

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan pustaka**

Tinjauan pustaka merujuk pada jurnal dan penelitian terdahulu yang memiliki relevansi pada penelitian ini. Penelitian terdahulu penting untuk membangun dasar teoritis dan membantu memilih metode yang sesuai dengan penelitian. Terdapat 10 penelitian terdahulu yang digunakan sebagai referensi, 10 penelitian tersebut membahas mengenai perancangan sistem menggunakan metode pengembangan perangkat lunak seperti *Rapid Application Development*, *Extreme Programming*, *Waterfall* dan *Scrum*. Topik dari penelitian terdahulu menjelaskan mengenai seberapa efektif peralihan dari proses bisnis yang belum terkomputerisasi menjadi sebuah sistem dan keberhasilan dari penerapan metode pengembangan dalam merancang sistem. Beberapa penelitian terdahulu yang relevan dengan penelitian ini telah diidentifikasi persamaan, perbedaan, dan hasil penelitian dalam Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
1	Rancang Bangun Sistem Tracking Jasa Laundry Sepatu di Clinix Shoes Care Berbasis Website. Febriyan Saragih, Yulia Agustina Dalimunthe, Husni Lubis 2021[9].	Memilik tema penelitian yang selaras yaitu jasa cuci sepatu	Perancangan sistem dilakukan dengan memanfaatkan kerangka kerja CodeIgniter dan menerapkan metode waterfall	Tidak terdapat menu untuk login sebagai customer	Sistem yang dikembangkan mampu menyelesaikan permasalahan yang terkait dengan transaksi laundry dan mendukung efisiensi dalam pengolahan informasi transaksi laundry dengan cepat dan sederhana.	Sistem informasi dibangun untuk mempermudah admin dalam pendataan customer dan melakukan <i>tracking</i> produk
2	Sistem Informasi Pelayanan Jasa Laundry Sepatu Benga Cleaning Shoes Store berbasis Web Silviana Fitria Hidarini, Muhtajuddin Danny, Asep Muhidin 2021 [10].	Pengujian sistem menggunakan metode <i>Blackbox Testing</i> .	Pengembangan sistem menggunakan pendekatan waterfall	Sistem memiliki batasan hanya dapat digunakan oleh admin dan hal ini tidak membantu pelanggan toko	Sistem informasi ini memungkinkan adanya pelayanan online kepada konsumen, meminimalisir kesalahan data, serta memudahkan proses pencatatan dan pembuatan laporan secara periodik.	Sistem informasi ini membantu melakukan pencatatan pada pendataan seperti daftar dan pendapatan setiap transaksi agar menjadi lebih terorganisir
3	Rancang Bangun Sistem Tracer study UNUSIA Berbasis Web Menggunakan Metode	Melakukan penelitian terkait penerapan metode Rapid application	Penelitian ini menguraikan penerapan metode RAD dalam proses	Penelitian tersebut tidak memberikan penjelasan mengenai	Dapat menjadi bahan acuan dalam pengembangan website terutama	Penelitian ini menunjukkan bahwa sistem yang dikembangkan

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
	Rapid Application Development [11]	development (RAD), pemodelan sistem menggunakan (UML)	pengembangan website yang bertujuan sebagai instrumen tracing alumni dan pusat karir	<i>framework</i> pemrograman yang digunakan dalam tahapan pengembangan website.	berasal dari metode pengujian dan perancangan sistem	mampu beroperasi sesuai dengan harapan pengguna sistem
4	Sistem Informasi Pelayanan Jasa Laundry Sepatu Queen Shoes Cleaning [12].	Perancangan sistem pada jasa laundry sepatu menggunakan bahasa pemrograman PHP dan pengujian sistem menggunakan pendekatan <i>Blackbox Testing</i>	Pengembangan sistem menggunakan pendekatan waterfall	Penelitian ini tidak menampilkan visualiasi website sebagai hasil perancangan yang dilakukan	Pengembangan sistem informasi berbasis web memiliki dapat meningkatkan efektifitas kinerja operasional bisnis jasa laundry sepatu.	Pembuatan sistem dapat menjalankan transaksi, mengelola data dan arsip dengan terorganisir melalui situs web tersebut, serta menyediakan kemudahan bagi pelanggan yang ingin mencuci sepatu tanpa perlu mengunjungi toko secara langsung.
5.	Metode <i>Extreme Programming</i> Dalam Pengembangan Aplikasi Legalisir Online Berbasis Web Service [13].	Metode <i>Extreme Programming (XP)</i> digunakan untuk merancang sistem, pengujian dilakukan dengan pendekatan <i>Blackbox Testing</i> dan <i>System usability scale</i>	Penelitian ini menerapkan metode <i>Extreme Programming (XP)</i> pengembangan sistem legalisir online	Perancangan difokuskan pada fungsionalitas sehingga tampilan pada sistem menjadi kurang menarik	Memberikan pemahaman yang lebih mendalam mengenai metode <i>Extreme Programming</i> dan pengujian sistem menggunakan <i>Blackbox Testing</i>	Peneliti berhasil menerapkan metode pengembangan <i>Extreme Programming (XP)</i> , hasil pengujian menunjukkan bahwa aplikasi berjalan dengan lancar dan mendapatkan standar kelayakan sistem "layak"

