BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah PT. Prima Info Sarana Media (Divisi CHIP LAB)

PT. Prima Info Sarana Media adalah penerbit utama majalah komputer bulanan CHIP Indonesia. Majalah ini adalah salah satu majalah komputer bulanan yang telah beredar sejak lama dan telah populer di masyarakat. Disamping penerbit, perusahaan ini juga bergerak sebagai konsultan di bidang teknologi komputer, salah satunya adalah dipercaya untuk menangani beberapa proyek pengadaan komputer untuk instansi/perusahaan. Salah satu divisi yang dipercaya oleh perusahaan ini untuk menangani proyek pengadaan tersebut, bernama divisi CHIP lab. Divisi inilah yang nantinya bertanggung jawab untuk menganalisa sebuah spesifikasi komputer.

2.1.2 Visi dan Misi PT. Prima Info Sarana Media

PT. Prima Info Sarana Media mempunyai visi dan misi dalam menjalankan tugasnya sebagai perusahaan yang bergerak di bidang media percetakan. Adapun visi dan misi PT. Prima Info Sarana Media adalah:

1. Visi

   Menjadi perusahaan yang terkemuka dalam bidang media Informasi dan Teknologi terkini.

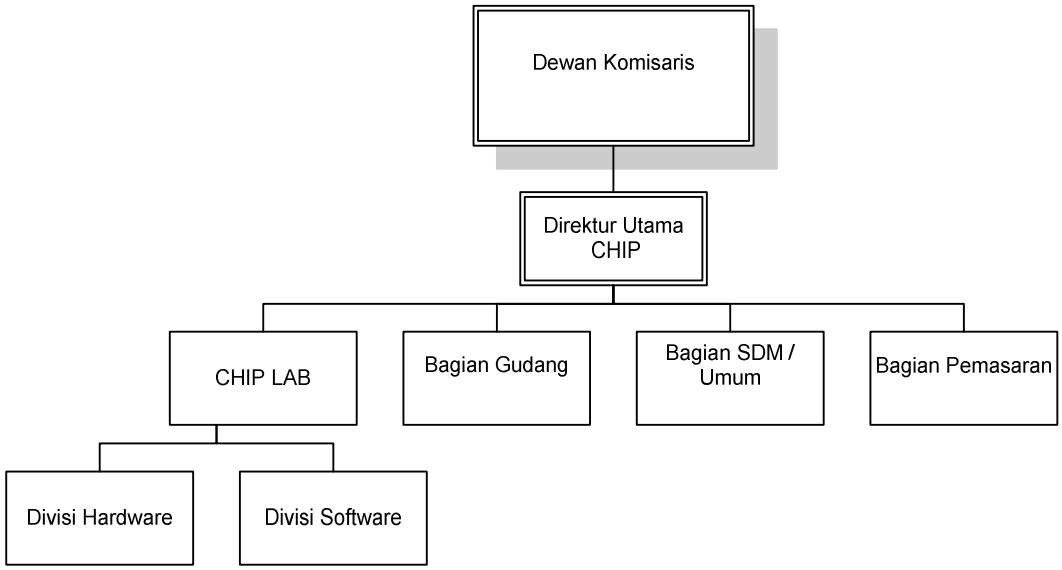
2. Misi

   Meningkatkan nilai perusahaan melalui peningkatan kualitas SDM, sistem manajemen perusahaan dan pertumbuhan profitabilitas usaha.

2.1.3 Struktur Organisasi PT. Prima Info Sarana Media

Organisasi perusahaan sangat penting dalam menjamin kelangsungan dan kelancaran mekanisme kerja perusahaan, adanya organisasi dimaksudkan untuk menciptakan suatu sistem pembagian kerja atau tugas yang sesuai dengan kebutuhan perusahaan sehingga mempermudah kegiatan operasional perusahaan dalam mencapai tujuan.

Struktur organisasi PT. Prima Info Sarana dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi PT. Prima Info Sarana Media

Uraian tugas dari struktur organisasi PT. Prima Info Sarana Media adalah sebagai berikut:

1. Direktur Utama

   Direktur utama bertugas membawahi semua struktur yang ada dibawahnya dan bertanggung jawab kepada dewan komisaris.

2. CHIP LAB

   CHIP LAB bertugas untuk melakukan pengujian produk Hardware dan Software didalam Laboratorium.

   2.1. Divisi *Hardware*

   Tugas dari divisi *Hardware* adalah melakukan penilaian *Hardware* secara ilmiah dengan melakukan pengujian-pengujian teknis secara detail.

