

## BAB 2

### LANDASAN TEORI

#### 2.1. Bahasa Jepang

Bahasa Jepang memiliki tiga jenis huruf, pertama huruf *hiragana*, *hiragana* berasal dari bahasa Jepang asli, dan dipakai untuk menyederhanakan huruf *kanji*. Huruf *hiragana* terbentuk dari garis - garis yang melengkung dan tidak bersudut tajam, dan aksara hiragana ini dikenal sebagai *onna de* atau “Tulisan Wanita” karena biasa digunakan untuk kaum wanita. Aksara hiragana ini biasa digunakan untuk menulis kata kerja, kata sifat, kata bantu, dan sebagai awalan atau akhiran sebagai penghalus kata . Bentuk-bentuk aksara Hiragana dapat dilihat pada tabel 2.1. [3][14].

Tabel 2. 1 Daftar Aksara Hiragana

Hiragana	Caba Baca	Hiragana	Cara Baca	Hiragana	Cara Baca
あ	a	ら	ra	きゃ	kya
い	i	り	ri	きゅ	kyu
う	u	る	ru	きょ	kyo
え	e	れ	re	しゃ	sha
お	o	ろ	ro	しゅ	shu
か	ka	わ	wa	しよ	sho
き	ki	ゐ	wi	ちゃ	cha
く	ku	ゑ	we	ちゅ	chu
け	ke	を	wo	ちよ	cho
こ	ko	が	ga	にゃ	nya

さ	sa	ぎ	gi	にゆ	nyu
し	shi	ぐ	gu	によ	nyo

す	su	げ	ge	ひゃ	hya
せ	se	ご	go	ひゅ	hyu
そ	so	ざ	za	ひい	hyo
た	ta	じ	ji	みゃ	mya
ち	chi	ず	zu	みゅ	myu
つ	tsu	ぜ	ze	みよ	myo
て	te	ぞ	zo	りゃ	rya
と	to	だ	da	りゅ	ryu
な	na	ぢ	di	りよ	ryo
に	ni	づ	du	ぎゃ	gya
ぬ	nu	で	de	ぎゅ	gyu
ね	ne	ど	do	ぎよ	gyo
の	no	ば	ba	じゃ	ja
は	ha	び	bi	じゅ	ju
ひ	hi	ぶ	bu	じよ	jo
ふ	fu	べ	be	ぢゃ	(ja)
へ	he	ぼ	bo	ぢゅ	(ju)
ほ	ho	ぱ	pa	ぢよ	(jo)
ま	ma	ぴ	pi	びゃ	bya
み	mi	ぷ	pu	びゅ	byu

む	mu	ペ	pe	びよ	byo
め	me	ぽ	po	びゃ	pya
も	mo	ん	n	びゅ	pyu
や	ya			びょ	pyo

ゆ	yu				
よ	yo				

Kedua, huruf *katakana*, berkebalikan dengan huruf hiragana yang cenderung tidak memiliki sudut dan garis melengkung, huruf *katakana* cenderung memiliki garis yang kaku dan bersudut tajam, sehingga dikatakan sebagai huruf laki-laki. Selain berfungsi menuliskan kata serapan dari bahasa asing, huruf *katakana* juga berfungsi menuliskan nama binatang, nama orang asing, nama tumbuhan, nama perusahaan / *brand*, dan penekanan dalam kalimat bahasa Jepang itu sendiri. Bentuk - bentuk huruf *katakana* dapat dilihat pada tabel 2.2. [3][14].

Tabel 2. 2 Daftar Aksara Katakana

Katakan	Caba Baca	Katakana	Cara Baca	Katakana	Cara Baca
ア	a	マ	ma	キャ	kya
イ	i	ミ	mi	キユ	kyu
ウ	u	ム	mu	キヨ	kyo
エ	e	メ	me	シャ	sha
オ	o	モ	mo	シュ	shu
カ	ka	ヤ	ya	シヨ	sho
キ	ki	ユ	yu	チャ	cha
ク	ku	ヨ	yo	チュ	chu
ケ	ke	ラ	ra	チヨ	cho
コ	ko	リ	ri	ニヤ	nya
サ	sa	ル	ru	ニユ	nyu

