

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem memiliki fungsi untuk menangani fungsi yang secara terus menerus dan berulang-ulang atau yang rutin terjadi. Umumnya sistem dibuat sebagai suatu alat untuk mempermudah dalam pekerjaan manusia.

Dalam sistem ini terdapat komponen yang saling berinteraksi dan bekerja sama dalam membentuk kesatuan untuk melakukan kegiatan atau menyelesaikan suatu tujuan tertentu dan komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2.1.1 Pengertian Sistem

Dengan adanya sistem maka lebih mempermudah dalam melakukan suatu pekerjaan. Secara umum suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lain dan terpadu untuk mencapai tujuan tertentu.

Menurut Davis (1985) dalam Ladjamudin (2013:3) “yang mendefinisikan sistem sebagai bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud”. Menurut Pratama (2014:7) “Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa dapat disimpulkan sistem adalahn sebagian prosedur yang saling berkaitan dan saling terhubung untuk mencapai beberapa sasaran dan maksud tertentu.

A. Karakteristik Sistem

Suatu dapat disebut sistem, jika mempunyai karakteristik tertentu. Sehingga terdapat beberapa faktor dalam karakteristik sistem yaitu komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran, pengolahan sistem, dan sasaran sistem.

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem (Sutabri, 2012:13) yaitu:

a. Komponen Sistem (*Component*)

Suatu sistem yang terdiri dari sejumlah komponen serta saling berinteraksi yang bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa bentuk subsistem. Setiap subsistem memiliki sifat-sifat yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut "supra system".

b. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya, atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang tidak dapat dipisahkan.

c. Lingkungan luar sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan, namun dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan

merupakan energi bagi sistem tersebut, sedangkan lingkungan luar yang merugikan harus dikendalikan, kalau sistem tersebut tidak dikendalikan maka akan mengganggu kelangsungan hidup suatu sistem tersebut.

d. Penghubung Sistem (*interface*)

Media yang menghubungkan antara suatu sistem dengan subsistem yang lainnya disebut dengan penghubung sistem atau *interface*. Penghubung sistem ini memungkinkan sumber daya yang mengalir dari suatu subsistem yang lainnya. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain, dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan Sistem (*input*)

Energi yang dimasukkan kedalam sistem disebut dengan masukan sistem, yang dapat berupa pemeliharaan (*maintance input*) dan sinyal (*signal Input*) sebagai contoh, didalam suatu sistem unit komputer, “program” adalah *maintance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah *signal input* yang akan diolah menjadi sebuah informasi.

f. Keluaran Sitem (*Output*)

Hasil dari sebuah energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna, keluaran dapat merupakan masukan untuk subsistem yang lain seperti sistem informasi.

g. Pengolahan Sistem (*process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

h. Sasaran Sistem (*Objective*)

Sebuah sistem memiliki tujuan atau sasaran yang pasti dan bersifat *deterministic*.

Jadi inti dari pembahasan diatas adalah suatu karakteristik sistem saling berhubungan dalam sistem seperti komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran, pengolahan sistem, sasaran sistem.

B. Klasifikasi Sistem

Sistem dapat diklasifikasikan menjadi beberapa bagian yang perlu kita ketahui agar dapat mendefinisikan sebuah sistem tersebut. Bagian-bagian tersebut yaitu sistem abstrak dan sistem fisik, sistem alamiah dan sistem buatan manusia, sistem deterministik dan sistem probobalistik, sistem terbuka dan sistem tertutup.

Menurut Sutabri (2012:22) “sistem adalah suatu bentuk integrasi antara satu komponen dengan komponen lain karena memiliki sasaran yang berbeda untuk setiap kasus yang terjadi didalam sistem”. Oleh sebab itu, sistem klasifikasi dari beberapa sudut pandang antaranya:

a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.

b. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang dan malam serta pergantian musim. Sedangkan sistem buatan merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut *Human Machine System*. Salah satu contohnya adalah sistem informasi berbasis komputer, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem Determinasi dan Sistem *Probabilistic*

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*. Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilistic*.

d. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya. Sedangkan sistem tertutup adalah sistem yang tidak berhubungan dan tidak dipengaruhi oleh lingkungan luarnya, sistem ini bekerja secara otomatis tanpa adanya campur dari pihak luar.

