

BAB II

LANDASAN TEORI

2.1 Bahasa dan Aksara Jepang

Bahasa Jepang merupakan bahasa yang digunakan oleh kurang lebih 130.000.000 orang penduduk Jepang dan oleh imigran ataupun emigran negara tersebut. Bangsa Jepang diduga pertama kali memperoleh seni menulis pada tahun 405 AD (*Anno Domini*, masehi). ketika seorang cendekiawan Korea datang ke Jepang membawa buku-buku berbahasa China dan mengajarkan Bangsa Jepang cara membaca dan menulis aksara China (Suski, 2011, p. 1).

Penulisan aksara Jepang memang didasari oleh aksara China, namun struktur tata bahasa kedua bahasa ini tidak sama. Dalam menuliskan aksara-aksara Jepang, urutan dan arah goresan pena sangat berpengaruh. Bahasa Jepang dituliskan dengan menggunakan kombinasi tiga buah aksara, yaitu aksara Hiragana, Katakana, dan Kanji.

2.1.1 Hiragana

Hiragana (平仮名/ひらがな) adalah salah satu dari dua aksara Kana pada aksara Jepang yang merupakan hasil penyederhanaan dari bentuk kursif aksara China (Kanji). Hiragana merupakan aksara silabis, setiap aksara melambangkan satu bunyi tanpa arti. Hiragana dasar terdiri dari 46 aksara dimana masing-masing aksara mewakili bunyi yang ada dalam Bahasa Jepang sehingga pada dasarnya semua kosakata dalam bahasa Jepang dapat ditulis sepenuhnya menggunakan aksara Hiragana. Fungsi dari aksara Hiragana adalah:

1. Sebagai partikel penanda yang berhubungan dengan tata bahasa Jepang.

Contoh: カレーを食べた。 (*Karē wo tabeta*). Aksara Hiragana 「を」 (wo)

pada kalimat tersebut berfungsi sebagai penanda objek (カレー *karē*) dari verba (食べた *tabeta*).

2. Ditulis disamping aksara Kanji kata kerja atau kata sifat untuk melengkapi bacaannya yang berguna sebagai konjugasi atau infleksi. Aksara Hiragana dalam kasus ini disebut dengan *okurigana* (送り仮名) yang secara harfiah berarti "aksara pendamping". Contoh: 行く (*iku*), yang berarti "pergi". Aksara Hiragana 「く」 (ku) pada kata tersebut adalah *okurigana*.

3. Ditulis di atas aksara Kanji yang berguna sebagai panduan membaca untuk aksara Kanji yang jarang dipakai, cara membacanya tidak beraturan, atau untuk anak kecil yang masih belum mempelajari aksara Kanji tersebut. Aksara Hiragana dalam kasus ini disebut dengan *furigana* (振り仮名).

Contoh: 日本語。 Aksara Hiragana pada kata tersebut (にほんご *nihongo*)

yang diletakan di atas aksara Kanji (日本語 *nihongo*) adalah *furigana* yang berfungsi sebagai pemandu fonetis dalam membaca aksara Kanji tersebut.

4. Digunakan untuk menulis kosakata yang aksara Kanjinya tidak umum atau jarang dipakai, atau aksara Kanjinya terlalu susah untuk ditulis. Contoh: cantik dalam bahasa Jepang adalah *kirei*, bila ditulis dengan aksara Kanji akan menjadi 綺麗. Aksara tersebut susah ditulis dan tidak umum sehingga biasanya ditulis hanya menggunakan aksara Hiragana menjadi きれい.

5. Sebagai aksara pertama yang dipelajari oleh pemula atau anak-anak sebelum mempelajari aksara Kanji.

Tabel 2.1 Hiragana Dasar

	a	i	u	e	o
	あ	い	う	え	お
k	か	き	く	け	こ
s	さ	し	す	せ	そ
t	た	ち	つ	て	と
n	な	に	ぬ	ね	の
h	は	ひ	ふ	へ	ほ
m	ま	み	む	め	も
y	や		ゆ		よ
r	ら	り	る	れ	ろ
w	わ				を

Ke-46 aksara Hiragana dasar belum mewakili keseluruhan bunyi yang ada pada bahasa Jepang. Masih ada lima buah konsonan yaitu g, z, d, b, dan p yang dibentuk dengan cara menambahkan *dakuten* (濁点) yang berbentuk dua buah garis seperti tanda petik dua (̄) dan *handakuten* (半濁点) yang berbentuk lingkaran kecil (̄) di atas aksara dasar konsonan k, s, t, dan h. Khusus untuk *handakuten* hanya dapat ditambahkan pada aksara konsonan h untuk membentuk bunyi konsonan p.

