

BAB II

LANDASAN TEORI

1.1. Implementasi

Menurut Usman dalam (Labib 2014:7) Implementasi adalah bermuara pada aktivitas aksi, tindakan atau adanya mekanisme dalam suatu sistem. Implementasi bukan sekedar aktivitas, tetapi suatu kegiatan terencana untuk mencapai suatu tujuan kegiatan.

Implementasi adalah tahapan untuk mewujudkan sebuah rancangan yang telah di desain pada tahapan awal dengan pendekatan tertentu dalam bentuk perangkat lunak.(Zainudin Zuhri, 2014).

Jadi, implementasi adalah tahapan dalam bentuk aktivitas untuk mewujudkan rancangan yang telah di desain pada tahapan awal untuk mencapai suatu tujuan kegiatan.

1.2. Aplikasi

Menurut Hengky W. Pramana (2012), aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.

Menurut Desy Anwar (2010:5), aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna, biasanya

dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolahan kata, lembar kerja dan pemutaran media.

Jadi, aplikasi adalah suatu unit perangkat lunak yang memanfaatkan kemampuan komputer itu sendiri yang digunakan untuk mengerjakan suatu tugas yang menguntungkan *user*.

1.3. Sertifikat TOEFL

Menurut Kamus Besar Bahasa Indonesia (KBBI) sertifikat berarti tanda atau surat keterangan (pernyataan) tertulis atau tercetak dari orang yang berwenang yang dapat digunakan sebagai bukti pemilikan atau suatu kejadian.

Test of English As Foreign Language (TOEFL) adalah ujian kemampuan berbahasa Inggris (logat Amerika) yang diperlukan untuk mendaftar masuk ke universitas di Amerika Serikat atau negara-negara lain di dunia. Tes ini diperuntukan bagi pendaftar atau pembicara yang bahasa ibunya bukan bahasa Inggris. Jenis tes bahasa Inggris TOEFL ini pada umumnya diperlukan untuk persyaratan masuk kuliah pada hampir semua universitas di Amerika Serikat dan Kanada baik untuk program *undergraduate* (S-1) maupun *graduate* (S-2 atau S-3). Selain itu, dewasa ini TOEFL sudah digunakan dalam dunia kerja sebagai salah satu mekanisme *recruitment* atau jenjang kenaikan pangkat.

Jadi, sertifikat TOEFL merupakan sebuah keterangan atau pernyataan kelulusan ujian TOEFL. Sertifikat ini biasanya dikeluarkan oleh lembaga penyelenggara tes, sehingga dapat dipertanggung jawabkan.

1.4. Pemalsuan

Menurut KBBI, pemalsuan adalah proses, cara, perbuatan memalsu;~ ijazah upaya atau tindakan memalsukan ijazah dengan meniru bentuk asli;~ intelektual pemalsuan isi surat atau tulisan;~ sertifikat upaya atau tindakan memalsukan sertifikat dengan membuat bentuk atau penandatanganan yang serupa dengan aslinya;~ tanda tangan upaya atau tindakan memalsukan tanda tangan dengan meniru bentuk tanda tangan yang dipalsukan;~ uang upaya atau tindakan memalsukan mata uang dengan mencetak uang yang mirip dengan aslinya;

Perbuatan pemalsuan merupakan suatu jenis pelanggaran terhadap kebenaran dan kepercayaan dengan tujuan memperoleh keuntungan bagi diri sendiri atau orang lain. Suatu pergaulan hidup yang teratur dalam masyarakat yang maju dan teratur tidak dapat berlangsung lama tanpa adanya jaminan kebenaran atas beberapa bukti surat dan dokumen-dokumen lainnya. Karenanya perbuatan pemalsuan merupakan ancaman bagi kelangsungan hidup dari masyarakat tersebut.

Masyarakat telah diciptakan untuk hidup bermasyarakat dalam suasana hidup bermasyarakat itulah ada perasaan saling ketergantungan satu sama lain. Didalamnya terdapat tuntutan kebiasaan, aspirasi, norma, nilai

kebutuhan dan sebagainya. Kesemuanya ini dapat berjalan sebagaimana mestinya jika ada keseimbangan pemahaman kondisi sosial tiap pribadi. Tetapi keseimbangan tersebut dapat goyah bilamana dalam masyarakat tersebut terdapat ancaman yang salah satunya berupa tindak kejahatan pemalsuan. (Harni Eka Putri B., 2013).

Selain itu pemalsuan telah diatur dalam pasal 263 Kitab Undang-undang Hukum Pidana (KUHP) yang berbunyi :

- (1) *Barang siapa membuat surat palsu atau memalsukan surat yang dapat menimbulkan sesuatu hak, perikatan atau pembebasan hutang, atau yang diperuntukan sebagai bukti daripada sesuatu hal dengan maksud untuk memakai atau menyuruh orang lain memakai surat tersebut seolah-olah isinya benar dan tidak palsu, diancam jika pemakaian tersebut dapat menimbulkan kerugian, karena pemalsuan surat, dengan pidana penjara paling lama enam bulan.*
- (2) *Diancam dengan pidana yang sama, barang siapa dengan sengaja memakai surat palsu atau yang dipalsukan seolah-olah sejati, jika pemakaian surat itu dapat menimbulkan kerugian.*

1.5. Kriptografi

Kata kriptografi (*cryptography*) berasal dari bahasa Yunani yaitu dari kata *kryptos* yang artinya tersembunyi dan *graphein* yang artinya menulis. Kriptografi dapat diartikan tulisan yang dirahasiakan atau dapat diartikan juga sebagai suatu ilmu ataupun seni yang mempelajari bagaimana sebuah

data, informasi dan dokumen dikonversi ke bentuk tertentu yang sulit dimengerti. (Hartini, 2014).

Menurut Rifki Sadikin dalam bukunya “*Kriptografi untuk keamanan Jaringan*” (2012) mengemukakan bahwa kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas.

Ilmu kriptografi sebenarnya sudah lama dipakai oleh orang zaman dulu. Dalam sejarah, beberapa kerajaan yang akan mengirim pesan ke kerajaan lainnya, isi dari pesan tersebut diubah sedemikian hingga orang lain kesulitan untuk membacanya. Hanya penerima saja yang dapat membaca isi pesan tersebut.

Tujuan utama dari kriptografi yang ingin dicapai antara lain :

1. **Secrecy**, informasi dengan aman. Pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima atau pihak-pihak yang memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematik yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
2. **Integrity**, informasi tidak rusak. Mampu mengenal/mendeteksi adanya manipulasi (penghapusan, perubahan atau penambahan) data yang tidak sah (oleh pihak lain).

