

BAB III

TINJAUAN PUSTAKA

3.1 Landasan Teori

3.1.1 Sistem Informasi

Menurut Irpan dan Asri (2017:259), Sistem informasi adalah kumpulan dari sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna.

Menurut Abdul (2014:9), Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.

Sistem informasi mengandung komponen-komponen seperti berikut

:

1. Perangkat keras (*Hardware*), yang mencakup peranti-peranti fisik seperti komputer dan printer.
2. Perangkat lunak (*Software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan membangkitkan keluaran yang dikehendaki.

4. Orang, yakni semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
5. Basis data (*Database*), yaitu kumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

Ada berbagai cara untuk mengelompokkan sistem informasi.

Klasifikasi yang umum dipakai antara lain didasarkan pada :

1. Level organisasi

Diantaranya ada jenis pribadi, kelompok kerja, dan perusahaan.

- 1) Area fungsional

Diantaranya adalah penjualan dan pemasaran, manufaktur (Produksi), keuangan, dan Akuntansi

- 2) Dukungan yang diberikan, dan

Diantaranya adalah Sistem pemrosesan transaksi (*transaction processing system* atau TPS), Sistem informasi manajemen (*management information system* atau MIS), Sistem otomasi perkantoran (*office automation system* atau OAS), Sistem pendukung keputusan (*decision support system* atau DSS), Sistem informasi eksekutif (*executive information system* atau EIS), Sistem pendukung kelompok (*group support system* atau

GSS), dan Sistem pendukung cerdas (*intelligent support system* atau ISS).

3) Arsitektur sistem informasi

Diantaranya adalah Sistem berbasis *mainframe*, Sistem komputer pribadi (PC) tunggal, dan Sistem tersebar atau sistem komputasi jaringan.

3.1.2 *Marketplace*

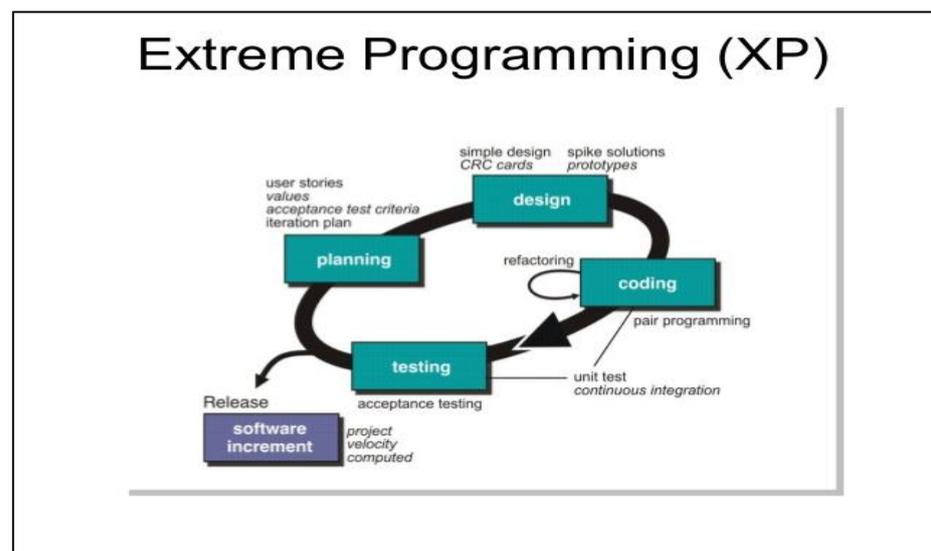
Menurut Betti (2017:243), *Marketplace* merupakan sebuah pasar virtual dimana pasar tersebut menjadi tempat bertemunya pembeli dan penjual untuk melakukan transaksi. *Marketplace* mempunyai fungsi yang sama dengan sebuah pasar tradisional, perbedaannya adalah *marketplace* lebih terkomputerisasi dengan menggunakan bantuan sebuah jaringan dalam mendukung sebuah pasar agar dapat dilakukan secara efisien dalam menyediakan update informasi dan layanan jasa untuk penjual dan pembeli yang berbeda-beda.

3.1.3 *Metode Extreme Programming(XP)*

Menurut Ahmad Fatoni, Dhany Dwi (2016:19), *Metode Extreme Programming* sering juga dikenal dengan metode XP. Metode ini dicetuskan oleh *Kent Beck*, seorang pakar *software engineering*. *Extreme programming* adalah model pengembangan perangkat lunak yang menyederhanakan berbagai tahapan pengembangan sistem menjadi lebih efisien, adaptif dan fleksibel. Nilai dasar metode *extreme programming* :

1. *Communication* : Memfokuskan komunikasi yang baik antara *programmer* dengan *user* maupun antar *programmer*.
2. *Courage* : Pengembang perangkat lunak harus selalu memiliki keyakinan, keberanian dan integritas dalam melakukan tugasnya.
3. *Simplicity* : Lakukan semua dengan sederhana.
4. *Feedback* : Mengandalkan *feedback* sehingga dibutuhkan anggota tim yang berkualitas.
5. *Quality Work* : Proses berkualitas berimplikasi pada perangkat lunak yang berkualitas sebagai hasil akhirnya.

Menurut Andrianto, dkk (2015:269), *Extreme Programming (XP)* menggunakan pendekatan *object oriented* dan mencakup seperangkat aturan yang terjadi dalam 4 kerangka kegiatan : *planning*, *design*, *coding*, and *testing*.



Gambar 3.1 Kerangka kerja *Extreme Programming*

Adapun tahapan dari metode *Extreme Programming* yaitu sebagai berikut :

1. *Planning*

Planning dimulai dengan mengumpulkan kebutuhan yang memungkinkan para anggota dari tim XP dapat memahami konteks bisnis dari perangkat lunak yang akan dibuat dan mendapat wawasan yang luas untuk *output* apa yang diperlukan serta fitur-fitur utama dari perangkat lunak. Tahap ini akan mengarah pada pembuatan “*Stories*” yang menggambarkan *output* yang dibutuhkan, fitur, dan fungsi dari perangkat lunak yang akan dibuat.

2. *Design*

Metode XP mengikuti aturan KIS (*Keep It Simple*). Desain yang sederhana selalu diutamakan dibandingkan dengan representasi yang kompleks. Jika terdapat design yang sulit, XP akan menerapkan *Spike Solution*, dimana pembuatan design dilakukan segera, dan dibuat langsung ketujuannya. XP juga mendukung adanya *refactoring* dimana kita dapat melakukan perubahan pada kode program untuk disederhanakan tanpa merubah cara kode tersebut bekerja.

3. *Coding*

Setelah “*Stories*” sudah dirancang dan desain awal sudah selesai, tim tidak langsung memulai *code*, tetapi terlebih dahulu merancang beberapa *unittest* yang digunakan untuk menjalankan “*stories*” dan

disertakan pada *software release* saat itu. Setelah itu, pengembang *focus* untuk mengimplementasikannya. XP juga menerapkan *Pair Programming* yaitu proses pengembangan program yang dilakukan secara berpasangan. Dua orang bekerja sama dalam satu komputer untuk menulis kode. Hal ini memberikan *real-time problem solving* dan *real-time quality assurance*.

4. *Testing*

Pada tahap *testing* dilakukan pengujian kode pada *unittest* yang telah dibuat sebelumnya. Pada metode XP dilakukan *acceptance test* atau biasa disebut *customer test*. Tes ini diberikan kepada *customer* akan menggunakan fitur dan fungsi sistem yang akan dibuat.

3.1.4 UML (*Unified Modeling Language*)

Menurut Rosa (2014:137), Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

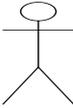
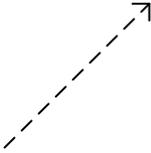
UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Macam – Macam diagram UML :

1) Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case diagram* merupakan sebuah pekerjaan tertentu. Berikut beberapa notasi yang digunakan dalam *use case diagram* diperhatikan pada tabel dibawah ini.

Tabel 3.1 Notasi *Use Case Diagram*

Simbol	Keterangan
	<p>Actor Menggambarkan pengguna perangkat keras, atau sistem informasi lain yang berinteraksi langsung dengan aplikasi perangkat lunak.</p>
	<p>Use Case Memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem.</p>
	<p>Dependency Suatu hubungan semantic antara dua <i>things</i> dimana perubahan pada satu <i>thing</i> (<i>independent</i>) mungkin mempengaruhi semantic <i>thing</i> (<i>independent</i>) lain.</p>
	<p>Association Hubungan structural yang menggambarkan mata rantai antar objek</p>

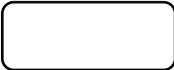
	<p>Generalitation Hubungan spesialisasi dimana objek dari elemen khusus (anak) merupakan pengganti untuk objek elemen umum (induk)</p>
	<p>Realization Hubungan antara antarmuka yang tersedia secara umum (<i>interface</i> atau <i>use case</i>) dengan penerapan detail dari antarmuka (<i>class</i>, <i>package</i>, atau <i>realization</i>).</p>

Sumber : Wibowo (2014)

2) Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas kedalam keadaan sesaat (*state*). *Activity diagram* juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Berikut ini merupakan simbol *Activity diagram* diperhatikan pada tabel dibawah ini.

