

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Dalam penyusunan penelitian ini, peninjauan terkait penelitian yang pernah dilakukan sebelumnya mengenai rancang bangun sistem informasi dengan berbagai metode pengembangan dan pengujian sistem yang berbeda-beda adalah hal yang penting. Pada penelitian ini penulis meninjau sebanyak 7 penelitian sebagai bahan acuan pemecahan masalah.

Pada penelitian yang berjudul “Aplikasi Manajemen Data Peminjaman dan Pengembalian Mobil di Pusfatja Lapan ” yang dilakukan oleh Priyo Utomo, Lukman dan Nia Damayanti, mengidentifikasi permasalahan yang terdapat pada Pusfatja LAPAN adalah terjadinya kehilangan berkas pada saat pelaporan bulanan kepada pimpinan diakibatkan *human error* serta kendala yang timbul adalah pada saat proses pendataan dan sering terjadi kesalahan pencatatan dan pelaporan. Berdasarkan permasalahan tersebut sistem informasi peminjaman ini diperlukan dengan tujuan untuk memudahkan *admin* dalam memproses pengolahan data yang ada saat ini. Perangkat aplikasi yang telah dibuat dengan bahasa pemrograman *Java NetBeans 8.0.2* dan penyimpanan data pada *database MySQL* dapat memberikan kelancaran dalam proses *input* dan penyimpanan data-data serta laporan-laporan yang diberikan kepada Pimpinan perusahaan. Metode pengembangan sistem yang digunakan yaitu *waterfall*. Hasil dari penelitian ini proses peminjaman pihak perusahaan menjadi mudah dalam proses pendataan data karyawan, data peminjaman mobil karyawan serta proses pendataan data pengembalian mobil yang lebih baik dan akurat [6].

Pada penelitian yang berjudul “Rancang Bangun Aplikasi Inventaris Aset Berbasis Android” yang dilakukan oleh Vera Alviani, Nurkhalik Wahdaniel Asbara dan Magfira Tunnisa, masalah yang timbul pada sistem inventaris yang menerapkan cara pencatatan dan pengembalian secara mandiri [2]. Sistem yang tidak efisien ini menyebabkan hilangnya beberapa inventaris sekolah. Dari

permasalahan tersebut penelitian ini berhasil merancang aplikasi inventaris aset sekolah, yang digunakan untuk memantau siklus peminjaman dan pengembalian peralatan. Aplikasi yang dirancang berbasis *android* dengan menggunakan metode perancangan *waterfall* dengan menggunakan bahasa pemrograman *Java*. Pada tahap pengujian, peneliti menggunakan metode *Blackbox Testing* kepada siswa SMA Negeri 1 Mamuju.

Penelitian yang dilakukan oleh Muhammad Faisal Asrozy, Indyah Hartami Santi dan Dimas Fanny Hebrasianto Permadi yang berjudul “Pengkombinasian Metode Fifo Dan Metode Fefo Pada Sistem Aplikasi Pengeluaran Stok Barang”, bertujuan untuk mengembangkan sebuah sistem yang dapat mempermudah pengelolaan stok barang dan dapat meminimalisir risiko kerugian dalam proses jual beli yang ada di Koperasi Karyawan Gemah Ripah. Dalam penelitian ini penulis mengembangkan sistem aplikasi *e-commerce* dengan menggunakan pengkombinasian metode *First In First Out* (FIFO) dan metode *First Expired First Out* (FEFO). Aplikasi yang dikembangkan berbasis *website* dengan metode pengembangan sistem yakni metode *scrum*. Bahasa pemrograman yang digunakan yakni *PHP* dan *HTML* dengan *MySQL* sebagai media penyimpanan. Hasil yang diperoleh dari penelitian ini adalah pengkombinasian metode *First In First Out* dan metode *First Expired First Out* pada sistem aplikasi *e-commerce* ini membuat manajemen pengelolaan stok produk dapat lebih efektif karena produk yang dijual pertama adalah stok produk yang pertama masuk dan stok produk yang masa *expired*-nya pendek terlebih dahulu. Dengan begitu pengelola dapat mengurangi risiko kerugian akibat stok lama yang *expired* [7].

Penelitian yang dilakukan oleh Fitri Wibowo, Freska Rolansa dan Tri Bowo Atmojo yang berjudul “Sistem Informasi Inventaris Untuk Peminjaman dan Pengembalian Alat di Laboratorium Teknik Informatika POLNEP Berbasis Web”, mengidentifikasi permasalahan yang ada pada Inventori laboratorium yakni proses administrasi yang masih menggunakan *paper-based*

atau *hardcopy*. Selain itu sistem manajemen peminjaman yang diterapkan memiliki beberapa permasalahan seperti bahan habis pakai yang kurang *update*. Dari permasalahan tersebut peneliti membuat sebuah sistem informasi yang dapat digunakan untuk menangani peminjaman alat dan mengelola persediaan habis pakai. Teknologi yang digunakan dalam aplikasi *mobile* yang dirancang yaitu dengan menggunakan *QR Code Scanner* yang berfungsi untuk mempercepat proses pendataan barang. Aplikasi yang dirancang akan menggunakan bahasa pemrograman *python* dengan memakai *framework Django* dan metode yang digunakan dalam aplikasi ini ialah *Rapid Application Development (RAD)* [8].

Penelitian yang dilakukan oleh Adelia Sekar Apsari, Luciana Andrawina dan Hilman Dwi Anggana yang berjudul “Perancangan Sistem Informasi Manajemen Aset Kantor di Fakultas Rekayasa Industri Telkom *University* Menggunakan Metode Scrum”, menerangkan bahwa fakultas rekayasa industri yang ada di Telkom University mengalami kesulitan dalam proses pengelolaan aset. Masalah tersebut disebabkan karena tidak adanya sistem yang terintegrasi yang dapat melakukan pencatatan, *tracking* dan peminjaman aset secara *online* dan *realtime*. Sehingga dari masalah tersebut dirancanglah sebuah sistem informasi yang dapat mengintegrasikan beberapa proses tersebut dalam bentuk aplikasi berbasis *website*. Metode pengembangan sistem yang digunakan ialah metode *scrum* yang merupakan salah satu implementasi dari metode *agile*, metode *scrum* memungkinkan pengembang untuk dapat bekerja lebih *flexible* terhadap perubahan tanpa menghilangkan fokus prioritas [9]. Pengujian fungsionalitas yang ada pada sistem informasi akan menggunakan metode *Blackbox Testing* dan untuk mengetahui sistem yang dirancang sesuai dengan kebutuhan *user* akan dilakukan pengujian dengan menggunakan metode *User Acceptance Testing (UAT)*.

Penelitian yang dilakukan oleh Sigit Teguh Prakoso, Sugeng Widodo dan Yekti Asmoro Kanthi yang diterapkan di kota Malang yang berjudul

“Perancangan Sistem Informasi Peminjaman Alat Berbasis Web pada Mapala Se-Kota Malang”, menemukan bahwa Mapala yang ada di kota Malang memerlukan sebuah sistem informasi untuk mempermudah proses peminjaman peralatan dari setiap organisasi Mapala yang ada di kota tersebut dan meminimalkan kesalahan pencatatan transaksi peminjaman di kota tersebut. Dalam penelitian ini disebutkan, sistem informasi peminjaman diperlukan karena setiap organisasi Mapala yang ada saling mendukung dalam keperluan sarana dan prasarana yang diperlukan ketika ingin melakukan kegiatan, namun karena setiap organisasi Mapala memiliki inventaris yang berbeda-beda, maka diperlukan sebuah sistem terpusat untuk mendata setiap *inventory* yang ada pada setiap organisasi Mapala di kota Malang. Sistem informasi yang dirancang berbasis *website* dengan menggunakan bahasa *PHP* melalui *framework CodeIgniter*, serta *MySQL* yang digunakan sebagai *database* [10].

Pada penelitian yang berjudul “Pengembangan Front-End Pada Aplikasi M-Banking Agen46 Dengan Teknologi Flutter”, bertujuan untuk mengembangkan sistem yang ada di salah satu layanan *e-banking* milik BNI yakni BNI Agent46. Pengembangan sistem yang ingin dilakukan ialah *multiplatform* agar sistem dapat memenuhi kebutuhan pengguna dan menjangkau *user* lebih banyak lagi dan pengembangan *interface* yang agar aplikasi lebih mudah dipahami pengguna dan menghasilkan tampilan aplikasi yang lebih menarik dari sebelumnya. Dari permasalahan tersebut, peneliti berfokus pada pengembangan *front-end* dan pengintegrasian *API Framework* yang digunakan pada penelitian ini ialah *flutter*, dengan menggunakan bahasa pemrograman *Dart*. Metode pengembangan sistem yang digunakan ialah implementasi dari metode *agile* yakni *scrum*. Hasil pengujian dari aplikasi ini yang dilakukan dengan metode *Blackbox testing* menghasilkan bahwa *API* dan pengiriman data transaksi dari *backend* ke *frontend* juga sudah sesuai dengan skema pengujian yang telah dibuat [11].

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Judul Penelitian	Perbedaan	Persamaan
1	Aplikasi Manajemen Data Peminjaman dan Pengembalian Mobil di Pusfatja Lapan	Cakupan objek penelitian termasuk dalam bidang transportasi. Aplikasi yang dirancang menggunakan bahasa pemrograman <i>java</i> dan <i>software</i> NetBeans. Metode pengembangan sistem yang digunakan ialah metode <i>waterfall</i> .	Solusi yang dihadirkan berupa digitalisasi proses peminjaman yang ada berupa aplikasi berbasis <i>mobile</i> .
2	Rancang Bangun Aplikasi Inventaris Aset Berbasis Android.	Metode pengembangan aplikasi yang digunakan ialah metode <i>waterfall</i> . Bahasa pemrograman yang digunakan ialah <i>Java</i> , <i>database MySQL</i> dan dengan menggunakan <i>software Android Studio</i> .	Persoalan yang diangkat ialah mengenai permasalahan dalam hal pencatatan barang. Solusi yang dihadirkan berupa digitalisasi proses peminjaman yang ada berupa aplikasi berbasis <i>mobile</i> . Pengujian yang dilakukan dengan menggunakan metode <i>black box</i> .
3	Pengkombinasian Metode Fifo Dan Metode Fefo Pada Sistem Aplikasi Pengeluaran Stok Barang	Objek yang diteliti yakni pendataan barang masuk dan keluar untuk memudahkan laporan keuangan. Sistem aplikasi yang dikembangkan menggunakan Bahasa	Metode pengembangan sistem yang digunakan ialah metode <i>scrum</i> . Pengujian yang dilakukan dengan menggunakan metode <i>Black box</i> .