2.1.2 Bahasa Pemograman

Dalam penulisan TA ini penulis menggunakan bahasa pemograman, bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer. Adapun bahasa pemograman yang dipakai dalam perancangan sistem

informasi laporan keuangan rutin sekolah ini adalah dengan menggunakan *Microsoft Visual Basic 6.0*.

Menurut Kusrini dan Kinoyo (2007:171) “*Visual Basic* adalah salah satu bahasa pemograman komputer. Bahasa pemograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu”.

Menurut Madcoms (2010:2) “*Microsoft visual basic 6.0* merupakan bahasa pemograman yang cukup populer dan mudah dipelajari”. *Microsoft visual basic 6.0* menyediakan fasilitas yang memungkinkan untuk menyusun sebuah program dengan memasang objek-objek grafis dalam sebuah *form*. Selain itu *Visual basic* juga menawarkan berbagai kemudahan dalam mengelola sebuah *database*.

Berdasarkan definisi diatas, dapat disimpulkan bahwa *Microsoft Visual Basic 6.0* merupakan salah satu pemograman komputer yang memungkinkan untuk menyusun sebuah program dengan memasang objek-objek grafis dalam sebuah *form*. Selain itu *Visual basic* juga menawarkan berbagai kemudahan dalam mengelola sebuah *database*.

2.1.3 Basis Data

Basis data kumpulan data yang tersimpan dalam komputer yang digunakan suatu program untuk memperoleh informasi dari basis data tersebut.

Menurut Rosa dan Salahudin (2015:42) “Basis data adalah sistem terkomputerisasi tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Sedangkan menurut Madcoms (2010:186) “*Database* adalah sekumpulan data yang terdiri atas satu

atau lebih tabel yang saling berhubungan antara satu dengan yang lain, dimana Anda atau *User* mempunyai wewenang untuk mengakses data tersebut, baik menambah, mengganti, menghapus, dan mengedit data dalam tabel-tabel tersebut.

Berdasarkan defenisi diatas dapat disimpulkan, bahwa *database* adalah sekumpulan sistem terkomputerisasi yang bisa mengakses untuk menambah, menggantu, menghapus, mengedit data dalam tabel. Adapun basis data yang digunakan dalam Tugas Akhir ini adalah *PhpMyAdmin* .

PhpMyAdmin merupakan salah satu perangkat lunak yang digunakan untuk melakukan pengolahan terhadap basis data, tabel-tabel (*entity*), atribut (*field*) dari *entity*, hubungan (*relation*) antar *entity*, indek, dan lain-lain.

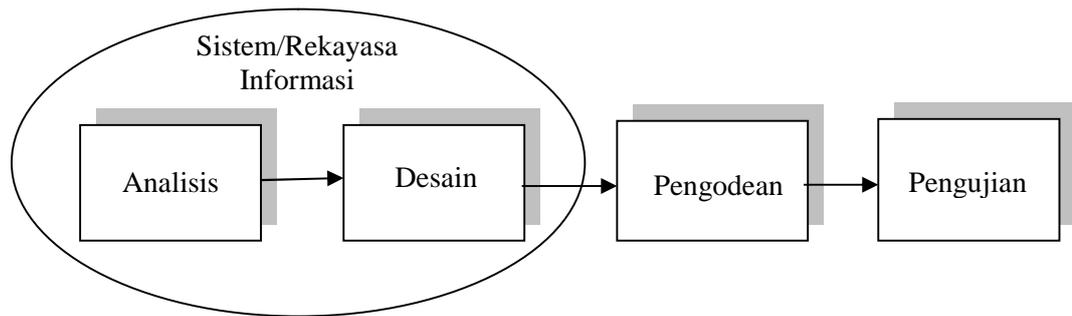
Menurut Rahman (2013:21) “*PhpMyAdmin* adalah sebuah *software* berbasis pemograman PHP yang dipergunakan sebagai administrator *MySQL* melalui *browser (web)* yang digunakan untuk menagemen *database*. Sedangkan menurut Sibero (2013:376) ”*PhpMyAdmin* adalah aplikasi *web* yang dibuat oleh phpmyadmin.net. *phpmyadmin* digunakan untuk administrasi *database MySQL*”.

