

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian yang akan dilakukan oleh peneliti yang berkaitan dengan judul membuat rancangan *website dashboard management* 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 Melky Agustina Sidaruk pada tahun 2022, dengan judul “Rancang Bangun *Website Dashboard Sales* Menggunakan metode *Scrum* (Studi Kasus PT.Tourezia Yogyakarta)“. Permasalahan dalam penelitian ini bahwa PT. Tourezia Yogyakarta masih manual dalam menginput data sehingga memakan waktu cukup lama serta visualisasi yang terbatas membuat sering terjadi kesalahan dalam perhitungan [6].

Penelitian yang ditulis oleh Dipta Arsyani Febby Suare dan Dedy Rahman Prehanto pada tahun 2022, dengan judul “Rancang Bangun Sistem Informasi Pengelolaan Administrasi Persuratan berbasis *Website* menggunakan Metode *Scrum*”. Kegiatan surat menyurat suatu organisasi sangat penting pada Ormawa FT UNESA. Organisasi harus mampu beradaptasi terhadap teknologi yang berkembang dalam pengarsipan surat menyurat [7].

Penelitian yang ditulis oleh Tri Setya Nugraha, Kusnadi dan Rifqi Hardian pada tahun 2021 dengan judul “ Rancang Bangun Sistem Informasi *Company Profile* dengan menggunakan metode *Scrum* Pada PT.Hasna Satya Negara Berbasis *Web* “. Permasalahan dalam penelitian pada PT. Hasna Satya mengalami permasalahan dalam penyebaran informasi yang masih dilakukan secara manual dengan menggunakan brosur pada tempat umum. PT. Hasna Satya sehingga mengalami kendala seperti keterbatasan karyawan dalam proses penyebaran dan biaya yang cukup banyak untuk kebutuhan cetak brosurnya [8].

Penelitian yang dilakukan oleh Jovan Aully Prabu pada tahun 2023, dengan judul “ Rancang Bangun Aplikasi Laporan Keuangan Kedai Kontekstual Kopi Berbasis *Website* Menggunakan Metode *Scrum* “. Permasalahan dalam penelitian yaitu sebuah kedai kopi yang masih memakai cara manual atau tulis tangan dalam mencatat laporan keuangan, sehingga membutuhkan waktu yang cukup lama, pemilik kedai juga mengalami kesulitan untuk mengetahui aktivitas penjualan pada kedai kopi dengan cara menghitung laporan manual pada buku catatan [9] .

Penelitian yang ditulis oleh Muhammmad Dwiki Reza, Depandi Enda, dan Ryci Rahmatil Fiska pada tahun 2022 dengan judul “Penerapan Metode *Scrum* pada Rancang Bangun Aplikasi Manajemen Koperasi (Studi Kasus : Koperasi Produksi Generasi Mandiri)”. Pada penelitian ini memiliki permasalahan pada Koperasi Produksi Generasi Mandiri yaitu pengelolaan data-data koperasi dan transaksi masih manual sehingga kesulitan dalam menemukan data-datanya yang menyebabkan waktu yang lama dalam pencarian. Penelitian ini menggunakan Metode *Scrum* dan menggunakan bahasa pemrograman *PHP* berbasis *Laravel Framework*. Aplikasi selesai dengan tepat waktu dengan etimasi waktu yang sesuai [10].

Penelitian yang ditulis oleh Nathanael Frederico Wibawanto pada tahun 2023 dengan judul penelitian “ Sistem Permohonan Ijin Berbasis *Website* Menggunakan *Framework Laravel* Dengan Metodologi *Scrum* “. Pada penelitian tersebut menggunakan metode *Scrum* dengan permasalahan pada PT. Televisi Kampus Universitas Dian Nuswantoro yaitu pada sistem pengajuan permohonan ijin dan cuti masih dilakukan dengan model konvensional yaitu menggunakan kertas. Sistem tersebut memiliki kelemahan adalah masih dilakukan prosesnya hanya dalam cakupan kantor serta rekap pendataan masih dilakukan secara manual [11].

Penelitian yang ditulis oleh Monica Efniasari pada tahun 2022 dengan judul penelitian “ Pengembangan Sistem Informasi Pelayanan Kesehatan Berbasis *Website* Menggunakan metode *Scrum* (Studi Kasus : Puskesmas Kisam Ilir)”. Metode pengembangan menggunakan *Scrum*, dengan permasalahan pada puskesmas masih menggunakan proses manual dalam pelayanan Kesehatan yaitu

dengan mengelola data pasien dengan menuliskan pada buku laporan pasien. Oleh sebab itu kinerja dari sistem informasi masih terkendala dalam proses pencarian data pasien, data pasien pasien bisa hilang dan rusak karena tempat penyimpanan berkas yang masih manual [12].