No	Judul Penelitian	Perbedaan	Persamaan
		pemrograman yakni PHP dan HTML dengan <i>MySQL</i> sebagai <i>database</i> .	
4	Sistem Informasi Inventaris Untuk Peminjaman dan Pengembalian Alat Di Laboratorium Teknik Informatika POLNEP Berbasis Web	Objek yang diteliti mengenai peminjaman alat-alat dan bahan habis pakai yang ada di laboratorium. Sistem yang dibangun berbasis <i>website</i> menggunakan <i>software</i> XAMPP, dan memakai bahasa pemrograman PHP. Metode yang digunakan dalam perancangan aplikasi ialah <i>workflow driven</i> .	Persoalan yang diangkat ialah mengenai pencatatan stok barang yang tidak <i>up-to-date</i> . Pengujian yang dilakukan untuk memastikan setiap menu dan <i>fitur</i> pada <i>interface</i> berfungsi menggunakan metode <i>Blackbox</i> .
5	Perancangan Sistem Informasi Manajemen Aset Kantor di Fakultas Rekayasa Industri Telkom University Menggunakan Metode Scrum	Penelitian ini tidak hanya mencakup peminjaman dan pendataan barang, namun juga mencakup pelacakan barang dan pengadaan barang. Sistem yang dibangun berbasis <i>website</i> .	Metode yang digunakan untuk pengembangan sistem ialah metode <i>scrum</i> . Pengujian yang dilakukan dengan menggunakan metode <i>Blackbox</i> dan <i>User Acceptance Testing</i> .

No	Judul Penelitian	Perbedaan	Persamaan
6	Perancangan Sistem Informasi Peminjaman Alat Berbasis Web pada Mapala Se-Kota Malang.	Objek penelitian yakni pendataan <i>inventory</i> pada setiap organisasi Mahasiswa Pencinta Alam (Mapala) yang ada di kota Malang. Sistem peminjaman yang dirancang berbasis <i>website</i> dengan menggunakan bahasa pemrograman PHP melalui <i>framework CodeIgniter</i> . Salah satu fitur yang ada dalam sistem peminjaman ini ialah <i>user</i> dapat melakukan transaksi atau pembayaran	Pengguna dari sistem peminjaman ini ialah mahasiswa yang tergabung dalam sebuah organisasi.
7	Pengembangan Front-End Pada Aplikasi M-Banking Agen46 Dengan Teknologi Flutter	Fokus utama pada penelitian ini ialah pengembangan fitur dengan integrasi API <i>backend</i> dengan <i>frontend</i> , dengan menerapkan transaksi <i>multiplatform</i> ke <i>e-wallet</i> . Selain pengembangan fitur penelitian ini juga berfokus terhadap pembaharuan <i>interface</i> , bukan pada digitalisasi proses yang sudah ada.	Pengembangan aplikasi ini menggunakan implementasi dari metode <i>agile</i> yaitu metode <i>scrum</i> . <i>Framework</i> yang digunakan peneliti ialah <i>flutter</i> dengan bahasa pemrograman <i>Dart</i> .

Berdasarkan penjelasan dari 7 literatur yang telah dibaca, ditemukan bahwa solusi untuk mentransformasi proses peminjaman barang di Bagian Logistik IT Telkom Purwokerto, yang masih menggunakan *hardcopy* menjadi sistem peminjaman digital adalah langkah yang tepat. Dengan adanya sistem yang terintegrasi mulai dari pencatatan peminjaman, pendataan stok, terbukti mengurangi terjadinya *miss data* dan *human error*, serta mempercepat proses penyediaan sarana dan prasarana. Selain itu, penelitian ini juga berpotensi untuk memberikan peningkatan kepada sistem informasi IT Telkom Purwokerto.

## 2.2 Dasar Teori

### 2.2.1 Logistik

Logistik adalah unit yang menangani proses *supply chain management* yang memiliki fungsi penting dalam perencanaan, pelaksanaan, dan pengendalian efektivitas dan efisiensi penyimpanan dan aliran barang, pelayanan dan informasi [12].

### 2.2.2 Bagian Logistik dan Manajemen Aset IT Telkom Purwokerto

Logistik dan Manajemen Aset merupakan unit di bawah Bidang II yang berfungsi sebagai unit penyedia sarana dan prasarana untuk mendukung proses pembelajaran serta memastikan bahwa keseluruhan aset lembaga tercatat, tertata dan berfungsi dengan baik melalui pemeliharaan yang berkesinambungan [13]. Prosedur peminjaman barang oleh mahasiswa kepada unit Logistik IT Telkom Purwokerto yakni sebagai berikut :

- a. Pengisian *form*, pada tahapan ini proses peminjaman dimulai dengan pengisian *form* pengajuan pengadaan barang dengan menyertakan nama unit, tanggal kegiatan, lembar pengesahan, nama alat, tanggal pengambilan dan jumlah kuantitas dari barang yang diajukan kepada bagian/unit logistik.



