

BAB II

TINJAUAN PUSTAKA

2.1. Kajian Pustaka

Penelitian ini dikerjakan dengan menggunakan sumber acuan dari penelitian sebelumnya sebagai panduan pembuatan. Pemanfaatan teknologi informasi sudah banyak diimplementasikan pada kehidupan sehari-hari terutama dalam pemesanan *Online*, salah satunya adalah *GrabFood*, *Pizza Hut Delivery*, dll. Semakin berkembangnya zaman, mulai muncul penelitian-penelitian perancangan aplikasi kantin elektronik.

Beberapa diantaranya adalah penelitian dengan judul “Pengembangan Aplikasi E-Kantin Berbasis *Mobile* dan *Web*”[4]. Penelitian ini menggunakan metodologi *waterfall* dalam penelitiannya [4]. Tujuan dari penelitian ini adalah untuk merancang sebuah aplikasi kantin elektronik yang bisa mempermudah proses pemesanan makanan dan minuman[4]. Hasil penelitian menunjukkan bahwa fungsionalitas dari sistem yang diciptakan berjalan dengan baik. Pemilik stan bisa melihat urutan pemesanan. Fitur pemilihan meja memudahkan dalam mencari identitas pemesan. Setiap transaksi tercatat dengan terstruktur. Kekurangan dari penelitian ini adalah untuk pengelola kantin diharuskan menggunakan *web*. Akan kurang efektif jika pengelola kantin untuk bisa mengakses kantinnya harus menggunakan laptop maupun dengan mengakses *web*.

Dilanjutkan dengan penelitian berikutnya dengan judul “Perancangan Aplikasi *MyCanteen* berbasis *Android* dengan Menggunakan Metode *Object Oriented Analysis* dan *Design* Pada Kantin Makan Nona Geprek Prabumulih”[1]. Tujuan dari penelitian ini adalah untuk mengatasi masalah pada proses pemesanan makanan dan minuman dan mengurangi kesalahan pemesanan antara penjual dan pembeli dengan membuat aplikasi *MyCanteen* yang berbasis *android*. Aplikasi ini dibangun menggunakan bahasa pemrograman *PHP* dan dikembangkan menggunakan *software Appsgeyser.com*[1]. Memanfaatkan metode *waterfall* dalam proses pengembangan. Kelemahan yang terdapat pada penelitian ini adalah pada bagian hasil dan pembahasan tidak dijelaskan bagaimana hasilnya, pada penelitian ini hanya melampirkan gambar saja tanpa memberikan kejelasan.

Lalu kemudian terdapat penelitian dengan judul “Rancang Bangun Aplikasi Penjualan *Online* Berbasis *Web* Menggunakan Metode *Scrum*”[9]. Penelitian ini diambil sebagai referensi karena topik yang masih berhubungan dengan penjualan *Online* dan metode yang digunakan sama. Tujuan dari penelitian ini adalah untuk merancang sebuah aplikasi penjualan *Online* yang berbasis *web* untuk menciptakan sistem yang bisa digunakan dan sesuai dengan tujuan yang telah dibuat[9]. Hasil dari penelitian ini menunjukkan bahwa aplikasi yang dirancang berbasis *web* adalah aplikasi penjualan yang dapat memasarkan produk secara *Online* dan membantu dalam mengelola proses transaksi[9]. Dengan melakukan pengembangan menggunakan *scrum* pembeli dapat mengakses *website* tanpa *login*, melakukan registrasi akun, *login* menggunakan validasi data, memasukkan produk ke dalam keranjang, dan melakukan pemesanan produk[9]. Adapun kelemahan yang teridentifikasi dari hasil penelitian adalah hanya admin saja yang dapat mengelola produk, dan aplikasi hanya dikembangkan untuk *web* saja.

