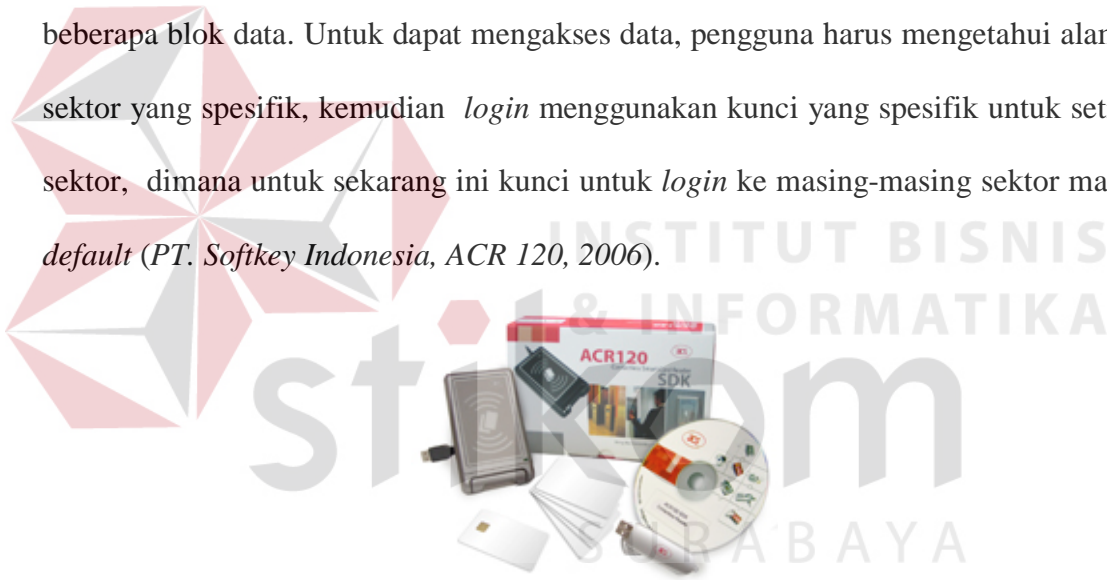


## BAB II

### LANDASAN TEORI

#### 2.1 *Contactless Smart Card*

Di dalam sebuah *smart card*, kita bisa memasukkan berbagai macam informasi. Diantaranya adalah data pribadi, data perusahaan, data transaksi, serta catatan kegiatan yang melibatkan penggunaan kartu tersebut. Media penyimpanan pada *contactless smart card* terdiri atas sejumlah sektor, dan setiap sektor terdiri atas beberapa blok data. Untuk dapat mengakses data, pengguna harus mengetahui alamat sektor yang spesifik, kemudian *login* menggunakan kunci yang spesifik untuk setiap sektor, dimana untuk sekarang ini kunci untuk *login* ke masing-masing sektor masih *default* (PT. Softkey Indonesia, ACR 120, 2006).



Gambar 2.1 *Smart Card*

Salah satu komponen yang penting dalam penggunaan teknologi ini adalah aspek keamanan. Data yang tersimpan biasanya data yang bersifat rahasia yang hanya boleh diakses oleh pihak yang berhak saja. Untuk itu perlu adanya mekanisme pengamanan yang spesifik untuk melindungi informasi yang disimpan didalamnya. Sebenarnya di dalam *mifare contactless smart card* ini sudah ada mekanisme

pengamanan yang cukup baik, tetapi bentuknya masih *standard*, sehingga setiap pihak yang mengetahui strukturnya akan dapat menembus lapisan keamanan ini. Untuk itu perlu dibuat sebuah mekanisme pengamanan yang spesifik hanya untuk satu perusahaan saja. Bentuk *contactless smart card* dapat dilihat pada Gambar 2.1.

Sampai sekarang, sebagian besar perusahaan *provider contactless smart card* masih menyimpan informasi pada kartu dalam bentuk *plaintexts*, sehingga apabila mengetahui struktur data di dalam kartu, maka akan dengan mudah data di dalam kartu dibaca dan dimodifikasi. Untuk itu diperlukan sebuah metode pengamanan yaitu perubahan konfigurasi, yang mengubah *password* setiap sektor menjadi unik. Tujuannya adalah supaya tidak semua orang bisa *login* ke sektor tersebut kemudian membaca dan memodifikasi isi didalamnya. Selain itu perlu diaplikasikan juga sebuah algoritma enkripsi yang mengenkripsi blok data yang ada di dalam *smart card*. Sehingga apabila lapisan keamanan pertama yaitu *password* sektor tersebut ditembus, maka data yang terbaca masih berupa *cipherteks*. Untuk lebih jelasnya, proses baca/ tulis kartu dapat dilihat pada Gambar 2.2.