シ	shi	レ	re	ニヨ	nyo
ス	su	ロ	ro	ヒヤ	hya

セ	se	ワ	wa	ヒユ	hyu
ソ	so	ヰ	wi	ヒヨ	hyo
タ	ta	ヱ	we	ミヤ	mya
チ	chi	ヲ	wo	ミュ	myu
ツ	tsu	ソ	n	ミヨ	myo
テ	te	ガ	ga	リヤ	rya
ト	to	ギ	gi	リュ	ryu
ナ	na	グ	gu	リヨ	ryo
ニ	ni	ゲ	ge	ギヤ	gya
ヌ	nu	ゴ	go	ギユ	gyu
ネ	ne	ザ	za	ギョ	gyo
ノ	no	ジ	ji	ジャ	ja
ハ	ha	ズ	zu	ジュ	ju
ヒ	hi	ゼ	ze	ジョ	jo
フ	fu	ゾ	zo	ビヤ	bya
ヘ	he	ダ	da	ビユ	byu
ホ	ho	ヂ	ji	ビヨ	byo
バ	ba	ヅ	zu	ピヤ	pya
ビ	bi	デ	de	ピユ	pyu
ブ	bu	ド	do	ピョ	pyo

べ	be	ピ	pi	ポ	po
ボ	bo	ブ	pu		
パ	pa	ペ	pe		

Ketiga adalah *kanji*, berbeda dengan *hiragana* dan *katakana* yang mewakili penyebutan satu suku kata, *kanji* mewakili sebuah makna dalam ujaran. Huruf *kanji* umum dipergunakan masyarakat Jepang, baik itu dibidang akademik maupun non-akademik. Huruf *kanji* berjumlah 1945 bentuk berbeda (tahun 1981), hal inilah yang menyebabkan huruf kanji disebut sebagai huruf yang sangat rumit dan sukar untuk dipelajari. Huruf *kanji* bukanlah huruf asli dari Jepang, melainkan hasil pengembangan dan modifikasi dari huruf China. Contoh huruf *kanji* bisa dilihat pada tabel 2.3. [3][14].

Tabel 2. 3 Daftar Akasara Kanji

Kanji	Kunyomi	Onyomi	Arti
薬	Kusuri	Yaku	Obat
木	Ki	Moku	Pohon
日	Hi	Ni	Matahari, Hari
林	Hayashi	Rin	Hutan
愛	-	Ai	Cinta
桜	Sakura	Ou, Yo	Bunga Sakura

Furigana adalah sistem penulisan untuk membantu cara baca sebuah kanji [15]. Biasanya menggunakan gabungan huruf kanji dengan huruf *hiragana*, bisa juga gabungan huruf latin dan huruf kanji. Contoh huruf kanji bisa dilihat pada tabel 2.4. Contoh Furigana.

