pada arus data.
4. Memasukkan lebih dari sembilan proses dalam sebuah diagram yang akan menyebabkan kebingungan dalam pembacaan.
5. Membuat ketidaksesuaian *decomposition* pada *child diagram*. Setiap *child diagram* harus mempunyai masukan dan keluaran yang sama dengan *parent process*.

3.4 Database

Menurut Marlinda (2004), *database* 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.

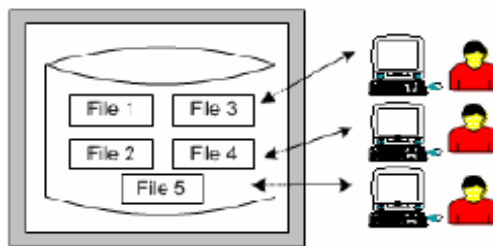
Terdapat beberapa aturan yang harus dipatuhi pada *file* basis data agar dapat memenuhi kriteria sebagai suatu basis data. Beberapa aturan itu berhubungan dengan (Kroenke, 1998):

1. Kerangkapan data, yaitu munculnya data-data yang sama secara berulang-ulang pada *file* basis data,
2. Inkonsistensi data, yaitu munculnya data yang tidak konsisten pada *field* yang sama untuk beberapa *file* dengan kunci yang sama,
3. Data terisolasi, disebabkan oleh pemakaian beberapa *file* basis data.
4. Keamanan data, berhubungan dengan masalah keamanan data dalam sistem basis data.
5. Integrasi data, berhubungan dengan kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian penuh.

3.4.1 Sistem Basis Data (DBMS)

Menurut Marlinda (2004), *Database Management Sistem* (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk mengelolanya. Basis data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data.

DBMS (*Database Management System*) kumpulan program yang digunakan *user* untuk me-management database (*create, maintain*).



Gambar 3.3 Proses *sharing* sistem basis data

Proses – proses yang terdapat dalam DBMS yaitu:

1. *Defining*: database mendefinisikan tipe data, struktur dan batasan (*constraint*) dari data yang disimpan dalam *database*.
2. *Manipulating*: database mencakup berbagai fungsi dan *query* untuk mendapatkan data yang dicari, termasuk operasi *insert*, *update* dan *delete* serta dalam *generate report data*.
3. *Sharing*: *database* dapat diatur untuk dapat *sharing multiple user* dan program untuk mengakses *database* secara bersama-sama.

Fungsi yang lebih penting dari DBMS adalah proteksi dan *maintain* database dalam jangka panjang.

- a. Proteksi: mengandung *system protection* yang menangani kondisi *malfunction* (*crash*) baik pada *hardware* ataupun *software*, juga mengandung *security protection* yang menangani pengaksesan oleh *user* terlarang.
- b. *Maintain*: mengandung sistem *maintaining* yang selalu meningkatkan kebutuhan perubahan tiap waktu.

3.4.2 Bagian-Bagian Basis Data

Membangun basis data adalah langkah awal dari pembuatan sebuah aplikasi.

Keberhasilan dalam membangun basis data akan menyebabkan program lebih

mudah dibaca, mudah dikembangkan dan mudah mengikuti perkembangan perangkat lunak.

Berikut ini diuraikan mengenai komponen-komponen yang terdapat dalam basis data:

1. Tabel, merupakan kumpulan dari suatu *field* dan *record* (kolom dan baris).
2. *Field*, adalah sebutan untuk mewakili suatu *record*.
3. *Record*, merupakan kumpulan elemen-elemen yang saling berkaitan yang berisi informasi tentang suatu isi data secara lengkap. Satu *record* mewakili satu data atau informasi tentang seseorang misalnya, nomor daftar, nama pendaftar, alamat, tanggal masuk.
4. *Primary Key*, merupakan kolom (*field*) yang menjadi titik acuan pada sebuah tabel yang bersifat unik, dimana dalam artian data tidak ada satu nilai pun yang sama atau kembar dalam tabel tersebut.
5. *Foreign Key* (kunci relasi), suatu kolom dalam tabel yang digunakan sebagai “kaitan” atau relasi untuk membentuk satu hubungan yang didapati dari tabel induk, umumnya hubungan yang terbentuk antar tabel adalah satu ke banyak (*one to many*).
6. *Index*, merupakan struktur basis data secara fisik, yang digunakan untuk mengoptimalkan pemrosesan data dan mempercepat proses pencarian data.

3.5 Visual Basic.net

Visual Basic .NET adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem *.NET Framework*, dengan menggunakan bahasa *BASIC*. Dengan menggunakan alat ini,

para programmer dapat membangun aplikasi Windows Forms, Aplikasi web berbasis ASP.NET, dan juga aplikasi *command-line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *Visual J#*), atau juga dapat diperoleh secara terpadu dalam *Microsoft Visual Studio .NET*.

Bahasa *Visual Basic .NET* sendiri menganut paradigma bahasa pemrograman berorientasi obyek yang dapat dilihat sebagai evolusi dari *Microsoft Visual Basic* versi sebelumnya yang diimplementasikan di atas *.NET Framework*. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.