

## **BAB II**

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

#### **2.1 Tinjauan Pustaka**

Penelitian yang akan dilakukan oleh peneliti yang berkaitan dengan judul membuat rancangan aplikasi solusi layanan jasa pada mitra OBRE-JEK, maka terdapat perbedaan dengan penelitian sebelumnya. Penelitian ini dilihat dari segi permasalahan yang diangkat, metode yang digunakan, atau komponen yang digunakan dalam penelitian.

Penelitian yang ditulis oleh Dianaros Pakel, Rachmadita Andreswari, dan Faishal Musfied Al Anshary pada tahun 2018, dengan judul “Pengembangan Aplikasi Mobile *Commerce* Jasa Dekorasi Berbasis Android Pada StartUp Balon Partner” penelitian ini menggunakan metode *Scrum*, permasalahan pada penelitian ini adalah *StartUp* Balon Partner masih menginputkan rincian pemesanan jasa dekorasinya secara manual melalui sosial media sehingga memakan waktu cukup lama serta sering terjadinya kesalahan dalam proses pengisian data[7].

Penelitian yang ditulis oleh Yafet Yahya Wagarefe dan Joko Suwarno pada tahun 2021, dengan judul “Aplikasi Sistem Pemesanan Laundry Berbasis Android Studi Kasus Bayu Laundry” penelitian ini menggunakan metode *Scrum*, sementara itu permasalahan pada penelitian ini pemilik usaha Bayu Laundry masih menggunakan sistem manual yang mengakibatkan muncul berbagai bentuk permasalahan lain seperti kesulitan saat mencari data dari setiap customer yang menggunakan jasa Laundry tersebut[8].

Penelitian yang ditulis oleh Anjar Pratama Putra dan Sirojul Munir S.Si, M.Kom. pada tahun 2022, dengan judul “Pengembangan Aplikasi Web Market Place Jasa Rumah Tangga (Beresin)” penelitian ini menggunakan metode *Scrum*, sementara itu permasalahan pada penelitian ini yaitu para penyedia jasa yang ada di daerah Cibinong dan Depok tidak memiliki sebuah tempat khusus

seperti kantor ataupun tempat usaha dan kesulitan dalam mencari jasa yang disediakan[9].

Penelitian yang ditulis oleh Ridwan Setiawan pada tahun 2020 dengan judul “Perancangan Elektronik Marketplace Event Organizer” penelitian ini menggunakan metode *Scrum*, sementara itu permasalahan pada penelitian ini yaitu belum adanya media promosi untuk meningkatkan pemasaran dan jangkauan konsumen selain itu dikarenakan keterbatasan jarak dan lokasi membuat konsumen sulit mencari jasa *Event Organizer* yang sesuai dengan kebutuhan[10].

Penelitian yang ditulis oleh Marcelliano R. A. Omega<sup>1</sup>, Gladly C. Rorimpandey, dan Vivi P. Rantung dengan judul “Aplikasi Yunit Laundry Menggunakan Framework *Scrum*” permasalahan pada penelitian ini yaitu belum ada penerapan teknologi informasi di usaha mikro kecil yang mengakibatkan kendala saat menghadapi permintaan yang semakin meningkat dari konsumen[11].

Tabel 2. 1 Penelitian Sebelumnya

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
1.	Pengembangan Aplikasi Mobile Commerce Jasa Dekorasi Berbasis Android Pada StartUp Balon Partner[7]	2018	StartUp Balon partner memiliki masalah dalam pemasaran produk dan jasa yang disediakan dan kesulitan dalam menjangkau konsumen.	Metode <i>Scrum</i>	Berdasarkan hasil penelitian Aplikasi Mobile Commerce Jasa Dekorasi Berbasis Android pada StartUp Balon Partner berhasil dikembangkan dan di implementasikan dan di uji coba menggunakan black box testing dan dapat berjalan dengan baik dan normal	Perbedaan pada penelitian adalah mengenai studi kasus mengenai aplikasi Mobile Commerce hanya untuk jasa dekorasi dari StartUp Balon Partner dan berfokus pada sisi konsumen sedangkan penelitian yang di lakukan peneliti yaitu membuat Aplikasi Layanan jasa dari sudut pandang mitra atau pelaku usaha untuk mengelola layanan jasa mitra OBRE-JEK.
2.	Aplikasi Sistem Pemesanan Laundry Berbasis Android Studi Kasus Bayu Laundry[8]	2021	Proses pencatatan Pesanan pada Bayu Laundry yang masih berupa fisik dan manual.	Metode <i>Scrum</i>	Berdasarkan hasil penelitian mampu mengimplementasikan aplikasi pemesanan jasa laundry pada usaha Bayu Laundry dengan kebutuhan fungsional untuk pemesanan jasa laundry.	Perbedaan penelitian dari Segi mengenai rancangan pengembangan aplikasi pemesanan laundry yang merancang aplikasi untuk Bayu laundry sedangkan penelitian yang di lakukan peneliti berfokus kepada perancangan aplikasi untuk