3. **Authentication**, informasi hanya dapat diakses oleh yang berhak. Mampu melakukan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/infromasi.

4. **Non-repudiation**, informasi yang sekali kirim. Dilayanan ini pesan hanya sekali kirim sehingga dapat menyangkal pihak lain pesan dari pihak yang bukan pengirim.

Dalam kriptografi ada beberapa istilah yang sering digunakan yaitu :

1. **Plaintext**, semua data yang belum diproses melalui suatu algoritma kriptografi. Plaintext dapat berupa teks, *image*, atau bentuk lain.

2. **Chipertext**, hasil proses algoritma kriptografi dari suatu plaintext.

3. **Enkripsi (Encryption)**, proses pengubahan dari plaintext ke chipertext dengan algoritma kriptografi tertent.

4. **Dekripsi (Decryption)**, proses pengubahan dari plaintext ke chipertext dengan algoritma kriptografi tertentu.

5. **Chiper**, suatu algoritma kriptografi yang dapat melakukan proses enkripsi dan dekripsi.

6. **Key**, parameterkunci yang digunakan oleh chiper dalam melakukan proses enkripsi dan dekripsi.

7. **Kriptosistem**

Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi

8. Kriptanalisis

Kriptanalisis merupakan suatu ilmu untuk mendapatkan plainteks tanpa harus mengetahui kunci secara wajar.

1.5.1. Kriptografi Modern

Kriptografi modern berorientasi bit sebab penyandian modern menggunakan media komputer untuk mengolah pesan. Pesan pada sandi modern tidak selalu berupa rangkaian karakter bisa saja berupa rangkaian bit seperti berkas video atau berkas gambar.

Kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Algoritma kriptografi modern terdiri dari dua bagian :

1. Kunci Simetri

Proses enkripsi dan dekripsi data secara keseluruhan menggunakan kunci yang sama. Istilah lain dari kriptografi kunci simetri ini adalah kriptografi kunci privat (*private key cryptography*), kriptografi kunci rahasia (*secret key cryptography*) atau kriptografi konvensional.

Skema ini berdasarkan jumlah data per proses dan alur pengolahan data didalamnya dibedakan menjadi dua kelas, yaitu block-chiper dan stream-chiper. Berikut merupakan contoh yang menggunakan kunci simetri antara lain :

1. Data Encryption Standard (DES)
 2. Advanced Encryption Standard (AES)
 3. International Data Encryption Algorithm (IDEA)
 4. RC4
2. Kunci Asimetri

Kunci asimetri menggunakan kunci yang berbeda dalam setiap proses enkripsi dan dekripsinya. Kunci ini sering disebut juga kunci publik (*public-key*) karena kunci enkripsinya dibuat supaya diketahui oleh orang lain dalam melakukan enkripsinya. Dalam tahap dekripsinya kunci yang digunakan hanya diketahui oleh yang berhak memiliki kunci rahasia. Entitas pengirim akan mengenkripsi dengan menggunakan kunci publik, sedangkan entitas penerima mendekripsikan menggunakan kunci privat.

Dari beberapa algoritma yang ada dalam kriptografi modern yang penulis gunakan dalam penelitian ini menggunakan algoritma RSA.

1.6. Algoritma Rivest Shamir Adleman (RSA)

Dari sekian banyak algoritma kriptografi kunci publik yang pernah dibuat, algoritma RSA merupakan algoritma yang paling populer dan cukup aman dikarenakan algoritma RSA melakukan pemfaktoran bilangan yang cukup besar.

Algoritma ini adalah algoritma pertama yang diketahui paling cocok untuk menandai (*signing*) dan untuk enkripsi (*encryption*) dan salah satu penemuan besar pertama dalam kriptografi kunci publik. RSA masih digunakan secara luas dalam protokol-protokol perdagangan elektronik, dan dipercaya sangat aman karena diberikan kunci-kunci yang cukup panjang dan penerapannya yang sangat *up-to-date* (mutakhir).

1.6.1. Perumusan Algoritma RSA

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci pribadi. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin. Dikarenakan makin besar ukuran kunci, maka makin besar juga kemungkinan kombinasi kunci yang bisa dijebol dengan metode memeriksa kombinasi satu persatu atau lebih dikenal dengan istilah brute force attack sehingga tidak ekonomis dan sulit untuk memecahkan sandi tersebut. (M. Zaki Riyanto dan Ardhi Ardian, 2008).

Terdapat beberapa besaran yang digunakan pada algoritma RSA, yakni :

1. p dan q bilangan prima (rahasia)
2. $N = p \times q$ (tidak rahasia)
3. $f(N) = (p-1)(q-1)$ (rahasia)

4. (kunci enkripsi) (tidak rahasia)
5. (kunci dekripsi) (rahasia)
6. X (plainteks) (rahasia)
7. Y (chiperteks) (tidak rahasia)

1. Proses Pembangkitan Kunci

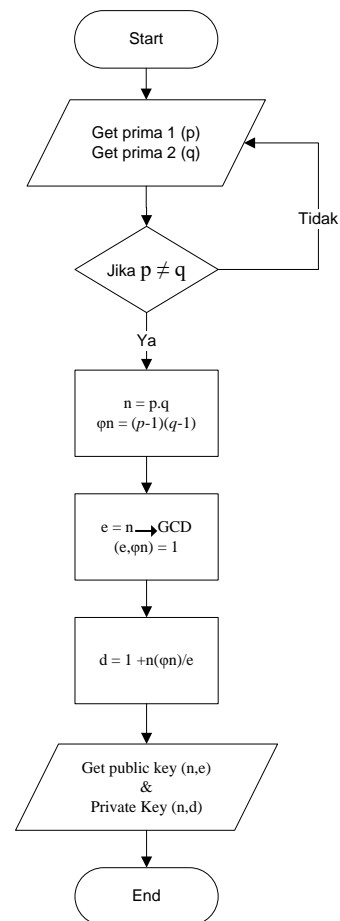
Berikut ini adalah langkah-langkah dalam membangkitkan kunci, yaitu sebagai berikut :

