

## **BAB II**

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

#### **2.1 Tinjauan Pustaka**

Pada bab ini, terdapat tinjauan terhadap penelitian sebelumnya dimana peneliti telah menemukan beberapa penelitian terdahulu yang membahas atau menjelaskan metode terkait. Berdasarkan sejumlah penelitian sebelumnya, dapat disimpulkan bahwa kemampuan untuk memprediksi suatu peristiwa memiliki manfaat yang signifikan dalam menyusun perencanaan, persiapan, dan memudahkan pengambilan keputusan. Beberapa studi terdahulu yang dianggap relevan oleh penulis dengan penelitian yang akan dilaksanakan.

Pertama, penelitian yang berjudul “Implementasi *Scrum Framework* Pada Manajemen Proyek Perancangan Sistem Informasi Pemesanan Jasa Inspeksi Mobil Bekas” dilakukan oleh Dian Ambar Wasesha pada tahun 2023. Penelitian ini dilakukan untuk membuat aplikasi *mobile* Android untuk manajemen proyek yang dapat membantu manajer proyek dalam memantau pekerjaan harian pekerja proyek dan memenuhi kebutuhan atau fitur klien yang selalu berubah dan bertambah. Dalam perancangan sistem pada penelitian ini, metode yang diterapkan adalah metode Scrum. Dari penelitian ini dihasilkan aplikasi Android yang dapat menampilkan detail pekerjaan dan penjadwalannya, desain alur sistem hingga *prototype* antar mukanya dan diharapkan dapat membantu calon pembeli mobil bekas dalam menghilangkan kekhawatiran terhadap mobil bekas yang akan dibeli[9].

Kedua, penelitian yang berjudul “Sistem Informasi Pelayanan Jasa Foto Berbasis Android (Studi Kasus : *ACE Photography Way Kanan*)” dilakukan oleh Aditya Dwi Saputra dan Rohmat Indra Borman pada tahun 2020. Penelitian ini dilakukan untuk membuat aplikasi yang mampu memberikan kemudahan terhadap pelanggan yang akan melakukan

pelayanan jasa foto di *ACE Photography*. Serta pelanggan memiliki akses untuk melihat dan memesan pelayanan jasa toko secara daring, kemudahan dalam proses pemesanan toko dan memberikan informasi jadwal pemotretan yang transparan kepada pelanggan. Dalam pengembangan sistem pada penelitian ini, metode yang diterapkan adalah metode Agile. Pengujian yang digunakan pada penelitian ini menerapkan metode pengujian *black-box* pada sistem evaluasi. Hasil yang diperoleh dengan menghitung 1 aspek yaitu fungsionalitas dari sistem dan berdasarkan kuesioner dengan persentase nilai sebesar 80% yang dinyatakan sangat layak. Berdasarkan hasil yang didapatkan, dapat disimpulkan bahwa aplikasi android yang dibuat telah mampu meningkatkan kinerja dan telah memenuhi syarat yang ditetapkan[10].

Ketiga, penelitian yang berjudul “Rancang Bangun Aplikasi Jasa Penyediaan Pelayanan Perawatan Perbaikan Menggunakan Metode Agile Berbasis Android Pada PT. Otani” dilakukan oleh Kevin Rocky Damara Pasaribu dan Abdul Meizar pada tahun 2023. Penelitian ini dilakukan untuk membangun sistem berbasis android yang mampu meningkatkan efisiensi para pekerja dalam melaksanakan tugas, sehingga dapat membantu mempermudah pelayanan perawatan dan perbaikan pada PT. Otani. Dalam membangun sistem pada penelitian ini, metode yang diterapkan adalah metode Scrum. Penelitian ini menerapkan metode pengujian *black-box* untuk mengevaluasi fungsionalitas dan kestabilan sistem. Hasil dari penelitian ini adalah aplikasi yang dibuat dapat memberikan suatu solusi yang meningkatkan efisiensi layanan perawatan dan perbaikan. Aplikasi ini memudahkan divisi untuk melaporkan masalah kepada teknisi dengan menyajikan informasi secara cepat dan akurat[11].