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
						mitra atau pelaku usaha OBRE-JEK kabupaten Brebes, Jawa Tengah.
3.	Pengembangan Aplikasi Web Market Place Jasa Rumah Tangga (Beresin)[9]	2022	Para warga Cibinong dan Depok yang menyediakan jasa kesulitan dalam mencari konsumen untuk jasa yang disediakan dan begitu juga sebaliknya.	Metode <i>Scrum</i>	Berdasarkan hasil penelitian aplikasi marketplace jasa rumah tangga (Beresin) mampu mengimplementasikan aplikasi yang dapat mengatasi permasalahan warga Cibinong dan Depok dalam mencari penyedia jasa dan membantu penyedia jasa dalam memasarkan layanan jasa rumah tangga.	Perbedaan penelitian yang dilakukan adalah yaitu mengenai rancangan pembuatan aplikasi yang dikembangkan dalam bentuk <i>Website</i> sedangkan penelitian yang diteliti peneliti dirancang dalam bentuk aplikasi mobile dengan sistem operasi <i>Android</i> .

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
4.	Perancangan Elektronik Marketplace Event Organizer[10]	2023	Permasalahan Konsumen yang mengalami kesulitan dan banyak kendala seperti pada saat mencari jasa <i>Event Organizer</i> .	Metode <i>Scrum</i>	Berdasarkan penelitian tersebut menghasilkan <i>e-marketplace Event Organizer</i> yang diimplementasikan dalam bentuk <i>website</i> dengan hasil uji kelayakan <i>website</i> tersebut sudah berjalan dan dapat digunakan.	Perbedaan pada penelitian adalah pada <i>study case</i> Perancangan Elektronik Marketplace <i>Event Organizer</i> dan implementasi dalam bentuk <i>website</i> . penelitian yang akan di lakukan peneliti yaitu membuat rancangan aplikasi layanan jasa pada mitra OBRE-JEK Berbasis <i>Android</i> .
5.	Aplikasi Yunit Laundry Menggunakan Framework Scrum[11]	2022	perkembangan UMK yang ada di Sulawesi Utara khususnya laundry belum diiringi dengan adanya penerapan teknologi yang bisa menopang operasional para pengusaha laundry.	Metode <i>Scrum</i>	Berdasarkan hasil penelitian Aplikasi Yunit Laundry berhasil untuk di implementasikan dengan baik dan mempermudah pengelolaan jasa laundry dan pencarian jasa laundry bagi konsumen.	Perbedaan pada penelitian adalah pada <i>study case</i> Aplikasi Yunit Laundry Menggunakan Framework Scrum. Sedangkan penelitian yang akan dilakukan peneliti yaitu merancang aplikasi layanan jasa pada mitra OBRE-JEK

Pada tabel 2.1 mengenai penelitian sebelumnya membandingkan permasalahan yang diteliti oleh peneliti dengan kata kunci sesuai dengan judul penelitian seperti aplikasi layanan jasa, rancang bangun, pemesanan dan *E-commerce*.

## 2.2 Landasan Teori

### 2.2.1 OBRE-JEK



Gambar 2. 1 Poster OBRE-JEK

OBRE-JEK (Ojek *Online* Brebes) adalah mitra yang bergerak dibidang jasa layanan ojek *online* lokal pada kabupaten Brebes, Jawa Tengah. Ojek *online* tersebut sudah berjalan kurang lebih satu tahun beroperasi di Brebes. OBRE-JEK menyediakan layanan ojek motor, mobil barang, makanan, dan lainnya. OBREJEK memiliki tujuan untuk meningkatkan taraf pendapatan masyarakat Brebes dengan ojek lokal.