- a. Pilih dua bilangan prima p dan q secara acak , dimana $p \neq q$. Bilangan ini harus cukup besar (minimal 100 digit) untuk memperoleh tingkat keamanan yang tinggi.
- b. Hitung $n = p \cdot q$ ($p \neq q$, karena jika $p=q$ maka nilai $n=p^2$ dan akan mudah mendapatkan nilai n), bilangan n disebut Parameter sekuriti.
- c. Hitung $\Phi n = (p-1)(q-1)$.
- d. Pilih bilangan bulat (*integer*) sebagai kunci publik antara satu dan Φn ($1 < e < \Phi n$) yang tidak mempunyai faktor pembagi dari Φn , sehingga $\gcd(e, \Phi n) = 1$. Dapat dikatakan bahwa e relatif prima terhadap Φn .
- e. Menghitung kunci dekripsi atau kunci privat d dengan kekongruenan $e \cdot d \equiv 1 \pmod{\Phi n}$, sehingga d dapat dihitung dengan cara yang sederhana dengan persamaan $d = \frac{1 \pmod{\Phi n}}{e}$

Dengan mencoba nilai $d = 0,1,2,3\dots$ sehingga memenuhi persamaan tersebut. Hasil dari algoritma tersebut adalah :

1. Kunci publik adalah pasangan (e,n)
2. Kunci privat adalah pasangan (d,n)

Jika digambarkan dengan flowchart maka sebagai berikut :

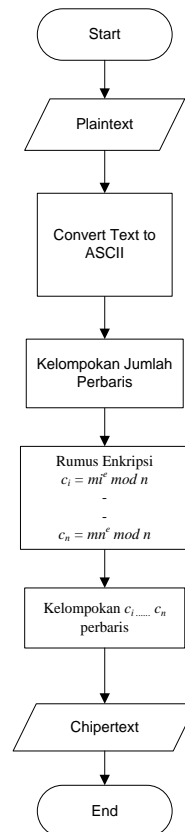


Gambar 2.1. *Flowchart* Proses Pembentukan Kunci

2. Proses Enkripsi

Untuk melakukan proses enkripsi, terlebih dahulu tentukan kunci publik (e,n) . Masukkan *plainteks* (m) atau pesan asli yang akan dienkripsi dan ubah ke dalam bentuk kode ASCII (desimal), pesan yang akan dienkripsi dibagi ke dalam blok-blok yang lebih kecil dari n , dikarenakan n harus lebih besar atau sama dengan m . Rumus enkripsinya adalah : $c = m^e \bmod n$.

Jika digunakan dengan *flowchart* untuk proses enkripsi maka sebagai berikut:



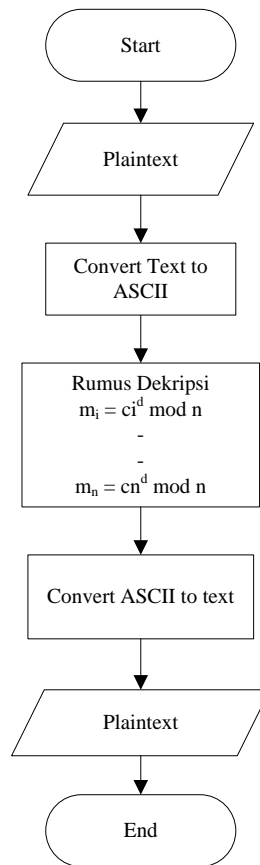
Gambar 2.2. *Flowchart* proses enkripsi

3. Proses Dekripsi

Setelah menerima pesan yang sudah terenkripsi maka penerima pesan akan melakukan proses dekripsi dengan rumus :

$$m = c^d \text{ mod } n$$

Jika digunakan dengan *flowchart* untuk proses dekripsi maka sebagai berikut :



Gambar 2.3. *Flowchart* proses dekripsi

1.7. Kode ASCII

Kode ASCII atau kepanjangan dari *American Standard Code for International Interchange* merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal. Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks.

Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit *significant* paling tinggi. Bit tambahan ini sering digunakan untuk uji prioritas. Jumlah kode ASCII adalah 255 kode. Kode

ASCII 0 - 127 merupakan kode ASCII untuk manipulasi teks. Kode ASCII 128 – 255 merupakan kode untuk manipulasi grafik. Tabel karakter ASCII yaitu sebagai berikut :

Tabel 2.1. Tabel Karakter ASCII (Sumber : Ghina Citro Nurwanti, 2015)

Karakter	Hexadesimal	Desimal	Keterangan
NUL	0000	0	Null (tidak tampak)
SOH	0001	1	Start of Heading (tidak tampak)
STX	0002	2	Start of Text (tidak tampak)
ETX	0003	3	End of Text (tidak tampak)
EOT	0004	4	End of Transmission (tidak tampak)
ENQ	0005	5	Enquiry (tidak tampak)
ACK	0006	6	Acknowledge (tidak tampak)
BEL	0007	7	Bell (tidak tampak)
BS	0008	8	Menghapus satu karakter di belakang kursor (Backspace)
HT	0009	9	Horizontal Tabulation
LF	000A	10	Pergantian Baris (Line Feed)
VT	000B	11	Tabulasi vertikal
FF	000C	12	Pergantian Baris (Form Feed)
CR	000D	13	Pergantian Baris (Carriage return)
SO	000E	14	Shift out (tidak tampak)
SI	000F	15	Shift in (tidak tampak)
DLE	0010	16	Data link escape (tidak tampak)

DC1	0011	17	Device Control 1 (tidak nampak)
DC2	0012	18	Device Control 2 (tidak nampak)
DC3	0013	19	Device Control 3 (tidak nampak)
DC4	0014	20	Device Control 4 (tidak nampak)
NAK	0015	21	Negative acknowledge (tidak nampak)
SYN	0016	22	Synchronour Idle (tidak nampak)
ETN	0017	23	End of transmission (tidak nampak)
CAN	0018	24	Cancel (tidak nampak)
EM	0019	25	End of Medium (tidak nampak)
SUB	001A	26	Substitute (tidak nampak)
ESC	001B	27	Escape (tidak nampak)
FS	001C	28	File separator
GS	001D	29	Group separator
RS	001E	30	Record Separator
US	001F	31	Unit Separator
SP	0020	32	Spasi
!	0021	33	Tanda seru (exclamation)
“	0022	34	Tanda kutip dua
\$	0023	35	Tanda pagar (kres)
\$	0024	36	Tanda mata uang dolar
%	0025	37	Tanda persen
&	0026	38	Karakter Ampersand (&)

