

## BAB II LANDASAN TEORI

### 2.1. Tinjauan Pustaka

Pada penelitian ini, penulis melakukan tinjauan pustaka pada penelitian sebelumnya. Sebagai bahan pendukung penelitian yang sedang dilakukan sekarang. Peneliti telah mengumpulkan beberapa tinjauan pustaka yang dapat dilihat pada tabel 2.1.

**Tabel 2. 1** Tinjauan Literatur

| No. 1             |  |
|-------------------|--|
| Penulis (Tahun)   | Amrin, Mita Diah Larasati, Irawan Satriadi (2020)  |
| Judul             | Model Waterfall Untuk Pengembangan Sistem Informasi Pengolahan Nilai Pada SMP Kartika XI-3 Jakarta Timur   |
| Metode Penelitian | <i>Waterfall</i>   |
| Kekurangan        | Pada SMP Kartika XI-3 Jakarta Timur sistem pengolahan nilai siswa masih dilakukan secara manual dan belum terkomputerisasi.  |
| Kelebihan         | Dengan adanya sistem pengolahan data nilai berbasis <i>web</i> dan sudah terkomputerisasi dapat membantu mengurangi kesalahan yang terjadi, dan meningkatkan kinerja Guru.   |
| Hasil             | Sistem informasi pengolahan nilai berbasis <i>web</i> pada SMP Kartika XI-3 Jakarta Timur yang sudah terkomputerisasi ini dapat membantu mengurangi kesalahan yang terjadi, dapat meningkatkan kinerja pegawai terutama dalam pengolahan data guru, data siswa dan pengolahan nilai siswa. |

**Tabel 2. 2** Tinjauan Literatur (Lanjutan)

| <b>No. 2</b>      |   |
|-------------------|---|
| Penulis (Tahun)   | Desty Dwi Pertiwi, Rohmat Taufiq (2020)   |
| Judul             | Analisis dan Desain Sistem Informasi Pengolahan Nilai Siswa di SMK Avicena Rajeg  |
| Metode Penelitian | Wawancara, Observasi, Kepustakaan dan <i>Unified Modeling Language</i> (UML)  |
| Kekurangan        | Pada SMK Avicena Rajeg proses pengelolaan nilai masih dilakukan dengan cara menggunakan <i>Microsoft Excel</i> belum adanya aplikasi khusus, sehingga banyak menyita waktu.   |
| Kelebihan         | Setelah diterapkannya sistem informasi pengolahan nilai pada SMK Avicena Rajeg dapat memudahkan Guru dalam mengelola nilai dan Kepala Sekolah dapat <i>me-review</i> laporan nilai.   |
| Hasil             | Perancangan dari sistem informasi pengolahan nilai pada SMK Avicena Rajeg memudahkan Guru dan Wali Kelas dalam melakukan proses pengolahan nilai siswa serta memudahkan Guru maupun Kepala Sekolah untuk <i>me-review</i> laporan pengolahan nilai, selain itu, siswa dalam melihat pengolahan nilai secara <i>online</i> . |

**Tabel 2. 3** Tinjauan Literatur (Lanjutan)

| <b>No. 3</b>      |  |
|-------------------|--|
| Penulis (Tahun)   | Dimas Aryo Anggoro, Yanuar Eko Adi Lukmana (2019)  |
| Judul             | Sistem Informasi Pengelolaan Data Nilai Siswa Pada SD Negeri Jambangan 1 Kabupaten Ngawi   |
| Metode Penelitian | <i>Waterfall</i>   |
| Kekurangan        | Prosedur pengolahan data nilai siswa disekolah ini masih dilakukan secara konvensional. Guru menyampaikan nilai siswa kepada wali kelas dalam bentuk lembaran kertas kemudian Guru wali kelas menuliskan hasil nilai ke dalam rapor, tentu hal tersebut menyita waktu yang cukup banyak. |
| Kelebihan         | setelah diterapkannya sistem informasi pengelolaan data nilai siswa pada SD Negeri 1 Jambangan dapat membantu pihak sekolah dalam mengelola data nilai siswa. Selain itu sistem ini dapat dioperasikan dengan mudah.   |
| Hasil             | Kesimpulan yang didapat, sistem ini dapat membantu pihak sekolah dalam mengelola data nilai siswa. Selain itu sistem ini dapat dioperasikan dengan mudah sehingga dapat mempercepat pihak sekolah dalam pengelolaan data nilai siswa.  |

Tabel 2. 4 Tinjauan Literatur (Lanjutan)

| No. 4             |   |
|-------------------|---|
| Penulis (Tahun)   | Desak Made Dwi Utami Putra, Mohamad Ariasidi (2019)   |
| Judul             | Rancang Bangun Sistem Informasi Pengolahan Nilai Rapor Berbasis Web pada SMK Negeri 1 Kotaraja  |
| Metode Penelitian | <i>Waterfall</i>  |
| Kekurangan        | Berdasarkan hasil wawancara dapat disimpulkan SMK Negeri 1 Kotaraja masih menggunakan cara konvensional dalam pengolahan nilai, dengan cara tersebut rentan terhadap masalah, memerlukan banyak waktu dan tenaga dalam melakukan pengolahan nilai.  |
| Kelebihan         | Setelah menerapkan pengolahan nilai rapor berbasis web, dan diuji menggunakan <i>blackbox testing</i> maka sistem dikatakan sesuai dengan rancangan dan berjalan dengan baik.   |
| Hasil             | Berdasarkan hasil dan pembahasan, sistem ini dirancang mulai dari tahap mengidentifikasi masalah, kemudian mengumpulkan data dan mengelolanya, tahap berikutnya adalah analisis sistem yang dibuat, sistem ini dirancang dengan menggunakan bahasa pemrograman <i>PHP</i> dan <i>MySQL</i> sebagai <i>datasenya</i> . Setelah dilakukan pengujian menggunakan <i>blackbox testing</i> . Maka sistem dikatakan sesuai dengan rancangan dan berjalan dengan baik. |

**Tabel 2. 5** Tinjauan Literatur (Lanjutan)

| <b>No. 5</b>      |   |
|-------------------|---|
| Penulis (Tahun)   | Aziz Setyawan, Ahmad Fauzi, Wandy Ubleeuw, Pas Mahyu (2019)   |
| Judul             | Sistem Pengolahan Data Nilai Berbasis Web pada Sekolah Menengah Pertama (SMP) Karel Sadsuitubun Langgur   |
| Metode Penelitian | <i>Waterfall</i>  |
| Kekurangan        | Proses pengolahan nilai siswa pada SMP Karel Sadsuitubun Langgur masih belum optimal dan masih menggunakan <i>Microsoft Excel</i> yang belum terintegrasi dengan <i>database</i> . Menyebabkan Guru, Siswa dan Orang tua siswa harus datang ke sekolah terlebih dahulu untuk mendapatkan informasi akademik.  |
| Kelebihan         | Setelah diterapkannya sistem pengolahan data nilai berbasis <i>web</i> ini dapat mempermudah kinerja Guru khususnya wali kelas dalam mencetak raport siswa.   |
| Hasil             | Sistem informasi pengolahan data nilai berbasis <i>web</i> ini dibangun dengan memiliki hak akses yaitu <i>admin</i> dan <i>user</i> . Siswa dan guru sebagai <i>user</i> dapat mengakses informasi data nilai secara <i>online</i> melalui media <i>browser internet</i> . Dengan adanya sistem informasi pengolahan nilai dapat mempermudah kinerja guru khusus nya wali kelas dalam mencetak raport siswa. |

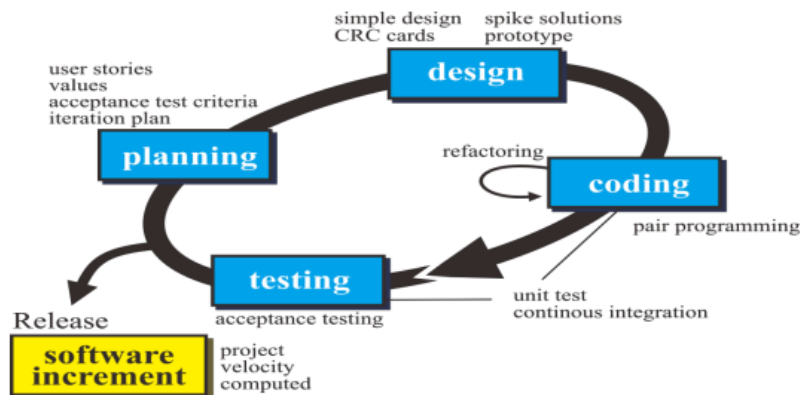
Kesimpulan yang dapat diambil dari tinjauan literatur di atas dengan penelitian adalah perbedaan pada metode pengembangan sistem yang dipakai. Pada penelitian yang akan diteliti, akan menggunakan metode pengembangan *Extreme Programming*

(XP) sebagai metode pengembangan penelitiannya, karena selain dapat digunakan untuk pengembangan aplikasi *web* sederhana, model ini juga dapat digunakan untuk mengembangkan pembuatan *prototype* awal atau untuk mempersiapkan tes modul yang dapat diimplementasikan pada desain halaman *web*. Sedangkan untuk pengujian sistem penulis menggunakan pengujian ISO 25010.