Keempat, penelitian yang berjudul “Perancangan Sistem Pemesanan Online Jasa *Print Out* Berbasis Android Menggunakan Metode Scrum” yang dilakukan oleh Clarissa Chandra dan Joko Susilo pada tahun 2021. Penelitian ini dilaksanakan dengan tujuan mengembangkan aplikasi *mobile* yang memfasilitasi pengguna dalam menemukan layanan *print-out*

di sekitarnya, dengan kemampuan melakukan pemesanan secara daring yang akan diterima oleh pemilik atau pengelola layanan *print-out*. Dalam membangun sistem pada penelitian ini, metode yang diterapkan adalah metode Scrum. Hasil dari penelitian ini adalah aplikasi yang dapat memfasilitasi pelanggan untuk dengan mudah mencari dan memesan layanan cetak, sambil menyajikan informasi mengenai status proses pemesanan yang dilakukan oleh pelanggan[12].

Kelima, penelitian yang berjudul “Aplikasi Yunit *Laundry* Menggunakan *Framework Scrum*” dilakukan oleh Marcelliano R. A. Omega, Gladly C. Rorompandey, dan Vivi P. Rantung pada tahun 2023. Penelitian ini dilakukan untuk mengembangkan aplikasi Yunit *Laundry* yang dirancang untuk memudahkan para pengusaha *laundry* dalam mengelola bisnisnya. Dalam mengembangkan sistem pada penelitian ini, metode yang diterapkan adalah metode Scrum. Hasil dari penelitian ini adalah dua aplikasi, yaitu aplikasi *website* yang diperuntukkan bagi admin untuk mengelola operasional *laundry*, dan aplikasi *mobile* berbasis Android yang ditujukan untuk pengguna yang ingin melakukan pemesanan *laundry*[13].

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No	Judul	Penulis	Tahun	Studi kasus	Metode/ Algoritma/ Pengujian	Hasil	Perbedaan dengan penelitian yang dilakukan
1	Implementasi Scrum Framework Pada Manajemen Proyek Perancangan Sistem Informasi Pemesanan Jasa Inspeksi Mobil Bekas[9]	Dian Ambar Wasesha	2023	Inspeksi Mobil Bekas	<i>Scrum</i>	Aplikasi Android yang dirancang mampu menampilkan detail tugas dan jadwal, mencakup desain alur sistem hingga <i>prototype</i> antarmukanya. Diharapkan aplikasi ini dapat membantu calon pembeli mobil bekas dalam mengatasi kekhawatiran terkait kondisi mobil bekas yang akan mereka beli.	Dalam penelitian sebelumnya, objek penelitiannya adalah <i>inspector</i> profesional mobil bekas. Namun, dalam penelitian ini, objek penelitiannya adalah pelanggan yang ingin memesan pelayanan jasa terhadap mitra UMKM dari OBRE.
2	Sistem Informasi Pelayanan Jasa Foto Berbasis Android (Studi Kasus : ACE <i>Photography Way Kanan</i> )[10]	Aditya Dwi Saputra dan Rohmat Indra Borman	2020	ACE <i>Photography Way Kanan</i>	<i>Agile</i>	Pada penelitian ini diperoleh hasil berupa skor presentase <i>black-box</i> sebesar 80% dengan indeks <Sangat Layak>. Berdasarkan kriteria tersebut, dapat disimpulkan bahwa aplikasi android ini telah memenuhi persyaratan yang ditetapkan.	Pada penelitian sebelumnya objek yang difokuskan adalah merancang aplikasi Android pemesanan jasa foto ACE <i>Photography Way Kanan</i> , Sedangkan penelitian ini difokuskan untuk merancang aplikasi Android pemesanan layanan jasa terhadap mitra UMKM