OBRE-JEK menggunakan aplikasi *whatsapp* sebagai pemesanan jika melakukan *order* layanan tersebut. OBRE-JEK ini merupakan *platfrom* sosial yang dikelola dengan tujuan membantu meningkatkan taraf pendapatan ojek lokal, dan menyediakan solusi dari umkm lokal yang menyediakan produk dan layanan jasa yang dikelola.

### **2.2.2 Layanan Jasa**

Layanan Jasa merupakan kegiatan atau aktivitas yang dilakukan oleh penyedia jasa yang memberikan manfaat kepada konsumen. Pelayanan jasa dapat berupa penggunaan keterampilan, pengetahuan dan rasa menyenangkan yang diberikan dalam memenuhi kebutuhan dan keinginan konsumen[12]. Oleh karena itu, kualitas pelayanan jasa yang baik dapat mempengaruhi kepuasan konsumen dan keberhasilan Perusahaan[13].

### **2.2.3 E-commerce**

*E-commerce*, singkatan dari elektronik commerce, merupakan model bisnis yang digunakan perusahaan atau orang untuk membeli atau menjual produk dan jasa secara online[14]. Selain itu *E-commerce* memberikan banyak manfaat, seperti memberikan kenyamanan saat seseorang untuk berbelanja atau pemesanan jasa tanpa harus datang ke toko fisik, dan mempermudah seseorang pemilik bisnis dalam mempromosikan bisnis dan untuk memperluas pasar untuk mencari customer[15].

### **2.2.4 Android**

*Android* adalah sebuah sistem operasi yang dikembangkan oleh Google untuk perangkat *mobile*. Android merupakan salah satu sistem operasi *mobile* yang paling populer di dunia[16]. Selain itu *Android* memberikan pendekatan komperhensif dalam pengembangan aplikasi, yang berarti sebuah aplikasi android yang dikembangkan dapat berfungsi di berbagai perangkat yang menggunakan sistem operasi *Android*, termasuk *smarphone*, *smartwatch*, tabelt, dan perangkat lainnya[17].

### **2.2.5 Unifed Modeling Language (UML)**

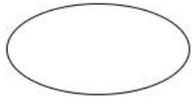
*Unified Modeling Language* (UML) adalah representasi visual dari model yang digunakan untuk membangun, merancang, menggambarkan, menjelaskan secara rinci, dan mendokumentasikan desain dari suatu sistem

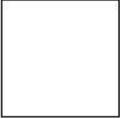
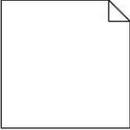
perangkat lunak. Beberapa jenis diagram UML antara lain Diagram *Use Case*, Diagram *Sequence*, dan Diagram *Class*. UML merupakan kumpulan diagram yang memberikan kerangka kerja untuk perancangan sistem perangkat lunak yang berfokus pada pendekatan berbasis objek[18]. Berikut merupakan penjelasan jenis diagram UML:

### 2.2.5.1 Use Case Diagram

*Use Case* Diagram adalah diagram yang menggambarkan hubungan antara sistem dan aktor. Diagram ini mengilustrasikan interaksi antara pengguna dengan sistem dengan menunjukkan bagaimana aktor terlibat dalam berbagai skenario penggunaan yang direpresentasikan dalam use case[19]. Pada dasarnya *Use Case* Diagram memvisualisasikan hubungan antara aktor (pengguna) dan fungsionalitas sistem (use case).

Tabel 2. 2 Simbol Use Case

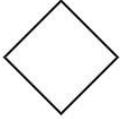
No	Simbol	Keterangan
1	 Aktor/ <i>Actor</i>	Menyatakan peran pengguna yang berinteraksi dengan sistem.
2	 Use Case	Urutan aksi yang ditampilkan sistem untuk hasil yang terukur bagi suatu <i>actor</i> .
3	 Association	Menghubungkan <i>object</i> satu dengan <i>object</i> yang lainnya.

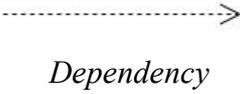
No	Simbol	Keterangan
4	 <i>Extend</i>	Menspesifikasikan usecase dengan memperluas perilaku <i>usecase</i> pada titik yang diberikan.
5	 <i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6	 <i>Collaboration</i>	Interaksi aturan dan elemen yang bekerja untuk menyediakan perilaku yang lebih besar.
7.	 <i>Note</i>	Elemen fisik saat aplikasi dijalankan dan mencerminkan sumber daya komputasi.

### 2.2.5.2 Class Diagram

*Class diagram* adalah penjelasan dari suatu proses pembuatan *database* yang akan digunakan suatu program[20].

Tabel 2. 3 Simbol *Class Diagram*

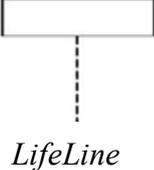
No	Simbol	Keterangan
1	 <i>Generalization</i>	Hubungan dimana objek turunan ( <i>descendent</i> ) mengadopsi perilaku dan struktur data dari objek yang merupakan objek induk ( <i>ancestor</i> )
2	 <i>Nary Association</i>	Upaya menghindari untuk asosiasi lebih dari 2 objek.
3	 <i>Class</i>	Himpunan objek-objek dengan atribut dan operasi yang sama.
4	 <i>Collaboartion</i>	Aksi-aksi yang ditampilkan sistem dengan hasil yang terukur bagi aktor.
5	 <i>Association</i>	Menghubungkan objek satu dengan objek yang lainnya.

6		Hubungan perubahan yang terjadi secara mandiri yang akan mempengaruhi elemen yang tidak mandiri.
---	---	--

### 2.2.5.3 Sequence Diagram

*Sequence Diagram* adalah sebuah diagram yang menggambarkan cara objek berinteraksi, dengan menunjukkan bagaimana objek-objek tersebut berkomunikasi dan berhubungan satu sama lain[21].

Tabel 2. 4 Simbol Sequence Diagram

No	Simbol	Keterangan
1		Entity yang berinteraksi melalui antarmuka.
2		Deskripsi dari komunikasi antar objek yang mencakup informasi terkait aplikasi yang sedang berlangsung.
3		Deskripsi dari komunikasi antar objek yang mencakup informasi terkait aplikasi yang sedang berlangsung.

#### 2.2.5.4 Activity Diagram

*Activity diagram* merupakan diagram yang menggambarkan aliran kegiatan dalam perancangan program, mulai dari tahap awal hingga tahap akhir, dengan mempertimbangkan berbagai kemungkinan yang mungkin terjadi[21]. *Activity diagram* digunakan untuk mengilustrasikan secara terstruktur kerangka aliran dalam perancangan sistem dengan cermat.

Tabel 2. 5 Simbol Activity Diagram

No	Simbol	Keterangan
1	 <i>Activity</i>	Menunjukkan aktivitas yang dilakukan sistem.
2	 <i>Action</i>	Menunjukkan state dari sistem yang mengeksekusi suatu aksi.
3	 <i>Initial Node</i>	Titik awal objek dari aktivitas atau alur kerja.
4	 <i>Activity Final Node</i>	Titik akhir dari aktivitas atau alur kerja.
5	 <i>Fork Node</i>	Merupakan cabang atau pembagian alur kerja.

### **2.2.6 Database**

*Database* merupakan suatu perorganisasian sekumpulan data atau informasi yang saling terkait dan disimpan secara elektronik, yang dapat diakses melalui program-program komputer untuk mendapatkan informasi yang dibutuhkan[22]. *Database* membantu mengurangi biaya perangkat, melindungi keamanan data dengan baik, efisien dan efektif dalam pengelolaan informasi, mendukung pengguna multi, serta memudahkan proses pengembangan aplikasi serta pengelolaan data secara terpusat[23]. Peneliti menggunakan *Firebase* sebagai *Database* untuk mengatur informasi yang digunakan dalam penelitian.

#### **2.2.6.1 Firebase**

*Firebase* adalah layanan yang disediakan oleh Google dengan tujuan memberikan kemudahan dan menyederhanakan proses pengembangan aplikasi bagi para pengembang.[24] *Firebase*, yang juga dikenal sebagai BaaS (Backend as a Service), merupakan solusi yang diberikan oleh Google untuk mempercepat proyek pengembangan oleh para developer[25].

### **2.2.7 Kotlin**

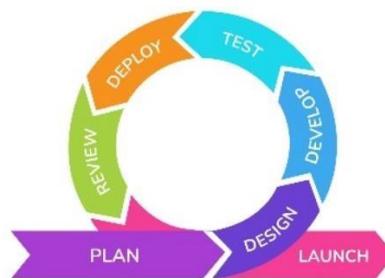
*Kotlin* adalah bahasa pemrograman modern yang merupakan salah satu Bahasa pemrograman yang populer dan sangat diminati oleh para pengembang, terutama yang berfokus pada pengembangan aplikasi Android[26]. *Kotlin* dikembangkan oleh JetBrains, Selain itu *Kotlin* mempunyai keunggulan yaitu bersifat ringan dan juga *open source*, mempunyai sintaks yang simpel dan ringkas, lebih aman dari *error*, dan terintegrasi dengan *Android Studio*. [27].