#### 1. Perubahan Konfigurasi *Smart Card*

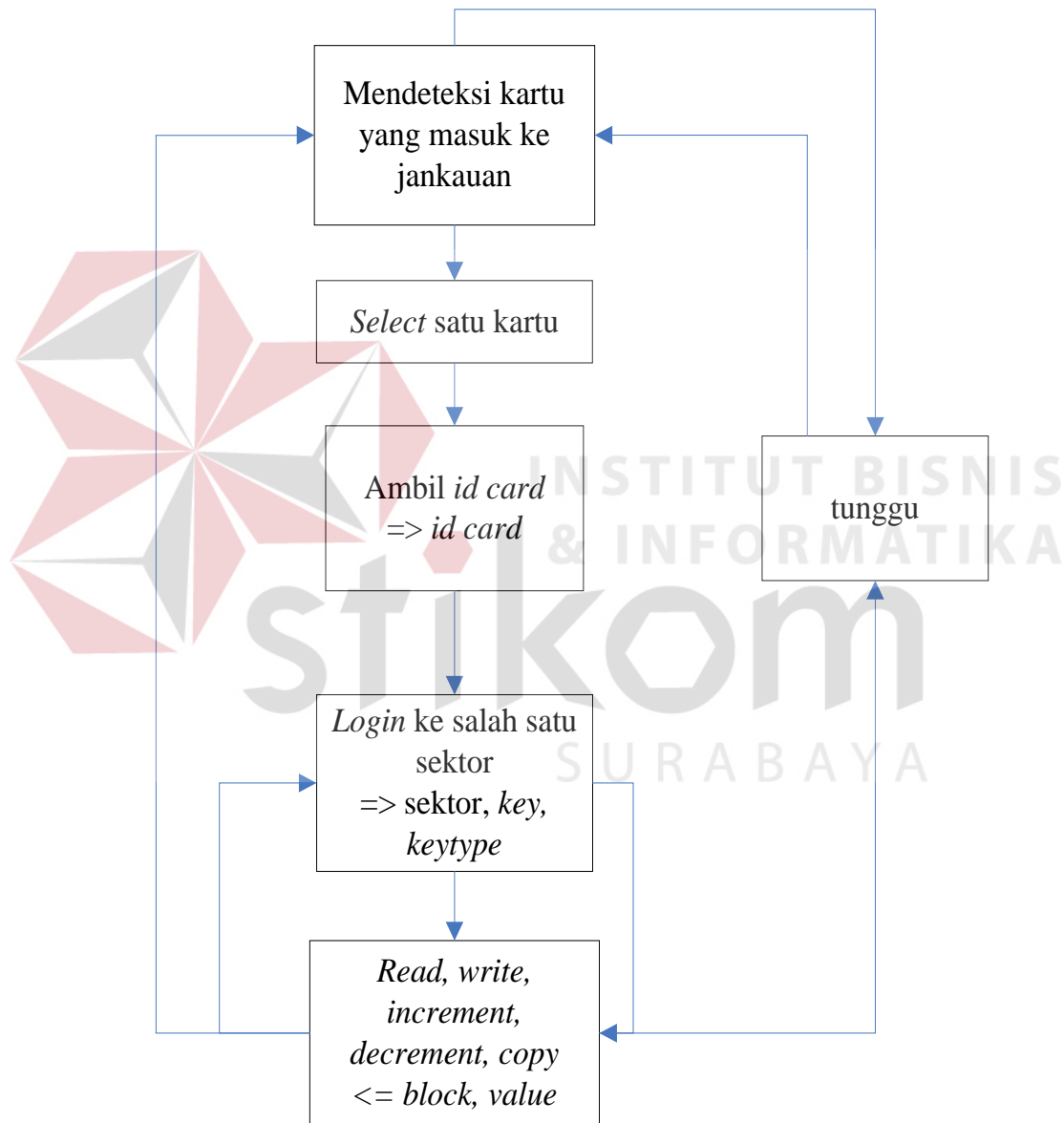
*Mifare contactless smart card* yang berukuran 1kb mempunyai 2 macam kunci *login* untuk dapat masuk ke dalam sektor, yaitu kunci A dan kunci B. Keduanya bisa dipakai untuk mengamankan kartu, tetapi umumnya, kunci yang dipakai hanyalah kunci A saja, karena dianggap dengan menggunakan satu macam kunci saja sudah relatif aman. Pada keadaan awal, kunci untuk masuk setiap sektor pada *Mifare contactless smart card* adalah *default*. Secara *default*, isi kuncinya adalah.

