BAB IV

METODE PENELITIAN

4.1 Lokasi dan Waktu Penelitian

4.1.1 Lokasi Penelitian

Penelitian dilakukan di PT. Teknologi Syariah Indonesia yang beralamat di Jl. Sukarela No. 04 KM 7 Palembang, Sumatera Selatan.

4.1.2 Waktu Penelitian

Waktu penelitian dilaksanakan pada tanggal 22 April 2018 sampai tanggal 6 Juni 2018.

Tabel 4.1 Jadwal Penelitian

Tahapan	Tahun 2018															
	Bulan Maret				Bulan April				Bulan Mei				Bulan Juni			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Fase *Inception*																
Fase *Elaboration*																
Fase *Contruction*																
Fase *Transition*																

4.2 Jenis Data

4.2.1 Data Primer

Sunyoto (2013:21), data primer adalah data asli yang dikumpulkan sendiri oleh peneliti untuk menjawab masalah penelitiannya secara khusus. Pada umum data primer ini sebelumnya belum tersedia, sehingga seorang peneliti harus melakukan

pengumpulan sendiri data ini berdasarkan kebutuhannya. Data primer didapat langsung oleh penulis dari direktur PT Teknologi Syariah Indonesia. Contoh data primer yang penulis ambil dari perusahaan lokasi riset adalah hasil wawancara dengan direktur perusahaan mengenai proses jalannya sistem informasi pengolahan re stok barang PT Teknologi Syariah Indonesia.

4.2.2 Data Sekunder

Sunyoto (2013:21), data sekunder adalah data yang bersumber dari catatan-catatan yang ada pada perusahaan dan dari sumber lainnya yaitu dengan mengadakan studi kepustakaan dengan mempelajari buku-buku yang ada hubungannya dengan objek penelitian. Data itu biasanya diperoleh dari perusahaan, laporan-laporan peneliti terdahulu, jurnal dan buku-buku sebagai referensi data pendukung didalam penelitian yang penulis lakukan. Data sekunder juga biasa disebut data tersedia. Data sekunder yang diperoleh dari PT Teknologi Syariah Indonesia meliputi sejarah singkat perusahaan, struktur organisasi, visi dan misi, tugas dan wewenang, data barang, data penjualan, data pembelian, data cabang, data *supplier*.

4.3 Teknik Pengumpulan Data

Dalam melakukan penyusunan laporan skripsi ini penulis menggunakan beberapa teknik pengumpulan data. Data yang digunakan dalam laporan adalah :

4.3.1 Observasi (Pengamatan)

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

Penulis melakukan pengamatan bahwa di PT. Teknologi Syariah Indonesia dalam pengolahan data masih mencatat manual pada buku dan *form,* ini sering terjadi kesalahan pencatatan serta sebagian ada menggunakan *spreadsheet* yang belum *user friendly.*

4.3.2 *Interview* (Wawancara)

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

Penulis menyimpulkan bahwa wawancara merupakan tanya jawab secara langsung antarasi penanya dengan si penjawab, penulis melakukan wawancara langsung kepada ibu Wulan yang bertugas sebagai staff kantor. Penulis mendapatkan informasi seringnya terjadi kesalahan dalam pencatatan manual, barang yang

terjual tidak sama dengan stok barang yang berkurang yang dicatat dibuku besar.

4.3.3 Studi Pustaka

Menurut Nazir (2014:79), studi kepustakaan merupakan langkah yang penting dimana setelah seorang peneliti menetapkan topik penelitian, langkah selanjutnya adalah melakukan pengkajian yang berkaitan dengan teori pada topik penelitian.

Penulis melakukan pengumpulan studi pustaka dengan cara membaca buku pada perpustakaan STMIK PalComTech Palembang serta sumber lain seperti buku, jurnal dan situs internet yang berhubungan dengan penelitian.

4.3.4 Dokumentasi

Menurut Nurlina (2012:90), cara memperoleh data mengenai hal-hal atau variable-variable yang berupa catatan, transkip, buku, surat kabar, majalah, prasasti, notulen, rapan, agenda, dan sebagainya.

4.4 Jenis Penelitian

Menurut Wandasari (2013:561), Data kualitatif adalah data yang berbentuk kata-kata, bukan dalam bentuk angka. Data kualitatif diperoleh melalui berbagai macam teknik pengumpulan data misalnya wawancara, analisis dokumen, diskusi terfokus, atau observasi yang telah dituangkan dalam catatan lapangan (transkip). Bentuk lain data kualitatif adalah gambar yang diperoleh melalui pemotretan atau rekaman video.

4.5 Alat dan Teknik Pengembangan Sistem

4.5.1 Alat Pengembangan Sistem

Adapun alat pengembangan sistem yang dilakukan dalam penelitian menggunakan bahasa pemodelan UML (*Unified Modelling Language*) terdiri dari *use case diagram, activity diagram* dan *class diagram.*

Menurut Hadi, Arlis, Hariyanto (2017:142) *Unifed Modeling Language (UML)* adalah bahasa standar untuk penulisan cetak biru perangkat lunak. UML daat digunakan untuk memvisualisasikan, menentukan, mengonstruksi, dan mendokumentasikan artifak-artifak suatu sistem *software-intensive.* Dengan kata lain, sama seperti arsitek bangunan membuat cetak biru untuk digunakan oleh perusahaan kontruksi, arsitek perangkat lunak dalam membangun perangkat lunak.

Menurut Salisah (2016:40) UML (*Unified Modelling Language*) adalah bahasa standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML merupakan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML lebih cocok untuk

penulisan piranti dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.

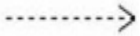
A. Pemodelan Proses

1) *Use Case Diagram*

Menurut Salisah (2016:40) *Use Case Diagram* bersifat statis. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna, simbol-simbol *Use Case Diagram* dapat dilihat pada tabel 4.2.

Tabel 4.2. Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan peran yang pengguna mainkan ketika berinteraksi dengan Use Case.

No	Gambar	Nama	Keterangan
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu element mandiri (*indepedent*) akan mempengaruhi element yang bergantung pada element yang tidak mandiri.
3		*Generaliza-tion*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *UseCase* sumber secara *eksplisit.*
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber

No	Gambar	Nama	Keterangan
			pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*UseCase*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

No	Gambar	Nama	Keterangan
9		*Collabora-tion*	Interaksi aturan-aturan dan element lain yang berkerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan element-elementnya (sinergi).
10		*Note*	Element fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber: Salisah (2016)

2) *Activity Diagram*

Menurut Salisah (2016:40) Diagram ini bersifat dinamis. *Activity Diagram* menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang terjadi pada beberapa eksekusi. Simbol-simbol *Activity diagram* dapat dilihat pada tabel 4.3.

Tabel 4.3. Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		*Activy*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari satu aksi
3		*Initial Node*	Bagaiman objek dibentuk dan diawali
4		*Actifity Final Node*	Bagaiman objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : Salisah (2016)

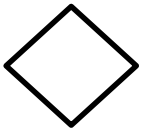
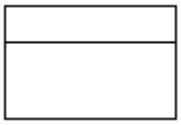
B. Pemodelan Data

Pemodelan data yang digunakan yaitu *class diagram.* Menurut Salisah (2016:40) *Class Diagram* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan

merupakan inti dari pengembangan dan desain berorientasi objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menunjukan interaksi antara kelas dalam sistem. Kelas mengandung informasi dan tingkah laku (*behavior*) yang berkaitan dengan informasi tersebut. Sebuah kelas pada diagram kelas dibuat untuk setiap tipe objek pada diagram sekuensial atau diagram kolaborasi. *Case tool* tertentu seperti *rational rose* membangkitkan struktur kode sumber untuk kelas-kelas kemudian para programmer menyempurnakan dengan bahasa pemrograman yang dipilih pada saat *coding*. Para analist menggunakan diagram ini untuk menunjukan detail sistem, sedangkan arsitek sistem menggunakan diagram ini untuk melihat rancangan sistem. Simbol-simbol *Class Diagram* dapat dilihat pada tabel 4.4.

Tabel 4.4. Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur

No	Gambar	Nama	Keterangan
			data dari objek yang ada di atasnya objek induk (*ancestor)*
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek

No	Gambar	Nama	Keterangan
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu element yang mandiri (*independent)* akan mempengaruhi element yang tidak mandiri
7		*Association*	Apa yang menghubungkan antar objek satu dengan objek lainnya

Sumber : Salisah (2016)

4.5.2 Teknik Pengembangan Sistem

Menurut Rosa dan Shalahudin (2016:125*) Rational Unified Process* (RUP) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), focus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan perstrukturan yang baik (*well structured*).

RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh *Rational Software* yang diakui oleh IBM di Bulan Februari 2003. RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Dalam *Rational Unified Process* terdapat empat tahap pengembangan perangkat lunak yaitu :

a. *Inception* (permulaan)

Pada tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requipments*). Berikut adalah tahap yang dibutuhkan pada tahap ini :

1. Memahami ruang lingkup dari proyek (termasuk pada biaya, waktu, kebutuhan, resiko dan lain sebagainya).
2. Membangun kasus bisnis yang dibutuhkan.

b. *Elaboration* (perluasan/perencanaan)

Pada tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang buat.

c. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Untuk pengujian sistem menggunakan *black box testing.*

Menurut Mustaqbal (2015), *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing. Black Box Testing* cenderung untuk menemukan hal-hal berikut :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan - pertanyaan berikut :

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid ?

2. Input seperti apa yang dapat menjadi bahan kasus uji yang baik ?
3. Apakah sistem sensitif pada input-input tertentu ?
4. Bagaimana sekumpulan data dapat diisolasi ?
5. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem ?
6. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem ?

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan *Black Box Testing*, antara lain:

1. *Equivalence Partitioning*
2. *Boundary Value Analysis/Limit Testing*
3. *Comparison Testing*
4. *Sample Testing*
5. *Robustness Testing*
6. *Behavior Testing*
7. *Requirement Testing*
8. *Performance Testing*
9. Uji Ketahanan (*Endurance Testing*)
10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*).

d. Transition (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan

produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas kemampuan operasional awal.