



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Mulyono (2010:1) menyatakan, “Komputer adalah seperangkat alat elektronik yang terdiri atas peralatan *input*, alat yang mengolah *input*, dan peralatan *output* yang memberikan informasi, serta bekerja secara otomatis.”

Hamacher (dalam Mulyono, 2010:1) menyatakan, “Komputer adalah mesin perhitungan elektronik yang cepat dan dapat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya dan menghasilkan output berupa informasi.”

Siallagan (2009:1) menyatakan, “Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*process*), memberikan informasi (*output*) dan terkoordinasi di bawah kontrol program yang tersimpan dalam memorinya.”

Dari ketiga pengertian tersebut, penulis menyimpulkan bahwa komputer adalah alat elektronik yang saling bekerja sama untuk menerima data, memprosesnya dan memberikan informasi.

2.1.2. Pengertian Perangkat Lunak (*Software*)

Sukanto dan Shalahuddin (2013:2) menyatakan, “Perangkat lunak (*software*) adalah program komputer yang terasosiasi dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Mulyono (2010:97) menyatakan, “Perangkat lunak (*software*) adalah program yang berisi instruksi atau perintah sebagai perantara, yang menghubungkan antara *hardware* dan *brainware* sehingga menghasilkan informasi sesuai keinginan *brainware*.”



Siallagan (2009:3) menyatakan, “Perangkat lunak (*software*) adalah program-program komputer yang berguna untuk menjalankan atau mengoperasikan suatu pekerjaan sesuai dengan yang dikehendaki.”

Dari ketiga pengertian tersebut, penulis menyimpulkan bahwa perangkat lunak (*software*) adalah program komputer yang berisi perintah mengoperasikan suatu pekerjaan sehingga menghasilkan informasi sesuai keinginan pengguna.

2.1.3. Pengertian Studi Kelayakan

Sutabri (2012:72) menyatakan, “Studi Kelayakan adalah proses mempelajari dan menganalisis masalah yang telah ditentukan sesuai dengan tujuan akhir yang akan dicapai.”

Sutabri (2012:72) juga menjelaskan beberapa kriteria utama yang digunakan untuk suatu studi kelayakan adalah sebagai berikut:

1. Kelayakan Teknis

Sebuah masalah mempunyai kelayakan teknis jika tim perancang sistem dapat menyelesaikan masalah tersebut dengan menggunakan *hardware* dan *software* yang tersedia, yaitu yang ada atau yang dapat dimiliki.

2. Kelayakan Operasi

Sebuah masalah mempunyai kelayakan operasi jika tim perancang sistem dapat menyelesaikan masalah dengan menggunakan sumber daya manusia dan prosedur yang tersedia, yaitu yang ada atau yang dapat dimiliki.

3. Kelayakan Ekonomis

Sebuah masalah mempunyai kelayakan ekonomis jika tim perancang sistem dapat menyelesaikan masalah tersebut dalam waktu dan anggaran biaya yang masuk akal.

2.1.4. Metode Pengembangan Sistem

Sukamto dan Shalahuddin (2013:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic*)



life cycle). Model air terjun (*waterfall*) menyediakan pendekatan alur hidup terurut sebagai berikut :

1. Analisis

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 tahap 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.

3. Pembuatan Kode Program (Pengodean)

Desain harus diterjemahkan kedalam program perangkat lunak. Hasil dari tahap ini adalah program sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak 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 atau pemeliharaan sistem

Pada tahap ini 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 pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.



2.1.5. Metode Pengujian Sistem

Yakub (2012:150) menjelaskan tentang metode pengujian sistem yaitu, *black box*. *Black box* merupakan pendekatan komplementer dari teknik *white box testing*, karena pengujian *black box* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi *input* yang sesuai dengan persyaratan fungsional suatu program.

2.2. Teori Judul

2.2.1. Pengertian Aplikasi

Asropudin (2013:6) menyatakan, “Aplikasi atau *application* adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu.”

Sutabri (2012:147) menyatakan, “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.”

Dari kedua pengertian tersebut, penulis menyimpulkan bahwa aplikasi adalah perangkat lunak yang dibuat dan difungsikan secara khusus untuk mengerjakan tugas yang dimilikinya.