Key A : FF FF FF FF FF FF (*Phillips*)

*Key B* : FF FF FF FF FF FF (*Phillips*)

*Access Condition* : FF 07 80 69

Hal ini sangat rawan, karena pada kondisi ini berarti setiap *programmer* pasti bisa mengakses isi setiap sektor pada kartu, bahkan mengubah isinya.



Gambar 2.2 Alur Baca/tulis Kartu

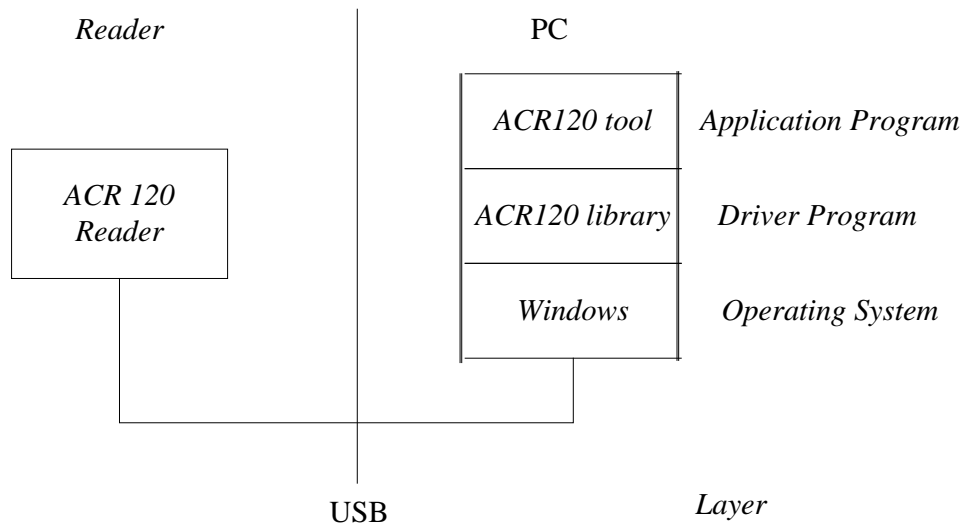
Dalam aplikasi ini, kunci untuk masuk ke sektor diubah dengan mengambil *card-id* sebagai umpan masukannya. Pada aplikasi ini, sektor yang digunakan adalah sektor keenam saja, sehingga perubahan kunci untuk masuk sektor hanya dilakukan pada sektor keenam saja.

Cara pemakaian umpan adalah dengan mendeteksi *card-id* yang terbaca dari kartu kemudian mengambil nilai *hash*-nya. Hasil keluaran nilai *hash* ini adalah 32 karakter heksadesimal. Dari urutan karakter ini, diambil 6 karakter pertama, yang kemudian dipakai untuk menggantikan kunci untuk *login* pada sektor pertama.

Kunci untuk masuk ke sektor terdapat pada blok terakhir dari setiap sektor, atau yang biasa disebut sebagai *sector trailer*. Pengubahan kunci *login* sektor pertama dilakukan dengan cara menimpa atau menuliskan karakter baru ke dalam 6 *byte* pertama blok ke-4 dari sektor tersebut. Hanya 6 *byte* pertama saja yang diubah, karena *byte* selanjutnya adalah *access condition* yang mengatur hak akses ke sektor tersebut. Setelah 6 *byte* tersebut diubah, maka otomatis pada proses *login* selanjutnya harus sudah menggunakan kunci yang baru.