Tabel 3.2 Simbol *Activity Diagram*

Simbol	Keterangan
	<p>Activity Menyatakan sebuah proses yang dilakukan di dalam sistem.</p>

	Start State Menyatakan status awal dari suatu proses diagram activity
	End State Menyatakan suatu akhir dari suatu proses diagram activity
	Transition Bagaimana perpindahan dari activity ke activity lain
	Swimlanes Siapa yang bertanggung jawab terhadap aktivitas
	Note Suatu simbol yang memberikan batasan dan komentar yang dikaitkan pada suatu elemen atau kumpulan elemen.

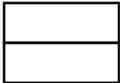
Sumber : Wibowo(2014)

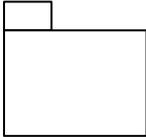
3) Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembang dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu nama

(dan stereotype), atribut, metoda. Berikut ini merupakan simbol *Class diagram* diperhatikan pada tabel dibawah ini.

Tabel 3.3 Simbol *Class Diagram*

Simbol	Keterangan
	<p><i>Class</i> Menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama, digunakan untuk mengimplementasikan elemen-elemen dari sistem yang sedang dibangun</p>
	<p><i>Interface</i> Merupakan operasi tanpa implementasi dari suatu kelas, implementasi operasi dalam interface dijabarkan oleh operasi dalam kelas.</p>
	<p><i>Package</i></p>

	Mekanisme umum mengorganisasikan elemen-elemen ke dalam kelompok-kelompok
---	---

Sumber : Wibowo(2014)

3.2 Penelitian Terdahulu

Berikut adalah hasil penelitian terdahulu yang dapat dilihat pada tabel berikut ini :

Tabel 3.5 Hasil Penelitian Terdahulu

No	Judul	Penulis/Tahun	Hasil
1	Model Sistem Informasi Pariwisata <i>Multiuser</i> Menggunakan Konsep <i>E-MALL</i>	Sri Handayaningsih (2010) ISSN: 1979-2328	Hasil dari penelitian ini adalah sebuah model yang menggambarkan konsep <i>e-mall</i> pada <i>website</i> dinas pariwisata, sebagai upaya dalam memaksimalkan informasi pada satu <i>website</i> .
2	Sistem Informasi Geografis Pariwisata Kota Semarang	Ambrina Kundyani, Ir. Kodrat Iman Satoto, M.T., Dr. Oky Dwi Nurhayati, ST. MT. (2008)	Penelitian ini menghasilkan suatu Sistem informasi geografis pariwisata kota semarang yang bermanfaat untuk mengetahui letak lokasi wisata, penginapan, restoran, tempat ibadah, pusat oleh-oleh dan event yang ada di kota semarang.
3	Perancangan Sistem Informasi Kuliner Di <i>Brother Caffe</i> Berbasis <i>Web</i>	Irfan Ahmad Fauzi, Asri Mulyani (2017) ISSN : 2302-7339	Menghasilkan suatu <i>website</i> yang memudahkan perusahaan dalam mempromosikan apa saja pelayanan yang akan diberikan kepada pelanggan.

4	Rancang Bangun Sistem Informasi Geografis Persebaran Lokasi Obyek Pariwisata Berbasis Web Dan Mobile Android (Studi Kasus Di Dinas Pariwisata Kabupaten Gianyar)	Ida Bagus Made Yogie Adnyana, Rissal Efendi (2014) ISSN : 2087-0868	Sistem informasi geografis ini dapat membantu Dinas Pariwisata Kabupaten Gianyar dalam mengelola obyek-obyek wisata yang ada di Kabupaten Gianyar.
---	--	--	--

1. Model Sistem Informasi Pariwisata *Multiuser* Menggunakan Konsep *E-MALL*

Pada penelitian pertama, hubungan dengan penelitian yang saya lakukan adalah sama-sama mengolah tentang sistem informasi pariwisata.

2. Sistem Informasi Geografis Pariwisata Kota Semarang

Untuk penelitian kedua ini, sama-sama membahas tentang sistem informasi pariwisata, tetapi di penelitian ini lebih khusus ke geografis atau letak dari objek wisata tersebut, dan pada penelitian ini memiliki perbedaan yaitu lokasi penelitian yang dilakukan.