No	Judul	Compare	Contrast	Criticize	Synthesis	Summarize
6	<i>Development of Warehouse Management System Using Extreme Programming</i> [14].	Melakukan penelitian terkait penerapan metode (XP) dalam membuat website <i>Warehouse Management System</i>	Dalam melakukan Perancangan sistem menggunakan visual basic 2013	Dalam penelitian ini, antarmuka pengguna aplikasi tidak dijelaskan dengan mendetail.	Dapat memberikan pemahaman yang lebih mendalam mengenai penerapan metode <i>Extreme Programming</i> dalam perancangan sistem	Menciptakan sebuah sistem dengan kemampuan mendukung proses bisnis, terutama dalam mencatat stok barang di gudang.
7	Implementasi <i>Extreme Programming</i> Pada Sistem Informasi Program Kreativitas Mahasiswa Berbasis Web[15].	Penerapan Metode <i>Extreme Programming (XP)</i> untuk merancang sistem informasi dan pengujian dilakukan dengan pendekatan <i>Blacbox Testing</i>	Metode <i>Extreme Programming (XP)</i> digunakan untuk pengembangan sistem informasi program kreativitas mahasiswa	Penelitian ini tidak memberikan penjelasan mengenai <i>framework</i> pemrograman yang digunakan dalam tahapan pengembangan website.	Penggunaan metode XP dapat memberikan fleksibilitas dalam pengembangan sistem.	Dengan menggunakan metode XP, penelitian ini berhasil merancang website yang dapat memudahkan pengguna menjalankan proses bisnis PKM.
8	Perancangan Sistem Informasi Warga Berbasis Web Pada Perumahan Pesona Wibawa Praja Dengan Metode Scrum[16]	Perancangan sistem informasi pendataan menggunakan bahasa pemrograman PHP dan database mysql	Penelitian ini menerapkan metode pengembangan agilile scrum dalam pengembangan sistem pendataan warga	Penelitian ini tidak memberikan penjelasan mengenai <i>framework</i> yang dipakai dalam melakukan perancangan sistem	Memberikan pemahaman mengenai efektifitas dan efisiensi sistem yang terkomputerisasi dalam melakukan pendataan	Artikel menekankan manfaat sistem dalam hal efisiensi dan efektifitasn pendataan warga bagi RW setempat untuk memaksimalkan pelayanan

No	Judul	<i>Compare</i>	<i>Contrast</i>	<i>Criticize</i>	<i>Synthesis</i>	<i>Summarize</i>
9	<i>Developing BuJel Application Using Extreme Programming (XP)Methodology [17].</i>	Penerapan Metode <i>Extreme Programming (XP)</i> dalam merancang aplikasi dan melakukan pengujian dengan pendekatan <i>Blackbox Testing</i>	Penggunaan bahasa python dalam perancangan aplikasi dan Sqlite untuk pengelolaan database	Penelitian ini tidak menyebutkan adanya batasan atau tantangan yang dihadapi selama proses pengembangan .	Menjadi bahan referensi dalam pengembangan aplikasi terutama berasal dari metodologi yang digunakan	Artikel penelitian ini memaparkan pengembangan aplikasi BuJel menggunakan metodologi <i>Extreme Programming</i> untuk mengatasi masalah limbah minyak masak (WCO) di Kecamatan Pancoran, Jakarta Selatan.
10	<i>An Extreme Programming Approach for Instructor Performance Evaluation System Development [18].</i>	menerapkan Metode <i>Extreme Programming (XP)</i> dalam mengembangkan sistem	Membahas mengenai perbandingan seluruh pendekatan pengembangan sistem pada metode agile	Tidak menjelaskan metode pengujian sistem yang digunakan	Dapat memberikan pemahaman yang lebih mendalam mengenai penerapan metode <i>Extreme Programming</i>	Pengujian yang dilakukan ini menunjukkan sistem telah terbukti efektif

Berdasarkan 10 penelitian sebelumnya yang telah diambil sebagai referensi pada penelitian ini. Terdapat 1 penelitian yang di jadikan sebagai acuan utama yaitu penelitian yang berjudul Sistem Informasi Pelayanan Jasa Laundry Sepatu Benga Cleaning Shoes Store berbasis Web yang ditulis oleh Silviana Fitria Hidarini, Muhtajuddin Danny, Asep Muhidin. Penelitian tersebut memiliki objek penelitian yang sama yakni, meneliti mengenai jasa *laundry* Sepatu, namun terdapat perbedaan penelitian yang terletak pada sistem pengembangan dalam penelitian tersebut mengadopsi metode waterfall dan tidak melibatkan tahap pengujian sistem.

## **2.2 Landasan Teori**

Landasan teori digunakan untuk menyertakan teori yang relevan untuk mendukung penelitian

### **2.2.1 Rancang Bangun**

Kegiatan rancang melibatkan tujuan merancang sistem guna mengatasi masalah yang dihadapi oleh pengguna, berdasarkan pemilihan sistem terbaik. Istilah "bangun" dalam konteks "pembangunan" merujuk pada tindakan menciptakan sistem baru atau menggantikan serta memperbaiki sistem yang sudah ada, baik secara keseluruhan maupun sebagian[19].Kegiatan rancang bangun meliputi perancangan, mengembangkan, dan mengatur komponen terpisah menjadi satu bagian yang lengkap. Rancang bangun melibatkan aktivitas mengubah hasil analisis menjadi sebuah aplikasi dan selanjutnya membuat atau memperbaiki aplikasi yang telah ada[20].

Berdasarkan penjelasan yang diberikan disimpulkan tahap rancang bangun termasuk peran yang penting pada pengembangan sistem setelah melakukan analisis. Tahap ini, dilakukan identifikasi kebutuhan perencanaan untuk membentuk suatu sistem yang lengkap dan berfungsi dengan baik. Proses ini mencakup pengGambaran, perencanaan, dan pengaturan komponen hardware dan software. Rancang bangun tidak hanya terbatas pada pembuatan sistem baru, tetapi juga melibatkan perbaikan pada sistem yang sudah ada.

### **2.2.2 Sistem Informasi**

Sistem informasi merupakan suatu rangkaian komponen di dalam suatu organisasi yang memiliki tujuan utama untuk menyediakan informasi[21]. Komponen pada sistem informasi terdiri dari kombinasi orang, fasilitas, teknologi, media, prosedur, dan pengendalian. Sistem ini bertujuan untuk mendapatkan jalur komunikasi penting, memproses jenis transaksi rutin tertentu, memberikan informasi kepada pemangku kepentingan dan pihak lain terkait dengan peristiwa internal dan eksternal yang signifikan. Selain itu, sistem ini menyediakan dasar informasi untuk mendukung pengambilan keputusan yang cerdas[22].

Berdasarkan penjelasan yang diberikan disimpulkan bahwa sistem Informasi adalah sebuah istem terstruktur yang bertugas untuk mengolah informasi yang memiliki nilai, dengan tujuan khusus, dan dapat diterima dengan baik oleh penerima sehingga mencapai tujuan yang diinginkan.

### **2.2.3 Bilashoe**

Bilashoe merupakan usaha yang bergerak dalam jasa laundry dan reparasi sepatu dengan memberikan pelayanan seperti deep clean treatement, swap sole, dan leather paint. Pada proses pengelolaan data transaksi, Bilashoe memiliki empat tahapan dalam pengelolaannya. Gambar 2.1 menunjukkan keempat tahapan proses pengelolaan data transaksi secara mendetail.



Gambar 2. 1 Proses transaksi Bilashoe

#### 2.2.4 Website

Sebuah *website* merupakan kumpulan halaman berisikan informasi serta dapat diakses melalui internet. Dalam konteks lain, *website* merupakan suatu rangkaian halaman web yang saling terhubung, berisikan berbagai informasi seperti data, teks, Gambar, animasi, video, atau kombinasi dari semua komponen tersebut, dan biasanya dibuat untuk keperluan personal atau organisasi[23]. Mendapati kesimpulan bahwa *website* merupakan media yang bisa digunakan untuk berbagai keperluan serta berisi berbagai informasi. *Website* dapat menjadi sarana untuk berbagi informasi, mempromosikan produk atau jasa, atau melayani pelanggan.

#### 2.2.5 PHP

*PHP* merupakan bahasa pemrograman *server-side* yang dirancang untuk pengembangan web yang bersifat *open source*. *PHP* disebut sebagai bahasa pemrograman *server-side* karena eksekusinya terjadi pada komputer server. Keadaan ini berbeda dengan bahasa pemrograman *client-side* seperti *JavaScript* yang dieksekusi pada *web browser* (client). Bahasa pemrograman *PHP* digunakan untuk memungkinkan komunikasi antara *website* dan *server database*[24].



### 2.2.6 *Laravel*

*Laravel* merupakan kerangka kerja berbasis bahasa pemrograman *PHP*, berfungsi untuk mendukung pengembangan *website* agar mencapai hasil yang optimal. Dengan menggunakan *Laravel*, *website* yang dikembangkan dapat menjadi lebih dinamis[25]. *Laravel* bersifat *open source* dengan menggunakan konsep *MVC (Model-View-Controller)*. *Laravel* memiliki beberapa kelemahan, salah satunya adalah ukuran file yang cukup besar. Di dalam *Laravel* terdapat file default seperti *vendor* yang tidak dapat dihapus sembarangan, sehingga dapat membuat ukuran *website* yang dibuat menjadi cukup besar[26].

### 2.2.7 *My Structured Query Language (MySQL)*

*MySQL*, merupakan kependekan dari *My Structured Query Language*, *MySQL* bersifat *open-source*, menyediakan *source code* yang digunakan untuk pembuatannya. Perangkat lunak ini mampu mengelola basis data dengan cepat, menampung volume data yang besar, dan dapat diakses oleh banyak pengguna [27]. *MySQL* termasuk dalam kategori berbagai Sistem Manajemen Basis Data (*DBMS*) seperti *Oracle*, *MSSQL*, *PostgreSQL*, dan sebagainya. *MySQL* merupakan platform manajemen basis data dengan memanfaatkan *SQL* untuk pengelolaan data [28].

### 2.2.8 *Unified Modelling Language (UML)*

*UML* merupakan bahasa visual yang dipakai untuk memodelkan serta menggambarkan interaksi antara pengguna dengan sistem melalui penggunaan diagram. *UML* digunakan untuk mewakili desain sistem dalam pemrograman berorientasi objek. Jenis diagram pada *UML* yang digunakan untuk melakukan pemodelan terdiri dari *Use Case diagram*, *Class diagram*, *Activity diagram* serta *Sequence diagram*[29].





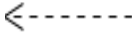
Beberapa jenis diagram yang umumnya dipergunakan dalam *Unified Modeling Language (UML)* di antaranya adalah sebagai berikut :

#### 1) *Use Case Diagram*

Serangkaian deskripsi terkait yang saling terhubung dan membentuk sistem secara terstruktur yang dieksekusi atau diawasi oleh suatu pihak yang terlibat, yang

disebut *actor*. *Usecase* digunakan untuk membentuk perilaku objek dalam suatu model dan diimplementasikan melalui suatu kerjasama[30]. Adapun simbol *Use Case* diagram dijelaskan pada Tabel 2.2







Tabel 2. 2 Simbol *Use Case* Diagram[30]