- b. Pengajuan persetujuan, selanjutnya diserahkan kepada pihak kemahasiswaan terlebih dahulu untuk disahkan sebagai permohonan lalu diserahkan kepada unit logistik.
- c. Pengecekan ketersediaan, bagian logistik selanjutnya akan memeriksa dan menentukan ketersediaan aset, apakah tersedia atau tidak sedang digunakan oleh unit lain, jika aset tersedia maka akan disetujui dan jika tidak maka akan diberitahu kepada unit yang mengajukan peminjaman.
- d. Pengambilan barang, selanjutnya pihak atau unit yang melakukan peminjaman barang/aset akan menerima barang yang sesuai dengan form pengajuan dengan ketentuan barang dapat diambil satu hari sebelum hari kegiatan
- e. Pengembalian barang, dilakukan oleh unit peminjam dengan ketentuan barang harus dikembalikan H+1 setelah kegiatan dilaksanakan.

### 2.2.3 Aplikasi *Mobile*

Aplikasi *mobile* adalah program aplikasi yang digunakan dengan sumber daya berbasis *web* yang menyediakan akses ke beragam informasi yang relevan menggunakan perangkat telepon seluler, *smartphone*, nirkabel dan perangkat lainnya [14].

### 2.2.4 *Android*.

*Android* adalah sistem operasi berbasis *linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *android, Inc.*, dengan dukungan finansial dari *Google*, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, perusahaan dengan didirikan *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler [15].

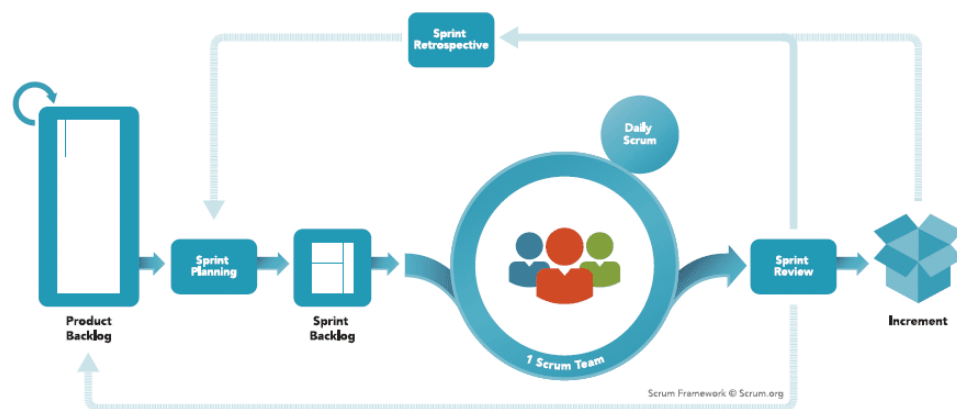
### 2.2.5 Software Development Life Cycle (SDLC)

SDLC (Software Development Life Cycle) adalah suatu pendekatan sistematis yang digunakan dalam pengembangan perangkat lunak. Pendekatan ini membantu tim pengembangan perangkat lunak dalam mengatur proses pembuatan, pengembangan, pengujian, dan pemeliharaan perangkat lunak secara efisien dan terstruktur. Tujuan dari SDLC adalah untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna, mengikuti tenggat waktu, dan berada dalam batas anggaran yang ditentukan [4].

### 2.2.6 Pengembangan Perangkat Lunak Metode *Scrum*

Metode pengembangan aplikasi *scrum* adalah metode pengembangan perangkat lunak yang *flexible* dalam menangani perubahan sistem. Selain tanggap dalam menangani perubahan, metode ini juga menggunakan prinsip untuk menekankan iterasi dalam proses pengembangan dan pengiriman produk kepada pemangku kepentingan [16].

## SCRUM FRAMEWORK



Gambar 2. 1 Tahapan Metode Scrum

Pada gambar 2.1 menampilkan Langkah-langkah yang dilakukan dalam pengembangan aplikasi menggunakan metode *scrum*, penjelasannya sebagai berikut :

- a. *Product Backlog*, pada tahapan ini peneliti membuat daftar terurut dari fitur-fitur yang harus ada dari sistem yang akan dibangun, serta menyertakan bobot tingkat kesulitan dan kegunaan dari fitur tersebut. Standar bobot yang digunakan pada *backlog* ditentukan oleh Tim Pengembang yang disepakati dengan *stakeholder* lainnya dalam hal ini *Product Owner* dan *Scrum Master* [17]. Contoh penerapannya yakni seperti pada table 2.2 dibawah ini.