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No	Judul	Penulis	Tahun	Studi kasus	Metode/ Algoritma/ Pengujian	Hasil	Perbedaan dengan penelitian yang dilakukan
							dari OBRE bagi pelanggan.
3	Rancang Bangun Aplikasi Jasa Penyediaan Pelayanan Perawatan Perbaikan Menggunakan Metode Agile Berbasis Android[11]	Kevin Rocky Damara Pasaribu	2023	PT. Otani	<i>Agile</i>	Dalam penelitian ini, hasil yang diperoleh adalah aplikasi yang dibuat memberikan kenyamanan dalam memberikan layanan perawatan dan perbaikan. Aplikasi ini mempermudah divisi untuk melaporkan masalah kepada teknisi dengan menyajikan data secara efisien dan akurat.	Pada penelitian sebelumnya, menggunakan metode agile untuk membangun aplikasi dan objek penelitiannya adalah jasa pelayanan perawatan perbaikan pada PT. OTANI. Sedangkan dalam penelitian ini, pendekatan yang digunakan adalah Scrum dan objek penelitiannya adalah pelanggan yang memesan layanan jasa terhadap mitra UMKM dari OBRE.
4	Perancangan Sistem Pemesanan <i>Online Jasa Print Out</i> Berbasis	Clarissa Chandra dan Joko Susilo	2021	Jasa Print Out di Jakarta Utara	<i>Scrum</i>	Sebuah aplikasi yang dapat memfasilitasi pelanggan untuk dengan mudah mencari dan memesan layanan cetak, sambil menyajikan informasi mengenai	Pada penelitian sebelumnya objek yang difokuskan adalah merancang aplikasi Android untuk

Tabel 2.1 Ringkasan Penelitian Sebelumnya

No	Judul	Penulis	Tahun	Studi kasus	Metode/ Algoritma/ Pengujian	Hasil	Perbedaan dengan penelitian yang dilakukan
	Android Menggunakan Metode Scrum[12]					status proses pemesanan yang dilakukan oleh pelanggan.	pemesanan layanan jasa <i>print out</i> yang ada di Jakarta Utara, Sedangkan penelitian ini difokuskan untuk merancang aplikasi Android pemesanan layanan jasa terhadap mitra UMKM dari OBRE.
5	Aplikasi Yunit Laundry Menggunakan Framework Scrum[13]	Marcelliano R. A. Omega, Gladly C. Rorompandey, dan Vivi P. Rantung	2023	Jasa Laundry di Sulawesi Utara	Scrum	Hasil dari penelitian ini mencakup dua aplikasi, yakni aplikasi situs web yang diperuntukkan bagi admin dalam mengelola operasional <i>laundry</i> , dan aplikasi seluler berbasis Android yang ditujukan bagi pengguna yang ingin melakukan pemesanan layanan <i>laundry</i> .	Pada penelitian sebelumnya, objek penelitiannya adalah aplikasi pengelolaan jasa <i>laundry</i> bagi admin dan aplikasi pemesanan jasa <i>laundry</i> bagi pelanggan. Namun, dalam penelitian ini, objek penelitiannya adalah pelanggan yang ingin memesan layanan jasa terhadap mitra UMKM dari OBRE.

## 2.2 Landasan Teori

### 2.2.1 OBRE (Ojek Brebes)

OBRE adalah sebuah usaha yang saat ini menyediakan layanan jasa transportasi ojek online dan yang memiliki misi untuk meningkatkan pemanfaatan sumber daya manusia yang ada di Kabupaten Brebes dengan menciptakan peluang pekerjaan bagi penduduk setempat, dan saat ini jasa layanannya berfokus pada area Brebes saja untuk mendongkrak perekonomian dari warga lokal yang ada di Brebes, terutama untuk kepala keluarga yang berpenghasilan Rp 11.000/hari. Usaha ini didirikan pada tanggal 15 November 2022, oleh Ibu Loly, Pak Hendra, Ibu Dwi, dan Pak Novel, yang beralamat kantor Jl. RA Kartini No.17, Saditan, Brebes, Jawa Tengah dan berada di bawah naungan Yayasan sosial bernama Yayasan Pemali Peduli Kasih. Usaha ini didirikan berawal dari keluhan-keluhan masyarakat sekitar karena sedikitnya lapangan pekerjaan untuk kepala keluarga karena rata-rata hanya untuk perempuan.

