

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Sebelumnya**

Penggunaan sistem informasi berbasis *website* kini sudah banyak dilakukan di berbagai bidang penelitian. Referensi dan rujukan yang dilakukan pada hasil penelitian sebelumnya yang telah dilakukan penelitian lain merupakan tujuan pada tinjauan Pustaka ini.

Salah satu penelitian terkait pada tahun 2020 dengan judul **“Pengembangan Sistem Informasi Akuntansi Penjualan Jasa *Laundry* Berbasis Web”** oleh Imam Soleh Marifati, Ubaidillah.

Dalam penelitian inilah penulis mengembangkan aplikasi berbasis web yang dapat digunakan oleh pengusaha *Laundry* dalam penjualan jasa *Laundry* kepada pelanggannya. Aplikasi berbasis web ini dapat diakses oleh pelanggan tanpa harus datang ke lokasi usaha *Laundry* sehingga memberi kemudahan. Dalam mengembangkan sistem informasi akuntansi, penulis menggunakan pendekatan Siklus Hidup Pengembangan Sistem (*System Development Life Cycle – SDLC*) yaitu siklus proses pengembangan dimulai dari tahap Dalam penelitian ini, peneliti mengembangkan sistem menggunakan pendekatan daur hidup pengembangan sistem (*System Development Life Cycle – SDLC*). Dari hidup pengembangan sistem yang dimulai dari tahap analisa sistem, desain konsep, desain fisik, implementasi dan konversi serta operasional dan pemeliharaan[2].

Penelitian berikutnya pada tahun 2020 dengan judul **“Pembuatan Aplikasi Manajemen UKM *Laundry* (Studi Kasus Water Lily *Laundry*)”** oleh Hery Herawan, Anindito Yoga Pratama, Esty Purnamasari, Lulu Chaerani Munggaran.

Pada penelitian ini penulis mengambil studi kasus di Water Lily *Laundry* yang merupakan salah satu UKM yang bergerak dalam bidang jasa *Laundry* yang berlokasi di Depok. Pemilik usaha UKM *Laundry* tersebut masih menggunakan sistem manual dalam menjalankan usahanya, sehingga manajemen dan pengontrolan usaha yang dilakukan pekerja kurang

optimal. Sistem yang dikembangkan dalam penelitian ini yaitu pembuatan aplikasi manajemen UKM *Laundry* pada *Water Lily Laundry*. Selain fitur master data seperti data paket *Laundry*, pengguna, pelanggan, dan juga master data stok bahan baku. Selain itu, pada aplikasi yang dikembangkan terdapat tambahan fitur yaitu dapat mengelola data karyawan, data jenis cucian, data pengeluaran non bahan baku, data bahan baku, data pembelian bahan baku, data uang, kemudian dapat menambahkan uang dan mengambil uang dari *box* uang, data transaksi, data pelunasan, data pengurangan bahan baku, data keluhan pelanggan, transaksi dan mengeluarkan data-data tersebut dalam bentuk laporan. Dalam pengaplikasian *website* tersebut terdapat juga laporan arus uang yang dapat diatur sesuai periode tertentu yang sangat berguna untuk mengecek pemasukan dan pengeluaran dalam periode tertentu dan mengetahui keuntungan UKM *Laundry*. Pembuatan aplikasi disesuaikan dengan desain sistem yang sudah dibuat sebelumnya. Aplikasi ini dibuat dengan bahasa pemrograman Java. Untuk kenyamanan pengguna, pembuatan desain aplikasi manajemen *Laundry* juga disesuaikan dengan kaidah-kaidah *Human Computer Interaction* (HCI). Ukuran *font* yang digunakan dibuat konsisten antara *form* yang satu dengan yang lainnya. Dalam perancangan sistem ini penulis menggunakan metode model sistem *waterfall* [7].

Penelitian berikutnya pada tahun 2016 dengan judul “**Pembangunan Sistem Informasi Pelayanan Jasa *Laundry* Berbasis Web Dengan Fitur *Mobile* Pada 21 *Laundry* Padang**” oleh Husnil Kamil, Audiah Duhani.

Pada penelitian ini penulis mengambil studi kasus di 21 *Laundry* Padang yang pengelolaan data transaksi belum menggunakan sistem terkomputerisasi. Proses pencatatan data pelanggan, data transaksi, dan pembuatan nota penyerahan yang masih dilakukan secara manual di mana setiap kegiatannya disimpan di dalam buku besar. Oleh karena sistem penyimpanan data masih berupa pengarsipan fisik sehingga sering terjadi penumpukan arsip yang dapat membuat keamanan data kurang terjamin. Proses pembuatan laporan transaksi, pencarian data yang lambat, dan penghitungan data dalam jumlah banyak sulit dilakukan dan menimbulkan

permasalahan dalam *administrasi*. Solusi yang ditawarkan untuk mengatasi permasalahan pada 21 *Laundry* Padang adalah dengan memanfaatkan teknologi informasi. Oleh karena itu, perlu dibangun sebuah sistem informasi berbasis web dengan fitur *mobile* pada 21 *Laundry* Padang yang nantinya diharapkan dapat menunjang aktivitas dan keberhasilan bisnis pada perusahaan. Penelitian berupa pengembangan perangkat lunak. Metodologi yang digunakan dalam pembangunan perangkat lunak ini mengikuti model pengembangan *waterfall*[8].

Penelitian berikutnya pada tahun 2020 dengan judul “**Perancangan Aplikasi Pencatatan Keuangan Studi Kasus Pada Amanah Laundry Bogor Menggunakan Agile Development Methods**” oleh Mohammad Fuad Farisi, Budi Praptono, Maria Dellarosawati Idawicaksati.

Pada penelitian ini penulis mengambil studi kasus di Amanah *Laundry* merupakan industri kecil menengah yang bergerak dibidang jasa cuci pakaian yang berada di Bojonggede Kabupaten Bogor dan masih dini dalam dunia pasar. Amanah *Laundry* yang masih membuat pencatatan keuangan secara manual atau dengan tulis tangan masih kurang efisien. Oleh sebab itu kelemahan pencatatan yang dilakukan dengan cara manual terkadang ke tidak akuratkan sering terjadi saat melakukan perhitungan antara pemasukan dan pengeluaran dalam pencatatan keuangan dan dokumen hasil pencatatan keuangan tidak rapi serta hilang. Dokumen keuangan seharusnya bisa memberikan informasi spesifik mengenai finansial usaha perusahaan. Oleh karena itu dibutuhkan sesuatu aplikasi yang bisa membantu dalam pembuatan pencatatan keuangan. Dengan dibuatnya aplikasi untuk pencatatan keuangan akan memberikan hasil yang efektif serta efisien bagi para pemilik usaha. Untuk membuat perancangan sistem aplikasi dibutuhkan sebuah metode yang sesuai dengan spesifikasi pada pengguna. Apabila pengguna membutuhkan pembuatan aplikasi dengan waktu yang cepat maka metode yang tepat yaitu metode *agile development*. Penyelesaian aplikasi dibuat dengan membuat proyek dengan keterangan kegiatan dan waktu pelaksanaan yang tersusun dalam *burndown chart grafik*. Metode *agile development methods* dengan

model *scrum* merupakan metode yang cocok untuk *user*, karena pengguna selalu dilibatkan dalam proses pengembangan sistem program. Agar kegiatan pengembangan sistem yang berkualitas dalam waktu yang singkat, metode *scrum* ini dapat digunakan oleh pemrogram untuk penyelesaian kebutuhan pengguna. Sehingga kebutuhan pengguna terpenuhi secara nyata[9].

Penelitian berikutnya pada tahun 2016 dengan judul “**Implementasi Metodologi SCRUM dalam Pengembangan Sistem Pembayaran Elektronik Pada Usaha Mikro Kecil Menengah**” oleh Taufiq Rizaldi, Dwi Putro Sarwo S., Hendra Yufit R.

Pada penelitian ini penulis permasalahan yang dihadapi oleh mitra UMKM adalah Kurangnya pemanfaatan teknologi informasi untuk pengelolaan sistem pembayaran, belum adanya sistem informasi pembayaran berbasis web, kurangnya pemahaman dalam pemanfaatan Teknologi Informasi untuk kepentingan tata kelola sistem pembayaran yang modern, pengelolaan sistem pembayaran yang belum terkomputerisasi menyebabkan rendahnya dokumentasi dan kualitas pelaporan data seperti laporan bulanan, siapa yang sedang menggunakan sistem, hingga rekapitulasi laporan pengelolaan sistem pembayaran. Dengan memanfaatkan metodologi dari SCRUM yang mempunyai sifat yang fleksibel untuk mengembangkan sebuah aplikasi pembayaran dan didukung dengan pendampingan, permasalahan pada pelaporan rincian pembayaran yang dialami oleh UMK akan berjalan lebih optimal dan efektif[10].

Penelitian berikutnya pada tahun 2019 dengan judul “**Pengembangan Sistem Informasi KPR Syariah Dengan Metode Scrum**” oleh Sri Hardani.

Pada penelitian ini penulis ingin mengembangkan layanan KPR syariah dan untuk meningkatkan tingginya minat masyarakat terhadap KPR syariah, maka diperlukan sarana pengelolaan yang tepat. Untuk KPR yang disediakan oleh perbankan syariah, tentunya sudah dikelola dengan sistem yang memadai. Sementara untuk KPR syariah yang disediakan oleh developer perumahan syariah, sistemnya masih sangat sederhana dan perlu

ditingkatkan. Penerapan aplikasi berbasis web akan sangat membantu baik untuk pengunjung (calon pembeli), pembeli, maupun developer (penjual). Calon pembeli dapat mengakses informasi tanpa harus menghubungi developer. Pembeli dapat melihat informasi *terupdate* terkait rumah yang dibeli, baik mengenai pembayaran maupun progres pembangunan rumahnya. Developer dapat mengelola data KPR dan memperoleh berbagai informasi terkait KPR yang dikelolanya, dengan mudah. Dalam proses pengembangan aplikasi, perlu diterapkan metode pengembangan aplikasi, agar proses pengembangan aplikasi dapat berjalan secara terencana dan efisien. Ada banyak metodologi yang dapat digunakan. Salah satu yang saat ini banyak digunakan adalah metode *Scrum*. *Scrum* merupakan metodologi baru dalam pengembangan perangkat lunak. Metode ini banyak digunakan karena mampu beradaptasi dengan perubahan yang terjadi selama proses pengembangan aplikasi[11].

Penelitian berikutnya pada tahun 2017 dengan judul “**Implementasi Metode *Scrum* Dalam Rancang Bangun Sistem Informasi Penjualan ( Study Kasus : Penjualan *Sparepart* Kendaraan )**” oleh Novri Hadinata, Muhammad Nasir.

Pada penelitian ini penulis ingin melakukan salah satu faktor pendorong perencanaan untuk pemanfaatan sistem informasi penjualan *sparepart* mobil dengan meningkatnya kebutuhan dalam fungsi bisnis yang dijalankan. Dalam perencanaan penerapan sistem informasi penjualan diharapkan perencanaan sistem yang dibuat sesuai yang diharapkan agar informasi dalam pengolahan data penjualan dapat diimplementasikan untuk ke tahap selanjutnya. Proses penjualan di mana sistem ini di rancang dengan menerapkan sebuah metode yaitu metode *Scrum* di mana metode ini banyak mengandung sprint dan beban kerja yang mana masing-masing sprint didorong oleh *Prioritized product backlog*. *Prioritized Product Backlog* terdiri dari fitur terbaru, perbaikan bug, dan hal lain yang memberikan kontribusi pada produk akhir. Metode *Scrum* merupakan suatu pendekatan

interaktif pada pengembangan perangkat lunak yang mengusung prinsip *agile* (ketangkasan)[12].

Penelitian berikutnya pada tahun 2014 dengan judul “**Implementasi Metodologi *Scrum* dalam Pembangunan Situs Harga Komoditas**” oleh Made Krisnanda. Penelitian ini penulis bertujuan untuk memberi pengetahuan mengenai kerangka kerja *Scrum* dan implementasinya pada pengembangan perangkat lunak khususnya situs yang diakses melalui internet. Munculnya Metodologi *Scrum* membuktikan bahwa dari hari ke hari proyek pengembangan sistem makin memerlukan fleksibilitas yang tinggi untuk dapat memenuhi kepuasan pelanggan. Hal ini terutama sering terjadi pada pengembangan aplikasi berbasis web, yang menjadi salah satu cara mencapai keunggulan berkompetisi[4]

Tabel 2.1 Penelitian Sebelumnya

No.	Judul Penelitian	Masalah	Metode	Hasil
1.	Perancangan Aplikasi Pencatatan Keuangan Studi Kasus Pada Amanah Laundry Bogor Menggunakan <i>Agile Development Methods</i> [9].	Masih kesulitan dalam pencatatan keuangan yang menggunakan cara manual membuat data tidak akurat saat melakukan perhitungan antara pemasukan dan pengeluaran dalam pembuatan laporan yang tidak rapi serta hilang.	Metode <i>Agile</i> model <i>Scrum</i>	Bahwa untuk membuat aplikasi berbasis <i>website</i> dibutuhkan sebuah metode yang mampu menyelesaikan program yang tersusun dengan rapi dan cepat terselesaikan yaitu dengan metode <i>agile development</i> dengan model <i>scrum</i> , selain itu peranan pengguna juga membantu dalam pengembangan sistem.
2.	Pengembangan Sistem Informasi KPR Syariah Dengan Metode <i>Scrum</i> [11].	Pada KPR syariah yang disediakan oleh developer perumahan syariah, sistemnya masih sangat sederhana dan perlu ditingkatkan.	Metode <i>Scrum</i> .	Selama pengembangan sistem, terdapat beberapa kali perubahan kebutuhan pengguna. Dengan menggunakan metode <i>scrum</i> , perubahan ini disampaikan pada <i>sprint review</i> yang diadakan setiap akhir <i>sprint</i> dan dihadiri oleh <i>stake holder, product owner, scrum master</i> , dan tim <i>scrum</i> . Dengan adanya <i>sprint review</i> , komunikasi antara pengembang dan <i>stake holder</i> menjadi lebih mudah.
3.	Implementasi Metodologi SCRUM dalam Pengembangan Sistem Pembayaran Elektronik Pada Usaha Mikro Kecil Menengah[10].	Pengelolaan sistem pembayaran yang belum terkomputerisasi menyebabkan rendahnya dokumentasi dan kualitas pelaporan data seperti laporan bulanan, siapa yang sedang menggunakan	Metode <i>Scrum</i> .	Penerapan metodologi SCRUM pada pengembangan pada permasalahan tersebut sesuai diterapkan sebab dapat mengantisipasi perubahan yang terjadi selama proses pengembangan

No.	Judul Penelitian	Masalah	Metode	Hasil
		sistem, hingga rekapitulasi laporan pengelolaan sistem pembayaran		aplikasi.
4.	Implementasi Metode <i>Scrum</i> Dalam Rancang Bangun Sistem Informasi Penjualan ( Study Kasus : Penjualan <i>Sperpart</i> Kendaraan ) [12].	Proses pengolahan data penjualan dan persediaan barang dicatat dalam sebuah berkas dalam bentuk lembaran kertas, dimana data tersebut akan disimpan pada biling arsip.	Metode <i>Scrum</i> .	Metodologi <i>Scrum</i> membuktikan bahwa dari hari ke hari proyek pengembangan sistem makin memerlukan fleksibilitas yang tinggi untuk dapat memenuhi kepuasan pelanggan.
5.	Pembangunan Sistem Informasi Pelayanan Jasa <i>Laundry</i> Berbasis Web Dengan Fitur <i>Mobile</i> Pada 21 <i>Laundry</i> Padang [8].	Pada <i>Laundry</i> tersebut masih menggunakan cara manual, sehingga penyimpanan data masih berupa pengarsipan fisik sering terjadi penumpukan arsip yang dapat membuat keamanan data kurang terjamin.	Metode <i>waterfall</i> .	Setiap melakukan transaksi sudah akurat, menampilkan informasi yang diperlukan, menghemat waktu mencari data, adanya laporan transaksi bulanan, melihat status cucian, dan menyimpan data selama bertahun tahun.
6.	Pengembangan Sistem Informasi Akuntansi Penjualan Jasa <i>Laundry</i> Berbasis Web [2].	Peningkatan layanan <i>Laundry</i> untuk pengusaha jasa <i>Laundry</i> dalam upaya peningkatan pelanggan yang ada di kota-kota besar	Metode <i>System Development Life Cycle</i>	Dengan adanya sistem informasi ini dapat memudahkan pelanggan dalam pemesanan, dan menjangkau pelanggan lebih luas dan praktis.
7.	Pembuatan Aplikasi Manajemen UKM <i>Laundry</i> (Studi Kasus Water Lily <i>Laundry</i> ) [7].	Sistem pada <i>Laundry</i> tersebut masih menggunakan manual dalam menjalankan usahanya, sehingga manajemen dan pengontrolan usaha yang dilakukan pekerja masih kurang optimal.	Metode <i>Waterfall</i> .	Dengan adanya sistem pada <i>Laundry</i> tersebut dapat memudahkan pemilik untuk memantau segala kegiatan manajemen yang dilakukan, sehingga pemilik <i>Laundry</i> tersebut tidak perlu datang setiap saat untuk melakukan pengecekan data.



Maka dari penelitian sebelumnya dapat disimpulkan penggunaan metode *scrum*, dalam rancang bangun sistem informasi dapat mengefisiensi waktu pada saat pengembangan dilakukan. Dikarenakan setiap pengembangan dilakukan adanya *sprint*, dimana *sprint* tersebut bertugas menjalankan *product backlog* menjadi fitur pada sistem, *sprint review* akan menjelaskan hasil yang telah dibuat sesuai dengan *product backlog*, dan *sprint retrospective* akan dilakukannya perubahan jika ada kebutuhan terbaru maupun fitur yang tidak sesuai dari *product owner*. Pemilihan metode *scrum* menjadi fleksibilitas yang membuat proses-proses pengembangan sistem memiliki tingkat kompleksitas yang cukup tinggi, kondisi tersebut sulit diantisipasi oleh metode pengembangan sistem konvensional seperti *waterfall* atau Metode *System Development Life Cycle*.

## 2.2 Dasar Teori

### 2.2.1 Laundry

Usaha mikro, kecil, dan menengah (UMKM) merupakan salah satu bagian penting dari perekonomian suatu negara ataupun daerah. Berbagai upaya pengembangan UMKM telah dilakukan, salah satunya dengan membangkitkan dan memperbanyak orang atau pengusaha baru di bidang UMKM, sehingga masyarakat bisa diberi keterampilan dengan harapan keterampilan tersebut menjadi sebuah ide dan terciptanya usaha kreatif yang memberi manfaat bagi perekonomian keluarga dan daerah tersebut[13]. *Laundry* ialah sebuah departemen *housekeeping* yang bertugas dan bertanggung jawab untuk memproses semua aktivitas pencucian baik untuk operasional hotel dan tamu hotel. *Laundry* adalah kata benda yang mengacu pada tindakan mencuci pakaian, tempat di mana mencuci dilakukan. *Laundry* juga diartikan sebagai kegiatan mencuci pakaian atau bahan tekstil lainnya dan juga sebagai sebuah tempat untuk mencuci pakaian atau bahan [1].

### 2.2.2 Website

*Website* adalah kumpulan dari halaman-halaman situs, yang terangkum dalam sebuah domain atau *subdomain*, yang tempatnya berada di dalam *World Wide Web* (WWW) di dalam internet. Sebuah halaman web biasanya berupa dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari server *website* untuk ditampilkan kepada para pemakai melalui *web browser*[14].

Web mengizinkan pemberian *highlight* (penyorotan atau penggaris bawah) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, frase, *movie clip*, atau file suara. *Web* dapat menghubungkan dari sembarang tempat dalam sebuah dokumen atau gambar ke sembarang tempat di dokumen lain. Dengan sebuah browser yang memiliki *Graphical User Interface* (GUI), *link* dapat di hubungkan ke tujuannya dengan menunjuk *link* tersebut dengan[15].

### 2.2.3 PHP

PHP adalah bahasa pemrograman skrip sederhana yang digunakan untuk pemrosesan HTML *Form* di dalam halaman web. Strukturnya sangat sederhana sehingga PHP dapat dengan mudah dipelajari *programmer* pemula, bahkan oleh orang-orang tanpa latar belakang Teknologi Informasi [16]. PHP atau yang memiliki kepanjangan PHP *Hypertext Preprocessor*, merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu *website* dinamis. PHP menyatu dengan kode HTML, PHP difungsikan sebagai proses, sehingga dengan adanya PHP tersebut, sebuah web akan sangat mudah di *maintenance*[17]. PHP termasuk bahasa program yang hanya bisa berjalan di sisi server, atau sering disebut *Side Server*

*Language*. Tanpa adanya server web yang terus berjalan dia tidak akan bisa dijalankan[18].