Tabel 2. 2 Product Backlog

No.	Backlog Item	Bobot
1	Fitur A	1
2	Fitur B	3
3	Fitur C	5
4	Fitur D	8

- b. *Sprint planning*, selanjutnya pengembang menentukan batas waktu atau durasi untuk mengerjakan sebuah runtutan pekerjaan yang bersifat *Increment*, di mana satu siklus *sprint* memiliki batasan maksimal 30 hari. Batasan waktu yang digunakan harus bersifat konstan, sebuah *sprint* baru dimulai segera setelah akhir dari *sprint* sebelumnya telah selesai. Estimasi waktu *sprint* yang digunakan pada *sprint* pertama dapat dijadikan acuan untuk evaluasi *sprint review*, apakah waktu pengerjaan *sprint* perlu ditingkatkan atau dikurangi sesuai dengan keputusan *stakeholder* dalam *scrum*.
- c. *Sprint Backlog*, selanjutnya pengembang menentukan *product backlog* yang secara fungsional berkaitan antar satu item *backlog* dengan *backlog* lainnya untuk ditargetkan selesai dalam satu masa *sprint* yang sama.

Tabel 2. 3 Sprint Backlog

No.	Target	Backlog item	Estimasi
1	Sprint 1	Fitur A	7 hari

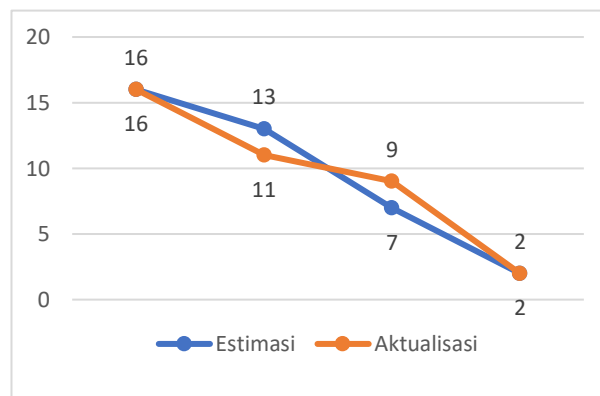
		Fitur B	
		Fitur C	
2	Sprint 2	Fitur D	7 hari
		Fitur E	

- d. *Daily Scrum*, tahap ini berupa pertemuan dengan durasi maksimal 15 menit, yang dimana pertemuan ini digunakan untuk mengawasi perkembangan yang telah dilakukan, apa hambatan yang ditemukan selama menjalani sprint dan membahas tujuan untuk pertemuan selanjutnya.

Tabel 2. 4 *Daily Scrum*

<i>Backlog ID</i>	Waktu		Pengerjaan	
	Planned	Aktual	Estimasi	Aktual
ID001	dd/mm/yyyy	dd/mm/yyyy	16	16
ID002	dd/mm/yyyy	dd/mm/yyyy	..	..
...	...	...	..	..

Dalam praktik pendokumentasian daily scrum, terdapat beberapa alat yang dapat membantu untuk memvisualisasi setiap progres dengan jelas, salah satunya ialah *burndown chart*. Burndown chart ialah cara yang digunakan untuk melihat secara sekilas bagaimana sprint sebenarnya berkembang dibandingkan dengan kecepatan kerja tim sebelumnya [18].

Gambar 2.2 *Burndown chart*

- e. *Sprint Review*, adalah fase akhir dari *sprint backlog*, untuk melihat apa saja yang telah dikerjakan dari sprint yang telah dilakukan dan menyesuaikan kembali apakah sudah sesuai dengan *sprint backlog*.

Tabel 2. 5 *Sprint Review*

No.	Target	Backlog item
1	Sprint 1	Fitur A
		Fitur B
		Fitur C
2	Sprint 2	Fitur D
		Fitur E

- f. *Sprint retrospective*, adalah tahapan yang diperlukan untuk mengevaluasi kinerja selama proses *scrum*, dimulai dari merancang *product backlog* hingga *sprint review*. Dari hasil evaluasi tersebut digunakan untuk menentukan kinerja selanjutnya.

Selain itu pengembangan aplikasi menggunakan metode *scrum* terdiri dari beberapa stakeholder yang memiliki tugas berbeda-beda, diantaranya ialah *product owner*, *scrum master* dan *developer*. Setiap *stakeholder* tersebut memiliki tugas yang berbeda, penjelasannya sebagai berikut :

a. *Product Owner*

*Product owner* bertugas untuk bertanggung jawab terhadap *product backlog*, serta pengambilan keputusan dalam menentukan prioritas dan fitur produk.

b. *Scrum Master*

*Scrum Master* adalah orang yang memastikan anggota pengembang dapat memahami prinsip-prinsip *scrum* sehingga tim dapat menjalankan proyek dengan menerapkan metode *scrum* dengan baik [17]. Prasyarat utama untuk menjadi *scrum master* ialah sebagai berikut :

- 1) Memiliki *clout*(pengaruh) & keahlian negosiasi
- 2) *People person* dan memiliki keahlian *servant leadership*

- 3) Kreatif dan memiliki keahlian memfasilitasi
- 4) Memiliki keahlian *situation awareness*
- 5) Memiliki pemikiran positif

c. Developer

Pengembang adalah orang yang mengimplementasikan fitur yang menjadi item *product backlog*, baik dari segi *design, analyst, develop, testing* dan lain sebagainya.

### 2.2.7 Figma

*Figma* adalah *design tool* untuk membuat *user interface* baik dari aplikasi *mobile, website, desktop* dan sebagainya. *Figma* bersifat *multi platform*, bisa digunakan di sistem operasi *windows* [19]. *Figma* merupakan sebuah platform yang terintegrasi dengan *cloud*, sehingga memungkinkan pengguna untuk berkolaborasi dalam *team project* untuk menghasilkan *prototype* dari produk digital dengan pengguna lainnya [20].

### 2.2.8 Bahasa Pemrograman Dart

Bahasa pemrograman *dart* adalah bahasa pemrograman yang dikembangkan oleh *Google* untuk *general-purpose programming* atau merancang aplikasi kebutuhan umum [21]. Bahasa *Dart* sangat cocok untuk pengembangan klien, dimana *dart* memprioritaskan pengembangan. Contoh fitur *Dart* yang memprioritaskan pengembangan aplikasi adalah fitur *hot reload*, di mana pengembang bisa melihat secara langsung perubahan tampilan selagi masih melakukan *coding* tanpa harus menunggu untuk *recompile* dan/atau *rebuild coding* yang sedang dibuat. *Dart* juga dapat dengan mudah mengoding untuk *web, mobile, dan juga desktop*. [22].

### 2.2.9 Flutter






Flutter merupakan sebuah *framework toolkit UI* portabel milik *Google* yang digunakan untuk membuat aplikasi yang dikompilasi secara *native* untuk seluler atau *mobile, web, dan desktop* dari satu basis kode. *Flutter*

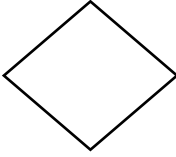


bekerja dengan kode yang ada, sudah digunakan oleh berbagai pengembang dan organisasi di seluruh dunia, dan juga gratis untuk pengguna untuk digunakan atau *open source* [23].

#### 2.2.10 Flowchart

*Flowchart* merupakan urutan logika dari suatu prosedur pemecahan masalah yang dituliskan dalam simbol-simbol tertentu. *Flowchart* bertujuan untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas. Simbol-simbol pada *Flowchart* ditunjukkan ke dalam Tabel 2.7 berikut [24].

Tabel 2.7 Simbol-simbol Flowchart


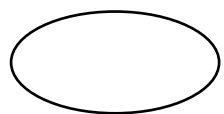
No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Awal atau akhir dari program.
2		Garis Alir	Aliran program.
3		<i>Preparation</i>	Penentuan nilai awal.
4		Proses	Pengolahan data.
5		<i>Input/Output</i>	Masukan/keluaran data, parameter, informasi.

No	Simbol	Nama	Keterangan
6		<i>Decision</i>	Perbandingan pernyataan dan penentuan pilihan untuk tindakan selanjutnya.
7		<i>On Page Connector</i>	Sambung antar bagian <i>flowchart</i> pada halaman yang sama
8		<i>Off Page Connector</i>	Sambung antar bagian <i>flowchart</i> pada halaman yang berbeda.

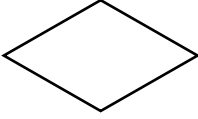

### 2.2.11 ERD

*ERD* (*Entity Relationship Diagram*) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek data yang mempunyai hubungan antar relasi [25]. Simbol-simbol pada ERD ditunjukkan kedalam Tabel 2.8 berikut.

Tabel 2.8 Simbol-simbol ERD

No	Simbol	Nama	Keterangan
1		Entity	Koleksi objek yang dapat dikenali dan didefinisikan secara unik.
2		Atribut	Properties dari entity atau relasi yang memberikan deskripsi lebih rinci.



No	Simbol	Nama	Keterangan
3		Relationship	Hubungan antar satu atau lebih entity.
4		Link	Penghubung antar himpunan, relasi dan atribut dari entity

#### 2.2.12 Firebase Realtime Database

*Firebase* merupakan *cloud service provider* yang dimiliki oleh Google yang berfungsi sebagai penyimpanan basis data *nonSQL*. *Firebase* menyediakan layanan yang ditawarkan oleh *Google* untuk mempermudah pengembangan aplikasi sebagai tempat penyimpanan yang bersifat *realtime*, yakni dengan *Firebase Realtime Database*. Data yang disimpan pada *firebase* akan diubah menjadi *node* dalam bentuk *JSON* [26].

#### 2.2.13 Blackbox Testing

Pengujian *Blackbox* merupakan pengujian yang didasarkan pada detail fungsionalitas pada aplikasi, tampilan, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh pengguna. Pengujian *Black Box* dilakukan untuk menemukan kesalahan baik dari *interface*, informasi, integrasi *API*, dan proses inisiasi dan terminasi [27]. Proses *Blackbox testing* dilakukan dengan beberapa kriteria, program yang telah dibuat dibuatkan skenario pengujian, dari skenario tersebut parameter pengujiannya ialah hasil yang diharapkan, lalu kesimpulan dari pengujian tersebut.

#### 2.2.14 User Acceptance Testing

*User Acceptance Testing* (UAT) adalah proses validasi apakah fitur-fitur yang dihadirkan dalam aplikasi bisa tergolong sebagai solusi dari sisi pengguna aplikasi nantinya. Proses UAT berbeda dengan pengujian sistem,

yang dimana tidak hanya memastikan aplikasi memenuhi spesifikasi sistem, namun tujuan proses validasi ini adalah agar pengembang aplikasi dapat memahami dari sisi pengguna apakah sudah memenuhi harapan atau tidak [28]. Untuk mendapatkan validasi dari *user*, melalui pengujian UAT memiliki beberapa jenis pengujian yakni sebagai berikut :

1. *Alpha Testing*, dilakukan oleh tim pengembang sebelum hasil perangkat lunak diserahkan kepada pengguna akhir, Alpha testing bertujuan untuk menemukan dan memperbaiki bug sebelum hasil perangkat lunak diluncurkan.
2. *Beta Testing*, merupakan pengujian yang dilakukan oleh sekelompok kecil pengguna akhir yang diambil dari pengguna nyata. Tujuan dari Beta testing ialah untuk mendapatkan umpan balik dari pengguna nyata dan menemukan bug yang terlewatkan setelah pengujian internal.
3. *Contract Acceptance Testing*, merupakan pengujian yang dilakukan untuk memastikan bahwa perangkat lunak memenuhi spesifikasi sesuai dengan kesepakatan pemangku bisnis.
4. *User Interface Testing*, bertujuan untuk memastikan bahwa antarmuka pengguna intuitif dan memenuhi harapan pengguna dari segi navigasi, kemudahan penggunaan dan estetika.
5. *Compliance Testing*, merupakan pengujian yang bertujuan untuk melakukan pengecekan apakah perangkat lunak mematuhi kebijakan perusahaan, standar keamanan dan peraturan pemerintah.
6. *Operational Acceptance Testing*, merupakan pengujian yang berfokus pada pengujian operasional seperti proses pemulihan pasca insiden, backup dan pemeliharaan sistem.
7. *Blackbox Testing*, merupakan pengujian yang berfokus pada input dan output yang digunakan oleh pengguna akhir untuk memvalidasi fungsionalitas sistem.
8. *Scenario Testing*, merupakan pengujian yang melibatkan skenario atau kasus penggunaan spesifik yang mencerminkan ketika perangkat lunak akan digunakan dalam dunia nyata.

9. *Exploratory Testing*, merupakan pengujian tidak terstruktur di mana pengguna atau penguji secara spontan mencoba beberapa fungsi perangkat untuk menemukan permasalahan tidak terduga.

#### 2.2.15 *Beta Testing Questionare*

Pengujian kuesioner digunakan untuk mengumpulkan umpan balik dari pengguna beta mengenai pengalaman mereka menggunakan produk atau perangkat lunak dalam tanpa melakukan pengujian kode. Kuesioner ini dirancang untuk mengidentifikasi masalah dan menilai kepuasan pengguna terhadap fitur-fitur yang ada yang diwakilkan dalam bobot pertanyaan yang diberikan kepada sampel pengguna akhir [29]. Beberapa tahapan yang dilakukan dalam beta testing questionare ialah sebagai berikut :

##### a. Pengisian Kuesioner

Pengujian validasi user diawali dengan pengisian kuesioner yang menggunakan skala *likert* sebagai kriteria penilaian, yang umumnya digunakan pada pembuatan kuesioner [30]. Kriteria penelitian seperti yang ditampilkan pada Tabel 2.7 berikut :

Tabel 2. 6 Kriteria Penilaian Kuesioner

<b>Jawaban</b>	<b>Bobot Nilai</b>
Sangat Setuju	5
Setuju	4
Cukup	3
Kurang Setuju	2
Sangat Tidak Setuju	1

Acuan dalam setiap pertanyaan adalah kriteria atau aspek yang digunakan untuk mengukur sejauh mana fitur atau sistem yang telah berhasil memenuhi harapan dan kebutuhan pengguna. Aspek yang menjadi acuan dalam pertanyaan yang ada pada kuesioner adalah sebagai berikut :

Tabel 2. 7 Aspek Pertanyaan Kuisisioner

No.	Pertanyaan
<b>A. Aspek Komunikasi Visual</b>	
1	Pertanyaan 1
2	Pertanyaan 2
<b>B. Aspek Fungsionalitas</b>	
4	Pertanyaan 1
5	Pertanyaan 2
<b>C. Aspek Kinerja Aplikasi</b>	
6	Pertanyaan 1
7	Pertanyaan 2

b. Penghitungan Nilai Responden

Setelah setiap responden selesai mengisi jawaban pada setiap pertanyaan yang ada pada Kuesioner, pertama dilakukan perhitungan untuk memperoleh total bobot dari setiap pertanyaan yang nantinya total bobot dari setiap pertanyaan tersebut akan digunakan untuk menghitung nilai rata-rata dari responden. Perhitungan bobot pertanyaan diperoleh dengan cara mengalikan jumlah jawaban x bobot nilai tersebut, contoh penerapannya ialah dapat dilihat pada Tabel 2.8 sebagai berikut :

Tabel 2. 8 Perhitungan Nilai Kriteria

Keterangan	Perhitungan bobot
Nilai bobot jawaban Sangat Setuju	= Jumlah jawaban x 5
Nilai bobot jawaban Setuju	= Jumlah jawaban x 4
Nilai bobot jawaban Cukup	= Jumlah jawaban x 3
Nilai bobot jawaban Tidak Setuju	= Jumlah jawaban x 2
Nilai bobot jawaban Sangat Tidak Setuju	= Jumlah jawaban x 1

Setelah memperoleh jumlah bobot dari setiap jawaban yang diberikan responden, data tersebut digunakan untuk mendapatkan nilai responden, dengan rumus perhitungan sebagai berikut :

Rumus 2. 1 Perhitungan Nilai Responden

$$\text{Nilai Responden} = \frac{\text{Jumlah Bobot Nilai Responden}}{\text{Total Responden}}$$

c. Perhitungan Hasil Kualitas Sistem

Setelah memperoleh nilai responden, maka selanjutnya nilai responden tersebut akan menjadi data nilai rata-rata untuk melakukan perhitungan presentase kualitas sistem pada setiap pertanyaan agar memperoleh hasil apakah kualitas sistem tersebut sudah layak bagi pengguna. Rumus perhitungannya ialah sebagai berikut :

Rumus 2. 2 Perhitungan Hasil Kualitas Sistem

$$\text{Presentase} = \frac{\text{Nilai rata-rata responden}}{\text{Bobot Maksimum}} \times 100\%$$

Hasil kualitas sistem, dapat diperoleh dengan menjumlahkan setiap presentase pertanyaan lalu dibagi dengan banyaknya jawaban, lalu dari hasil perhitungan presentase pertanyaan yang telah diperoleh dapat disimpulkan apakah sistem yang telah dirancang telah sesuai dengan kebutuhan pengguna berdasarkan kriteria interpretasi nilai berikut [31].

Tabel 2. 9 Kriteria Hasil Kualitas

<b>Hasil Presentase</b>	<b>Perhitungan bobot</b>
81 – 100%	Sangat baik
61 – 80%	Baik
41 – 60%	Cukup baik
21 - 40%	Kurang baik
0 - 20%	Sangat kurang baik

### 2.2.16 *Stable Testing*

*Stable testing* adalah proses evaluasi untuk memastikan perangkat lunak yang dirancang berfungsi dengan baik dan stabil tanpa ada kendala yang signifikan dan telah memenuhi standar kualitas yang diperlukan sebelum dirilis ke pengguna. Tujuan utama dari pengujian ini adalah untuk menilai kinerja dan ketahanan perangkat lunak terhadap kegagalan selama penggunaan normal maupun dalam kondisi ekstrim sebelum perangkat lunak dirilis ke lingkungan produksi atau ke pengguna akhir [29]. Bentuk pengujian dari *stable testing* ialah sebagai berikut :

- 1) *Load Testing* : Pengujian ini dilakukan untuk mengevaluasi kinerja perangkat lunak ketika dioperasikan di bawah beban tertentu yang meningkat secara bertahap. Tujuannya adalah untuk memastikan bahwa perangkat lunak dapat menangani volume kerja yang diharapkan tanpa mengalami penurunan kinerja yang signifikan.
- 2) *Stress Testing* : Pengujian tekanan menguji perangkat lunak di luar batas operasional normalnya untuk melihat bagaimana ia menangani kondisi yang ekstrem, seperti lonjakan lalu lintas data yang tiba-tiba atau penggunaan sumber daya yang berlebihan. Hal ini bertujuan untuk mengidentifikasi titik kegagalan dan memastikan perangkat lunak dapat pulih dari kondisi tersebut.
- 3) *Performance Testing* : Pengujian ini fokus pada evaluasi berbagai aspek kinerja perangkat lunak, termasuk waktu respons, throughput, dan efisiensi penggunaan sumber daya (seperti CPU dan memori). Pengujian ini bertujuan untuk memastikan bahwa perangkat lunak memenuhi spesifikasi kinerja yang telah ditetapkan.
- 4) *Reliability Testing* : Pengujian keandalan bertujuan untuk mengevaluasi kemampuan perangkat lunak untuk berfungsi secara konsisten dan andal selama periode waktu yang lama. Ini termasuk pengujian untuk

mendeteksi kebocoran memori, crash, atau malfungsi lainnya yang dapat terjadi seiring waktu.

- 5) *Capacity Testing* : Pengujian ini dilakukan untuk menentukan kapasitas maksimum perangkat lunak, yaitu jumlah maksimum pengguna atau transaksi yang dapat ditangani oleh sistem tanpa penurunan kinerja yang signifikan. Hal ini penting untuk merencanakan skala sistem dan memastikan perangkat lunak dapat memenuhi permintaan yang diharapkan.
- 6) *Regression Testing* : Pengujian regresi dilakukan untuk memastikan bahwa perubahan atau pembaruan pada perangkat lunak tidak mengakibatkan kerusakan pada fungsi yang sudah ada. Ini penting untuk menjaga stabilitas perangkat lunak selama siklus pengembangan dan perawatan.