## 2.2. *Extreme Programming (XP)*

Menurut (Suryantara, 2018) *Extreme Programming* adalah metode pengembangan perangkat lunak yang bertujuan untuk meningkatkan kualitas perangkat lunak dan daya tanggap terhadap perubahan kebutuhan pelanggan.

Dalam metode ini, pengembang harus mengikuti empat langkah berikut untuk menyelesaikan pekerjaan pengembangan :



**Gambar 2. 1** Tahapan Metode *Extreme Programming*

## 1. Perencanaan (*Planning*)

Tahapan ini merupakan tahapan yang diperlukan sebelum pengembang membuat sistem. Tahap ini sangat penting karena diperlukan perencanaan atau analisa kebutuhan pengguna pada saat pembuatan sistem. Pengembang memutuskan bagaimana membangun hasil cerita pengguna sesuai dengan janji yang disepakati, adapun *story-story* yang dilakukan dengan cara :

1. Pengguna menceritakan apa permasalahan pada sistem yang digunakan dan sistem seperti apa yang akan dibangun.
2. Berdasarkan hasil cerita pengguna maka peneliti menentukan poin pada bagian *value* untuk memutuskan apa saja yang akan dibangun.
3. Dari hasil kesepakatan tersebut maka peneliti menentukan *acceptance test criteria* yaitu menentukan kriteria apa saja yang nantinya sebagai acuan terhadap sistem yang akan diuji.
4. Sehingga hasil peneliti menyimpulkan berapa kali akan dilakukan *release* dan perbaikan pada tahap *iteration plan* merencanakan berapa kali akan dilakukan uji terhadap sistem yang dibangun.

## 2. Perancangan (*Design*)

Setelah tahap perencanaan selesai, tahap selanjutnya adalah desain. Pada tahap ini pengembang merancang dengan membuat model yang diawali dengan pemodelan sistem, dilanjutkan dengan pemodelan arsitektur dan pemodelan basis data untuk memberikan gambaran tentang sistem yang akan dibangun.

1. *Simple Design* adalah pengembang membangun perangkat lunak dengan desain yang sederhana. Dimulai dengan desain yang sederhana dilakukan menggunakan UML seperti *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.
2. *CRC Card* digunakan untuk mengenali dan mengatur *object oriented class* yang sesuai dengan pengembangan. Jika pada saat perancangan terdapat ketidaksesuaian maupun perbaikan maka akan dilakukan.
3. *Spike solution* jika pada saat praktiknya, desain yang dibuat sangatlah sulit. Maka *Extreme Programming* akan menggunakan *spike solution* dimana pembuatan desain akan dibuat langsung ketujuannya.
4. *Prototype* adalah bagian perancangan berupa *user interface* dalam bentuk *wireframing* untuk mempermudah pengguna melihat desain sistem.

### **3. Pengkodean (*Coding*)**

Pada proses pengkodean peneliti menyesuaikan terhadap *story* pengguna sehingga sistem yang dibangun sesuai, proses pengkodean yang dilakukan yaitu :

1. *Pair Programming* merupakan tahap sistem dibangun dengan bahasa pemrograman dan media penyimpanan yang telah disepakati.
2. *Refactory* merupakan tahapan yang dilakukan ketika terjadi ketidaksesuaian kode program sehingga dilakukan perbaikan guna mendapatkan hasil yang sesuai.



#### **4. Pengujian (*Testing*)**

Tahap pengujian dilakukan oleh pengguna sebagai *user* dengan melakukan uji sesuai dengan *acceptance test* yang telah ditentukan dan disetujui. Unit test yang telah dibuat fokus pada keseluruhan fitur dan fungsional sistem. Sehingga sistem dapat disimpulkan telah sesuai dan dapat dirilis.

#### **2.3. *Unified Modeling Language (UML)***





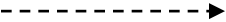
Menurut (Sukamto dan Shalahuddin, 2018), UML (*Unified Modelling Language*) merupakan bahasa standar yang banyak digunakan di industry untuk mendefinisikan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur pemrograman berorientasi objek. UML adalah bahasa visual yang digunakan untuk memodelkan dan mendukung teks. UML hanya digunakan untuk pemodelan. Oleh karena itu, pengguna UML tidak terbatas pada metode tertentu, meskipun UML digunakan terutama dalam metode berorientasi objek.

Terdapat tiga diagram dalam UML yang memiliki fungsi masing-masing yaitu :

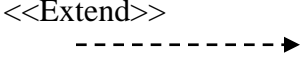
##### **2.3.1. *Use Case Diagram***

Menurut (Sukamto dan Shalahuddin, 2018), *Use Case* adalah model perilaku (*behaviour*) sistem yang akan dibuat. *Use case* menggambarkan interaksi antara satu atau lebih aktor dan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa yang ada di sistem dan siapa yang memiliki akses ke fungsi tersebut.

Tabel 2. 6 Simbol *Use Case Diagram*

| No | Simbol   | Keterangan  |
|----|--|---|
| 1  |                   | <i>Use Case</i> : Fungsionalitas yang disediakan system sebagai unit-unit atau actor, yang biasanya dinyatakan dengan kata kerja diawal frase nama <i>use case</i> .  |
| 2  |                   | Orang, atau proses yang akan berinteraksi dengan sistem informasi yang akan dibuat, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang, yang biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |
| 3  |                 | Asosiasi merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.   |
| 4  |                 | Generalisasi merupakan hubungan umum-khusus antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum.  |
| 5  | <<Include>><br> | <<Include>> simbol ini menunjukkan bahwa <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.  |

**Tabel 2. 7** Simbol *Use Case Diagram* (Lanjutan)

|   |   |  |
|---|---|--|
| 6 |  | <<Extend>> merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu. |
|---|---|--|

Sukamto & Shalahuddin (2018)

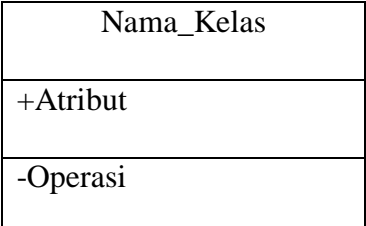
### 2.3.2. *Class Diagram*

Menurut (Sukamto dan Shalahuddin, 2018), *Class Diagram* menggambarkan struktur sistem dengan mendefinisikan kelas-kelas yang diperlukan untuk membangun sistem. Kelas memiliki apa yang disebut atribut, *method* atau operasi.

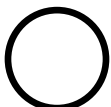


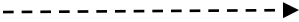


Berikut adalah penjelasan dari atribut dan *method* :

1. Atribut merupakan *variable* yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi yang dimiliki oleh suatu kelas.

**Tabel 2. 8** Simbol *Class Diagram*

| No | Simbol  | Keterangan                        |
|----|---|-----------------------------------|
| 1  |  | Kelas yang terdapat pada stuktur. |

Tabel 2. 9 Simbol *Class Diagram* (Lanjutan)



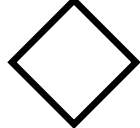

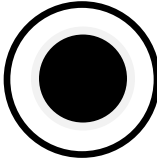
|   |   |   |
|---|---|---|
| 2 | <p><i>Interface</i> (Antar muka)</p>             | <i>Interface</i> sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.  |
| 3 | <p>Asosiasi Berarah</p>                          | Asosiasi berarah, relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan simbol. |
| 4 | <p>Generalisasi</p>                              | Generalisasi, relasi antar kelas dengan makna generalisasi –spesialisasi (umum-khusus).   |
| 5 | <p><i>Dependency</i><br/>(Ketergantungan)</p>  | Relasi antar kelas dengan makna ketergantungan antar kelas.   |
| 6 | <p>Agregasi</p>                                | Agregasi, relasi antar kelas dengan makna semua bagian.   |
| 7 | <p>Asosiasi</p>                                | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .  |

Sukamto dan Shalahuddin (2018)


### 2.3.3. Activity Diagram

Menurut (Sukanto dan Shalahuddin, 2018), *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau alur aktivitas sistem atau proses bisnis atau menu diperangkat lunak. *Activity diagram* menggambarkan aktivitas sistem, bukan aktivitas yang dilakukan oleh aktor. Berikut simbol yang ada pada *activity digaram* :

**Tabel 2. 10** Simbol *Activity Diagram*

| No | Simbol  | Keterangan   |
|----|---|--|
| 1  |    | Status awal, sebuah diagram aktivitas yang memiliki status awal.                               |
| 2  |  | Aktivitas, yang dilakukan oleh sistem, yang biasanya diawali dengan kata kerja.                |
| 3  |  | <i>Decision</i> , asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.      |
| 4  |  | <i>Join</i> , asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 5  |  | Status akhir, sebuah diagram aktivitas memiliki sebuah status akhir                            |

**Tabel 2. 11** Simbol *Activity Diagram* (Lanjutan)

|   |   |  |
|---|---|--|
| 6 |  | <i>Swimlane</i> , memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
|---|---|--|

Sukamto dan Shalahuddin (2018)

#### 2.4. *Laravel*

*Laravel* adalah sebuah *framework PHP* bersifat *open source* yang dirilis dibawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*). *Laravel* adalah pengembangan *website* berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pemeliharaan. *Laravel* merupakan *framework PHP* terbaik yang dikembangkan oleh Taylor Otwell.