Penelitian selanjutnya adalah penelitian dengan judul “Implementasi e-Kantin di Fakultas Teknik Universitas Pancasila”[2]. Dengan menggunakan metodologi *waterfall*, penelitian ini bertujuan untuk memudahkan antara penjual dan pembeli agar dapat lebih praktis dan efisien dalam proses transaksi pada sebuah kantin [2]. Dari penelitian ini, ditemukan bahwa aplikasi *e-canteen* yang dibangun memudahkan konsumen, terutama mahasiswa teknik Universitas Pancasila[2]. Aplikasi yang telah dibuat juga memudahkan penjual dalam menjalankan proses transaksi[2]. Hasil penelitian menunjukkan bahwa ada beberapa kekurangan yaitu tampilan aplikasi masih kurang rapi, pembahasan proses aplikasi yang kurang jelas, dan tidak adanya informasi tentang pengujian yang dilakukan selama penelitian.

Selanjutnya dari penelitian dengan judul “Pengembangan Aplikasi *MyCanteen* dengan Pembayaran Non-Tunai Berbasis *Android* Untuk Siswa (Studi Kasus: SMA Negeri 5 Malang)”[3]. Penelitian ini menggunakan metodologi *waterfall* dalam penelitiannya, adapun tujuannya adalah menyelesaikan permasalahan kantin dalam pemesanan dan kesulitan yang dihadapi pembeli dengan membuat sebuah aplikasi *MyCanteen* untuk memudahkan dalam transaksi dengan pembayaran yang dilakukan secara non-tunai[3]. Hasil dari penelitian ini menunjukkan bahwa

Aplikasi yang dirancang dengan menggunakan metode *Waterfall* memiliki fitur seperti *Sign In*, *Sign Up*, Pemilihan Stan, Pemilihan Menu, Penambahan Pesanan, Pembayaran, Riwayat Pesanan, Melihat Saldo, Akun Saya, dan *Log Out*[3]. Hasil penelitian menunjukkan bahwa aplikasi yang dirancang menggunakan metode *waterfall* memiliki fitur pembayaran menggunakan *QR Code* dan dilakukan pengujian dengan metode *blackbox*. Dari hasil pengujian, perangkat lunak tersebut terbukti memenuhi spesifikasi yang diharapkan[3]. Kekurangan yang ditemui pada penelitian ini adalah tidak adanya tempat untuk menyimpan gambar yang akan di *upload* untuk menu yang akan ditambahkan. Serta tidak adanya notifikasi dan tidak adanya rekap pesanan bulanan.

Penelitian berikutnya dengan judul “Sistem Informasi Kantin Elektronik (E- Canteen) Politeknik Negeri Tanah Laut Berbasis Web *Mobile*”[12]. Penelitian ini menggunakan metode *waterfall* dan bertujuan untuk memecahkan masalah efisiensi pemesanan dan mempercepat proses yang sebelumnya dilakukan secara manual[12]. Kesimpulan dari penelitian adalah perancangan sebuah sistem informasi kantin elektronik berdasarkan teknologi *web mobile* untuk pemesanan juga sewa secara *Online* di kantin politeknik negeri tanah laut yang berhasil mencapai tujuan sesuai harapan[12]. Kekurangan yang didapat dalam penelitian adalah tahapan pengujian *BlackBox* hanya menguji hasil yang diharapkan saja, tidak sesuai dengan spesifikasi yang ada, sehingga sulit untuk memastikan apakah sistem tersebut dapat bekerja dengan efisien.

Dilanjutkan dengan penelitian dengan judul “*E-Kantin* UNIKOM Sebagai Layanan Pemesanan Berbasis Web *E-Kantin* UNIKOM As *Web-Based Order Services*”[13]. Metodologi yang digunakan adalah *protoype* dan tujuan penelitian adalah untuk memecahkan masalah waktu tunggu dalam pemesanan yang lama dengan membangun sistem pemesanan makanan dan minuman sehingga pesanan bisa dilakukan sebelum jam waktu makan[13]. hasil dari penelitian ini adalah perancangan aplikasi *e-kantin* yang diusulkan untuk mengubah cara pencatatan secara konvensional menjadi komputerisasi memberikan kemudahan bagi mahasiswanya untuk melakukan pesan[13]. Kekurangan dari penelitian ini adalah hasil pengujian yang tidak dicantumkan sehingga tidak menjelaskan apakah aplikasi bekerja atau tidak.

Untuk melihat ringkasan dari penelitian sebelumnya, tabel ulasan literatur dari penelitian terdahulu bisa diamati pada bagian tabel 2.1 ini :

Tabel 2. 1 Literatur *review* dari penelitian sebelumnya

| Peneliti | Judul | Metode | Hasil |
|---|--|--|---|
| Andri, Robin, Muhammad Ridho (2019) | Pengembangan Aplikasi E-Kantin Berbasis <i>Mobile</i> dan <i>Web</i> [4] | <i>Waterfall</i> | Hasil dari sistem yang dibuat memberikan kinerja yang baik. Penyewa stan bisa meninjau urutan pemesanan melalui sistem.. Fitur pemilihan meja memudahkan dalam mencari identitas pemesan. Setiap transaksi tercatat dengan terstruktur. Kekurangan dari penelitian ini adalah untuk pengelola kantin diharuskan menggunakan <i>web</i> dan berdasarkan hasil gambar menggunakan laptop/PC. Akan kurang efektif jika pengelola kantin untuk bisa mengakses kantinnya harus menggunakan laptop maupun dengan mengakses <i>web</i> . |
| Andi Christian, Rishi Suprianto, Della Salsabillah Putri (2021) | Perancangan Aplikasi <i>Mycanteen</i> Berbasis <i>Android</i> dengan Menggunakan Metode <i>Object Oriented Analysis</i> dan Design (<i>Ooad</i>) Pada Kantin Makan Nona Geprek Prabumulih[1] | <i>Object Oriented Analysis & Design</i> | Aplikasi diciptakan menggunakan bahasa pemrograman PHP dan perangkat lunak <i>Appsgeyser.com</i> . Memanfaatkan metode <i>waterfall</i> dalam proses pengembangan. Kelemahan yang terdapat pada penelitian ini adalah pada bagian hasil dan pembahasan tidak dijelaskan bagaimana hasilnya, pada penelitian ini hanya melampirkan gambar saja tanpa memberikan kejelasan. |
| Aryanata Andipradana, Kristoko Dwi Hartomo (2021) | Rancang Bangun Aplikasi Penjualan <i>Online</i> Berbasis <i>Web</i> Menggunakan Metode <i>Scrum</i> [9] | <i>Scrum</i> | Aplikasi yang dibuat berbasis <i>web</i> bertujuan sebagai solusi pemasaran produk dapat dipesan secara <i>online</i> dan mempermudah pengelolaan proses transaksi. Dengan metode pengembangan <i>scrum</i> , pengguna dapat mengakses situs <i>web</i> tanpa harus log in, melakukan registrasi akun, log in dengan menggunakan validasi data, memasukkan produk ke keranjang belanja, dan melakukan pemesanan produk. Kurangnya dalam penelitian ini adalah hanya administrator saja yang memiliki akses untuk |

