

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem**

Menurut Herlambang (2005:116), definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, sistem merupakan kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai tujuan tertentu.

Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya.

#### **2.2 Aplikasi**

Menurut Noviansyah (2008:4) aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi suatu pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk suatu tugas khusus dapat dibedakan menjadi dua jenis, yaitu:

1. Aplikasi Software Spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.

2. Aplikasi Software Paket, suatu program dengan dokumentasi tergabung yang dirancang untuk masalah tertentu.

### 2.3 Pendataan

Pendataan menurut Herlambang (2005), merupakan suatu proses pencatatan keterangan yang benar dan nyata tentang sesuatu, baik manusia, benda, lingkungan, maupun kejadian tertentu. Pencatatan ini dimaksudkan sebagai suatu dokumentasi atau arsip yang dapat digunakan untuk suatu keperluan di masa depan. Adapun keperluan utama yang lazim menjadi pengagas suatu pendataan adalah pembuatan laporan. Pembuatan laporan dimaksudkan sebagai dasar atau bahan pertimbangan bagi pemimpin organisasi/perusahaan untuk mengambil suatu keputusan.

### 2.4 Barcode

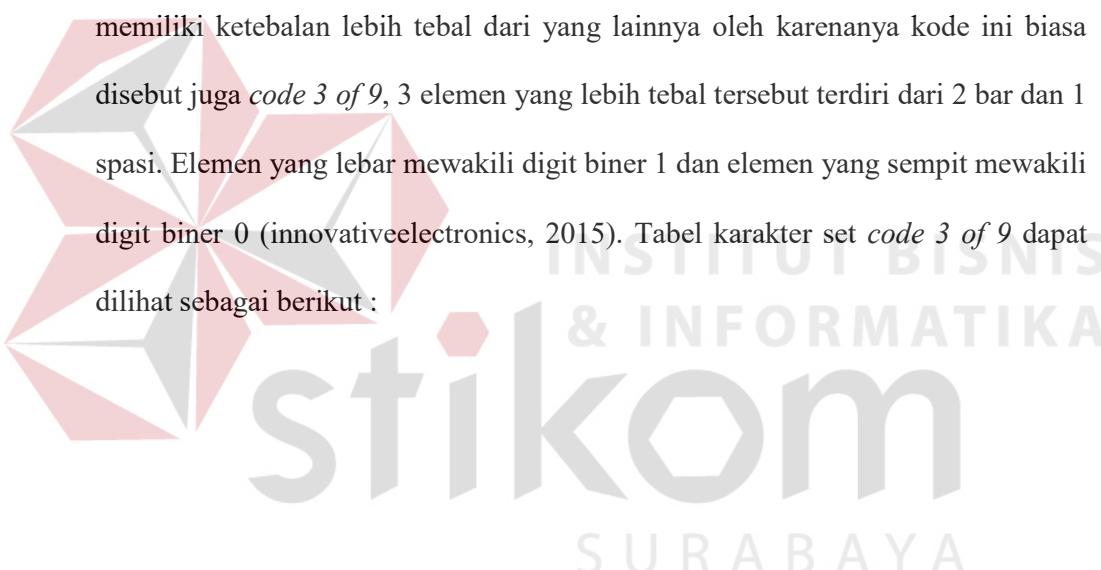
(Malik, Wijaya dan Taufiq, 2010) dalam bukunya, ada banyak definisi *barcode*, tetapi semua definisi tersebut merujuk pada satu hal yang sama. Untuk lebih jelasnya bisa dilihat dibawah ini.

1. *Barcode* secara harfiah berarti kode berbentuk garis.
2. Sebagai kumpulan kode yang berbentuk garis, masing-masing ketebalan setiap garis berbeda sesuai dengan isi kodennya.
3. Informasi terbacakan mesin (*machine readable*) dalam format visual yang tercetak. Umumnya barcode berbentuk garis-garis vertikal tipis tebal yang terpisah oleh jarak tertentu.
4. Sejenis kode yang mewakili data atau informasi tertentu, biasanya jenis dan harga barang, seperti makanan dan buku. Kode berbentuk batangan balok dan

berwarna hitam putih ini mengandung satu kumpulan yang berlainan ukuran yang disusun sedimikian rupa. Kode ini dicetak di atas stiker atau di kotak pembungkus batang.

Dalam penelitian ini akan menggunakan jenis barcode *Code 39/ 3 of 9*.