Berdasarkan pengertian menurut para ahli diatas dapat di simpulkan, bahwa *phpMyAdmin* adalah sebuah *software* berbasis pemograman digunakan untuk administrasi *database MySQL*.

2.1.4 Model Pengembangan Perangkat Lunak

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

Menurut Rosa dan Salahudin (2015:28) mengemukakan bahwa “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 Shalahuddin(2015:29)

Gambar II.1 Ilustrasi Model Waterfall

Dari Gambar II.1 dapat dijelaskan sebagai berikut:

1. Analisa kebutuhan perangkat lunak

Menganalisa suatu kebutuhan si pemakai pada sistem perangkat lunak. Pada tahapan perlu di dokumentasikan.

2. Desain

Menggambarkan suatu sistem yang akan dibuat agar dapat memenuhi semua kebutuhan pengguna (*user*).

3. Pembuatan kode program

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

4. Pengujian

Pengujian terhadap suatu perangkat lunak dilakukan agar semua bagian dipastikan telah diuji. Hal ini dilakukan agar dalam pembuatan sistem tidak terjadi suatu kesalahan yang mengakibatkan sistem tersebut menjadi *error*, dan memastikan bahwa keluaran yang dihasilkan sesuai dengan apa yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Terkadang sebuah perangkat lunak dapat terjadi perubahan setelah dikirim ke pengguna. Sebab adanya kesalahan tersebut dikarenakan sistem tidak terdeteksi saat perangkat lunak beradaptasi dengan lingkungan yang baru.

2.2 Teori Pendukung

Adapun teori pendukung yang digunakan untuk menjadi dasar penelitian tugas akhir ini bersumber pada buku dan jurnal penelitian. Sebagai referensi untuk menunjang atau memperdalam pemahaman terhadap informasi-informasi yang disajikan.

2.2.1 Sistem Informasi

Dalam proses pengembangan sistem informasi, diperlukan adanya pemahaman mengenai konsep konsep dasar dari sistem informasi tersebut. Menurut Sutabri (2012:46) “Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Menurut Ladjamudin (2013:13) “Sistem informasi didefinisikan sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi”.

Berdasarkan pengertian sistem informasi dari para ahli dapat di simpulkan, bahwa sistem informasi diatas adalah suatu sistem di dalam organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.

2.2.2 Laporan Keuangan

Menurut Sugiono dan Untung (2008:3) “Laporan keuangan pada perusahaan merupakan hasil akhir dari kegiatan akuntansi (siklus akuntansi) yang mencerminkan kondisi keuangan dan hasil operasi perusahaan”. Sedangkan menurut Priyatno (2009:89) “Laporan keuangan merupakan laporan yang menginformasikan tentang posisi keuangan perusahaan yang tersusun secara rinci dan lengkap meliputi Neraca, Laba Rugi, Neraca Saldo dan sebagainya”.

Bedasarkan defenisi diatas dapat simpulkan, bahwa laporan keuangan adalah hasil akhir dari kegiatan akuntansi yang tersusun secara rinci dan lengkap.

2.2.3 Diagram Alir Data

Menurut Ladjamudin (2013:64) ”Diagram Alir Data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih besar”. Sedangkan menurut Pratama (2014:49) “Diagram Alir Data adalah diagram pemodelan suatu perangkat lunak, yang mana didalamnya terdapat sejumlah notasi dengan aliran-aliran dari data ke sistem. Adanya aliran data ini menjadikan kita lebih memahami mengenai sistem secara terstruktur dan lebih jelas”.

Berdasarkan pengertian dari para ahli dapat di simpulkan, bahwa Diagram Alir Data merupakan model dari sistem perangkat lunak, menggambarkan pembagian sistem ke modul yang lebih besar.

1. Konsep Dasar Diagram Alir Data

Diagram alir data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Ada beberapa langkah dalam membuat Diagram Alir Data menurut Ladjamudin (2013:64), yaitu:

a. Diagram Konteks

Diagram yang terdiri dari suatu proses dan menggambarkan suatu sistem secara keseluruhan.

b. Diagram Nol

Diagram nol juga menggambarkan pandangan secara menyeluruh dari sistem yang dibuat, namun dalam penjelasannya dilakukan secara lebih terinci.

c. Diagram Detail

Diagram yang menggambarkan arus secara lebih mendetail lagi dari tahapan proses yang ada di dalam diagram nol.

2. Simbol atau lambang yang digunakan dalam membuat diagram alir data yang lazim digunakan, menurut Ladjamudin (2013:67) DAD terdiri dari empat buah simbol yaitu:

a. KesatuanLuar (*External Entity*)

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

b. Arus Data (*Data Flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem.

c. Proses (*process*)

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.

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.

3. Aturan main Digram Alir Data yang baku dan berlaku dalam penggunaan diagram alir data untuk membuat model sistem menurut Ladjamudin (2013:75), hal-hal tersebut adalah :

- a. Arus data tidak boleh dari entitas luar langsung menuju entitas luar lainnya, tanpa melalui suatu proses.
- b. Arus data tidak boleh dari simpanan data langsung menuju ke entitas luar, tanpa melalui suatu proses.
- c. Arus data tidak boleh dari simpanan data langsung menuju ke simpanan data lainnya, tanpa melalui suatu proses.
- d. Arus data dari satu proses langsung menuju proses lainnya, tanpa melalui suatu simpanan data, sebaiknya / sebisa mungkin dihindari.

4. Langkah-langkah mengembangkan DAD menurut Ladjamudin (2013:64) adalah sebagai berikut :

a. Diagram Konteks

Diagram konteks adalah diagram yg terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem.

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

Diagram nol adalah diagram yang menggambarkan proses dari data *flow* diagram.

c. Diagram Rinci (*Level Diagram*)

Diagram rinci adalah diagram yang mengurai proses apa yang ada dalam diagram nol atau diagram level di atasnya.

2.2.4 Entity Relationship Diagram (ERD)

Dalam perancangan basis data, ERD merupakan teknik pemodelan yang sering digunakan dalam mengembangkan basis data yang sedang dirancang.

1. Definisi Entity Relationship Diagram (ERD)

Menurut Ladjamudin (2013:142) “ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak”. ERD ini berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem.

Menurut Rosa dan Salahudin (2015:50) “ERD adalah bentuk paling awal dalam melakukan perancangan basis data resional. Jika menggunakan OODMBS maka perancangan ERD tidak diperlukan”.

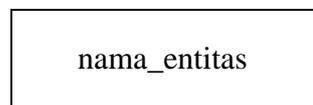
Berdasarkan pengertian dari para ahli di atas dapat disimpulkan, bahwa ERD adalah suatu model jaringan dalam melakukan perancangan basis data resional.

2. Komponen *Entity Relationship Diagram* (ERD)

simbol-simbol yang digunakan pada ERD adalah sebagai berikut :

1. Entitas

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

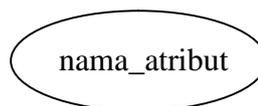


Sumber: *Ladjamudin(2013:149)*

Gambar II.6 Entitas

2. Atribut

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



Sumber: *Ladjamudin(2013:149)*

Gambar II.2 Atribut

3. Relasi

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



Sumber: *Ladjamudin(2013:149)*

Gambar II.3 Relasi

1. Garis atau *Link*

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



Sumber: *Ladjamudin(2013:149)*

Gambar II.4 Garis/*Link*

5. Kardinalitas

Kardinalitas adalah relasi yang meunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas yang lain. Ada 3 macam kardinalitas menurut Ladjamudin (2013:147), 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 bisa dilihat dari entitas yang pertama maupun entitas yang kedua.

3. Derajat *Relationship* (*Relationship Degree*)

Relationship degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*.

a. *Unary relationship*

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

b. *Binary relationship*

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

c. *Ternary Relationship*

Ternary relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sepihak.

d. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*.

e. Kardinalitas (*Cardinality*)

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

2.2.5 Kamus Data

Menurut Rosa dan Shalahuddin (2013:73) “kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Sedangkan menurut Ladjamudin (2013:70) “kamus data berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses”.