Sebagai sebuah *framework PHP*, *laravel* hadir sebagai *platform* pengembangan *web open source*. Ekspresi *Laravel* dan sintaks yang elegan juga sangat menarik. ini dirancang khusus untuk menyederhanakan dan mempercepat proses pengembangan web (Saputra, 2018).

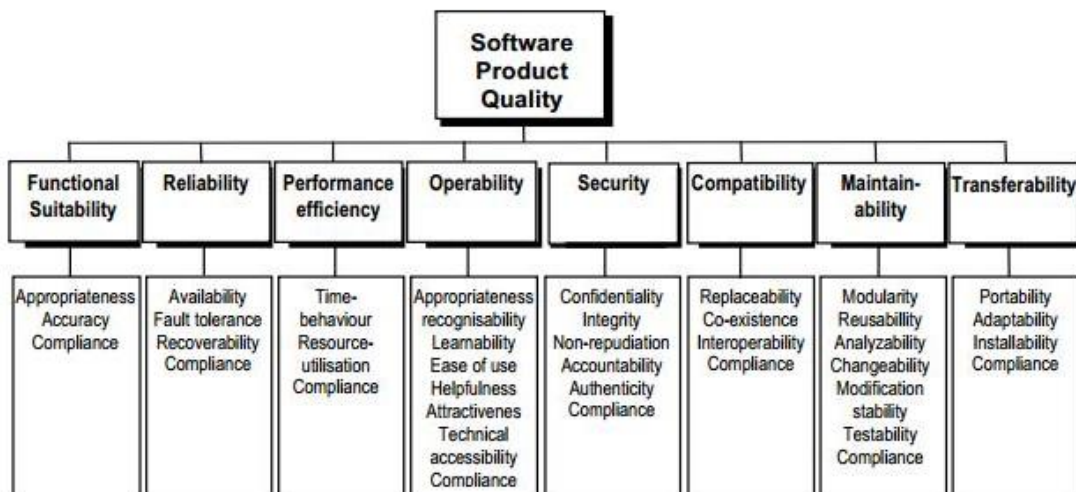
#### 2.5. **Blackbox Testing**

Menurut (Sutarno, 2018), *black box testing* adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas dari sisi pengguna tanpa mengetahui detail implementasi atau kode program dan struktur internal perangkat lunak. Metode ini bertujuan untuk menemukan *bug* atau kesalahan dalam perangkat

lunak yang dapat mempengaruhi pengalaman pengguna. Hal ini dilakukan untuk memastikan bahwa perangkat lunak dapat berfungsi sesuai dengan spesifikasi dan kebutuhan pengguna.

## 2.6. Pengujian ISO 25010

Menurut (Harun, 2020), ISO 25010 terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat statis perangkat lunak dan sifat dinamis sistem komputer, seperti yang ditunjukkan pada gambar dibawah ini :



**Gambar 2. 2 ISO 25010**

Berdasarkan gambar diatas, dapat dijelaskan mengenai delapan karakteristik tersebut, 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 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.
4. *Usability*, merupakan tingkat dimana sistem atau produk mudah dimengerti, mudah dipakai dan menarik untuk digunakan.
5. *Security*, merupakan tingkat dimana 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 sistem atau produk dapat dimodifikasi, yang meliputi perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.
8. *Portability*, merupakan tingkat dimana sistem atau produk dapat dipindahkan dari satu ruang ke ruang lainnya.

Penelitian ini berfokus pada dua aspek pengujian berdasarkan ISO 25010 yaitu aspek pengujian *functionality*, dan pengujian *usability*. Alasan melakukan pengujian tersebut adalah untuk menguji apakah sistem yang telah dibuat mudah dimengerti dan mudah digunakan oleh pengguna (*usability*). Setelah pengujian *usability* maka



pengujian selanjutnya yaitu fungsi dari menu yang terdapat pada sistem, apakah berfungsi sesuai dengan yang telah diinginkan atau tidak (*functionality*).