*Code 39* dapat mengkodekan karakter *alphanumeric* yaitu angka decimal dan huruf besar serta tambahan karakter spesial -.\*\$/%+. Satu karakter dalam *Code 39* terdiri dari 9 elemen yaitu, 5 bar (garis vertical hitam) dan 4 spasi (garis vertical putih) yang disusun bergantian antara bar dan spasi. 3 dan 9 elemen tersebut memiliki ketebalan lebih tebal dari yang lainnya oleh karenanya kode ini biasa disebut juga *code 3 of 9*, 3 elemen yang lebih tebal tersebut terdiri dari 2 bar dan 1 spasi. Elemen yang lebar mewakili digit biner 1 dan elemen yang sempit mewakili digit biner 0 (innovativeelectronics, 2015). Tabel karakter set *code 3 of 9* dapat dilihat sebagai berikut :



ASCII	Karakter Set										Nilai Karakter
	B	S	B	S	B	S	B	S	B		
0	0	0	0	1	1	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	0	0	1	1	2
3	1	0	1	1	0	0	0	0	0	0	3
4	0	0	0	1	1	0	0	0	1	1	4
5	1	0	0	1	1	0	0	0	0	0	5
6	0	0	1	1	1	0	0	0	0	0	6
7	0	0	0	1	0	0	1	0	1	1	7
8	1	0	0	1	0	0	1	0	0	0	8
9	0	0	1	1	0	0	1	0	0	0	9
A	1	0	0	0	0	1	0	0	1	10	
B	0	0	1	0	0	1	0	0	1	11	
C	1	0	1	0	0	1	0	0	0	12	
D	0	0	0	0	1	1	0	0	1	13	
E	1	0	0	0	1	1	0	0	0	14	
F	0	0	1	0	1	1	0	0	0	15	
G	0	0	0	0	0	1	1	0	1	16	
H	1	0	0	0	0	1	1	0	0	17	
I	0	0	1	0	0	1	1	0	0	18	
J	0	0	0	0	1	1	1	0	0	19	
K	1	0	0	0	0	0	0	1	1	20	
L	0	0	1	0	0	0	0	1	1	21	
M	1	0	1	0	0	0	0	1	0	22	
N	0	0	0	0	1	0	0	1	1	23	
O	1	0	0	0	1	0	0	1	0	24	
P	0	0	1	0	1	0	0	1	0	25	
Q	0	0	0	0	0	0	1	1	1	26	
R	1	0	0	0	0	0	1	1	0	27	
S	0	0	1	0	0	0	1	1	0	28	
T	0	0	0	0	1	0	1	1	0	29	
U	1	1	0	0	0	0	0	0	1	30	
V	0	1	1	0	0	0	0	0	1	31	
W	1	1	1	0	0	0	0	0	0	32	
X	0	1	0	0	1	0	0	0	1	33	
Y	1	1	0	0	1	0	0	0	0	34	
Z	0	1	1	0	1	0	0	0	0	35	
-	0	1	0	0	0	0	1	0	1	36	
.	1	1	0	0	0	0	1	0	0	37	
SPACE	0	1	1	0	0	0	1	0	0	38	
*	0	1	0	0	1	0	1	0	0	-	
\$	0	1	0	1	0	1	0	0	0	39	
/	0	1	0	1	0	0	0	1	0	40	
+	0	1	0	0	0	1	0	1	0	41	
%	0	0	0	1	0	1	0	1	0	42	

Gambar 2.1. Karakter Set Code 3 of 9

Struktur penulisan barcode *code 39* dapat dilihat pada gambar sebagai berikut :



Gambar 2.2. Struktur Penulisan Barcode Code 39

Keterangan :

X : Ketebalan elemen yang sempit (minimum 0.19mm).

QZ : Quiet Zone dengan ketebalan minimum 10 kali X.

SC : Start Character (karakter\*).

ICG : Inter Character Gap dengan ketebalan 1 kali X.

CC : Check Character.

PC : Stop Character (karakter\*).

Untuk dapat membedakan garis vertikal lebar dan sempit maka perbandingan ketebalan antara garis vertikal lebar dan sempit minimum 2:1, dimana perbandingan 3:1 akan lebih baik. *Check character code 3 of 9* adalah sisa dari jumlah seluruh nilai karakter di modulus dengan total jumlah nilai karakter yaitu 43, sebagai contoh:

Message	:	B02
Karakter	:	B 0 2
Nilai karakter	:	11 0 2
Jumlah	:	$11 + 0 + 2 = 13$
13 mode 43	=	13

13 adalah nilai dari karakter D, maka hasil dari penjumlahan *Message + check character* adalah B02D.

Pada umumnya setiap *check character* telah terintegrasi di dalam *software* penulisan *barcode* yang bisa didapatkan secara cuma-cuma pada *internet*.

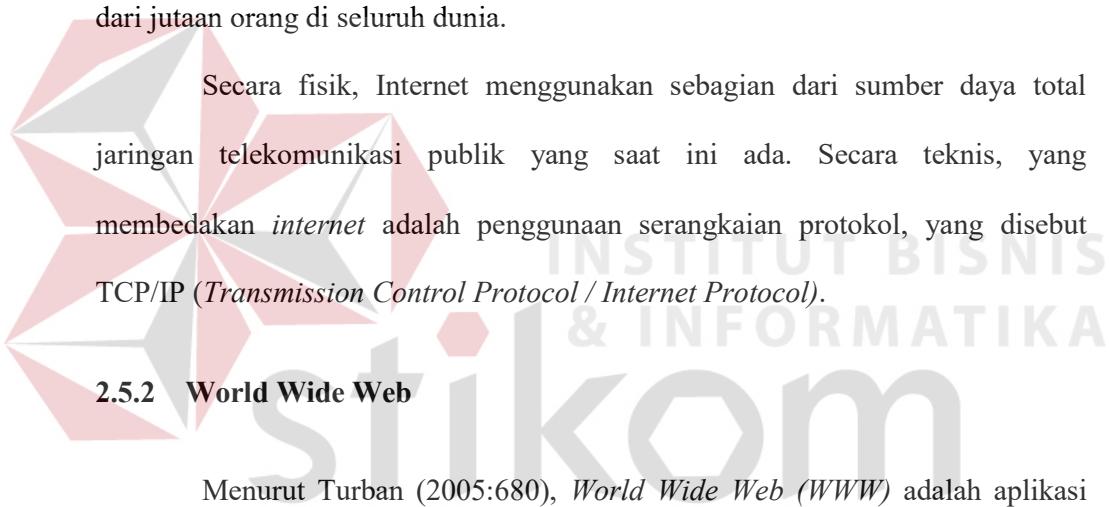
## 2.5 Sistem Berbasis Web

Menurut Turban (2005:69), Sistem berbasis *web* adalah aplikasi atau layanan yang berada dalam *server* dan dapat diakses dengan menggunakan penjelajah *web* dan karenanya dapat diakses dari mana saja melalui *internet*. Satu-

satunya peranti lunak sisi klien yang dibutuhkan untuk mengakses dan menjalankan aplikasi berbasis *web* adalah lingkungan penjelajah *web*, dan berbagai aplikasi tersebut harus sesuai dengan *protocol internet*-nya.

### 2.5.1 Internet

Menurut Turban (2005:69), *Internet* adalah sistem global jaringan komputer sebuah jaringan untuk berbagai jaringan. Kini, *Internet* adalah fasilitas publik yang kooperatif dan dapat berjalan sendiri, yang dapat diakses oleh ratusan dari jutaan orang di seluruh dunia.



Secara fisik, *Internet* menggunakan sebagian dari sumber daya total jaringan telekomunikasi publik yang saat ini ada. Secara teknis, yang membedakan *internet* adalah penggunaan serangkaian protokol, yang disebut TCP/IP (*Transmission Control Protocol / Internet Protocol*).

### 2.5.2 World Wide Web

Menurut Turban (2005:680), *World Wide Web (WWW)* adalah aplikasi yang menggunakan fungsi transport tersebut. *Web* adalah sistem dengan standar yang diterima secara *universal* untuk menyimpan, menelusuri, memformat dan menampilkan informasi melalui arsitektur klien / *server*.

### 2.5.3 Web 2.0

Istilah *Web 2.0* pertama kali diperkenalkan oleh O'Reilly Media pada tahun 2004 sebagai teknologi *Web* generasi kedua yang mengedepankan kolaborasi dan *sharing* informasi secara *online*. Menurut Tim O'Reilly (2005) , *Web 2.0* dapat didefinisikan sebagai revolusi bisnis diindustri komputer yang

disebabkan oleh penggunaan *internet* sebagai *platform*, dan merupakan suatu percobaan untuk memahami berbagai aturan untuk mencapai keberhasilan pada *platform* baru tersebut.

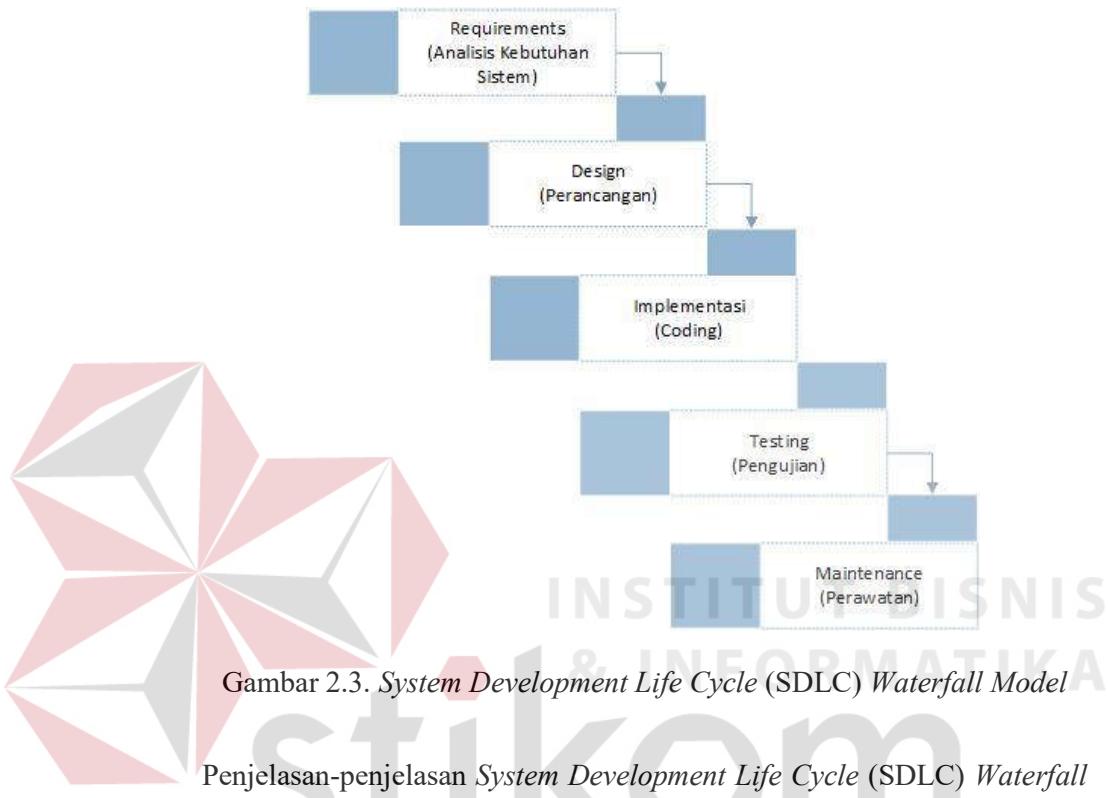
Berbagai layanan berbasis *web* seperti jejaring sosial, wiki dan folksonomies (misalnya: “flickr.com”, “del.icio.us”) merupakan teknologi *Web* 2.0 yang menambah interaktifitas di antara para pengguna *web*. Pada umumnya, *Website* yang dibangun dengan menggunakan teknologi *Web* 2.0 memiliki fitur-fitur sebagai berikut:

1. *CSS (Cascading Style Sheets)*.
2. Aplikasi Rich *Internet* atau berbasis Ajax
3. Markup XHTML
4. Sindikasi dan agregasi data menggunakan RSS/Atom
5. URL yang valid
6. Folksonomies
7. Aplikasi wiki pada sebagian atau seluruh *Website*
8. XML Web-Service API

## 2.6 Pengembangan Sistem

Menurut Pressman (2001), Model *System Development Life Cycle (SDLC)* ini biasa disebut juga dengan model *waterfall* atau disebut juga *classic life cycle*. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah *Requirements* (analisis sistem), *Analysis* (analisis kebutuhan sistem), *Design* (perancangan), *Coding* (implementasi), *Testing* (pengujian) dan *Maintenance* (perawatan). Model ini

memungkinkan proses pengembangan lebih terlihat. Hal ini dikarenakan bentuknya yang bertingkat ke bawah dari satu fase ke fase lainnya, model ini dikenal dengan model *waterfall*, seperti terlihat pada gambar 5 berikut ini:



1. *Requirement* (Analisis Kebutuhan Sistem)

Pada tahap awal ini dilakukan analisa guna menggali secara mendalam kebutuhan yang akan dibutuhkan. Kebutuhan ada bermacam-macam seperti halnya kebutuhan informasi bisnis, kebutuhan data dan kebutuhan user itu sendiri. Kebutuhan itu sendiri sebenarnya dibedakan menjadi tiga jenis kebutuhan. Pertama tentang kebutuhan teknologi. Dari hal ini dilakukan analisa mengenai kebutuhan teknologi yang diperlukan dalam pengembangan suatu sistem, seperti halnya data penyimpanan informasi / *database*. Kedua

kebutuhan informasi, contohnya seperti informasi mengenai visi dan misi perusahaan, sejarah perusahaan, latar belakang perusahaan. Ketiga, Kebutuhan *user*. Dalam hal ini dilakukan analisa terkait kebutuhan user dan kategori user.

## 2. *Design* (Perancangan)

Selanjutnya, hasil analisa kebutuhan sistem tersebut akan dibuat sebuah *design database*, DFD, ERD, antarmuka pengguna/ *Graphical User Interface (GUI)* dan jaringan yang dibutuhkan untuk sistem. Selain itu juga perlu dirancang struktur datanya, arsitektur perangkat lunak, detil prosedur dan karakteristik tampilan yang akan disajikan. Proses ini menterjemahkan kebutuhan sistem ke dalam sebuah model perangkat lunak yang dapat diperkirakan kualitasnya sebelum memulai tahap implementasi.

## 3. *Implementation (Coding)*

Rancangan yang telah dibuat dalam tahap sebelumnya akan diterjemahkan ke dalam suatu bentuk atau bahasa yang dapat dibaca dan diterjemahkan oleh komputer untuk diolah. Tahap ini juga dapat disebut dengan tahap implementasi, yaitu tahap yang mengkonversi hasil perancangan sebelumnya ke dalam sebuah bahasa pemrograman yang dimengerti oleh komputer. Kemudian komputer akan menjalankan fungsi-fungsi yang telah didefinisikan sehingga mampu memberikan layanan-layanan kepada penggunanya.

## 4. *Testing* (Pengujian)

*Testing* dapat digunakan untuk memastikan kesesuaian dalam proses *input*, sehingga dapat menghasilkan *output* yang sesuai. Pada tahap ini terdapat 2

metode pengujian perangkat yang dapat digunakan, yaitu: metode *black-box* dan *white-box*. Pengujian dengan metode *black-box* merupakan pengujian yang menekankan pada fungsionalitas dari sebuah perangkat lunak tanpa harus mengetahui bagaimana struktur di dalam perangkat lunak tersebut. Sebuah perangkat lunak yang diuji menggunakan metode *black-box* dikatakan berhasil jika fungsi-fungsi yang ada telah memenuhi spesifikasi kebutuhan yang telah dibuat sebelumnya. Pengujian dengan menggunakan metode *white-box* yaitu menguji struktur internal perangkat lunak dengan melakukan pengujian pada algoritma yang digunakan oleh perangkat lunak.

#### 5. *Maintenance* (Perawatan)

Tahap terakhir dari metode SDLC ini adalah *maintenance*. Pada tahap ini, jika sistem sudah sesuai dengan tujuan yang ditentukan dan dapat menyelesaikan masalah pada pendataan jemaat dan aktivitas gerejawi, maka akan diberikan kepada pengguna. Setelah digunakan dalam periode tertentu, pasti terdapat penyesuaian atau perubahan sesuai dengan keadaan yang diinginkan, sehingga membutuhkan perubahan terhadap sistem tersebut. Tahap ini dapat pula diartikan sebagai tahap penggunaan perangkat lunak yang disertai dengan perawatan dan perbaikan. Perawatan dan perbaikan suatu perangkat lunak diperlukan, karena dalam prakteknya ketika perangkat lunak digunakan terkadang masih terdapat kekurangan.

### 2.7 Testing dan Implementasi Sistem

Menurut Romeo (2003:3), *testing software* adalah proses mengoperasikan *software* dalam suatu kondisi yang dikendalikan untuk :

1. Verifikasi. Apakah telah berlaku sebagaimana yang ditetapkan (menurut spesifikasi)?
2. Mendeteksi *error*.
3. Validasi. Apakah spesifikasi yang ditetapkan telah memenuhi keinginan atau kebutuhan pengguna yang sebenarnya?

Menurut Romeo (2003:33), *test case* merupakan tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya. Metode testing ini dibagi menjadi dua, yaitu *white box testing* dan *black box testing*.

## 2.8 Blackbox Testing

Menurut Romeo. (2003), *Black box testing* atau *behavioral testing* atau *specification-based testing*, *input/ output testing* atau *functional testing* dilakukan tanpa sepengetahuan detil struktur internal dari sistem atau komponen yang dites.

*Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*.

Menggunakan *black box testing*, perekayasa *software* dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Kategori *error* dapat diketahui melalui *black box testing*, antara lain :

1. Fungsi yang hilang atau tidak benar.
2. *Error* dari antar-muka.
3. *Error* dari struktur data atau akses *eksternal database*.
4. *Error* dari kinerja atau tingkah laku.
5. *Error* dari inisialisasi dan terminasi.