

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Laporan Akhir di Teknik Komputer**

Mahasiswa perlu melakukan konsultasi atau bimbingan dalam pengerjaan laporan akhir kepada dosen pembimbing yang telah ditunjuk oleh ketua program studi Jurusan Teknik Komputer Polstri agar dapat membantu mahasiswa dalam mengerjakan Laporan Akhir menjadi lebih terarah dan lebih fokus sehingga hasil yang diharapkan dapat tercapai.

Penyusunan laporan akhir bertujuan memantapkan wawasan pengetahuan dan pengalaman mahasiswa dalam melakukan kegiatan ilmiah, dengan cara memberi kesempatan memformulasikan ide, konsep, pola berpikir, dan kreativitas yang dikemas secara komprehensif, dan mengomunikasikan dalam format yang lazim digunakan di kalangan masyarakat ilmiah (Anik Ghufron, dkk, 2013: 2).

Di Jurusan Teknik Komputer Polstri terdiri dari beberapa tahap, yaitu:

- a. Pengajuan proposal oleh mahasiswa kepada Ketua Program Studi.
- b. Penunjukkan dosen pembimbing oleh Ketua Program Studi.
- c. Proses bimbingan laporan akhir mahasiswa dan dosen pembimbing.

#### **2.2 Sistem Informasi**

Menurut Kenneth C Laudon dan Jane P Laudon (2008:15), Sistem informasi adalah sekumpulan komponen yang saling berhubungan, mengumpulkan (atau mendapatkan), memproses, menyimpan, dan mendistribusikan informasi untuk menunjang pengambilan keputusan dan pengawasan dalam suatu organisasi. Menurut Aprillita Dwiyani (2013: 2), sistem informasi adalah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, membantu dan mendukung kegiatan operasi, bersifat manajerial dari suatu organisasi dan membantu mempermudah penyediaan laporan yang diperlukan.

Al Fatta (2007: 10) mengemukakan bahwa komponen-komponen yang menyusun suatu sistem informasi terdiri dari:

- a. Perangkat keras, yaitu perangkat keras komponen untuk melengkapi kegiatan memasukan data, memroses data, dan keluaran data.
- b. Perangkat lunak, yaitu program dan instruksi yang diberikan ke komputer.
- c. Database, yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.

- d. Telekomunikasi, yaitu komunikasi yang menghubungkan antara pengguna sistem dengan sistem komputer secara bersama-sama ke dalam suatu jaringan kerja yang efektif.
- e. Manusia, yaitu personel dari sistem informasi, meliputi admin, analis, programer dan operator.

Al Fatta (2007: 12) mengemukakan bahwa sistem informasi dapat dibagi menjadi beberapa tipe yaitu:

a. Sistem Pemrosesan Transaksi (*Transaction Processing System*)

*Transaction Processing System* atau Sistem Pemrosesan Transaksi adalah sistem informasi terkomputerisasi yang dikembangkan untuk memroses sejumlah data untuk transaksi bisnis rutin.

b. Sistem Informasi Manajemen (SIM)

Sistem Informasi Manajemen adalah sebuah sistem informasi yang berfungsi untuk membantu perencanaan, pengendalian, dan pengambilan keputusan dengan menyediakan resume rutin dan laporan-laporan tertentu.

c. Sistem Penunjang Keputusan (*Decision Support System*)

*Decision Support System* atau Sistem Penunjang Keputusan adalah sistem informasi yang berfungsi mengombinasikan data dan model analisis canggih atau peralatan data analisis untuk mendukung pengambilan keputusan yang semiterstruktur maupaun tidak terstruktur.

d. Sistem Pakar (*Expert System*)

*Expert System* atau Sistem Pakar adalah representasi pengetahuan yang menggambarkan cara seorang ahli dalam mendekati suatu masalah.

### 2.3 Basis Data

Basis data terdiri dari dua kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang tempat bersarang dan berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya (Syahril Akbar, 2010: 43).

Menurut Firman Kemal P., (2010: 22-23), basis data dapat didefinisikan dalam sejumlah sudut pandang, diantaranya:

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- c. Kumpulan file/tipe/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

#### **2.4 Model Basis Data Relasional**

Abdul Kadir (2003: 270-276) mengemukakan bahwa model basis data adalah sekumpulan konsep terintegrasi yang dipakai untuk menjabarkan data, hubungan antar data, dan kekangan terhadap data yang digunakan untuk menjaga konsistensi. Model data terbagi menjadi 4 macam, yaitu:

- a. Model data hirarkis.
- b. Model data jaringan.
- c. Model data relasional.
- d. Model data berbasis objek.