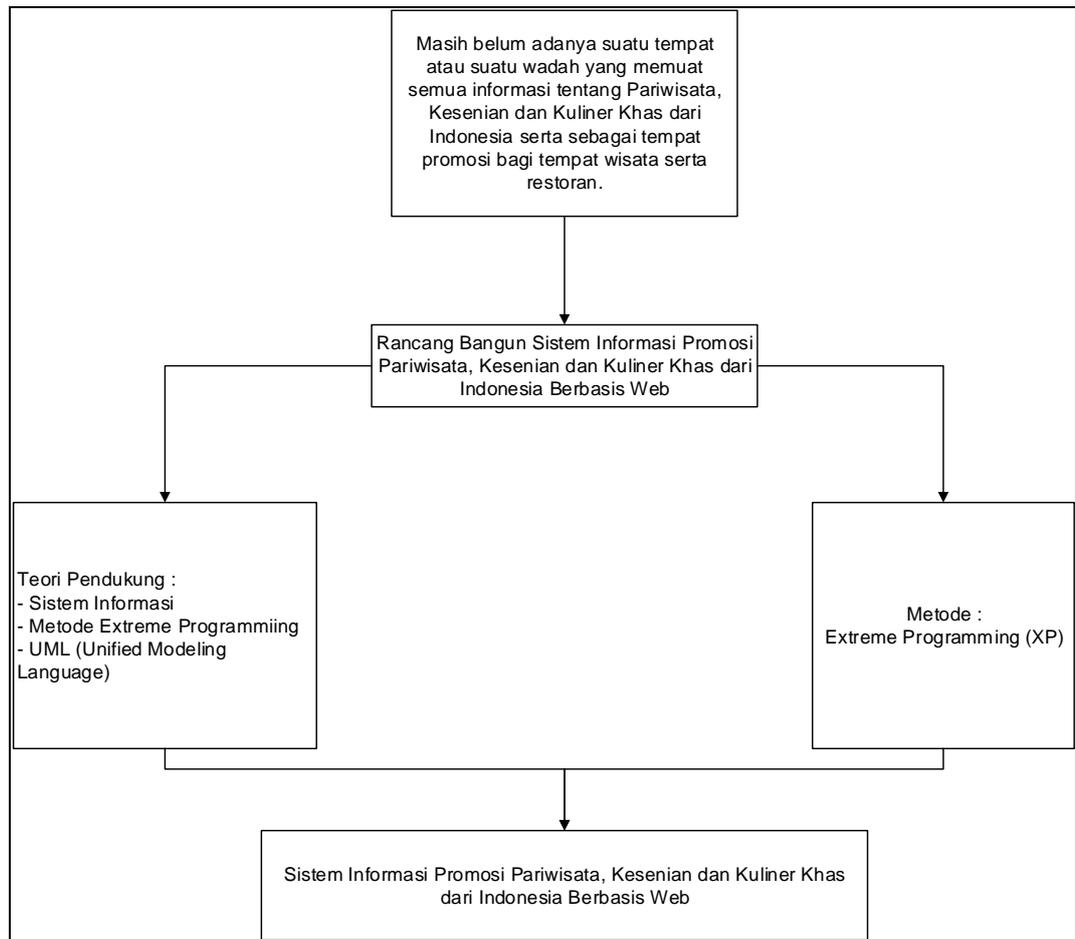
3. Perancangan Sistem Informasi Kuliner Di *Brother Caffe* Berbasis Web

Untuk penelitian yang ketiga, membahas tentang sistem informasi kuliner, yang memiliki kesamaan dengan penelitian yang saya lakukan, untuk lokasi penelitian berbeda dengan lokasi penelitian yang saya lakukan, dan di penelitian ini juga sama berbasis *website*.

4. Rancang Bangun Sistem Informasi Geografis Persebaran Lokasi Obyek Pariwisata Berbasis Web Dan Mobile Android (Studi Kasus Di Dinas Pariwisata Kabupaten Gianyar)

Untuk penelitian yang terakhir ini memiliki kesamaan dengan penelitian yang saya lakukan, yaitu membahas tentang sistem informasi pariwisata, tetapi di penelitian ini lebih terkhusus ke persebaran lokasi nya atau tempat dari obyek wisata tersebut, dan di penelitian ini juga sama-sama berbasis *website*.

3.3 Kerangka Penelitian



Gambar 3.2 Kerangka Pemikiran