

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam penelitian ini, penulis melakukan analisis literatur dari penelitian-penelitian sebelumnya sebagai sumber dukungan untuk penelitian yang akan dilakukan oleh penulis. Berikut ini adalah review literatur yang telah diteliti sebelumnya:

Tabel 2.1. Daftar Literatur

No	Penulis	Tahun	Judul
1	Lasma Divathama Tobing, Herman Tolle	2021	Perancangan User Experience Aplikasi E-Koperasi Sekolah berbasis Android dengan menggunakan Metode Human Centered Design (HCD)
2	Asep Saepulloh, Mohammad Adeyadi	2019	Aplikasi Scanner Berbasis Android Untuk Menampilkan Data Id Card Menggunakan Barcode
3	D Kustiawan, W N Cholifah, R Destriana, N Heriyani	2022	Rancang Bangun Sistem Informasi Akuntansi Pengelolaan Koperasi Menggunakan Metode Extreme Programming
4	Reza Alamsyah Harahap, Fajar Satriya Segarawasesa	2023	Analisis Implementasi Sistem Informasi Akuntansi Penjualan Kredit Pada Koperasi Guru dan Pegawai di MAN 1 Sleman
5	Nuzula Roliana Putri, Tintin Herlina	2022	Sistem Informasi Penjualan di Koperasi Siswa MTs Negeri Banyuwangi Berbasis Website

2.1.1. Tinjauan Literatur 1

Penelitian yang dilakukan oleh Lasma Divathama Tobing, Herman Tolle pada tahun 2021 yang berjudul Perancangan User Experience Aplikasi E-Koperasi Sekolah berbasis Android dengan menggunakan Metode Human Centered Design (HCD) dari Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Penelitian ini mencari solusi untuk meningkatkan kegiatan koperasi sekolah terkait masalah sistem yang digunakan. Koperasi sekolah berperan dalam menjual dan membeli barang yang dibutuhkan oleh anggota sekolah, namun keterbatasan waktu dalam melakukan kegiatan menjadi kendala. Oleh karena itu, sebuah solusi diusulkan dengan menggunakan Perancangan User Experience berbasis Human Centered Design (HCD). Solusi ini fokus pada kebutuhan dan aktivitas para pengguna, seperti para pembeli, kasir, suplier, dan admin koperasi. Evaluasi desain menggunakan User Experience Questionnaire (UEQ) menunjukkan hasil positif, namun Heuristic Evaluation menemukan beberapa masalah yang perlu diperbaiki. Secara keseluruhan, desain solusi ini memberikan persepsi positif dalam hal User Experience.

2.1.2. Tinjauan Literatur 2

Penelitian yang dilakukan oleh Asep Saepulloh, Mohammad Adeyadi pada tahun 2019 yang berjudul Aplikasi Scanner Berbasis Android Untuk Menampilkan Data Id Card Menggunakan Barcode dari Jurnal Manajemen Dan Teknik Informatika (JUMANTAKA). Penelitian ini berfokus pada masalah perkembangan pesat teknologi dan sistem informasi. Dalam perkembangan teknologi saat ini, kebutuhan akan informasi yang bermanfaat, benar, akurat, cepat, dan tepat menjadi sangat penting. Barcode telah menjadi salah satu cara untuk menyimpan informasi di

kemasan dan data-data lainnya. Namun, penggunaan barcode scanner konvensional di supermarket telah berkembang menjadi lebih efisien dengan adanya aplikasi android yang dapat melakukan scanning barcode. Penelitian ini bertujuan untuk menganalisis cara pembuatan scanner barcode menggunakan aplikasi android untuk meningkatkan efektivitas dan efisiensi penggunaan barcode.

2.1.3. Tinjauan Literatur 3

Penelitian yang dilakukan oleh D Kustiawan, W N Cholifah, R Destriana, N Heriyani pada tahun 2022 yang berjudul Rancang Bangun Sistem Informasi Akuntansi Pengelolaan Koperasi Menggunakan Metode Extreme Programming dari Jurnal Teknologi dan Informasi. Penelitian ini mengidentifikasi masalah dalam pengelolaan koperasi terkait perhitungan harga pokok, transaksi pembelian dan penjualan yang dicatat secara manual, kurangnya pencatatan pemasukan dan pengeluaran, dan penggunaan metode konvensional untuk mencatat persediaan bahan. Untuk mengatasi permasalahan tersebut, penelitian mengusulkan pengembangan Sistem Informasi Akuntansi (SIA) berbasis website dengan menerapkan metode Extreme Programming (XP). SIA ini bertujuan untuk membantu koperasi mengelola proses bisnis dan pengambilan keputusan dengan efisien. Hasil pengujian sistem menunjukkan bahwa SIA telah berfungsi sesuai dengan kebutuhan, dengan nilai black-box testing mencapai 100%.

2.1.4. Tinjauan Literatur 4

Penelitian yang dilakukan oleh Reza Alamsyah Harahap, Fajar Satriya Segarawasesa pada tahun 2023 yang berjudul Analisis Implementasi Sistem Informasi Akuntansi Penjualan Kredit Pada Koperasi Guru dan Pegawai di MAN 1 Sleman dari Jurnal Ilmiah Multidisiplin. Tujuan dari penelitian ini adalah untuk

menganalisis implementasi sistem informasi akuntansi penjualan kredit, yang mana penjualan kredit juga menjadi salah satu sumber pendapatan tetap pada Koperasi Amanah MAN 1 Sleman, guna mengetahui efektivitas sistem informasi akuntansi penjualan kredit. . Penelitian ini merupakan penelitian deskriptif analisis dengan pendekatan kualitatif, karena diperlukan observasi langsung terkait objek dan subjek yang diteliti. Data penelitian ini diperoleh melalui observasi, wawancara, dan dokumentasi di Koperasi Amanah MAN 1 Sleman. Hasil penelitian menunjukkan bahwa sistem informasi akuntansi penjualan kredit pada Koperasi Amanah MAN 1 Sleman masih belum efektif dan memadai, fungsi sistem penjualan kredit masih memadai dengan adanya tugas rangkap dan uraian tugas yang tidak sesuai dengan pekerjaan. di lapangan sehingga tugas yang dilaksanakan menjadi tidak efektif.

2.1.5. Tinjauan Literatur 5

Penelitian yang dilakukan oleh Nuzula Roliana Putri, Tintin Herlina pada tahun 2022 yang berjudul Sistem Informasi Penjualan di Koperasi Siswa MTs Negeri Banyuwangi Berbasis Website dari Jurnal Informatika dan Komputer. Penelitian ini membahas tentang Koperasi MTs Negeri Banyuwangi yang lebih banyak menggunakan tenaga kerja manusia untuk menyelesaikan pencatatan transaksi penjualan barang kepada siswa, guru, pegawai, dan yayasan. Keterbatasan fisik yang dimiliki sumber daya manusia pada koperasi MTs Negeri Banyuwangi menjadi salah satu penyebab terjadinya kesalahan (human error) dalam membuat daftar setoran penjualan per hari dan laporan per bulannya yang dilakukan secara manual. Permasalahan human error ini akan berdampak pada keakuratan data. Dari permasalahan tersebut perlu dibuatkan suatu sistem

informasi penjualan di koperasi MTs Negeri Banyuwangi Berbasis Website. Dengan menggunakan bahasa pemrograman PHP dan database MySQL. Tujuan dibuatnya sistem informasi ini adalah diharapkan dapat membantu system penjualan di koperasi MTs Negeri Banyuwangidan meminimalisir terjadinya kesalahan (human eror).

2.2. Koperasi Sekolah

Koperasi merupakan salah satu bentuk organisasi nirlaba yang umumnya terdapat di masyarakat. Anggotanya terdiri dari sejumlah individu atau badan yang ikut serta dalam kegiatan usaha tersebut dengan prinsip berdasarkan nilai-nilai kekeluargaan (Alfani, 2016). Kriteria yang berlaku pada koperasi bermacam-macam jenisnya, seperti koperasi simpan pinjam, koperasi karyawan, dan koperasi sekolah. Koperasi sekolah merupakan jenis koperasi yang didirikan oleh sekolah itu sendiri, dengan anggota siswa dan guru (Cholil & Putri, 2019).

Pada umumnya koperasi sekolah merupakan usaha retail yang fokus pada penjualan produk-produk yang berkaitan dengan kebutuhan sekolah. Dalam hal metode pembayaran, koperasi perdagangan biasanya menerapkan dua jenis metode yaitu pembayaran tunai dan pembayaran kredit (Harahap & Segarawasesa, 2023).

Untuk memudahkan dalam memahami koperasi, ada beberapa istilah yang sering digunakan dalam konteks koperasi, seperti yang dijelaskan dalam Kamus Istilah Akuntansi:

- a. Tagihan (*receivables*) : mencakup jumlah tagihan dari para langganan, jumlah piutang dagang, dan piutang wesel
- b. Utang (kewajiban) : *account payable, debt, indebtedness, payable, lid-bility* uang, barang atau jasa yang harus diserahkan kepada orang lain berdasarkan

suatu perjanjian, tersebut atau tersirat, yang me-nimbulkan kewajiban legal untuk membayar atau menyerahkan

- c. Piutang (*receivable*) : tuntutan terhadap debitor, biasanya timbul sebagai hasil penjualan barang dan jasa yang belum di-terima pembayarannya
- d. Kredit (*credit*) : kesanggupan untuk membeli atau meminiam dengan pertim-bangan adanya suatu perjianjian untuk membayar di dalam suatu periode tertentu, yang seringkali tidak ditetapkan dengan ketat, di-ikuti dengan pengiriman barang

2.3. Sistem Informasi Penjualan

Sistem informasi penjualan adalah sub sistem informasi bisnis yang mencakup serangkaian prosedur untuk melaksanakan, mencatat, menghitung, dan menyediakan dokumen dan informasi terkait penjualan, untuk keperluan manajemen dan pihak-pihak terkait, mulai dari penerimaan order penjualan hingga pencatatan tagihan atau kredit dagang(Faqih & Wahyudi, 2022). Berdasarkan pengertian di atas, dapat disimpulkan bahwa komponen-komponen dari sistem informasi penjualan secara umum meliputi:

- a. Pencatatan transaksi penjualan: Proses untuk mencatat setiap transaksi penjualan yang terjadi, termasuk detail produk atau layanan yang dijual, jumlah, harga, tanggal, dan identitas pelanggan.
- b. Kalkulasi jumlah dan harga: Menghitung total jumlah dan harga dari setiap transaksi penjualan, termasuk menghitung diskon atau pajak yang relevan.
- c. Pembuatan nota penjualan: Membuat dokumen nota penjualan yang berisi informasi lengkap tentang transaksi penjualan, yang kemudian diberikan kepada pelanggan sebagai bukti pembelian.

- d. Pembuatan laporan penjualan: Menyediakan laporan yang mencatat informasi secara komprehensif tentang seluruh aktivitas penjualan, seperti laporan penjualan harian, mingguan, bulanan, atau tahunan. Laporan ini dapat membantu manajemen dalam mengambil keputusan bisnis yang tepat.

2.4. E-Koperasi

E-Koperasi merupakan penjabaran dari kata *Electronic* Koperasi yang mengandung arti bahwa koperasi menggunakan fasilitas *Electronic* yang memanfaatkan jaringan komputer melalui internet sebagai penunjang pengoperasiannya. Maka dapat ditarik kesimpulan bahwa E-Koperasi bisa mengandung arti bahwa Pengelolaan Koperasi yang menggunakan media internet dalam mengelola kegiatannya (Rahayu et al., 2017).

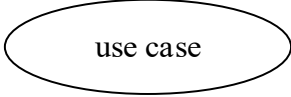
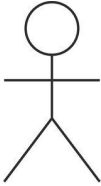
2.5. *Unified Modelling Language* (UML)


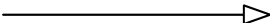
UML adalah bahasa visual yang digunakan untuk memodelkan dan berkomunikasi tentang sistem dengan menggunakan diagram dan teks pendukung (Fadallah & Rosyida, 2018). Bahasa pemodelan Unified Modeling Language (UML) terdiri dari berbagai jenis diagram yang harus diikuti aturan tertentu dalam pembuatannya. Diagram-diagram ini terdiri dari elemen-elemen grafis yang digunakan untuk membentuk satu kesatuan dalam menggambarkan perangkat lunak. Setiap diagram UML mewakili perspektif yang berbeda dari sistem dan fokus pada apa yang dilakukan sistem, bukan bagaimana sistem bekerja secara internal (Aryasa & Elly Kurniawan, 2019). Adapun dibawah ini merupakan beberapa jenis UML yang digunakan :

2.5.1. Use Case Diagram

Use case diagram digunakan untuk menggambarkan interaksi antara aktor atau pelaku dengan sistem melalui penggunaan kasus. Aktor dapat berupa individu, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dikembangkan. Diagram ini memiliki tujuan untuk menunjukkan fungsi-fungsi dari sistem atau persyaratan sistem dari sudut pandang pengguna (Fadallah & Rosyida, 2018). Adapun simbol-simbol yang terdapat pada *Use case diagram* terdapat pada tabel dibawah ini.

Tabel 2.2. Tabel *Use Case Diagram*

Nama	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan oleh sistem sebagai unit yang bertukar pesan antar unit atau actor
Aktor / <i>Actor</i>		Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat
<i>Extend</i>	-- <<extend>>-->	Hubungan <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>use case</i> tambahan
<i>Include</i>	-- <<include>>-->	Hubungan <i>use case</i> tambahan ke <i>use case</i> , dimana <i>use case</i> tambahan membutuhkan <i>use case</i> untuk berfungsi secara fungsional atau sebagai syarat agar <i>use case</i> ini berfungsi



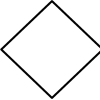
Nama	Simbol	Deskripsi
<i>Asosiasi</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi dalam <i>use case</i> atau <i>use case</i> berinteraksi dengan aktor
<i>Generalisasi</i>		Hubungan generalisasi dan spesialisasi (umum-spesifik) antara dua <i>use case</i> dimana satu fungsi lebih umum daripada yang lain

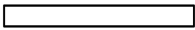

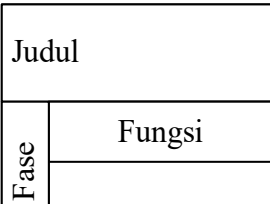
Sumber : (Ariani Sukamto & Shalahuddin, 2014)

2.5.2. Activity Diagram

Activity diagram adalah jenis diagram yang digunakan untuk memodelkan alur atau urutan dari sebuah proses bisnis atau aktivitas dalam perangkat lunak. Diagram ini menggunakan simbol-simbol untuk menggambarkan aktivitas-aktivitas atau tindakan-tindakan yang terlibat dalam proses tersebut (Fadallah & Rosyida, 2018). Adapun simbol-simbol yang terdapat pada *Activity diagram* terdapat pada tabel dibawah ini.

Tabel 2.3. Tabel *Activity Diagram*

Nama	Simbol	Deskripsi
Status Awal		Status awal aktivitas sistem dari sebuah diagram aktivitas
Aktivitas		Suatu aktivitas yang dilakukan oleh sistem, yang biasanya dimulai dengan kata kerja
Percabangan		Percabangan, jika pilihan aktivitas lebih dari satu

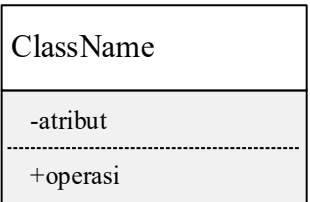
Nama	Simbol	Deskripsi
Penggabungan		Penggabungan, lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir		Status akhir yang dilakukan sistem dari sebuah diagram aktivitas
<i>Swimlane</i>		Memisahkan organisasi yang bertanggung jawab pada aktivitas yang terjadi

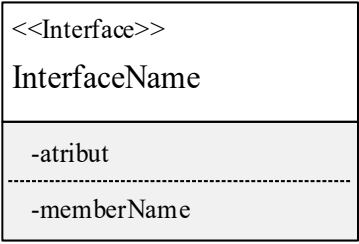
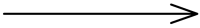
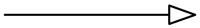
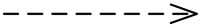
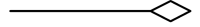
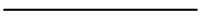
Sumber : (Ariani Sukamto & Shalahuddin, 2014)

2.5.3. Class Diagram

Class diagram adalah diagram yang digunakan untuk menggambarkan hubungan dan interaksi antara kelas-kelas dalam suatu sistem. Pembuatan class diagram didasarkan pada diagram use case dan diagram sequential yang telah dibuat sebelumnya (Fadallah & Rosyida, 2018). Adapun simbol-simbol yang terdapat pada *Class diagram* terdapat pada tabel dibawah ini.

Tabel 2.4. Tabel *Class Diagram*

Nama	Simbol	Deskripsi
<i>Class</i>		Kelas yang terdapat pada struktur. Atribut merupakan variable yang dimiliki suatu kelas. Operasi adalah perilaku yang dimiliki suatu kelas.

Nama	Simbol	Deskripsi
<i>Interface</i>		Sebuah entitas yang harus diwariskan untuk kelas
<i>Asociation</i>		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain
<i>Generalisasi</i>		Relasi antar kelas dengan makna generalisasi-spesialisasi
<i>Dependency</i>		Relasi antar kelas dengan makna ketergantungan
<i>Agregasi</i>		Relasi antar kelas dengan makna semua bagian
<i>Asosiasi</i>		Relasi antar kelas dengan makna umum

Sumber : (Ariani Sukamto & Shalahuddin, 2014)

2.6. *Android*

Sistem operasi *Android* adalah *platform seluler* yang mengalami modifikasi dari *kernel Linux* dan didesain secara khusus untuk digunakan pada perangkat seluler dengan layar sentuh, seperti telepon pintar dan tablet. Sistem ini menyediakan berbagai fitur canggih yang dapat digunakan oleh pengguna (Shonta et al., 2022).

Awalnya, *Android* dikembangkan oleh perusahaan bernama *Android Inc.*, dengan dukungan finansial dari *Google*, yang kemudian mengakuisisinya pada tahun 2005. Sistem operasi ini kemudian secara resmi dirilis pada tahun 2007. Antarmuka pengguna pada *Android* menggunakan pemrosesan langsung yang memungkinkan pengguna untuk menggunakan input sentuh untuk meniru tindakan

dunia nyata, seperti menggesek, menekuk, mencubit, dan memindahkan objek di layar (Aryasa & Elly Kurniawan, 2019).

2.7. Java

Java adalah bahasa pemrograman yang sangat fleksibel untuk pengembangan perangkat lunak yang dapat berjalan di berbagai perangkat seperti perangkat seluler, komputer desktop, dan server. Pada tahun 1995, Java diperkenalkan sebagai bahasa pemrograman berorientasi objek oleh *Sun Microsystems Inc.* yang dipimpin oleh *James Gosling* saat pembuatannya.

Salah satu tujuan utama dalam pembuatan bahasa pemrograman Java adalah menciptakan bahasa pemrograman yang dapat berjalan di berbagai perangkat tanpa harus terikat pada suatu platform tertentu. Hal ini membuat Java menjadi portable dan platform independent, artinya bahasa ini tidak tergantung pada mesin atau sistem operasi tertentu (Maiyana, 2018). Sebagai hasilnya, kode yang ditulis dalam bahasa Java dapat dengan mudah dijalankan di berbagai lingkungan yang berbeda tanpa perlu melakukan perubahan pada kode itu sendiri. Keunggulan ini menjadikan Java sebagai salah satu bahasa pemrograman yang sangat populer dan digunakan secara luas dalam pengembangan aplikasi dan perangkat lunak di berbagai platform.

2.8. Barcode

Barcode merupakan sekelompok data optik yang dapat diinterpretasikan oleh perangkat atau mesin. Cara kerja barcode adalah dengan menangkap data dari ruang antara garis-garis paralel dan lebar garis yang membentuk suatu pola, yang dikenal sebagai barcode atau simbol linear atau 1D. Selain itu, dalam bentuk gambar,

barcode juga bisa berbentuk kode matriks atau simbol 2D(Saepulloh & Adeyadi, 2019).

Barcode biasanya berbentuk batang dengan garis berwarna hitam yang terletak di atas latar belakang berwarna putih. Barcode digunakan untuk menulis dan menyimpan informasi, dan dapat dengan cepat diakses dan dibaca oleh perangkat pemindai atau scanner khusus. Informasi yang tersimpan dalam barcode bisa berupa kode unik, nomor identifikasi, angka, atau teks tertentu yang berkaitan dengan produk atau item yang memiliki barcode tersebut. Barcode sangat umum digunakan dalam berbagai bidang, termasuk perdagangan, manufaktur, logistik, dan pengelolaan inventaris.

2.9. *Laravel*

Laravel adalah sebuah kerangka kerja Bahasa pemrograman *PHP open source* yang dikembangkan oleh Taylor Otwell dengan konsep MVC (*Model-View-Controller*) untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan dan perbaikan, serta meningkatkan produktivitas kerja melalui sintaks yang bersih dan fungsional yang dapat menghemat banyak waktu implementasi (Luthfi, 2017).

Laravel menyediakan banyak fitur modern yang sangat membantu pengembang dalam membuat aplikasi. Beberapa keuntungan yang dimiliki oleh *Laravel* termasuk penggunaan *Command Line Interface (CLI) Artisan*, menggunakan manajemen paket *PHP Composer*, dan kemampuan untuk menulis kode program yang lebih pendek, lebih mudah dipahami, dan lebih ekspresif (Bin Tahir et al., 2019). Selain itu, *Laravel* juga membawa peningkatan dalam

pengembangan web dengan *PHP*, menjadikan *PHP* lebih kuat, cepat, aman, dan sederhana (Nugroho & Nugroho, 2021).

2.10. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang bersifat *open-source* (sumber terbuka) dan dikembangkan oleh perusahaan *Oracle Corporation*. Dikenal karena popularitasnya, *MySQL* sering digunakan sebagai basis data dalam pengembangan aplikasi web karena memiliki kecepatan, stabilitas, dan kemudahan penggunaan yang sangat diapresiasi. *MySQL* dapat dioperasikan pada berbagai platform, termasuk *Linux*, *Windows*, dan *Mac OS*, sehingga memberikan fleksibilitas bagi pengguna dalam memilih sistem operasi yang paling sesuai untuk kebutuhan mereka. Selain itu, *MySQL* juga mendukung berbagai bahasa pemrograman seperti *PHP*, *Java*, *C++*, dan lainnya, sehingga memungkinkan integrasi yang lancar dengan berbagai aplikasi dan lingkungan pengembangan (Fadila et al., 2021).

2.11. REST API (*Representational State Transfer Application Programming Interface*)

REST API adalah jenis antarmuka pemrograman aplikasi yang mengikuti pola arsitektur representasi sederhana dan tanpa sesi pada protokol *HTTP*. REST API memungkinkan komunikasi antara aplikasi dan server web, di mana permintaan dan respons dikirimkan dalam format data seperti *JSON* atau *XML* (Hasanuddin et al., 2022).

Tujuan dari desain REST API adalah untuk memudahkan penggunaan, mudah dipahami, dan dapat diakses melalui berbagai platform dan bahasa pemrograman. Dengan demikian, REST API memungkinkan aplikasi dari berbagai

sumber untuk berkomunikasi secara efisien tanpa memperhatikan platform atau bahasa pemrograman yang digunakan.

Untuk berinteraksi dengan sumber daya pada server web, REST API menggunakan empat jenis metode *HTTP*, yaitu *GET* untuk membaca data, *POST* untuk membuat data baru, *PUT* untuk memperbarui data, dan *DELETE* untuk menghapus data. Penggunaan metode *HTTP* ini memungkinkan aplikasi untuk melakukan operasi CRUD (*Create, Read, Update, Delete*) terhadap sumber daya pada server secara konsisten dan standar.

2.12. Visual Studio Code

Visual Studio Code adalah teks editor *open-source* yang dikembangkan oleh *Microsoft* dan dapat digunakan di semua sistem operasi (*multiplatform*), termasuk *Windows*, *Macintosh*, dan *Linux*. Editor ini mendukung berbagai bahasa pemrograman dan menjadi pilihan populer bagi para pengembang (Septianl, 2021).

Selain itu, *Visual Studio Code* memiliki fitur *marketplace* yang memungkinkan pengembang untuk mencari dan menggunakan plugin tambahan dalam berbagai kategori. Pengguna dapat menemukan berbagai tema, snippet, dan bahkan plugin yang menggunakan teknologi kecerdasan buatan (*Artificial Intelligence*) untuk membantu dalam proses penulisan kode program. Fitur-fitur ini sangat membantu meningkatkan produktivitas dan kualitas pekerjaan bagi seorang pengembang, karena mereka dapat mengakses alat-alat tambahan yang sesuai dengan kebutuhan proyek mereka. Dengan berbagai fitur unggulan ini, *Visual Studio Code* telah menjadi salah satu teks editor favorit di kalangan komunitas pengembang perangkat lunak.

2.13. *Android Studio*

Android Studio adalah sebuah *Integrated Development Environment* (IDE) *open-source* resmi yang dirancang khusus untuk pengembangan aplikasi berbasis *Android*. IDE ini merupakan pengganti dari *Eclipse* sebagai IDE utama untuk pengembangan aplikasi *Android* (Tjandra & Setiyawati, 2019) dan dipublikasikan oleh *Google* pada tanggal 16 Mei 2013 (Maiyana, 2018).

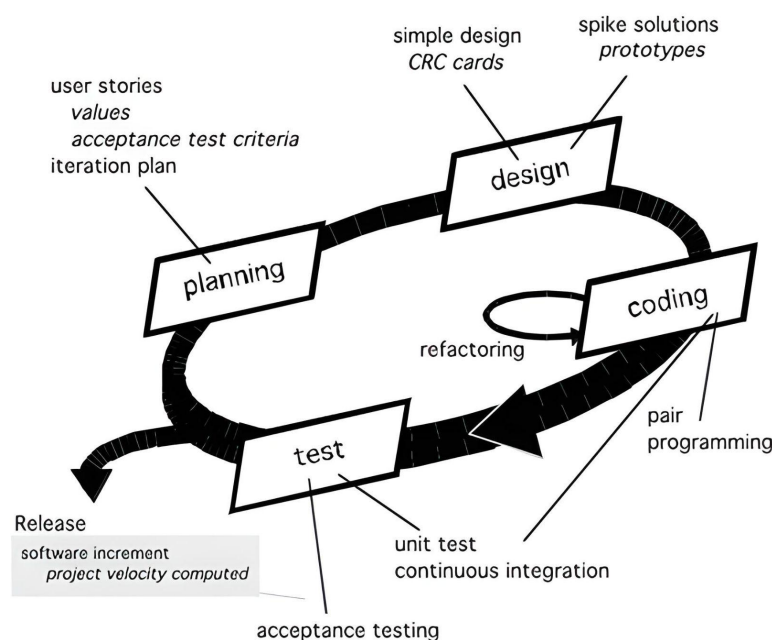
Android Studio menyediakan berbagai fitur canggih untuk mendukung pengembangan aplikasi *Android*. Beberapa fitur tersebut antara lain *system build* berbasis *Gradle* yang fleksibel, emulator yang cepat dan lengkap, serta lingkungan terpadu untuk mengembangkan aplikasi untuk semua perangkat *Android*. IDE ini juga memungkinkan pengembang untuk menerapkan perubahan langsung pada aplikasi yang sedang berjalan, menggunakan template kode, mengintegrasikan dengan *GitHub*, serta menyediakan framework dan alat pengujian yang lengkap. Alat lint juga disediakan untuk merekam performa aplikasi, dan terdapat dukungan *C++* dan *NDK* untuk pengembangan yang lebih kompleks. Selain itu, *Android Studio* memiliki dukungan bawaan untuk *Google Cloud Platform*.