### 2.2.8 *Android Studio*

*Android Studio* merupakan salah satu *Integrated Development Environment* (IDE) yang digunakan oleh developer dalam pengembangan aplikasi *Android* yang berdasarkan IntelliJ IDEA[26]. *Android Studio* mempunyai berbagai fitur yang meningkatkan produktivitas dalam saat pengembangan aplikasi *Android* di antaranya bersifat *open source*, ramah terhadap pengembang pemula maupun yang berpengalaman, emulator yang cepat dan kaya fitur dan terintegrasi dengan github yang mempermudah dalam dokumentasi kode[28].

### 2.2.9 *Agile*

*Agile* merupakan metode dalam manajemen proyek yang menggunakan siklus pengembangan yang singkat atau dikatakan dengan *Sprint* dengan tujuan untuk meningkatkan dalam suatu sistem pengembangan produk dan layanan. Pengembangan *Agile* memiliki banyak model dan kerangka kerja diantaranya *Scrum*, *Extreme Programming (XP)*, *Dynamic Systems Development*, *Adaptive Software Development*, *Feature Driven Development* dan *kanban*[29].



Gambar 2. 2 Siklus Agile Model

Pada gambar 1.2 Siklus dalam metodologi Agile adalah serangkaian tahap yang mencakup *planning*, *design*, *development*, *test*, *deploy*, *review*, dan jika sudah tidak diperlukan, maka dilanjutkan dengan *launch* produk atau sistem yang telah dikembangkan. Model Agile ini beroperasi secara

berkelanjutan, menyesuaikan diri dengan permintaan backlog yang terus berkembang.

### 2.2.10 *Scrum* Model

*Scrum* adalah kerangka kerja yang digunakan untuk mengembangkan suatu produk yang kompleks dan dapat berubah sesuai dengan kebutuhan, Selain itu *Scrum* memiliki visi yang berfokus pada pembuatan produk yang berkualitas tinggi secara kreativitas maupun produktivitas[30]. *Scrum* merupakan model kerangka kerja yang didasari dengan teori empirisme, yang mengutamakan bahwa pengetahuan diperoleh dari hasil pengalaman sehingga dalam pengambilan keputusan juga harus berdasarkan atas pengetahuan tersebut[31]. *Scrum* didasarkan pada tiga pilar yang penting yaitu:

#### A. Transparansi

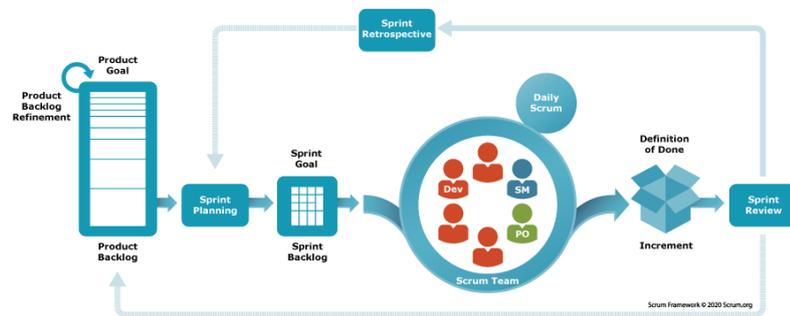
Transparansi merupakan prinsip utama dalam *Scrum* yang menekankan pada kebutuhan Proses dan hasil kerja yang muncul dapat diketahui saat sedang melakukan pekerjaan maupun saat menerima hasil kerja. *Scrum* memiliki tiga artefak formal yang menjadi keputusan penting yaitu dengan dasar yang memiliki transparansi yang minim dapat menghasilkan keputusan yang memperkecil nilai dan memperbesar risiko.

#### B. Inspeksi

Artefak *scrum* yang merupakan tujuan yang telah disepakati harus sering diperiksa untuk mengidentifikasi masalah yang terjadi secara seksama.

#### C. Adaptasi

*Scrum* Team diharapkan dapat beradaptasi secepatnya untuk mempelajari hal yang baru melalui inspeksi.



