

## BAB II LANDASAN TEORI

### 1.1 Tinjauan Pustaka

Dalam penelitian ini, penulis melakukan tinjauan pustaka terhadap penelitian-penelitian terdahulu sebagai penunjang penelitian yang akan dilakukan oleh penulis. Di bawah ini adalah tinjauan literatur yang telah diteliti sebelumnya:

**Tabel 2.1** Daftar Literatur

No	Penulis	Tahun	Judul
1	Lie, Ignatius Rivaldo Naufal, Muhammad Afif Kurniawan, Alek Bayu	2022	Perancangan Aplikasi Raporku Sekolah Dasar Berbasis <i>Android</i>
2	Fadillah, M Haris Khoirul Moenir, Ardianto	2021	Rancang Bangun Sistem Informasi <i>E-Rapor</i> Berbasis <i>Web</i> Dengan Metode <i>Waterfall</i> (Studi Kasus : SMKN 1 Kabupaten Tangerang)
3	Nurman Hidayat Kusuma Hati	2021	Penerapan Metode <i>Rapid Application Development (RAD)</i> dalam Rancang Bangun Sistem Informasi Rapor <i>Online</i> (SIRALINE)
4	Pengolahan, Aplikasi Nilai, S M A Medan, Mariana	2020	Aplikasi Pengolahan Data Nilai Rapor Berbasis <i>Web</i> di SMA Katolik Mariana Medan
5	Puspaningrum, Ajeng Savitri Neneng, Neneng Saputri, Intan Ariany, Fenty	2020	Pengembangan <i>E-Rapor</i> Kurikulum 2013 Berbasis <i>Web</i> Pada Sma Tunas Mekar Indonesia

### **2.1.1 Tinjauan Literatur 1**

Berdasarkan penelitian yang telah dilakukan oleh Lie, Ignatius Rivaldo Naufal, Muhammad Afif Paniza, Mehri Kurniawan, Alek Bayu pada tahun 2022 dari Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang pada penelitian ini Permasalahan yang ada adalah semenjak munculnya wabah Covid 19 ini sangat mengganggu kegiatan belajar mengajar di sekolah, masalah lain yang juga mengganggu adalah proses pembagian rapor siswa kepada orang tua wali murid karena sebelum adanya wabah covid 19 pembagian rapor dilakukan secara *offline* kini sangat sulit dilakukan mengingat adanya aturan pemerintah mengenai pembelajaran dilakukan *via online*. Untuk mengatasi masalah tersebut maka diperlukannya suatu sistem untuk mengolah nilai rapor siswa yaitu dengan membangun sistem rapor berbasis *mobile*. Dalam pembuatan desain tatap muka aplikasi rapor ini digunakan aplikasi *Figma* dalam bentuk aplikasi *web*.

Hasil dari penelitian ini adalah perancangan sistem aplikasi Raporku berbasis *mobile* yang di desain secara menarik serta mudah dioperasikan oleh *user* dan dapat dengan mudah siswa dan wali murid melihat hasil nilai secara cepat dan akurat.

### **2.1.2 Tinjauan Literatur 2**

Berdasarkan penelitian yang telah dilakukan oleh M. Fadillah, dan A. Moneir pada tahun 2021 dari Program Studi Teknik Informatika, Universitas Pamulang pada penelitian ini permasalahan yang dibahas yaitu belum adanya sistem pengolahan nilai rapor yang dapat memberikan informasi terkait nilai kepada siswa melalui internet, sistem yang berjalan saat ini masih mempunyai kelemahan sering kali ditemukan permasalahan pada data nilai siswa karena pengolahan yang dilakukan masih manual yaitu dengan cara menghitung jumlah rata-rata nilai

menggunakan kalkulator dan ditulis pada buku daftar kelas. Untuk mengatasi masalah tersebut maka perlu dilakukan perbaikan sistem yang berjalan dengan kebutuhan saat ini yaitu dengan adanya sistem khusus yang digunakan untuk pengolahan nilai siswa sehingga proses pengolahan data menjadi lebih mudah dan cepat. Perancangan sistem ini menggunakan model *waterfall*, yang terdiri dari 5 tahapan yaitu : Analisis, merancang/mendesain, penerapan *coding*, melakukan integrasi, mengoperasikan program.