No	Simbol	Keterangan
1	<i>Use Case</i> 	MengGambarkan aktivitas yang disediakan oleh sistem yang saling berinteraksi antar unit atau aktor.
2	Aktor / <i>Actor</i> 	Actor merupakan pengguna sistem. Meskipun representasi simbolis dari aktor berupa Gambar orang, hal tersebut belum menjamin bahwa aktor tersebut benar-benar merupakan orang.
3	<i>Association</i> 	Merupakan garis hubung antara aktor dan <i>Use Case</i> , menandakan interaksi komunikasi di antara keduanya melalui pertukaran pesan
4	<i>Include</i>  <<include>>	Include merupakan notasi relasi dari suatu <i>Use Case</i> mencakup seluruh fungsi dari <i>Use Case</i> lainnya.
5	<i>Extend</i>  <<extend>>	MengGambarkan hubungan pada suatu <i>Use Case</i> yang menjadi tambahan fungsional dari <i>Use Case</i> lainnya jika suatu kondisi tertentu dipenuhi.

## 2) *Activity Diagram*

Diagram aktivitas mengilustrasikan proses kegiatan dalam sistem yang sedang dirancang. Diagram ini serupa dengan flowchart karena memodelkan aliran kerja dari satu kegiatan dengan kegiatan lainnya[30]. Adapun simbol *Activity diagram* dijelaskan oleh Tabel 2.3.

Tabel 2. 3 Simbol *Activity Diagram*[30]


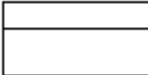

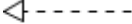


No	Simbol	Keterangan
1.	Initial 	Menyatakan tahap awal dari suatu kegiatan pada diagram
2.	<i>Activity Final</i> 	Menyatakan status akhir dari suatu aktivitas pada diagram
3.	Action 	MengGambarkan tindakan yang sedang dilakukan di dalam sistem
4.	<i>Activity</i> 	Menyatakan aktivitas yang dilakukan oleh sistem
5.	Decision 	Merupakan asosiasi percabangan yang menyatakan kondisi dari suatu proses
6.	Fork 	Merupakan percabangan dari satu alur aktivitas menjadi beberapa alur aktivitas

## 3) *Class Diagram*

Diagram kelas merupakan representasi struktural sistem dalam hal mendefinisikan diagram yang ingin dibuat dalam mengembangkan sistem. Kelas diagram mencakup atribut dan operasi dengan maksud memfasilitasi pembuat

program untuk membentuk korelasi hubungan antara perancangan desain dan sistem[31]. Terdapat simbol diagram kelas dijelaskan oleh Tabel 2.4.



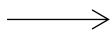
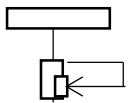



Tabel 2. 4 Simbol *Class Diagram*[30]

No	Simbol	Keterangan
1.	Generalization 	menunjukkan hubungan antara kelas induk (dan kelas anak
2.	<i>Class</i> 	Merujuk pada sekelompok objek yang memiliki sifat dan fungsi yang serupa.
3.	Association 	Asosiasi mengGambarkan hubungan antara dua kelas dan dapat memiliki arah
4.	Realization 	MengGambarkan implementasi dari sebuah antarmuka atau kelas abstrak.
5.	Dependency 	Menunjukkan perubahan pada satu kelas dapat memengaruhi kelas lainnya.
6.	Composition 	Menunjukkan satu kelas merupakan bagian integral dari kelas lain dan tidak bisa berdiri sendiri tanpa kelas tersebut.

#### 4) *Sequence Diagram*

*Sequence diagram* mengGambarkan urutan interaksi objek yang diatur dalam waktu secara berurutan dengan langkah-langkah yang terorganisir untuk mencapai tujuan sesuai dengan *Use Case*. Semua interaksi pesan dari *Use Case* yang telah dibuat sebelumnya harus tercakup dalam diagram urutan [31]. Adapun simbol diagram *Sequence* ditunjukkan oleh Tabel 2.5

Tabel 2. 5 Simbol *Sequence Diagram*[30]

No	Simbol	Keterangan
1.	Actor 	Actor merujuk pada entitas pengguna yang sedang berinteraksi dengan sistem
2.	LifeLine 	Memberikan Gambaran tempat mulai dan berakhirnya sebuah message
3.	Message 	Menunjukkan komunikasi atau pertukaran pesan antara objek-objek yang terlibat dalam suatu skenario atau <i>Use Case</i> .
4.	Self Message 	Menunjukkan pesan yang dikirim oleh objek kepada dirinya sendiri. Terdapat situasi di mana objek menginisiasi suatu aksi atau operasi pada dirinya sendiri, atau ketika objek membuat keputusan dan mengeksekusi sesuatu secara internal.
5.	Entity Class 	Menunjukkan adanya hubungan kegiatan yang sedang dilakukan oleh aktor
6.	Boundary Class 	Menunjukkan aktivitas dari penggunaan form
7.	Control Class 	Sebagai penghubung pada boundary dengan Tabel