Tabel 2. 4 Contoh Furigana

Kanji	Hiragana	Romanji	Arti
日本	にほん 日本	Nihon 日本	Jepang

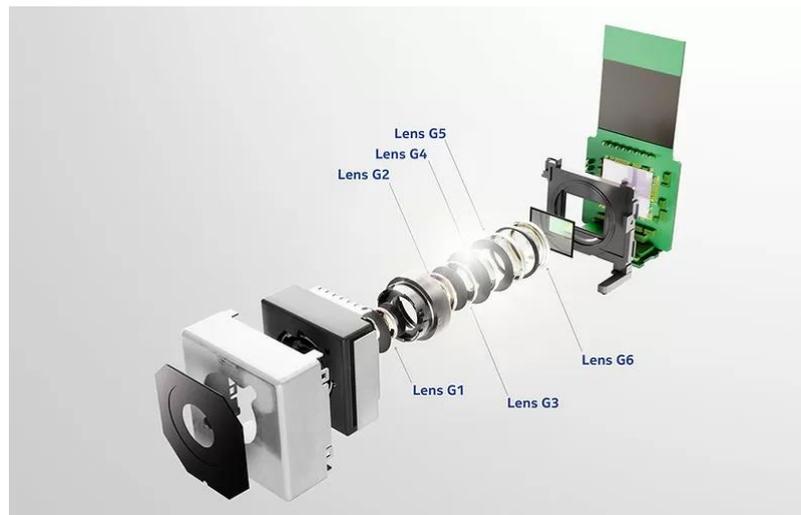
### 2.1. Penerjemah

Penerjemah adalah sebuah proses alih bahasa dari bahasa sumber ke dalam bahasa sasaran dengan isi dan gaya bahasa yang sama atau harus ada unsur padanan (*equivalence*). Untuk menghasilkan suatu terjemahan yang baik adalah dengan memperhatikan berbagai hal agar hasil terjemahan tersebut bisa diterima dan tidak merubah pesan serta makna yang dimaksud oleh bahasa sumber atau bahasa asal. Pada dasarnya penerjemahan merupakan cara untuk mencari padanan kata atau kalimat yang akan diterjemahkan [16].

### 2.2. Kamera Mobile

Kamera adalah salah satu komponen penting pada smartphone masa kini. Kinerja hardware yang disematkan pada smartphone semakin hari semakin baik dengan kualitas citra yang mampu memanjakan penggunanya. Hasil foto pada smartphone selain mudah diingat, terkadang memiliki peran penting dan selalu menarik untuk dilihat dan diamati karena memiliki nilai dokumentasi yang tinggi.

Kamera memiliki fungsi untuk menangkap citra dari apa yang terlihat di depannya, kemudian di cetak ke dalam lembaran kertas. Tapi pada smartphone kertas tersebut digantikan dengan sensor kamera digital yang mirip seperti panel surya. Sensor tersebut terdiri dari jutaan pixel merah, biru, dan hijau atau dikenal sebagai megapixel. Ketika cahaya mengenai pixel, sensor mengubahnya menjadi energi dan komputer yang dibangun di dalam kamera membaca seberapa banyak energi yang dihasilkan. Mengukur berapa banyak energi yang dimiliki setiap pixel memungkinkan sensor untuk menentukan area gambar mana yang terang dan gelap [11].



Gambar 2. 1 Komponen Kamera Ponsel

## 2.1. API

*Application Programming Interface* (API) memberikan konsep fungsi antarmuka pemrograman aplikasi, yang menjadi salah satu cara agar suatu aplikasi dapat diakses dan dimanfaatkan oleh pihak lain dan saling komunikasi antar sistem meskipun berbeda platform. Web Service sebagai API yang berperan dalam memberikan akses tersebut dalam proses pengambilan data. Melalui arsitektur Representational State Transfer (ReST) yang dioperasikan melalui Hypertext Transfer Protocol (HTTP), berisikan file JSON [17]. API terbagi menjadi 4 jenis, yaitu public API, private API, partner API dan composite API [18]. Untuk masing - masing penjelasan sebagai berikut :

### 1. Public API

Public API merupakan jenis API terbuka yang tersedia untuk digunakan oleh siapa saja, baik seorang pengembang aplikasi atau eksternal bisnis. API jenis ini umumnya melibatkan otentikasi dan otorisasi moderat, dan memonetisasi API dengan mengenakan biaya per panggilan akses yang dipakai publik untuk memanfaatkan API.

### 2. Private API

Private API merupakan API yang digunakan di dalam suatu perusahaan secara eksklusif untuk menghubungkan sistem dan data dalam bisnis.

Private API secara khusus dapat menghadirkan keamanan dan otentikasi yang lemah atau bahkan tidak ada keamanan sama sekali.

1. Partner API

Partner API merupakan API dari pihak ketiga atau mitra yang memfasilitasi kegiatan B2B (business to business) secara khusus lewat perjanjian tertentu. Penggunaan API dari pihak eksternal ini dilakukan secara selektif dalam beberapa bisnis yang ingin membagikan data dengan perusahaan lain secara khusus.

2. Composite API

Composite API secara umum merupakan penggabungan dua atau lebih API untuk menyusun urutan operasi yang terkait atau saling bergantung satu sama lain.