Hasil dari penelitian ini adalah sistem rapor yang terkomputerisasi dan dapat diakses secara *online* agar dapat meminimalisir terjadinya kesalahan data dan manipulasi nilai siswa.

### **2.1.3 Tinjauan Literatur 3**

Oleh Nurman Hidayat dan Kusuma Hati pada tahun 2021 dari STMIK Antar Bangsa pada penelitian ini masalah yang terdapat adalah proses pengolahan nilai rapor yang ada pada SD Islam Imam Syafi'I belum efisien karena masing-masing guru mata pelajaran masih mengelola nilai menggunakan *Microsoft Excel* secara terpisah lalu kemudian diberikan kepada guru wali kelas untuk disatukan hingga menjadi buku rapor. Tentunya hal ini akan mempersulit wali kelas dalam proses pengelolaan nilai harus menghitung rata-rata nilai secara manual berdasarkan nilai masing-masing siswa dan dengan mata pelajaran yang berbeda-beda. Maka dari permasalahan tersebut diperlukannya sistem informasi berbasis website pada SD Islam Imam Syafi'I dalam membantu pengelolaan nilai rapor agar dapat memberikan kemudahan bagi guru mata pelajaran dan wali kelas dengan harapan memberikan kemudahan, kecepatan proses pengolahan data nilai siswa pada tiap akhir semester.

Hasil dari penelitian ini adalah sistem informasi rapor sebagai salah satu media *online* akademik untuk memberikan informasi pencapaian belajar siswa pada tiap akhir semester dan dapat membantu sekolah dalam melakukan pengarsipan, penginputan data siswa, guru dan mata pelajaran dengan sangat baik.

#### **2.1.4 Tinjauan Literatur 4**

Berdasarkan penelitian yang telah dilakukan oleh Pengolahan, Aplikasi Nilai, SMA Medan, Mariana pada tahun 2020 dari Program Studi Sistem Informasi, Universitas Sari Mutiara Indonesia Medan pada penelitian ini permasalahan yang terdapat adalah belum sepenuhnya pengolahan nilai siswa terkomputerisasi, sistem yang berjalan saat ini adalah menerima berkas nilai dari masing-masing guru mata pelajaran lalu dicatat pada buku nilai dan dilakukan penyalinan di buku rapor setelah proses penyalinan tersebut, rapor dibagikan kepada wali murid. Untuk mengatasi masalah tersebut maka diperlukannya sistem untuk mengolah data nilai siswa agar dapat memberikan *alternatif* bagi guru mata pelajaran dan wali kelas dalam proses penginputan nilai. Pada perancangan sistem ini digunakan metode penelitian yaitu penerapan *system development live cycle*.

Hasil dari penelitian ini adalah aplikasi pengolahan nilai rapor dengan memberikan kemudahan bagi guru dalam penginputan nilai siswa serta dapat menyimpan data guru, siswa, jadwal, kelas, mata pelajaran dengan mudah dan aman.

#### **2.1.5 Tinjauan Literatur 5**

Berdasarkan penelitian yang telah dilakukan oleh Puspaningrum, Ajeng Savitri, Neneng, Neneng Saputri, Intan Ariany, Fenty pada tahun 2020 dari Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia pada penelitian ini

permasalahan yang terdapat adalah proses pengolahan nilai siswa yang dilakukan masih manual dengan cara menuliskan nilai siswa pada kertas absensi kemudian di rekap dengan aplikasi *Microsoft Excel* yang kemudian berkas nilai tersebut diberikan kepada wali kelas dengan menggunakan *flashdisk* dan *email*. Tentunya, proses tersebut menyebabkan masalah adanya virus pada perangkat komputer yang digunakan oleh guru. Untuk mengatasi masalah tersebut pada penelitian ini mengembangkan suatu aplikasi yaitu E-Rapor Berbasis *Web* dalam upaya membantu SMA Tunas Mekar Indonesia untuk melakukan proses penilaian kegiatan belajar siswa yang akan memberikan kemudahan dan meringankan pekerjaan guru dan wali kelas karena tidak perlu lagi mengirimkan rekap nilai siswa menggunakan *flashdisk*, karena sudah dapat diinputkan secara langsung dan sudah terintegrasi dengan data yang lain.