Tabel 2. 1 Penelitian Sebelumnya

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
1.	Rancang Bangun <i>Website Dashboard Sales</i> Menggunakan Metode <i>Scrum</i> (Studi Kasus PT.Tourezia Yogyakarta) [6].	2022	Proses input data secara manual sehingga menyebabkan visualisasi yang terbatas membuat terjadi kesalahan dalam perhitungan	Metode <i>Scrum</i>	Berdasarkan hasil penelitian <i>Website Dashboard sales</i> berhasil dikembangkan dan diimplementasikan. Dengan menggunakan 3 <i>product backlog</i> dengan waktu pengerjaan 15 hari serta total waktu keseluruhan 85 Jam. Dengan hasil uji <i>website</i> tersebut berjalan dengan baik dan diterima oleh mitra.	Perbedaan pada penelitian adalah studi kasus mengenai <i>Website Dashboard Sales</i> pada PT.Tourezia Yogyakarta sedangkan penelitian oleh peneliti dengan studi kasus membuat rancangan <i>dashboard website management</i> OBRE-JEK Kabupaten Brebes, Jawa Tengah. Pada penelitian yang dilakukan peneliti akan menggunakan pengujian <i>laravel dusk testing</i> .

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
2.	Rancang Bangun Sistem Informasi Pengelolaan Administrasi Persuratan berbasis <i>Website</i> menggunakan Metode <i>Scrum</i> [7].	2022	Kegiatan surat menyurat suatu organisasi sangat penting pada Ormawa FT UNESA. Organisasi harus mampu beradaptasi terhadap teknologi yang berkembang dalam pengarsipan surat menyurat.	Metode <i>Scrum</i>	Berdasarkan penelitian Sistem Informasi Pengelolaan Administrasi Persuratan berbasis <i>website</i> berhasil untuk diimplementasikan dengan baik. Dengan hasil uji kelayakan <i>website</i> tersebut sudah berjalan dan dapat digunakan.	Perbedaan penelitian dari segi mengenai Rancang Bangun Sistem Informasi Pengelolaan Administrasi Persuratan berbasis <i>website</i> . Sedangkan penelitian yang dilakukan peneliti studi kasus membuat rancangan <i>dashboard website management</i> OBRE-JEK Kabupaten Brebes, Jawa Tengah. Pada penelitian yang dilakukan peneliti akan menggunakan pengujian <i>laravel dusk testing</i> .

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
3.	Rancang Bangun Sistem Informasi <i>Company Profile</i> dengan menggunakan Metode <i>Scrum</i> Pada PT.Hasna Satya Negara Berbasis <i>Web</i> [8].	2021	PT. Hasna Satya mengalami permasalahan dalam penyebaran informasi yang masih dilakukan secara manual.	Metode <i>Scrum</i>	Berdasarkan penelitian mampu mengimplementasikan website sistem informasi pada PT. Hasna Satya dengan kebutuhan fungsional dan non-fungsional untuk penyampaian informasi.	Perbedaan penelitian yang dilakukan dengan <i>study case</i> Rancang Bangun Sistem Informasi <i>Company Profile</i> dengan menggunakan Metode <i>Scrum</i> Pada PT.Hasna Satya Negara Berbasis <i>Web</i> . Sedangkan penelitian yang akan dilakukan peneliti yaitu membuat racangan <i>dashboard website management</i> OBRE-JEK Kabupaten Brebes, Jawa Tengah. Pada penelitian yang dilakukan peneliti

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
						akan menggunakan pengujian <i>laravel dusk testing</i> .
4.	Rancang Bangun Aplikasi Laporan Keuangan Kedai Kontekstual Kopi Berbasis <i>Website</i> Menggunakan Metode <i>Scrum</i> [9].	2023	kedai kopi yang masih memakai cara manual atau tulis tangan dalam mencatat laporan keuangan, serta mengalami kesulitan untuk mengetahui aktivitas penjualan pada kedai kopi dengan cara menghitung laporan manual pada buku catatannya.	Metode <i>Scrum</i>	Berdasarkan penelitian tersebut menghasilkan <i>website</i> laporan keuangan dengan uji coba menggunakan <i>black box testing</i> yang menghasilkan rangkaian proses yang dapat diuji dengan berjalan dengan baik dan normal.	Perbedaan pada penelitian adalah pada <i>study case</i> mengenai rancang bangun aplikasi keuangan kedai kontekstual kopi sedangkan penelitian oleh peneliti dengan studi kasus membuat rancangan <i>dashboard website management</i> OBRE-JEK Kabupaten Brebes, Jawa Tengah. Pada penelitian yang