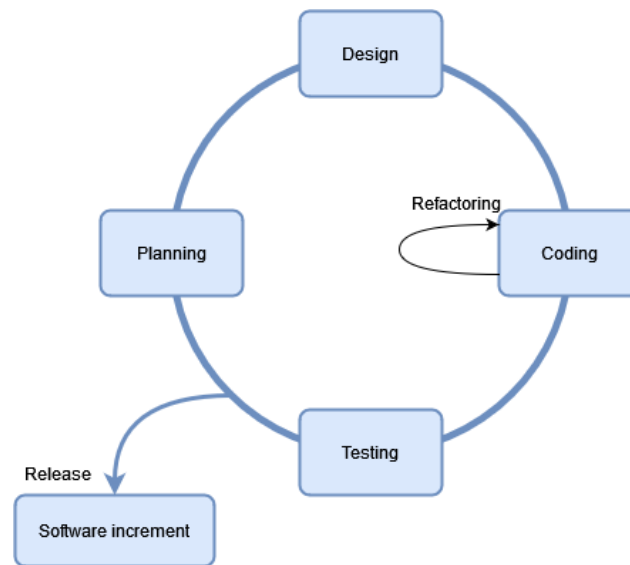
### 2.2.9 Blackbox Testing

Pengujian *black box* adalah metode pengujian fungsional tanpa memerlukan pengetahuan mengenai struktur internal program. Metode ini berguna untuk mengidentifikasi masalah seperti kesalahan fungsi, ketidaksesuaian antarmuka, dan

kecacatan fungsi lainnya[32]. Pengujian difokuskan pada spesifikasi fungsional perangkat lunak, yaitu apa yang dapat dilakukan oleh penguji. Dalam pengujian ini, menetapkan sejumlah kondisi input dan melaksanakan uji coba terhadap spesifikasi fungsional program. Pengujian *black box* memiliki kelebihan dan kelemahan dalam penerapannya. Salah satu kelebihannya adalah kemampuannya dalam menemukan aspek-aspek yang tidak memenuhi spesifikasi kebutuhan yang telah ditetapkan dalam pengembangan perangkat lunak. Namun, kelemahan dari pengujian *black box* terletak pada keterbatasan pengetahuan penguji mengenai perangkat lunak yang diuji, sehingga pengujian tidak dapat dilakukan secara menyeluruh[33].

#### **2.2.10 *Extreme Programming***

Metode *Extreme Programming* merupakan teknik pengembangan sistem yang termasuk dari salah satu metode *agile* yang ditemukan oleh seorang pakar *software engineering* yaitu Kent Beck pada tahun 1996. Metode ini memiliki prinsip – prinsip dasar meliputi *Communication, Courage, Simplicity, Feedback dan Quality work*. Prinsip-prinsip tersebut membuat metode *Extreme Programming* mampu untuk menciptakan lingkungan pengembangan perangkat lunak yang adaptif, responsif terhadap perubahan, dan berfokus pada kualitas. *Extreme Programming* memiliki empat tahapan dalam pelaksanaannya yaitu *planning, design, coding, dan testing*[34]. Gambar 2.2 menunjukkan tahapan metode *Extreme Programming (XP)*.



Gambar 2. 2 Metode *Extreme Programming* [34]

### 1) *Planning*

Tahap *planning* dalam metode *XP* melibatkan kolaborasi aktif antara tim pengembang dan pengguna. Bersama-sama, mereka menentukan fitur-fitur yang harus diimplementasikan dalam proyek. Perencanaan melibatkan penentuan prioritas fitur dan pembagian pekerjaan ke dalam iterasi yang pendek. *Planning* di *XP* bersifat adaptif dan dapat disesuaikan seiring berjalannya proyek untuk mengakomodasi perubahan kebutuhan pelanggan[34]. Kegiatan yang dilakukan pada tahap *planning* yaitu melakukan analisis kebutuhan seperti kebutuhan fungsional dan non fungsional. Untuk mendapatkan pemahaman sistem yang akan dikembangkan dapat melalui wawancara dengan pihak terkait. Pengolahan data wawancara dapat dijadikan dalam bentuk dokumentasi berupa *user story*. *User story* ini akan memberikan Gambaran mengenai output, fitur dan fungsi yang dibutuhkan [35]. Tabel 2.6 menunjukkan bentuk dokumentasi *user stories*.