#### 2.2.4 *CodeIgniter*

*Codeigniter* adalah sebuah *framework* untuk web yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi web yang kompleks. *Codeigniter* dapat mempercepat proses pembuatan web , karena semua *class* dan modul yang dibutuhkan sudah ada dan *programmer* hanya tinggal menggunakannya kembali pada aplikasi web yang akan dibuat[19].

Aplikasi yang dikembangkan dengan mengimplementasikan *Framework CodeIgniter* (CI) yang mendukung teknik pemrograman *Model-View-Controller* (MVC) untuk mengurangi waktu pengembangan aplikasi. Menggunakan *CodeIgniter*, pengembang tidak perlu lagi mengembangkan aplikasi dari awal dan dapat langsung fokus pada kode yang membangun fungsi aplikasi yang dikembangkan. Model MVC memungkinkan pengembang untuk mengakses sumber daya *database* beberapa server. Sesuai dengan aturan sistem terpadu, dan untuk mengurangi data yang berlebihan harus menggunakan layanan web yang dapat mengakses *database* secara bersamaan[20].

#### 2.2.5 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP Server, *MySQL* database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan *Perl*. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, PHP dan *Perl*. [15].

XAMPP berperan sebagai server web pada komputer lokal. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi website tanpa harus *online* atau terakses dengan internet[21].

### 2.2.6 *PhpMyAdmin*

*PhpMyAdmin* adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen *MySQL*. Dengan menggunakan *PhpMyAdmin*, dapat membuat *database*, membuat tabel, *insert*, menghapus dan *update* data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual[22]. Karena berbasis web, maka *PhpMyAdmin* dapat dijalankan dibanyak *Operating System*, selama dapat menjalankan web server dan *MySQL*[18].


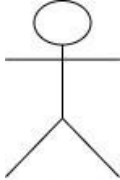



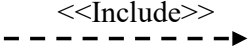
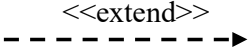
### 2.2.7 UML

UML adalah salah satu *tool* atau model untuk merancang pengembangan *software* yang berbasis *object-oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software*[23]. Terdapat gambaran untuk membuat UML (*Unified Modeling Language*) dengan beberapa tipe yaitu *use case*, *class diagram*, *sequence diagram* dan *activity diagram*[24].

#### a. *Use Case*

*Use Case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem. Digunakan untuk memodelkan suatu sistem atau perangkat lunak dilihat dari sisi pengguna yang ada pada sistem[25].




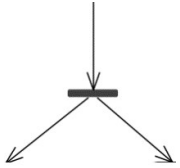
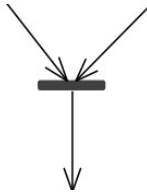

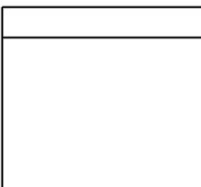
Tabel 2.2 Simbol-simbol pada Use case Diagram

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	Aktor adalah <i>Abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem, Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap use case.
	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan
Gambar	Keterangan
	bukannya mengindikasikan data.
	Generalisasi disebut juga inheritance(pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
	<i>Include</i> , merupakan di dalam use case lain ( <i>required</i> ) atau pemanggilan use case oleh use case lain.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

b. *Activity Diagram*

*Activity diagram* digunakan untuk memodelkan aspek dinamis dari sistem. *Activity diagram* berfungsi memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas pada suatu proses. *Activity diagram* dibuat untuk menggambarkan aktivitas aktor atau *user*[26].

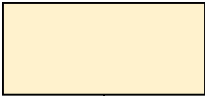


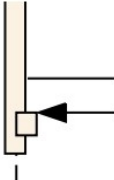
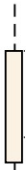

Tabel 2.3 Simbol-Simbol *Activity Diagram*

Gambar	Keterangan
	<i>StartPoint</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>EndPoint</i> , diletakkan pada bagian bawah aktivitas yang menandakan akhir dari aktivitas.
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, true atau false.
	<i>Swimlane</i> , pembagian Activity diagram untuk menunjukkan siapa melakukan apa.

### c. *Sequence Diagram*

*Sequence Diagram* menerangkan serangkaian hubungan yang terjadi antara aktor dan objek-objek serta kelas objek pada saat *user* akan melakukan kegiatan. Dalam diagram ini aktor yang terlibat adalah *user* yang menggunakan komputer atau telepon seluler. Sedangkan objek yang berinteraksi dengan aktor[24].


Tabel 2.4 Simbol-Simbol *Sequence Diagram*


Gambar	Keterangan
	objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Return Message</i> , simbol menerima pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

d. *Class Diagram*

*Class diagram* adalah diagram untuk menggambarkan *class-class* yang ada pada aplikasi[25]. *Class Diagram* berfungsi sebagai pendeskripsian jenis objek yang ada dalam sistem dan berbagai macam hubungan statis yang terjadi[26].

Tabel 4.5 Simbol-Simbol *Class Diagram*

Gambar	Keterangan
	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.

Gambar	Keterangan
	Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara elemen-elemen <i>hardware</i> .

## 2.2.8 Metode Pengujian Sistem

### 2.2.8.1 *Whitebox Testing*

Pengujian White Box, adalah suatu metode pengujian aplikasi yang menggunakan penjelasan struktur kontrol sebagai bagian dari component-level design untuk membuat test cases. White Box[27]. Pengujian white box ini dilakukan dengan melihat pure kode tanpa melihat tampilan interface dari halaman sistem[28]. Adapun beberapa teknik dalam pengujian whitebox testing.

Cyclomatic Complexity adalah besaran perangkat lunak yang menyediakan acuan kuantitatif kompleksitas suatu logika dalam program. Rumus menghitung Cyclomatic Complexity adalah sebagai berikut [27]:

$$V(G) = E - N + 2$$

Keterangan :

$$V(G) = \text{Cyclomatic complexity}$$

E = Jumlah *edges* pada *flowgraph*

N = Jumlah *nodes* pada *flowgraph*

### 2.2.8.2 *BlackBox Testing*

Metode Black Box adalah metode yang ketika melakukan pengujian tidak perlu melihat dari kode pemrograman, tidak seperti White Box Testing yang mengutamakan kode pemrograman yang digunakan beserta dengan kelas dan fungsinya[29]. Pada proses Black Box Testing dengan cara



mencoba program yang telah dibuat, dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan[30].

### 2.2.8.3 *Usability Metrics*

*Usability Metrics* adalah "sistem atau standar pengukuran" yang direpresentasikan dalam satuan yang dapat digunakan untuk menggambarkan lebih dari satu atribut. *Usability Metrics* sangat berguna untuk mengukur kegunaan selama evaluasi kegunaan perangkat lunak, situs web, dan aplikasi[31].

Sebagian besar *Usability Metrics* dihitung berdasarkan data yang dikumpulkan selama pengujian kegunaan. Pengguna diminta untuk menyelesaikan tugas sementara peneliti mengamati perilaku pengguna dan membuat catatan. Tugas dapat berupa sesuatu seperti "Bagaimana cara transaksi", atau "Daftar di situs web"[32]. Ada pun untuk penghitungan *Usability Metrics* antara lain:

#### a. *Completion Rate Effectiveness*

*Effectiveness* dapat dihitung dengan menghitung tingkat penyelesaian dengan angka "1" dan jika pengguna tidak berhasil maka nilai yang ditetapkan "0". Selain itu, dapat dikumpulkan selama setiap tahap perkembangan. Efektivitas dengan demikian dapat direpresentasikan sebagai persentase dengan menggunakan rumus sederhana ini[31]:

$$Effectiveness = \frac{\text{Tugas yang berhasil diselesaikan}}{\text{Jumlah total tugas yang dilakukan}} \times 100\%$$

Keterangan:

*Effectiveness* = efektivitas

Tugas yang berhasil diselesaikan = Seluruh tugas yang dikerjakan oleh pengguna.

Jumlah total tugas yang dilakukan = Seluruh tugas yang diberikan.

b. *Time-Based Efficiency*

*Time-Based* perhitungan seberapa efisien tugas yang dilakukan oleh pengguna dalam waktu yang dikerjakan. Dengan rumus dibawah ini digunakan untuk menghitung efisiensi waktu.

$$\bar{P}_t = \frac{\sum_{j=1}^R \sum_{i=1}^R \frac{n_{ij}}{t_{ij}}}{NR}$$

Keterangan:

$\bar{P}_t$  : *time-based efficiency*

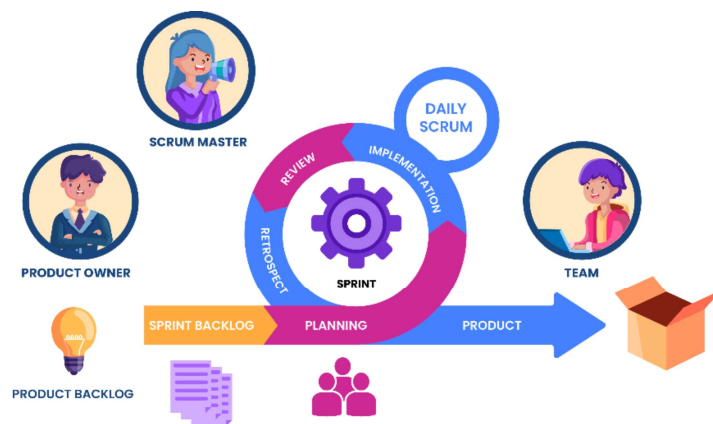
R : Jumlah responden

N : Jumlah total tugas

$n_{ij}$  : Hasil tugas i oleh pengguna j; jika pengguna berhasil menyelesaikan tugas, maka  $n_{ij} = 1$ , jika tidak, maka  $n_{ij} = 0$

$t_{ij}$ : Waktu yang dihabiskan oleh pengguna j untuk menyelesaikan tugas i. Jika tugas tidak berhasil diselesaikan, maka waktu diukur hingga saat pengguna keluar dari tugas.

### 2.2.9 Metode Scrum



Gambar 2.1 Tahapan Scrum

Metode Kerangka kerja *Scrum*, adalah jenis baru dari metodologi *Agile* yang diharapkan dapat meningkatkan kecepatan dan fleksibilitas dalam pengembangan perangkat lunak[10]. Proses pengembangan sistem memiliki tingkat kompleksitas yang tinggi. Sering kali pengguna atau lingkungan organisasi menuntut perubahan secara tiba-tiba, yang tidak diantisipasi oleh tim pengembang. Kondisi ini sulit diantisipasi oleh metode pengembangan sistem konvensional seperti *Waterfall* atau *Spiral*[4].

Berikut ini penjelasan dari penerapan pada metode *Scrum*:

a. *Product Backlog*

Langkah pertama dari *Scrum* adalah penentuan dari fitur berdasarkan prioritasnya oleh manajer proyek.

b. *Sprint Planning Meeting*

Setelah menentukan *Product Backlog* diadakan briefing oleh *scrum team* di setiap awal *Sprint* untuk mengevaluasi *Product Backlog*, mendiskusikan tujuan dan misi dari tiap fitur sesuai dengan keinginan *Product Owner*. Pada bagian ini, setiap anggota *scrum team* menentukan berapa jam yang dibutuhkan tiap anggota untuk melaksanakan pekerjaan pada tiap fiturnya.

c. *Daily Scrum*

Sebagai praktik untuk memantau kinerja, setiap hari perlu diadakan *meeting* untuk pelaporan apa saja yang sudah tiap anggota tim kerjakan. Pada tahap ini tidak semua anggota tim harus hadir, namun lebih kepada anggota yang terlibat khusus pada fitur yang sedang dikembangkan. Pada setiap akhir *meeting*, waktu penyelesaian diperbaharui untuk mengetahui sisa pengerjaan yang harus dikejar selama *Sprint* berjalan[3].

d. *Sprint Review*

Setiap *Sprint* selalu berakhir dengan mendemonstrasikan dan mempresentasikan fitur-fitur yang telah dikerjakan. Hal tersebut

dilakukan untuk memastikan bahwa fitur-fitur tersebut dapat bekerja dengan baik.

e. *Sprint Retrospective*

Harapan yang ingin dicapai pada *Sprint Retrospective* adalah adanya perbaikan tindakan sehingga Sprint berikutnya dapat dikerjakan dengan lebih baik lagi. Perbaikan-perbaikan tersebut harus di-implementasi-kan pada Sprint berikutnya[33].