Semua fitur yang disediakan oleh *Android Studio* bertujuan untuk meningkatkan produktivitas dan efisiensi dalam proses pengembangan aplikasi *Android*. Dengan antarmuka yang ramah pengguna dan alat yang lengkap, IDE ini menjadi pilihan favorit bagi para pengembang dalam menciptakan aplikasi *Android* yang inovatif dan berkualitas.

2.14. Metode *Extreme Programming* (XP)

Extreme Programming (XP) adalah bagian dari pendekatan *agile development* yang berfokus pada aktivitas pengkodean, yang merupakan tahapan

krusial dalam pengembangan perangkat lunak. XP menonjolkan responsivitas terhadap perubahan dan menggunakan iterasi yang dapat diulang-ulang sesuai kebutuhan. Setiap tahapan dalam XP dapat diselesaikan dengan cepat dan diulangi dalam durasi yang bervariasi sesuai dengan kebutuhan proyek. Proses pengembangan perangkat lunak dengan XP dimulai dari perencanaan, perancangan, pengkodean, hingga pengujian (Kustiawan, 2022). Adapun tahapan yang terdapat pada *Extreme Programming* terdapat pada gambar dibawah ini.



Gambar 2.1 *Extreme Programming*

Adapun tahapan-tahapan *software development* menggunakan *Extreme Programming* (XP) adalah sebagai berikut:

1. *Planning* (perencanaan)

Pada fase awal ini, langkah pertama yang harus diambil oleh pengembang adalah perencanaan sebelum memulai pembuatan sistem. Dalam tahap ini, fokus utama adalah merencanakan dan menganalisis kebutuhan pengguna yang akan terlibat dalam proses pengembangan sistem. Pengembang

membuat keputusan tentang bagaimana mewujudkan cerita yang disampaikan oleh pengguna sesuai dengan kesepakatan bersama. Berikut adalah langkah yang dapat diambil untuk mewujudkan cerita tersebut:

- a) Pengguna akan menjelaskan permasalahan yang dialami dalam penggunaan sistem dan bagaimana sistem yang diinginkan seharusnya dibangun.
- b) Berdasarkan deskripsi cerita pengguna, peneliti menentukan poin-poin kunci pada bagian nilai yang berfungsi sebagai panduan dalam menentukan elemen-elemen yang akan dikembangkan.
- c) Setelah kesepakatan tercapai, peneliti menetapkan kriteria uji penerimaan sebagai pedoman untuk pengujian sistem yang akan dilakukan.
- d) Selanjutnya, peneliti membuat keputusan tentang seberapa sering perilisan dan perbaikan akan dilakukan dalam tahap iterasi berikutnya, yang dikenal sebagai tahap perencanaan iterasi. Ini mencakup perencanaan pengujian sistem yang akan dibangun.

2. *Design* (perancangan)

Setelah tahap perencanaan selesai, langkah berikutnya adalah perancangan. Pada tahap ini, pengembang melakukan perancangan dengan membuat model yang dimulai dari pemodelan sistem, dilanjutkan dengan pemodelan arsitektur, dan pemodelan basis data. Tujuannya adalah memberikan gambaran yang jelas mengenai sistem yang akan dibangun.

- a) Perancangan Sederhana merupakan langkah dalam membangun perangkat lunak dengan desain yang simpel. Tahap ini dimulai dengan menggunakan

menggunakan UML seperti *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.

- b) *CRC Card* fungsinya untuk mengatur dan mengenali *object oriented class* yang disesuaikan dengan pengembangannya. Jika terjadi ketidaksesuaian selama perancangan, perbaikan akan dilakukan.
- c) *Prototipe* merupakan bagian dari perancangan yang mencakup antarmuka pengguna dalam bentuk *wireframe*. Ini bertujuan untuk mempermudah pengguna dalam melihat desain secara visual.

3. *Coding* (pengkodean)

Pada tahap pengkodean, peneliti menyesuaikan proses dengan cerita pengguna yang telah dibuat pada tahap perencanaan, sehingga sistem yang dibangun sesuai dengan kebutuhan. Proses pengkodean melibatkan langkah-langkah berikut:

- a) *Pair programming* adalah langkah di mana sistem yang akan dibangun dikodekan menggunakan bahasa pemrograman dan media penyimpanan yang telah disepakati.
- b) *Refactory* adalah tahap di mana perbaikan dilakukan ketika terjadi ketidaksesuaian dalam kode program untuk mencapai hasil yang sesuai.

4. *Test* (pengujian)

Tahap pengujian, atau sering disebut sebagai *testing*, akan dilakukan oleh pengguna sebagai pengujian akhir. Pengguna akan melakukan pengujian sesuai dengan *acceptance test* yang telah ditentukan dan disepakati sebelumnya. *Unit test* yang telah berhasil dibuat akan berfokus pada keseluruhan fitur dan *fungsiionalitas sistem*. Hal ini bertujuan untuk

memastikan bahwa sistem telah memenuhi kebutuhan yang diinginkan dan siap untuk diluncurkan.