2.2.2. Pengertian Data

Hartono (2013:15) menyatakan, “Data adalah hasil pengukuran dan pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan, atau kejadian.”

Asropudin (2013:22) menyatakan, “Data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.”

Sutarman (2009:14) menyatakan, “Data adalah fakta dari sesuatu pernyataan yang berasal dari kenyataan dimana pernyataan tersebut merupakan hasil pengukuran atau pengamatan.”

Dari ketiga pengertian tersebut, penulis menyimpulkan bahwa data adalah suatu fakta dari hasil pengukuran dan pencatatan yang dapat diolah sehingga menghasilkan informasi.



2.2.3. Pengertian Pengolahan Data

Kristanto (2008:8) menyatakan, “Pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan data menjadi informasi yang memiliki kegunaan.”

Sutarman (2012:4) menyatakan, “Pengolahan data adalah perhitungan atau transformasi data menjadi informasi yang mudah dimengerti dan dipahami serta sesuai dengan yang diinginkan.”

Dari kedua pengertian tersebut, penulis menyimpulkan bahwa pengolahan data adalah perhitungan atau transformasi yang digunakan untuk menggambarkan perubahan data menjadi informasi yang mudah dimengerti dan dipahami.

2.2.4. Pengertian Administrasi

Salim (2013:16) menyatakan, “Administrasi adalah usaha dan kegiatan yg meliputi penetapan tujuan serta penetapan cara-cara penyelenggaraan pembinaan organisasi.”

2.2.5. Pengertian Bimbingan

Salim (2013:100) menyatakan, “Bimbingan adalah petunjuk atau penjelasan cara mengerjakan sesuatu.”

2.2.6. Pengertian Jamaah

Salim (2013:285) menyatakan, “Jamaah adalah kelompok, kumpulan, jemaat”

2.2.7. Pengertian Haji

Mulyono dan Rofi'ie (2013:8) menyatakan, “Haji secara bahasa, dapat diartikan mengunjungi, menuju dan ziarah. Sedangkan secara istilah *syara'*, haji adalah berkunjung ke Baitullah (Ka'bah) dan tempat lainnya (mas'a, Arafah, Muzdalifah dan Mina) dalam waktu tertentu untuk mengerjakan amalan-amalan, seperti thawaf, sa'i, wukuf di Arafah dan beberapa amalan lainnya.”



2.2.8. Pengertian Umrah

Mulyono dan Rofi'ie (2013:8) menyatakan, “Umrah secara bahasa adalah ziarah atau mendatangi suatu tempat. Umrah secara istilah adalah mendatangi Baitullah al-Haram untuk melaksanakan thawaf, sa'i dan mencukur atau menggunting rambut.”

2.2.9. Pengertian Judul Secara Keseluruhan

Aplikasi Pengolahan Data Administrasi Jamaah Haji pada Bimbingan Haji dan Umrah PT Arraudhah Wisata Imani Palembang adalah suatu aplikasi yang dibuat menggunakan aplikasi pemrograman *PHP* dan *database MySQL* sehingga mengoptimalkan dalam penyimpanan data administrasi jamaah haji karena diproses secara komputerisasi dan menggunakan aplikasi pemrograman.

2.3. Teori Program

2.3.1. Basis Data (*Database*)

Priyadi (2014:2) menyatakan, “Secara umum, pengertian basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital.”

Sukamto dan Shalahuddin (2013:43) menyatakan, “Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat yang bertujuan memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”

Heryanto (2012:1) menyatakan, “Basis data adalah kumpulan dari tabel. Satu table mempresentasikan suatu entitas tertentu. Suatu entitas terdiri dari beberapa atribut.”

Dari ketiga pengertian tersebut, penulis menyimpulkan bahwa basis data adalah kumpulan fakta berupa table-tabel untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Priyadi (2014:3) juga menjelaskan beberapa elemen basis data, diantaranya adalah sebagai berikut:



1. Tabel

Pada suatu basis data, tabel direpresentasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.

2. *Field*

Pada tabel basis data, kolom merupakan representasi untuk sebuah nama *field* yang pembacaan datanya dilakukan secara vertikal.

3. *Record*

Pada tabel basis data, baris merupakan suatu representasi untuk sebuah *record* yang pembacaan datanya dilakukan secara horizontal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabel tersebut.

4. Kardinalitas

Kardinalitas merupakan batasan dari banyaknya hubungan, yang dapat dilakukan oleh himpunan entitas dalam melakukan relasi dengan himpunan entitas lainnya. Variasi kemungkinan untuk melakukan relasi yang dimiliki oleh kardinalitas terdiri dari empat macam, yaitu:

- a. Satu ke satu \longrightarrow (1:1).
- b. Satu ke banyak \longrightarrow (1:N).
- c. Banyak ke satu \longrightarrow (N:1).
- d. Banyak ke banyak \longrightarrow (N:N).

2.3.2. Sekilas Tentang PHP

2.3.2.1. Pengertian *Hypertext Preprocessor (PHP)*

Saputra (2013:1) menyatakan, “*PHP* adalah singkatan dari (*Hypertext Preprocessor*) yang merupakan suatu bahasa pemrograman yang hanya dapat berjalan pada sisi *server* (*Server Side Scripting*).”

MADCOMS (2011:216) menyatakan, “*PHP* adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *webserver* dan berfungsi sebagai pengolah data pada sebuah *server*.”



Badiyanto (2013:32) menyatakan “*PHP* adalah bahasa skrip yang dapat ditanamkan atau disisipkan dalam *HTML* yang banyak dipakai untuk membuat situs *web dinamis* dan membangun sebuah *CMS*.”

Dari ketiga pengertian di atas, dapat disimpulkan bahwa *PHP* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi yang hanya dapat berjalan pada sisi *server* dan banyak dipakai untuk membuat situs *web dinamis*.

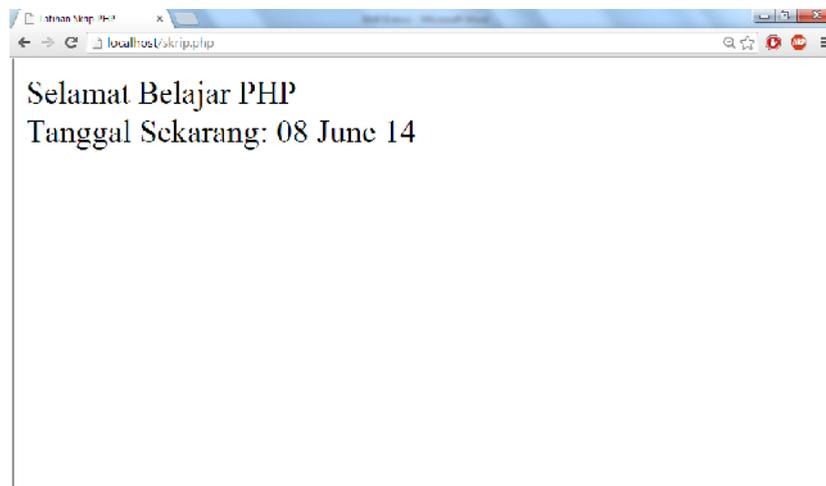
2.3.2.2. Skrip *PHP*

Kadir (2008:3) menjelaskan tentang skrip *PHP* berkedudukan sebagai tag dalam bahasa *HTML*. Sebagaimana diketahui, *HTML* adalah bahasa standar untuk membuat halaman-halaman *web*.

Sebagai contoh, berikut adalah contoh kode *PHP* yang berada di dalam kode *HTML*:

```
<HTML>
<HEAD>
<TITLE> Latihan Skrip PHP </TITLE>
</HEAD>
<BODY>
Selamat Belajar PHP
<BR>
<?php
printf
    ("Tanggal Sekarang: %s ", Date("d F y"));
?>
</BODY>
</HTML>
```

Bila dijalankan melalui *browser*, kode tersebut membentuk tampilan seperti berikut:

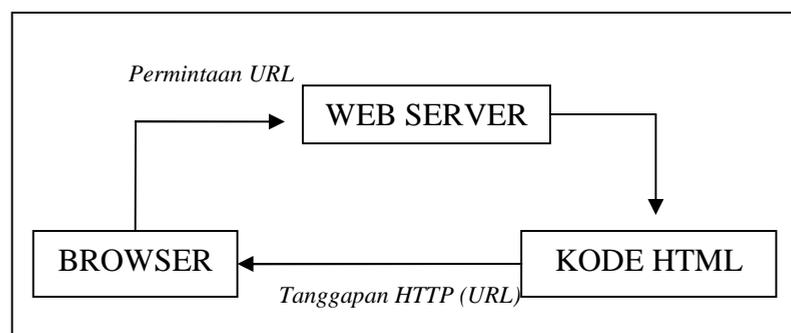


Gambar 2.1. Tampilan Kode *PHP* pada *Browser*

2.3.2.3. Konsep Kerja *PHP*

Saputra (2013:4) menjelaskan bahwa tentang konsep kerja *PHP* adalah sebagai berikut:

- a. Server membaca permintaan dari *client/browser*.
- b. Kemudian dilanjutkan untuk mencari halaman/*page* pada *server*.
- c. *Server* melakukan instruksi yang diberikan oleh php untuk melakukan modifikasi pada halaman/*page*.
- d. Selanjutnya hasil modifikasi tersebut akan dikembangkan kepada *client/browser*.



(Sumber : Saputra, 2013:5)

Gambar 2.2. Cara Kerja *PHP*



2.3.2.4. Tipe Data *PHP*

MADCOMS (2011:221) menjelaskan bahwa terdapat beberapa tipe data *PHP*, yaitu:

a. Tipe Data *Integer*

Tipe data ini berisikan data semua bilangan bulat yang besarnya range sama dengan data pada bahasa C, yaitu antara **-2, 147, 483, 648** sampa **+2, 147, 483, 647** pada platform **32bit**. Apabila data di luar kisaran tersebut maka *PHP* secara otomatis mengkonversikan data tersebut dari tipe ***Integer*** menjadi tipe ***Floating Point***.

b. Tipe Data *Floating Point*

Tipe data ini berisikan bilangan pecahan atau bilangan desimal. Kisaran data *floating* adalah antara **1.7E-308** sampai **1.7E+308**. Bentuk dari data ini adalah desimal ataupun dalam bentuk pangkat.

c. Variabel

Dalam program *PHP*, variabel digunakan untuk menyimpan data sementara baik jenis *string*, *integer*, maupun *array*. Variabel dinyatakan dengan tanda \$ di depan nama variabel. Nama variabel dapat berupa huruf, angka maupun garis bawah. Namun dalam penulisannya, variabel harus diawali dengan huruf atau garis bawah (_), kemudian diikuti huruf atau angka. Penulisan variabel yang diawali angka tidak dibenarkan.

2.3.3. Sekilas Tentang *MySQL*

2.3.3.1. Pengertian *MySQL*

Saputra (2012:77) menyatakan, “*MySQL* adalah salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman *PHP*.”

Kadir (2008:348) menyatakan, “*MySQL* adalah salah satu jenis *database server* yang sangat populer yang bersifat *open source* dan termasuk jenis *RDBMS (Relational Database Management System)*.”



Dari kedua pengertian tersebut, dapat disimpulkan bahwa *MySQL* adalah jenis *software* pada *Relational Database Management System* (RDBMS) pada sebuah manajemen *database* sebagai basis data.

2.3.3.2. Fungsi-Fungsi *MySQL*

Kadir (2008:360), menjelaskan tentang sejumlah fungsi yang berawalan *mysql* yang digunakan untuk mengakses *database server MySQL* sebagai berikut:

Tabel 2.1. Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan
1.	<i>MySQL_connect()</i>	Membuka koneksi ke <i>server MySQL</i>
2.	<i>MySQL_close()</i>	Menutup koneksi <i>MySQL</i>
3.	<i>MySQL_select_db()</i>	Membuka/mengaktifkan suatu basis data
4.	<i>MySQL_query()</i>	Mengeksekusi <i>string query</i>
5.	<i>MySQL_db_query()</i>	Melakukan <i>query</i>
6.	<i>MySQL_num_fields()</i>	Menghitung jumlah <i>field</i> dalam tabel tertentu
7.	<i>MySQL_fetch_assoc()</i>	Mengambil hasil <i>query</i> sebelumnya
8.	<i>MySQL_fetch_result()</i>	Membebaskan <i>resource</i> dari hasil <i>query</i>
9.	<i>MySQL_fetch_row()</i>	Menampilkan hasil <i>query</i> ke dalam <i>array</i>
10.	<i>MySQL_fetch_array()</i>	Menampilkan hasil <i>query</i> sebagai <i>array</i>
11.	<i>MySQL_fetch_field()</i>	Menampilkan informasi <i>field</i> termasuk tipe datanya
12.	<i>MySQL_affected_rows()</i>	Menampilkan jumlah baris yang dikenai operasi <i>SQL</i>
13..	<i>MySQL_errno</i>	Menampilkan kode kesalahan (<i>error number</i>) dari perintah <i>SQL</i>
14.	<i>MySQL_create_db()</i>	Membuat <i>database MySQL</i>
15.	<i>MySQL_drop_db()</i>	Menghapus <i>database MySQL</i>
16.	<i>MySQL_list_dbs()</i>	Menghasilkan seluruh daftar <i>database MySQL</i>

Lanjutan Tabel 2.1. Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan
17.	<i>MySQL_list_tables()</i>	Menampilkan seluruh tabel dalam <i>database</i> tertentu
18.	<i>MySQL_list_fields()</i>	Menampilkan seluruh <i>field</i> dalam tabel tertentu

(Sumber : Kadir, 2008:360)

2.3.4. Sekilas Tentang *Sublime Text 2*

2.3.4.1. Pengertian *Sublime Text 2*

Sibarani menyatakan, “*Sublime Text 2* adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform* operating system dengan menggunakan teknologi *Phyton API*.”

2.3.4.2. Fitur-Fitur *Sublime Text 2*

Sibarani menjelaskan tentang fitur-fitur yang ada dalam *Sublime Text 2* adalah sebagai berikut:

1. *Goto Anything*

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa *keystrokes*.

2. *Multiple Selections*

Fitur ini memungkinkan *user* untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

3. *Command Pallete*

Dengan hanya beberapa *keystrokes*, *user* dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu.

4. *Distraction Free Model*

Bila *user* memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu *user* dengan memberikan tampilan layar penuh.



5. *Split Editing*

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan *editing* perpecahan. Mengedit sisi file dengan sisi, atau mengedit dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang *user* inginkan.

6. *Instant Project Switch*

Menangkap semua file yang dimasukkan kedalam *project* pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam *project* lainnya dengan cepat.

7. *Plugin API*

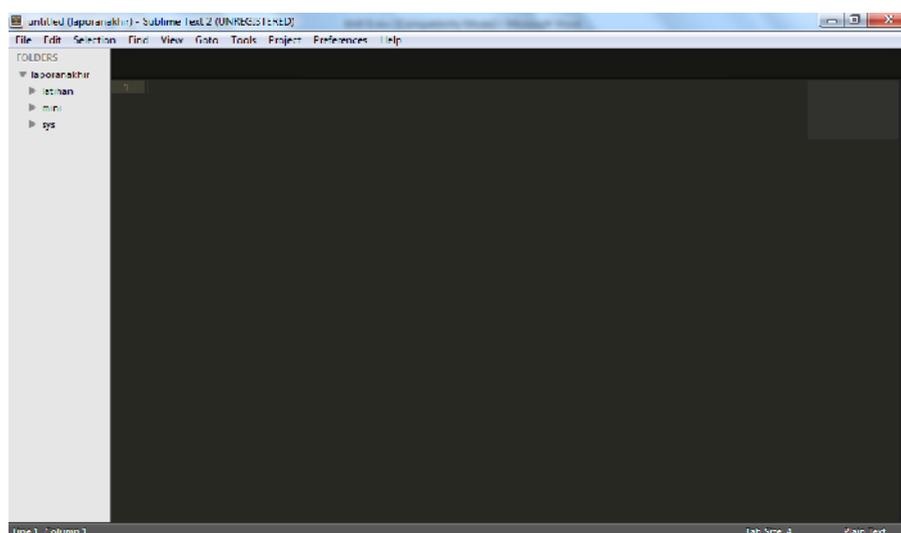
Dilengkapi dengan plugin API berbasis Phyton sehingga membuat aplikasi ini sangat tangguh.

8. *Customize Anything*

Aplikasi ini memberikan *user* fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini.

9. *Cross Platform*

Aplikasi ini dapat berjalan hampir disemua *operating system* modern seperti Windows, OS X, dan Linux *based operating system*.



Gambar 2.3. Ruang Kerja *Sublime Text 2*



2.3.5. Pengertian *JavaScript*

Saputra (2012:3) menyatakan, “*JavaScript* adalah bahasa *scripting* yang handal yang berjalan di sisi *client*.”

Badiyanto (2013:27) menyatakan, “*JavaScript* adalah bahasa pemrograman *java*, perintah-perintahnya ditulis dengan kode yang disebut skrip. *Java* adalah bahasa pemrograman berorientasi objek, sedangkan *script* adalah serangkaian instruksi program.”

Dari kedua pengertian tersebut, penulis menyimpulkan bahwa *JavaScript* adalah bahasa *scripting* yang perintahnya ditulis dengan kode-kode.

2.4. Teori Khusus

2.4.1. Pengertian *Data Flow Diagram* (DFD)

Saputra (2013:118) menyatakan, “*Data flow diagram* yang disingkat *DFD* merupakan suatu diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas.”

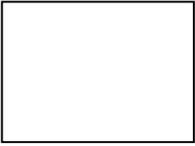
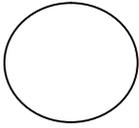
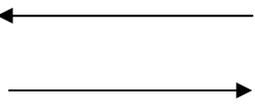
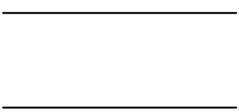
Sukamto dan Shalahuddin (2013:70) menyatakan, “*Data flow diagram* adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*).”

Sutabri (2012:117) menyatakan, “*Data flow diagram* adalah salah satu *network* yang menggambarkan sistem automat/komputerisasi, manualisasi atau gabungan dari keduanya, yang penggambarannya disusun dalam bentuk kumpulan komponen sistem yang saling berhubungan sesuai dengan aturan mainnya.”

Dari ketiga pengertian tersebut, penulis menyimpulkan bahwa *data flow diagram* merupakan suatu cara atau model logika data untuk membuat rancangan sebuah sistem yang menggambarkan dari mana asal data dan kemana tujuan data dan interaksi antar data.

Saputra (2013:119) menjelaskan 4 simbol DFD dalam versi Yourdon/De Marco sebagai berikut:

Tabel 2.2. Simbol-simbol *Data Flow Diagram*

No.	Simbol	Nama Simbol	Keterangan
1.		Entitas Eksternal	Entitas eksternal dapat berupa orang/unit terkait yang berinteraksi dengan sistem, tetapi di luar sistem.
2.		Proses	Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
3.		Aliran Data	Aliran data dengan arah khusus dari sumber ke tujuan.
4.		Data Store	Penyimpanan data atau tempat data di-refer oleh proses.

(Sumber: Saputra, 2013:119)

Sukamto dan Shalahuddin (2013:72) menjelaskan tahapan-tahapan perancangan dengan menggunakan *DFD* sebagai berikut:

1. *DFD* Level 0 atau sering disebut juga *Context Diagram*.

DFD Level 0 menggambarkan sistem yang dibuat sebagai entitas tunggal yang berinteraksi dengan orang maupun sistem lain. *DFD* Level 0 menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. *DFD* Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. *DFD* Level 1 merupakan hasil *breakdown* *DFD* level 0 yang sebelumnya sudah dibuat.



3. DFD Level 2

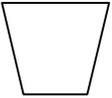
Modul-modul pada *DFD* Level 1 dapat di-*breakdown* menjadi *DFD* Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah *DFD* Level 2 sama dengan jumlah modul pada *DFD* Level di atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan *DFD* Level 1 atau Level 2.

2.4.2. Pengertian *Blockchart*

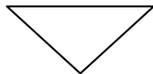
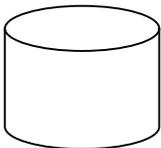
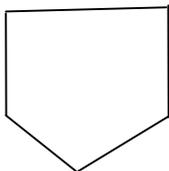
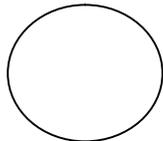
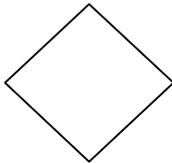
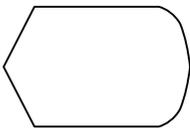
Kristanto (2008:75) menyatakan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu.”