'	0027	39	Karakter Apostrof
(0028	40	Tanda kurung buka
)	0029	41	Tanda kurung tutup
*	002A	42	Karakter asterisk (bintang)
+	002B	43	Tanda Tambah (plus)
,	002C	44	Karakter koma
-	002D	45	Karakter hyphen (strip)
.	002E	46	Tanda titik
/	002F	47	Garis miring (<i>slash</i>)
0	0030	48	Angka nol
1	0031	49	Angka satu
2	0032	50	Angka dua
3	0033	51	Angka Tiga
4	0034	52	Angka Empat
5	0035	53	Angka Lima
6	0036	54	Angka Enam
7	0037	55	Angka Tujuh
8	0038	56	Angka Delapan
9	0039	57	Angka Sembilan
:	003A	58	Tanda titik dua
;	003B	59	Tanda titik koma
<	003C	60	Tanda lebih kecil

=	003D	61	Tanda sama dengan
>	003E	62	Tanda lebih besar
?	003F	63	Tanda tanya
@	0040	64	A Keong (@)
A	0041	65	Huruf latin A kapital
B	0042	66	Huruf latin B kapital
C	0043	67	Huruf latin C kapital
D	0044	68	Huruf latin D kapital
E	0045	69	Huruf latin E kapital
F	0046	70	Huruf latin F kapital
G	0047	71	Huruf latin G kapital
H	0048	72	Huruf latin H kapital
I	0049	73	Huruf latin I kapital
J	004A	74	Huruf latin J kapital
K	004B	75	Huruf latin K kapital
L	004C	76	Huruf latin L kapital
M	004D	77	Huruf latin M kapital
N	004E	78	Huruf latin N kapital
O	004F	79	Huruf latin O kapital
P	0050	80	Huruf latin P kapital
Q	0051	81	Huruf latin Q kapital
R	0052	82	Huruf latin R kapital

S	0053	83	Huruf latin S kapital
T	0054	84	Huruf latin T kapital
U	0055	85	Huruf latin U kapital
V	0056	86	Huruf latin V kapital
W	0057	87	Huruf latin W kapital
X	0058	88	Huruf latin X kapital
Y	0059	89	Huruf latin Y kapital
Z	005A	90	Huruf latin Z kapital
[005B	91	Kurung siku kiri
\	005C	92	Garis miring terbalik (<i>Backslash</i>)
]	005D	93	Kurung siku kanan
^	005E	94	Tanda pangkat
_	005F	95	Garis Bawah (<i>underscore</i>)
`	0060	96	Tanda petik satu
a	0061	97	Huruf latin a kecil
b	0062	98	Huruf latin b kecil
v	0063	99	Huruf latin c kecil
d	0064	100	Huruf latin d kecil
e	0065	101	Huruf latin e kecil
f	0066	102	Huruf latin f kecil
g	0067	103	Huruf latin g kecil
h	0068	104	Huruf latin h kecil

i	0069	105	Huruf latin i kecil
j	006A	106	Huruf latin j kecil
k	006B	107	Huruf latin k kecil
l	006C	108	Huruf latin l kecil
m	006D	109	Huruf latin m kecil
n	006E	110	Huruf latin n kecil
o	006F	111	Huruf latin o kecil
p	0070	112	Huruf latin p kecil
q	0071	113	Huruf latin q kecil
r	0072	114	Huruf latin r kecil
s	0073	115	Huruf latin s kecil
t	0074	116	Huruf latin t kecil
u	0075	117	Huruf latin u kecil
v	0076	118	Huruf latin v kecil
w	0077	119	Huruf latin w kecil
x	0078	120	Huruf latin x kecil
y	0079	121	Huruf latin y kecil
z	007A	122	Huruf latin z kecil
{	007B	123	Kurung kurawal buka
	007C	124	Garis vertikal (pipa)
}	007D	125	Kurung kurawal tutup
~	007E	126	Karakter gelombang (tilde)

DEL	007F	127	Delete
	0080	128	Dicadangkan
	0081	129	Dicadangkan
	0082	130	Dicadangkan
	0083	131	Dicadangkan
IND	0084	132	Index
NEL	0085	133	Next Lin
SSA	0086	134	Start of selected Area
ESA	0087	135	End of selected Area
	0088	136	Character Tabulation Set
	0089	137	Character tabulation with justification
	008A	138	Line tabulation set
PLD	008B	139	Partial line down
PLU	008C	140	Partial line up
	008D	141	Reverse Line feed
SS2	008E	142	Single shift two
SS3	008F	143	Single shift three
DCS	0090	144	Device control string
PU1	0091	145	Private use one
PU2	0092	146	Private use two
STS	0093	147	Set transmit state

CCH	0094	148	Cancel character
MW	0095	149	Message Waiting
	0096	150	Start of guarded area
	0097	151	End of guarded area
	0098	152	Start of String
	0099	153	Dicadangkan
	009A	154	Single Character Introducer
CSI	009B	155	Control sequence Introducer
ST	009C	156	String terminator
OSC	009D	157	Operating system command
PM	009E	158	Privacy Message
APC	009F	159	Application program command
	00A0	160	Spasi yang bukan pemisah kata
ı	00A1	161	Tanda seru terbalik
¢	00A2	162	Tanda sen (Cent)
£	00A3	163	Tanda Poundsterling
¤	00A4	164	Tanda mata uang (<i>Currency</i>)
¥	00A5	165	Tanda Yen
¦	00A6	166	Garis tegak putus-putus (<i>broken bar</i>)
§	00A7	167	Section sign
¨	00A8	168	Diaeresis
©	00A9	169	Tanda hak cipta (<i>Copyright</i>)

ª	00AA	170	Feminine ordinal indicator
«	00AB	171	Left-pointing double angle quotation mark
¬	00AC	172	Not sign
	00AD	173	Tanda strip (<i>hyphen</i>)
®	00AE	174	Tanda merk terdaftar
-	00AF	175	Macron
º	00BA	176	Tanda derajat
±	00B1	177	Tanda kurang lebih (<i>plus-minus</i>)
²	00B2	178	Tanda kuadrat (pangkat dua)
³	00B3	179	Tanda kubik (pangkat tiga)
´	00B4	180	Acute Accent
µ	00B5	181	Micro sign
¶	00B6	182	Pilcrow sign
·	00B7	183	Middle dot

1.8. *Quick Respond Code (QR-Code / Kode QR)*

1.8.1. Sejarah QR-Code

QR-Code berasal dari negeri Jepang dan hanya baru-baru mulai menjadi populer di Timur Tengah dan Eropa. Barcode yang kita lihat pada setiap produk komersial yang sangat bermanfaat sebagai kecepatan membaca merk, akurasi tertinggi dan fungsi mereka adalah kunci. Sebagai barcode mencapai puncaknya dan mulai

membaisakan ke seluruh dunia, kebutuhan untuk lebih banyak data dan jenis karakter yang akan disimpan tak terelak kan. Pengembang mulai mengikat untuk memperluas pada jumlah saat bar di barcode dan bagaimana posisi mereka berada untuk memungkinkan kapasitas data yang lebih lanjut. Kebutuhan barcode yang lebih kecil juga merupakan faktor lain yang menentukan dalam pembangunan QR-Code di masa mendatang. (Ida Bagus Putu Wirajaya Kusuma, 2012).

