

## **BAB II** **TINJAUAN PUSTAKA**

### **1.1 Sistem Informasi**

Definisi menurut Nas (2018) Sistem informasi adalah sistem pemrosesan data, merupakan sistem buatan manusia yang biasanya terdiri dari sekumpulan komponen (baik manual maupun berbasis komputer) yang terintegrasi untuk mengumpulkan, menyimpan, dan mengelola data serta menyediakan informasi mengenai saldo persediaan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan data yang terintegrasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

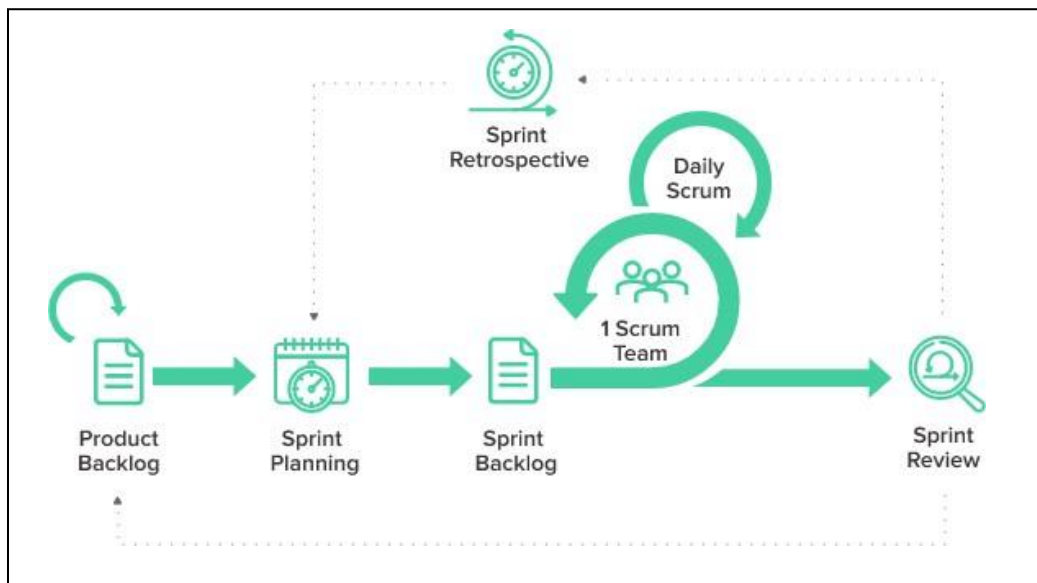
### **1.2 Pariwisata**

Menurut Suwena and Widayatmaja(2017)Pariwisata adalah keseluruhan jaringan dan gejala-gejala yang berkaitan dengan tinggalnya orang asing disuatu tempat dengan syarat orang tersebut tidak melakukan suatu pekerjaan yang penting (*Major Activity*) yang memberi keuntungan yang bersifat permanent maupun sementara.Jadi dapat di katakan pada dasarnya pariwisata itu motif kegiatannya adalah untuk mengisi waktu luang, untuk bersenang-senang, bersantai, studi, kegiatan Agama, dan mungkin untuk kegiatan olahraga. Selain itu semua kegiatan tersebut dapat memberi keuntungan bagi pelakunya baik secara fisik maupun psikis baik sementara maupun dalam jangka waktu lama,serta untuk prospek jangka panjangnya (*sustainable tourism*).

### **1.3 Metode Scrum**

Menurut Adi & Permana (2015) dalam penelitiannya yang berjudul “*Scrum Method Implementation in a Software Development Project Management* ” dalam(Julianto, 2019) menyatakan bahwa *Scrum* pertama kali dikembangkan oleh Schwaber dan Sutherland pada tahun 1993 dan tujuannya adalah menjadi metodologi pengembangan yang mengikuti prinsip-prinsip metodologi *Agile*.

Menurut (Schwaber & Sutherland, 2017) *Scrum* adalah suatu metodologi atau kerangka kerja yang terstruktur untuk mendukung pengembangan produk yang kompleks. *Scrum* terdiri dari sebuah tim yang memiliki peran dan tugas masing- masing. Setiap komponen dalam kerangka melayani tujuan tertentu dan sangat penting untuk kesuksesan penggunaan *Scrum*.



