

BAB IV

METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian

4.1.1 Lokasi Penelitian

Untuk mendapatkan data-data yang dibutuhkan, maka penulis melakukan riset ke perusahaan-perusahaan advertising kota Palembang antara lain perusahaan CV. Citra Sriwijaya Advertising, Studio Kreasindo, Lisatiga Printing, Gracia Jaya Advertising, Grafindo Digital Printing, Anggrek Advertising, Pelangi Advertising dan Imagica Imaging Digital Printing.

4.1.2 Waktu Penelitian

Waktu penelitian dilakukan dari tanggal 2 April 2018 sampai tanggal 31 Juli 2018 pada hari kerja senin-sabtu dari jam 09:00 -16.30 WIB.

4.2 Jenis Data

Dalam penulisan laporan ini, penulis menggunakan beberapa jenis data dalam pengumpulan datanya, yang terdiri dari :

4.2.1 Data Primer

Disini penulis mendapatkan data primer dengan cara melakukan wawancara langsung dengan Bapak Erik Stefano selaku pemilik perusahaan CV. Citra Sriwijaya Advertising dan Studio Kreasindo, Bapak Herman Hasim selaku pemilik Grafindo Digital Printing,

Bapak Beni Admiral selaku pemilik Lisatiga Printing, Bapak Rudi selaku pemilik Anggrek Advertising, Bapak Salim Robertus selaku pemilik Pelangi Advertising, Bapak Anton selaku pemilik Gracia Jaya Advertising, Bapak Fransen selaku pemilik Imagica Imaging Digital Printing dimana wawancara ini dilakukan untuk mendapatkan informasi tentang produk percetakan beserta harga pada perusahaan. Contoh data primer yang didapat dalam penelitian ini berupa hasil wawancara dan bahan materi untuk pembuatan aplikasi *e-marketplace* produk perusahaan advertising kota Palembang berbasis web.

4.2.2 Data Sekunder

Penulis disini mendapatkan data sekunder dengan cara membaca jurnal-jurnal yang telah ada. Contoh data sekunder dapat kita ambil referensi untuk penulis yaitu data dari berbagai buku, dokumen, internet, media cetak, dan dari jurnal penelitian terdahulu. Dan penulis mendapatkan data sekunder ini adalah beberapa judul yang dijadikan referensi dan kiat-kiat untuk menyelesaikan penelitian penulis.

4.3 Teknik Pengumpulan Data

Dalam penulisan laporan ini penulis menggunakan beberapa metode teknik pengumpulan data, yang terdiri dari :

4.3.1 Observasi

Teknik observasi dilakukan bersama-sama dengan pengumpulan kebutuhan sistem yang lain. Penulis mengamati

perusahaan *advertising*/percetakan diantaranya yaitu Citra Sriwijaya Advertising, Studio Kreasindo, Lisatiga Printing, Gracia Jaya Advertising, Grafindo Digital Printing, Anggrek Advertising, Pelangi Advertising dan Imagica Imaging Digital Printing, adapun yang penulis amati adalah sebagai berikut :

1. Informasi tentang perusahaan
2. Cara pemesanan produk pada perusahaan

4.3.2 Wawancara (*Interview*)

Penulis melakukan wawancara langsung dengan Bapak Erik Stefano selaku pemilik perusahaan CV. Citra Sriwijaya Advertising dan Studio Kreasindo, Bapak Herman Hasim selaku pemilik Grafindo Digital Printing, Bapak Beni Admiral selaku pemilik Lisatiga Printing, Bapak Rudi selaku pemilik Anggrek Advertising, Bapak Salim Robertus selaku pemilik Pelangi Advertising, Bapak Anton selaku pemilik Gracia Jaya Advertising, Bapak Fransen selaku pemilik Imagica Imaging Digital Printing, wawancara tersebut dilakukan pada tanggal 22 Mei 2018 dan pada tanggal 25 Mei 2018 yang bertempat di masing-masing perusahaan percetakan, dengan tujuan untuk mendapatkan informasi tentang produk, harga produk dan informasi perusahaan.

4.3.3 Studi Pustaka

Penulis melakukan studi pustaka terhadap penelitian-penelitian terdahulu, yang digunakan sebagai data sekunder atau teori

pendukung yang penulis butuhkan dalam penelitian ini, meliputi teori aplikasi, *e-marketplace* dan metode *Rapid Application Development (RAD)*.

4.4 Alat dan Pengembangan Sistem

4.4.1 Alat Pengembangan Sistem

4.4.1.1 UML (*Unified Modelling Language*)

Alat pengembangan sistem yang digunakan adalah *Unified Modelling Language (UML)*. Menurut Rosa dan Shalahuddin (2014:137), *Unified Modelling Language (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 digunakan pada metodologi berorientasi objek.

a. *Use Case Diagram*

Menurut Rosa dan Shalahuddin (2014:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi

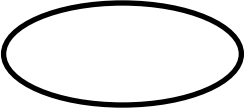
apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

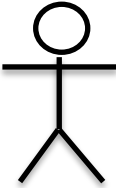

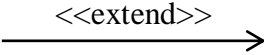
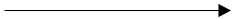
Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua ahal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

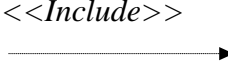
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor.

symbol-simbol *use case* menurut (Rosa, 2014:156), dapat dilihat pada tabel 6.1.

Tabel 6.1 Simbol – Simbol Use Case

Nama Komponen	Simbol	Keterangan
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>

Nama Komponen	Simbol	Keterangan
<i>Actor</i>		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 frase nama aktor.
<i>Association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi pada aktor.
Ektensi / <i>extend</i>		Relasi <i>use case</i> tambahan sebuah <i>use case</i> diaman <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama dapan yang sama dengan <i>use case</i> yang tambahkan.
Generalisasi / <i>generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.

Nama Komponen	Simbol	Keterangan
<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> ini.

Sumber: Rosa dan Shalahuddin (2014:156)

b. Class Diagram

Menurut Rosa dan Shalahuddin (2014:141), Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut ini :

1. Kelas *main*

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem.

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendahuluan *use case*.



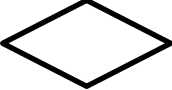



Kelas yang menangani fungsi-fungsi yang harus ada di ambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data.

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Simbol *Class Diagram* dapat dilihat pada tabel 6.2.

Tabel 6.2. *Class Diagram*

Nama Komponen	Simbol	Keterangan
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / decision		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / join		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa dan Shalahuddin (2014:162)

4.4.1.2 State Transition Diagram (STD)

State transition diagram (STD) merupakan suatu *modelling tool* yang menggambarkan sifat ketergantungan terhadap

waktu pada sistem. STD digunakan untuk mengidentifikasi sebagaimana sistem harus berperilaku seperti resiko dari kejadian eksternal.

4.4.2 Teknik Pengembangan Sistem

Teknik yang digunakan untuk mengembangkan sistem adalah *Rapid Application Development* (RAD). Menurut Wahyuningrum, Januarita (2014:81), *Rapid Application Development* (RAD) merupakan metode pengembangan sistem informasi dengan waktu singkat, sehingga dinilai tepat digunakan dalam pembangunan *e-commerce*. RAD menggunakan metode *iteratif* (berulang) dalam mengembangkan sistem dimana *working model* (model bekerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna. Dalam pengembangan sistem informasi normal, memerlukan waktu minimal 180 hari, namun dengan menggunakan metode RAD, sistem dapat diselesaikan dalam waktu 30-90 hari. Metode RAD memiliki 3 tahapan yaitu :

1. Rencana Kebutuhan (*Requirement Planning*) : *User* dan *analyst* melakukan pertemuan untuk mengidentifikasi tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Pada tahap ini merupakan hal terpenting yaitu adanya keterlibatan dari kedua belah pihak.

2. Proses Desain Sistem (*Design System*) : Pada tahap ini keaktifan *user* yang terlibat menentukan untuk mencapai tujuan karena pada proses ini melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Seorang *user* dapat langsung memberikan komentar apabila terdapat ketidaksesuaian pada desain, merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahapan ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain.
3. Implementasi (*Implementation*) : Tahapan ini adalah tahapan *programmer* yang mengembangkan desain suatu program yang telah disetujui oleh *user* dan *analyst*. Sebelum diaplikasikan pada suatu organisasi terlebih dahulu dilakukan proses pengujian terhadap program tersebut apakah ada kesalahan atau tidak. Pada tahap ini *user* biasa memberikan tanggapan akan sistem yang sudah dibuat serta mendapat persetujuan mengenai sistem tersebut.

4.5 Alat dan Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak sebagai suatu elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Adapun teknik yang digunakan oleh penulis adalah pengujian *Black-Box*.

Pengujian *Black-Box* berfokus pada penyerahan fungsional perangkat lunak dengan demikian pengujian *Black-Box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya dengan menggunakan semua persyaratan fungsional untuk satu program. Pengujian *Black-Box* merupakan alternatif dari teknik *White Box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkapkan kelas kesalahan dari pada metode *White Box*.

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan *interface*.
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan lahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.

Pengujian *Black-Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black-Box* merupakan metode perancangan data uji yang

didasarkan pada spesifikasi perangkat lunak. Data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan.