

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Pada dasarnya, sistem secara umum dapat didefinisikan sebagai suatu totalitas himpunan bagian-bagian yang satu sama lain saling berhubungandanterbentuk sedemikian rupa sehingga menjadi satu kesatuan yang terpadu untuk mencapai suatu tujuan tertentu. Sistem merupakan bagian terpenting dalam perkembangan ilmu pengetahuan sehingga banyak para ahli mengalihkan perhatian kepada pembelajaran mengenai sistem.

Menurut Djahir dan Pratita (2014:46) “sistem haruslah terdiri atas berbagai komponen/ elemen yang saling berhubungan/ berkaitan sehingga membentuk satu kesatuan yang utuh guna mencapai suatu tujuany yang telah ditentukan”. Sedangkan menurut Hartono (2013:9) “sistem adalah suatu himpunan dari berbagai bagian ata elemen,yang saling berhubungan secara terorganisasi berdasar fungsi-fungsinya, menjadi satu kesatuan”.

Menurut Gaol (2008:9) “sistem adalah hubungan satu unit dengan unit-unit yang saling berhubungan satu sama lainnya dan yang tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan. Apabila satu unit macet/terganggu, unit lainnya pun akan terganggu untuk mencapai tujuan yang ditetapkan tersebut”.

Dari beberapa pendapat diatas, penulis dapat menyimpulkan bahwa sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sistem didalam hidup tidak hanya digunakan sebagai alat bagi manajemen tetapi juga bisa digunakan sebagai senjata, pola berfikir dan yang penting lagi sistem bisa dijual.

2.1.1 Karakteristik Sistem

Suatu sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu.

Karakteristik sistem terdiri dari beberapa, antara lain:

1. Komponen (*components*). Bagian-bagian atau elemen-elemen, yang dapat berupa benda atau manusia, berbentuk nyata atau abstrak, dan disebut subsistem.
2. Penghubung antarbagian (*interface*). Sesuatu yang bertugas menjembatani satu bagian dengan bagian lain, dan memungkinkan terjadinya interaksi/komunikasi antarbagian.
3. Batas (*boundary*). Sesuatu yang membedakan antara satu sistem dengan sistem atau sistem-sistem lainnya.
4. Lingkaran (*enviroment*). Segala sesuatu yang berada diluar sistem dan dapat bersifat menguntungkan atau merugikan sistem yang bersangkutan.
5. Masukan (*input*). Sesuatu yang merupakan bahan untuk diolah atau diproses oleh sistem.
6. Mekanisme pengolahan data (*processing*). Perangkat dan prosedur untuk mengubah masukan menjadi keluaran dan menampilkannya.
7. Keluaran (*output*). Berbagai macam bentuk hasil atau produk yang dikeluarkan dari pengolahan.
8. Tujuan (*goal/objective*). Sesuatu keadaan yang ingin dicapai oleh sistem, baik dalam jangka pendek maupun jangka panjang.

9. Sensor dan kendali (*sensor & control*). Sesuatu yang bertugas memantau dan menginformasikan perubahan-perubahan didalam lingkungan dan dalam diri sistem kepada sistem.
10. Umpan-balik (*feedback*). Informasi tentang perubahan-perubahan lingkungan dan perubahan-perubahan (penyimpangan) dalam diri sistem.

2.1.2 Klasifikasi Sistem

Menurut Ladjammuddin (2013:6) “sistem merupakan bentuk integrasi anatara satu komponen dengan komponen lainnya. Karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada didalam sistem tersebut”. Oleh karena itu sistem dapat diklasifikasikan kedalam beberapa sudut pandang. Seperti contoh sistem yang bersifat abstrak, sistem alamiah, sistem yang bersifat *deterministic* dan sistem yang bersifat terbuka, adapun penjelasan lebih detail dan rinci akan dipaparkan dibawah ini:

1. Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologi, yaitu sistem yang berupa pemikiran-pemikiran hubungan manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi karna proses alam tidak dibuat oleh manusia (ditentukan dan tunduk kepada kehendak sang pencipta alam). Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem manusia yang melibatkan interaksi manusia dengan mesin disebut *human-machine system* atau ada yang menyebut

dengan *man machine system*. Sistem informasi merupakan contoh *man-machine system*. Karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksidiantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem tertentu relatif stabil/konstan dalam jangka waktu yang lama. 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. Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyatannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lain”.

2.1.3 Sistem Informasi

Menurut Ladjamuddin (2013:13) “sistem informasi dapat didefinisikan sebagai berikut :

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/ atau untuk mengendalikan organisasi.
3. Suatu sistem didalam suatu organisasi yang mempertemukankebutuhan pengolahan transaksi, mendukung operasi,bersifat manajerial,dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertetu dengan laporan-laporan yang diperlukan”.

2.1.4 Sitem Database

Menurut Mesran (2009:141) “*database* dapat diartikan sebagai kumpulan data yang terdiri atas satu atau lebih tabel yang terintegrasi satu sama lain, di mana setiap pemakai (*user*) diberi wewenang untuk dapat mengakses (mengubah, menghapus, menganalisis, menambah, memperbaiki) data dalam tabel tersebut”.

Aplikasi *basisdata* yang penulis gunakan dalam Tugas Akhir ini adalah *PhpMyAdmin*. Menurut Hikmah, Deddy Supriadi dan Tuti alawiyah (2015:2)”*PhpMyAdmin* merupakan aplikasi yang dapat digunakan untuk membuat database, pengguna (*user*), memodifikasi tabel, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*).. Sedangkan menurut Sibero (2013:376) “aplikasi yang dibuat oleh *phpmyadmin.net*. *phpmyadmin* digunakan untuk administrasi data base *MySQL*”.

Dari beberapa pendapat diatas penulis dapat menyimpulkan bawah *PhpMyAdmin* adalah aplikasi pengolah data yang digunakan untuk pembuatan *database*.

2.1.5 Bahasa Pemrograman

Dalam penulisan Tugas Akhir ini, penulis menggunakan bahasa pemrograman, bahasa ini dapat digunakan oleh programmer untk menentukan secara persisi data mana yang akan diolah oleh komputer. Adapun bahasa pemrograman yang penulis terapkan dalam perancangan sistem pencatatan penjualan untuk CV. Mitra Jaya ini menggunakan *Microsoft Visual Basic 6.0*.

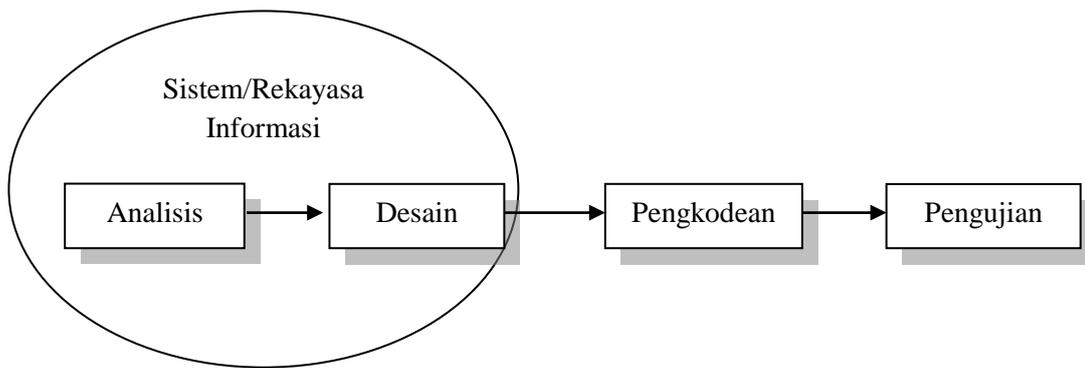
Menurut Fauzi dan Miftakul Amin(2012:1)“ *Visual Basic 6.0* merupakan bahasa pemrograman yang mudah digunakan untuk pengembangan aplikasi, baik aplikasi kecil maupun aplikasi besar”. Sedangkan menurut Madcoms (2010:2) “Microsoft Visual Basic 6.0 merupakan bahasa pemrogman yang cukup populer dan mudah untuk dipelajari.”

