

BAB II

LANDASAN TEORI

2.1 Pengertian Wisata dan Wisatawan

Menurut Kamus Besar Bahasa Indonesia (KBBI), wisata merupakan berpergian bersama-sama dengan tujuan bersenang-senang, menambah pengetahuan, dan lainnya. (Badan Pengembangan dan Pembinaan Bahasa, 2019a). Sedangkan wisatawan adalah orang yang mengadakan perjalanan dari tempat kediamannya tanpa menetap di tempat yang didatanginya atau hanya untuk sementara waktu tinggal di tempat yang didatanginya (Badan Pengembangan dan Pembinaan Bahasa, 2019).

2.2 Sistem Rekomendasi

Sistem rekomendasi merupakan model aplikasi dari hasil observasi terhadap keadaan dan keinginan pelanggan. Oleh karena itu sistem rekomendasi memerlukan model rekomendasi yang tepat agar yang direkomendasikan sesuai dengan keinginan pelanggan, serta mempermudah pelanggan mengambil keputusan yang tepat dalam menentukan produk yang akan digunakannya (Kadyanan, 2017).

Menurut Chandra (Prasetya, 2017), sistem rekomendasi membantu pengguna untuk mengidentifikasi produk yang sesuai dengan kebutuhan, kesenangan, dan keinginan *user*. Sedangkan menurut Aryani, Susilo & Setiawan (Aryani, Susilo, & Setiawan, 2019), sistem rekomendasi adalah program perantara atau perwakilan yang secara

cerdas menyusun daftar dari informasi yang diperlukan dan mencocokkan berdasarkan keinginan dari pengguna.

2.3 PHP

Menurut Astria Firman, Hans F. Wowor, dan Xaverius Najoran (Firman, Wowor, & Najoran, 2016), PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skrip HTML. Bahasa PHP juga dapat dikatakan sebagai gambaran beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari.

2.4 MySQL

Menurut Herman Yuliansyah (Yuliansyah, 2014), MySQL adalah sebuah *database management system* (DBMS) populer yang memiliki fungsi sebagai *relational database management system* (RDBMS).

2.5 Pengertian UML (*Unified Modeling Language*)


UML bukanlah hasil dari satu individu tetapi, merupakan upaya kolektif dari banyak praktisi, ahli metodologi, pemikir, dan penulis. UML pertama kali diusulkan sekitar tahun 1995 sebagai kombinasi dari tiga metode (proses) yang paling populer saat itu, *Booch*, *Object Modeling Technique*, dan *Objectory*. UML sendiri berguna



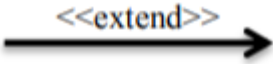
dengan tujuan untuk memodelkan, mengembangkan, dan memelihara sistem perangkat lunak (Unhelkar, 2018).


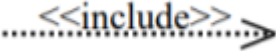


2.5.1 Use Case Diagram

Use case diagram adalah model persyaratan sistem pada level tinggi. *Use Case Diagram* digunakan untuk memvisualisasikan *Use Case*, sektor yang terkait beserta dengan interaksinya. Model visual dari penggunaan *Use Case* memfasilitasi pemahaman proses bisnis dan bantuan dalam komunikasi dengan pemangku kepentingan. Spesifikasi dan dokumentasi *Use Case* yang ditunjukkan dalam bentuk diagram dapat membentuk inti dari pemodelan persyaratan (Unhelkar, 2018).

Tabel 2. 1 Simbol Use Case Diagram

Nomor & Nama Simbol	Simbol	Deskripsi
<i>Use Case</i>		<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal <i>frase</i> nama <i>Use Case</i></p>

Nomor & Nama Simbol	Simbol	Deskripsi
Aktor / <i>Actor</i>		<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama aktor</p>
Asosiasi / <i>Assosiation</i>		<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
Ekstensi / <i>Extend</i>		<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dinamakan <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>intherince</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>


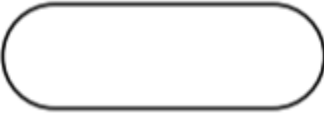


Nomor & Nama Simbol	Simbol	Deskripsi
<i>Inherits / Generalize</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i>		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>
<i>System Boundary</i>		Simbol yang memisahkan aktor dari <i>use case</i> . Semua aktor ada di luar batas dan kasus penggunaan ada di dalamnya.
<i>Notes</i>		Simbol ini dapat digunakan untuk menjelaskan hubungan <i>use case</i> , memberikan informasi tambahan tentang aktor (sebagai bentuk catatan)

Sumber : (Unhelkar, 2018).