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
						dilakukan peneliti akan menggunakan pengujian <i>laravel dusk testing</i> .
5.	Penerapan Metode <i>Scrum</i> pada Rancang Bangun Aplikasi <i>Management</i> Koperasi (Studi Kasus : Koperasi Produksi Generasi Mandiri) [10].	2022	Koperasi Produksi Generasi Mandiri memiliki permasalahan yaitu pengelolaan data-data koperasi dan transaksi masih manual sehingga kesulitan dalam menemukan data-datanya yang menyebabkan waktu yang lama dalam pencarian.	Metode <i>Scrum</i>	Berdasarkan hasil penelitian tersebut menghasilkan Aplikasi <i>Management</i> Koperasi dengan menggunakan bahasa pemrograman <i>PHP</i> berbasis <i>Laravel Framework</i> dengan pengujian sistem menggunakan <i>Black Box Testing</i> .	Perbedaan penelitian adalah pada studi kasus yaitu mengenai aplikasi koperasi sedangkan penelitian oleh peneliti dengan studi kasus membuat rancangan <i>dashboard website management OBRE-JEK Kabupaten Brebes, Jawa Tengah</i> dengan tambahan <i>framework tailwind</i> dan pengujian

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
						tambahan <i>laravel dusk testing</i>
6.	Sistem Permohonan Ijin Berbasis <i>Website</i> Menggunakan <i>Framework Laravel</i> Dengan Metodologi <i>Scrum</i> [11].	2023	Sistem pengajuan permohonan ijin dan cuti masih dilakukan dengan model konvensional yaitu menggunakan kertas. Proses rekap data manual serta pengajuan cuti hanya dilakukan pada cakupan kantor.	Metode <i>Scrum</i>	Berdasarkan Penelitian pengembangan sistem pengajuan permohonan ijin dan cuti berbasis <i>website</i> dapat diimplementasikan dengan baik sehingga mendapatkan presentasi fungsionalitas komponen <i>valid</i> bernilai 100% .	Perbedaan pada penelitian pada studi kasus yaitu sistem pengajuan permohonan ijin berbasis <i>website</i> sedangkan studi kasus penelitian oleh peneliti adalah rancangan <i>dashboard website management</i> OBRE-JEK Kabupaten Brebes, Jawa Tengah. Pada penelitian yang dilakukan peneliti akan menggunakan

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
						pengujian <i>laravel dusk testing</i> .
7.	Pengembangan Sistem Informasi Pelayanan Kesehatan Berbasis <i>Website</i> Menggunakan metode <i>Scrum</i> (Studi Kasus : Puskesmas Kisam Ilir)” [12].	2022	Proses mengelola data pasien masih secara manual dengan menuliskan pada buku laporan pasien. Sehingga kinerja masih terkendala dalam proses pencarian data pasien, data pasien pasien bisa hilang dan rusak karena tempat penyimpanan berkas yang masih manual.	Metode <i>Scrum</i>	Berdasarkan penelitian pengembangan sistem informasi pelayanan Kesehatan berbasis <i>website</i> di puskesmas Kisam Ilir dapat di implementasikan dengan pengujian menggunakan ISO 25010 berjalan dengan baik dengan kelayakan Usabilitynya mencapai nilai 83%.	Perbedaan penelitian sebelumnya adalah pada studi kasus mengenai pengembangan sistem informasi puskesmas dengan berbasis <i>website</i> dan menggunakan uji coba dengan ISO 25010 sedangkan penelitian yang dilakukan oleh peneliti dengan studi kasus racangan <i>dashboard website management</i> OBRE-JEK Kabupaten Brebes, Jawa

No	Judul	Tahun	Masalah	Metode	Hasil	Perbedaan
						Tengah dan menggunakan testing dengan <i>blackbox testing</i> dan <i>laravel dusk testing</i>

Pada tabel 2.1 mengenai penelitian sebelumnya membandingkan permasalahan yang diteliti oleh peneliti dengan kata kunci sesuai dengan judul penelitian seperti *dashboard*, *laravel*, rancang bangun, pengujian sistem dan *management*.

2.2 Landasan Teori

2.2.1 Rancang Bangun

Rancang bangun merupakan perencanaan, penggambaran, dan pembuatan sketsa atau pengaturan untuk beberapa elemen yang terpisah yang berfungsi dalam kesatuan utuh. Dapat diartikan kembali rancang bangun adalah pengembangan suatu sistem dengan tujuan mampu menciptakan sistem baru, memelihara sistem ada dan merancang pembuatan perangkat lunak [13].

2.2.2 OBRE-JEK

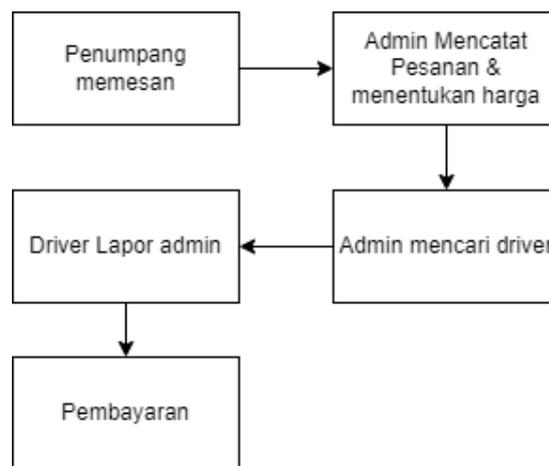


Gambar 2. 1 Poster OBRE-JEK

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. OBRE-JEK 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 *platform* sosial yang dikelola dengan tujuan membantu meningkatkan taraf pendapatan ojek lokal. Dengan mitra tidak meminta hasil pendapatan kepada *driver* atau 100% uang milik *driver*. Sistem pendataan pada OBRE-JEK masih dilakukan melalui *history* chat aplikasi *whatsapp* dan disimpan ke *email*.

2.2.2.1 Alur OBRE-JEK



Gambar 2. 2 Alur Pemesanan OBRE-JEK

Alur pemesanan OBRE-JEK adalah pelanggan menghubungi admin melalui nomor *whatsapp*, kemudian admin akan menanyakan posisi, alamat, nama, dari penumpang. Setelah itu admin akan menentukan harga setelah pelanggan setuju admin menghubungi *driver* melalui *grup* yang sudah disediakan. Admin akan memilih *driver* terdekat dari posisi penumpang. Admin melakukan pemantauan perjalanan ke penumpang dengan *Google Maps*, jika *driver* sudah bersama penumpang lapor ke admin, admin kembali memantau lewat chat. Penumpang membayar sesuai dengan kesepakatan harga kepada *driver*. Hasil dari pembayaran, 100% milik *driver* mitra tidak meminta.

2.2.3 Transaksi

Transaksi adalah proses melibatkan dua belah pihak dengan melakukan pertukaran, pinjam meminjam dan kegiatan lainnya. Transaksi penelitian pada pertukaran data dengan sistem. Proses transaksi memerlukan administrasi atau bukti pencatatan dalam Menyusun data dan informasi secara sistematis [14]. Transaksi dalam penelitian ini adalah antara admin dan *driver* ataupun admin dan penumpang dalam mengelola orderan penumpang yang akan diambil, serta menentukan tarif untuk penumpang.

2.2.4 Website

Website merupakan halaman yang berisi informasi yang terhubung dengan satu sama lain yang dapat diakses menggunakan *internet*. *Website* dapat bersifat statis dan dapat bersifat dinamis. Statis berarti isi dalam suatu *website* bersifat tetap, tidak diubah, dan informasi hanya searah. Dinamis berarti *webiste* yang dapat dirubah-rubah, serta isi dan informasinya memiliki dua arah berasal dari pemilik dan pengguna *website* [15].

2.2.4.1 Management Website

Management merupakan aktivitas mengelolah kegiatan ataupun aktivitas agar informasi dapat tersusun secara sistematis, terpantau, dan rinci. *Website* adalah halaman yang digunakan untuk menampilkan informasi berupa teks, gambar, animasi, suara, atau gabungan yang bersifat statis maupun dinamis dengan membentuk rangkaian sistem dengan dihubungkan oleh halaman-halaman yang lainnya [16]. *Management website* merupakan tampilan atau aktivitas dalam mengelola informasi atau data suatu perusahaan agar tersusun secara sistematis dan dipantau seluruh kinerja dengan baik.

2.2.4.2 Dashboard

Dashboard merupakan model sistem yang digunakan untuk menyajikan suatu informasi pada perusahaan dengan memberikan tampilan

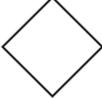
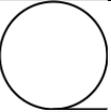
antar muka yang digunakan untuk menampung informasi berupa grafik, laporan, diagram yang bersifat *realtime* dan sesuai [3]. *Dashboard* adalah tampilan visual yang mencakup informasi berupa tampilan agar kinerja suatu perusahaan dapat dimonitor dengan baik. Kinerja yang digunakan pada *dashboard* dapat mengintegritaskan data dengan menyajikan laporan informasi yang berkualitas.

2.2.5 Flowchart

Flowchart adalah Langkah-langkah dan urutan atau prosedur dari suatu program yang divisualisasikan melalui bagan alir. *Flowchart* digunakan untuk memecahkan suatu masalah menjadi lebih kecil dalam menganalisis perancangan suatu sistem. *Flowchart* memiliki simbol berikut *table 2.2 flowchat* beserta fungsi atau kegunaannya [17].

Tabel 2. 2 Simbol *Flowchart*

No	Simbol	Arti
1.	 Input/Output	Merepresentasikan <i>input</i> data atau <i>Output</i> data yang di proses.
2.	 Proses	Mereperestasikan proses program berjalan.
3.	 Penghubung	Merepresentasikan Operasi
4.	 Flow/Anak Panah	Merepresentasikan alur kerja

No	Simbol	Arti
5.	 <i>Decision/</i> Keputusan	Menunjukkan kondisi dua kemungkinan <i>yes or no</i> atau <i>true or false</i>
6.	 Penjelasan	Digunakan untuk memberikan komentar tambahan pada diagram
7.	 <i>Punched card</i>	Input/output yang menggunakan kartu
8.	 <i>Prefined Process</i>	Rincian operasi berada di tempat lain.
9.	 Terminal Poin	Mengawali/mengakhiri <i>Flowchart</i>
10.	 <i>Preparation</i>	Proses Inisialisasi / Pemberian harga awal.
11.	 Dokumen	I/O dalam format yang dicetak
12.	 <i>Magnetic Tape</i>	I/O yang menggunakan pita <i>magnetic</i>

No	Simbol	Arti
13.	 <i>Punched Tape</i>	I/O yang menggunakan pita kertas berlubang.