### 2.2.2 Android

Android merupakan suatu sistem operasi yang berlandaskan pada kernel Linux[14] yang digunakan di *smartphone* dan Android memiliki fungsi yang serupa dengan sistem operasi Symbian pada perangkat Nokia, iOS di produk Apple, serta BlackBerry OS. Android Inc. merupakan perusahaan yang awalnya mengembangkan Android sebelum akhirnya diakuisisi oleh Google pada tahun 2005[15]. Android menyediakan platform terbuka bagi para pengembang yang bersifat *open source* yang dimana siapapun bisa mengembangkannya dan menciptakan aplikasi android mereka sendiri dan bersifat gratis, serta merupakan sistem operasi yang sangat terkenal[16].

### 2.2.3 Android Studio



Gambar 2.1 Android Studio [17]

Android Studio merupakan *platform* pengembangan yang disediakan oleh Google untuk sistem operasi Android. *Platform* ini merupakan integrasi resmi yang dibangun berdasarkan perangkat lunak JetBrains IntelliJ IDEA dan Android Studio dirancang khusus untuk mempermudah proses pengembangan aplikasi Android[18]. Dan digunakan untuk membuat kode atau mengedit kode secara efektif untuk pengembangan aplikasi android[19].

### 2.2.4 Kotlin



Gambar 2.2 Kotlin [20]

Sumber: <https://kotlinlang.org/docs/kotlin-brand-assets.html>

Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) dan dikembangkan oleh JetBrains. Bahasa ini praktis karena menggabungkan elemen-elemen pemrograman berorientasi objek (*Object Oriented*) dan pemrograman fungsional. Salah satu keunggulan utama Kotlin adalah interoperabilitas, memungkinkan penggunaan bersama dengan bahasa pemrograman Java dalam satu proyek. Selain digunakan untuk mengembangkan aplikasi Android, Kotlin juga dapat diterapkan dalam pembuatan aplikasi *desktop*, web, dan *backend*[21][22].

### 2.2.5 Firebase

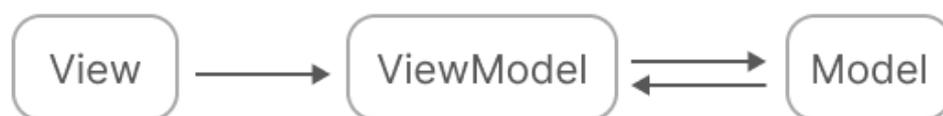


Gambar 2.3 Firebase [23]

Sumber: <https://firebase.google.com/>

Firebase merupakan penyedia layanan awan (*Cloud Service Provider*) dan *backend* sebagai layanan (*backend as a service*) yang ditawarkan oleh Google. *Platform* ini memberikan kemudahan bagi pengembang dalam membuat aplikasi berbasis web maupun *mobile*[24]. Dengan menggunakan Firebase, pengembang dapat memusatkan perhatian tanpa perlu menyisihkan upaya besar dalam pengembangan aplikasi. Dua fitur yang mencolok dari Firebase meliputi *Firebase Remote Config* dan *Firebase Realtime Database*[25].

### 2.2.6 MVVM (*Model-View-ViewModel*)



Design Pattern MVVM Layer

Gambar 2.4 MVVM [26]

Sumber: <https://www.dicoding.com/blog/tips-design-pattern-mvvm/>

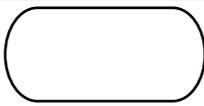
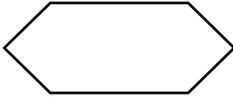
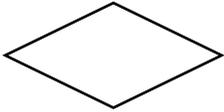
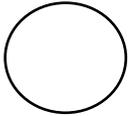
MVVM adalah suatu pola desain arsitektur atau paradigma terbaru yang digunakan untuk mengelola fokus pengembangan aplikasi Android yang bertujuan sebagai pemisah antarmuka pengguna dari logika bisnis (*back-end*). Sejak Google memperkenalkan MVVM pada acara Google I/O tahun 2018,

perusahaan tersebut secara konsisten mendorong penggunaan pola ini dalam pengembangan aplikasi android studio[27].