   2.2. Divisi *Software*

   Membuat dan melakukan pengujian *Software* yang masuk kedalam CHIP Test Center.

3. Bagian Gudang

   Mengkoordinasi semua alat yang dipakai untuk pekerjaan *benchmarking*, *overclocking*, dan *audition.*

4. Bagian SDM / Umum

   Bagian SDM / umum bertugas mengurus masalah yang berhubungan dengan karyawan dan tenaga ahli.

5. Bagian Pemasaran

   Bagian pemasaran bertugas mengikuti semua tender untuk mendapatkan pekerjaan pada suatu proyek.

2.2 Konsep Dasar Sistem

Dalam perancangan suatu sistem informasi diarahkan kepada pemanfaatan teknologi secara maksimal yang terdiri dari beberapa elemen atau komponen yang membentuk jaringan kerja dan mempunyai tujuan yang ingin dicapai. Pendekatan yang menekankan pada prosedur, mendefinisikan sebuah sistem sebagai berikut [1]:

"*Suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai tujuan*".

Suatu sistem dapat terdiri dari sistem bagian (*subsystems*). Sebagai misal, sistem komputer dapat terdiri dari subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukkan, alat pemroses, alat keluaran dan simpanan luar. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan hingga tujuan/sasaran sistem tersebut dapat tercapai. Interaksi dari subsistem-subsistem sedemikian rupa, sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi (*integrated*).

2.2.1 Karakteristik Sistem

Dengan adanya pengertian sistem tersebut maka sifat-sifat atau karakteristik sistem dapat ditentukan sebagai berikut :

1. Adanya komponen sistem

   Dimana komponen ini terdiri dari subsistem-subsistem yang saling berhubungan atau berinteraksi membentuk suatu kesatuan.

2. Adanya batasan sistem

   Batasan sistem (Boundary) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

3. Lingkungan luar sistem

   Lingkungan luar sistem (Environment) apapun diluar batas dari sistem yang mempengaruhi operasi sistem, baik itu yang menguntungkan ataupun yang merugikan.

4. Penghubung sistem

   Merupakan suatu media penghubung antar satu subsistem dengan subsistem yang lain sehingga dapat berintegrasi membentuk satu kesatuan.

5. Masukan sistem

   Masukan adalah merupakan masukan sistem yang akan diolah menjadi informasi, misalnya program dan data.

6. Keluaran sistem

Keluaran adalah hasil dari proses yang terjadi dan diklarifikasikan menjadi keluaran yang berguna, yaitu infomasi dan sisa pembuangan, misalnya panas yang dihasilkan oleh komputer.

7. Pengolahan sistem

   Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem

   Sistem pasti mempunyai tujuan atau sasaran yang akan menentukan masukan dan keluaran

2.2.2 Klasifikasi Sistem

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

1. Sistem diklasifikasikan sebagai Sistem Abstrak dan Sistem Fisik
   a. Sistem Abstrak

      Adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.

   b. Sistem Fisik

      Adalah sistem dari unsur-unsur yang bekerja bersama untuk mencapai suatu tujuan tertentu dimana sistem-sistem tersebut lebih sekedar kerangka-kerangka konsep belaka.

2. Sistem diklasifikasikan sebagai Sistem Alamiah dan Sistem Buatan Manusia

   a. Sistem Alamiah

      Adalah sistem yang terjadi melalui proses alam dan tidak dibuat oleh manusia.

   b. Sistem Buatan Manusia

      Adalah sistem yang dirancang manusia dan melibatkan interaksi antara manusia dan mesin.

3. Sistem diklasifikasikan sebagai Sistem Tertentu dan Sistem Tak Tentu

   a. Sistem Tertentu

      Adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, interasi antara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan.

   b. Sistem Tak Tentu

      Adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai Sistem Tertutup dan Sistem Terbuka

   a. Sistem Tertutup

      Adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya.

   b. Sistem Terbuka

      Adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.3 Perancangan Sistem (*System Design*)

Proses perancangan diperlukan untuk menghasilkan suatu rancangan sistem yang baik, karena dengan adanya rancangan yang tepat akan menghasilkan sistem yang stabil dan mudah dikembangkan di masa mendatang. . Perancangan yang kurang baik akan mengakibatkan sistem yang dibangun harus dirombak total atau sistem yang dibangun akan sangat berlebihan dari kebutuhan yang diperlukan. Tahap perancangan disebut juga tahap pemecahan masalah, yaitu dengan menyusun suatu alur sistem, masukan, prosedur proses, keluaran dan database.