## 2. Enkripsi Data dalam *Smart Card*

Aplikasi yang dibuat pada tugas akhir ini akan ditanamkan pada komputer yang terhubung dengan *card reader device*. *Device* itulah yang akan digunakan untuk baca sekaligus tulis data pada *smart card*. Pada kenyataannya, *card reader* inilah yang akan berinteraksi langsung dengan *user*. Aplikasi yang dibuat ini termasuk pada *ACR120 tool*, yaitu aplikasi yang berfungsi untuk mengendalikan proses-proses yang dilakukan oleh *card reader* melalui *device driver*. Hubungan antara *reader* dan aplikasi dijelaskan pada Gambar 2.3.



Gambar 2.3 Hubungan Antara *Reader* dan Aplikasi

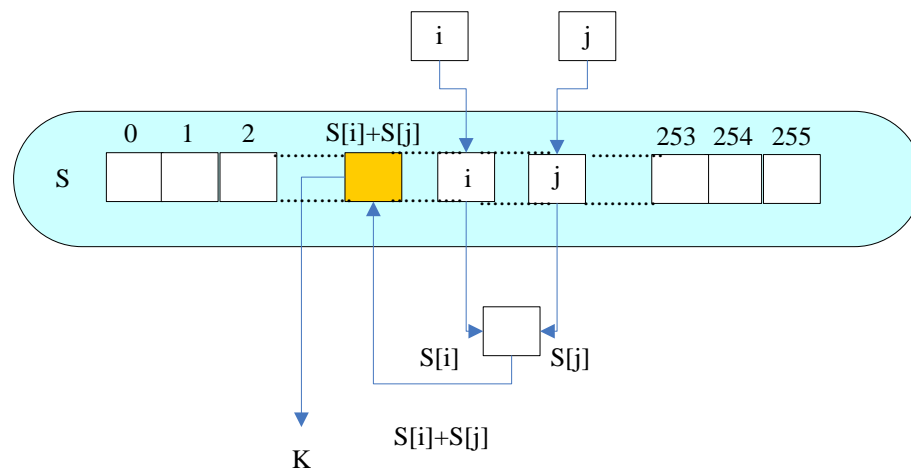
Salah satu hal yang menjadi masalah penting dalam proses transaksi menggunakan *smart card* adalah *processing time*, yaitu waktu yang dibutuhkan dari mulai *card reader* mendeteksi bahwa ada kartu di dalam jangkauan sensornya, pembacaan, autentikasi, transaksi, sampai pemutusan hubungan antara *reader* dengan kartu. *User* pasti akan merasa lebih nyaman dan tenang apabila proses transaksi yang dilakukan itu cepat selesai. Untuk itu, maka pemrosesan transaksi pada kartu harus seefisien dan sesimpel mungkin, termasuk proses enkripsi dan dekripsinya tidak boleh terlalu signifikan menambah *processing time*-nya.

Data disimpan di dalam *smart card* dalam bentuk blok-blok data yang masing-masing berisi *byte* data. Dan penyusunan didalamnya diletakkan dalam sektor-sektor yang terpisah. Sedangkan untuk pembacaan datanya, harus melalui proses *login* terlebih dahulu ke dalam sektor yang diinginkan. Sekali *login* untuk sekali pengaksesan ke satu sektor. Setelah *login* berhasil, baru kemudian sistem dapat melaksanakan operasi baca/ tulis terhadap kartu tersebut. Untuk itu, maka algoritma

enkripsi yang tepat adalah RC4. Selain itu, memiliki kebutuhan bahwa algoritma enkripsi itu tidak boleh terlalu rumit dan membebani *card reader*, maka harus dipilih algoritma yang cukup sederhana, tapi kuat.

Untuk melakukan enkripsi dengan menggunakan algoritma RC4, ada beberapa parameter yang dipergunakan. Yaitu blok permutasi *S-Box*, serta kunci U. Supaya setiap proses enkripsi memiliki proses yang unik, maka blok permutasi S itu diisi dengan suatu parameter yang unik pula, yaitu *id card*. *Id card* adalah 4 *byte* unik yang ada di sektor pertama blok pertama pada setiap *Mifare contactless smart card*. *Id card* tersebut harus dimodifikasi supaya panjangnya menjadi 256 *bytes*. Caranya adalah dengan mengambil nilai *hash id card* tersebut, kemudian mengisikannya ke blok permutasi *S-Box*. Fungsi *hash* yang digunakan adalah MD5. Nilai *hash* yang dihasilkan oleh fungsi MD5 berukuran 32 *byte*, sementara ukuran blok permutasi *S-Box* adalah 256 *byte*, untuk itu nilai *hash* yang didapat harus diisikan secara berulang sebanyak 8 kali. Hal ini dilakukan untuk menambah performa keamanannya. Karena apabila *id card* tersebut hanya diambil mentah-mentah, maka kemungkinan terbaknya parameter algoritma itu akan sangat mudah.