2.5.2 Activity Diagram

Activity diagram menggambarkan aliran kerja, atau proses, dalam suatu sistem. Karena itu, mereka tergambar seperti *flowchart*. *Activity Diagram* mendokumentasikan perilaku internal yang ada dalam *Use Case*, antara *Use Case*, atau bisnis secara keseluruhan (Unhelkar, 2018)

Tabel 2. 2 Simbol Activity Diagram

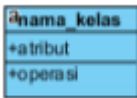



Nomor & Nama Simbol	Simbol	Deskripsi
<i>Start</i>		Simbol <i>start</i> untuk menyatakan awal dari suatu proses
<i>Activity</i>		Simbol <i>activity</i> menyatakan aksi yang dilakukan dalam suatu arsitektur sistem.
<i>Decision</i>		Simbol <i>decision</i> digunakan untuk menyatakan kondisi dari suatu proses
<i>Stop</i>		Simbol <i>stop</i> untuk menyatakan akhir dari suatu proses


Sumber : (Unhelkar, 2018)

2.5.3 Class Diagram

Class diagram adalah salah satu diagram yang paling populer bagi *software engineering*. *Class Diagram* mewakili entitas kunci dalam bisnis serta domain teknis. *Class Diagram* sifatnya sangat struktural dan statis. *Class Diagram* dapat menunjukkan kelas yang ada pada tingkat bisnis, juga dengan kelas pada tingkat teknis yang berasal dari bahasa implementasi (misalnya *Java* atau *C++*) (Unhelkar, 2018).

Tabel 2. 3 Simbol Class Diagram

Nomor & Nama Simbol	Simbol	Deskripsi
Kelas		Kelas pada struktur sistem
Asosiasi / Association		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Inheritance		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Agregasi / Aggregation		Relasi antarkelas dengan makna semua bagian (whole-part)

Nomor & Nama Simbol	Simbol	Deskripsi
Multiplicity		Relasi antara dua kelas yang membawa informasi tentang jumlah objek (contoh) yang diperhitungkan pada setiap akhir asosiasi.

Sumber : (Unhelkar, 2018).

2.6 Metode Sistem Pendukung Keputusan *Analytical Hierarchy Process*

Menurut Evan Rosiska (Rosiska, 2018), Metode *Analytical Hierarchy Process* (AHP) merupakan metode yang banyak digunakan dalam kasus pembobotan kriteria dan penentuan prioritas setiap kriteria berdasarkan matriks perbandingan berpasangan.

2.7 Metode Sistem Pendukung Keputusan *Weighted Product* (WP)

Menurut Aliy Hafiz (Aliy Hafiz & Muhammad Ma'mur, 2018), metode *Weighted Product* merupakan suatu metode pengambilan keputusan yang efisien dalam perhitungan, selain itu waktu yang dibutuhkan lebih singkat dan banyak digunakan untuk menyelesaikan permasalahan dengan melakukan perkalian antar nilai kriteria yang telah ditentukan, yang dimana nilai dari setiap kriteria harus dipangkatkan terlebih dahulu dengan bobot kriteria yang telah ditetapkan diawal.

2.8 Metode Sistem Pendukung Keputusan *TOPSIS*

Menurut Darsono Nababan (Nababan & Rahim, 2018), *TOPSIS* adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang. *TOPSIS* menggunakan prinsip bahwa alternatif yang terpilih

harus mempunyai jarak terdekat dari solusi ideal positif dan jarak terpanjang (terjauh) dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* (jarak antara dua titik) untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal.