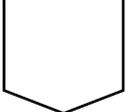
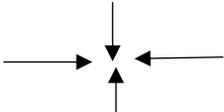
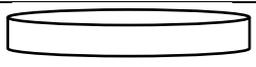
### 2.2.7 Flowchart

*Flowchart* adalah suatu metode representasi visual yang menggunakan simbol-simbol tertentu. Metode ini digunakan untuk menjelaskan tahap-tahap pemecahan masalah dengan mudah dipahami, mudah digunakan, dan sesuai dengan standar[28]. *Flowchart* memberikan gambaran simbol-simbol yang digunakan untuk menggambarkan hubungan antara proses secara detail dalam pembuatan program, sehingga program tersebut terstruktur dengan baik[29]. Simbol-simbol tersebut memiliki fungsi dan arti tertentu, yang menjadikan *flowchart* sebagai representasi visual yang membantu memahami dan mengilustrasikan urutan proses dalam suatu program[30].

Tabel 2.2 Simbol *Flowchart*

No	Simbol	Keterangan
1		<i>Terminal</i> : Melambangkan titik awal atau akhir suatu proses atau algoritma[31], [32], [33].
2		Proses : Mewakili aktivitas atau pengolahan data yang dilakukan dalam proses[31].
3		<i>Input/Output</i> : Mewakili masukan atau keluaran dalam proses[32], [33].
4		<i>Preparation</i> : Mewakili langkah di mana persiapan atau inisiasi suatu proses atau kegiatan dilakukan[31].
5		<i>Decision</i> : Mewakili suatu keputusan yang memiliki dua kemungkinan hasil, umumnya berupa opsi ya atau tidak. Arah jalur keluar dari keputusan yang memberikan pilihan langkah berikutnya berdasarkan hasil keputusan tersebut[31].
6		<i>Connector</i> : Merupakan suatu kondisi yang menunjukkan sambungan pada langkah tertentu ke langkah berikutnya pada halaman yang sama[33], [34].

Tabel 2.2 Simbol *Flowchart*

No	Simbol	Keterangan
7		<i>Off-page Connector</i> : Menunjukkan titik koneksi antara halaman atau bagian flowchart yang berbeda[31].
8		<i>Flow</i> : Menunjukkan arah alur dari satu langkah ke langkah berikutnya dari sebuah program[31].
9		Dokumen : Mewakili dokumen atau laporan[32].
10		<i>Predefined Process</i> : Mewakili sebuah proses atau serangkaian langkah-langkah yang telah ditentukan sebelumnya dan tidak memerlukan rincian lebih lanjut dalam <i>flowchart</i> . Ini dapat mencakup subrutin, fungsi, atau prosedur yang sudah terdefinisi dengan jelas[32].
11		Mewakili penyimpanan data, seperti <i>database</i> [33].

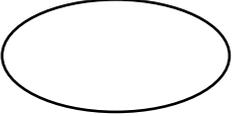
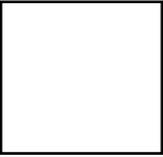
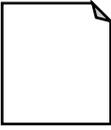
### 2.2.8 *Unified Modeling Language (UML)*

*Unified Modeling Language* merupakan sebuah teknik dalam rekayasa perangkat lunak yang digunakan untuk mengilustrasikan bagaimana sistem bekerja, fungsi-fungsi yang dimilikinya, tujuan, dan mekanisme pengendalian yang diterapkan[35]. UML memiliki kemampuan untuk menggambarkan, menentukan, membentuk, dan mendokumentasikan berbagai aspek dari sistem perangkat lunak. Dengan analogi, mirip dengan peran seorang arsitek yang merancang dokumen desain untuk perusahaan konstruksi membangun suatu bangunan, arsitek perangkat lunak menggunakan diagram UML untuk membimbing para pengembang dalam membangun perangkat lunak[36]. Pada penelitian ini terdapat empat jenis diagram UML yang digunakan, yang melibatkan Diagram *Use Case*, Diagram *Sequence*, Diagram *Class*, dan *Activity Diagram*. Berikut merupakan penjelasan dari jenis diagram UML:

### 2.2.8.1 Use Case Diagram

*Use Case Diagram* adalah gambaran visual dari perilaku sistem informasi yang sedang dibangun, menjelaskan kegunaan sistem dari perspektif individu di luar sistem atau yang dikenal sebagai aktor. Sebagai suatu grafik, *diagram use case* memberikan gambaran tentang bagaimana sistem akan digunakan dan memberikan keuntungan dari perspektif para aktor atau pihak eksternal[35], [36].