Setelah larik permutasi *S-Box* dan kunci U siap, maka dilakukan proses *key-scheduling* yang prosesnya dapat dilihat pada Gambar 2.4 yaitu proses untuk membangkitkan aliran-*bit*-kunci. Pada saat *card reader* melakukan proses penulisan, *reader* akan langsung menuju ke sektor yang diinginkan. *Login* ke sektor tersebut diinisialisasi, dan bersamaan dengan proses penulisan *byte-byte*-nya, maka dilakukan proses *pseudo-random generation algorithm*.



Gambar 2.4 Key Scheduling

Jadi pada setiap *bytenya*, terjadi 4 proses yang berjalan berurutan, yaitu *generate* aliran-bit-kunci, meng-XOR-kan dengan *byte* yang akan dituliskan, menuliskan *byte* yang sudah dienkripsi ke dalam *memory smart card*, kemudian mengacak ulang isi kotak permutasi *S-Box*. Proses ini diulangi sebanyak jumlah *byte* yang akan dituliskan.

## 2.2 Sewa Bagi Hasil

Sewa bagi hasil adalah sebuah sistem dimana pedagang dan pengelola saling bekerja sama saling menguntungkan dan sama-sama pula menanggung rugi. Sistem ini adalah sebuah sistem yang sangat cepat sekali bagi pengelola dalam mencari para pedagang atau *tenant* untuk mengisi gerai-gerai yang telah disiapkan. Langkah ini diambil agar keseluruhan operasional dan peraturan menjadi hak *exclusive* pengelola dan pedagang harus mematuhi jika ingin bergabung atau masih ingin bergabung. Di tahap awal pengelola memasarkan *food court* kepada para

pedagang dengan keunggulan dari lokasi *food court* yang dikelolanya. Kesepakatan antara pedagang dan pengelola *food court* memiliki syarat sebagai berikut.

1. Pedagang dikenakan prosentase bagi hasil dari omzet kotor yang biasanya 30%.
2. Pedagang akan dikenakan deposit dan itu berlaku juga di sistem sewa, yang dimaksudkan adalah selama masa sewa atau kerja sama pedagang tidak begitu saja memutuskan masa kerja sama atau sewa, karena resikonya uang deposit akan hilang.
3. Perjanjian kerja sama kedua belah pihak diatas materai.
4. Bilamana ada perbaikan, pengelola memberikan waktu pengerjaan maksimal satu minggu sebelum operasional.
5. Pedagang harus membayar listrik dan air yang digunakan sesuai dengan meteran yang akan dipasang di setiap gerai.

### **2.3 Food Court**

Secara definisi stan berjualan adalah sebuah ruangan yang dirancang khusus, lengkap dengan meja untuk menyajikan, tempat untuk menyimpan barang, dan perlengkapan lain yang bersih, aman, dan higienis, yang berguna untuk memenuhi kebutuhan publik, baik publik lokal, internasional, maupun pelaku perjalanan. *Food court* adalah sebuah tempat yang luas dimana berkumpul para pedagang makanan dari semua tingkatan yang terkordinir. *Food court* dilengkapi dengan aneka fasilitas yaitu.

1. Penyejuk udara atau AC.
2. Televisi.



3. Penerangan yang dapat disesuaikan dengan suasananya.
4. Panggung hiburan.
5. Wastafel.
6. Toilet.

Pengelola menyediakan unit-unit/ gerai-gerai bagi para pedagang dengan ukuran yang tidak terlalu luas yang disebut Gerai. Pedagang dapat memproduksi dan sekaligus menjual produknya di area tersebut dimana kesegaran dan kebersihannya dapat terjaga, karena udaranya bersih dari polusi. Stan makanan yang didesain dengan baik akan membuat para pengunjung nyaman, sebaliknya stan makanan yang tidak didesain dengan baik akan membuat para pengunjung merasa tidak nyaman dan ingin beranjak sesegera mungkin (wikipedia.co.id).

#### **2.4 Sistem Informasi**

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Herlambang dan Tanuwijaya (2005), sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedural dan pendekatan secara komponen. Berdasarkan pendekatan prosedural, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi yaitu *input*, proses, *output*. Menurut

Herlambang dan Tanuwijaya (2005), data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya, sehingga sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya.

Data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relatif tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya, atau sebaliknya. Pengertian sistem informasi itu sendiri adalah suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya.

Sebuah sistem terintegrasi atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi. Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen dan basis data.

## **2.5 Analisis dan Perancangan Sistem**

Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang

terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Kendall dan Kendall, 2003). Tahap analisis sistem dilakukan setelah tahap perencanaan sistem (*system planning*) dan sebelum tahap perencanaan sistem (*system design*). Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini juga akan menyebabkan kesalahan di tahap selanjutnya.

Dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem adalah.

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

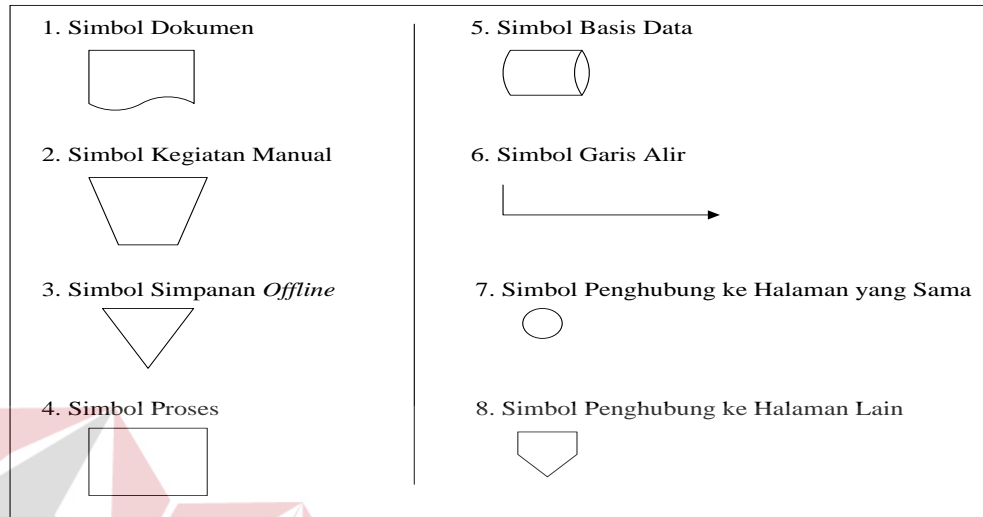
Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan perancangan sistem. Analisis dan Perancangan Sistem dipergunakan untuk menganalisis, merancang, dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi (Kendall dan Kendall, 2003).

### **2.5.1 System Flow**

*System flow* atau bagan alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan urutan-urutan

dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dikerjakan sistem.

Simbol-simbol yang digunakan dalam *system flow* ditunjukkan pada Gambar 2.5.



Gambar 2.5 Simbol-simbol pada *Sistem Flow*

1. Simbol dokumen

Menunjukkan dokumen *input* dan *output* baik untuk proses manual atau komputer.

2. Simbol kegiatan manual

Menunjukkan pekerjaan manual.

3. Simbol simpanan *offline*

Menunjukkan *file* non-komputer yang diarsip.

4. Simbol proses

Menunjukkan kegiatan proses dari operasi program komputer.

5. Simbol basis data

Menunjukkan tempat untuk menyimpan data hasil operasi komputer.

6. Simbol garis alir

Menunjukkan arus dari proses.

## 7. Simbol penghubung

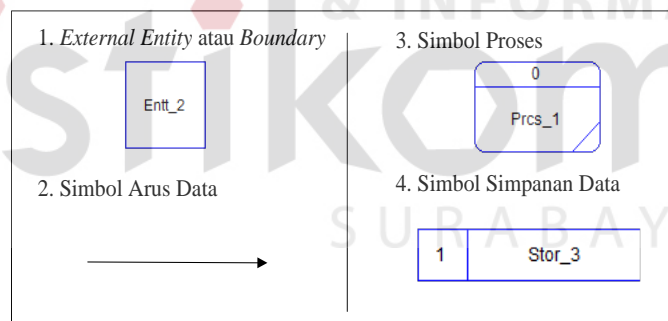
Menunjukkan penghubung ke halaman yang masih sama atau ke halaman lain.

