

BAB IV

METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian

4.1.1 Lokasi Penelitian

Tempat penelitian ini dilakukan di kota Palembang dengan merangkum serta mendata semua informasi tentang pariwisata, kesenian, dan kuliner khas.

4.1.2 Waktu Penelitian

Adapun waktu pelaksanaan penelitian dilaksanakan dari bulan Maret sampai dengan bulan Juni.

NO	URAIAN	BULAN															
		MARET				APRIL				MEI				JUNI			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	<i>Planning</i>	■	■	■	■												
2	<i>Design</i>					■	■	■	■								
3	<i>Coding</i>									■	■	■	■	■	■		
4	<i>Testing</i>														■	■	■

Gambar 4.1 Jadwal Penelitian

4.2 Jenis Pengumpulan data

4.2.1 Data Primer

Menurut Adi (2015:12), Data primer adalah data yang diperoleh peneliti sendiri dari subjek atau objek yang diteliti melalui pengamatan, wawancara atau eksperimen. Penulis mengumpulkan data tentang pariwisata, kesenian dan kuliner khas

dari bertanya kepada masyarakat. Adapun data yang diperoleh yaitu data nama-nama objek pariwisata, kesenian dan kuliner khas.

4.2.2 Data Sekunder

Menurut Sugiarto (2015:89), Data sekunder adalah data yang diperoleh dengan cara mengkaji berbagai literature dan hasil penelitian yang terkait. Data sekunder digunakan sebagai data pendukung yang memperkuat data primer yang diperoleh dari hasil observasi dan wawancara. Data sekunder umumnya berupa bukti, atau laporan historis yang tersusun dalam arsip atau data dokumenter yang dipublikasikan dan yang tidak dipublikasikan.

Jenis data yang digunakan adalah data sekunder dimana data ini dapat dari hasil penelitian sebelumnya kemudian akan dipakai dan dikembangkan kedalam aplikasi yang akan dibuat.

4.3 Teknik Pengumpulan Data

4.3.1 Pengamatan (Observasi)

Menurut Nazir (2014:154), Pengumpulan data dengan observasi langsung atau dengan pengamatan langsung adalah cara pengambilan data dengan menggunakan mata tanpa ada pertolongan alat standar lain untuk keperluan tersebut.

Dalam hal ini penulis melakukan observasi pada pariwisata, kesenian, dan kuliner khas dengan datang langsung ke lokasi.

4.3.2 Wawancara

Menurut Nazir (2014:170), Wawancara adalah proses memperoleh keterangan untuk tujuan penelitian dengan cara tanya jawab, sambil bertatap muka antara si penanya atau pewawancara dengan si penjawab atau responden dengan menggunakan alat yang dinamakan *interview guide* (panduan wawancara).

Dalam metode ini penulis melakukan wawancara dengan penjaga tempat objek wisata tentang harga dan jam operasional, dengan karyawan restoran tentang harga dari makanan atau minuman yang dijual.

4.3.3 Dokumentasi

Menurut Sugiarto dan Afifuddin (2015:88), Dokumentasi adalah teknik pengumpulan data dengan cara mencari bukti-bukti dari sumber nonmanusia terkait dengan objek yang diteliti.

Menurut Sugiyono dan Sugiarto (2015:88), Yaitu yang berupa tulisan, gambar, atau karya-karya monumental dari seseorang.

Dalam hal ini penulis mendapatkan data objek pariwisata, kesenian, dan kuliner khas dari website pemerintah kota Palembang, serta mendokumentasikan dalam bentuk foto dan video.

4.3.4 Studi Pustaka

Menurut Nazir (2014:79), Studi kepustakaan adalah teknik pengumpulan data atau analisis data yang pernah digunakan oleh peneliti-peneliti terdahulu.

Dalam hal ini penulis mengumpulkan jurnal yang berkaitan dengan judul penelitian yang penulis lakukan.

4.4 Alat Pengembangan Sistem

4.4.1 Permodelan Proses

4.4.1.1 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

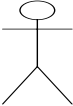
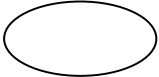
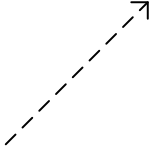


tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

4.4.2 Permodelan Data

4.4.2.1 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 4.1 Notasi *Use Case Diagram*

Simbol	Keterangan
	Actor Menggambarkan pengguna perangkat keras, atau sistem informasi lain yang berinteraksi langsung dengan aplikasi perangkat lunak.
	Use Case Memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem.
	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.
	Association Hubungan structural yang menggambarkan mata rantai antar objek
	Generalization Hubungan spesialisasi dimana objek







	dari elemen khusus (anak) merupakan pengganti untuk objek elemen umum (induk)
----->	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>).

Sumber : Wibowo (2014)

4.4.2.2 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 ke dalam 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 4.2 Simbol *Activity Diagram*

Simbol	Keterangan
	Activity Menyatakan sebuah proses yang dilakukan di dalam sistem.
	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.

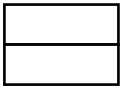

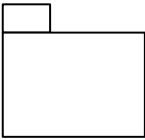
Sumber : Wibowo(2014)

4.4.2.3 *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 4.3 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> Mekanisme umum mengorganisasikan elemen-elemen ke dalam kelompok-kelompok</p>

4.4.3 Metode *Extreme Programming* (XP)

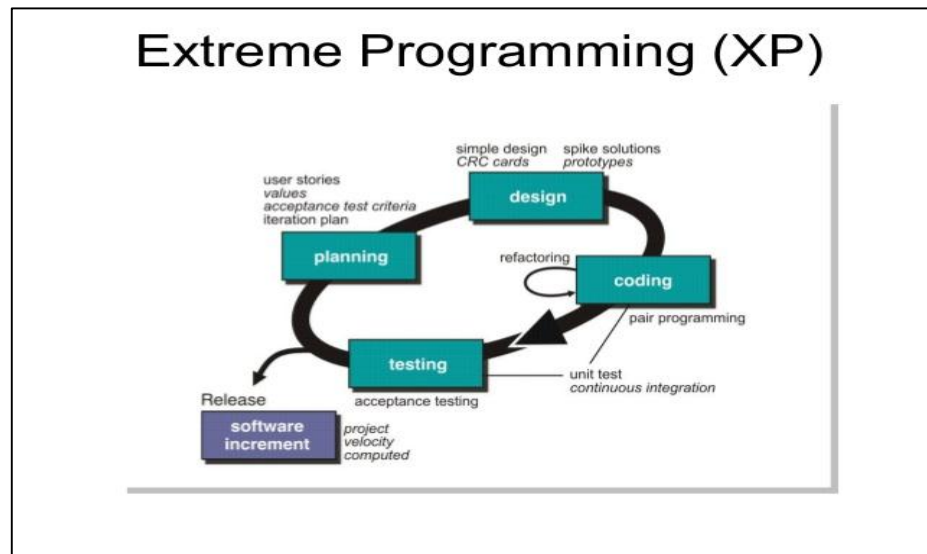
Menurut Ahmad Fatoni, Dhany Dwi (2016:19), Metode *Extreme Programming* sering juga dikenal dengan metode XP.

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 4.2 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 Smile*). 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.

4.4.4 Metode Pengujian Sistem

1. *Black Box*

Menurut Rosa (2014:275), Yaitu menguji perangkat lunak segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.