Model data relasional merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami oleh pengguna. Model ini menggunakan sekumpulan tabel dengan masing-masing tabel tersusun atas sejumlah baris dan kolom. Pada model data relasional, kaitan antara 2 buah tabel disebut relasi. Halhal yang perlu diperhatikan dalam menyusun tabel-tabel dalam basis data relasional adalah sebagai berikut:

- a. **Atribut Kunci**

Penentuan kunci merupakan hal yang paling penting dalam basis data relasional. Kunci sebagai metode untuk mengakses suatu baris tertentu sekaligus menjadi pengenal unik (tidak ada yang sama) rekaman di dalam suatu tabel. Kunci dapat berupa sebuah atribut atau gabungan dari beberapa atribut.

Menurut Andri Kristanto (2003: 76-77), atribut kunci bagi menjadi:

1) Kunci kandidat (*Candidat key*)

Kunci kandidat adalah suatu atribut yang mengidentifikasi secara unik dari suatu kejadian yang sifatnya khusus dari suatu entiti.

Kunci kandidat ini tidak ganda.

2) Kunci primer (*Primary key*)

Kunci primer adalah kunci kandidat yang dipilih untuk mewakili setiap kejadian dari suatu entiti.

3) Kunci alternatif (*Alternative key*)

Kunci alternatif adalah kunci kandidat yang tidak dipilih sebagai kunci primer. Kunci alternatif lebih diorientasikan untuk pengurutan rekaman di dalam sebuah tabel atau entiti.

4) Kunci tamu (*Foreign key*)

Kunci tamu adalah kunci primer suatu entiti yang dihubungkan pada entiti lain. Biasanya jika ada dua tabel, kunci primer akan diletakkan pada entiti induk sedangkan pada entiti anak diletakkan kunci tamu.

b. Normalisasi

Menurut Kronke (1992) dalam Abdul Kadir (1999: 65) mengemukakan bahwa normalisasi sebagai proses untuk mengubah suatu tabel yang memiliki masalah tertentu ke dalam 2 tabel atau lebih yang tak memiliki masalah tersebut. Tujuan dari normalisasi adalah untuk mengurangi redudansi dan ketidakkonsistenan data. Aturan-aturan normalisasi dinyatakan dalam istilah bentuk normal. Bentuk normal adalah suatu aturan yang dikenakan pada tabel-tabel dalam basis data. Suatu tabel dikatakan berada dalam bentuk normal, jika memenuhi aturan-aturan pada level normalisasi yang bersesuaian. Menurut Abdul Kadir (2001: 54) aturan-aturan dalam masing-masing normalisasi tersebut adalah sebagai berikut:

1) Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, mungkin saja data tidak lengkap atau data terduplikasi dan data dikumpulkan apa adanya.

2) Bentuk normal pertama

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom.

3) Bentuk normal kedua

Suatu tabel berada dalam bentuk normal kedua (2NF) jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu kunci primer yang sama.

4) Bentuk normal ketiga

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom yang bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

## **2.5 Database Management Sistem (DBMS)**

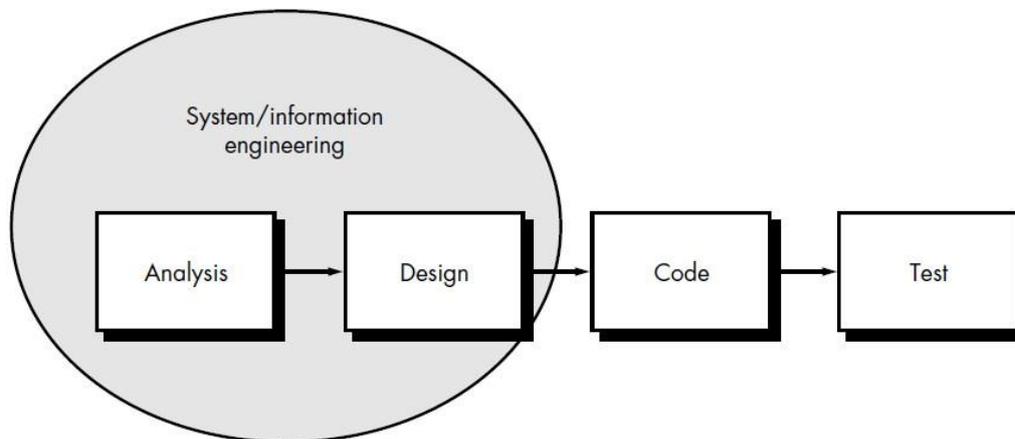
Menurut Syahril Akbar (2010: 46), pengelolaan basis data secara spesifik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus/spesifik. Perangkat inilah yang disebut DBMS yang menentukan bagaimana data diorganisasi, disimpan, diubah, dan diambil kembali. DBMS menerapkan mekanisme pengamanan data secara bersama, pemaksaan keakuratan/ konsistensi data dan sebagainya.

DBMS berfungsi menyediakan sebuah cara untuk menyimpan dan menerima kembali informasi database secara mudah dan efisien. Raharjo (2011: 10) memaparkan macam-macam program DBMS yang saat ini sering digunakan adalah MySQL, Oracle, Interbase/Firebird, IDM DB2.

## **2.6 Model Perancangan Sistem**

Perancangan sistem merupakan suatu gambaran secara umum kepada pengguna/user tentang sistem yang akan dibuat. Desain sistem akan dibuat untuk menggambarkan komponen-komponen sistem informasi yang akan didesain secara rinci. Model perancangan sistem yang akan dipakai menggunakan metode analisis sistem terstruktur atau lebih dikenal dengan model *Waterfall*.

Model *Waterfall* sering disebut dengan “*Classic Life Cycle*”. Ini merupakan model yang muncul pertama kali, sekitar tahun 1970 sehingga sering dianggap kuno tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *Analysis, Desain, Code, Test*.



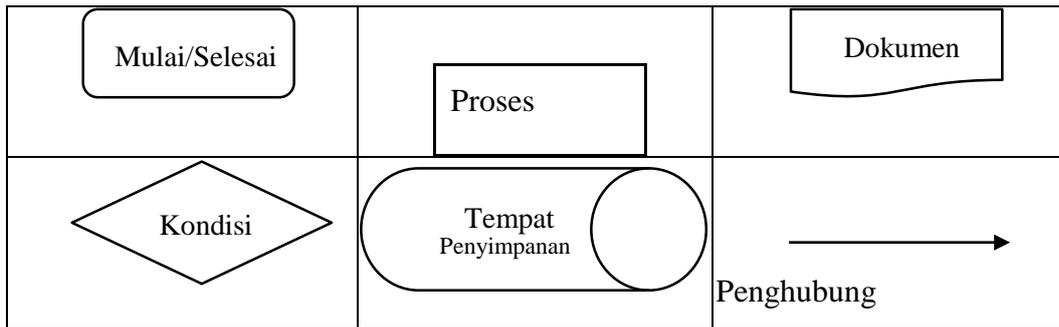
**Gambar 2.1 Model *Waterfall* Presmann**

Gambar 2.1 merupakan tahapan secara umum dari model *Waterfall*. Menurut Pressman (2001: 28) model *Waterfall* diuraikan dengan tahap-tahap sebagai berikut:

- a. *Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan perancangan sistem.
- b. *Design* adalah tahap penerjemah atau tahap perancangan dari keperluankeperluan yang dianalisis dalam bentuk yang lebih mudah dimengerti oleh pemakai.
- c. *Code* adalah tahap implementasi dari hasil sistem yang telah dirancang dalam bahasa pemograman yang telah ditentukan dan digunakan dalam pembuatan sistem.
- d. *Test* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian dilakukan agar fungsi-fungsi dalam sistem bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

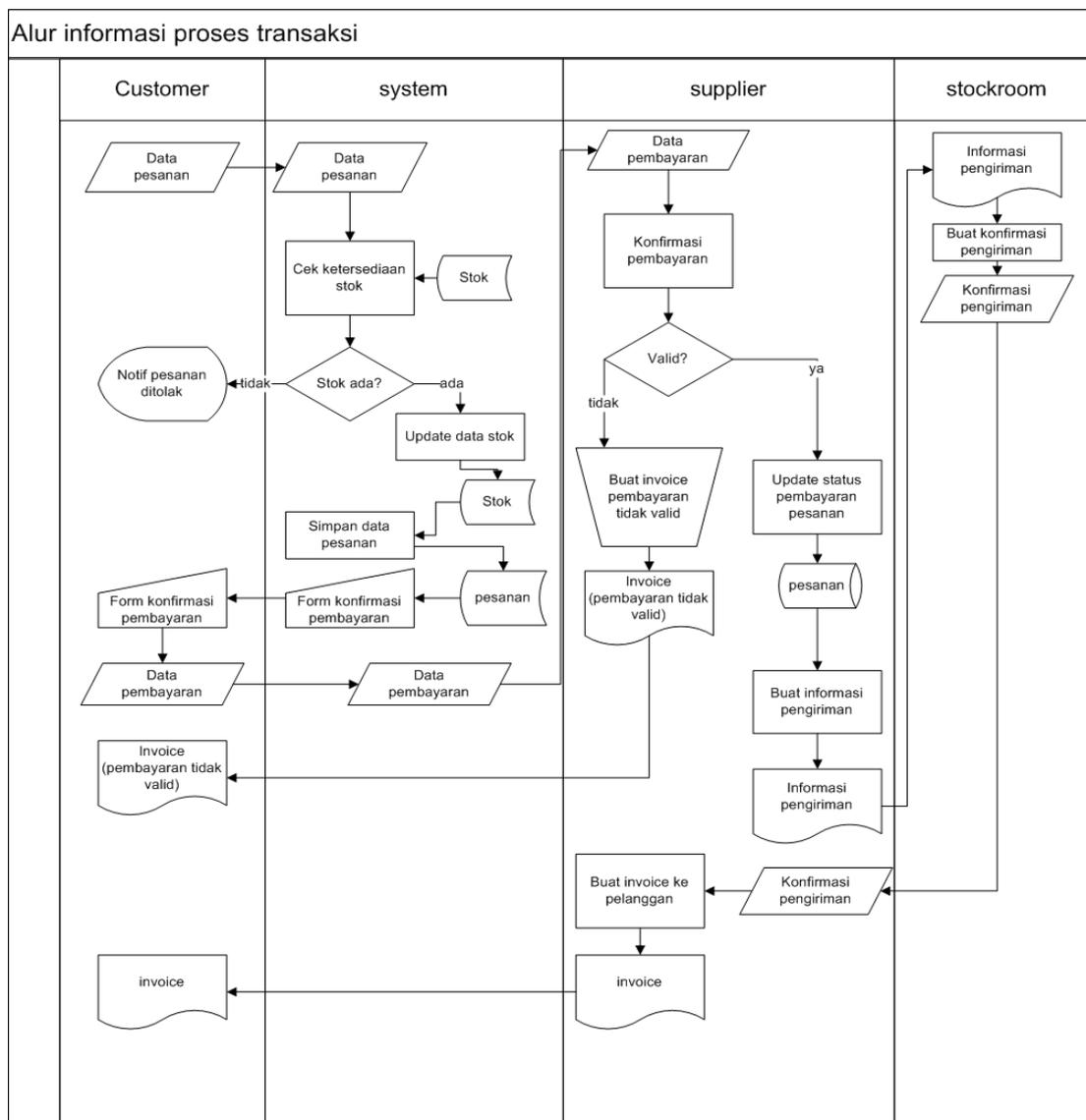
## **2.7 *Flowmap***

Menurut Firman (2010: 25), *flowmap* adalah gambaran hubungan antara entitas yang terlihat berupa alihan-alihan dokumen yang ada. Bagan alir dokumen merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusannya. Keterangan gambar dalam flowmap disajikan pada Gambar 2.2



**Gambar 2.2** Keterangan Gambar Dalam Flowmap

Contoh flowmap disajikan pada Gambar 2.3.



**Gambar 2.3** Contoh Flowmap

## 2.8 Data Flow Diagram (DFD)

Firman (2010: 26) menyampaikan bahwa DFD memperlihatkan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis & design*). DFD dapat menggambarkan arus data di dalam sistem dengan terstruktur. DFD memberikan informasi tambahan yang digunakan selama analisis domain informasi dan berfungsi sebagai dasar bagi pemodelan fungsi.

Al-Bahra bin Ladjamuddin (2005: 67-75), elemen dasar dari DFD adalah sebagai berikut:

a. Kesatuan luar (*External entity*)

Kesatuan luar adalah sesuatu yang berada di luar sistem tetapi memberikan data ke dalam sistem atau sebaliknya. Kesatuan luar tidak termasuk bagian dari sistem. Pedoman pemberian nama kesatuan luar adalah sebagai berikut:

- 1) Nama kesatuan luar berupa kata benda.
- 2) Kesatuan luar tidak boleh memiliki nama yang sama kecuali memang objeknya sama.

b. Proses (*Process*)

Proses merupakan kegiatan atau kerja yang dilakukan oleh sistem. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Pedoman pemberian nama proses adalah sebagai berikut:

- 1) Nama proses terdiri dari kata kerja dan kata benda yang mencerminkan fungsi proses tersebut.
- 2) Kata proses tidak boleh digunakan sebagai bagian dari nama suatu proses.
- 3) Tidak boleh ada beberapa proses yang memiliki nama yang sama.
- 4) Proses harus diberi nomor.

c. Simpanan data (*Data store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Pedoman pemberian nama simpanan data adalah sebagai berikut:

- 1) Nama harus mencerminkan simpanan data tersebut.
- 2) Bila namanya lebih dari satu kata, maka harus diberi tanda sambung.

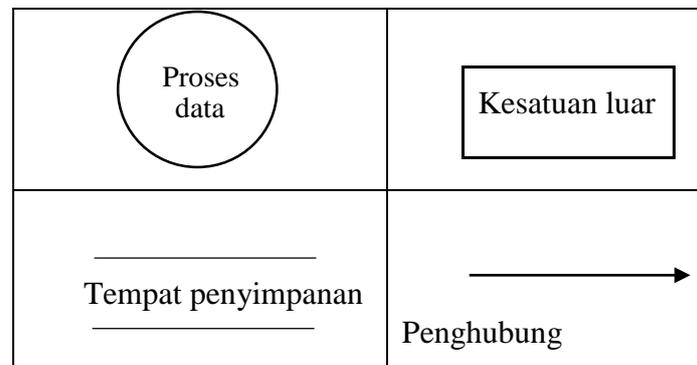
d. Arus data (*Data flow*)

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. Pedoman pemberian nama aliran data adalah sebagai berikut:

- 1) Nama aliran data yang terdiri dari beberapa kata dihubungkan dengan garis sambung.
- 2) Sedapat mungkin nama aliran data ditulis lengkap.
- 3) Tidak boleh ada aliran data dari kesatuan luar ke simpanan data atau sebaliknya, hubungan kesatuan luar dengan simpanan data harus melalui proses.

e. Simbol DFD

Simbol DFD disajikan pada Gambar 2.4.



**Gambar 2.4. Keterangan Gambar Dalam DFD**

Menurut Agus Winarto (2007: 2) DFD dibagi menjadi 2 jenis, yaitu:

a. Diagram Konteks

Diagram konteks merupakan DFD Level 0, yaitu diagram yang paling sederhana dari sebuah sistem informasi yang menggambarkan aliran data dari kesatuan luar ke dalam sistem dan sebaliknya. Contoh

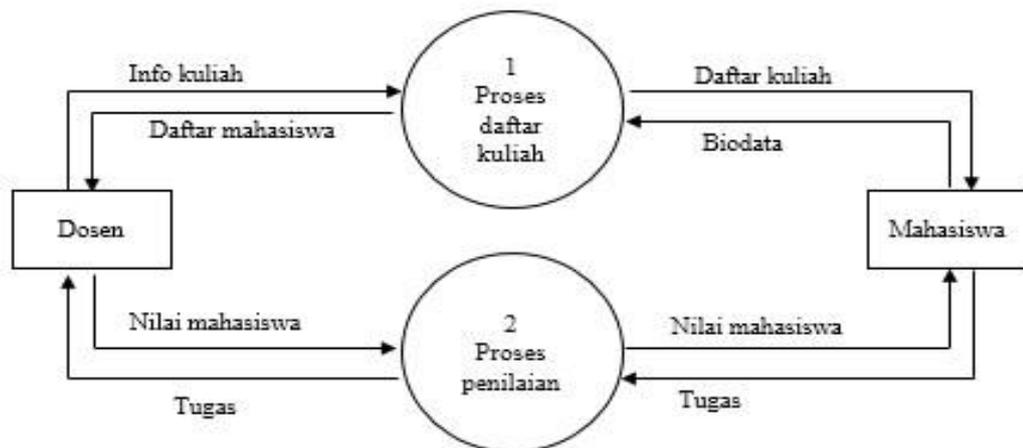
Diagram Konteks disajikan pada Gambar 2.5.



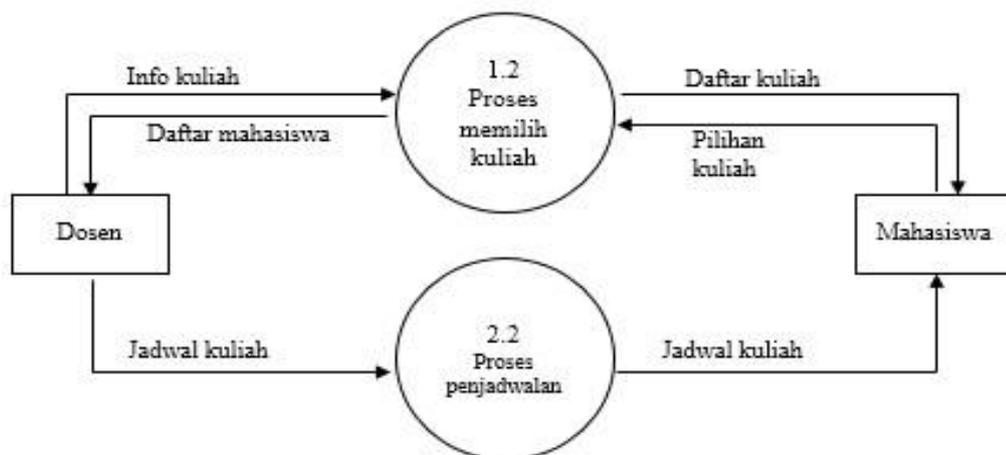
**Gambar 2.5. Contoh Diagram Konteks**

b. DFDLeveln

Diagram Level n merupakan diagram yang digunakan untuk menggambarkan diagram hasil pengembangan dari Diagram Konteks ke dalam komponen yang lebih detail. Contoh DFD Level 1 disajikan pada Gambar 2.6, dan contoh DFD Level 2 disajikan pada Gambar 2.7.



Gambar 2.6. Contoh DFD Level 1



Gambar 2.7. Contoh DFD Level 2

## 2.9 Hypertext Preprocessor (PHP)

Menurut Badiyanto (2013: 32), PHP: *Hypertext Preprocessor* (PHP) adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML/PHP banyak digunakan untuk membuat situs web dinamis.

Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang dan menulis halaman web menjadi dinamis dengan cepat. PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995, seorang programmer C. Pada waktu itu, PHP masih bernama

*Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Menurut Deni Sutaji (2012: 2), aturan penulisan skrip PHP adalah sebagai berikut:

- a. Embedded script, yaitu dengan cara meletakkan tag PHP diantara tag-tag HTML.

Contohnya:

```
<html>
<body>
<?php echo "Belajar"; ?>
<body>
</html>
```

- b. Non embedded script, yaitu dengan cara semua skrip HTML diletakkan dalam skrip PHP. Contohnya:

```
<?php
echo"<html>";
echo"<body>";
echo"Belajar PHP";
echo"</body>";
echo "</html>";
```

Deni Sutaji (2012: 3) menjelaskan bahwa PHP memiliki variabel yang digunakan sebagai tempat penyimpanan data sementara. Data tersebut akan hilang setelah program selesai dieksekusi. Aturan penggunaan nama variabel adalah sebagai berikut:

- a. Diawali dengan karakter \$.
- b. Bersifat *case sensitive*, jadi \$Var berbeda dengan \$var atau \$VAR.
- c. Karakter pertama harus berupa huruf atau garis bawah ( \_ ).
- d. Karakter berikutnya boleh huruf, angka atau garis bawah ( \_ ).

Berikut adalah beberapa contoh penulisan dan pendeklarasian variabel:

```
$namadepan = "Ali Shodikin";
```

```
$harga_barang = 3500;
```

```
$nilai3 = 3.45;
```

## 2.10 MySQL

Menurut Aprillita (2013: 2), MySQL adalah salah satu perangkat lunak Database Management System (DBMS) yang sering digunakan saat ini, yang didistribusikan secara gratis di bawah lisensi GPL (GNU General Public License). Sehingga setiap orang mudah untuk mendapatkan dan bebas untuk menggunakan MySQL.

Menurut Deni Sutaji (2011: 44), SQL dibagi menjadi tiga bentuk query, yaitu :

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

DDL adalah sebuah Metode Query SQL yang berguna untuk mendefinisikan data pada sebuah database. Berikut *query* yang termasuk DDL:

- 1) CREATE digunakan untuk melakukan pembuatan tabel dan database.
- 2) DROP digunakan untuk melakukan penghapusan tabel maupun database.
- 3) ALTER digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah Field (add), mengganti nama Field (change) ataupun menamakannya kembali (rename), serta menghapus (drop).

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

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian database yang telah ada atau yang telah dibuat sebelumnya.

Berikut *query* yang termasuk DML:

- 1) SELECT digunakan untuk menampilkan data pada tabel.
- 2) INSERT digunakan untuk melakukan penginputan / pemasukan data pada tabel database.
- 3) UPDATE digunakan untuk melakukan perubahan atau peremajaan terhadap data yang ada pada tabel.
- 4) DELETE digunakan untuk melakukan penghapusan data pada tabel.

### c. DCL (*Data Control Language*)

DCL adalah sebuah metode Query yang dapat digunakan untuk mengendalikan eksekusi perintah. Biasanya berhubungan dengan pengaturan hak akses. Berikut *query* yang termasuk DCL:

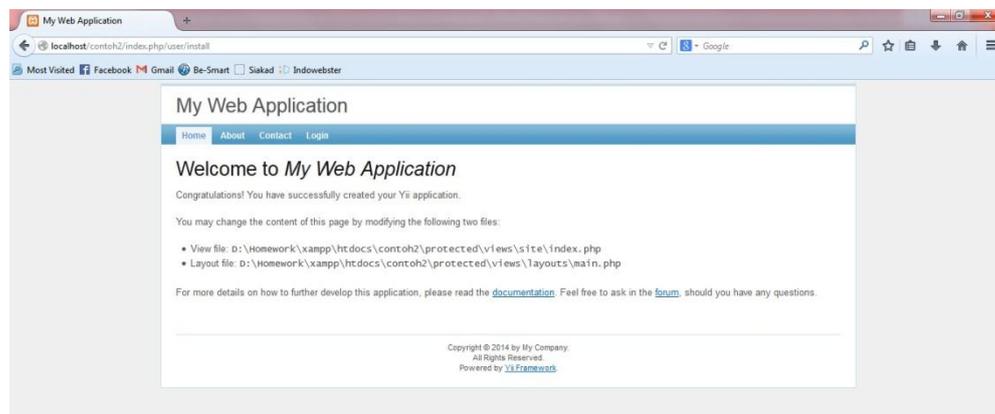
- 1) GRANT : Digunakan untuk memberikan hak akses (*privilage*) kepada user tertentu.

2) REVOKE : Digunakan untuk mencabut hak akses dari user tertentu

## 2.11 Framework Yii

Johan I, dkk (2011: 7) memaparkan bahwa *framework* adalah desain yang bisa digunakan kembali dan biasanya dinyatakan sebagai satu set abstraksi *class* yang mengatur bagaimana *class* saling terhubung. Perancangan pada *framework* dibuat sedemikian rupa sehingga sebagian atau seluruh software dapat digunakan kembali. *Framework* merupakan kerangka kode yang dapat disimpulkan dengan *classes* yang spesifik atau dengan fungsi yang telah dirancang untuk mengatasi masalah yang dihadapi.

*Framework* Yii adalah kerangka kerja PHP berbasis komponen yang dibuat oleh Qiang Xue tanggal 1 Januari 2008 untuk pengembangan aplikasi *web* berskala besar. *Framework* Yii menyediakan *reusability* maksimum dalam pemrograman *web* dan mampu meningkatkan kecepatan dalam membuat aplikasi *web*. Nama Yii diambil dari singkatan “Yes It Is!” (Badiyanto, 2013: 7). Aplikasi *web* yang dibangun menggunakan Yii menggunakan arsitektur *Model-View-Controller* (MVC). Contoh aplikasi Yii disajikan pada Gambar 2.8.



**Gambar 2.8. Aplikasi Yii**

*Model-View-Controller* (MVC) adalah model pembuatan program yang menerapkan arsitektur yang memisahkan proses, tampilan, dan bagian yang menghubungkan proses dengan tampilan. MVC bertujuan untuk memisahkan proses bisnis dari pertimbangan antarmuka *user* agar para pengembang bisa lebih mudah setiap bagian tanpa memengaruhi yang lain (Badiyanto, 2013: 49). Dalam MVC, *model* menggambarkan informasi (data) dan proses bisnis, *view* (tampilan) berisi elemen antarmuka *user* seperti teks, formulir masukan, sementara *controller* mengatur komunikasi antara *model* dengan *view*. Badiyanto, (2013: 49-55) menjelaskan tentang *Model-View-Controller* sebagai berikut:

a. *Model*

*Model* merupakan kelas yang mendasari logika proses dalam aplikasi perangkat lunak dan kelas yang terkait dengannya. *Model* adalah suatu objek yang tidak mengandung informasi tentang *user interface*. *Model* juga merupakan suatu kelas yang berisi metode/fungsi dan digunakan untuk menyimpan data dan aturan bisnis yang relevan. Pengguna dapat membuat *model* berdasarkan tabel menggunakan *Model Generator*. Contoh penggunaan *Model Generator* disajikan pada Gambar 2.9.

The screenshot shows the 'yii code generator' interface. On the left, there is a sidebar with a 'Generators' menu containing links for 'Controller Generator', 'Crud Generator', 'Form Generator', 'Model Generator', and 'Module Generator'. The main area is titled 'Model Generator' and contains the following fields and options:

- Database Connection \***: db
- Table Prefix**: [empty]
- Table Name \***: authassignment
- Model Class \***: Authassignment
- Base Class \***: CActiveRecord
- Model Path \***: application.models
- Build Relations**:
- Use Column Comments as Attribute Labels**:
- Code Template \***: default (D:\Homework\xampp\htdocs\yii\framework\yii\generators\model\templates\default)

Below the form are 'Preview' and 'Generate' buttons. At the bottom, there is a table with the following content:

Code File	Generate
models/Authassignment.php	new <input checked="" type="checkbox"/>

**Gambar 2.9. Model Generator Yii**

b. *View*

*View* merupakan kumpulan dari kelas yang mewakili unsur-unsur dalam antarmuka *user*. Dalam *view*, terdapat nama yang dipakai untuk mengidentifikasi *file* skrip tampilan saat dipanggil lewat fungsi *render*. Nama *view* sama seperti nama *file* skrip *view*-nya. Sebagai contoh, *file view edit* merujuk pada *file* skrip bernama `edit.php`. Pengguna dapat membuat *file view* yang terdiri dari *create*, *read*, *update*, dan *delete* berdasarkan *model* menggunakan *Crud Generator*. Contoh penggunaan *Crud Generator* disajikan pada Gambar 2.10.