Berdasarkan pengertian dari para ahli di atas dapat disimpulkan, bahwa kamus data adalah kumpulan dasar elemen data yang membantu untuk mengartikan aplikasi secara detail dan mengorganisir semua elemen data yang digunakan.

Ada beberapa hal yang harus dimuat dalam pembuatan kamus data menurut Ladjamudin (2013:71) 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 yang lainnya.

3. Bentuk Data

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

4. Arus Data

Arus data menunjukkan darimana 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 tentang arus data tersebut.

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

Tabel II. Notasi Tipe Data

Nota	Keterangan
X	Setiap karakter
9	Angka <i>numeric</i>
A	Karakter alphabet
Z	Angka nol tampilkan sebagai spasi kosong
.	Titik sebagai penulisan ribuan
,	Koma sebagai pemecah pecahan
-	<i>Hypen</i> sebagai tanda penghubung

Sumber : Yakub (2012 : 169)

Tabel II.2 Notasi Struktur Data

Notasi	Keterangan
=	Terdiri
+	Dan(and)
()	Pilih (ya atau tidak)
{ }	Pengulangan proses (Iterasi)
[]	Pilih salah atau pilihan
	Pemisah pilihan didalam tanda []
*	Keterangan atau catatan
@	Petunjuk

Sumber : Yakub (2012 : 170)

2.2.6 Pengkodean (*Pseudocode*)

Pengkodean sering digunakan dalam perancangan suatu program dimana kode yang dibuat tersusun dari aturan-aturan yang dirancang berdasarkan elemen-elemen tertentu yang digunakan oleh perancang.

Menurut Sutabri dalam Puspitawati dan Anggraini (2011:96) “sistem pengkodean terdiri dari himpunan karakter simbol-simbol yang dapat diterima dan telah dinyatakan digunakan untuk mengidentifikasi objek tertentu”. Sedangkan menurut Ladjamudin (2013:298) “*Pseudocode* merupakan kode yang mirip dengan kode pemrograman yang sebenarnya. *Pseudocode* berasal dari kata (imitansi atau mirip atau menyerupai), dan *code* (program)”.

Berdasarkan defenisi diatas dapat disimpulkan, bahwa pengkodean atau *pseudocode* adalah kode yang mirip dengan kode pemrograman yang sebenarnya terdiri dari himpunan karakter simbol-simbol yang dapat diterima dan telah dinyatakan digunakan untuk mengidentifikasi objek tertentu

Menurut Jogiyanto (2009:384) ada beberapa 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 atau penambahan *item* baru dapat tetap diwakili oleh kodenya.

4. Efisien

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

5. Konsisten

Kode harus konsisten dengan kode yang telah digunakan.

6. Harus distandarisasi

Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi.

7. Hindari Penggunaan Spasi

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

8. Hidari 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.

2.2.7 Kunci Elemen Data (*Key*)

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

Adapun jenis-jenis *key* menurut Ladjamudin (2013:138) adalah 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 atribut minimal, disebut dengan *candidate key*. *Candidate key* tidak boleh berisi atribut dari tabel yang 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*.

2.2.8 Hierarchy Plus Input-Process-Output (HIPO)

Menurut Fatta (2007:147) “HIPO merupakan tehnik untuk mendokumentasikan pengembangan suatu sistem yang dikendalikan oleh IBM”. Menurut Ladjamudin (2013:211) “HIPO merupakan sebuah teknik yang digunakan sebagai alat desain dan mendokumentasikan sistem pemrograman yang dibentuk dengan menekankan pada fungsi-fungsi sistem yang akan mempercepat pencarian prosedur dalam suatu sistem”.

Adapun beberapa manfaat dari penggunaan HIPO menurut Ladjamudin (2013:211) antara lain sebagai berikut:

1. Seorang manajer dapat menggunakan HIPO untuk memperoleh gambaran umum sistem.
2. Seorang programmer 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.