2.3.1 Bagan Alir Dokumen (*flow map*)

Bagan alir dokumen merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Beberapa simbol yang digunakan di bagan alir dokumen antara lain :

a. Dokumen (*document*)

   Simbol ini menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer.

b. Kegiatan Manual (*Manual Operation*)

   Simbol ini menunjukkan pekerjaan manual yang dilakukan oleh orang.

c. Simpanan *Off line* (*off-line Storage*)

   Simbol ini menunjukkan file non-komputer yang diarsipkan.

d. Proses (*Process*)

Simbol ini menunjukkan kegiatan proses dari operasi program komputer.

e. Simpanan Data (*data storage*)

Simbol ini menunjukkan tempat penyimpanan data.

f. Penghubung (*connector*)

Simbol ini menunjukkan penghubung kehalaman yang masih sama atau kehalaman lain.

2.3.2 Flow Chart (Bagan Alir)

Flow chart atau bagan alir adalah bagan (*chart*) yang menunjukan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Ada lima macam bagan alir yang biasa dipakai, yaitu :

2.3.2.1 System Flow Chart (Bagan Alir Sistem)

Bagan alir sistem (*System Flowchart*) merupakan bagan yang menunjukan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan-urutan dari prosedur yang ada di dalam sistem. Bagan alir sistem menunjukan apa yang sedang dikerjakan di sistem.

2.3.2.2 Document Flow chart (Bagan Alir Dokumen)

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (form flowchart atau *paperwork flowchart*) merupakan bagan alir yang menunjukan arus data laporan dan formulir termasuk tembusan-tembusan. Bagan alir ini menggunakan simbol yang sama.

2.3.2.3 Schematic Flow Chart (Bagan Alir Skematik)

Bagan alir skematik *(schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarkannya.

2.3.2.4 Program Flow Chart (Bagan Alir Program)

Bagan alir program (*program flow chart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan Alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*).

Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika.

2.3.2.5 Process Flow Chart (Bagan Alir Proses)

Bagan alir proses (*Process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses menggunakan 5 buah simbol tersendiri.

2.3.3 Diagram Konteks

Diagram konteks adalah model atau gambar yang menggambarkan hubungan sistem dengan lingkungan sistem. Untuk menggambarkan diagram konteks, kita deskripsikan data apa saja yang dibutuhkan oleh sistem dan dari mana sumbernya, serta informasi apa saja yang akan dihasilkan oleh sistem tersebut dan kemana informasi tersebut akan diberikan.

Diagram konteks adalah kasus khusus dari DFD atau bagian dari DFD yang berfungsi memetakan modul lingkungan yang di presentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

2.3.4 Diagram Arus Data (*Data Flow Diagram)*.

DFD *(Data Flow Diagram)* digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika

tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telepon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan (misalnya *harddisk*, *Diskette, CD,* dan sebagainya).

Beberapa simbol yang digunakan di DFD antara lain :

a. Kesatuan Luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external* entity) merupakan kesatuan di lingkungan luar sistem dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.

b. Arus Data (*Data Flow*)

Arus data di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

c. Proses

Proses adalah kegiatan atau kerja 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. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul. Simpanan Data (*Data Store*)

Simpanan data merupakan simpanan dari data yang dapat berupa :

- Suatu *file* atau *database* di sistem komputer
- Suatu arsip atau catatan manual.
- Suatu kotak tempat data di meja seseorang.
- Suatu tabel acuan manual.
- Suatu agenda atau buku.

Simpanan data di DFD dapat disimbolkan dengan sepasang garis horizontal paralel yang kedua ujungnya terbuka atau salah satu ujungnya tertutup.

2.3.5 Kamus Data

Kamus Data atau *Data Dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data digunakan untuk mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat berdasarkan arus data yang ada dalam *DFD* (*Data Flow Diagram*) dan hanya ditunjukkan arus datanya saja.

2.4 Desain Basis Data (*Database*)

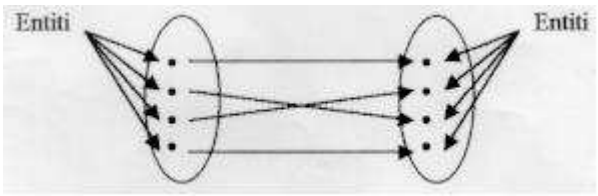
Basis data merupakan kumpulan dari data-data yang saling terkait dan saling berhubungan satu dengan lainnya. Basis data adalah kumpulan kumpulan *file* yang saling berkaitan.