**yii code generator** [help](#) | [webapp](#) | [yii](#) | [logout](#)

**Generators**

- [Controller Generator](#)
- [Crud Generator](#)**
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

### Crud Generator

This generator generates a controller and views that implement CRUD operations for the specified data model.

Fields with \* are required. Click on the highlighted fields to edit them.

Model Class \*  
Mahasiswa

Controller ID \*  
mahasiswa

Base Controller Class \*  
Controller

Code Template \*  
default (D:\Homework\kampus\htdocs\yii\framework\yii\generators\crud\templates\default)

Code File	Generate <input type="checkbox"/>
<a href="#">controllers\MahasiswaController.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswa_form.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswa_search.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswa_view.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswaadmin.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswacreate.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswaindex.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswaupdate.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswaview.php (diff)</a>	overwrite <input type="checkbox"/>

**Gambar 2.10. Crud Generator Yii**

c. *Controller*

*Controller* merupakan kelas yang menghubungkan *model* dan *view*, dan digunakan untuk berkomunikasi antara kelas dalam *model* dan *view*. *Controller* mempunyai *action* standar. Ketika permintaan *user* tidak menetapkan *action* mana yang dijalankan, program akan menjalankan *action* standar. Pengguna dapat membuat *controller* berdasarkan *model* menggunakan *Controller Generator*. Contoh penggunaan *Controller*

*Generator* disajikan pada Gambar 2.11.

**yii code generator** [help](#) | [webapp](#) | [yii](#) | [logout](#)

**Generators**

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

### Controller Generator

This generator helps you to quickly generate a new controller class, one or several controller actions and their corresponding views.

Fields with \* are required. Click on the highlighted fields to edit them.

Controller ID \*  
Mahasiswa

Base Class \*  
Controller

Action IDs  
index

Code Template \*  
default (D:\Homework\xampp\htdocs\yii\framework\yii\generators\controller\templates\default)

Code File	Generate <input type="checkbox"/>
<a href="#">controllers\MahasiswaController.php (diff)</a>	overwrite <input type="checkbox"/>
<a href="#">views\mahasiswa\index.php (diff)</a>	overwrite <input type="checkbox"/>

**Gambar 2.11. Controller Generator Yii**

#### d. Alur kerja MVC

Alur kerja yang dilakukan dalam MVC saat membuka halaman *web* adalah sebagai berikut:

- 1) *User* melakukan permintaan dengan memanggil URL misalnya `http://localhost/bimbinganonline/index.php?r=site/login` (test adalah nama aplikasi yang dibuat), dan *web server* memberikan responsi permintaan dengan menjalankan skrip *bootstrap* `index.php`.
- 2) Skrip *bootstrap* membuat sebuah instan aplikasi dan menjalankannya.
- 3) Aplikasi mendapat rincian informasi permintaan *user* dari komponen aplikasi bernama *request*.
- 4) Aplikasi menentukan *controller* dan aksi yang terdapat pada fungsi *action* yang diminta dengan bantuan komponen aplikasi bernama `urlManager`. Dalam contoh ini, *controller* adalah `Site` yang merujuk pada kelas `SiteController`, dan aksi adalah `login.php` yang arti sebenarnya ditentukan oleh *controller*.
- 5) Aplikasi membuat *instance controller* yang diminta untuk menangani permintaan *user*. *Controller* menentukan aksi `login.php` merujuk pada sebuah metode bernama `actionLogin` dalam kelas *controller*. Kemudian membuat dan menjalankan *filter* terkait aksi ini. Aksi dijalankan jika diijinkan oleh *filter*.
- 6) Aksi membaca *model* `LoginForm` dengan parameter `$model` yang terdapat pada *controller* dalam `actionLogin` diambil dari pernyataan:

```
$model->attributes=$_POST['LoginForm'];
```

- 7) Aksi menyiapkan *view* (tampilan) bernama `login.php` dengan *model* `LoginForm`.
- 8) *View* membaca dan menampilkan atribut *model* `login.php`.
- 9) *View* menjalankan beberapa *widget* dan *view* menyiapkan hasil yang dipasangkan dalam *layout* (tata letak).