Tabel 2.2 Hiragana dengan *Dakutan* dan *Handakuten*

	a	i	u	e	o
k→g	が	ぎ	ぐ	げ	ご
s→z	ざ	じ	ず	ぜ	ぞ
t→d	だ	ぢ	づ	で	ど
h→b	ば	び	ぶ	べ	ぼ
h→p	ぱ	ぴ	ふ	ペ	ぽ

Selain konsonan yang telah dijelaskan, Hiragana dasar dapat digabungkan dengan aksara ya (や), yu (ゆ), dan yo (よ) kecil untuk membentuk konsonan dengan bunyi seperti kya, kyu, kyo, dan lainnya.

Tabel 2.3 Hiragana dengan Kombinasi Ya, Yu, dan Yo

	k	s	ch	n	h	m	r	g	j	b	p
ya	きや sha	しや cha	ちや cha	にや	ひや	みや	りや	ぎや ja	じや ja	びや ju	ぴや ju
yu	きゅ shu	しゅ chu	ちゅ chu	にゅ	ひゅ	みゅ	りゅ	ぎゅ ju	じゅ ju	びゅ ju	ぴゅ ju
yo	きょ sho	しょ cho	ちょ cho	にょ	ひょ	みょ	りょ	ぎょ jo	じょ jo	びょ jo	ぴょ jo

Jika terdapat tsu (つ) kecil diantara aksara Hiragana, maka aksara yang terletak setelah つ kecil tersebut harus diucapkan dengan hitungan dua suku kata. Pengucapan seperti ini disebut dengan *sokuon* (そくおん) atau suara konsonan kembar. Contoh, まつて akan menjadi *matte* (dibaca mat-te).

Perpanjangan bunyi vokal atau yang disebut dengan *chou'on* (ちゅうおん) suatu aksara Hiragana pada Bahasa Jepang dilakukan dengan menambahkan

aksara Hiragana a (あ), i (い), dan u (う). Perpanjangan bunyi ini sangat penting, karena kata yang memiliki perpanjangan bunyi (*chou'on*) dan yang tidak memiliki perpanjangan bunyi (*seion* atau vokal pendek) memiliki dua arti yang berbeda. Contoh, おばさん(*obasan*) memiliki arti tante sedangkan おばあさん (*obaasan*) memiliki arti nenek. Selain memperpanjang bunyi i, aksara Hiragana i (い) juga dipergunakan untuk memperpanjang bunyi e. Aksara Hiragana u (う) juga dipergunakan untuk memperpanjang bunyi o selain untuk bunyi u. Terkadang bunyi e dan o juga diperpanjang juga dengan menggunakan aksara Hiragana mereka sendiri, seperti pada おねえさん (*oneesan*/kakak perempuan) dan おおきい (ooiki/besar), namun penggunaan seperti ini tidak banyak (Kim, 2012).

2.1.2 Katakana

Katakana (片仮名/カタカナ) merupakan salah satu dari dua aksara Kana pada aksara Jepang selain Hiragana. Aksara Katakana berasal dari fragmentasi aksara-aksara Kanji. Sama seperti dengan aksara Hiragana, aksara Katakana juga merupakan aksara silabis, setiap aksara melambangkan satu bunyi tanpa arti.

Fungsi dari aksara Katakana adalah:

1. Untuk menulis kata serapan/kata pinjaman dari bahasa asing yang disebut dengan *gairaigo* (外来語). Contoh: アイスクリーム (*aisukurīmu*), yang diserap dari bahasa Inggris “Ice Cream” yang berarti “es krim”.
2. Untuk menuliskan kata-kata atau nama-nama asing, seperti nama negara atau nama orang. Contoh: インドネシア (*indonesia*) yang berarti “Indonesia”, ダニエル・ラドクリフ (*danieru radokurifu*) yang berarti “Daniel Radcliffe”.

3. Sebagai penekanan dan penegasan suatu kata, fungsinya sama seperti kata yang digaris bawah, dimiringkan, ditebalkan, atau dibesarkan dalam bahasa Indonesia.
4. Untuk menuliskan onomatope, yaitu kata atau sekelompok kata yang menirukan bunyi-bunyi dari sumber yang digambarkannya. Contoh: suara mengetuk pintu ドンドン (*dondon*), suara hentakan kaki ドスンドスン (*dosundosun*), suara kucing ニヤー (*nyaa*), dan lainnya.
5. Untuk menuliskan nama produk, perusahaan, atau hal komersial lainnya.