Kristanto (2008:75) juga menjelaskan tentang simbol-simbol yang sering digunakan dalam *block chart* dapat dilihat pada tabel berikut ini:

Tabel 2.3. Simbol-simbol *Blockchart*

No	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan.
2.		Multi dokumen.
3.		Proses manual.
4.		Proses yang dilakukan oleh komputer.

Lanjutan Tabel 2.3. Simbol-simbol *Blockchart*

No	Simbol	Keterangan
5.		Menandakan dokumen yang diarsipkan (arsip manual).
6.		Data penyimpanan (<i>data storage</i>).
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukan data secara manual.

(Sumber : Kristanto, 2008:75)

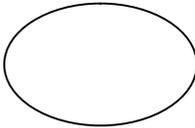
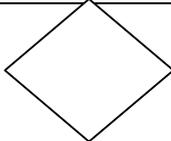


2.4.3. Pengertian *Entity Relational Diagram (ERD)*

Sukamto dan Shalahuddin (2013:50) menyatakan, “*Entity Relationship Diagram (ERD)* digunakan untuk pemodelan basis data relasional. *ERD* memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker), Ian (dikembangkan oleh Palmer, Harry Ellis), notasi Crow’s Foot, dan beberapa notasi lainnya.”

Berikut adalah simbol-simbol yang digunakan pada *ERD* dengan notasi Chen:

Tabel 2.4. Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Keterangan
1.		Entitas merupakan data inti yang akan disimpan.
2.		Atribut atau <i>field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.		Relasi, menghubungkan anatar entitas dan biasanya diawali dengan kata kerja.
4.		Asosiasi, penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> .

(Sumber : Sukamto dan Shalahuddin, 2013:50)



2.4.4. Pengertian *Flow Chart*

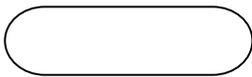
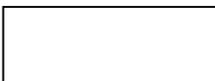
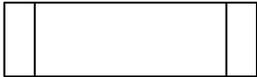
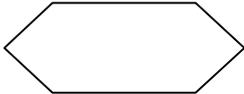
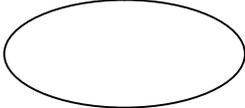
Saputra (2013:120) menyatakan, “*Flow Chart* merupakan suatu diagram yang menggambarkan alur kerja suatu sistem.”

Siallagan (2009:6) menyatakan, “*Flow Chart* adalah diagram alir yang mempergunakan simbol atau tanda untuk menyelesaikan masalah.”

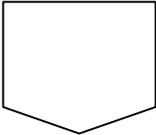
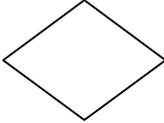
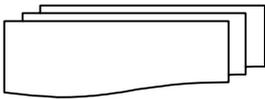
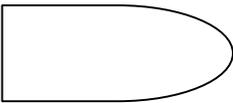
Dari kedua pengertian di atas, penulis menyimpulkan bahwa *flow chart* merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya.

Siallagan (2009:6) juga menjelaskan tentang simbol-simbol yang sering digunakan dalam *flow chart* dapat dilihat pada tabel berikut ini:

Tabel 2.5. Simbol-simbol *Flowchart*

No.	Simbol	Fungsi
1.		Terminal menyatakan awal atau akhir dari suatu algoritma.
2.		Menyatakan proses.
3.		Proses yang terdefinisi atau sub program.
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5.		Menyatakan masukan dan keluaran (<i>input/output</i>).
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.

Lanjutan Tabel 2.5. Simbol-simbol *Flowchart*

No.	Simbol	Fungsi
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>decision</i> (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).
15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

(Sumber : Siallagan, 2009:6)



2.4.5. Pengertian Kamus Data

Sukamto dan Shalahuddin (2013:73) menyatakan, “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).”

Sukamto dan Shalahuddin (2013:74) menjelaskan bahwa kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, yaitu sebagai berikut:

Tabel 2.6. Simbol-simbol dalam Kamus Data

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	AND atau dan
3	()	data opsional
4	[]	baik...atau...
5	*...*	batas komentar
6	{ }n	n kali diulang / bernilai banyak

(Sumber : Sukamto dan Shalahuddin, 2013:74)