Tabel 2. 6 Contoh *User Story*[35]

No	User Story
User Story 1	Deskripsi <i>user story</i>
User Story 2	Deskripsi <i>user story</i>

Berikut merupakan bentuk *Acceptance Test Criteria* ditunjukkan pada Tabel 2.7

Tabel 2. 7 Contoh *Acceptance Test Criteria* [35]

No	<i>Acceptance Test Criteria</i>
User Story 1	Deskripsi <i>Acceptance Test Criteria</i>
User Story 2	Deskripsi <i>user story</i>

berikut merupakan bentuk dari dokumentasi kebutuhan fungsional sistem ditunjukkan pada Tabel 2.8

Tabel 2. 8 Contoh Kebutuhan sistem [15]

No	Judul	Deskripsi
1	Menu sistem	Deskripsi kebutuhan fungsional
2	Menu sistem	Deskripsi kebutuhan fungsional

## 2) *Design*

Pada tahap desain dalam *Extreme Programming (XP)* melakukan perancangan dasar untuk fitur-fitur yang akan diimplementasikan, memastikan bahwa desain tersebut tidak hanya memenuhi kebutuhan fungsional, tetapi juga memperhatikan aspek non-fungsional. Prinsip-prinsip desain yang menekankan kesederhanaan dan penulisan kode yang bersih menjadi pedoman utama, bertujuan untuk memastikan bahwa perangkat lunak yang dihasilkan dapat dengan mudah disesuaikan atau diperbaiki di masa mendatang. Dengan fokus pada kejelasan dan kemudahan pemeliharaan, tahap desain menjadi fondasi dalam siklus pengembangan yang memastikan bahwa produk akhir dapat terus beradaptasi dengan perubahan kebutuhan [34]. Perancangan desain terdiri dari tiga pemodelan yaitu pemodelan sistem, perancangan user interface dan pemodelan basis data. Pemodelan sistem di representasikan melalui diagram *UML* yang terdiri dari diagram *Use Case*, *Activity* diagram, diagram *Sequence*, perancangan *user interface* berupa *wireframe* dan pemodelan basis data[36].

## 3) *Coding*

Dalam tahap *coding*, desain yang telah dirancang sebelumnya mulai di implementasikan dengan menuliskan kode untuk fitur-fitur yang telah direncanakan. Menerapkan prinsip-prinsip seperti *Pair Programming*, di mana dua pengembang bekerja secara bersama-sama di satu komputer, menjadi langkah yang



diambil untuk meningkatkan kualitas dan keamanan kode. Pada tahap ini sangat menekankan fokus pada kesederhanaan, kejelasan, dan uji coba otomatis sebagai bagian integral dari proses kodifikasi. Dengan demikian, maka dapat memastikan bahwa setiap baris kode tidak hanya berfungsi dengan baik, tetapi juga dapat dengan mudah dimengerti, dikelola, dan diuji secara efisien [34].

#### 4) *Testing*

Pada tahap pengujian dalam *Extreme Programming (XP)*, Setiap fitur yang diimplementasikan mendapatkan pengujian otomatis secara berkelanjutan, memastikan bahwa perangkat lunak dapat beroperasi sesuai dengan kebutuhan. Pada tahap ini acceptance test yang dilakukan oleh pengguna berfokus kepada fitur dan fungsi sistem untuk memastikan bahwa perangkat lunak tidak hanya memenuhi kebutuhan fungsional, tetapi juga sesuai dengan keinginan pengguna. Metode XP memastikan bahwa setiap bagian dari perangkat lunak yang telah diuji dan divalidasi dapat mencapai kualitas yang optimal [34].