Contoh: トヨタ (Toyota).

6. Untuk menuliskan nama klasifikasi tanaman dan hewan. Contoh: バラ (*bara*) yang berarti bunga mawar.

Tabel 2.4 Katakana Dasar

a	i	u	e	o
ア	イ	ウ	エ	オ
k	カ	キ	ク	ケ
コ				
s	サ	シ	ス	セ
ソ				
t	タ	チ	ツ	テ
ト	chi	tsu		
n	ナ	ニ	ヌ	ネ
ノ				
h	ハ	ヒ	フ	ヘ
ホ		fu		
m	マ	ミ	ム	メ
モ				
y	ヤ		ユ	ヨ
r	ラ	リ	ル	レ
ロ				



Sama seperti Hiragana, Katakana dasar dapat ditambahkan dengan *dakuten* (濁点) dan *handakuten* (半濁点) untuk membentuk konsonan g, z, d, b, dan p dari aksara dasar konsonan k, s, t, dan h.

Tabel 2.5 Katakana dengan *Dakuten* dan *Handakuten*

The diagram features a central point with a flower-like shape extending outwards. Each petal contains a Katakana character and its Romanized form. The petals are arranged in a circular pattern.

	a	i	u	e	o
k→g	ガ	ギ	グ	ゲ	ゴ
s→z	ザ	ジ	ズ	ゼ	ゾ
t→d	ダ	ヂ	ヅ	ヂ	ド
h→b	バ	ビ	ブ	ベ	ボ
h→p	パ	ピ	プ	ペ	ポ

Selain itu aksara Katakana dasar juga dapat dikombinasikan dengan ya (ヤ), yu (ュ), dan yo (ョ) kecil untuk membentuk aksara dengan bunyi seperti kya, kyu, dan kyo sama seperti kombinasi aksara Hiragana.

Tabel 2.6 Katakana dengan Kombinasi Ya, Yu, dan Yo

	k	s	ch	n	h	m	r	g	j	b	p
ya	キヤ	シヤ	チヤ	ニヤ	ヒヤ	ミヤ	リヤ	ギヤ	ジャ	ビヤ	ピヤ
	sha	cha							ja		
yu	キュ	シュ	チュ	ニュ	ヒュ	ミュ	リュ	ギュ	ジュ	ビュ	ピュ
	shu	chu							ju		
yo	キョ	ショ	チョ	ニョ	ヒョ	ミョ	リョ	ギョ	ジョ	ビョ	ピョ
	sho	cho							jo		

2.1.3 Kanji

Aksara Kanji (漢字) merupakan aksara China yang dipergunakan dalam penulisan bahasa Jepang. Aksara Kanji merupakan aksara morfemis, setiap aksara melambangkan suatu arti (ide) dan beberapa bacaan. Beberapa aksara Kanji memiliki arti dan cara baca yang berbeda dengan aksara China (Hanzi tradisional: 漢字; Hanzi sederhana: 汉字; Pinyin: *hanzi*) yang memiliki bentuk yang sama dengan aksara Kanji tersebut. Cara baca aksara Kanji dengan cara baca China disebut dengan *on'yomi* (音読み) sedangkan cara baca Jepang disebut *kun'yomi* (訓読み). Aksara Kanji yang penulisannya digabungkan dengan aksara Kanji lain akan dibaca dengan cara *on'yomi*, sementara apabila aksara Kanji yang berdiri sendiri atau yang diikuti oleh *okurigana* (aksara Hiragana yang mengikuti aksara Kanji) akan dibaca dengan cara *kun'yomi* (Kim, 2012, p. 22).

Contoh, 力 akan dibaca *chikara* (cara *kun'yomi*) apabila berdiri sendiri atau bila diikuti dengan *okurigana* seperti 力ずくで (*chikarazuku de*), namun apabila digabungkan dengan aksara Kanji lain seperti 腕力 (*wanryoku*) maka aksara Kanji 力 akan dibaca *ryoku* (cara *on'yomi*).

Terdapat sekitar 40.000 aksara Kanji, namun hanya sekitar 2.000 aksara Kanji yang benar-benar dipergunakan dalam penggunaan Bahasa Jepang sehari-hari (Kim, 2012, p. 12). Aksara Kanji digunakan untuk membedakan homofon, dimana terdapat banyak kata yang memiliki pengucapan yang sama (sehingga bila ditulis menggunakan aksara silabis Hiragana atau Katakana akan menjadi sama) namun memiliki arti yang berbeda (Kim, 2012, p. 12).

Berikut adalah daftar 50 aksara Kanji yang terdapat di dalam aplikasi yang dibuat:

Tabel 2.7 Lima Puluh Aksara Kanji Dasar

#	Kanji	Strokes	Arti	On	Kun
1	一	1	satu	ichi, itsu	hito-tsu
2	二	2	dua	ni, ji	futa-tsu
3	三	3	tiga	san	mit-tsu
4	四	5	empat	shi	yot-tsu, yon
5	五	4	lima	go	itsu-tsu
6	六	4	enam	roku	mut-tsu
7	七	2	tujuh	shichi	nana-tsu
8	八	2	delapan	hachi	yat-tsu
9	九	2	sembilan	ku or kyū	kokono-tsu
10	十	2	sepuluh	jū	tō
11	百	6	seratus	hyaku	momo
12	千	3	seribu	sen	chi
13	上	3	atas	jō	ue
14	下	3	bawah	ka, ge	shita, shimo, moto
15	左	5	kiri	sa	hidari
16	右	5	kanan	u, yū	migi
17	中	4	di dalam, di tengah	chū, jū	naka
18	大	3	besar	dai, tai	ō-kii
19	小	3	kecil	shō	chii-sai, ko, o
20	月	4	bulan	gatsu or getsu	tsuki
21	日	4	hari, matahari	nichi, jitsu	hi, ka
22	年	6	tahun	nen	toshi
23	早	6	awal	sō, satsu	haya-i
24	木	4	pohon	moku or boku	ki
25	林	8	hutan	rin	hayashi
26	山	3	gunung	san, zan	yama
27	川	3	sungai	sen	kawa
28	土	3	tanah	do, to	tsuchi
29	空	8	langit	kū	sora, a-ku, kara
30	田	5	sawah	den	da, ta
31	天	4	surga	ten	ame, ama
32	生	5	kehidupan	sei, shō	i-kiru, u-mu, nama
33	花	7	bunga	ka	hana
34	草	9	rumput	sō	kusa
35	虫	6	serangga	chū	mushi

#	Kanji	Strokes	Arti	On	Kun
36	犬	4	anjing	ken	inu
37	人	2	orang	jin, nin	hito
38	名	6	nama	mei or myō	na
39	女	3	perempuan	jo, nyō	on'na
40	男	7	laki-laki	dan or nan	otoko
41	子	3	anak	shi, su	ko
42	目	5	mata	moku	me
43	耳	6	telinga	ji, ni	mimi
44	口	3	mulut	kou	kuchi
45	手	4	tangan	shu	te
46	足	7	kaki	soku	ashi
47	見	7	melihat	ken, gen	mi-ru
48	音	9	suara	on	ne, oto
49	力	2	tenaga	riki, ryoku	chikara
50	気	6	udara	ki, ke	iki

2.2 Model Pembelajaran Interaktif

Model pembelajaran Interaktif adalah suatu cara atau teknik pembelajaran yang digunakan guru pada saat menyajikan bahan pelajaran dimana guru pemeran utama dalam menciptakan situasi interaktif yang edukatif, yakni interaksi antara guru dengan siswa, siswa dengan siswa dan dengan sumber pembelajaran dalam menunjang tercapainya tujuan belajar.

Syarat model pembelajaran interaktif (Ahmad Sabari, 2005) :

1. Dapat membangkitkan motivasi, minat atau gairah belajar siswa.
2. Dapat merangsang keinginan siswa untuk belajar lebih lanjut.
3. Dapat memberikan kesempatan bagi siswa untuk memberikan tanggapannya terhadap materi yang disampaikan.
4. Dapat menjamin perkembangan kegiatan kepribadian siswa.
5. Dapat mendidik siswa dalam teknik belajar sendiri dan cara memperoleh pengetahuan melalui usaha pribadi.