2.9 Metode Pengembangan Sistem *Rapid Application Development* (RAD)

RAD merupakan model proses perangkat lunak yang menekankan pada daur pengembangan hidup yang singkat. RAD merupakan versi adaptasi cepat dari model *waterfall*, dengan menggunakan pendekatan konstruksi komponen (Pressman, 2012)..



Gambar 2. 1 Rapid Application Development

Sumber : (Pressman, 2012)

Tahapan RAD terdiri dari 3 tahap yang terstruktur dan saling bergantung disetiap tahap, yaitu :

1.) *Requirement Planning* (Perencanaan Persyaratan)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut (Kenneth E. Kendall, 2010).

2.) *Design Workshop*

Fase ini adalah fase untuk merancang dan memperbaiki yang digambarkan sebagai *workshop*. Fase ini juga digunakan untuk menyempurnakan desain. Penganalisis dan pemrogram dapat membuat pola desain yang dapat divisualisasikan. Programmer dan analis membangun dan menunjukkan visual yang telah dibuat guna mendapatkan *feedback* dari pengguna. Nantinya *feedback* tersebut akan digunakan untuk menyempurkan prototipe yang telah dirancang berdasarkan *feedback* pengguna. Apabila seorang pengembang sistem / pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi (Kenneth E. Kendall, 2010).

3.) *Implementation* (Penerapan)

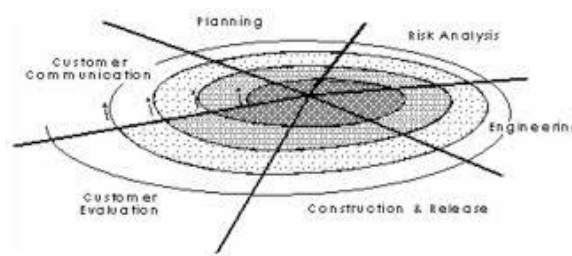
Pada fase implementasi atau penerapan ini, penganalisis bekerja dengan pengguna selama *workshop* (tahap ke 2) dan merancang aspek-aspek bisnis.

Nantinya sistem ini akan diuji dan diperkenalkan kepada pengguna. Dan setelah aspek-aspek diatas sudah terpenuhi dan diuji, kemudian sistem ini sudah siap diperkenalkan (Kenneth E. Kendall, 2010).

- a.) Sebagai sistem yang baru dibangun, sistem ini akan diuji dan diperkenalkan kepada pihak pengguna
- b.) Ketika membuat sistem baru, tidak perlu menjalankan sistem yang lama

2.10 Metode Pengembangan Sistem *Spiral*

Metode *Spiral* (*Spiral Model*) adalah model proses software yang evolusioner yang merangkai sifat iterative dari prototipe dengan cara control dan aspek sistematis dari model sekuensial linier. Model ini berpotensi untuk pengembangan versi pertambahan *software* secara cepat. Dalam pengembangan sistem informasi berbasis web, model ini digunakan untuk menyelesaikan secara global terlebih dahulu, kemudian untuk fitur dari sistem akan dikembangkan kemudian. (Utomo & Alfaridzi, 2018)



Gambar 2.2 Spiral

Sumber : (Utomo & Alfaridzi, 2018)

2.11 Metode Pengembangan Sistem *Agile Scrum*

Metode *Scrum* adalah metode pengembangan peranti lunak secara cepat (*agile*). Prinsip *scrum* sesuai dengan prinsip-prinsip yang terdapat pada metode pengembangan peranti lunak, seperti pemenuhan kebutuhan, analisa, desain, dan penyampaian (*delivery*) (Mahendra & Eby Yanto, 2018).