Tabel 2.3 Simbol *Use Case*

No	Simbol	Keterangan
1		Aktor: Mewakili entitas diluar sistem yang berinteraksi dengan sistem. Aktor bisa menjadi pengguna sistem atau sistem eksternal lainnya[37], [38]
2		<i>Use Case</i> : Mewakili suatu fungsi atau aktivitas yang dilakukan oleh sistem. <i>Use case</i> menggambarkan suatu fungsionalitas yang dapat dijalankan oleh satu atau lebih aktor[37].
3		<i>Association</i> : Menunjukkan interaksi aktor dalam suatu <i>use case</i> tertentu[37].
4		<i>Extend</i> : Mengindikasikan bahwa suatu <i>use case</i> mampu memperluas fungsi dari <i>use case</i> lain pada titik tertentu[37].
5		<i>System</i> : Menandai batas sistem dan memisahkan <i>use case</i> yang terdapat di dalam sistem dengan aktor yang berada di luar sistem[38].
6		<i>Collaboration</i> : Kolaborasi antara aturan dan komponen-komponen yang berfungsi bersama untuk menghasilkan tingkah laku yang lebih kompleks[39].
7		<i>Note</i> : Memberikan penjelasan atau catatan tambahan terkait <i>case</i> atau hubungan[39].

### **2.2.8.2 Class Diagram**

*Class Diagram* merupakan bagian dari jenis diagram struktur UML, difungsikan untuk dengan jelas mengilustrasikan struktur dan rincian mengenai class, termasuk atribut, metode, dan keterkaitan antar objek. Diagram ini bersifat statis, artinya penekanannya bukan pada menjelaskan bagaimana interaksi antar kelas terjadi, melainkan lebih kepada mendeskripsikan hubungan yang ada di antara kelas-kelas tersebut[40].

### **2.2.8.3 Activity Diagram**

*Activity Diagram* adalah gambaran visual yang memperlihatkan cara operasional suatu objek atau sistem. Sebuah diagram aktivitas menggambarkan secara terstruktur alur proses kerja dari *use case* yang sedang diproses, dimulai dari titik awal hingga titik akhir. Tiap aktivitas direpresentasikan dengan notasi sesuai dengan fungsinya[41].

### **2.2.8.4 Sequence Diagram**

*Sequence Diagram* adalah gambaran grafis yang menggambarkan kolaborasi dan interaksi dinamis antara sejumlah objek. Tujuannya adalah untuk memvisualisasikan serangkaian pesan yang saling dikirim antar objek serta hubungan interaktif di antara objek-objek tersebut[42], [43].

## **2.2.9 Scrum**

Scrum adalah metode perancangan atau sebuah kerangka responsif tambahan yang digunakan dalam pengembangan perangkat lunak untuk mengelola produk atau mengembangkan aplikasi[44].

Dalam metode scrum terdapat *role*, yaitu:

### 1. *Product Owner*

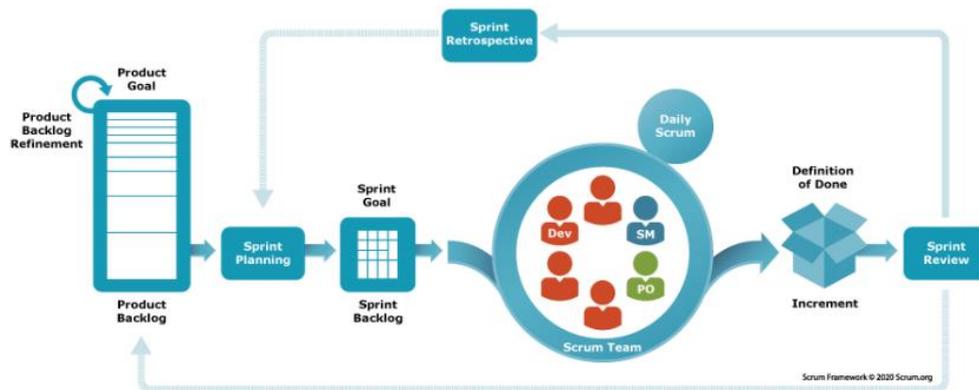
*Product Owner* merujuk pada seseorang yang bertanggung jawab untuk menyediakan informasi, kebutuhan, dan data yang diperlukan dalam pengembangan aplikasi yang diinginkan.