### 2.5.2 Data Flow Diagram

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan terstruktur dan jelas (Kendall dan Kendall, 2003).

#### 1. Simbol-Symbol yang digunakan dalam DFD

Di dalam penggambaran DFD terdapat beberapa simbol yang digunakan seperti yang dapat dilihat pada Gambar 2.6



Gambar 2.6 Simbol-Symbol pada DFD

#### a. *External Entity* atau *Boundary*

Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. *External entity* tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untu

satu bagian maka bagian lain yang masih terkait menjadi *external entity* (Al-Bahra, 2005).

*b. Arus Data*

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir di antara proses, data store dan menunjukkan arus data dari dataa yang berupa masukan untuk sistem atau hasil proses sistem (Al-Bahra, 2005).

*c. Proses*

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data ke luar. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa keluaran. Proses sering disebut *bubble* (Al-Bahra, 2005).

*d. Simpanan Data*

Simpanan data merupakan tempat penyimpanan data pengikat data yang ada dalam sistem. Data storee dapat disimbolkan denan sepasan dua garis sejajar aatau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke database (Al-Bahra, 2005).

*2. Context Diagram*

Menurut Raymond dan George (2009) context diagram menerapkan sistem dalam suatu konteks lingkungan. Diagram ini terdiri atas satu simbol proses tunggal yang

melambangkan keseluruhan sistem. Diagram ini menunjukkan arus data yang mengarah dan keluar dari terminator.

## 2.6 Sistem Basis Data

Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/ berkumpul. Data adalah representasi fakta dunia nyata yang mewakili suatu subyek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/ tabel/ arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/ arsip, sedangkan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/ arsip. Perbedaannya terletak pada media penyimpanan yang digunakan. Jika lemari arsip

menggunakan lemari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti *disk* (disket atau *harddisk*).

Hal ini merupakan konsekuensi yang logis karena lemari arsip langsung dikelola/ ditangani oleh manusia, sementara basis data dikelola/ ditangani melalui perantara alat/ mesin pintar elektronis (yang kita kenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metoda/ cara yang dapat digunakan dalam upaya penyimpanan (Fathansyah, 1999).

### **2.6.1 Database Management System**

Menurut Indrajani (2009), *Database Management System* (DBMS) mulai dikenalkan di tahun 1960 saat pesawat Apollo mendarat di bulan. Pada waktu itu tidak ada sistem yang mampu menangani dan mengatur sejumlah informasi yang diperlukan proyek tersebut. Lembaga Ilmu Penerbangan Amerika Utara, yaitu North American Aviation (sekarang Rockwell International), sebagai pemborong utama untuk proyek perangkat luna, memperkenalkan Guam (Generalized Update Access Method). Guam didasarkan pada konsep bahwa komponen yang lebih kecil merupakan awal bagian dari komponen yang lebih besar, dan seterusnya sampai produk dirakit. Struktur ini dikenal dengan nama struktur hirarki dan digambarkan seperti pohon terbalik.



## 2.6.2 Bahasa-bahasa yang Terdapat dalam DBMS

### a. *Data Definition Language* (DDL)

struktur/ skema basis data yang menggambarkan/ mewakili desain basis data secara keseluruhan dispesifikasikan dengan bahasa khusus yang disebut *Data Definition Language* (DDL). Dengan bahasa inilah kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel dan sebagainya. Hasil dari kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam *file* khusus yang disebut Kamus Data (*Data Dictionary*).

Kamus Data merupakan suatu metadata (superdata) yaitu data yang mendeskripsikan data sesungguhnya. Kamus data ini akan selalu diakses dalam suatu operasi basis data sebelum suatu *file* data yang sesungguhnya diakses (Fathansyah, 1999).

### b. *Data Manipulation Language* (DML)

merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa:

1. Penyisipan/ penambahan data baru ke suatu basis data.
2. Penghapusan data dari suatu basis data.
3. Pengubahan data di suatu basis data.