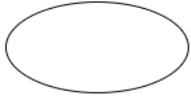
2.2.6 Unified Modeling Language (UML)

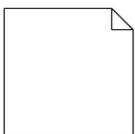
Unified Modeling Language (UML) adalah visualisasi model yang digunakan untuk membangun, merancang, menggambarkan, menspesifikasikan, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Jenis diagram UML yaitu *Diagram Use Case*, *Diagram Sequence*, *Diagram Class* [18]. UML merupakan kumpulan diagram yang berfungsi untuk melakukan perancangan perangkat lunak berbasis objek [19]. Berikut merupakan penjelasan jenis diagram UML :

2.2.7.1 Use Case Diagram

Use Case Diagram merupakan urutan yang saling terhubung antara sistem dan aktor atau biasanya akan dijalankan dengan menggambarkan gambaran interaksi antara *user* dengan sistem [19]. *Use case* adalah diagram yang menjelaskan hubungan antara aktor dan *use case* [20].

Tabel 2. 3 Simbol *Use Case*

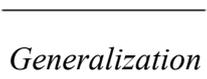
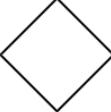
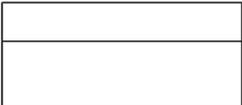
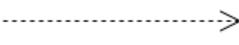
No	Simbol	Keterangan
1	 Aktor/ <i>Actor</i>	Menspesifikasikan peran pengguna untuk berinteraksi dengan sistem.
2		Urutan aksi yang ditampilkan sistem untuk

No	Simbol	Keterangan
	<i>Use Case</i>	hasil yang terukur bagi suatu <i>actor</i> .
3	 <i>Association</i>	Menghubungkan <i>object</i> satu dengan <i>object</i> yang lainnya.
4	 <i>Extend</i>	Menspesifikasikan <i>usecase</i> 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.

2.2.7.2 Class Diagram

Class diagram adalah penjelasan dari suatu proses pembuatan database yang akan digunakan suatu program [19].

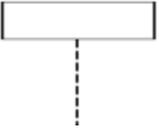
Tabel 2. 4 *Class Diagram*

No	Simbol	Keterangan
1	 <i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya 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	 <i>Dependency</i>	Hubuangan perubahan yang terjadi secara mandiri yang akan mempengaruhi elemen yang tidak mandiri.

2.2.7.3 *Sequence Diagram*

Sequence Diagram merupakan diagram yang menjelaskan interaksi dari objek dengan menunjukkan hubungan atau komunikasi antar objek [19].

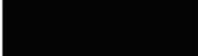
Tabel 2. 5 *Sequence Diagram*

No	Simbol	Keterangan
1	 <i>LifeLine</i>	Objek <i>entity</i> antarmuka yang saling berinteraksi.
2	 <i>Message</i>	Spesifikasi dari komunikasi anatar objek yang memuat informasi-informasi tentang aplikasi yang terjadi.
3	 <i>Message</i>	Spesifikasi dari komunikasi antara objek yang memuat informasi-informasi tentang aplikasi yang terjadi.

2.2.7.4 Activity Diagram

Activity diagram merupakan penjelasan aliran kegiatan program yang dirancang, dalam perancangan awal hingga akhir program, dengan berbagai kemungkinan terjadi [19]. *Activity diagram* adalah gambaran kerangka *flow* yang terstruktur serta dirancang untuk sistem dengan baik [20].

Tabel 2. 6 Simbol *Activity Diagram*

No	Simbol	Keterangan
1	 <i>Actifity</i>	Menunjukkan kelas berinteraksi satu sama lain.
2	 <i>Action</i>	Menunjukkan <i>state</i> dari sistem dieksekusi dari suatu aksi.
3	 <i>Initial Node</i>	Awal objek dibentuk atau diawali
4	 <i>Activity Final Node</i>	Akhir dari <i>object</i> .
5	 <i>Fork Node</i>	Mengubah aliran tertentu menjadi beberapa aliran.

2.2.7 Database

Database merupakan kumpulan data atau informasi yang disimpan pada komputer secara sistem serta digunakan untuk memperoleh informasi yang diambil melalui program dari komputer [16]. *Database* digunakan oleh suatu sistem untuk menyimpan data atau informasi dengan kapasitas lebih besar dan menampung semua jenis karakter dan tipe datanya. *Database* memiliki kelebihan yaitu mampu mengurangi *budget* perangkat, keamanan data yang aman, efektif dan efisien, *Multi-user*, memudahkan pembuatan aplikasi, serta kontrol data yang terpusat. Peneliti menggunakan menggunakan *MySQL* dan *Xampp* sebagai pengelolaan data atau informasi yang akan digunakan dalam penelitian.

2.2.8.1 XAMPP

Xampp merupakan aplikasi yang digunakan sebagai server *localhost* atau berdiri sendiri. *Xampp* memiliki beberapa program yaitu *Apache HTTP Server*, *MySQL database* dan penerjemah bahasa ke dalam bahasa pemrograman *PHP*, *Perl*. *Xampp* tersedia didalam GNU (*General Public License*), adalah *web server* yang digunakan untuk memudahkan halaman web dinamis[21] .

2.2.8.2 MYSQL

MySQL merupakan *software* berbasis Struktur *Query Language (SQL)* yang termasuk dari *Database Management System* yang bersifat *Open Source*. *Mysql* adalah *software* RDBMS (*Relational Database Management System*) atau *server database* yang digunakan dalam mengelola *database* dengan efektif, mampu menampung data yang cukup besar, *multi-user* diartikan dapat diakses oleh *user* , *multi-threaded* atau melakukan proses dengan bersamaan atau sikron [18][16].

2.2.8 HTML

HTML merupakan kepanjangan dari *Hypertext Markup Language*. HTML merupakan bahasa *markup* standar dalam membuat suatu *website* yang memungkinkan *user* membuat bagian heading, paragraf, link, atau kerangka yang dibutuhkan Ketika akan membangun *website*. HTML dirancang menggunakan konsep markup yang dikenal *Standard Generalized Markup Language*. HTML merupakan struktur kode sederhana dan mudah dipahami dengan *tag*, atribut sederhana dan dasar [19] [22].

2.2.9 Tailwind

Tailwind CSS adalah salah satu *framework* yang digunakan dalam perancangan dan pengembangan *front-end* dengan waktu yang singkat. *Tailwind* CSS berbeda dengan *framework* CSS lainnya seperti *Bootstrap*,

karena *Tailwind* CSS prioritas utama daripada *Bootstrap* yang menggunakan komponen yang dapat modifikasi dari gaya tampilan *website*. *Tailwind* CSS menggunakan konsep *utility first* atau dapat menyederhanakan CSS dengan kode yang telah disediakan, *Tailwind* dapat didesain dan dapat merubah formatnya [23].

2.2.10 PHP

PHP merupakan kepanjangan dari PHP (*Hypertext Preprocessor*), PHP merupakan bahasa pemrograman yang biasa digunakan dalam proses pembuatan dan pengembangan sebuah *website* dan biasa digunakan pada HTML [21]. PHP memiliki persamaan dengan bahasa pemrograman C dan Perl dengan perintah-perintah sederhana untuk membangun suatu web. Penggunaan PHP dapat mempermudah dalam *maintenance* sebuah *website* serta proses *update* dapat dijalankan dengan aplikasi yang dibangun dengan *script* [12].

2.2.11 *Laravel*

Laravel adalah *framework* bahasa pemrograman PHP yang mempunyai banyak dokumentasi sehingga membantu *developer* dalam mengembangkan suatu sistem berbasis *website* dengan menerapkan konsep MVC (*Model View Controller*). *Laravel* digunakan *developer* dalam *development website* karena meningkatkan kualitas aplikasi, mengurangi *budget* pengembangan, memudahkan dalam *maintenance* serta mampu meningkatkan produktifitas pekerjaan dengan kode program yang rapih dan terstruktur. *Laravel* memiliki kelebihan diantaranya [24] [25] :

- a. Menggunakan *Command Line Interface* (CLI).
- b. Menggunakan *Package Manager PHP Compressor*.
- c. Penulisan kode yang rapih, singkat, dan terstruktur.
- d. Mudah dipahami oleh *developer*.

2.2.12 Agile

Agile merupakan metode dalam *management* proyek dengan menggunakan siklus pengembangan yang singkat atau dikatakan dengan *sprint* dengan tujuan meningkatkan dalam suatu pengembangan sistem pada produk dan layanan. Pengembangan *agile* memiliki model diantaranya *agile modelling*, *crystal*, *dynamic system development*, *adaptive software development*, *feature driven development*, *extreme programming (XP)*, dan *Scrum* [26].



Gambar 2. 3 Siklus Agile Model

Pada Gambar 2.2 merupakan siklus dari *agile methodology* dengan siklus adalah *planning*, *design*, *development*, *test*, *deploy*, *review*, jika sudah tidak diperlukan maka *finish* atau *launching* produk dari sistem yang dikembangkan. *Agile* model dilakukan terus-menerus sesuai dengan kebutuhan *backlog*.

2.2.13 Scrum Model

Scrum model bagian dari metodologi *agile*, yang merupakan metodologi dalam pengembangan perangkat lunak dengan mengutamakan progres dari suatu proyek. *Scrum* model adalah kerangka kerja dalam menyelesaikan permasalahan yang kompleks dengan setiap proyek memiliki perubahan atau progresnya untuk menghasilkan produk dengan kualitas baiknya. *Scrum* model bersifat ringan, mudah dipahami, namun sulit dikuasai. *Scrum* memiliki tiga pilar yaitu [27] :

A. Transparansi

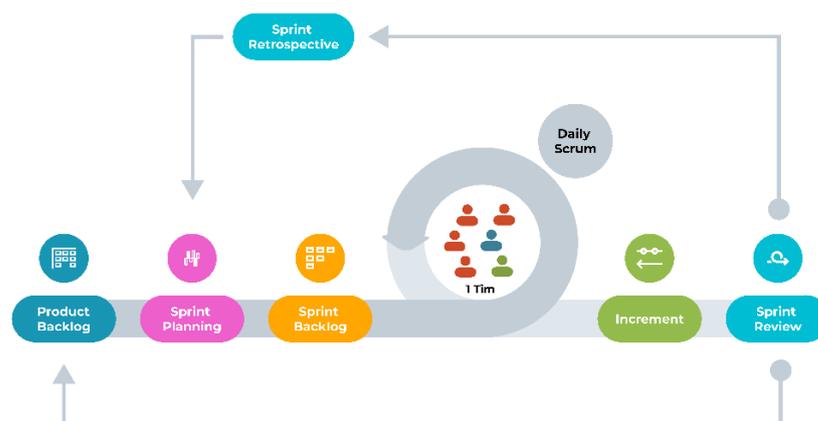
Proses dan pekerjaan yang muncul dapat dilihat saat melakukan pekerjaan serta menerima hasil pekerjaan. Scrum memiliki keputusan penting dengan dasar tiga artefak formal yaitu artefak yang memiliki transparansi rendah dapat mengarah kepada keputusan yang mengurangi nilai dan meningkatkan risiko.

B. Inspeksi

Artefak scrum merupakan tujuan yang disepakati harus sering diperiksa untuk mendeteksi masalah terjadi secara seksama.

C. Adaptasi

Scrum Team diharapkan untuk beradaptasi seketika untuk mempelajari sesuatu yang baru melalui inspeksi.



Gambar 2. 4 Siklus Scrum Model

Pada gambar 2.3 merupakan siklus dari *Scrum* Model diperoleh dari scrum.org dengan tahapan yaitu *product backlog*, *Sprint Planning*, *Daily Scrum*, *Increment*, *Sprint Review* atau Kembali ke *Sprint Retrospective* atau *Product Backlog* sesuai dengan kebutuhan yang diperlukan dalam pengembangan sistem berikut penjelasannya [27] [28] :

A. *Product Backlog*

Product Backlog merupakan daftar atau susunan yang dibutuhkan untuk meningkatkan suatu produk. *Product backlog* merupakan tanggung jawab yang dipegang oleh *product owner* sebab

daftar fitur yang diinginkan ada pada *product owner*. Sebelum menentukan product backlog adalah menentukan user story dimana user story merupakan alur atau rancangan yang diinginkan oleh seorang product owner terkait sistem. Developers memiliki tanggung jawab untuk menentukan skala ukuran untuk menentukan prioritas taks yang ditentukan sesuai dengan kesepakatan dengan product owner untuk menentukan urgensi fitur yang dibuat meliputi [27] :

1. High memiliki arti sistem yang dibuat adalah fitur utama atau website yang dibuat.
2. Medium memiliki arti sistem yang dibuat adalah merupakan yang harus
3. Low memiliki arti fitur opsional jika ada sistem tersebut akan menjadi lebih baik.

B. *Sprint Planning*

Sprint planning merupakan kegiatan yang mengatur pekerjaan yang akan dilakukan selama *sprint* dimulai. *Sprint planning* memiliki batasan waktu maksimum delapan jam untuk *sprint* selama satu bulan. Pembagian waktu pada *sprint* bergantung dengan kebutuhan, durasi satu *sprint* dua minggu ataupun empat minggu (satu bulan). Durasi *sprint* diperbolehkan satu minggu dalam satu kali *sprint* semua tergantung dengan kebutuhan [29] .

C. *Sprint Backlog*

Sprint backlog merupakan rencana untuk *developers*, dimana memiliki gambaran waktu dari pekerjaan yang direncanakan *Developers* selama satu *sprint* untuk mencapai *sprint goal*. *Sprint backlog* harus memiliki perincian yang cukup sehingga dapat memeriksa *daily scrum*. Pada tahapan *sprint backlog* menentukan *story point* yang digunakan untuk menggambarkan estimasi keseluruhan usaha yang diperlukan dalam menyelesaikan suatu

product backlog. Dalam menentukan *story point* terdapat pendekatan yang digunakan dengan satuan waktu, tingkat kerumitan, dan pendekatan lainnya yang disesuaikan dengan jenis pekerjaannya. *Story point* dapat ditentukan dengan menggunakan deret *Fibonacci* yang terdiri dari 1,2,3,5,8 dan seterusnya. Dalam menentukan *story point* ada beberapa hal yang perlu diperhatikan [30] :

1. Banyaknya Pekerjaan yang akan dilakukan.
2. Kerumitan pekerjaan
3. Resiko ketidak pastian.

D. *Daily Scrum*

Daily Scrum memiliki tujuan untuk memeriksa kemajuan menuju *sprint goal* dan menyesuaikan *Sprint backlog*, serta merencanakan rancangan kerja yang akan datang. *Daily scrum* digunakan untuk meningkatkan komunikasi, mengidentifikasi hambatan, mendorong pengambilan keputusan dengan cepat, dan menghilangkan kebutuhan untuk keperluan pertemuan lainnya.