### 2.1. OCR Space API

OCR Space API merupakan layanan public API untuk melakukan OCR (*Optical Character Recognize*) atau sebuah proses untuk mendeteksi teks, angka, maupun sebuah pola karakter dari sebuah gambar. Salah satu teks yang dapat dikenali OCR Space adalah Bahasa Jepang, yang merupakan salah satu tulisan yang mempunyai banyak karakter unik [19] .

Untuk menggunakan OCR Space API ini memerlukan sebuah API *key* yang dapat diperoleh dengan cara membuat akun pada situs resmi OCR Space API. Terdapat beberapa tingkatan pengguna salah satu nya adalah FREE dengan batasan penggunaan 25.000 *request*/ bulan, dan ukuran file maksimal 1 MB. Sedangkan untuk pengguna PRO mampu melakukan request sampai 300.000 per bulan, dan ukuran file maksimal 5 MB dengan harga \$30/bulan. OCR Space API dapat memproses beberapa jenis file yaitu : PDF, GIF, PNG, JPG, TIFF, dan BMP.

Bahasa yang dapat diproses dan kode bahasa pada OCR Space API antara lain adalah :

1. Arabic = ara
2. Bulgarian = bul
3. Chinese (Simplified) = chs

1. Chinese (Traditional) = cht
2. Croatian = hrv
3. Czech = cze
4. Danish = dan
5. Dutch = dut
6. English = eng
7. Finnish = fin
8. French = fre
9. German = ger
10. Greek = gre
11. Hungarian = hun
12. Korean = kor
13. Italian = ita
14. Japanese = jpn
15. Polish = pol
16. Portuguese = por
17. Russian = rus
18. Slovenian = slv
19. Spanish = spa
20. Swedish = swe
21. Turkish = tur
22. Hindi = hin
23. Kannada = kan
24. Persian (Fari) = per
25. Telugu = tel
26. Tamil = tam
27. Thai = tai
28. Vietnamese = vie

## 2.1. Google Translate API

Google Translate API adalah sebuah mesin penerjemah yang dikembangkan oleh raksasa teknologi Google. Google translate mampu menerjemahkan 133 bahasa dengan tingkat akurasi mencapai 85% akurat. Hasil terjemahan google translate dinilai tepat karena google translate telah mengalami peningkatan kualitas penerjemahan dari metode *statistical machine translation* menjadi *neural mechanic translation* [12].

Google Translate API adalah salah satu layanan dari Google Cloud Platform yang bersifat public API dan membutuhkan sebuah API Key. Google Translate API mampu menerjemahkan lebih 1.000.000 karakter. Google Translate ini dilengkapi dengan fitur auto-detect dan auto-suggestion, dimana mampu mendekteksi dari bahasa mana berasal dan mampu mengkoreksi kata jika terjadi kesalahan pengetikan.

## 2.2. Visual Studio Code

VS Code adalah perangkat lunak teks editor populer yang dibuat oleh Microsoft untuk sistem operasi Linux, Windows, dan Mac. Bahasa pemrograman yang didukung adalah Javascript, Typescript, NodeJs, dan bahasa pemrograman lainnya dengan bantuan plugin yang bisa dipasang lewat marketplace VS code seperti C++, C#, Python, Go, Java, dsb.

Salah satu keunggulan dari VSCode adalah kustomisasi ruang kerja, dimana pengguna dapat menyunting sesuai dengan kebutuhan. Belum lagi *extension* yang ditawarkan juga beragam, seperti *intellisense*, *git integration*, *debugging*, *emmet* dan banyak lagi *extension* yang akan membantu selama proses pengembangan. Pembaruan versi VSCode ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VSCode dengan teks editor lainnya.

## 2.1. Bahasa Pemrograman Javascript dan JSON

JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan website agar lebih dinamis dan interaktif. JavaScript dapat meningkatkan fungsionalitas pada halaman web, sehingga memungkinkan untuk membuat sebuah *tools*, bahkan *games* di website. JavaScript atau JS merupakan bahasa pemrograman jenis *interpreter*, sehingga tidak memerlukan *compiler* untuk menjalankannya. JavaScript memiliki fitur-fitur seperti berorientasi objek, client-side, high-level programming, dan loosely typed.

JSON atau Javascript Object Notation adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah untuk dibuat oleh komputer. JSON format teks yang tidak bergantung dengan bahasa pemrograman apapun, karena menggunakan gaya bahasa umum digunakan oleh bahasa pemrograman C, C++, C#, Java, Javascript, Pearl, Python dan lain - lain. Oleh karena sifatnya flexible, menjadikan JSON ideal sebagai bahasa pertukaran-data. Selain menjadi bahasa pertukaran-data, JSON digunakan sebagai configuration bahkan database seperti MongoDB.

## 2.2. Svelte JS

Svelte JS adalah sebuah framework JavaScript untuk menulis komponen antarmuka atau *user interface*. Cara kerja framework Svelte JS lebih mirip seperti *compiler*, dimana kode yang ditulis di svelte akan dikonversikan ke dalam javascript yang sudah dioptimasi. Salah satu fitur kunci dari Svelte adalah ukuran aplikasi yang ditulis menjadi lebih kecil karena proses kompilasi yang dilakukan. Dan semua itu dicapai tanpa bantuan virtual DOM sehingga ukuran aplikasi Svelte menjadi relatif lebih kecil dan ringan.

## 2.3. Tailwind CSS

Tailwind CSS adalah framework CSS dengan konsep utility-first untuk membangun desain antarmuka khusus dengan cepat. Salah satu kelebihan dari Tailwind CSS adalah didesain untuk disesuaikan, pasalnya Tailwind CSS menggunakan

*class - class* kecil sehingga style terkesan unik. Dan Kelebihan lainnya adalah responsif dan component friendly.

## 2.1. PWA (Progressive Web Apps)

Progressive Web App ( PWA ) adalah sebuah aplikasi yang dibangun dengan melakukan optimasi pada sebuah website. Optimasi yang dilakukan tidak hanya akan membuat website menjadi lebih cepat namun juga mampu memberikan pengalaman layaknya menggunakan aplikasi mobile (native apps), seperti splash screen, push notification, akses kamera, penggunaan gps, offline mode dan lain - lain. PWA memungkinkan untuk dipasang lintas platform dengan satu kode sumber [20].

## 2.2. Local Storage

Local Storage merupakan salah satu cara yang dapat digunakan untuk menyimpan data di web browser. Kelebihan dari local storage adalah data yang disimpan tidak memiliki masa kadaluarsa bahkan ketika browser ditutup. Local Storage dapat menyimpan data sampai 10 MB, dengan seperti JSON.

A screenshot of a code editor with a dark background and light text. The code is as follows:

```
localStorage.setItem("str", "XYZ");  
const getStr = localStorage.getItem("str");  
console.log("localStorage.getItem(): ", getStr);
```

The words "Key" and "Value" are written in red above the strings "str" and "XYZ" respectively in the first line of code. The strings "str" and "XYZ" are enclosed in red boxes.

Gambar 2. 2 Contoh Implementasi Localstorage

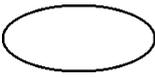
## 2.1. UML (Unified Modeling Language)

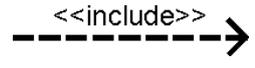
UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, kebutuhan, dan pendokumentasian sistem. UML dapat mempermudah pengembangan perangkat lunak serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Jenis diagram pada UML sebagai berikut [21].

### 2.13.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat dengan menggambarkan hubungan interaksi antara sistem dan aktor. Use case diagram bekerja dengan mendeskripsikan tipe interaksi antara aktor sebuah sistem dengan sistem itu sendiri. Simbol - simbol yang ada pada use case diagram bisa dilihat pada table 2.5[21].

Tabel 2. 5 Simbol pada Use Case Diagram

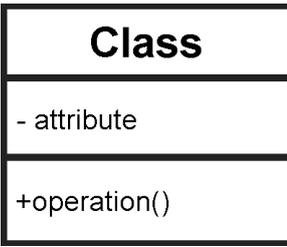
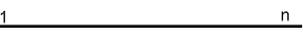
Simbol	Deskripsi
	<b>Aktor</b> : Mewakili peran orang / user, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	<b>Use Case</b> : Abstraksi dan interaksi antara sistem dan aktor
	<b>Association</b> : Abstraksi dari penghubung antara aktor dengan use case.
	<b>Generalisasi</b> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case

	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case tambahan fungsional dari use case lainnya jika suatu kondisi tertentu

### 2.13.1 Class Diagram

Class diagram merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Class diagram terdiri dari atribut dan operasi dengan tujuan pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai [21]. Untuk simbol - simbol class diagram bisa dilihat pada tabel 2.6.