Gambar 2. 3 Siklus *Scrum* Model

Pada gambar 1.3 ini adalah siklus dari *Scrum* Model diperoleh dari [scrum.org](http://scrum.org) dengan beberapa tahap yaitu *product backlog*, *Sprint Planning*, *Daily Scrum*, *Increment*, *Sprint Review* atau Kembali ke *Sprint Retrospective* atau *Product Backlog* sesuai dengan kebutuhan yang dibutuhkan dalam proses pengembangan sistem berikut penjelasannya:

#### A. *Product Backlog*

*Product Backlog* adalah sebuah kumpulan atau susunan elemen-elemen penting yang dibutuhkan untuk menyempurnakan sebuah produk. *Product backlog* adalah tanggung jawab product owner karena daftar fitur yang diinginkan oleh product owner berada dalam kendalinya. Sebelum menyusun *product backlog*, diperlukan penentuan user story. User story adalah deskripsi singkat yang menggambarkan apa yang diinginkan oleh product owner terkait sistem. Para developer bertanggung jawab untuk menentukan skala prioritas tugas-tugas yang telah disepakati bersama product owner berdasarkan tingkat urgensinya. Prioritas fitur-fitur ini dibagi menjadi tiga kategori utama:

1. High: Sistem yang dibuat adalah fitur utama atau inti dari aplikasi yang dibangun.

2. Medium: Sistem yang dibuat adalah fitur penting yang harus ada.
3. Low: Fitur tambahan yang bersifat opsional; kehadirannya akan membuat sistem menjadi lebih baik.

#### B. *Sprint Planning*

*Sprint Planning* adalah proses perencanaan untuk mengatur tugas-tugas yang harus dilakukan pada awal *Sprint* sesuai dengan *Product Backlog*. Pada *Sprint* satu bulan, perencanaan *Sprint* harus diselesaikan dalam jangka waktu maksimal delapan jam. Pembagian waktu dalam *Sprint* disesuaikan dengan kebutuhan proyek, dengan durasi yang dapat berkisar dari dua minggu hingga empat minggu (satu bulan). *Sprint* juga bisa berdurasi satu minggu per siklus, semuanya tergantung pada kebutuhan yang ada.

#### C. *Sprint Backlog*

*Sprint backlog* adalah rancangan untuk developer, yang memiliki *timeline* pekerjaan yang dirancang oleh developer selama *Sprint* untuk memenuhi *Sprint goal*. *Sprint backlog* harus memiliki detail yang memadai sehingga dapat diperiksa saat *daily scrum*. Pada tahap *Sprint backlog*, penentuan story point digunakan untuk menggambarkan estimasi keseluruhan usaha yang diperlukan untuk menyelesaikan suatu *product backlog*. Dalam menentukan story point, terdapat beberapa pendekatan yang digunakan, seperti berdasarkan satuan waktu, tingkat kerumitan, dan pendekatan lainnya yang disesuaikan dengan jenis pekerjaan. Story point dapat ditentukan menggunakan deret Fibonacci, yang terdiri dari angka 1, 2, 3, 5, 8, dan seterusnya. Ada beberapa faktor yang perlu diperhatikan dalam menentukan story point, yaitu:

1. Jumlah pekerjaan yang akan dilakukan.
2. Tingkat kerumitan pekerjaan.
3. Risiko ketidakpastian.

#### D. *Daily Scrum*

*Daily Scrum* bertujuan untuk mengecek progress terhadap *Sprint* goal dan penyesuaian *Sprint* backlog, serta perencanaan terhadap rancangan kerja di waktu yang akan datang. *Daily scrum* digunakan untuk memperkuat komunikasi, mengidentifikasi kendala, memotivasi pengambilan keputusan yang cepat, dan mengurangi kebutuhan akan pertemuan tambahan.

#### E. Increment

*Increment* adalah langkah menuju produk goal, *increment* adalah pelengkap untuk semua *increment* sebelumnya dan telah divalidasi secara keseluruhan dengan menjamin fungsionalitas semua *increment* bekerja dengan bersamaan.

#### F. *Sprint* Review

*Sprint* Review adalah proses pemeriksaan hasil *Sprint* dan penentuan penyesuaian berikutnya. *Sprint* Review mempunyai batas waktu empat jam untuk *Sprint* selama satu bulan.