6. Dapat menanamkan nilai-nilai dan sikap siswa dalam kehidupan sehari-hari.

Macam-macam model pembelajaran interaktif:

1. *Picture and picture.*
2. *Numbered head together.*
3. *Student teams-achievement divisons (STAD).*
4. *Jigsaw.*
5. Mencari pasangan.
6. *Think pair and share.*

2.3 Android

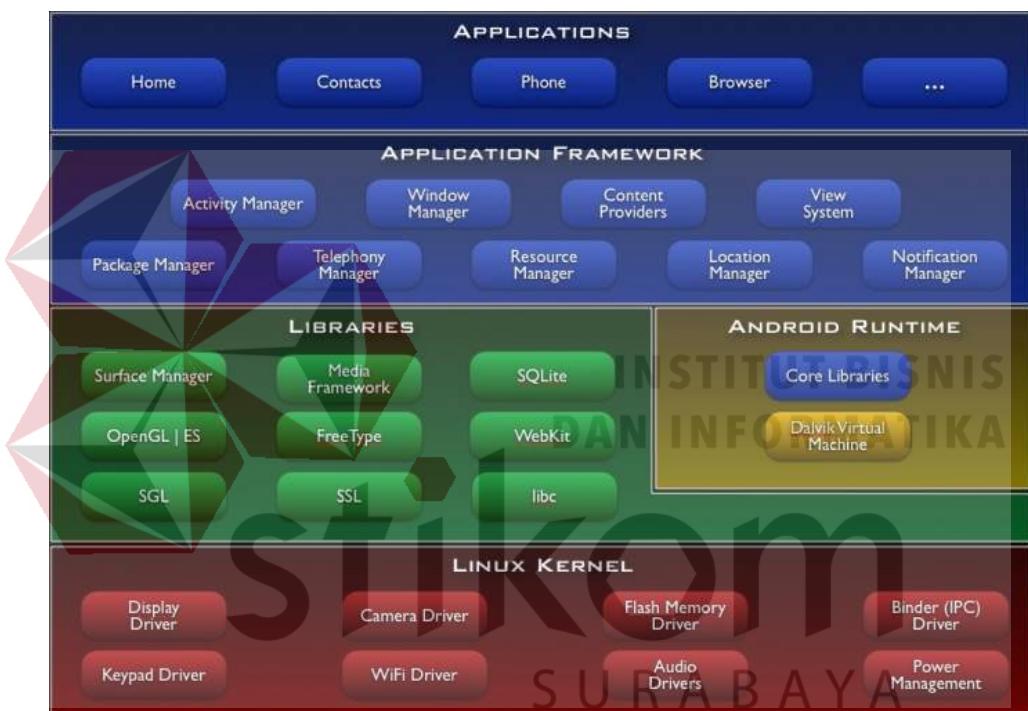
Android dirilis pada tahun 2007 saat Google di bawah aliansi Handset. Dengan dirilisnya Android, Google menyediakan berbagai *development kit* dan tutorial untuk membantu calon pengembang ke sistem baru. Berkas-berkas bantuan yaitu *software development kit* (SDK), dan bahkan komunitas pengembang dapat ditemukan di website Google Android, <http://code.google.com/android>. (DiMarzio, 2008:5)

Android adalah sebuah sistem operasi berbasiskan java, yang berjalan pada linux 2.6 kernel (DiMarzio, 2008:7). Sistem operasi Android untuk peralatan *mobile* dibuat dapat digunakan pada *open source software* dibawah lisensi Apache 2008. Android memuat komponen dalam (*platform*) dan komponen luar (Linux kernel dan WebKit, dibawah lisensi GPLv2 dan LGPL serta variasi komponen lainnya atau projek, hak cipta oleh para pemilik lainnya) (Blackduck, 2012:2).

Gambar 1 menunjukkan arsitektur Android.

Android tidak hanya sebuah sistem operasi tetapi juga termasuk sebuah *middleware* dan sebuah *array of application* untuk pengguna. Beberapa dari fitur

pendukung adalah Dalvik Virtual Machine, membangun di *browser* dan mendukung database, media, kamera, GPS, map, dan fitur lainnya (Dhruv Mohindra, 2008:3). Sistem ini sangat ringan dan memiliki banyak fitur, antara lain *speech input*, *Text-to-Speech* (TTS), *accelerated 3D graphics engine* (berdasarkan pada dukungan hardware), mendukung *database* yang didukung oleh SQLite, dan sebuah *web browser* yang terintegrasi. (DiMarzio, 2008:7)



Gambar 2.1 Arsitektur Android

Menurut Android SDK Docs (2012), ketika seseorang mengembangkan aplikasi di Android, diharapkan memahami *platform* pada umumnya untuk manajemen perubahan API (*application programming interface*). Tingkat API diharapkan agar dikenali terlebih dahulu untuk memastikan kompatibilitas aplikasi dengan perangkat yang mungkin akan diinstal. Tingkat API adalah nilai integer yang secara unik mengidentifikasi kerangka revisi API yang ditawarkan

oleh versi dari *platform* Android. Pembaruan kerangka API dirancang sehingga API baru tetap kompatibel dengan versi API sebelumnya, kecuali dalam kasus-kasus terisolasi dimana aplikasi menggunakan bagian dari API yang kemudian dihapus karena alasan tertentu. Setiap versi *platform* Android mendukung satu tingkat API.