2.4.1 ERD (*Entity Relationship Diagram*)

ERD *(Entity Relationship Diagram)* berfungsi untuk menggambarkan relasi dari dua file atau dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi, antara lain :

a. *One to One Relationship*

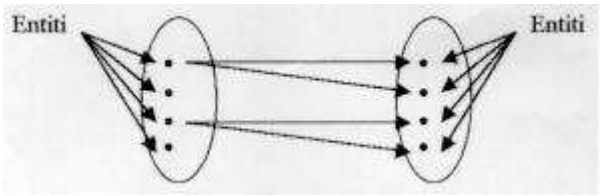
Hubungan antara file pertama dan file kedua adalah satu berbanding satu pada **Gambar 2.2.**



Gambar 2.2 Relasi satu ke satu

b. *One to Many Relationship*

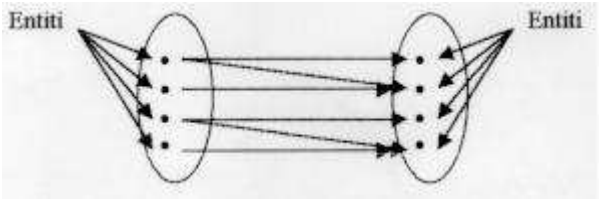
Hubungan antara file pertama dan file kedua adalah satu berbanding banyak atau dapat pula dibalik menjadi banyak lawan satu seperti pada **Gambar 2.3.**



Gambar 2.3 Relasi satu ke banyak

c. *Many to Many Relationship*

Hubungan antara file pertama dan file kedua adalah banyak berbanding banyak pada **Gambar 2.4.**



Gambar 2.4 Relasi banyak ke banyak

2.4.2 Normalisasi

Normalisasi merupakan peralatan yang digunakan untuk melakukan proses pengelompokkan data menjadi tabel-tabel yang menunjukkan entitas dan relasinya.

Dalam proses normalisasi, persyaratan sebuah tabel masih harus dipecah didasarkan adanya kesulitan kondisi pengorganisasian data seperti untuk menambah atau menyisipkan, menghapus atau mengubah, serta pembacaan data dari tabel tersebut. Bila masih ada kesulitan, maka tabel harus dipecah menjadi beberapa lagi, dan dilakukan proses normalisasi kembali sampai diperoleh tabel yang optimal.

Secara umum proses normalisasi dibagi dalam tiga tahap, yaitu tahap tidak normal (Unnormal), normalisasi tahap pertama, normalisasi tahap kedua dan normalisasi tahap ketiga. Di bawah ini akan dijelaskan mengenai ketiga bentuk tersebut dan akan dimulai dengan bentuk tidak normal.

Normalisasi adalah proses yang berkaitan dengan model data relational untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat. Hasil dari proses normalisasi adalah himpunan-himpunan

data dalam bentuk normal (normal form). Ada beberapa bentuk normal, yaitu :

1. Bentuk Tidak Normal (*Unnormalized Form*)

   Bentuk ini merupakan rancangan awal dari pembuatan suatu database. Pada Tahap ini, semua data yang ada direkam tanpa format tertentu. Data bisa jadi mengalami duplikasi. Suatu relasi memenuhi 1-NF jika dan hanya jika setiap atribut dari relasi tersebut hanya memiliki nilai tunggal dalam satu baris atau record seperti pada **Gambar 2.5.**

StudentID	Student	CourseID
61521	Jones	MAT231, ECO220, HST211
61300	Devi	HST211
61425	Tony	ENG202, MAT231
61230	Paula	MAT231, ENG202

Gambar 2.5 Bentuk tidak Unnormalized Form (Non 1-NF table)

2. Bentuk Normal Pertama (*1NF*)

   Suatu tabel dikatakan dalam bentuk normal pertama jika dan hanya jika setiap atribut bernilai tunggal (*Atomic Value*) untuk setiap barisnya. Suatu relasi memenuhi 2-NF jika dan hanya jika :

   a. Memenuhi 1-NF.
   b. Setiap atribut yang bukan kunci utama tergantung secara fungsional terhadap semua atribut kunci dan bukan hanya sebagian atribut.

   Jika suatu relasi memenuhi 1-NF dan relasi tersebut memiliki tepat satu atribut yang membentuk kunci utama, maka relasi tersebut memenuhi 2-NF.