1.8.2. Pengertian QR-Code

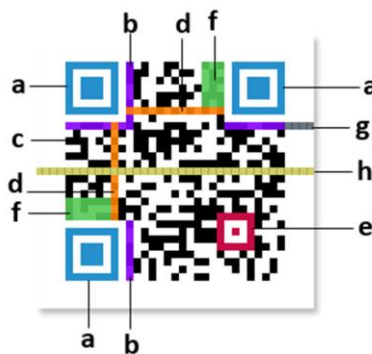
QR-Code atau Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh *Denso Wave*, sebuah divisi *Denso Corporation* yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai. QR merupakan singkatan dari *Quick Response* atau respon cepat yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang yang hanya menyimpan informasi secara horizaontal dan vertikal, oleh karena itu secara otomatis kode QR dapat menampung informasi lebih banyak daripada kode batang. (Ida Bagus Putu Wirajaya Kusuma, 2012).

Penggunaan kode QR sudah sangat lazim di Jepang. Hal ini dikarenakan kemampuannya menyimpan data yang lebih besar dari pada kode batang sehingga mampu mengkodekan informasi dalam

bahasa Jepang sebab dapat menampung huruf kanji. Kode QR telah mendapatkan standarisasi internasional dan standarisasi dari Jepang berupa ISO/IEC18004 dan JIS-X-0510 dan telah digunakan secara luas melalui ponsel di Jepang. (Ida Bagus Putu Wirajaya Kusuma, 2012).

1.8.3. Gambar QR-Code

Kode QR memiliki beberapa bagian yang masing-masing memiliki arti dan tugas yang berbeda. Untuk mengetahui arti dari setiap bagian pada QR Code dapat dilihat pada gambar di bawah ini :



Gambar 2.4 Bagian-bagian QR Code (Sumber : Moh. Lukman Sholeh dan Lutfi Ali Muharom, 2016)

Beberapa penjelasan anatomi *QR Code* Menurut Ariadi (2011) antara lain :

- a. *Finder Pattern* berfungsi untuk identifikasi letak *QR Code*.
- b. *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.
- c. *Data* berfungsi untuk menyimpan data yang dikodekan.

- d. *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat *QR Code*, berbentuk modul hitam putih.
- e. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan *QR Code* terutama distorsi non linier.
- f. *Version Information* adalah versi dari sebuah *QR Code*.
- g. *Quiet Zone* merupakan daerah kosong di bagian terluar *QR Code* yang mempermudah mengenali pengenalan *QR* oleh sensor *CCD*.
- h. *QR Code version* adalah versi dari *QR Code* yang digunakan.

1.8.4. Kelebihan QR-Code

QR-Code telah menjadi lebih populer daripada Barcode. Hal ini membuat penggunaan dan keragaman QR-Code jauh lebih menarik daripada bagian counter tua mereka, yaitu Barcode. Dan beberapa kelebihan QR-Code adalah sebagai berikut :

1. QR-Code dapat menyimpan hingga 7.089 karakter.
2. QR-Code memiliki kapasitas tinggi dalam data pengkodean, yaitu mampu menyimpan semua jenis data, seperti data numerik, data alfabitis, kanji, kana, hiragana, simbol dan kode biner.
3. QR-Code berupa kumpulan batang dengan rentang secara horizontal dan vertikal. Sehingga tidak membutuhkan space yang terlalu besar dalam pencetakan.


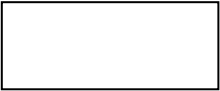
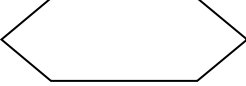

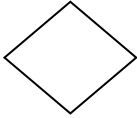
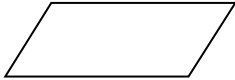
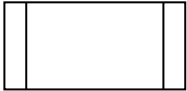
4. Secara statistik QR-Code mampu encoding jumlah data yang sama di sekitar ruang sepersepuluh dari barcode tradisional. Seperti mini QR-Code disebut *Micro QR-Codes*.
5. Secara fitur dari QR-Code kita tidak perlu untuk memindai mereka dari satu sudut tertentu. QR-Code mampu omnidirectional (360 derajat) dengan kecepatan tinggi membaca. QR-Code *scanner* mampu menentukan cara yang benar untuk memecahkan kode isi dalam kode QR karena tiga kotak khusus yang ditempatkan di sudut-sudut simbol.
6. QR-Code juga tahan terhadap kerusakan, sebab kode QR mampu memperbaiki kesalahan sampai dengan 30%. Oleh karena itu, walaupun sebagian simbol kode QR kotor ataupun rusak, data tetap dapat disimpan dan dibaca.

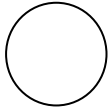
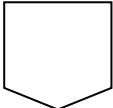
1.9. Flowchart

Menurut Sulindawati dan Muhammad Fathoni (2010) dalam jurnalnya yang berjudul “Pengantar Analisa Perancangan Sistem” mengatakan bahwa flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Menurut (Fairuz El Said, 2010), flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) didalam program atau prosedur system secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Tabel 2.2 Simbol Flowchart

Simbol	Dekripsi
	Terminator , simbol untuk menunjukkan awal atau akhir dari aliran proses
	Processing , menunjukkan pengolahan aritmatika dan pemindahan data
	Preparation , memberikan nilai awal pada suatu variabel atau counter
	Connector , tanda panah yang menunjukkan arah aliran dari satu proses ke proses yang lain
	Decision , menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak
	Input / Output , simbol untuk menunjukkan data yang menjadi input atau output proses.
	Sub-process , Simbol untuk menunjukkan bahwa dalam langkah yang dimaksudkan terdapat <i>flowchart</i> lain yang menggambarkan langkah

	tersebut lebih rinci.
	CConnector , penghubung pada halaman yang sama.
	Off page Connector , penghubung pada halaman yang berbeda.

1.10. *Unified Modelling Language (UML)*

UML adalah bahasa spesifikasi ystandar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untk mendukung pengemabngan sistem. (Rosa A.S & Shalahuddin M., 2013).

Dalam penggunaannya UML bergantung kepada level abstraksi penggunaannya. Jadi belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualisasikan.

Alat bantu yang dalam perancangan berorientasi objek berbasiskan UML menggunakan beberapa diagram berikut ini :

1. **Use Case Diagram**


Use case diagram menggambarkan interaksi antara sistem, sistem eksternal, dan pengguna. Dengan kata lain, secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna mengharapkan interaksi dengan sistem itu. Use case




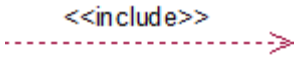
narrative digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

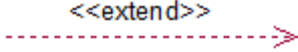
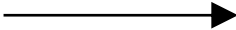
Use case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berak menggunakan fungsi-fungsi tersebut. (Rosa A.S. & Shalahuddin M., 2013).

Syarat penamaan pada use case adalah nama di definisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

Tabel 2.3 Simbol diagram Use case

Simbol	Deskripsi
Use Case  NewUseCase	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase atau use case.
Aktor / actor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah

 nama aktor	gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi / association tanpa panah 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor; gambar dengan garis tanpa panah mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
Asosiasi / association dengan panah 	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
Include 	Include merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case lain. Menggunakan include dalam relasi use case tambahan dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

<p>Ekstensi / extend</p> 	<p>Extend merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi. Relasi use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.</p>
<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>

Setiap use case dilengkapi dengan skenario. Skenario use case adalah alur jalannya proses use case dari sisi aktor dan sistem. Berikut adalah format tabel skenario use case.

Tabel 2.4 Skenario use case

Aksi Aktor	Reaksi Sistem
Skenario Normal	

Skenario Alternatif	


Skenario use case dibuat per use case terkecil, misalkan untuk generalisasi maka skenario yang dibuat adalah use case yang lebih khusus. Skenario normal adalah skenario bila sistem berjalan normal tanpa terjadi kesalahan atau *error*. Sedangkan skenario alternatif adalah skenario bila sistem tidak berjalan normal, atau mengalami *error*. Skenario normal dan skenario alternatif dapat lebih dari satu. Alur dari skenario inilah yang nantinya menjadi dasar pembuatan diagram sekuen.

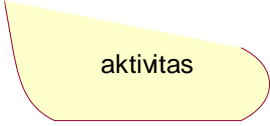
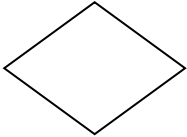
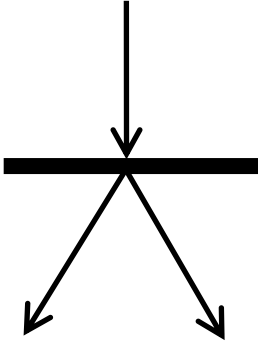
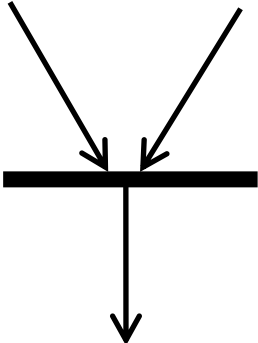
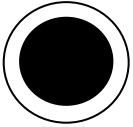
2. Activity Diagram

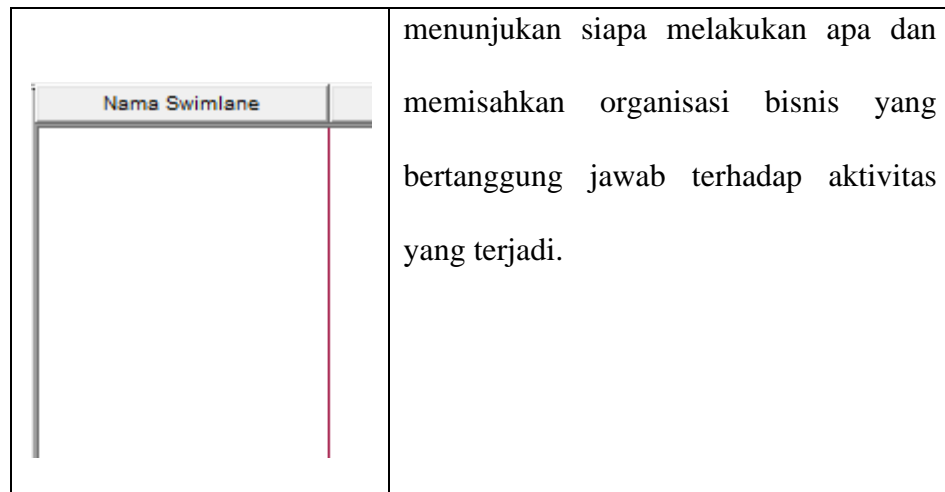
Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S & Shalahuddin M., 2013).

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.5 Simbol diagram aktivitas

Simbol	Deskripsi
Status awal / start point 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal.

<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Decision point</p> 	<p>Asosiasi percabangan diaman jika ada pilihan aktivitas lebih dari satu.</p>
<p>Fork / percabangan</p> 	<p>Fork (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
<p>Join / penggabungan</p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.</p>
<p>Status Akhir / End point</p> 	<p>Akhir dari suatu aktivitas</p>
<p>Swimlane</p>	<p>Pembagian aktivitas diagram untuk</p>



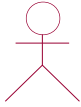
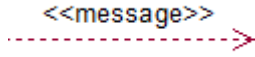
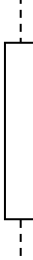

3. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sequence juga dibutuhkan untuk melihat skenario yang ada pada use case. (Rosa A.S & Shalahuddin M., 2013).

Banyaknya diagram sequence yang harus digambarkan adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sequence. Sehingga semakin banyak use case yang didefinisikan maka diagram sequence yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sequence :

Tabel 2.6 Simbol sequence diagram

Simbol	Dekripsi
<p data-bbox="448 459 528 488">Aktor</p>  <p data-bbox="501 660 635 689">nama aktor</p>	<p data-bbox="810 459 1369 929">Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="448 974 528 1003">Objek</p> <div data-bbox="448 1021 762 1115" style="border: 1px solid black; padding: 2px;"> <p data-bbox="467 1037 743 1066">nama objek : nama kelas</p> </div>	<p data-bbox="810 974 1369 1003">Menyatakan objek yang berinteraksi pesan</p>
<p data-bbox="448 1164 560 1193">Message</p> 	<p data-bbox="810 1164 1369 1265">Message, simbol mengirim pesan antar kelas</p>
<p data-bbox="448 1366 587 1395">Activation</p> 	<p data-bbox="810 1366 1369 1624">Activation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.</p>
<p data-bbox="448 1736 555 1765">Lifeline</p> 	<p data-bbox="810 1736 1369 1982">Lifeline, menyatakan kehidupan suatu objek. Garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.</p>

--	--

4. Class Diagram

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. (Rosa A.S & Shalahuddin M., 2013).