Gambar 2.4 Tahapan-Tahapan Metode Scrum

### 2.3.1 Tahapan-Tahapan Metode *Scrum*

Adapun tahapan-tahapan dalam *Scrum* dalam (Schwaber & Sutherland, 2017) adalah sebagai berikut ini:

#### 1. ***Product Backlog***

Peneliti sistem akan mengumpulkan dan menyusun semua kebutuhan sistem dan permintaan pengguna terhadap sistem, misalnya fitur-fitur yang dibutuhkan oleh pengguna terhadap sistem. *Product backlog* berada dalam tanggung jawab *product owner*. Setelah targetnya ditetapkan, semua kebutuhan dan permintaan akan dibagikan menjadi poin-poin kecil yang mana setiap poin tersebut mempunyai tingkat layak untuk dikembangkan.

#### 2. ***Sprint Planning***

*Sprint Planning* merupakan sebuah langkah yang wajib dilaksanakan setiap saat akan memulainya sprint baru. Pada langkah tersebut peneliti akan menyusun pekerjaan-pekerjaan apa saja yang harus diselesaikan dalam 1 *sprint*.

#### 3. ***Sprint Backlog***

Perencanaan *sprint* dilakukan dalam pertemuan/*meeting* antara pemilik produk dan tim developer, yang akan berkolaborasi untuk memilih produk yang akan dikembangkan *backlog* untuk dimasukkan kedalam proses *sprint*. Hasil dari pertemuan tersebut adalah *sprint backlog*.

#### 4. ***Sprint***

Dalam *Scrum*, *Sprint* adalah sebuah kerangka waktu yang berdurasi maksimal 1 bulan untuk mengembangkan produk yang berpotensi untuk dirilis. Dalam *Sprint* terdapat 2 bagian pekerjaan, yaitu:

a. Pertemuan Harian (*Daily Standup Meeting*)

Merupakan pertemuan dimana setiap 24 jam (1 hari), tim pengembang bertemu untuk membahas proses pengembangan produk.

b. Refleksi *Sprint*

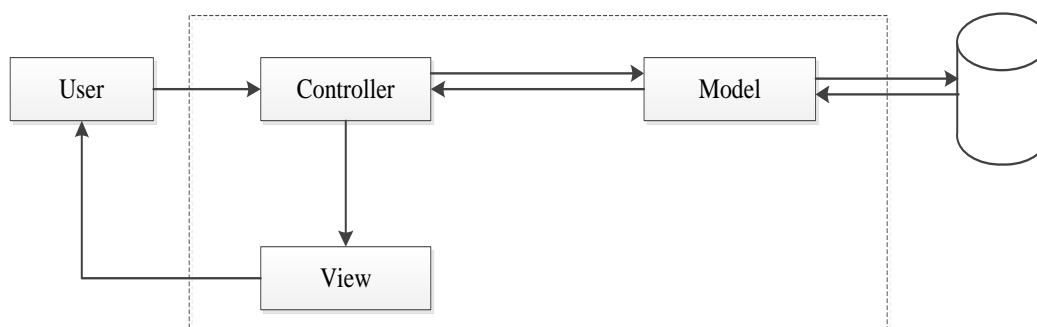
Merupakan pertemuan yang dilakukan setiap bulannya, yang bertujuan untuk membahas hal dari *Sprint Backlog* yang telah berjalan dan telah berhasil dikerjakan, serta dapat memperbaiki dan meningkatkan kualitas produk pada *Sprint* yang berikutnya.

5. **Working Increment (*Sprint Rivew*)**

*Increment* merupakan hasil dari seluruh hal dalam *product backlog* yang telah selesai dikerjakan pada seluruh *sprint*

#### 1.4 *CodeIgniter*

*CodeIgniter* menurut Raharjo (2018) merupakan *framework* untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. *CodeIgniter* memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat mengembangkan dalam perangkat *web*, dekstop maupun *mobile*. *CodeIgniter* memiliki konsep atau pola *Model-View-Controller* (MVC) sehingga kode-kode dapat di sederhanakan.



