



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Judul

##### 2.1.1 Pengertian Implementasi

Menurut Ali (2017:51), “Implementasi merupakan penyediaan sarana untuk melaksanakan sesuatu yang menimbulkan dampak atau akibat terhadap sesuatu”.

Sedangkan Menurut *Browne dan Wildavsky* (dalam Firdianti, 2018:19), mengemukakan bahwa “Implementasi adalah perluasan aktivitas yang saling menyesuaikan”.

Berdasarkan pengertian yang diungkapkan oleh pengarang di atas dapat disimpulkan bahwa implementasi merupakan suatu kegiatan atau aktivitas yang menimbulkan dampak atau akibat terhadap sesuatu yang saling menyesuaikan.

##### 2.1.2 Pengertian Metode *Weight Product*

Latif, dkk. (2018:26) menjelaskan, metode *Weight Product* (WP) merupakan salah satu metode yang sederhana dengan perkalian untuk menghubungkan rating atribut dimana setiap atribut harus dipangkatkan dengan bobot atribut yang bersangkutan. Hal tersebut dinamakan normalisasi”.

Menurut Risawandi dan Rahim (dalam Limbong, dkk. 2020:38), *Weight Product* (WP) merupakan metode sistem pengambilan keputusan yang dilakukan dengan cara melakukan perkalian, guna menghubungkan rating atribut dan setiap atribut lebih dahulu harus dipangkatkan dengan bobot atribut yang sesuai, apakah positif atau negatif.

Berdasarkan pengertian yang diungkapkan oleh pengarang diatas, dapat disimpulkan bahwa Metode *Weight Product* (WP) adalah metode dalam sistem pengambilan keputusan dengan perkalian rating atribut, dimana rating atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan.



Tahapan dalam menggunakan metode *Weight Product* (WP) yaitu: (Limbong, dkk. 2020:39).

1. Menentukan tingkat prioritas bobot setiap kriteria
2. Menghitung bobot kriteria ( $W_j$ )
3. Melakukan perhitungan vektor S
4. Melakukan perhitungan vektor V, dengan cara (hasil perhitungan masing-masing vektor  $S_i$  dibagi dengan jumlah hasil perhitungan seluruh vektor S)
5. Dari hasil perhitungan vektor V, sebagai dasar pengambilan keputusan. Nilai V, terbesar merupakan pilihan alternatif yang terbaik.

Rumus perhitungan bobot sehingga memperoleh total bobot  $\sum w_j = 1$

$$W_j = \frac{W_j}{\sum_{j=1}^n W_j}$$

Rumus vektor  $S_i$

$$S_i = \prod_{j=1}^n X_{ij} W_j, \text{ dengan } i = 1, 2, 3, \dots, n$$

$w_j$  adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut beban/biaya.

Rumus vektor untuk melakukan perankingan

$$V_i = \frac{S_i}{S_1 + S_2 + S_3 + \dots + S_n}, \text{ dengan } i = 1, 2, 3, \dots, n$$

### 2.1.3 Pengertian *Self-Assesment*

Menurut Mubarak, dkk. (2019:5), *Self-Assesment System* adalah sistem pemungutan pajak dimana wajib pajak menentukan sendiri jumlah pajak yang dibayar, dan dilaporkan.

Lihin dan Wijana (2019:2), mengemukakan bahwa Ketentuan Umum Perpajakan/KUP adalah ketentuan formal perpajakan di Indonesia yang menganut sistem *self-assesment*. Sistem ini memberikan kepercayaan penuh kepada wajib



pajak untuk menghitung, memperhitungkan, menyetor, dan melaporkan seluruh pajak yang menjadi kewajibannya.

Sistem *Self-Assesment* adalah wajib pajak sendiri yang menghitung, menetapkan, menyetorkan, dan melaporkan pajak yang terutang. (Karmila, 2018:15).

Berdasarkan pengertian yang diungkapkan oleh pengarang, penulis menyimpulkan bahwa *self-assesment* adalah sistem yang memberikan wewenang pada wajib pajak untuk menentukan sendiri jumlah pajak, menghitung, memperhitungkan, menyetor, dan melaporkan seluruh pajak yang menjadi kewajibannya.

#### **2.1.4 Pengertian Kepatuhan Wajib Pajak**

Menurut Safri Numantu (dalam Pohan, 2016:544), Kepatuhan dapat didefinisikan sebagai suatu keadaan saat wajib pajak memenuhi semua kewajiban perpajakan dan melaksanakan hak perpajakannya.

Kiryanto (dalam Salman dan Tjaraka, 2019:242) mengatakan, “Kepatuhan wajib pajak adalah tingkah laku Wajib Pajak yang memasukkan dan melaporkan pada waktunya informasi yang diperlukan, mengisi secara benar jumlah pajak yang terutang, dan membayar pajak pada waktunya, tanpa ada tindakan pemaksaan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa kepatuhan wajib pajak adalah orang pribadi, badan atau pemotong yang memenuhi semua kewajiban perpajakan dan melaksanakan hak perpajakannya dengan cara melaporkan pada waktunya, mengisi secara benar jumlah pajak yang terutang, dan membayar pajak pada waktunya, tanpa ada tindakan pemaksaan.

#### **2.1.5 Pengertian Pajak Hotel**

Salman dan Tjaraka (2019:191), menjelaskan bahwa Pajak Hotel merupakan dana/iuran yang dipungut atas penyedia jasa penginapan yang disediakan sebuah badan usaha tertentu yang jumlah ruang/kamarnya lebih dari 10. Pajak tersebut dikenakan atas fasilitas yang disediakan oleh hotel tersebut. Tarif pajak hotel



dikenakan sebesar 10% dari jumlah yang harus dibayarkan dan masa pajak hotel adalah 1 bulan.

Pajak Hotel adalah pajak atas pelayanan yang disediakan oleh hotel. Hotel adalah fasilitas penyediaan jasa penginapan/peristirahatan. (Tambunan dan Wilson, 2019:311).

Menurut Ayza (2017:94), Pajak Hotel adalah pajak yang dipungut oleh pemerintah daerah kabupaten atau kota atas pelayanan yang disediakan oleh Hotel dengan pembayaran, termasuk jasa penunjang sebagai kelengkapan Hotel yang sifatnya memberikan kemudahan dan kenyamanan, termasuk fasilitas olahraga dan hiburan.

Berdasarkan pernyataan yang diungkapkan oleh pengarang, penulis menyimpulkan bahwa Pajak Hotel adalah dana/iuran pajak atas pelayanan yang disediakan oleh Hotel dengan pembayaran, termasuk jasa penunjang sebagai kelengkapan Hotel yang sifatnya memberikan kemudahan dan kenyamanan, dengan tarif pajak hotel sebesar 10% dan masa pajak hotel 1 bulan.

## **2.2 Teori Program**

### **2.2.1 Basis Data (*Database*)**

Menurut Firly (2019:110), “*Database* adalah sekumpulan informasi yang diorganisasikan sehingga mudah diakses, dikelola dan diperbaharui”.

Basis data merupakan kumpulan berbagai informasi yang tersimpan dalam komputer secara sistematis sehingga bisa diperiksa dengan menggunakan suatu aplikasi komputer guna memperoleh informasi dari basis data tersebut. (Limbong, dkk. 2020:34).

Berdasarkan pengertian yang diungkapkan oleh pengarang, penulis mengambil kesimpulan bahwa basis data (*database*) adalah kumpulan informasi yang diangkat atau diambil dari kejadian nyata (fakta) yang saling berhubungan sehingga mudah diakses, dikelola dan diperbaharui untuk memenuhi kebutuhan penggunaannya.



### 2.2.2 Framework Laravel

*Framework* adalah komponen programmer yang siap digunakan ulang kapan saja sehingga programmer tidak harus membuat script yang sama untuk tugas yang sama. (Yudhanto dan Prasetyo, 2019:10).

Laravel merupakan salah satu *framework* PHP paling populer yang akan memudahkan programmer dalam membuat aplikasi *web* berbasis PHP. (Abdulloh, 2018:3).

Menurut Badiyanto (2018:2), “*Framework* secara sederhana dapat diartikan sebagai kumpulan dari fungsi atau prosedur dan *class* untuk tujuan tertentu yang sudah siap untuk digunakan sehingga mempermudah dan mempercepat *programmer* dalam membuat program tanpa harus membuat fungsi atau *class* dari awal”.

Menurut Yudhanto dan Prasetyo (2018:17), Laravel adalah salah satu *framework* PHP yang dikembangkan oleh Taylor Otwell, proyek laravel dimulai pada april 2011.

Terdapat beberapa keunggulan dari *framework* Laravel yaitu sebagai berikut: (Abdulloh, 2017:3).

1. Laravel memiliki banyak fitur yang tidak dimiliki oleh *framework* lain.
2. Laravel merupakan *framework* PHP yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti sehingga *programmer* pemula sekalipun akan mudah paham kegunaan suatu sintaks walaupun belum mempelajarinya.
3. Laravel memiliki dokumentasi yang cukup lengkap, bahkan setiap versinya memiliki dokumentasi tersendiri mulai dari cara instalasi hingga penggunaan fitur-fiturnya.
4. Laravel digunakan oleh banyak *programmer* sehingga banyak *library* yang mendukung Laravel yang diciptakan para *programmer* pecinta Laravel.
5. Laravel didukung oleh Composer sehingga *library-library* diperoleh dengan mudah dari internet menggunakan Composer.
6. Laravel memiliki *template engine* tersendiri yang diberi nama *blade* yang memudahkan kita menampilkan data pada *template* HTML.



Ada beberapa fitur yang dimiliki oleh *framework* Laravel yaitu sebagai berikut (Supardi dan Sulaeman, 2019:2).

- a) **Bundles** yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam diaplikasi.
- b) **Eloquent ORM** merupakan penerapan PHP lanjutan dari pola “*active record*” menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*.
- c) **Application Logic** merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* maupun sebagai bagian dari deklarasi *route*.
- d) **Reverse Routing**, mendefinisikan hubungan antara *link* dan *route*.
- e) **Restful Controller**, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
- f) **Class Auto Loading**, menyediakan otomatis *loading* untuk kelas-kelas PHP.
- g) **View Composers** adalah kode unit *logical* yang dapat dieksekusi ketika *view* sedang *loading*.
- h) **IoCContainer** memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip *control* pembalik.
- i) **Migrations** menyediakan versi sistem *control* untuk skema *database*.
- j) **Unit Testing** banyak tes untuk mendeteksi dan mencegah regresi.
- k) **Automatic Pagination** menyederhanakan tugas dari penerapan halaman.

### 2.2.3 Website

Abdulloh (2018:1), mengatakan bahwa *Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia.

Menurut Sa'ad (2020:5), *Website* merupakan kumpulan file yang terletak pada komputer yang terhubung ke internet.

*Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara, dan video atau gabungan



dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang diseluruh dunia. (Enterprise, 2017:1).

Berdasarkan pengertian yang diungkapkan pengarang, dapat disimpulkan bahwa *website* adalah kumpulan halaman yang berisi informasi data digital yang terletak pada komputer yang terhubung ke internet yang dapat diakses dan dilihat oleh semua orang di seluruh dunia.

#### **2.2.4 Pengertian MySQL**

Menurut Kadir (2018:170), “MySQL (baca: “Mai-es-kyu-el”) merupakan sistem manajemen *database* terkenal yang sekarang dimiliki oleh *Oracle* dan salah satu produknya bernama MySQL Community Server bersifat *Open Source*”.

MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, Kepopuleran ini karena ditunjang karena performansi *query database* yang saat itu bisa dikatakan paling cepat, dan jarang bermasalah. (Abdulloh, 2017:301).

Dari pernyataan di atas, penulis menyimpulkan bahwa MySQL adalah suatu sistem atau media yang berguna untuk manajemen atau mengelola *database*, dan MySQL merupakan sistem manajemen basis data yang paling populer di lingkungan Linux serta MySQL bersifat *Open Source*.

#### **2.2.5 Hypertext Preprocessor (PHP)**

Menurut Supono (2018:3), (PHP; *Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML.

Sedangkan Menurut Kadir (2018:358), “PHP merupakan bahasa pemrograman skrip yang diletakkan dalam server yang biasa digunakan untuk membuat aplikasi Web yang bersifat dinamis”.

Dari pernyataan diatas, penulis menyimpulkan PHP merupakan bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang diletakkan dalam server yang dapat ditambahkan ke dalam



HTML, untuk membuat dan pengembangan suatu *website* atau menghasilkan halaman *website* yang dinamis.

### 2.2.6 Laragon

Menurut Harianto, dkk. (2019:14), Laragon adalah perangkat lunak yang mendukung banyak sistem operasi, berfungsi sebagai server diri sendiri/localhost laragon menyediakan services, tools, fitur mulai dari Apache, MySQL, PHP Server, Mamchaced, Redis, Composer, Xdebug, PhpMyAdmin, Cmder dan Laravel.

### 2.2.7 Sublime Text

*Sublime text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi *phyton* API. (Azis, dkk. 2019:153).

Menurut Rerung (2018:7), Sublime text adalah aplikasi yang gratis maupun yang berbayar. Sublime text mempunyai banyak keunggulan seperti:

- a. *Auto-Complation*,
- b. *Minimap/Document Map*,
- c. *Goto Defenition*,
- d. *Split Editing*,
- e. *Column Editing*,
- f. *Multi Editing, dll.*

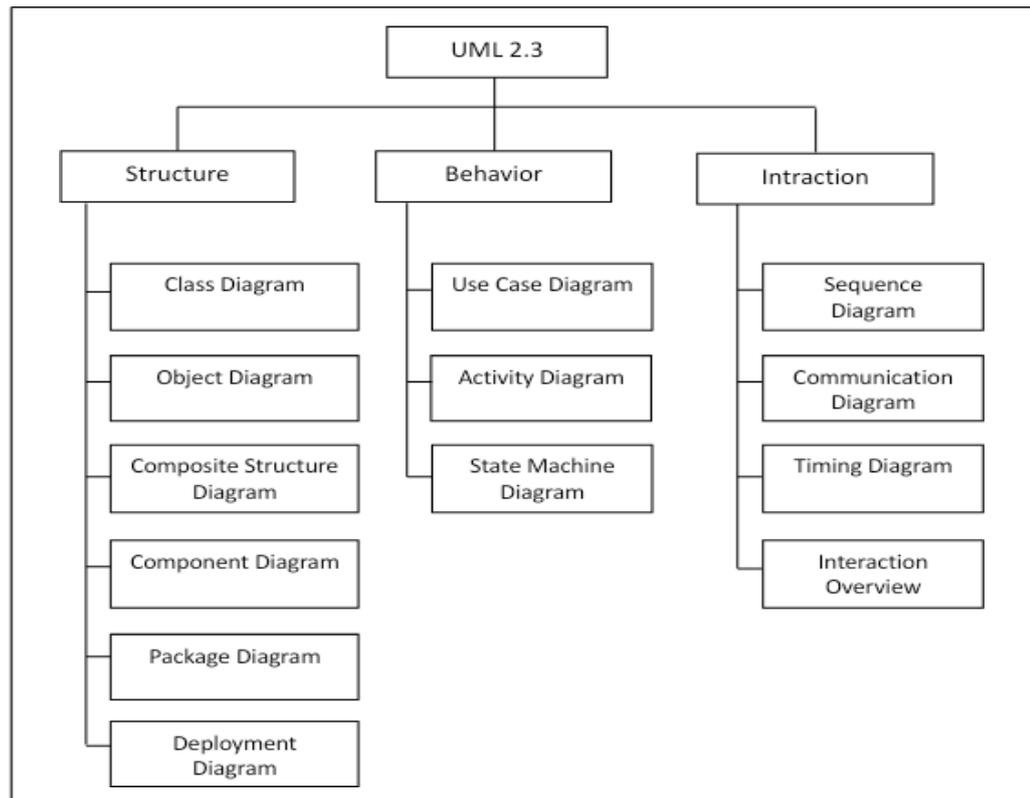
Berdasarkan pengertian yang diungkapkan oleh pengarang di atas, penulis menyimpulkan bahwa *Sublime text* adalah aplikasi editor yang gratis maupun yang berbayar yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi *phyton* API.



## 2.3 Teori Khusus

### 2.3.1 Unified Modeling Language (UML)

Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah.



**Gambar 2.1** Macam-macam Diagram UML

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141):

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

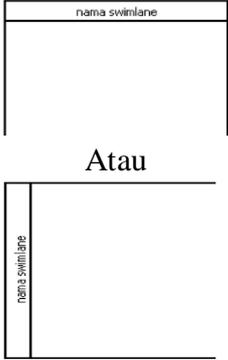


### 2.3.1.1 Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *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.

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut: (Sukamto dan Shalahuddin, 2018:162).

**Tabel 2.1** Simbol-simbol pada *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4.	penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Sukamto dan Shalahuddin 2018:162)



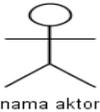
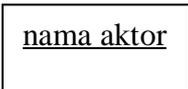
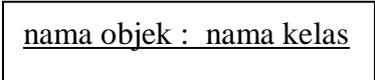
### 2.3.1.2 Sequence Diagram

Sukanto dan Shalahuddin (2018:165), menjelaskan bahwa diagram *sequence* 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 beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

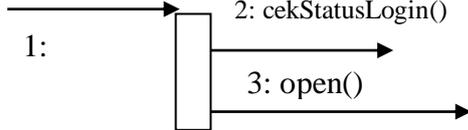
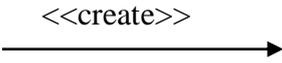
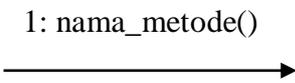
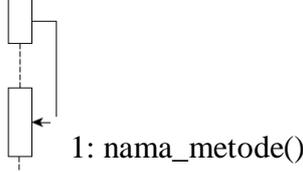
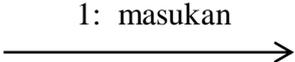
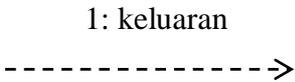
Banyaknya diagram *sequence* yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan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

Berikut simbol-simbol pada *Sequence Diagram*:

**Tabel 2.2** Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	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.
2.	<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.



No	Simbol	Deskripsi
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  maka cekStatusLogin() dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif.
5.	Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe call 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe send 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe return 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

(Sumber: Sukamto dan Shalahuddin 2018:165-166)

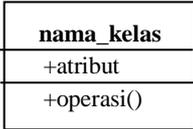


### 2.3.1.3 Class Diagram

Sukamto dan Shalahuddin (2018:141), menyebutkan *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram Class* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Sukamto dan Shalahuddin (2018:146), Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

**Tabel 2.3** Simbol-simbol pada *Class Diagram*

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2.	antarmuka / interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
4.	asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7.	<i>Link</i> 	Relasi antar <i>node</i>

(Sumber: Sukamto dan Shalahuddin 2018:146-147)

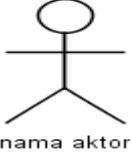
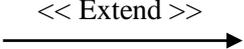


### 2.3.1.4 Use Case Diagram

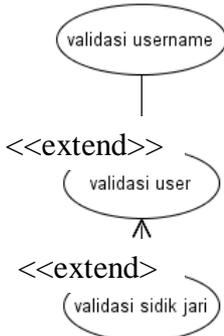
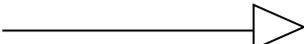
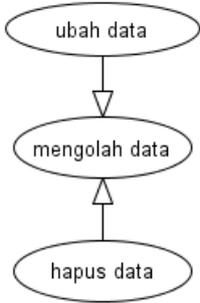
Menurut Sukamto dan Shalahuddin (2018:155), “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Simbol-simbol yang digunakan dalam *Use Case Diagram* adalah sebagai berikut: (Sukamto dan Shalahuddin, 2018:156).

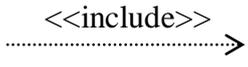
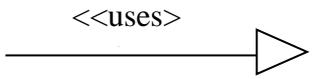
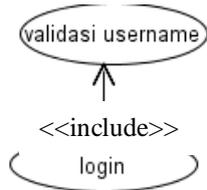
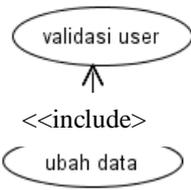
**Tabel 2.4** Simbol-simbol pada *Use Case Diagram*

No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i></p>
2.	<p>aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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>
3.	<p>asosiasi / <i>association</i></p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>.</p>
4.	<p>Ekstensi / <i>Extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i>, biasanya <i>use case</i> tambahan</p>



No	Simbol	Deskripsi
		<p>memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>
5.	<p>Generalisasi / <i>generalization</i></p> 	<p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>



No	Simbol	Deskripsi
6.	<p>menggunakan / include / uses</p> <p style="text-align: center;">    </p>	<p>relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</li> </ul> <p style="text-align: center;">  </p> <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <p style="text-align: center;">  </p> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

(Sumber: Sukamto dan Shalahuddin 2018:156-158)