E. *Increment*

Increment merupakan pijakan menuju *product goal*, penambahan adalah tambahan semua *increment* sebelumnya dan telah diverifikasi secara keseluruhan dengan memastikan fungsi semua *increment* berfungsi secara bersamaan.

F. *Sprint Review*

Sprint review merupakan untuk memeriksa hasil dari sprint dan menentukan adaptasi selanjutnya. *Sprint Review* memiliki batasan waktu yaitu empat jam untuk *sprint* selama satu bulan.

G. *Sprint Retrospective*

Sprint Retrospective merupakan cara untuk meningkatkan kualitas dan efektivitas. *Sprint retrospective* dilakukan scrum team untuk mengkaji bagaimana berjalannya sprint terakhir yang berkaitan dengan individu, interaksi, proses, alat, dan *Definition of done*. *Sprint retro* memiliki batasan waktu yaitu 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 merupakan orang yang memiliki kuasa atau pengetahuan mengenai produk yang akan dikembangkan .

B. *Scrum Master*

Scrum master merupakan orang yang memiliki tanggung jawab atau pemimpin yang memiliki keahlian mengenai *scrum* untuk mengarahkan tim *scrum* saat bekerja dengan tujuan yang benar dalam proses pengembangan perangkat lunak.

C. *Development Team*

Development Team merupakan orang-orang yang mengimplementasikan produk dalam pengembangan sebuah sistem kedalam program sesuai dengan kebutuhan sistem.

Scrum model memiliki pendekatan yang digunakan dalam aktivitas pengembangannya yaitu *Backlog* merupakan *list* kebutuhan sistem yang diinginkan oleh *clien*, *Sprint* merupakan fase kerja yang digunakan dalam pengembangan sistem sesuai dengan kebutuhan *list* pada *backlog* dengan waktu pada *time-box*, *Scrum meetings* merupakan aktivitas *meeting* atau pertemuan rutin untuk mengevaluasi pekerjaan, kendala, serta taget dari apa

yang akan dikerjakan selanjutnya, dan demo merupakan *review* yang berkaitan dalam peningkatan perangkat lunak kepada *client* [31].

2.2.14 Pengujian Sistem

Pengujian sistem merupakan proses penting untuk memeriksa kesalahan – kesalahan pada sistem yang dibangun dan untuk mengurangi terjadinya kerugian pada sistem. Pengujian sistem berfungsi untuk mengurangi kesalahan yang ada sehingga sistem dapat diperbaiki menjadi lebih baik. Pengujian sistem terbagi menjadi 2 yaitu *Black Box Testing* dan *Laravel Dusk Testing*[32].

2.2.15.1. *Blackbox Testing*

Blackbox Testing adalah salah satu metode untuk menguji perangkat lunak yang telah dibangun. Pada pengujian perangkat lunak ini tidak menguji desain, kode program, namun hanya untuk menguji fitur berfungsi dan *output* dari perangkat lunak sesuai dengan spesifikasi kebutuhan [33]. *Blackbox Testing* berfokus pada pengujian spesifikasi fungsionalitas dari suatu perangkat lunak. *Blackbox testing* digunakan untuk mendeteksi masalah berikut [34]:

1. Fungsi yang salah atau hilang
2. Kesalahan pada *interface*
3. Kesalahan struktur data atau basis data
4. Kesalahan deklarasi dan terminasi

Blackbox Testing digunakan untuk mengetahui kelemahan dari sistem untuk mengetahui *output* sesuai dengan input yang diharapkan setelah dieksekusi dan untuk menghindari kekurangan dan kesalahan aplikasi sebelum digunakan oleh *user*.

2.2.15.2. *Laravel Dusk Testing*

Laravel Dusk Testing merupakan fitur yang dimiliki oleh *laravel* yang berfungsi untuk melakukan pengujian terhadap suatu sistem. *Laravel Dusk* menguji sistem secara terprogram dengan menggunakan *browser* untuk melakukan pengujiannya untuk memastikan sistem berjalan dengan baik pada *browser* [35]. Selain itu pengujian ini dilakukan untuk menguji efisiensi dari fungsi yang dibuat dengan hasil yang diperoleh akan digunakan untuk memperbaiki fungsi yang dibuat. Pengujian ini dilakukan dengan membuat program sesuai perintah yang akan dilakukan [36]. *Dusk* akan menampilkan hasil dari *testing* yang sudah selesai dengan hasil berupa informasi *success* atau *error* [35].