Hasil Dari penelitian ini adalah pengembangan metode *prototype* dan implementasi sistem dengan *PHP* dan *MySQL* dapat membantu sekolah dalam memberikan kemudahan untuk melakukan pengolahan data nilai rapor siswa.

## **2.2 Pengembangan Sistem**

### **2.2.1 Rapid Application Development**

*Rapid Application Development* (RAD) merupakan bagian dari pengembangan model *System Development Life Cycle* (SDLC) model pengembangan perangkat lunak dengan menekankan pada siklus pengembangan sangat singkat, dapat menjadi acuan untuk melakukan suatu pengembangan sistem informasi yang baik pada segi kecepatan dan ketepatan (Rosa dan Shalahuddin, 2018). Berikut adalah tahapan pengembangan sistem *Rapid Application Development* (RAD):

1. Menentukan *Project Requirements*

Menentukan *project* requirement merupakan proses awal yang dilakukan untuk menentukan kebutuhan apa saja yang diperlukan untuk membangun sebuah *project*, setelah mendapatkan kebutuhan yang jelas, barulah menentukan hal-hal yang mendetail. Contoh seperti tujuan, timeline, dan budget yang diperlukan. Pada intinya tahapan awal ini berguna untuk memberikan gambaran pada *project* yang akan di kerjakan. Pada tahap ini, akan dilakukan pengumpulan informasi dengan melakukan observasi dan wawancara pada di Madrasah Aliyah Miftahul Huda Nambahdadi untuk mengetahui apa saja kebutuhan dan bagaimana cara mengatasi kebutuhan user tersebut.

2. Membuat *Prototype*

Pada tahapan selanjutnya adalah membuat *prototype*, dimana *developer* akan membuat *prototype* dari aplikasi yang diinginkan. Lengkap dengan fitur dan fungsi yang berbeda-beda. Tujuannya, untuk memberitahu apakah *prototype* yang dibuat sudah sesuai dengan keinginan dan kebutuhan *user*.

3. *Rapid Construction* dan Pengumpulan *Feedback*

Pada tahapan ketiga ini pengumpulan *feedback* yang diberikan oleh user. *Feedback* yang dimaksud disini mencakup fitur, fungsi, visual, dan juga *interface* dari program yang sedang dikembangkan. Setelah itu, *Prototype* akan dikembangkan lagi sampai *client* memberikan persetujuan untuk finalisasi produk. Dan seperti bahasan sebelumnya, kedua tahapan ini akan diulang terus-menerus, sampai hasilnya maksimal dengan keinginan *client*.

#### 4. Implementasi dan finalisasi

Langkah terakhir adalah implementasi hasil *feedback* yang diberikan *user* dan membuat *project* akhir. Dimana tugas utama *develover* adalah menutupi kekurangan yang mungkin terjadi ketika proses pengembangan aplikasi, tugas ini penting karena melakukan optimasi untuk stabilitas aplikasinya, memperbaiki *interface*, hingga melakukan *maintenance* dan menyusun dokumentasi.

### 2.3 Website

*Website* merupakan sekumpulan halaman *web*, yang biasanya berada dalam sebuah domain yang tempatnya berada di *World Wide Web* (WWW) di internet. Halaman *web* merupakan dokumen yang ditulis dengan format *Hyper Text Markup Language* (HTML), yang dapat diakses melalui *HTTP*, sebuah protokol yang menyediakan informasi dari server situs *web* untuk dapat ditampilkan kepada pengguna melalui *browser* pada perangkat komputer atau *handphone*, *website* juga digunakan untuk menyajikan informasi berupa teks, animasi, gambar dan suara baik statis maupun dinamis (Nurhadi, 2017).

### 2.4 Android

*Android* adalah sistem operasi berbasis Linux yang digunakan untuk perangkat seluler atau tablet. *android* pada awalnya dikembangkan oleh *Android Inc* dan didukung oleh Finansial dari *Google*, sistem operasi ini telah resmi dirilis pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance* dari perusahaan perangkat keras, perangkat lunak dan telekomunikasi, dengan tujuan untuk memajukan perangkat selular (Yudhayanto dan Wijayanto, 2017).