2.15. *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak di mana pengujian dilakukan tanpa memperhatikan detail implementasi atau kode program yang digunakan. Tujuannya adalah untuk mengevaluasi apakah program atau sistem telah memenuhi persyaratan dan spesifikasi fungsional yang telah ditentukan sebelumnya. Dalam metode ini, pengujian dilakukan dengan menggunakan masukan tertentu dan mengamati keluaran yang dihasilkan. Sehingga, *Black Box Testing* bertujuan untuk memeriksa apakah program berjalan sesuai dengan fungsi yang diharapkan tanpa memperhatikan bagaimana program tersebut diimplementasikan (Ahrizal et al., 2020).

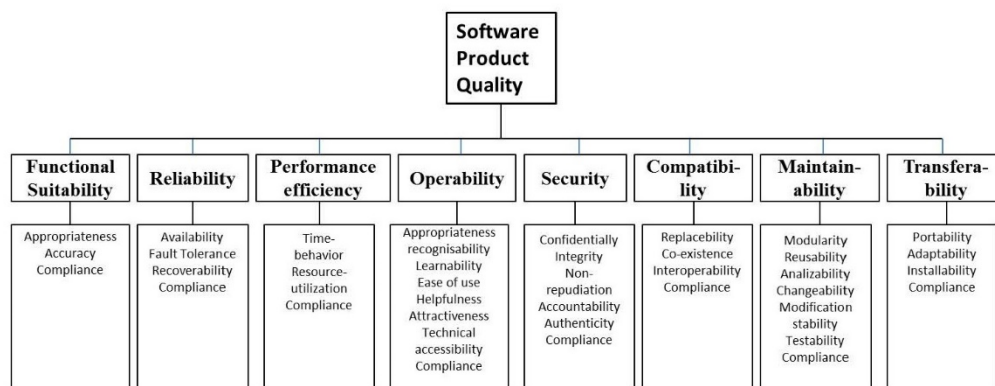
Dengan metode *Black Box Testing*, pengujian dilakukan dari perspektif pengguna atau pengguna potensial, dan tidak memerlukan pengetahuan tentang struktur internal atau algoritma yang digunakan dalam program. Pendekatan ini memungkinkan pengujian yang lebih obyektif dan memastikan bahwa perangkat lunak berfungsi dengan baik sesuai dengan kebutuhan dan harapan pengguna. *Black Box Testing* juga membantu dalam mengidentifikasi bug dan kelemahan dalam perangkat lunak secara keseluruhan tanpa harus mengetahui rincian teknis implementasinya.

2.16. Pengujian 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. 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 sub-karakteristik yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer (Harun, 2020). Berikut ini karakteristik dan sub-karakteristik yaitu sebagai berikut :



Gambar 2.2 Model ISO 25010

1. Functional Suitability, karakteristik ini mewakili sejauh mana suatu produk atau sistem menyediakan fungsi yang dapat memenuhi kebutuhan untuk digunakan dalam kondisi tertentu. Karakteristik ini terdiri dari sub-karakteristik :
 - a. Functional completeness, merupakan tingkat yang mengatur fungsi-fungsi mencakup semua tugas yang ditentukan dan tujuan pengguna.

- b. Functional correctness, merupakan tingkat di mana produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang diperlukan.
 - c. Functional appropriateness, merupakan tingkat di mana fungsi yang tersedia mampu memfasilitasi pencapaian tugas dan tujuan tertentu
2. Reliability, merupakan tingkat di mana suatu sistem, produk atau komponen melakukan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu. Karakteristik ini terdiri dari sub-karakteristik berikut :
- a. Maturity, merupakan tingkat di mana suatu sistem, produk atau komponen memenuhi kebutuhan akan keandalan di bawah operasi normal.
 - b. Availability, merupakan tingkat di mana suatu sistem, produk atau komponen operasional dan dapat diakses ketika diperlukan untuk digunakan.
 - c. Fault Tolerance, merupakan tingkat dimana suatu sistem, produk atau komponen beroperasi sebagaimana dimaksud meskipun ada kesalahan perangkat keras atau perangkat lunak.
 - d. Recoverability, merupakan tingkat dimana suatu sistem, produk atau sistem dapat memulihkan data yang terkena dampak langsung dan membangun kembali keadaan yang diinginkan dari sistem.
3. Performance Efficiency, karakteristik ini mewakili kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang ditentukan. Karakteristik ini terdiri dari sub-karakteristik berikut :

- a. Time behavior, merupakan tingkat di mana respons, waktu proses dan tingkat keluaran suatu produk atau sistem ketika menjalankan fungsinya memenuhi persyaratan.
 - b. Resource utilization, merupakan tingkat di mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem ketika menjalankan fungsinya memenuhi persyaratan.
 - c. Capacity, merupakan tingkat di mana batas maksimum produk atau parameter sistem memenuhi persyaratan.
4. Usability, merupakan tingkat di mana produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam konteks penggunaan yang ditentukan.
- Karakteristik ini terdiri dari sub-karakteristik berikut :
- a. Appropriateness recognizability, merupakan tingkat di mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai untuk kebutuhan mereka.
 - b. Learnability, merupakan tingkat di mana suatu produk atau sistem dapat digunakan oleh pengguna mencapai tujuan pembelajaran tertentu untuk menggunakan produk atau sistem dengan efektifitas, efisiensi, bebas dari risiko dan kepuasan dalam konteks penggunaan yang ditentukan.
 - c. Operability, merupakan tingkat di mana produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikontrol.
 - d. User error protection, merupakan tingkat di mana sistem melindungi pengguna dari membuat kesalahan.

- e. User interface aesthetics, merupakan tingkat di mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
 - f. Accessibility, merupakan tingkat di mana suatu produk atau sistem dapat digunakan oleh orang-orang dengan jangkauan terluas karakteristik dan kemampuan untuk mencapai tujuan tertentu dalam konteks penggunaan yang ditentukan.
5. Security, merupakan tingkat di mana suatu produk atau sistem melindungi data dan informasi sehingga orang atau produk atau sistem lain memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasinya.

Karakteristik ini terdiri dari sub-karakteristik berikut:

- a. Confidentiality, merupakan tingkat di mana suatu produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
- b. Integrity, merupakan tingkat di mana suatu produk atau sistem atau komponen mencegah akses yang tidak sah atau modifikasi dari program atau data komputer.
- c. Non-repudiation, merupakan tingkat di mana tindakan atau peristiwa dapat dibuktikan telah terjadi, sehingga peristiwa atau tindakan tidak dapat ditolak kemudian.
- d. Accountability, merupakan tingkat di mana tindakan suatu entitas dapat dilacak secara unik kepada entitas.
- e. Authenticity, merupakan tingkat di mana identitas subjek atau sumber daya dapat dibuktikan menjadi yang diklaim.

6. *Compatibility*, merupakan tingkat di mana suatu produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen lain, atau melakukan fungsi yang diperlukan sambil berbagi perangkat keras atau perangkat lunak yang sama. Karakteristik ini terdiri dari sub-karakteristik berikut :

- a. *Coexistence*, merupakan tingkat di mana suatu produk dapat melakukan fungsi yang diperlukan secara efisien sambil berbagi lingkungan dan sumber daya umum dengan produk lain tanpa dampak yang merugikan pada produk lain.
- b. *Interoperability*, merupakan tingkat di mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah ditukar.

7. *Maintainability*, karakteristik ini mewakili tingkat efektifitas dan efisiensi dari suatu produk atau sistem dapat dimodifikasi untuk perbaikan atau menyesuaikannya dengan perubahan lingkungan sesuai persyaratan.

Karakteristik ini terdiri dari sub-karakteristik berikut:

- a. *Modularity*, merupakan tingkat di mana sistem atau program komputer terdiri dari komponen-komponen terpisah sehingga perubahan pada satu komponen memiliki dampak minimal pada komponen lain.
- b. *Reusability*, merupakan tingkat di mana aset dapat digunakan di lebih dari satu sistem, atau dalam membangun aset lain.
- c. *Analysability*, merupakan tingkat keefektifan dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau sistem dari

- perubahan yang dimaksudkan untuk satu atau lebih dari bagian-bagiannya, atau untuk mendiagnosis suatu produk untuk kekurangan atau penyebab kegagalan, atau untuk mengidentifikasi bagian yang akan dimodifikasi.
- d. Modifiability, merupakan tingkat dimana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa memperkenalkan cacat atau menurunkan kualitas produk yang ada.
 - e. Testability, merupakan tingkat efektivitas dan efisiensi dengan mana kriteria pengujian dapat ditetapkan untuk suatu sistem, produk atau komponen dan tes dapat dilakukan untuk menentukan apakah kriteria tersebut telah dipenuhi.
8. Portability, merupakan tingkat efektivitas dan efisiensi dengan mana suatu sistem, produk atau komponen dapat ditransfer dari satu perangkat keras, perangkat lunak atau lingkungan operasional atau penggunaan lainnya ke yang lain. Karakteristik ini terdiri dari sub-karakteristik berikut:
- a. Adaptability, merupakan tingkat di mana suatu produk atau sistem dapat secara efektif dan efisien diadaptasi untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lain yang berbeda atau berkembang.
 - b. Installability, merupakan tingkat efektivitas dan efisiensi di mana produk atau sistem dapat berhasil dipasang dan / atau dihapus di lingkungan tertentu.

- c. Replaceability, merupakan tingkat di mana suatu produk dapat menggantikan produk perangkat lunak lain yang ditentukan untuk tujuan yang sama dalam lingkungan yang sama.

2.17. Skala Likert

Skala Likert adalah skala yang digunakan dalam mengukur sikap, pendapat, dan persepsi seseorang atau orang tentang fenomena social. Dengan Skala Likert, maka variable yang akan diukur dijabarkan menjadi indicator variable. Kemudian indicator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrument yang dapat berupa pernyataan atau pertanyaan (Bahrun et al., 2017).

Karakteristik dari skala ini yang membedakan dari skala lain adalah pilihan dari masing-masing pertanyaan dari instrument yang digunakan berupa pilihan yang mempunyai gradasi dari sangat positif samapai sangat negatif, seperti sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat tidak setuju. Dalam skala likert bentuk penyajian yang dapat digunakan terbagi menjadi dua yaitu pilihan ganda dan bentuk checklist (Bahrun et al., 2017).

Maka dapat disimpulkan bahwa, skala likert merupakan sebuah skala yang dapat digunakan untuk mengukur sikap, pendapat, ataupun persepsi seseorang maupun sekelompok orang tentang suatu fenomena sosial. Skala likert mempunyai karakteristik yaitu pilihan yang menggunakan beberapa pernyataan, misalnya seperti sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat tidak setuju. Karakteristik tersebut yang membedakan skala ini dengan skala yang lainnya.