



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Sugiarto (2014:13), “Komputer berasal dari Bahasa Inggris, yaitu dari kata *to compute* yang berarti alat hitung”.

Asropudin (2013:19), “Komputer adalah alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing-masing komputer.”

Dapat disimpulkan bahwa komputer alat yang dipakai untuk mengolah data menurut prosedur yang telah ditentukan.

2.1.2. Pengertian Internet

Asropudin (2013:48), “Internet adalah istilah yang dipakai untuk menunjukkan network tingkat dunia yang terdiri dari komputer dan layanan servis yang pemakai nya sekitar 50 juta pemakai dan puluhan sistem informasi termasuk e-mail, Gopher, FTP dan Word Wide Web”.

Yugianto dan Rachman (2012:36), “Internet adalah suatu sistem jaringan komunikasi beberapa komputer yang terhubung tanpa batas waktu maupun tempat, sehingga dapat dikatakan sebagai komunitas jaringan global.”

2.1.3. Pengertian Basis Data (*Database*)

Rosa dan Shalahuddin (2014:02), “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau membuat informasi data tersedia saat dibutuhkan.”

Raharjo (2011:3), “*database* didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat.”



2.1.4. Pengertian Aplikasi Web

Kadir (2009:2), “Aplikasi web adalah jenis aplikasi yang diakses melalui browser, biasanya Internet Explorer dan Mozilla Firefox .”

Sujatmiko (2012:23), “Aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti internet atau intranet.”

Kesimpulannya, Aplikasi Web adalah sekumpulan halaman web yang mampu berinteraksi dengan pengunjung, dengan sesama halaman web dan dengan berbagai sumber data yang ada di server web.

2.1.5. Pengertian Perangkat Lunak

Rosa dan Shalahuddin (2014:02), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).”

Sujatmiko (2012:256), “Perangkat Lunak adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya.”

Kesimpulannya, Perangkat Lunak adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan.

2.1.6. Metode Pengembangan Sistem

Rosa dan Shalahuddin (2014:28), menjelaskan metode pengembangan sistem yang sering digunakan dalam tahapan pengembangan sistem yaitu model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*).

Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Berikut ini tahapan-tahapan model air terjun :

a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dipahami perangkat lunak



seperti apa yang dibutuhkan oleh pengguna. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

b. Desain

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

c. Pembuatan kode program

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

d. Pengujian

Pengujian focus pada perangkat lunak secara segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji.

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

Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.2. Teori Khusus

2.2.1. Pengertian DFD (*Data Flow Diagram*)

Indrajani (2015:27) “*Data Flow Diagram* adalah sebuah alat yang menggambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut.”

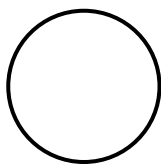
DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan



menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

Rosa dan Shalahuddin, (2014:71), menjelaskan notasi pada DFD adalah sebagai berikut :

Tabel 2.1. Simbol-simbol Data Flow Diagram

No	Simbol	Keterangan
1.		Proses atau fungsi atau prosedur; pada permodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka permodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja.
2.		<i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada permodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka permodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical sData Model</i> (PDM)) catatan: nama yang diberikan pada sebuah penyimpanan biasanya kata benda



3.		Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. catatan: nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda.
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>). catatan: nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”.

(Sumber: Rosa dan Shalahudin (2014:71))

Kristanto (2008:68) menjelaskan beberapa petunjuk yang dapat dipakai dalam proses pembuatan simbol DFD. Petunjuk-petunjuk itu adalah sebagai berikut:

1. Penamaan yang jelas
 - a. Sebaiknya menggunakan nama yang mengacu pada fungsi, yaitu gabungan antara kata kerja yang spesifik.
 - b. Jangan menggunakan nama terlalu umum.
 - c. Gunakan nama yang mudah dimengerti oleh pemakai.
2. Memberi nomor pada proses
 - a. Nomor yang diberikan pada proses tidak harus menjadi nomor urut.
 - b. Penomoran dimaksudkan sebagai identifikasi proses dan memudahkan penurunan (level yang lebih rendah) ke proses berikutnya.



3. Penggambaran kembali
 - a. Ukuran dan bentuk lingkaran tetap sama.
 - b. Panah yang melengkung dan lurus tidak menjadi masalah.
4. Hindarilah proses yang mempunyai masukan tetapi tidak mempunyai keluaran begitu pula sebaliknya, hindarilah proses yang mempunyai keluaran tetapi tidak mempunyai masukan.
5. Hati-hati dengan aliran data dan proses yang tidak dinamai. Aliran proses yang tidak diberi nama dapat mengakibatkan elemen data yang saling tidak berhubungan akan menjadi satu.

Rosa dan Shalahuddin (2014:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan Data Flow Diagram yaitu:

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

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

2. Membuat 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. Membuat 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 kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.

4. Membuat DFD Level 3 dan seterusnya

DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

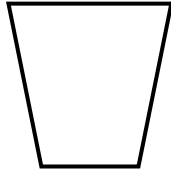
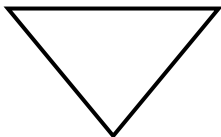


2.2.2. Pengertian *Blockchart*

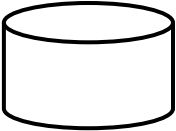
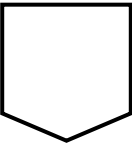
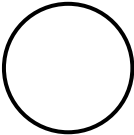
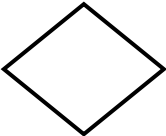
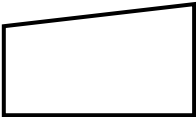
Kristanto (2008:75) menjelaskan, “*Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *blockchart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi.”

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

Tabel 2.2. Simbol-simbol dalam *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 dilakukan oleh komputer.
5.		Menandakan dokumen yang diarsipkan (arsip manual)



6.		Data penyimpanan (<i>Storage</i>)
7.		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik.
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain.
9.		Terminal yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama.
10.		Terminal yang menandakan awal dan akhir dari suatu aliran.
11.		Pengambilan keputusan (<i>Decision</i>).
12.		Layar peraga (<i>monitor</i>).
13.		Pemasukkan data secara manual.

(Sumber: Kristanto (2008:75))

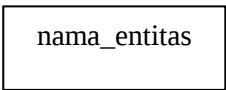
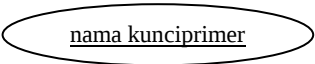
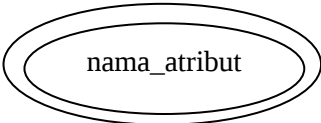
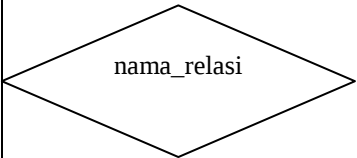
2.2.3. Pengertian ERD (*Entity Rational Diagram*)

Rosa dan Shalahuddin, (2013:50), “ERD adalah permodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk permodelan basis data relasional.”

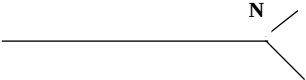


Rosa dan Shalahuddin, (2013:50), menjelaskan simbol-simbol yang digunakan dalam ERD, yaitu:

Tabel 2.3. Simbol-simbol *EntityRelationship Diagram* (ERD)

No.	Simbol	Keterangan
1.	Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	Atribut multivalai / <i>multivalve</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.



6.	Asosiasi / <i>association</i> 	Penghubung antara relasi dan entitas di mana 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 onetomany menghubungkan entitas A dan entitas B.
----	--	--

(Sumber: Rosa dan Shalahudin (2014:50))

2.2.4. Pengertian *Flowchart*

Lamhot (2015:14) menjelaskan, “*Flowchart* adalah langkah-langkah menyelesaikan masalah yang dituliskan dalam simbol-simbol tertentu.”

Menurut Lamhot (2015:14-16), Gambaran simbol-simbol *Flowchart* sebagai berikut:

Tabel 2.4. Simbol-simbol data *Flowchart*

No.	Simbol	Nama	Fungsi
1		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
2		<i>Input/output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya.
3		<i>Proccess</i>	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.



4		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya atau tidak.
5		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7		<i>Predefined process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8		<i>Fancher Card</i>	Menyatakan input berasal dari kartu atau output data ke kartu.
9		<i>Punch Tape</i>	
10		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui Printer)
11		<i>Flow</i>	Menyatakan jalannya arus suatu proses

(Sumber: Lamhot (2015:14))

2.2.5. Pengertian Kamus Data

Rosa dan Shalahuddin, (2014:73), “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)."

Rosa dan Shalahuddin, (2014:74), menjelaskan simbol-simbol yang digunakan dalam kamus data, yaitu :

Tabel 2.5. Simbol-simbol dalam Kamus Data

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

(Sumber: Rosa dan Shalahuddin (2014:74))

2.3. Pengertian Judul

2.3.1. Pengertian Aplikasi

Sujatmiko (2012:23), "Aplikasi adalah program komputer yang dibuat oleh sebuah perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu."

2.3.2. Pengertian Pengaduan

Kamus Besar Bahasa Indonesia menjelaskan "Pengaduan adalah ungkapan rasa tidak puas akan hal-hal yang tidak begitu penting tetapi perlu di perhatikan".

2.3.3. Pengertian Keluhan

Kamus Besar Bahasa Indonesia menjelaskan "Keluhan adalah ungkapan yang keluar karena perasaan ketidakpuasan terhadap suatu layanan tertentu".



2.3.4. Pengertian Pelanggan

Nasution (2004:104), “Pelanggan adalah orang yang membeli atau menggunakan produk atau jasa suatu perusahaan”.

2.3.5. Pengertian Kantor Pos

Kantor Pos adalah fasilitas fisik tidak bergerak untuk melayani penerimaan, pengumpulan, penyortiran, dan pengantaran surat dan paket pos.

2.3.6. Pengertian Aplikasi Pengaduan Keluhan Pelanggan Pada PT Pos Indonesia (Persero) Cabang Palembang Sumatera Selatan

Aplikasi Pengaduan Keluhan Pelanggan pada PT Pos Indonesia (Persero) Cabang Palembang Sumatera Selatan adalah program yang digunakan untuk menampung semua keluhan pelanggan terhadap kepuasan menggunakan layanan jasa pos dan memudahkan perusahaan untuk mengakses mengetahui apa-apa saja yang harus diperbaiki demi menjaga keamanan dan kenyamanan pelanggan terhadap PT Pos Indonesia (Persero) Cabang Palembang Sumatera Selatan.

2.4. Teori Program

2.4.1. Pengertian HTML

Rudiyanto (2011:43), “*Hypertext Markup Language* merupakan salah satu format yang digunakan dalam pembuatan dokumen dan aplikasi yang berjalan di halaman web.”

Sujatmiko (2012:128), “HTML(*Hypertext Markup Language*) adalah bahasa komputer yang digunakan untuk membuat sebuah halaman *web page* atau program yang digunakan untuk menulis (membuat) halaman web di internet.”

2.4.2. Pengertian PHP (*Hypertext Preprocessor*)

Rudiyanto (2011:43), “PHP (*Hypertext Processing*) merupakan bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis.”



Sujatmiko (2012:213), “PHP adalah bahasa pemrograman yang bisa kita gunakan untuk membuat aplikasi web.”

2.4.2.1. Tipe Data *PHP*

Risnandar dkk (2013:60), PHP mempunyai tipe data dasar yaitu:

1. *Tipe Data Integer*

Integer merupakan tipe data yang mencakup semua bilangan bulat. Range bilangan *integer* adalah antara -2.147.483.647 sampai dengan 2.147.483.647.

Contoh *integer* :

\$biaya = 1000;

\$a = 1234;

\$a = -123;

2. *Tipe Data FloatingPoint*

Floatingpoint merupakan tipe data yang mencakup semua bilangan desimal (bilangan yang memiliki angka dibelakang koma). Range bilangan floatingpoint antara 1e-308 sampai dengan 1e308 (10^{-308} s.d. 10^{308}).

Contoh floatingpoint :

\$a = 1.234;

\$a = 1.2e3;

3. *Tipe Data Character*

Character merupakan tipe yang digunakan untuk menyimpan data-data yang berupa karakter (1 huruf). Penulisannya biasanya diapitnya dengannya tandanya kutipnya satunya ('...').

Contoh *character* :

\$abjad = 'a';

\$abjad = 'b';

Jika karakter-karakter berkumpul membentuk suatu kata, kalimat atau paragraf maka tipe tadi tidak bisa lagi disebut sebagai *character* tapi bertipe *string*. *String* merupakan tipe data tersendiri dan tidak dapat dikelompokkan menjadi tipe data dasar. Penulisannya biasa diapit dengan tanda kutip 2 ("...").

Contoh *string* :



```
$nama = "gus";
```

```
$alamat = "Jalan Tanjung";
```

4. Boolean

Boolean menyimpan data yang bernilai logika (benar atau salah, 0 atau 1, ya atau tidak).

2.4.2.2. Script PHP

Utomo (2009:18), "PHP merupakan *embedded language* pada *HTML*". *PHP* dapat dimasukkan pada kode *HTML* dalam satu *file*. Berikut ini contoh sintaks dasar penggunaan *PHP* :

```
<!DOCTYPE html>
<html>
<head>
  <title>Hallo Dunia PHP</title>
</head>
<body>
  <h1>Hallo PHP</h1>
  <?php
    echo "<h2>Hallo, dunia!</h2>";
    phpInfo();
  ?>
</body>
</html>
```

2.4.3. Pengertian MySQL

Raharjo (2011:21), "MySQL merupakan *software RDBMS* (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*)."



2.4.4. Pengertian CSS (*Cascading Style Sheets*)

Saputra dkk (2012:6) menjelaskan, “CSS (*Cascading Style Sheets*) merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web agar lebih rapi, terstruktur dan seragam.”

2.4.5. Hubungan *HTML*, *CSS*, *PHP*, dan *MySQL*

Ketika *user* akan mengakses data perusahaan maka *PHP* akan berusaha untuk mengambil data dari *MySQL* menggunakan salah satu fungsi yang ada di *PHP* yaitu *mysqli_query()*. Setelah data diambil, maka *PHP* akan mengirimkan hasilnya ke *web server* yang berbasis *HTML* dengan menggunakan salah satu fungsi *PHP* yaitu *echo()*. Selanjutnya *HTML* bertugas untuk menampilkan halaman web ke browser sesuai kebutuhan menggunakan tag-tag *HTML* dan dipercantik oleh *CSS*.

2.4.6. *Javascript*

Suryana dan Koesheriyatin (2014:101), “*Java Script* adalah bahasa *script* berdasarkan pada objek yang memperolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen *HTM*”.

Sidik (2011:1), “*Javascript* adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokumen *HTML* yang ditampilkan dalam browser menjadi lebih interaktif, tidak sekedar indah saja.”

Javascript merupakan bahasa pemrograman *client side*. Kode *javascript* dituliskan dalam bentuk fungsi yang ditaruh di tak `<head>` yang dibuka dengan tag `<script type="text/javascript" src="alamat.js">` dan diakhir dengan tag `</script>`.

2.4.6.1 Tipe Data, Variabel dan Operator *JavaScript*

Sidik (2011:31) menjelaskan variabel dan operator yang biasa digunakan *javascript*:

1. Tipe Data



Tipe data yang dapat diproses dalam *Javascript*, dibedakan menjadi :

a. Numerik

Data numerik adalah data berupa bilangan atau angka yang dapat dikenai operasi matematika, seperti penjumlahan, pengurangan, perkalian, atau pembagian.

b. *String*

Data *string* adalah semua data berupa huruf, tanda baca, angka, dan rangkaian huruf dan angka.

c. *Boolean*

Data *boolean* adalah data yang terdiri atau dua nilai, benar atau salah, *true* atau *false*.

d. *Date*

Data *date* adalah data khusus yang menyatakan tentang waktu – waktu tanggal dan jam.

e. Objek

Tipe data objek adalah tipe data yang mendefinisikan objek.

f. *RegExp*

Tipe data *RegExp* adalah tipe data yang digunakan untuk mendefinisikan pola atau *regular expression (regex)*.

g. *Null* dan *undefine*

Tipe data *null* dan *undefine* adalah tipe data khusus. *Null* digunakan untuk mendefinisikan data kosong (*null*), sedangkan *undefined* didefinisikan untuk data yang belum didefinisikan untuk data yang belum didefinisikan, bernilai *null*.

2. Variabel

Variabel adalah tempat untuk menyimpan data atau informasi yang akan diproses, berada dalam memori komputer.

a. Variabel Tunggal

Variabel tunggal adalah variabel yang digunakan untuk menyimpan satu data.

b. *Array*



Array adalah variabel jamak, yaitu satu nama variabel yang dapat memiliki banyak data; beberapa variabel yang diacu dengan satu nama.

c. Lingkup dan Umur Variabel

Tempat mendeklarasi suatu variabel akan menentukan lingkup atau ruang variabel itu dikenali dalam program.

3. Operator

Operator adalah alat yang digunakan untuk melakukan pemrosesan data.

a. Operator Berdasar Jumlah *Operand*.

Operator menurut jumlah *operand* yang dimanipulasinya :

- i. *Unary* adalah operator yang beroperasi ada *operand* tunggal.
- ii. *Binary operand* yang beroperasi untuk dua *operand*.

b. Operator Berdasar Tipe Data

1. Operator *Aritmatika*

Operator *aritmatika* digunakan untuk melakukan pemrosesan data numerik hasilnya adalah data numerik.

2. Operator *Assignment*

Operator *assignment* adalah operator yang digunakan untuk menyatakan suatu pernyataan perintah *assignment* dan ekspresi *aritmatika* dalam bentuk yang disingkat.

3. Operator Bit (*Bitwise Operator*)

4. Operator *String*

Operator *string* adalah operator yang digunakan untuk melakukan pemrosesan data *string* hasilnya berupa data *string*.

5. Operator Perbandingan

Operator perbandingan digunakan untuk melakukan perbandingan dua data, hasilnya berupa nilai *boolean* benar atau salah.

6. Operator Logika

Operator logika (*boolean*) digunakan untuk melakukan pemrosesan data *boolean* hasilnya adalah data *boolean*.