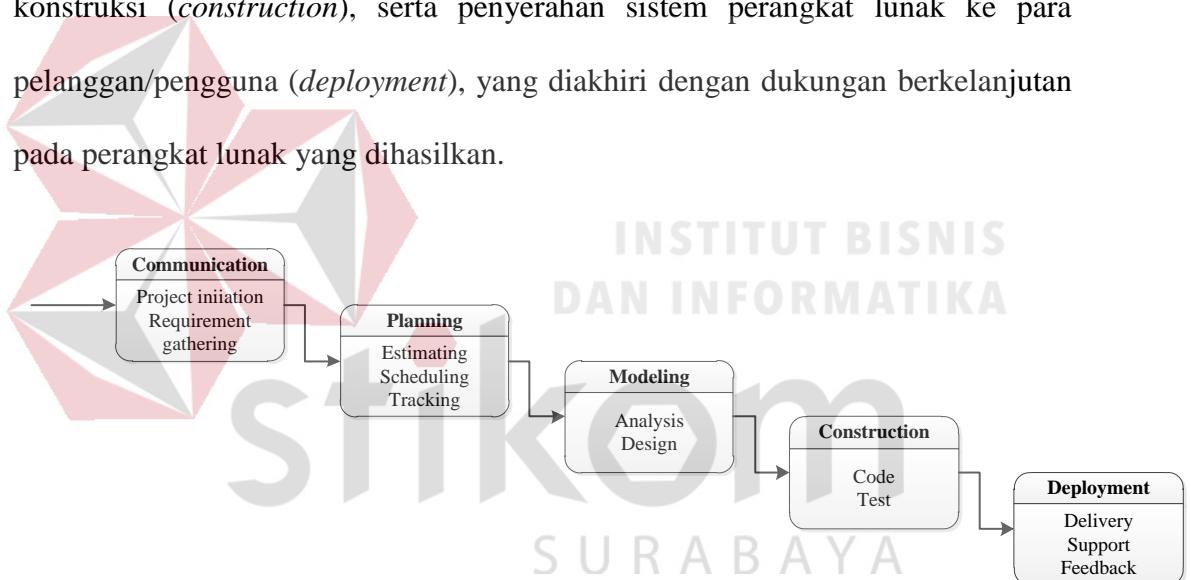
2.4 UML

Menurut Nugroho (2005,16), pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku komunikasi yang baik akan terjalin dengan mudah antaranggota tim pengembang sistem/perangkat lunak dan antara anggota tim pengembang dengan para pengguna (*end-user*). Untuk melakukan pemodelan sistem/perangkat lunak, dalam buku ini akan menggunakan notasi-notasi UML (*Unified Modeling Language*) yang akan digambarkan secara elektronik (dengan bantuan komputer) lewat sarana perangkat lunak. Dengan pemodelan menggunakan UML ini, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaahan bagaimana objek-objek dalam sistem saling mengirim pesan (*message*) dan saling bekerjasama satu sama lain.
3. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
4. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

2.5 System Development Life Cycle

Menurut Pressman (2015), *System Development Life Cycle* (SDLC) ini biasanya disebut juga dengan model *waterfall*. Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak. Pengembangan perangkat lunak dimulai dari spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan.



Gambar 2.2 Model Pengembangan *Waterfall* (Pressman, 2015)

Gambar 2.2 menunjukkan tahapan umum dari model proses *waterfall*. Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2015) memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai

dari level kebutuhan sistem lalu menuju ke tahap *Communication, Planning, Modeling, Construction, dan Deployment*.

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam Model *Waterfall* menurut Pressman (2015):

a. *Communication*

Langkah pertama diawali dengan komunikasi kepada konsumen/pengguna.

Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. *Planning*

Setelah proses *communication* ini, kemudian menetapkan rencana untuk penggerjaan *software* yang meliputi tugas-tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal penggerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

d. *Construction*

Construction merupakan proses membuat kode (*code generation*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan

dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