   Rasionalisasi 2-NF :

   a. Memiliki semantik yang lebih eksplisit dari 1-NF.

b. Mencegah beberapa kondisi anomali dalam update data pada **Gambar 2.6**

StudentID	Student	CourseID
61521	Jones	MAT231
61521	Jones	ECO220
61521	Jones	HST211
61300	Devi	HST211
61425	Tony	ENG202
61425	Tony	MAT231
61230	Paula	MAT231
61230	Paula	ENG202

Gambar 2.6 Bentuk 1-NF table

3. Bentuk Normal Kedua (*2NF*)

Suatu tabel dikatakan dalam bentuk normal kedua jika dan hanya jika berada pada bentuk normal pertama dan semua atribut bukan kunci memiliki dependensi sepenuhnya terhadap kunci primer.

Suatu relasi memenuhi 2-NF jika dan hanya jika :

a. Memenuhi 1-NF.

b. Setiap atribut yang bukan kunci utama tergantung secara fungsional terhadap semua atribut kunci dan bukan hanya sebagian atribut.

Jika suatu relasi memenuhi 1-NF dan relasi tersebut memiliki tepat satu atribut yang membentuk kunci utama, maka relasi tersebut memenuhi 2-NF.

Rasionalisasi 2-NF :

a. Memiliki semantik yang lebih eksplisit dari 1-NF.

b. Mencegah beberapa kondisi anomali dalam update data.

StudentID	Student	BirthDate	Course	CourseID	Credit	Grade	Weight
61521	Jones	12/05/77	Math	MAT231	3	B	3
61521	Jones	12/05/77	Economic	ECO220	3	A	4
61521	Jones	12/05/77	History	HST211	2	B	3
61300	Devi	04/28/78	History	HST211	2	A	4
61425	Tony	11/01/78	English	ENG202	2	C	2
61425	Tony	11/01/78	Math	MAT231	3	B	3
61230	Paula	06/14/77	Math	MAT231	3	B	3
61230	Paula	06/14/77	English	ENG202	2	C	2

Gambar 2.7 Bentuk 2-NF table (satisfying 1-NF).

Ketergantungan fungsional dilakukan untuk :

a. StudentID => Student, BirthDate (SC1).

b. CourseID => Course, Credit (SC2).

c. StudentID, CourseID => Grade (SC3, SC3A).

d. Grade => Weight (SC3B).

StudentID	Student	BirthDate
61521	Jones	12/05/77
61300	Devi	04/28/78
61425	Tony	11/01/78
61230	Paula	06/14/77

SC1 table.

CourseID	Course	Credit
MAT231	Math	3
ECO220	Economic	3
HST211	History	2
ENG202	English	2

SC2 table.

StudentID	CourseID	Grade	Weight
61521	MAT231	B	3
61521	ECO220	A	4
61521	HST211	B	3
61300	HST211	A	4
61425	ENG202	C	2
61425	MAT231	B	3
61230	MAT231	B	3
61230	ENG202	C	2

SC3 table.

Gambar 2.8 Tabel yang memenuhi 2-NF.

StudentID	CourseID	Grade
61521	MAT231	B
61521	ECO220	A
61521	HST211	B
61300	HST211	A
61425	ENG202	C
61425	MAT231	B
61230	MAT231	B
61230	ENG202	C

SC3B table.

Grade	Weight
A	4
B	3
C	2
D	1
E	0

SC3B table.

Gambar 2.9 Tabel yang memenuhi 3-NF.

Akhirnya semua tabel SC1, SC2, SC3A, SC3B berada dalam kondisi 3-NF, sehingga semua databases mengalami kondisi 3-NF.

4. Bentuk Normal Ketiga (*3NF / Third Normal Form*)

   Suatu tabel dikatakan dalam bentuk normal ketiga jika berada pada bentuk normal kedua dan setiap atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci primer.

   Suatu relasi memenuhi bentuk III (3-NF) jika dan hanya jika :

   a. Relasi tersebut memenuhi 2-NF.
   b. Setiap atribut bukan kunci tidak tergantung secara fungsional kepada atribut bukan kunci yang lain dalam relasi tersebut.

   Suatu relasi yang memenuhi 2-Nf dan hanya memiliki satu atribut bukan kunci selalu memenuhi 3-NF.

   Kegunaan normalisasi :

a. Meminimasi pengulangan informasi.

b. Memudahkan indentifikasi entiti / obyek.