#### G. *Sprint* Retrospective

*Sprint* Retrospective merupakan salah satu cara untuk meningkatkan kualitas dan keefektifan. *Sprint* retrospective dilakukan oleh tim *scrum* untuk meninjau kembali bagaimana proses *Sprint* terakhir yang berkaitan dengan individu, interaksi, proses, alat, dan Definition of done. *Sprint* retrospective mempunyai batas waktu tiga jam untuk *Sprint* selama satu bulan.

*Scrum* memiliki 3 role yaitu *product owner*, *scrum master*, dan *development team*. Berikut penjelasan dari *role* yang berada pada *scrum*:

A. Product Owner

*Product Owner* adalah orang yang mempunyai kekuasaan atau pengetahuan tentang produk yang akan dikembangkan.

B. Scrum Master

*Scrum master* adalah orang yang bertanggung jawab sebagai pemimpin yang memiliki keahlian seputar *scrum* untuk memandu *scrum team* agar dapat bekerja dengan tujuan yang tepat dalam proses pengembangan perangkat lunak..

C. Developer

*Developer* adalah orang yang mengimplementasikan produk dalam pengembangan suatu sistem tersebut ke dalam program sesuai dengan spesifikasi yang dibutuhkan oleh sistem tersebut.

*Scrum model* mempunyai pendekatan yang diterapkan dalam kegiatan pengembangannya, yaitu *Backlog* yang merupakan daftar kebutuhan sistem yang diinginkan oleh klien, *Sprints* adalah fase-fase kerja yang digunakan dalam pengembangan sistem sesuai dengan kebutuhan yang ada pada daftar di *backlog* dengan waktu pada *time-box*, *Scrum meeting* adalah kegiatan rapat atau pertemuan rutin untuk mengevaluasi pekerjaan, kendala, dan target apa yang akan dikerjakan berikutnya, dan *demo* adalah kegiatan peninjauan ulang terkait penyempurnaan perangkat lunak kepada klien[32].

### **2.2.11 Pengujian Sistem**

Pengujian sistem merupakan proses penting untuk mengetahui adanya kemungkinan adanya kesalahan pada sistem yang dikembangkan dan untuk mengurangi terjadinya kegagalan pada sistem. Pengujian sistem berfungsi untuk mengurangi kesalahan yang ada sehingga sistem dapat diperbaiki agar menjadi semakin baik. Pengujian sistem aplikasi ini di uji menggunakan metode *Black Box Testing*[33].

### **2.2.12 System Usability Scale**

Metode Penilaian *Usability System* (SUS) adalah pendekatan evaluasi yang digunakan untuk mendapatkan pandangan subjektif dari pengguna terkait seberapa efektif dan mudah digunakan suatu sistem. Skor yang dihasilkan dari SUS dapat digunakan sebagai kriteria untuk menilai apakah sebuah aplikasi layak digunakan. Kuesioner SUS terdiri dari 10 pernyataan yang dirancang untuk mengevaluasi pengalaman pengguna dengan sistem atau produk tertentu. Kuesioner ini mencakup pertanyaan positif dan negatif yang dirancang untuk menggali berbagai aspek usability dan kepuasan pengguna, dengan tujuan untuk memberikan gambaran menyeluruh tentang pengalaman pengguna secara subjektif.[34]. Pernyataan dengan nomor ganjil menyoroti aspek-aspek yang dianggap positif atau menguntungkan, sementara pernyataan dengan nomor genap menggambarkan aspek-aspek yang dianggap negatif atau mengganggu. Saat mengisi survei, peserta diminta memberikan tanggapan menggunakan skala Likert yang meliputi pilihan dari "sangat setuju", "setuju", "netral", "tidak setuju", hingga "sangat tidak setuju".

### **2.2.13 Black Box Testing**

*Blackbox Testing* adalah salah satu metode untuk menguji perangkat lunak yang telah diselesaikan. Pada dasarnya *Blackbox Testing* tidak menguji desain, kode program, melainkan hanya untuk menguji fitur-fitur

yang berfungsi dan *output* dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan[35]. Selain itu *Black box Testing* berfungsi untuk mendeteksi kelemahan dari sistem sehingga dapat mengetahui *output* yang sesuai dengan *input* yang diinginkan setelah di eksekusi dan untuk menghindari adanya kekurangan dan error pada aplikasi sebelum digunakan oleh *user*[36].