Class diagram secara khas meliputi : kelas (class), relasi, associations, generalization dan aggregation, atribut (attributes), operasi (operations/method) dan visibility.

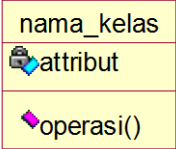


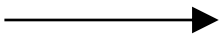
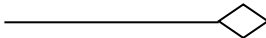
Hubungan antar kelas mempunyai keterangan yang disebut dengan multiplicity atau kardinaliti.

Tabel 2.7 Multiplicity dan Kardinaliti

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.8 Simbol diagram kelas

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Asosiasi / Association</p> 	Relasi antar kelas dengan makna umum asosiasi biasanya juga disertai dengan multiplicity
<p>Asosiasi berarah / Direct Association</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Agregasi / Agregation</p> 	Relasi antar kelas dengan makna semua- bagian (<i>whole part</i>).

1.11. Rational Rose

Menurut Adi Nugroho (2009) *Rational Rose* merupakan perangkat lunak pemodelan secara visual yang memiliki banyak kemampuan (power-full) untuk pembentukan sistem berorientasi objek yang menggunakan *Unified Modelling Language* (UML).

Istilah-istilah yang digunakan dalam rational rose, yaitu :

1. *View*, menunjukkan perbedaan dari berbagai aspek-aspek satu sistem yang dimodelkan.
2. Diagram, merupakan graph yang menjelaskan tentang isi dari sebuah *view*.
3. Elemen, elemen model konsep digunakan diagram merupakan elemen-elemen model yang menyatakan konsep-konsep berorientasi objek secara umum.

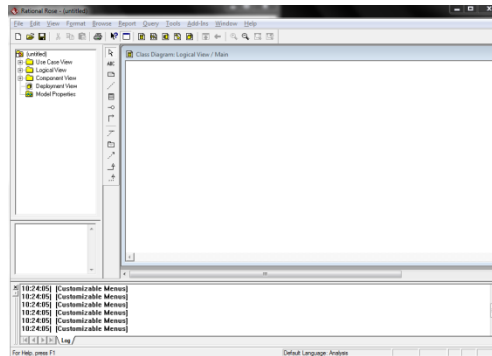
Berikut *interface* pada *Rational Rose* dapat kita lihat pada gambar :



Gambar 2.17 Tampilan Awal *Rational Rose* (Adi Nugroho. 2009)

Rational Unified Process (RUP) merupakan pilihan tepat untuk membuat suatu design yang lengkap dengan cara mudah karena model ini merupakan template, sehingga kita dapat langsung melakukan perubahan dengan mengganti *use case*, *actors*, *class diagram* yang telah ada pada template RUP.

Gambar di bawah ini merupakan tampilan utama dari *software Rational Rose* :



Gambar 2.14 Tampilan Software Rational Rose (Sukmawan Alfi Dwi, 2008)

1.12. MySQL

Menurut Bunafit Nugroho (2013:26), “MySQL adalah software atau program Database Server”. Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (*query*) dalam *database server* termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam software *database server* lain seperti SQL Server, Oracle, PostgreSQL dan lainnya.

SQL adalah bahasa standar yang digunakan untuk mengakses database server. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses database menjadi lebih *user-friendly* dibandingkan dengan menggunakan dBase atau Clipper yang masih menggunakan perintah-perintah pemrograman. (MADCOMS, 2009).

1.13. XAMPP

Menurut Bunafit Nugroho (2008 : 2), XAMPP adalah suatu bundel *web server* yang populer digunakan untuk coba-coba di *windows* karena

kemudahan instalasinya. Bundel program *open source* tersebut berisi antara lain server web Apache, interpreter PHP, dan basis data MySQL. Setelah menginstal XAMPP kita bisa memulai pemrograman PHP di komputer sendiri maupun mencoba menginstal aplikasi web.

1.14. Browser Web

Browser web adalah *software* yang digunakan untuk menampilkan informasi dari server web. *Software* ini kini telah dikembangkan dengan menggunakan *user interface* grafis, sehingga pemakai dengan melakukan *point* dan *click* dapat pindah antar dokumen. (-+Betha Sidik dan Husni I. Pohan, 2010:5).

1.15. PHP

PHP berasal dari kata “*Hypertext Preprocessor*”, yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP banyak dipakai untuk membuat program situs web dinamis.

PHP merupakan sekumpulan skrip atau bahasa program memiliki fungsi utama yaitu mampu mengumpulkan dan mengevaluasi hasil survei atau bentuk apapun ke *server database* dan pada tahap selanjutnya akan menciptakan efek beruntun. Efek beruntun PHP ini berupa tindakan dari skrip lain yang akan melakukan komunikasi dengan *database*, mengumpulkan dan mengelompokan informasi, kemudian menampilkannya

pada saat ada tamu website memerlukannya (menampilkan informasi sesuai permintaan user). (Mundzir M.F., 2014).

1.16. HTML

HTML (Hyper Text Mark Up Language) merupakan suatu metode-untuk mengimplementasikan konsep hypertext dalam suatu naskah atau dokumen. HTML sendiri bukan tergolong pada suatu bahasa pemrograman karena sifatnya yang hanya memberikan tanda (marking up) pada suatu naskah teks dan bukan sebagai program.

Dokumen HTML merupakan dokumen yang disajikan dalam browser web. Dokumen ini umumnya berisi informasi atau pun interface aplikasi di dalam internet. (Rabby Larry, 2012).

1.17. Javascript

Javascript adalah bahasa yang digunakan untuk membuat program agar dokumen HTML yang ditampilkan dalam browser menjadi lebih interaktif, tidak sekedar indah saja. Javascript memberikan beberapa fungsionalitas ke dalam halaman web, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antarmuka web.

Javascript merupakan bahasa skrip, bahasa yang tidak memerlukan kompiler untuk menjalankannya, cukup dengan interpreter. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan.

Javascript adalah bahasa pemrograman yang ringan dan mudah digunakan. Dengan adanya javascript ini maka halaman web dapat menjadi suatu program aplikasi dengan antarmuka web.