Dari beberapa pendapat diatas penulis dapat menyimpulkan bahwa microsoft visual basic adalah bahasa pemrograman komputer yang mudah dipelajari dan populer.

2.1.6 Model Pengembangan Perangkat Lunak

Untuk pengembangan perangkat lunak, disiplin ilmu yang sering digunakan sebagai dasar pemodelan pengembangan perangkat lunak menggunakan model *waterfall*.

Menurut Rosa dan Salahudin (2014:28) “ model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).



Sumber: Rosa dan Salahudin (2014:29)

Gambar II.1 Ilustrasi Model *Waterfall*

Dari gambar diatas dapat dijelaskan bahawa :

1. Analisi Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada taha ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam progam perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara secara dari segi logik. Dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2 Teori Pendukung

Adapun teori pendukung yang dimaksud untuk merancang model sistem yang baru pada penulisan Tugas Akhir ini adalah :

2.2.1 Diagram Alir Data (DAD)

Menurut Ladjammuddin (2013:63) “DAD menampilkan kegiatan sistem lengkap dengan komponen-komponen yang menunjukkan secara tegas file-file yang dipakai, unsur sumber atau tujuan data, serta aliran data dari satu proses ke proses yang lainnya”.

1. Adapun simbol dari diagram alir data menurut ladjammuddin (2013:67) sebagai berikut :

a. Kesatuan luar (*External Entity*)

Sesuatu yang berada diluar sistem, tetapi ia memberikan data ke dalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kitak notasi.

b. Arus Data (*Data Flow*)

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 diantara proses, data store dan menunjukkan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

c. Proses (*Process*)

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. 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 data keluaran. Proses sering pula disebut bubble.

d. Simpanan data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data pengikat data yang ada dalam sistem. Data store dapat disimbolkan dengan sepasang dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka”.

2. Konsep dasar Diagram Alir Data (DAD)

Diagram Alir Data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Diagram Alir Data dibagi menjadi tiga :

a. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh boundary (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses tidak boleh ada store data diagram konteks.

b. Diagram Nol/Zero (*Overview Diagram*)

Diagram nol adalah diagram yang menggambarkan proses dari dataflow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan eksternal entity.

c. Diagram Rinci (*Level Diagram*)

Diagram rinci adalah diagram yang menguraikan proses apa yang ada didalam diagram zero atau diagram level diatasnya.

2.2.2 *Entity Relationship Diagram (ERD)*

1. Definisi *Entity Relationship Diagram (ERD)*

Menurut Komputer (2008:30) adalah “sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD diatas”.

Sedangkan menurut ladjamudin (2013:142) “satu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”.

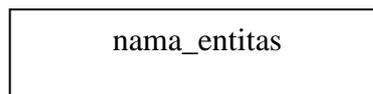
Dari beberapa pendapat diatas penulis dapat menyimpulkan bahwa, ERD adalah konsep yang memetakan susunan data yang disimpan di dalam sistem secara abstrak.

2. Komponen *Entity Relationship Diagram* (ERD)

Simbol-simbol yang digunakan pada ERD menurut ladjamudin adalah :

1. Entitas

Entitas benda yang mempunyai data dan harus disimpan datanya agar dapat diakses oleh perangkat komputer.

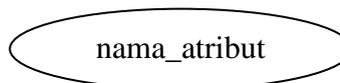


Sumber : Ladjamudin (2013:149)

Gambar II.2 Entitas

2. Atribut

Atribut merupakan *field* data kolom data yang butuh disimpan dalam suatu entitas.

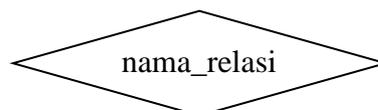


Sumber : Ladjamudin (2013:149)

Gambar II.3 Atribut

3. Relasi

Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.