**Gambar 1.1** Arsitektur *Model-View-Controller* (MVC)

Berdasarkan arsitektur tersebut diketahui konsep *Model-View-Controller* yaitu:

1. *Model*

*Model* merupakan suatu fungsi yang digunakan mengelola database pada sistem untuk menangani struktur data dari *controller*.

## 2. View

*View* merupakan bagian untuk mengelola tampilan dari *website* atau dapat disebut sebagai *user interfac* yang diatur bagian *controller*.

## 3. Controller

*Controller* merupakan kunci dalam konsep MVC dengan fungsi untuk menghubungkan *model* dengan *view*.

### 2.4.1 Web Based

*Web Based* menurut Urbietta *et al.* (2019) adalah aplikasi yang dibuat berbasis web yang membutuhkan *web server* dan *browser* untuk menjalankannya. Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).

### 2.4.2 Perl Hypertext Preprocessor (PHP)

Menurut Subagja (2018) *Perl Hypertext Preprocessor* adalah bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis.

Jadi *Perl Hypertext Preprocessor* merupakan bahasa pemograman *web server-side* yang bersifat *open source* atau mudah dikembangkan. *Perl Hypertext Preprocessor* adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*.

### 2.4.3 MySql

Menurut MySQL (2018) *MySQL* adalah singkatan dari *Structue Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan basis data.

Menurut Amin (2018) mendefinisikan *mysql* *MYSQL* adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

*MySQL* merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

## 2.5 Alat Pengembang Sistem (*Unified Modelling Language*)


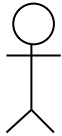

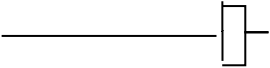
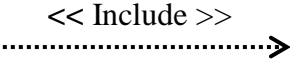
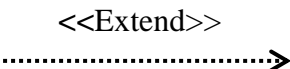
Alat pengembang sistem merupakan konsep desain yang digunakan untuk menggambarkan sistem dengan menggunakan diagram. Penyesuaian alat yang digunakan

harus sesuai dengan metode pengembangan yang dilakukan salah satunya adalah penerapan *Unified Modelling Language*. Menurut (Rosa and Shalahuddin, 2019) UML (*unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

### 2.5.1 Use Case Diagram

Menurut Rosa and Shalahuddin (2019) *Use Case* adalah *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.1 berikut ini:

**Tabel 1.1** Simbol *Use Case Diagram*



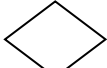

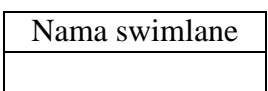

No	Simbol	Deskripsi
1.		Usecase Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi ( <i>generalization</i> ) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.		Ekstensi ( <i>extend</i> ) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

**Sumber** : (Rosa and Shalahuddin, 2019)

### 2.5.2 Activity Diagram

Menurut Rosa and Shalahuddin (2019) *activity diagram* adalah *activity Diagram* menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activitydiagram* dapat dilihat pada tabel 2.3 berikut ini :

**Tabel 1.2** Simbol *Activity Diagram*

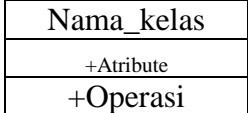
No.	Simbol	Keterangan
1.		Status awal aktivitas sitem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan ( <i>Decision</i> ) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		Penggabungan ( <i>Join</i> ) merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.		Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
6.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.