| Peneliti | Judul | Metode | Hasil |
|--|---|-------------------|---|
| | | | mengelola produk. kemudian bahasa yang digunakan menggunakan bahasa pemrograman versi lama, serta pengembangan hanya dilakukan pada <i>web</i> saja. |
| Adityo Cahyo Nugroho, Gusti Rahana Putra, Desti Fitriati (2019) | Implementasi E-Kantin Di Fakultas Teknik Universitas Pancasila[2] | <i>Waterfall</i> | Aplikasi e-kantin yang telah dibuat memberikan kemudahan konsumen terkhusus mahasiswa teknik universitas Pancasila. Aplikasi yang telah dibuat juga memudahkan penjual dalam menjalankan proses transaksi. Kelemahan dalam penelitian ini adalah pada desain tampilan aplikasi yang masih berantakan serta tidak jelasnya informasi bagaimana proses-proses aplikasi di dalam pembahasan, serta tidak adanya informasi pengujian yang dilakukan pada penelitian tersebut |
| Vega Revaldy, Widhy Hayuhardhika Nugraha Putra, Buce Trias Hanggara (2020) | Pengembangan Aplikasi <i>Mycanteen</i> dengan Pembayaran <i>Non-Tunai</i> Berbasis <i>Android</i> Untuk Siswa (Studi Kasus: Sma Negeri 5 Malang)[3] | <i>Waterfall</i> | Aplikasi yang dibuat menggunakan metode <i>waterfall</i> memiliki beberapa fitur seperti <i>sign in</i> , <i>sign up</i> , pemilihan stan, pemilihan menu, penambahan pemesanan, pembayaran, melihat riwayat pesanan, melihat saldo, akses akun pribadi, dan <i>logout</i> . Pembayaran dilakukan menggunakan QR Code dan pengujian dilakukan dengan metode <i>blackbox</i> . Hasil pengujian menunjukkan bahwa perangkat lunak memenuhi spesifikasi yang diharapkan. Kekurangan yang ditemui pada penelitian ini adalah tidak adanya tempat untuk menyimpan gambar yang akan di <i>upload</i> untuk menu yang akan ditambahkan. Serta tidak adanya notifikasi dan tidak adanya recap pesanan bulanan |
| Siti Purnama, Khairul Anwar Hafizd, Rabini Sayyidati (2020) | Sistem Informasi Kantin Elektronik (<i>E-Canteen</i>) Politeknik Negeri Tanah Laut Berbasis <i>Web Mobile</i> [12] | <i>waterfall.</i> | Sistem informasi kantin elektronik berbasis <i>web mobile</i> telah dibuat dan berhasil memberikan solusi untuk transaksi pemesanan dan penyewaan produk di kantin elektronik di Politeknik Negeri Tanah Laut, sehingga dapat memenuhi harapan dari hasil yang diinginkan. Kekurangan yang didapat dalam penelitian |

| Peneliti | Judul | Metode | Hasil |
|---|--|------------------|--|
| | | | adalah tahapan pengujian <i>BlackBox</i> hanya menguji hasil yang diharapkan saja, Tidak sesuai dengan spesifikasi yang ditentukan, sehingga sulit untuk memastikan apakah sistem yang dibuat akan bekerja dengan efisien. |
| Akhammad Fatah Rusdi, Agus Nursikuwagus (2018) | E-Kantin UNIKOM Sebagai Layanan Pemesanan Berbasis <i>Web Mycanteen</i> UNIKOM As <i>Web-Based Order Services</i> [13] | <i>Prototype</i> | Perancangan sistem yang diusulkan untuk mengubah cara pencatatan secara konvensional menjadi komputerisasi memberikan kemudahan bagi mahasiswanya untuk melakukan pesanan. Kemudian dapat memudahkan penjual untuk melakukan proses transaksi. Kekurangan dari penelitian ini adalah hasil pengujian yang tidak dicantumkan sehingga tidak menjelaskan apakah aplikasi bekerja atau tidak. |

2.2. Dasar Teori

2.2.1. Kantin

Kantin ialah tempat yang mana bisa dimanfaatkan oleh para pelajar sebagai sebuah tempat untuk berkumpul untuk sekedar bersosialisasi maupun untuk memesan dan melakukan transaksi makanan dan minuman[2]. Kantin juga merupakan sebuah keberadaan wajib yang harus ada pada suatu kampus[2]. Kantin juga dapat di artikan sebagai tempat yang terdapat pada suatu area sekolah maupun perguruan tinggi yang menyediakan pilihan-pilihan menu yang bagus sebagai pemenuhan kebutuhan juga kesehatan manusia yang disajikan oleh penjual kantin sekolah[3].

2.2.2. Android

Android adalah sistem operasi yang berdasar pada sistem operasi Linux yang digunakan pada ponsel dan tablet[14]. Pada awalnya, *android* dikembangkan oleh sebuah perusahaan *startup* bernama *Android Inc.*, yang didirikan oleh Oleg Andry Rubin, Rich Miner, Nick Sears, dan Chris White di California. Pada tahun 2005, *Google* membeli *Android* dan mengambil alih pengembangannya hingga saat ini. *Google* merilis versi *beta Android SDK (System Development Kit)* pada November 2007[15]. Sistem operasi *mobile* yang muncul seiring dengan pertumbuhan sistem operasi lain memberikan peluang bagi *developer* untuk membuat dan mengembangkan aplikasi mereka sendiri yang dapat digunakan pada beragam jenis perangkat *mobile*[3][14].

2.2.2.1. Android Studio

Android Studio adalah sebuah *Integrated Development Environment (IDE)* yang didukung secara resmi dan bersifat *open source*, dan dapat digunakan untuk membuat aplikasi yang didasarkan pada *android*[3]. *Android Studio* pertama kali dirilis pada konferensi *Google I/O* tanggal 16 Mei 2013, dan dapat digunakan secara gratis dengan lisensi *Apache 2.0*[15]. Pengembangan aplikasi *android* menggunakan *Android Studio* diperkaya dengan berbagai fitur yang meningkatkan produktivitas, seperti sistem versi berbasis *Gradle* yang fleksibel, *emulator* yang cepat dan lengkap, lingkungan pengembangan yang terintegrasi untuk semua perangkat *android*, *instant run* untuk mempercepat proses pengujian aplikasi,

template kode dan integrasi *GitHub* untuk memudahkan pembuatan fitur dan mengimpor kode, kerangka kerja pengujian dan *debugging* yang luas, alat *Lint* untuk meningkatkan kinerja, kegunaan, dan kompatibilitas, dukungan untuk bahasa *C++* dan *NDK*, dukungan bawaan untuk *Google Cloud Platform*, serta integrasi yang mudah dengan *Google Cloud Messaging* dan *App Engine*[16].

2.2.3. Kotlin

Kotlin merupakan satu dari banyak bahasa pemrograman yang didukung oleh Mesin Virtual Java (JVM) [15]. *Kotlin* juga adalah sebuah bahasa pemrograman untuk *Android* yang menggabungkan prinsip pragmatis orientasi objek (OO) dan fungsional[15]. Kemudian, Bahasa pemrograman *Kotlin* dapat berintegrasi dengan bahasa pemrograman Java dan digunakan untuk pengembangan aplikasi pada berbagai platform seperti *desktop*, *web*, dan *backend*[15]. *Kotlin* dikembangkan oleh *JetBrains*, perusahaan yang juga mengembangkan *IntelliJ IDEA*[15]. Setelahnya *JetBrains* mempublikasikan *kotlin* sebagai *open source* setelah banyak pengembangan yang saat ini masih dalam tahap pengembangan. *Google* sepenuhnya mendukung penggunaan *kotlin* oleh pengembang aplikasi *Android*[15].

2.2.4. Firebase

Firebase ialah salah satu *API* yang diterbitkan oleh *Google*, yang memungkinkan untuk dimanfaatkan untuk menyimpan dan sinkronisasi data pada aplikasi yang salah satunya pada *android*[3]. Salah satu fiturnya yaitu *Realtime Database* yang mampu menyimpan dan memanggil data dengan cepat[3]. *Firebase Database* adalah penyimpanan basis data *non-SQL* yang memungkinkan pengguna untuk menyimpan beberapa jenis data, seperti *String*, *Long*, dan *Boolean*. Data disimpan dalam bentuk objek *JSON tree*, tidak seperti basis data *SQL* yang menggunakan tabel dan baris[18]. Ketika data ditambahkan, data akan menjadi *node* dalam struktur *JSON*. Setiap *node* berisi data dan dapat memiliki cabang-cabang lain yang berisi data juga. Proses memasukkan data ke *Firebase Database* disebut dengan *push*[18].

2.2.5. Unified Modeling Language

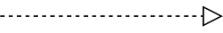
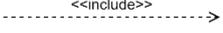
UML (Unified Modeling Language) adalah salah satu metode pemodelan *visual* yang digunakan dalam proses desain dan pembuatan perangkat lunak

berbasis *objek*[16]. UML adalah standar pembuatan atau semacam cetak biru yang berisi proses bisnis pembuatan kelas dalam bahasa tertentu. Ada beberapa *diagram* UML yang sering digunakan selama proses pengembangan suatu sistem, di antaranya adalah[16]:

2.2.5.1. Use Case Diagram

Use Case Diagram adalah gambaran dari tindakan dan interaksi yang diinginkan antara aktor dan sistem dalam suatu sistem[16]. Dalam sebuah *Use Case*, biasanya memiliki *actor* yang dapat digambarkan sebagai entitas baik manusia ataupun sistem yang melakukan aktivitas pada sistem[16]. Terdapat beberapa komponen yang ada didalam sebuah *activity*, diantaranya *actor*, *use case*, *association*, *generalisasi*, *include*, dan *extend*[17]. Penjelasan lebih detail bisa dilihat pada tabel 2.2 berikut :

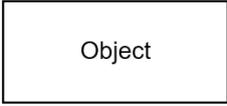
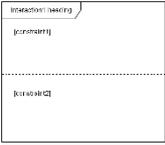
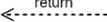
Tabel 2. 2 Komponen-komponen *Use Case Diagram*[17]

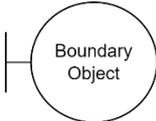
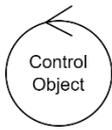
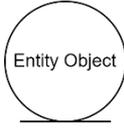
| Simbol | Keterangan |
|---|---|
|  Actor | Aktor komponen <i>use case</i> yang mewakili entitas yang berinteraksi dengan sistem. Aktor dapat berupa orang, sistem lain, atau alat yang terlibat dalam skenario penggunaan <i>use case</i> . |
|  Use Case | <i>Use Case</i> merupakan representasi grafis dari fungsionalitas sistem yang menunjukkan interaksi antara aktor dan sistem |
|  | <i>Association</i> adalah salah satu relasi dalam <i>diagram use case</i> yang menunjukkan hubungan antara aktor dan <i>use case</i> . <i>Association</i> merepresentasikan keterlibatan atau interaksi antara aktor dan <i>use case</i> yang terkait |
|  | Generalisasi menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> |
|  | <i>Include</i> merupakan relasi antara <i>use case</i> yang menunjukkan bahwa fungsionalitas dari <i>use case</i> yang satu sepenuhnya tergantung pada fungsionalitas <i>use case</i> lain. |
|  | <i>Extend</i> merupakan relasi antara <i>use case</i> yang menunjukkan bahwa suatu <i>use case</i> (yang disebut <i>use case</i> tambahan) dapat dieksekusi hanya jika kondisi tertentu terpenuhi dalam <i>use case</i> lain (yang disebut <i>use case</i> utama) |

2.2.5.2. *Sequence Diagram*

Sequence Diagram adalah representasi visual dari hubungan dan pesan yang diterima dan dikirim antara objek dalam sistem dan lingkungannya dalam urutan waktu[16]. Terdapat beberapa komponen dalam *sequence diagram*, diantaranya adalah *actor*, *activation box*, *object*, *lifelines*, *option loop*, *synchronous*, *reply message*, *delete message*, *boundary*, *control*, dan *entity*[18]. Untuk penjelasan lebih lengkap bisa dilihat pada tabel 2.3 berikut :

Tabel 2. 3 Komponen-komponen *Sequence Diagram*[18]

| Simbol | Keterangan |
|--|--|
|  Actor | Komponen <i>stick figure</i> dalam <i>sequence diagram</i> dikenal sebagai aktor, yang mewakili pengguna yang terlibat dalam interaksi dengan sistem, baik secara internal maupun eksternal |
|  | <i>Activation box</i> adalah suatu komponen persegi panjang pada <i>sequence diagram</i> yang mewakili durasi waktu yang dibutuhkan oleh suatu objek untuk menyelesaikan tugas tertentu. Semakin lama waktu yang dibutuhkan, maka <i>activation box</i> akan semakin panjang |
|  Object | <i>Object</i> merupakan sebuah komponen berbentuk kotak dalam <i>diagram</i> yang digunakan untuk mendemonstrasikan bagaimana sebuah objek akan berinteraksi dalam suatu sistem dan bagaimana perilakunya dalam konteks yang diberikan |
|  :Object | <i>Lifelines</i> adalah komponen <i>diagram</i> yang berbentuk garis putus-putus, digunakan untuk menunjukkan urutan kejadian yang terjadi pada objek selama pembuatan <i>diagram</i> |
|  | <i>Option loop</i> adalah komponen <i>diagram</i> yang sangat berguna untuk menggambarkan bagaimana sistem berperilaku dalam situasi tertentu, terutama jika ada percabangan atau alternatif jalur yang harus dipertimbangkan |
|  | <i>Synchronous message</i> adalah salah satu komponen <i>diagram</i> yang dilambangkan dengan simbol panah tebal ke kanan, dan digunakan untuk menunjukkan bahwa pengirim harus menunggu <i>respon</i> pesan sebelum melanjutkan |
|  | <i>Reply message</i> adalah salah satu komponen <i>diagram</i> yang dilambangkan dengan simbol panah putus-putus ke kiri, dan berfungsi sebagai balasan atas suatu panggilan tertentu |

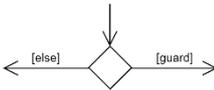
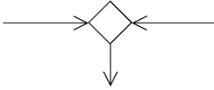
| Simbol | Keterangan |
|--|--|
|  | <p><i>Delete message</i> merupakan komponen <i>diagram</i> yang dilambangkan dengan tanda cakra di tengahnya terdapat garis putus-putus, dan berfungsi untuk menghapus objek tertentu</p> |
|  | <p>Komponen <i>boundary</i> dalam <i>diagram</i> biasanya terletak pada batas luar sistem, dan dapat berupa antarmuka pengguna atau alat lain yang digunakan untuk berinteraksi antara pengguna dan sistem</p> |
|  | <p>Komponen <i>control</i> dalam sebuah sistem bertanggung jawab untuk mengatur alur informasi dalam skenario bisnis, dan dapat mempengaruhi perilaku teknis dari sistem</p> |
|  | <p><i>Entity</i> adalah komponen sistem yang berperan menyimpan informasi atau data, dan biasanya direpresentasikan sebagai objek model atau <i>beans</i></p> |

2.2.5.3. Activity Diagram

Activity Diagram adalah sebuah representasi visual yang menggambarkan urutan dari aktivitas yang dilakukan oleh suatu sistem, menunjukkan aliran aktivitas secara berurutan[16]. Pada sebuah *activity diagram*, terdapat beberapa komponen yang ada didalamnya, seperti *initial state/start*, *final state/end*, *activity*, *transisi/association*, *decision*, dan *merge*[19]. Untuk penjelasan lebih lengkap mengenai komponen-komponen tersebut, bisa dilihat pada tabel 2.4 berikut :

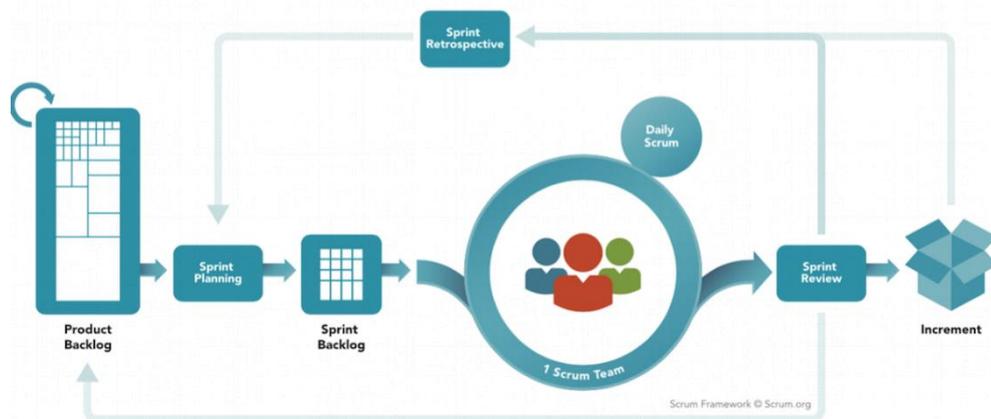
Tabel 2. 4 Komponen-komponen *Activity Diagram*[19]

| Simbol | Keterangan |
|---|--|
|  | <p><i>Initial state</i> merupakan titik awal atau <i>starting point</i> dari alur kerja dalam sebuah <i>activity diagram</i>, dan hanya terdapat satu <i>initial state</i> pada setiap <i>diagramnya</i></p> |

| Simbol | Keterangan |
|---|---|
|  | <p><i>Final state</i> merupakan komponen yang menunjukkan akhir dari suatu alur kerja. Pada satu <i>activity diagram</i>, bisa terdapat lebih dari satu <i>final state</i> tergantung pada desain alur kerja</p> |
|  | <p><i>Activity diagram</i> adalah sebuah <i>diagram</i> yang digunakan untuk memodelkan urutan aktivitas atau tugas dalam suatu alur kerja atau proses bisnis. <i>Diagram</i> ini membantu dalam menggambarkan hubungan antara aktivitas-aktivitas tersebut, sehingga memudahkan dalam analisis proses bisnis atau alur kerja dalam sebuah sistem</p> |
|  | <p>Transisi/<i>Association</i> pada <i>Activity Diagram</i> menghubungkan antara dua aktivitas atau lebih dan menunjukkan urutan dari aktivitas tersebut. Komponen ini berfungsi untuk menggabungkan aktivitas selanjutnya setelah aktivitas sebelumnya dan menunjukkan aliran kerja atau alur yang harus diikuti oleh sistem</p> |
|  | <p>Keputusan atau "<i>Decision</i>" dalam <i>activity diagram</i> digunakan untuk merepresentasikan sebuah pilihan atau kondisi dimana alur kerja dapat bercabang menjadi dua atau lebih jalur</p> |
|  | <p><i>Merge</i> berfungsi untuk menggabungkan kembali aliran kerja yang sebelumnya telah dipecah oleh suatu <i>decision point</i>, sehingga mengembalikan aliran kerja menjadi satu jalur tunggal.</p> |

2.2.6. Metode Scrum

Scrum adalah sebuah pendekatan pengembangan perangkat lunak yang menggunakan prinsip-prinsip agil. Ini memungkinkan untuk menghasilkan perangkat lunak sesuai dengan kebutuhan pengguna dan memiliki fleksibilitas untuk mengatasi perubahan. *Scrum* juga sangat cocok untuk proyek skala kecil [11]. Berikut merupakan alur kerja dari metode *scrum* sesuai Gambar 2.1 :



Gambar 2. 1 Scrum DailyLifeCycle[22].

2.2.7. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan bagian penting dalam proses pengembangan perangkat lunak. Pengujian *BlackBox* digunakan pada penelitian ini untuk memastikan kualitas perangkat lunak yang dibuat.

2.2.7.1. BlackBox Testing

BlackBox Testing adalah metode pengujian yang fokus pada uji coba fungsionalitas aplikasi tanpa memperhatikan desain dan implementasi kode program. Ini dilakukan untuk menemukan masalah seperti kesalahan dalam fungsi aplikasi dan *item* aplikasi yang hilang[20]. Jadi dengan *BlackBox* testing, kita memungkinkan melakukan uji terhadap kinerja dan fungsionalitas aplikasi yang sudah dibuat tanpa perlu mempelajari bagaimana aplikasi itu dibuat dan bagaimana kode programnya bekerja[20]. Hasil dari perhitungan kelayakan diterjemahkan menjadi nilai kualitatif melalui interpretasi menggunakan Skala *Likert* seperti ditunjukkan dalam tabel 2.5[21]:

Tabel 2. 5 Interpretasi Skala *Likert*

| Nomor | Persentase | Interpretasi |
|-------|----------------------|--------------|
| 1 | Nilai 0 hingga 20% | Sangat Buruk |
| 2 | Nilai 21 hingga 40% | Buruk |
| 3 | Nilai 41 hingga 60% | Cukup |
| 4 | Nilai 61 hingga 80% | Baik |
| 5 | Nilai 81 hingga 100% | Sangat Baik |