2.4.3 Relasi Tabel

Di dalam sebuah database, setiap tabel memiliki sebuah field yang memiliki nilai unik untuk setiap baris. Field ini ditandai dengan icon bergambar kunci di depan namanya. Baris-baris yang berhubungan pada tabel mengulangi kunci primer (*primary key*) dari baris yang dihubungkannya pada tabel lain. Salinan dari kunci primer di dalam tabel-tabel yang lain disebut dengan kunci asing (*foreign key*). Kunci asing ini tidak perlu bersifat unik, dan semua field bisa menjadi kunci asing. Yang membuat sebuah field merupakan kunci asing adalah jika dia sesuai dengan kunci primer pada tabel lain.

2.4.4 Struktur File (*file structure*)

Struktur *file* merupakan struktur dari perancangan *database* yang akan digunakan, *file-file* disusun berdasarkan kelas datanya agar dapat memudahkan dalam penyimpanan data.

2.5 Perancangan (Rancang Bangun) Program

Perancangan program dimaksudkan untuk memudahkan pemrogram komputer dalam melaksanakan tugasnya. Pada langkah ini terdapat empat bagian,

yaitu perancangan *input / output*, pengkodean, struktur menu dan kebutuhan sistem.

2.5.1 Perancangan Masukan/Keluaran (*Input / Output*)

a. Perancangan Masukan (*Input*)

   Perancangan *input* merupakan awal dimulainya proses informasi. Pada perancangan input ini dilakukan tahapan perancangan *interface* dari program, yaitu bagimana merancang bentuk dari program sistem informasi yang mudah untuk diakses, cepat dan memperhitungkan efisiensi waktu dan hardware komputer pengguna.

b. Perancangan Keluaran (*Output*)

   Keluaran (*output*) merupakan produk dari sistem informasi yang dapat dilihat. dilakukan perancangan output sistem, bagaimana mendesain sebuah output yang didapat dari beberapa proses pengeditan data yang ada. Output ini berupa tampilan dalam bentuk *paper / printer* output. Contoh output adalah faktur, check, tanda terima pembayaran dan sebagainya.

2.5.2 Pengkodean (*coding*)

Pengkodean digunakan untuk tujuan mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat dibentuk dari kumpulan angka, hurut dan karakter-karakter khusus (misalnya %, /, -, $, #, &, :, dan

sebagainya). Angka merupakan simbol yang banyak digunakan pada sistem pengkodean.

Ada beberapa hal yang perlu diperhatikan dalam pengkodean, yaitu :

a. Harus mudah diingat.

   Supaya kode mudah diingat, maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan obyek yang diwakilnya.

b. Harus unik.

   Kode harus unik untuk masing-masing item yang diwakilinya. Unik berarti tidak ada kode yang kembar.

c. Harus fleksibel.

   Kode harus fleksibel sehingga memungkinkan perubahan-perubahan atau penambahan item baru tetap dapat diwakili oleh kode.

d. Harus efisien.

   Kode harus sependek mungkin, selain mudah diingat juga akan efisien bila direkam di simpanan luar komputer.

e. Harus konsisten.

   Bilamana mungkin, kode harus konsisten dengan kode yang telah dipergunakan.

f. Harus distandarisasi.

   Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi. Kode yang tidak standar akan mengakibatkan kebingungan, salah

pengertian dan dapat cenderung terjadi kesalahan pemakaian bagi yang menggunakan kode tersebut.

g. Spasi dihindari.

   Spasi di dalam kode sebaiknya dihindari, karena dapat menyebabkan kesalahan di dalam menggunakannya.

h. Hindari karakter yang mirip.

   Karakter-karakter yang hampir serupa bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode.

i. Panjang kode harus sama.

   Masing-masing kode yang sejenis harus mempunyai panjang yang sama.

2.5.3 Struktur Menu

Menu banyak digunakan dalam dialog karena merupakan jalur pemakai (*user interface*) yang mudah dipahami dan mudah digunakan. Menu berisi dengan beberapa alternatif atau pilihan yang disajikan kepada *user*. *User* dapat memilih pilihan di menu dengan cara menekan tombol angka atau huruf yang dihubungkan dengan pilihan tersebut. Jika pilihan dari menu terlalu banyak, maka dapat diorganisasikan secara berjenjang.

2.5.4 Kebutuhan Sistem

Kebutuhan-kebutuhan sistem (*systems requirements*) yang harus di perhatikan dalam merancang sistem informasi antara lain :

1. Kebutuhan Perangkat Lunak (*Software Requirements*)

   Sistem harus disesuaikan dengan ketersediaan perangkat lunak yang digunakan oleh perusahaan dan dapat mengikuti perubahan (*upgrade*) perangkat lunak yang dilakukan oleh perusahaan.

2. Kebutuhan Perangkat Keras (*Hardware Requirements*)

   Sistem harus disesuaikan dengan ketersediaan perangkat keras yang digunakan oleh perusahaan dan dapat mengikuti perubahan (*upgrade*) perangkat keras yang dilakukan oleh perusahaan.

2.5 Metode Analytical Hierarcy Process (AHP)

2.5.1 Sistem Pendukung Keputusan (Decisions Support System)

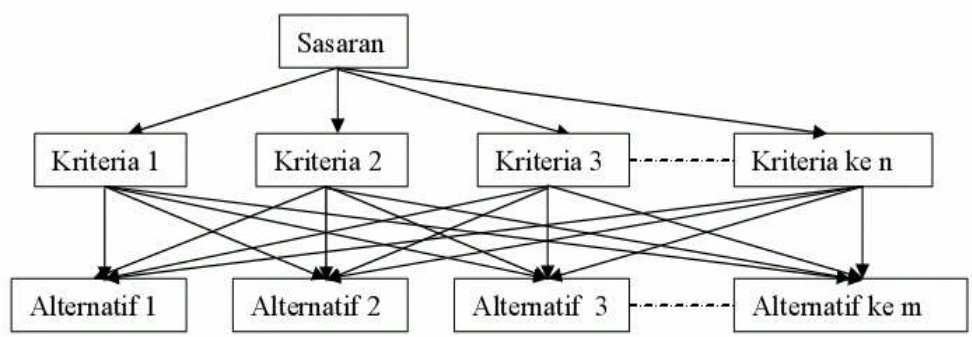
Dewasa ini perkembangan teknologi informasi sudah sedemikian pesat. Perkembangan yang pesat tidak hanya teknologi perangkat keras dan perangkat lunak saja, tetapi metode komputasi juga ikut berkembang. Salah satu metode komputasi yang cukup berkembang saat ini adalah metode sistem pendukung keputusan (*Decisions Support System*).

Dalam teknologi informasi, sistem pengambilan keputusan merupakan cabang ilmu yang letaknya diantara sistem informasi dan sistem cerdas. Banyak metode yang

dapat digunakan dalam sistem pengambilan keputusan. Salah satu metode tersebut yang digunakan dalam penelitian ini adalah metode *Analytical Hierarchy Process* (AHP). Konsep metode AHP adalah merubah nilai-nilai kualitatif menjadi nilai kuantitatif. Sehingga keputusan-keputusan yang diambil bisa lebih obyektif. Metode AHP mula-mula dikembangkan di Amerika pada tahun 1970 dalam hal perencanaan kekuatan militer untuk menghadapi berbagai kemungkinan (*contingency planning*). [2]

2.5.2 Langkah-Langkah Metode AHP

Metode AHP merupakan salah satu model untuk pengambilan keputusan yang dapat membantu kerangka berfikir manusia. Metode ini mula-mula dikembangkan oleh Thomas L. Saaty pada tahun 70-an. Dasar berpikirnya metode AHP adalah proses membentuk skor secara numerik untuk menyusun rangking setiap alternatif keputusan berbasis pada bagaimana sebaiknya alternatif itu dicocokkan dengan kriteria pembuat keputusan. Adapun struktur hirarki AHP ditampilkan pada **Gambar 2.10.**



Gambar 2.10 Struktur AHP

Adapun langkah-langkah metode AHP adalah :

1. Menentukan jenis-jenis kriteria penilaian.
2. Menyusun kriteria-kriteria tersebut dalam bentuk matriks berpasangan.
3. Menjumlah matriks kolom.
4. Menghitung nilai elemen kolom kriteria dengan rumus masing-masing elemen kolom dibagi dengan jumlah matriks kolom.
5. Menghitung nilai prioritas kriteria dengan rumus menjumlah matriks baris hasil langkah ke-4 dan hasilnya 5 dibagi dengan jumlah kriteria
6. Menentukan alternatif-alternatif yang akan menjadi pilihan.
7. Menyusun alternatif-alternatif yang telah ditentukan dalam bentuk matriks berpasangan untuk masing-masing kriteria. Sehingga akan ada sebanyak *n* buah matriks berpasangan antar alternatif.
8. Masing-masing matriks berpasangan antar alternatif sebanyak *n* buah matriks, masing-masing matriksnya dijumlah per kolomnya.
9. Menghitung nilai prioritas alternatif masing-masing matriks berpasangan antar alternatif dengan rumus seperti langkah 4 dan langkah 5.
10. Menguji konsistensi setiap matriks berpasangan antar alternatif dengan rumus masing-masing elemen matriks berpasangan pada langkah 2 dikalikan dengan nilai prioritas kriteria. Hasilnya masing-masing baris dijumlah, kemudian hasilnya dibagi dengan masing-masing nilai prioritas kriteria sebanyak $\alpha 1, \alpha 2, \alpha 3, \ldots, \alpha n$
11. Menghitung Lamda max dengan rumus langkah ke 4 dan hasilnya 5 dibagi dengan jumlah kriteria.

$$\lambda \text{ max} = \frac{\sum \lambda}{n} \qquad \textbf{(2.1)}$$

12. Menghitung CI (*Consistency Index*) dengan rumus

$$CI = \frac{\lambda \text{ max-n}}{n - 1} \qquad \textbf{(2.2)}$$

13. Menghitung CR (*Consistency Ratio*) dengan rumus

$$CR = \frac{CI}{RC} \qquad \textbf{(2.3)}$$

dimana RC (*Random Consistency*) adalah nilai yang berasal dari tabel random di **Tabel 2.3.** [2]

Tabel 2.3 Random Consistency

n	1	2	3	4	5	6	7	8	9	10	11
RC	0,00	0,00	0.58	0,90	1,12	1,24	1,32	1,41	1,45	1,49	1,51

Sumber : Saaty, 1986.

Jika CR < 0,1 maka nilai perbandingan berpasangan pada matriks kriteria yang diberikan konsisten. Jika CR > 0,1 maka nilai perbandingan berpasangan pada matriks kriteria yang diberikan tidak konsisten. Sehingga jika tidak konsisten, maka pengisian nilai-nilai pada matriks berpasangan pada unsur kriteria maupun alternatif harus diulang.

2.6 Sekilas Microsoft Visual Basic 6.0

Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari

pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang berbasis DOS dan tidak mempunyai kemampuan menggunakan metode GUI dalam basis Windows. Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP). [8]

Dengan menggunakan metode Graphical User Interface (GUI), Visual Basic memudahkan pemrogram untuk berinteraksi langsung dengan elemen-elemen untuk setiap bentuk pemrograman. Sebagai program yang berbasis Windows, Visual Basic mempunyai kemampuan untuk berinteraksi dengan seluruh aplikasi Windows, seperti Microsoft Word, Microsoft Excel, Microsoft Acces, dan sebagainya.

Visual Basic ialah bahasa pemrograman yang bersifat *event-driven.* Event driven artinya program menunggu sampai adanya respon dari pemakai berupa kejadian tertentu, misalnya tombol diklik, atau menu dipilih. Ketika event terdeteksi, event yang berhubungan akan melakukan aksi sesuai dengan kode yang diberikan.

Visual basic merupakan bahasa pemrograman tercepat dan termudah untuk membuat suatu aplikasi dalam Microsoft Windows. Dari sekian banyak bahasa pemrograman visual yang ada saat ini, Visual Basic merupakan salah satu yang termudah untuk dipelajari dan handal.

2.7 Sekilas Microsoft SQL Server 2000

Microsoft SQL Server 2000 adalah salah satu nama *database* yang paling populer di kalangan pengembang perangkat lunak. Terlepas dari keunggulan strategi pemasaran dari perusahaan Microsoft atau bukan, yang jelas *database* ini mempunyai banyak pendukung, terbukti dari banyaknya tenaga profesional yang menguasai dan mampu mengoperasikan *database* ini dengan baik. Para administrator *database* dan *programmer* tidak menemui kesulitan sama sekali untuk menemukan literatur dan dokumentasi baik berupa buku maupun informasi di internet mengenai *database* ini dengan berbagai macam topik pembahasan. Para pembuat *software* pihak ketiga juga berlomba-lomba membuat berbagai macam program yang mendukung pengoperasian *database* ini seperti *software backup, replikasi, profiler, audit trail, database modeler, debugger, business intelligence*, dan sebagainya. Bahkan tidak sedikit perusahaan pembuat software besar yang menggunakan SQL Server sebagai *database* utamanya, seperti Great Plains dan Siebel yang terkenal dengan aplikasi ERP dan CRM-nya

SQL Server memiliki sistem berarsitektur terbuka yang memungkinkan para pengembang program memperluas dan menambahkan fungsi-fungsi ke dalam *database* tersebut. Microsoft telah menyediakan *Software Development Kit* (SDK) bersama dokumentasi yang lengkap untuk membuka kesempatan yang luas kepada para programmer untuk mempelajari dan memanfaatkan komponen dan obyek di dalam SDK tersebut guna membuat aplikasi-aplikasi secara elegan.