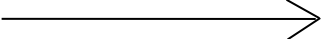
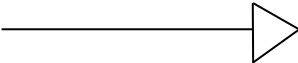
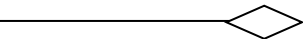
**Sumber :** (Rosa and Shalahuddin, 2019)

### 2.5.3 Class Diagram

Menurut Rosa and Shalahuddin (2019) *Class Diagram* adalah *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada tabel 2.3 berikut ini :

**Tabel 1.3** Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.

2.	Antar Muka/Interface  Nama_Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / Asociation 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / Directed Association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Agregasi / aggregation 	Relasi antar kelas dengan maksna semua bagian ( <i>whole-part</i> )

**Sumber:** (Rosa and Shalahuddin, 2019).

## 2.6 User Acceptance Test

Menurut Ayunda *et al.*, (2017), *User Acceptance Test* merupakan proses verifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai untuk pengguna. Proses ini berbeda dengan pengujian sistem (memastikan *software* tidak *crash* dan sesuai dengan dokumen permintaan pengguna), melainkan memastikan bahwa solusi dalam sistem tersebut akan bekerja untuk pengguna (yaitu, tes bahwa pengguna menerima solusi di dalam sistem).

*User Acceptance Test* umumnya dilakukan oleh klien atau pengguna akhir, biasanya tidak fokus pada identifikasi masalah sederhana seperti kesalahan ejaan, maupun di cacat *showstopper*, seperti *crash* perangkat lunak. Penguji dan pengembang mengidentifikasi dan memperbaiki masalah ini selama tahap awal pengujian fungsionalitas, pengujian saat integrasi dan pada tahap sistem testing.

## 2.7 Tinjauan Pustaka

Tinjauan pustaka digunakan sebagai refrensi dalam penelitian yang dapat dilihat sebagai berikut:

**Tabel 1.4** Tinjauan Pustaka

NO.	Judul Jurnal	Penulis	Link Jurnal dan Nama Jurnal	Metode
1.	IMPLEMENTASI	Novri	<u>N</u> Hadinata, <u>M</u> Nasir -	Metode

	METODE SCRUM DALAM RANCANG BANGUN SISTEM INFORMASI PENJUALAN (STUDY KASUS : PENJUALAN SPERPART KENDARAAN)	Hadinata1 , Muhammad Nasir2	JURNAL ILMIAH BETRIK ..., 2017 - ejournal.lppmsttpagaralam.ac.id	scrum
2.	Analisis Penerapan Metode Scrum Pada Sistem Informasi Manajemen Proyek Dalam Industri & Organisasi Digital	Hisyam Rahmawan Suharno1 , Nyoman Gunantara2 , Made Sudarma	HR Suharno, N Gunantara... - Majalah Ilmiah ..., 2020 - pdfs.semanticscholar.org	Metode scrum
3.	Penerapan Metode SCRUM dalam Pengembangan Sistem Informasi Layanan Kawasan	Muhamad Hanif Muslim*2 , Farham Harvianto#3 , Setiawan Utama#4	<u>W Warkim</u> , MH Muslim, <u>F Harvianto...</u> - JuTISI (Jurnal Teknik ..., 2020 - journal.maranatha.edu	Metode scrum
4.	PETA DIGITAL OBJEK PARIWISATA DAN KULINER DI KABUPATEN JEPARA BERBASIS MOBILE	Riza Abi Emawan1 , Nova Rijati,SSi, M.Kom2	R ABI EMAWAN - eprints.dinus.ac.id	Metode scrum



	MENGGUNAKAN METODE SCRUM			
5.	Pengembangan sistem promosi pariwisata	Bambang Tjahjo Utomo	BT Utomo - GLOBAL, 2017 - ejournal.unsub.ac.id	Metode scrum
6.	Implementasi Sistem Informasi Multimedia dalam Meningkatkan UMKM di Kota Pagar Alam	iskandar Yogi Isro Mukti	<u>YI Mukti</u> - JURNAL ILMIAH BETRIK: Besemah ..., 2021 - ejournal.lppmsttpagaralam.ac.id	Metode scrum
7.	RUNNING TEXT ERA 4.0 SEBAGAI ALAT PEMANTAUAN KONDISI LINGKUNGAN KAWASAN WISATA BEDENGAN DESA SELOREJO	<i>Alief Sias, Mahfud Jiono, Yogi Dwi Mahand, Soraya Norma Mustika</i>	<u>QA Sias, M Jiono,</u> YD Mahand... - E-Prosiding ..., 2020 - conference.um.ac.id	Model scrum