Sumber : Ladjamudin (2013:149)

Gambar II.4 Relasi

4. Garis atau *Link*

Garis berfungsi sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.



Sumber : Ladjamudi (2013:149)

Gambar II.5 Garis/Link

5. Kardinalitas

Menurut ladjamudi (2013:147) ada 3 macam kardinalitas, yaitu :

a. One to One

Hubungan satu ke satu dinyatakan dengan satu kejadian pada entitas pertama dan mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua atau sebaliknya.

b. One to Many atau Many to One

Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan entitas yang kedua. Sebaliknya untuk satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan entitas yang pertama.

c. Many to Many

Hubungan banyak ke banyak terjadi jika sebuah entitas mempunyai banyak hubungan dengan entitas lainnya. Kejadian tersebut nisa dilihat dari entitas yang pertama.

3. Derajat Relationship (*Relationship Degree*)

Derajat Relationship atau *Relatonship Degree* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.

1. Unary Relationship

Unary Relationship adalah model *relationship* yang terjadi diantara entity yang berasal dari entity set yang sama.

2. *Binary Relationship*

Binary Relationship adalah model *relationship* antara *instance_instance* dari suatu tipe entitas (dua *entity* yang berasal dari entity yang sama).

3. *Tenary Relationship*

Tenary Relationship merupakan *relationship* antara *instance_instance* dari tiga tipe entitas secara sepihak.

4. Atribut

Atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*.

5. Kardinalitas (*Cardinality*)

Kardinalitas Relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain.

2.2.3 Kamus Data

Menurut Ladjamudin (2013:73) “ kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persisi sehingga pemakai dan penganalisi sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses”.

Menurut Rosa dan Shalahuddin (2014:73) kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan

keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)".

Dari beberapa pendapat diatas penulis dapat menyimpulkan bahwa kamus data adalah sebuah kata yang membantu untuk mengartikan aplikasi secara detail dan data yang mengalir pada sistem perangkat lunak sehingga input dan output dapat di pahami secara mudah.

Menurut Iadjudin (2013:71) ada hal-hal yang harus dimuat untuk pembuatan kamus data, antara lain :

1. Nama Arus Data

Kamus data yang dibuat berdasarkan arus data yang mengalir di DAD, serta nama arus data juga harus dicatat dalam kamus data.

2. Alias Data

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

3. Bentuk Data

Bentuk data dipergunakan untuk mengelompokkan kamus data ke dalam kegunaannya sewaktu perancangan sistem.

4. Arus Data

Arus data menunjukkan dari mana data mengalir dan kemana data menuju. Keterangan arus data ini perlu dicatat di kamus data untuk memudahkan mencari arus data di DAD.

5. Penjelasan

Penjelasan dapat diisi dengan keterangan-keterangan arus data tersebut.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, sebagai berikut :

Tabel II.1
Notasi Tipe Data

Notasi	Keterangan
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau...
{ }	N kali diulang/ bernilai banyak
()	Data opsional
...	Batas komentar

Sumber : Rosa dan Shalahudin

2.2.4 Pengkodean (*Pseudocode*)

Menurut Ladjamudim (2013:298) “*Pseudocode* merupakan kode yang mirip dengan kode pemrograman yang sebenarnya. *Pseudocode* berasal dari kata (imitasi atau mirip atau menyerupai), *code* (program)”. Sedangkan menurut Sitorus (2015:22)” Pseudocode adalah kode semu mirip atau menyerupai kode program yang sebenarnya dengan menggunakan suatu bahasa pemrograman tertentu. Pseudocode menggunakan simbo-simbol yang mirip atau menyerupai kode program dengan bahasa pemrograman tertentu”.

Dari beberapa pendapat di atas penulis dapat menyimpulkan bahwa pseudocode adalah cara untuk menuliskan kode atau tanda yang hampir sama dengan kode pemrograman yang sebenarnya dan terdiri dari karakter simbol-simbol yang mudah dipahami.

Menurut Jogiyanto (2009:384) ada beberapa hal yang harus diperhatikan dalam merancang kode, yaitu :

1. Mudah diingat

Supaya kode mudah diingat maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan objek yang diwakili dengan kodenya.

2. Unik

Kode harus unik untuk masing-masing item yang diwakilinya.

3. Fleksibel

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

4. Efisien

Kode harus sesingkat mungkin, selain mudah diingat juga akan efisien direkam dan disimpan diluar komputer.

5. Konsisten

Kode harus konsisten dengan kode yang telah digunakan.

6. Harus di standarisasi

Kode harus di standarisasi untuk seluruh tingkatan dan departemen dalam organisasi.

7. Hindari penggunaan spasi

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

8. Hindari karakter mirip

Karakter-karakter yang hampir mirip bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode. Masing-masing kode yang sejenis harus memiliki panjang kode yang sama.

2.2.5 Kunci Elemen Data (*Key*)

Menurut Ladjamudin (2013:138) “elemen *record* yang dipakai untuk menemukan *record* tersebut pada waktu akses, atau bisa juga digunakan untuk mengidentifikasi setiap *entity/record*/baris”.

Jenis-jenis *key* menurut Ladjamudin (2013:138) sebagai berikut :

1. *Superkey*

Superkey merupakan satu atau lebih atribut dari suatu tabel yang dapat digunakan untuk mengidentifikasi *entity/record* dari tabel tersebut secara unik.

2. *Candidate Key*

Superkey dengan jumlah minimal, disebut dengan *candidate key*. *Candidate key* tidak boleh berisi atribut dari tabel lain, sehingga *candidate key* sudah pasti *superkey* namun belum tentu sebaliknya.

3. *Primary Key*

Disebut *primary key* apabila *key* tersebut lebih natural untuk digunakan sebagai acuan, lebih sederhana dan juga terjamin keunikannya.

4. *Alternate Key*

Setiap atribut dari *candidate key* yang tidak terpilih menjadi *primary key*, maka atribut-atribut tersebut dinamakan *alternate key*.

5. *Foreign Key*

Foreign key merupakan sembarang atribut yang menunjuk pada *primary key* pada tabel yang lain.

6. *External Key*

External Key merupakan suatu himpunan *lexical* yang nilai-nilainya selalu mengidentifikasi suatu *object instance*.

2.2.6 Crystal Report

Menurut Madcoms (2010:234) “program yang terpisah dengan microsoft Visual Basic, tetapi keduanya dapat dihubungkan”. Sedangkan menurut Mesran(2010:173) “Crystal Report merupakan sebuah produk dari seagate yang dideasin khusus untuk membuat sebuah laporan”.

Dari beberapa pendapat diatas penulis dapat menyimpulkan bahwa Crystal report adalah aplikasi pembantu yang mempermudah dan dapat di hubungkan dengan Microsoft Visual Basic untuk membuat laporan.

2.2.7 Hireracy Plus Input-Process-Output (HIPO)

Menurut Ladjamudin (2013:211) “HIPO merupakan sebuah teknik yang digunakan sebagai alat desain dan mendokumentasikan sistem pemrograman yang dibentuk dengan menekankan pada fungsi sistem yang akan mempercepat pencarian prosedur dalam suatu sistem”.

Beberapa manfaat dari penggunaan HIPO menurut Ladjamudin (2013:211) yaitu :

1. Seorang manajer dapat menggunakan HIPO untuk memperoleh gambaran umum sistem.
2. Seorang programer aplikasi dapat menggunakan HIPO untuk menentukan fungsi-fungsi program.

3. Seorang programmer pemeliharaan dapat menggunakan HIPO untuk mencari fungsi-fungsi yang akan dimodifikasi dengan cepat.

Menurut pendapat diatas dapat disimpulkan bahwa *Hirerachy Plus Input Process Output* (HIPO) adalah alat dokumentasi program yang banyak digunakan sebagai alat desain untuk mempercepat pencarian prosedur dalam suatu sistem.