Pada level fisik, kita harus mendefinisikan algoritma yang memungkinkan pengaksesan yang efisien terhadap data. Pada level yang lebih tinggi, yang dipentingkan bukan hanya efisiensi akses, tetapi juga efisiensi interaksi manusia (pemakai) dengan sistem (kemudahan permintaan akses).

*Data Manipulation Language* (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada 2 (dua) jenis DML:

- a. Prosedural, yang mensyaratkan agar pemakai menentukan data apa yang diinginkan serta bagaimana cara mendapatkannya.
- b. Nonprosedural, yang membuat pemakai dapat emnentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya (Fathansyah, 1999).

### **2.6.3 Entity Relationship Diagram**

*Entity Relationship Diagram* (ERD) yang berisi komponen Himpunan Entitas dan Himpunan Relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari “dunia nyata” yang kita tinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Diagram Entity-Relationship* (*Diagram E-R*). Notasi-notasi simbolik di dalam *Diagram E-R* yang dapat kita gunakan yaitu:

- a. Persegi panjang, menyatakan himpunan entitas.
- b. Lingkaran/elip, menyatakan atribut (atribut yang berfungsi sebagai *key* digarisbawahi).
- c. Belah ketupat, menyatakan himpunan relasi.
- d. Garis, sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dan atributnya.

- e. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak) (Fathansyah, 1999) .

## 2.7 *Visual Basic .Net 2005*

Secara sederhana gambaran dari proses pengembangan aplikasi adalah bahwa *Visual Studio .NET 2005* menyediakan area atau lingkungan pengembangan terintegrasi atau sering disebut dengan *Interface Development Environment (IDE)* (Yuswanto, 2005). Lingkungan ini menyediakan *tool* untuk mendesain, menjalankan dan mencari kesalahan program dari aplikasi yang dibuat. Untuk melakukan *setting property* dari masing-masing kontrol, dapat melihat menu *Properties Window* dan double klik pada *form* untuk memasukan kode program.

Pada *Visual Studio .NET 2005* tampilan lingkungan pada berbagai bahasa pemrogramannya dibuat identik. Hal ini untuk mempermudah dalam mendesain dan mengelola suatu *solution* atau proyek yang menggunakan lebih dari satu bahasa pemrograman pada saat yang sama.

Ada perbedaan lingkungan yang ditempati untuk aplikasi yang dibuat dengan *Visual Basic 6* dengan *Visual Basic .NET 2005*. Pada *Visual Basic 6*, keseluruhan program aplikasi disebut dengan proyek (*Project*) dan umumnya hasil akhir dari proyek tersebut adalah *file .exe* atau *.dll*. Sedangkan pada *Visual Basic .NET 2005* menggunakan *solution* yang berisi lebih dari satu proyek dan juga berisi gambar, *file XML* serta elemen lainnya yang merupakan bagian dari program. Dalam satu *solution*

dapat dibuat lebih dari *file .exe* atau *.dll* dari aplikasi yang dibuat oleh bahasa pemrograman yang berbeda-beda.

## 2.8 Microsoft SQL Server 2005

*SQL Server 2005* atau disebut juga dengan *SQL Server 9* merupakan pengembangan dari *SQL Server* versi 2000. Pada *SQL Server 2000* disediakan beberapa *tools* yang dapat digunakan oleh para *developer*, yaitu:

1. *SQL Server Management Studio*
2. *SQL Computer Manager*
3. *Sqlcmd (SQL Command)*
4. *SQL Management Object*

*SQL Server 2005* merupakan aplikasi *database* produk *Microsoft* yang memiliki fitur-fitur baru sehingga membuatnya menjadi suatu *platform database* yang sempurna, khususnya dalam mengolah data yang berskala besar dan aplikasi *e-commerce*. Fitur-fitur tersebut antara lain:

1. *Notification Services*

Fitur ini digunakan untuk mengirimkan dan menerima pesan atau pemberitahuan (*notification*) dengan tepat waktu dari *database* ke ribuan pengguna.

2. *Reporting Services*

Fitur ini digunakan untuk membuat laporan dari *SQL Server 2005*.

### 3. *Services Broker*

Fitur ini, menyediakan infrastruktur yang digunakan untuk membangun keamanan pada aplikasi berkinerja tinggi (Yuswanto, 2005).