Javascript merupakan bahasa pemrograman yang tidak membutuhkan lisensi untuk dapat menggunakannya. Sehingga Javascript dapat langsung digunakan dalam pembuatan aplikasi berbasis web.

Umumnya program Javascript adalah program yang ditanamkan kedalam sebuah halaman web, sehingga halaman web menjadi sebuah aplikasi yang berjalan di dalam web browser. (Betha Sidik, 2011).

1.18. CSS

CSS (*Cascading Style Sheet*) adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokume yang ditulis dalam bahasa mirip *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML, termasuk SVG dan XUL. Spesifikasi CSS diatur oleh Wordl Wide Web Consortium (W3C). (Rabby Larry, 2012).

CSS digunakan untuk menentukan warna, jenis hurur, tata letak dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa *markup* lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini deapat meningkatkan aksesibilitas isi, memberikan lebih

banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi.

1.19. Pengujian Sistem

1.19.1. *Blackbox Testing*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut: fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan interface, (3) kesalahan dalam struktur data atau akses database eksternal, (4) kesalahan kinerja, (5) inisialisasi dan kesalahan terminasi (Roger S. Pressman, 2012).

Klasifikasi *Black box testing* mencakup beberapa pengujian, yang akan saya gunakan pada penelitian ini antara lain yaitu (Janner Simarmata, 2010) :

1. Pengujian usability (*usability testing*). Pengujian ini dilakukan jika antarmuka pengguna dari aplikasinya penting dan harus spesifik untuk jenis pengguna tertentu.
2. Pengujian domain (*domain testing*). Pengujian ini merupakan penjelasan yang paling sering menjelaskan teknik pengujian.

3. Pengujian skenario (*scenario testing*). Pengujian ini adalah pengujian yang realistis, kredibel dan memotivasi stakeholder, tantangan untuk program dan mempermudah penguji untuk melakukan evaluasi.
4. Penerimaan pengguna (*user acceptance*). Pada pengujian ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan.
5. Pengujian beta (*beta testing*). Pada pengujian ini, perangkat lunak didistribusikan sebagai sebuah versi beta dengan pengguna yang menguji aplikasi di situs mereka.

1.19.2. Whitebox Testing

“Pengujian kotak putih atau disebut juga pengujian kotak kaca (*glass box testing*) merupakan sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan perangkat komponen untuk menghasilkan *test case*”. (S. Pressman, 2012). Prinsip dari keluaran tipe testing ini adalah: Klasifikasi *white box testing* mencakup beberapa pengujian, yaitu:






1. Pengujian unit. Pengujian ini berada pada tingkat yang sangat dasar seperti ketika unit kode dikembangkan atau fungsi tertentu dibangun. Pengujian unit berkaitan dengan unit secara keseluruhan.

2. Analisis statis dan dinamis. Analisis statis dilibatkan melalui kode untuk mengetahui segala kemungkinan cacat dalam kode, sedangkan analisis dinamis akan melibatkan pelaksanaan kode dan penganalisisan hasilnya.
3. Cakupan pernyataan. Dalam hal ini jenis pengujian kode dijalankan dengan setiap pernyataan dari aplikasi yang dijalankan minimal sekali. Hal tersebut membantu dalam memastikan semua pernyataan untuk dijalankan tanpa efek samping.
4. Cakupan cabang. Pengujian ini membantu pemvalidasian semua cabang di dalam kode dan memastikan bahwa tidak ada yang mengarah ke percabangan perilaku abnormal dari aplikasi.
5. Pengujian mutasi. Pada pengujian ini aplikasi diuji untuk kode yang telah dimodifikasi setelah pemasangan bug/cacat tertentu. Hal ini juga membantu dalam menemukan kode dan strategi pengodean yang dapat membantu dalam mengembangkan fungsi secara efektif.

Teknik *white box* yang akan digunakan pada penelitian ini yaitu Pengujian basis path. Metode ini memungkinkan penguji dapat mengukur kompleksitas logis dari desain procedural dan menggunakannya sebagai pedoman untuk menerapkan himpunan basis dari semua jalur eksekusi.

1. Notasi Diagram Alir. Notasi yang digunakan untuk

menggambarakan jalur eksekusi adalah notasi diagram alir (atau grafik program), yang menggunakan notasi lingkaran (simpul atau node) dan anak panah (link atau edge).

Notasi	Arti
	Skema Sequence
	Skema If
	Skema While (...) DO (...)
	Skema Repeat (...) Until (...)
	Skema Case (...) Of

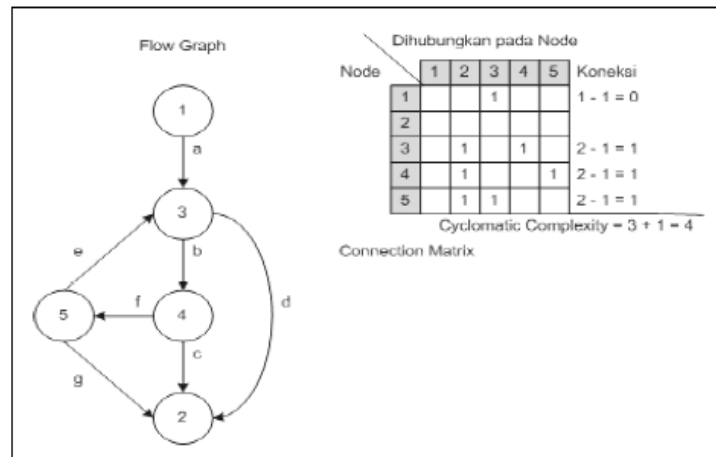
Gambar 2.6 Notasi Diagram Alir (Sumber : Janner

Simarmata, 2010)

2. Kompleksitas Siklomatis adalah metrics perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program, nilai yang didapat akan menentukan jumlah jalur independen dalam himpunan path, serta akan memberi nilai batas atas bagi jumlah pengujian yang harus dilakukan, untuk memastikan bahwa semua pernyataan telah dieksekusi sedikitnya satu kali.

3. Matriks Grafis (Graph Matrik). Bentuk struktur data yang sering digunakan untuk menggambarkan pengujian adalah dengan matriks grafis. Matriks grafis adalah matriks bujursangkar yang berukuran

sama dengan jumlah simpul pada grafik alir. Inputan dalam matriks harus bersesuaian dengan arah sisi dengan simpul.



Gambar 2.7 Matriks Grafis (Sumber : Janner Simarmata, 2010)