2.12 Penelitian Terdahulu

Tabel 2. 4 Penelitian Terdahulu

Nama Peneliti	Nama Jurnal	Judul Penelitian	Hasil Penelitian	Hasil Yang Diambil
Listiani, Friska Natalia	ULTIMA InfoSys, Vol. X, No. 1	Rancang Bangun Aplikasi Manpower Request Bagian Recruitment Pada PT. Dynapack Asia (2019)	Penelitian yang dibuat adalah sistem rekrutmen pada perusahaan PT. Dynapack Asia untuk meningkatkan efisiensi dalam rekrutmen dan pencatatan kebutuhan tenaga kerja khususnya pada divisi <i>Human Resource</i> . Metode pengembangan sistem yang dipakai adalah <i>Rapid Application</i>	Yang diambil dari jurnal ini adalah penggunaan pemrograman berorientasi object dengan menggunakan bahasa pemrograman <i>PHP</i> .

Nama Peneliti	Nama Jurnal	Judul Penelitian	Hasil Penelitian	Hasil Yang Diambil
			<i>Development</i> dan menggunakan metode pengumpulan data melalui observasi serta wawancara di Dynapack Asia.	
Evan Rosiska	JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi) Vol. 2 No.2 (2018) 479-485	Penerapan Metode Analytical Hierarchy Process (AHP) dalam Menentukan Mitra Usaha Berprestasi	Metode penelitian yang digunakan oleh pembuat jurnal ini adalah metode penelitian Kualitatif. Pembuatan penelitian ini menggunakan matriks manual <i>Microsoft Excel</i> dan diuji kedalam sebuah <i>software</i> Super Decisions	Materi yang dapat diambil pada jurnal ini adalah penggunaan metode AHP yang banyak digunakan dalam kasus pembobotan kriteria dan penentuan prioritas setiap kriteria berdasarkan matriks perbandingan berpasangan.
Jijon Raphita Sagala	Jurnal Mantik Penusa : Vol. 2, No.1 , Page 87-90	Model <i>Rapid Application Development</i> (RAD) Dalam Pengembangan Sistem Informasi Penjadwalan Belajar Mengajar (2018)	Metode penelitian yang digunakan oleh Jijon adalah dengan menggunakan metode pengembangan sistem RAD. Didalam jurnal ini juga	Pada jurnal ini, yang di ambil adalah keuntungan dalam menggunakan metode <i>Rapid Application Development</i> (RAD).

Nama Peneliti	Nama Jurnal	Judul Penelitian	Hasil Penelitian	Hasil Yang Diambil
			menjelaskan bagaimana penggunaan dari RAD itu sendiri, serta terdapat kelebihan dan kekurangan dalam menggunakan metode RAD.	

Berdasarkan penelitian terdahulu pada tabel 2.1, metode *Rapid Application Development* (RAD) merupakan metode yang sangat berguna pada kondisi dimana user kurang memahami kebutuhan-kebutuhan apa saja yang digunakan pada proses pengembangan aplikasi. Selain itu, RAD juga mempunyai kemampuan untuk menggunakan komponen yang sudah ada dan dapat membuat waktu pengerjaan menjadi lebih singkat. Dalam mengembangkan sistem aplikasi rekomendasi tempat wisata ini, *tools* yang dapat digunakan adalah dengan menggunakan bahasa pemrograman *PHP* dan juga menggunakan basis data *MySQL* untuk digunakan sebagai media penyimpanan *database* yang berguna untuk mendukung perancangan dan pengembangan sistem aplikasi rekomendasi tempat wisata ini.

2.13 Rumus *Analytical Hierarchy Process* (AHP)

Berikut dibawah ini merupakan rumus dari *Analytical Hierarchy Process* (AHP) :

$$CI = \frac{(\lambda_{maks} - n)}{n - 1}$$

$$CR = \frac{CI}{RI}$$

Rumus 2.1 *Analytical Hierarchy Process*

Sumber: (Kevin; Mulyawan, 2019)

Keterangan :

CI = *Consistency Index*

N = banyak kriteria

CR = *Consistency Ratio*

RI = *Random Index*

Jika $CR \leq 0.1$ maka nilai perbandingan sudah konsisten. Namun jika nilai CR lebih dari 0.1 maka nilai perbandingan belum konsisten.