### 2.2.11 *System Usability Scale*

System Usability Scale (*SUS*) merupakan metode yang digunakan untuk menilai sejauh mana sistem atau produk dianggap *usable* atau bermanfaat oleh penggunanya. Pendekatan ini berfokus pada persepsi pengguna sebagai dasar pengukuran. Dikembangkan oleh John Brooke pada tahun 1986, *SUS* telah menjadi salah satu standar evaluasi usability yang banyak digunakan. Metode ini melibatkan penggunaan kuesioner yang terdiri dari 10 pertanyaan, dengan skala 5 poin, yang dirancang untuk mengevaluasi aspek-aspek seperti *efficiency*, *effectiveness*, dan *satisfaction* [37]. Berikut instrumen pertanyaan ditunjukkan pada Tabel 2.9

Tabel 2. 9 Instrumen *SUS*[38]

No	Pertanyaan	Rasio
1	Saya berpikir akan menggunakan sistem ini lagi	1 – 5
2	Saya merasa sistem ini rumit untuk digunakan	1 – 5
3	Saya merasa sistem ini mudah digunakan	1 – 5
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini	1 – 5

No	Pertanyaan	Rasio
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya	1 – 5
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).	1 – 5
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat	1 – 5
8	Saya merasa sistem ini membingungkan	1 – 5
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini	1 – 5
10	Saya perlu membiasakan diri sebelum menggunakan	1 – 5

Responden akan memilih salah satu dari lima rasio untuk menyatakan sejauh mana persetujuan pada pernyataan yang berkaitan dengan sistem atau fungsi yang sedang diuji [38]. Skala penilaian ditunjukkan pada Tabel 2.10.

Tabel 2. 10 Skala Penilaian[37]

No	Skala	Skor	Keterangan
1.	STS	1	Sangat Tidak Setuju
2.	TS	2	Tidak Setuju
3.	N	3	Netral
4.	S	4	Setuju
5.	SS	5	Sangat Setuju

Perhitungan hasil pengujian dengan instrumen *SUS* diterapkan dengan mengikuti beberapa aturan [38]. Tabel 2.11 menunjukkan aturan dan rumus yang digunakan.

Tabel 2. 11 Aturan *SUS*[37]

No	Aturan	Rumus
1.	Apabila nomor pertanyaan adalah ganjil tingkat respons dari para responden dikurangkan sebesar 1.	$\Sigma Px - 1$
2.	jika nomor pertanyaan adalah genap, skala jawaban responden dikurangkan dari 5	$\Sigma 5 - Pn$

No	Aturan	Rumus
3.	Total dari konversi tersebut dihitung pada tiap responden dan dikalikan nilai 2,5	$(\Sigma \text{ skor ganjil} - \Sigma \text{ skor genap}) \times 2,5$

Setelah mengetahui rasio yang diberikan setiap responden, langkah berikutnya yaitu mencari rata-rata skor dengan menjumlahkan semua hasil skor kemudian membaginya dengan jumlah responden yang terlibat [38]. Perhitungan skor *SUS* ditunjukkan pada Persamaan 2.1

$$X = \Sigma x/n \quad (2.1)$$

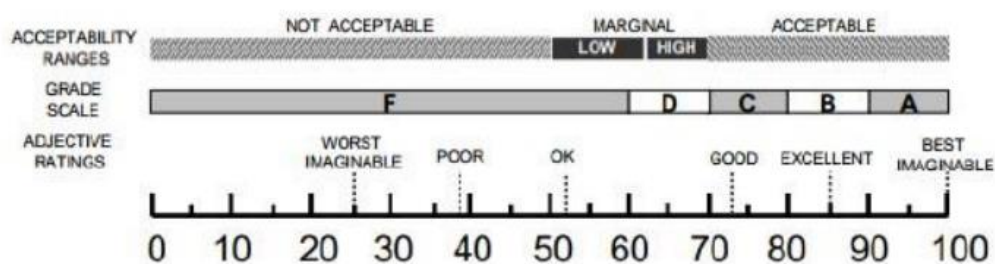
Keterangan:

X = Skor rata-rata

$\Sigma x$  = Jumlah skor system usability scale

N = Jumlah responden

Hasil rata rata pada perhitungan skor *SUS* disimpulkan dengan menginterpretasi hasil dalam acceptability range, grade scale dan adjective rating[38]. Seperti pada Gambar 2.3



Gambar 2. 3 Interpretasi Skor