Berdasarkan data yang dihimpun dari situs website Statista, *android* adalah *Smartphone* cerdas yang memiliki pengguna paling banyak sejak tahun 2010. Pada tahun 2021 *android* menjadi pemegang pangsa pasar lebih dari 70%. Diperkirakan *android* akan memepertahankan kepemimpinannya di pasaran (Statista, 2022).

## **2.5 PHP**

*PHP* merupakan bahasa pemrograman yang digunakan dalam proses pembuatan *website* dan dapat digunakan bersama dengan *HTML*. Bahasa pemrograman *PHP* bersifat *serverside* yang artinya, kode/*script* pada *PHP* tidak dapat ditampilkan di halaman website dengan begitu saja, memerlukan proses oleh *web server* selanjutnya ditampilkan pada halaman *website* di *browser* (Pamungkas, 2017).

## **2.6 SQL**

*Structured Query Language* yang juga disebut SQL merupakan bahasa yang terstruktur khusus untuk digunakan dalam mengelola database. *Structured Query Language* (SQL) didefinisikan pertama kalinya oleh *American National Standards Institute* (ANSI) pada tahun 1986. SQL sendiri merupakan manajemen database bersifat *open source* dan *relational* yang artinya, data yang dikelola pada database akan diletakkan ke dalam beberapa tabel terpisah sehingga proses manipulasi data dapat dilakukan dengan cepat dan mudah (Rosa dan Shalahuddin, 2018).



## 2.7 XAMPP

*XAMPP* merupakan singkatan dari *Multi Platform (X)*, *Apache (A)*, *MySQL (M)*, *PHP (P)*, *Perl (P)*. aplikasi ringan dan sangat memudahkan untuk *developer web* untuk membuat *web server* lokal dengan berbagai macam kebutuhan. *XAMPP* dapat berjalan di berbagai macam sistem operasi seperti *Linux*, *Windows* dan *Mac OS* (Asyikin, 2019). Komponen yang ada pada *XAMPP* adalah sebagai berikut:

### 1. *Apache*

*Apache* dibuat oleh *Apache Software Foundation* dan dapat digunakan secara gratis. *Apache* merupakan sebuah *web server* yang memproses permintaan *client* dan meneruskan ke halaman yang dituju melalui *protocol HTTP*.

### 2. *MySQL*

*MySQL* merupakan aplikasi *DBMS (Database Management System)* yang sangat populer. Semua aplikasi *web* baik yang mudah maupun yang rumit membutuhkan *database* yang digunakan untuk menyimpan serta memproses data maka dari itu peran *MySQL* untuk memudahkan proses tersebut.

### 3. *PHP*

*PHP* merupakan singkatan dari *Hypertext Preprocessor* sebuah bahasa pemrograman *web* yang berjalan pada *server* serta sangat *powerfull* dan dapat dikombinasikan dengan *MySQL*.

### 4. *Perl*

*Perl* adalah bahasa pemrograman tingkat tinggi seperti *PHP* yang dapat digunakan untuk membuat aplikasi *web*.

## 5. *PhpMyAdmin*

*phpMyAdmin* digunakan untuk mengelola *database*, dalam hal ini *MySQL* melalui *web browser*. Sedangkan *Filezilla* digunakan untuk membuat *file server*.

## 2.8 *Codeignitier*

*Codeigniter* adalah *Web Application Framework* yang dapat digunakan untuk mempermudah para *developer* untuk mengembangkan aplikasi berbasis *web*. *Codeigniter* berisikan sekumpulan kode berupa *library* dan *tools* yang menjadi satu dalam suatu kerangka kerja (*Framework*). *Codeigniter* merupakan *framework web* yang digunakan untuk pemrograman *PHP* yang dirancang oleh Rick Ellis pada tahun 2006. *Codeigniter* menerapkan pola arsitektur *Model-View-Controller* (MVC) yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (Asyikin, 2019).

## 2.9 *Android Studio*

*Android Studio* adalah *software tools Integrated Development Environment* (IDE) yang digunakan untuk pengembangan aplikasi berbasis *Mobile Android*. *Android studio* pertama kali diluncurkan pada tanggal 16 Mei 2013 oleh Produk Manager Google, *Ellie Powers*. *Android studio* bersifat gratis dibawah naungan *Apache License 2.0*. *Android Studio* dapat berjalan di berbagai sistem operasi seperti *Windows*, *Mac OS* dan *Linux* (Yudhayanto dan Wijayanto, 2017).

## 2.10 ISO 25010

pengujian ISO 25010 merupakan bagian dari *Systems and Software Quality Requirements and Evaluation (SQuaRE)* yang merupakan versi lanjutan dari ISO 91261 yang telah direvisi secara teknis dengan menambahkan beberapa struktur dan

bagian dari standar model kualitas (Bass, Clements dan Kazman, 2022). Tujuan dari penggunaan kualitas ini adalah untuk mengukur sejauh mana produk atau sistem tersebut bisa digunakan oleh pengguna untuk memenuhi kebutuhan dalam mencapai tujuan yang diinginkan dengan efisiensi, efektivitas, kepuasan dalam konteks penggunaan yang spesifik, dan bebas dari resiko.

Pengujian dengan ISO 25010 terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer yaitu sebagai berikut:

1. *Functional Suitability*, merupakan sistem atau produk yang memberikan fungsional untuk memenuhi kebutuhan saat sistem atau produk tersebut digunakan pada keadaan tertentu.
2. *Reliability*, merupakan tingkat dimana suatu sistem atau produk dapat mempertahankan kinerjanya pada level tertentu ketika digunakan pada keadaan tertentu.
3. *Performance Efficiency*, merupakan tingkat dimana sistem atau produk menyediakan performa yang baik dengan sejumlah *resource* yang akan digunakan pada sistem atau produk.
4. *Usability*, merupakan tingkat dimana pada suatu sistem atau produk mudah dimengerti, mudah dipakai, dan menarik untuk digunakan.
5. *Security*, merupakan tingkat dimana pada suatu sistem atau produk menyediakan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, ataupun pengungkapan yang berbahaya.
6. *Compatibility*, merupakan kemampuan pada suatu komponen atau sistem untuk bertukar informasi.

7. *Maintainability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dimodifikasi, yang meliputi perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.
8. *Portability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dipindahkan dari satu ruang ke ruang lainnya.

### **2.11 Rapor**

Rapor merupakan laporan nilai siswa yang telah didapatkan oleh siswa selama menempuh pendidikan dalam satu semester. Pencatatan nilai rapor melibatkan berbagai pihak di sekolah, diantaranya wali kelas, guru pelajaran dan pejabat kurikulum (Zulfiandry dan Mahmud, 2017).

### **2.12 Petunjuk Teknis Penilaian**

Berdasarkan hasil wawancara yang telah dilakukan di Madrasah Aliyah Miftahul Huda Nambahdadi bersama Bapak Lukman Hakim, S.Pd.I selaku kepala sekolah, penulis mendapatkan data petunjuk teknis penilaian sebagai pedoman guna menentukan strategi penilaian dalam implementasi rapor kurikulum 2013. Adapun data yang terdapat pada petunjuk teknis penilaian sebagai berikut:

1. Bentuk penilaian pendidik

Bentuk penilaian pendidik merupakan bentuk penilaian yang harus diberikan guru kepada siswa yang meliputi nilai ulangan, nilai sikap, nilai pengetahuan dan keterampilan, nilai akhir semester.

2. Penilaian Kelas

Penilaian kelas merupakan suatu kegiatan guru yang berkaitan dengan pengambilan keputusan tentang pencapaian kompetensi atau hasil belajar

peserta didik yang mengikuti proses pembelajaran. Adapun rumus penentuan nilai siswa sebagai berikut:

$$RNT=(N1+N2+N3+N4+N5+N6+N7/7)$$

$$RNU=(NU1+NU2+NU3+NU4+NU5+NU6+NU7/7)$$

$$NA=(RNT+RNU+PTS+UAS/4)$$

**Keterangan:**

RNT = Rata Nilai Tugas

RNU = Rata Nilai Ulangan

PTS = Penilaian Tengah Semester

UAS = Ujian Akhir Semester

3. Predikat Penilaian

Predikat penilaian merupakan nilai yang mewakili capaian siswa secara kualitatif berdasarkan KKM (Kriteria Ketuntasan Minimal) tiap mata pelajaran. Kriteria penentuan penilaian siswa dapat dilihat pada Tabel 2.2.