### 2. *Scrum Master*

*Scrum Master* merupakan individu yang memandu tim dalam mengelola proses Scrum dengan tujuan memastikan penerapan metode scrum berjalan dengan baik.

### 3. Tim Scrum

Tim Scrum adalah tim yang melaksanakan seluruh rangkaian proyek, termasuk analisis bisnis, pengembangan, serta pengujian sesuai dengan arahan dan keinginan *Product Owner*.



Gambar 2.5 Tahapan Siklus Scrum[45]

Pada gambar 2.1 merupakan tahapan dari siklus Scrum Model yang diperoleh dari scrum.org dengan tahapan *product backlog*, *sprint planning*, *daily scrum*, *increment*, *sprint review* atau kembali pada *sprint retrospective* atau *product backlog* sesuai kebutuhan yang diperlukan. Berikut penjelasan dari tahapan yang akan dilaksanakan[44], [46]:

#### 1. *Product Backlog*

Membuat rincian prioritas pada fitur-fitur yang akan dibuat dan merupakan kegiatan yang akan dilakukan selama *sprint* dimulai.

## 2. *Sprint Planning*

Pada tahap ini, dilakukan perencanaan dan penjadwalan kegiatan yang akan dilakukan sesuai dengan prioritas dalam *Backlog*, selama periode waktu yang telah ditentukan. Dan selanjutnya membuat *sprint backlog* yang didalamnya menjabarkan setiap fitur yang akan diterapkan menjadi *task-task* yang lebih detail.

## 3. *Sprint Backlog*

Merupakan rencana yang dibuat sebagai gambaran tentang pekerjaan yang direncanakan pengembang untuk diselesaikan selama *sprint* untuk mencapai *sprint goal*. Oleh karena itu, *Sprint Backlog* perlu memiliki detail yang memadai sehingga dapat memeriksa perkembangan progress dalam *daily scrum*.

## 4. *Daily Scrum (Sprint Execution)*

Pada tahap ini, melakukan pengembangan aplikasi sesuai dengan *sprint planning* dan memeriksa kemajuan dengan membahas pekerjaan yang telah dilakukan serta membahas kendala yang mungkin muncul selama proses pengembangan proyek.

## 5. *Increment*

*Increment* merupakan titik langkah konkret menuju *Product Goal*. Setiap *increment* adalah penambahan terhadap *increment* sebelumnya dan menjalani pengujian komprehensif untuk memastikan bahwa semua *increment* berinteraksi dengan baik.

## 6. *Sprint Review*

Pada tahap ini, memperlihatkan hasil aplikasi kepada *Product Owner* dan *Scrum Master* untuk menilai apakah hasilnya sudah sesuai dengan daftar kebutuhan produk. Jika ada kesalahan atau tambahan fitur yang diperlukan, ini akan dimasukkan ke dalam sprint berikutnya.

## 7. *Sprint Retrospective*

Tahapan terakhir adalah evaluasi yang dilakukan oleh tim Scrum setelah selesai membuat aplikasi. Melakukan evaluasi apakah ada hambatan selama proses pengerjaan, dengan tujuan untuk meningkatkan Sprint berikutnya menjadi lebih baik[44].

### 2.2.10 *Black-box Testing*

*Black-box testing* adalah metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas perangkat lunak tanpa memeriksa secara detail desain dan kode program. Dalam pengujian ini, penekanan diberikan pada aspek-aspek fungsional perangkat lunak, termasuk pemeriksaan kesesuaian fungsi, *input*, dan *output* perangkat lunak dengan spesifikasi yang telah ditetapkan, baik pada *level* unit maupun ketika perangkat lunak telah terintegrasi[47].

Pengujian BlackBox dilakukan untuk memastikan bahwa aplikasi berfungsi sesuai dengan harapan setelah dieksekusi dan untuk mengidentifikasi serta mengatasi kekurangan aplikasi sebelum digunakan oleh pengguna.