Tabel 2. 6 Simbol pada Class Diagram

Simbol	Deskripsi
	<p><b>Kelas / Class :</b> Penggambaran dari struktur sistem yang akan ditampilkan dalam sistem informasi</p> <p><b>Attribute :</b> Penggambaran mengenai keadaan suatu objek di dalam kelas</p> <p><b>Operation :</b> Penggambaran mengenai suatu fungsi di dalam kelas</p>
	<b>Asosiasi / Association :</b> Relasi antar kelas dengan makna umum, biasanya juga disertai dengan <i>multiplicities</i>
	<b>Generalisasi / Generalized :</b> Relasi yang menghubungkan antara sub kelas dengan super kelas.

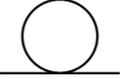
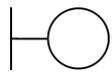
### 2.13.2 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Gambar Sequence diagram dibuat minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting

semua use case yang telah didefinisikan interaksinya pesan sudah dicakup pada sequence diagram sehingga semakin banyak use case yang didefinisikan, maka sequence diagram yang harus dibuat juga semakin banyak

[21]. Untuk contoh simbol - simbol pada sequence diagram bisa dilihat pada tabel 2.7.

Tabel 2. 7 Simbol pada Sequence Diagram

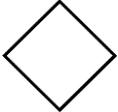
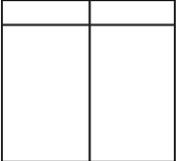
Simbol	Deskripsi
	<b>Aktor</b> : Mewakili peran orang / user, sistem yang lain, atau alat ketika berinteraksi dengan sistem
	<b>Entity Class</b> : Menggambarkan hubungan yang akan dilakukan
	<b>Boundary Class</b> : Menggambarkan hubungan suatu <i>form</i> , <i>report</i> , <i>hardware interface</i> seperti <i>printer</i> atau <i>scanner</i> .
	<b>Control Class</b> : Menggambarkan penghubung <i>boundary</i> dengan <i>table</i>
	<b>A Focus of Control &amp; lifetime</b> : Menggambarkan tempat mulai dan berakhirnya message
	<b>Message</b> : Menggambarkan pengiriman pesan

### 2.13.1 Activity Diagram

Activity diagram merupakan diagram yang menggambarkan workflow atau aktivitas dari sebuah sistem yang ada pada perangkat lunak [21]. Untuk contoh simbol - simbol activity diagram bisa dilihat pada tabel 2.8.

Tabel 2. 8 Simbol pada Activity Diagram

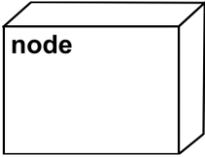
Simbol	Deskripsi
	<b>Status Awal</b> : Mewakili awal dari sebuah alur

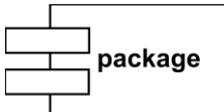
	<b>Aktivitas</b> : Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	<b>Percabangan / Decision</b> : Percabangan dimana ada pilihan aktivitas lebih dari satu.
	<b>Status Akhir</b> : Mewakili akhir dari sebuah alur
	<b>Penggabungan / Join</b> : Penggabungan aktivitas lebih dari satu menjadi satu aktivitas.
	<b>Swimlane</b> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### 2.13.1 Development Diagram

Development diagram merupakan adalah suatu diagram yang dapat digunakan untuk menggambarkan urutan proses atau kegiatan yang terjadi dalam suatu sistem. Development diagram biasanya digunakan untuk menggambarkan proses bisnis, perancangan sistem, atau pembangunan program. Untuk contoh simbol - simbol development diagram bisa dilihat pada tabel 2.9.

Tabel 2. 9 Simbol pada Flowchart

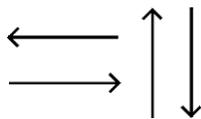
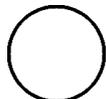
Simbol	Keterangan
	<b>Node</b> : Mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ) Berisi komponen – komponen, atau artifact.

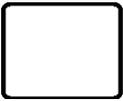
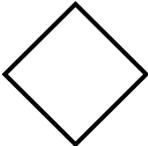
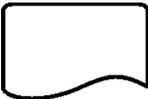
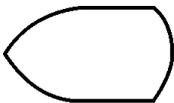
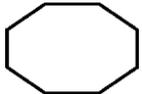
	<b>Package :</b> Sebuah modular
	<b>Artifact :</b> Berisi file yang yang bisa dieksekusi.
	<b>Link :</b> Relasi antar node
	<b>Dependency / kebergantungan :</b> Kebergantungan antar node , arah panah mengarah pada node yang digunakan

## 2.1. Flowchart

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Flowchart juga berfungsi sebagai gambaran alur sebuah program dari satu proses ke proses lain. Sehingga, alur program menjadi mudah dipahami dan menyederhanakan rangkaian prosedur. Setiap langkah digambarkan dalam bentuk diagram atau simbol dan dihubungkan dengan garis atau arah panah [22]. Berikut Simbol - simbol yang ada pada flowchart bisa dilihat pada 2.10.

Tabel 2. 10 Simbol pada Flowchart

Simbol	Keterangan
	<b>Flow / Connection Line :</b> Simbol yang digunakan untuk menggabungkan antara simbol satu dengan simbol lainnya.
	<b>On-Page Reference :</b> Simbol untuk keluar - masuk atau penyambung proses dalam lembar kerja yang sama.

	<b>Process</b> : Simbol yang menyatakan suatu proses yang dilakukan komputer
	<b>Off-Page Reference</b> : Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda
	<b>Terminator</b> : Simbol yang menyatakan awal atau akhir suatu program
	<b>Decision</b> : Simbol yang menunjukkan kondisi tertentu suatu proses yang dilakukan komputer
	<b>Input/output</b> : Simbol yang menyatakan proses input atau output tanpa tergantung peralatan
	<b>Manual operation</b> : Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer
	<b>Document</b> : Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik atau output yang perlu dicetak
	<b>Predefined Process</b> : Simbol untuk pelaksanaan suatu bagian sub-program atau prosedur
	<b>Display</b> : Simbol yang menyatakan peralatan output yang digunakan
	<b>Preparation</b> : Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal

## 2.1. Black Box Testing

*Black box testing* adalah metode pengujian perangkat lunak yang dilakukan tanpa memperhatikan struktur atau kode program di balik aplikasi atau sistem yang diuji. Metode ini didasarkan pada pengujian fungsionalitas dan spesifikasi eksternal

sistem tanpa memperhatikan rincian internal dari perangkat lunak atau sistem yang diuji.

*Black box testing* juga didasarkan pada prinsip bahwa pengguna atau pengguna potensial aplikasi atau sistem tidak perlu mengetahui detail teknis dari perangkat lunak atau sistem yang digunakan. Sebaliknya, pengguna hanya perlu tahu bagaimana cara menggunakan aplikasi atau sistem tersebut untuk memenuhi kebutuhan mereka. Metode pengujian ini juga dilakukan untuk menguji aplikasi atau sistem pada berbagai kondisi, seperti saat menerima input yang tidak valid, input yang sangat besar, input yang sangat kecil, atau ketika sistem bekerja di bawah beban tinggi. Hal ini memastikan bahwa sistem dapat bekerja dengan baik di berbagai kondisi dan dapat memenuhi kebutuhan pengguna secara efektif.

Dalam keseluruhan, landasan teori dari *black box testing* adalah pengujian fungsionalitas dan spesifikasi eksternal dari sistem atau aplikasi tanpa memperhatikan detail teknis dari perangkat lunak atau sistem yang diuji. Metode ini digunakan untuk memastikan bahwa aplikasi atau sistem dapat memenuhi kebutuhan pengguna secara efektif dan dapat bekerja dengan baik di berbagai kondisi[23].