**Tabel 2.2** Predikat Penilaian

<b>RENTANG PREDIKAT</b>			
A (Sangat Baik)	B (Baik)	C (Cukup)	D (Perlu Bimbingan)
>92-100	>83-91	75-82	<75



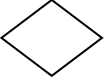


**2.13 Unified Modelling Language (UML)**

*Unified Modelling Language* (UML) merupakan suatu bahasa yang menjadi standar dalam visualisasi, merancang dan mendokumentasikan perangkat lunak. Dengan menggunakan UML maka kita dapat membuat model rancangan perangkat lunak yang nantinya akan dijalankan di beberapa sistem operasi. UML mendefinisikan sekumpulan bentuk notasi yang menggambarkan berbagai diagram dan setiap bentuk memiliki makna tertentu (Sugiarti, 2018).

## 1. Activity Diagram

Activity Diagram merupakan gambaran alur aktivitas di dalam sebuah sistem yang akan dibuat, memberikan gambaran bagaimana setiap aliran dimulai, keputusan yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut simbol-simbol yang ada pada *class diagram* dapat dilihat pada Tabel 2.3.

**Tabel 2.3** Simbol *Activity Diagram*

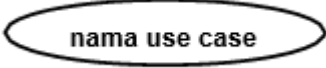



Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

## 2. Use Case Diagram



Use Case Diagram merupakan gambaran fungsi sistem dari perspektif pengguna dengan cara yang mudah dipahami. Diagram ini menggambarkan kumpulan kasus penggunaan, aktor, dan hubungan mereka. Use case adalah

hubungan antara fungsionalitas sistem dan aktor internal/eksternal sistem. Berikut simbol-simbol yang ada pada *use case* diagram dapat dilihat pada Tabel 2.4.

**Tabel 2.4** Simbol *Use Case* Diagram

Simbol	Deskripsi
<p data-bbox="448 528 572 562"><i>Use Case</i></p> 	<p data-bbox="722 528 1362 779">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p data-bbox="435 880 588 913">Aktor/<i>actor</i></p>  <p data-bbox="432 1263 592 1296">Nama Aktor</p>	<p data-bbox="722 880 1362 1240">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>
<p data-bbox="379 1377 644 1411"><i>Asosiasi/association</i></p> 	<p data-bbox="722 1377 1362 1518">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p data-bbox="336 1590 691 1624"><i>Generalisasi/generalization</i></p> 	<p data-bbox="722 1590 1362 1787">Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>

**Tabel 2.4** Simbol *Use Case* Diagram (lanjutan)

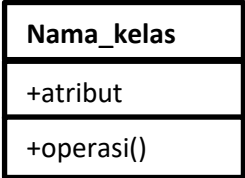


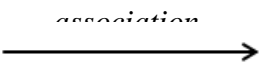

Simbol	Deskripsi
<p data-bbox="408 450 616 483">Ekstensi/<i>extend</i></p> <p data-bbox="432 674 592 707">&lt;&lt;extend&gt;&gt;</p> 	<p data-bbox="735 383 1370 853">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan</p>
<p data-bbox="331 898 692 931">Menggunakan/<i>Include/uses</i></p> <p data-bbox="427 1189 596 1223">&lt;&lt;include&gt;&gt;</p> 	<p data-bbox="735 898 1370 1144">Relasi <i>use case</i> 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>

### 3. *Class* Diagram



*Class* Diagram digunakan untuk menampilkan kelas-kelas dan paket-paket di dalam sistem. *Class* Diagram memberikan gambaran sistem secara statis. *Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem Berikut simbol-simbol yang ada pada *class* diagram dapat dilihat pada Tabel 2.5.



**Tabel 2.5** Simbol *Class* Diagram

Simbol	Deskripsi
<p>Kelas</p> 	<p>Kelas pada struktur system</p>
<p>Antarmuka/<i>Interfaces</i></p>  <p><b>Nama_interfaces</b></p>	<p>Sama dengan konsep <i>interfaces</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi/<i>association</i></p> 	<p>Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>.</p>
<p>Asosiasi berarah/<i>directed association</i></p> 	<p>Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i></p>
<p>Generalisasi</p> 	<p>Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)</p>

**Tabel 2.5** Simbol *Class Diagram* (lanjutan)

<b>Simbol</b>	<b>Deskripsi</b>
Kebergantungan/ <i>dependency</i> 	Kebergantungan antarkelas
Agregasi/ <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )