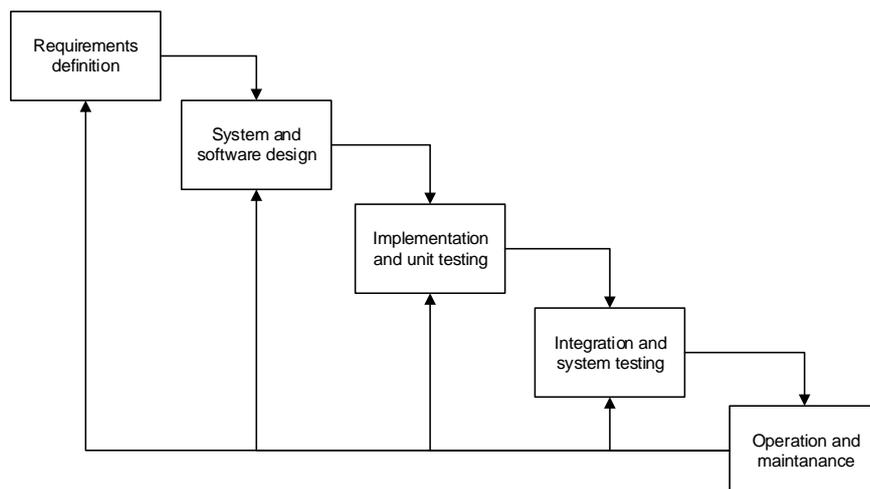


BAB III

METODE PENELITIAN

3.1. Desain Penelitian

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* (Sasmito, 2017 : 9) merupakan model pengembangan sistem informasi yang *systematic* dan sekuensial. Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut:



Gambar 3. 1 Metode Waterfall

1. *Requirements analysis and definition*

Pelayanan dalam sistem serta kendala kendala yang ada pada sistem dan tujuan yang akan di terapkan setelah hasil dari sebuah perundingan dengan user

yang diteruskan akan mendapatkan sebuah pendefinisian secara rinci dan bermanfaat dalam mengelompokkan *system*.

2. *System and software design*

Dalam melakukan proses perancangan sistem untuk mendistribusikan sebuah keperluan-keperluan yang akan dibutuhkan oleh *hardware* ataupun *software* untuk dapat membuat suatu kerangka sistem secara keseluruhannya. Perancangan *software* yang di libatkan dalam melakukan pengidentifikasian serta dalam membuat gambar yang abstraksi sistem dasar dari software dan hubungannya.

3. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer .

5. *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance*

melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

3.1.1. SDLC (System Development Life Cycle)

Adapun tahapan dalam SDLC (*System Development Life Cycle*) Menurut (Adnyana & Efendi, 2018: 11) sebagai berikut :

1. Tahap Perencanaan Sistem (*System Planning*)

Tahap perencanaan adalah tahap awal pengembangan sistem yang mendefinisikan perkiraan kebutuhan-kebutuhan sumber daya seperti perangkat fisik, manusia, metode (teknik dan operasi), dan anggaran yang sifatnya masih umum (belum detail/rinci).

2. Tahap Analisis Sistem (*System Analysis*)

Tahap analisis sistem adalah tahap penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui.

3. Tahap Perancangan/Desain Sistem (*System Design*)

Tahap desain sistem adalah tahap setelah analisis sistem yang menentukan proses dan data yang diperlukan oleh sistem baru. Desain sistem dibedakan menjadi dua macam, yaitu desain sistem umum dan desain sistem terinci.

4. Tahap Penerapan/Implementasi Sistem (*System Implementation*)

Tahap implementasi atau penerapan adalah tahap dimana desain sistem dibentuk menjadi suatu kode (program) yang siap untuk dioperasikan.

5. Tahap Pemeliharaan/Perawatan Sistem

Tahap pemeliharaan/perawatan sistem merupakan tahap yang dilakukan setelah tahap implementasi yang meliputi penggunaan sistem, audit sistem, penjagaan sistem, perbaikan sistem dan peningkatan sistem.

3.2. Objek Penelitian

3.2.1. Sejarah Singkat Perusahaan

Pelabuhan CPO Kabil Batam yang berkedudukan di pulau Batam. Pelabuhan CPO Kabil ini adalah salah satu pelabuhan yang dimiliki oleh Badan Perusahaan Batam dan satu-satunya pelabuhan cargo curah cair di Batam. Pelabuhan ini juga pernah dijadikan pelabuhan penyebrangan dari Batam ke Pulau Bintan sebelum Pelabuhan Telaga Punggur beroperasi. Pelabuhan ini beralamat di Jl. Raya Pelabuhan, Kabil, Nongsa, Kota Batam, provinsi Kepulauan Riau dengan koordinat 01-07'-00 LU / 104-10'-05" BT. Pelabuhan CPO Kabil adalah pelabuhan yang membantu mengembangkan industri dibidang perminyakan. Pelabuhan ini memiliki ukuran dengan panjang dermaga 410 mtr, lebar dermaga 350 mtr, dan dalam dermaga 9-19 mtr.

3.2.2. Visi dan Misi Perusahaan

Visi :

Menjadikan kawasan Batam sebagai kawasan ekonomi terkemuka Asia

Pasifik dan kontributor utama pembangunan ekonomi nasional.

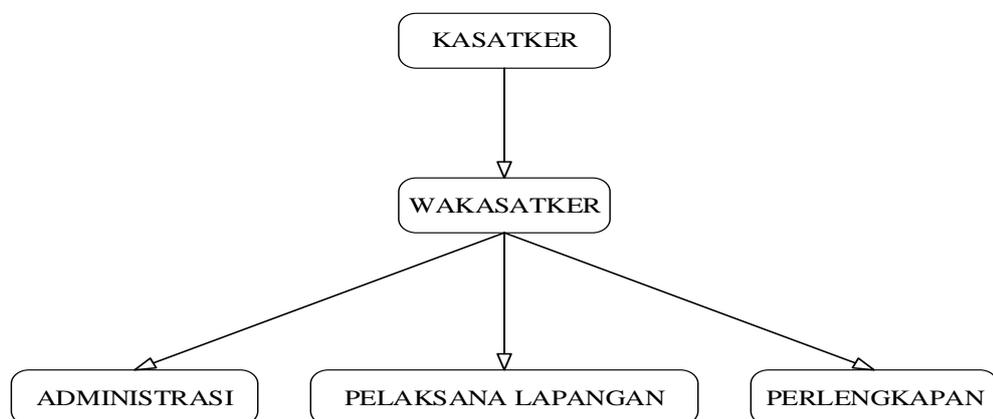
Misi :

1. Mewujudkan pulau Batam sebagai pulau industri hijau berorientasi Ekspor.
2. Mewujudkan pulau Batam menjadi kawasan wisata bahari unggul dan transshipment perdagangan internasional

3.2.3. Struktur Organisasi Pelabuhan

Struktur organisasi adalah suatu susunan komponen-komponen kerja dalam sebuah organisasi perusahaan. Struktur organisasi menunjukkan bahwa adanya pembagian kerja dan bagaimana fungsi atau kegiatan-kegiatan berbeda yang dikoordinasikan.

Struktur organisasi dapat menggambarkan secara jelas pemisahan kegiatan dari pekerjaan antara yang satu dengan kegiatan yang lainnya. Berikut susunan struktur organisasi pada Pelabuhan CPO Kabil :



Gambar 3. 2 Gambar struktur organisasi

Fungsi bagian-bagian dari struktur organisasi Pelabuhan CPO Kabil bagian pendataan dalam aktifitas yang di lakukan meliputi:

1. Kasatker:
 - a. Sebagai pimpinan tertinggi didalam struktur pelabuhan, harus dapat memberikan panutan kepada anggota satuan kerja dan bertanggung jawab penuh terhadap pelabuhan.
 - b. Meminta pertanggung jawaban setiap anggota satuan kerja atas tugas dan kewajiban yang di bebankan kepada masing-masing anggota.
 - c. Menentukan tujuan perusahaan untuk jangka pendek dan jangka panjang.
2. Wakasatker
 - a. Melakukan penyusunan keuangan Pelabuhan.
 - b. Berhubungan dengan pihak internal ataupun eksternal berkaitan dengan pembayaran jasa pelabuhan.
 - c. Melakukan pemeriksaan absensi pegawai.
 - d. Menerima hasil rekap jasa penggunaan dermaga.
 - e. Membuat laporan tentang pendapatan dan pengeluaran pelabuhan.
3. Administrasi
 - a. Melakukan penginputan data kapal yang akan menggunakan pelabuhan.
 - b. Memberikan informasi kepada pelaksana lapangan.
 - c. Menyediakan dan membuat nota pembayaran penggunaan jasa pelabuhan.
 - d. Menyimpan arsip-arsip dengan baik.
4. Pelaksana lapangan
 - a. Mengontrol jalannya aktifitas pelabuhan.
 - b. Memeriksa kelengkapan safety pekerja.

5. Perlengkapan

- a. Memeriksa kekurangan perlengkapan pelabuhan.

3.3. Analisis SWOT Program Yang Sedang Berjalan

Analisis SWOT adalah identifikasi berbagai faktor secara sistematis untuk merumuskan strategi perusahaan, Analisis SWOT meliputi dua tahap analisis lingkungan. Yaitu analisis faktor-faktor internal/ IFAS (*Internal Factor Analysis Summary*) diantaranya kekuatan (*Strength*) dan kelemahan (*weakness*) dan yang kedua analisis faktor-faktor *eksternal*/ EFAS (*Eksternal Faktor Analysis Summary*) yaitu peluang (*Opportunities*) dan ancaman (*Threats*) bagi perusahaan (Yusuf, 2015 : 115)

Evaluasi Sistem yang berjalan pada Pelabuhan CPO Kabil Batam adalah sebagai berikut:

1. Kekuatan Sistem (*Strength*)

- a. Proses pendataan rencana kedatangan kapal lebih cepat.
- b. Biaya yang dikeluarkan relatif lebih murah karena hanya membutuhkan kuitansi.

2. Kelemahan Sistem (*Weakness*)

- a. Data laporan tidak akurat.
- b. Membutuhkan waktu yang lama untuk proses kegiatan.
- c. Penyusunan laporan tidak dapat dilakukan dengan otomatis.

3. Peluang Sistem (*Opportunity*)

- a. Suatu sistem harus mampu melakukan input data, transaksi dan langsung menyimpan data tersebut ke dalam *database*.
- b. Mampu memberikan laporan dan kegiatan bulanan yang dibutuhkan dengan cepat dan akurat.

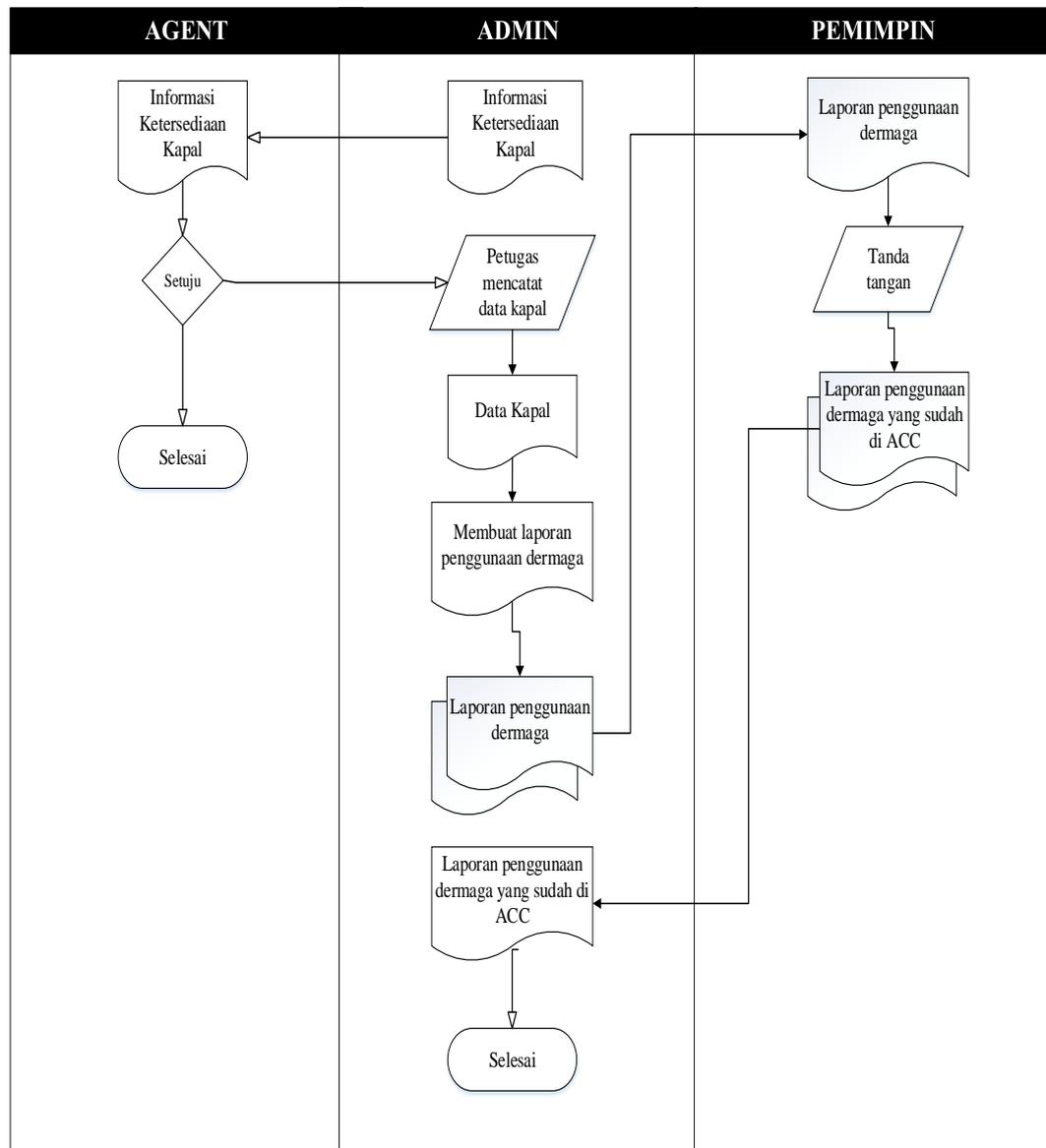
4. Ancaman (*Threat*)

- a. Munculnya sistem baru yang lebih baik dan efisien
- b. Ancaman dari segi keamanan data.
- c. Bencana alam yang mengakibatkan kerusakan fisik.

3.4. Analisis Sistem Yang Sedang Berjalan

Saat ini Pelabuhan CPO Kabil Batam antrian kapal yang tidak teratur. Mengatur waktu proses perizinan kurang efisien dan dokumen kelengkapan kapal. Pendataan kapal masih secara manual yaitu dengan menggunakan media papan dan Untuk penyimpanan data kapal masih tersimpan pada satu buku laporan.

3.5. Aliran Sistem Informasi Yang Sedang Berjalan



Gambar 3. 3 Aliran Sistem Informasi yang sedang berjalan.

Deskripsi dari Aliran Sistem Informasi yang sedang berjalan adalah sebagai berikut:

1. Agent mencari informasi tentang antrian jasa penggunaan pelabuhan.
2. Admin mengecek antrian kapal dan memberi informasi tentang pelabuhan.
3. Agent menerima informasi rencana kedatangan kapal.
4. Admin mencatat data kapal di papan informasi dan membuat laporan.
5. Admin memberikan laporan kepada pimpinan.
6. Pimpinan menerima laporan.

3.6. Permasalahan Yang Sedang Di Hadapi

1. Antrian kapal yang tidak teratur, mengatur waktu proses perizinan kurang efisien dan dokumen kelengkapan kapal.
2. Pendataan kapal masih secara manual yaitu dengan menggunakan media papan.
3. Untuk penyimpanan data kapal masih tersimpan pada satu buku laporan.
4. Memperlambat kinerja petugas dilapangan.

3.7. Usulan Pemecahan Masalah

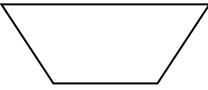
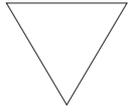
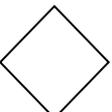
Sebagaimana yang telah dijelaskan diatas, dari permasalahan yang sedang dihadapi dapat ditarik kesimpulan bahwa permasalahan pokok yang dihadapi adalah mengenai pendataan kapal serta pengolahan data yang masih dilakukan

secara konvensional menggunakan buku laporan dan media papan. Untuk itu, melalui proses penelitian yang sudah penulis lakukan dan koordinasi dengan pihak petugas lapangan, maka peneliti memberikan ide atau masukan untuk membuat sebuah sistem informasi yang mampu mencatat mengenai pendataan dan pengelolaan data secara otomatis dan langsung menyimpan data tersebut ke dalam *database* serta proses rekap data yang mudah. Selain itu, sistem ini bisa membuat laporan tentang pendataan dan pengolahan data lainnya sesuai kebutuhan dari Pelabuhan CPO Kabil. Dengan adanya sistem ini diharapkan akan mempermudah pihak Pelabuhan dalam mengontrol proses pengolahan dan pendataan data.

3.7.1. Aliran Sistem Informasi

Dalam jurnalnya (Ismael, 2017 : 149), Aliran Sistem Informasi (ASI) merupakan bagan yang menunjukkan arus pekerjaan serta keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur prosedur yang ada dalam sistem adapun simbol-simbol yang dapat digunakan pada Aliran Sistem Informasi (ASI) ini adalah sebagai berikut:

SIMBOL	Nama	KETERANGAN
	Dokumen	Menunjukkan dokumen yang digunakan untuk <i>input</i> dan <i>output</i> , baik secara manual, mekanik atau menggunakan computer

	Kegiatan manual	Menunjukkan pekerjaan yang dikerjakan secara manual
	Proses Komputer	Menunjukkan kegiatan proses dari operasi program komputer
	File Storage/Arsip	Menggambarkan penyimpanan data baik dalam bentuk arsip atau file computer
	<i>Disket</i>	<i>Input/output</i> dengan menggunakan <i>disket</i>
	<i>Decision</i>	Simbol keputusan yang digunakan untuk suatu penyeleksian kondisi di dalam program
	Terminal	<i>Input</i> dengan menggunakan <i>keyboard</i>
	Alur Garis	Menunjukkan alur dari proses

	<p>Simbol Penghubung</p>	<p>Digunakan untuk penghubung ke halaman yang masih sama atau ke halaman lain</p>
---	--------------------------	---

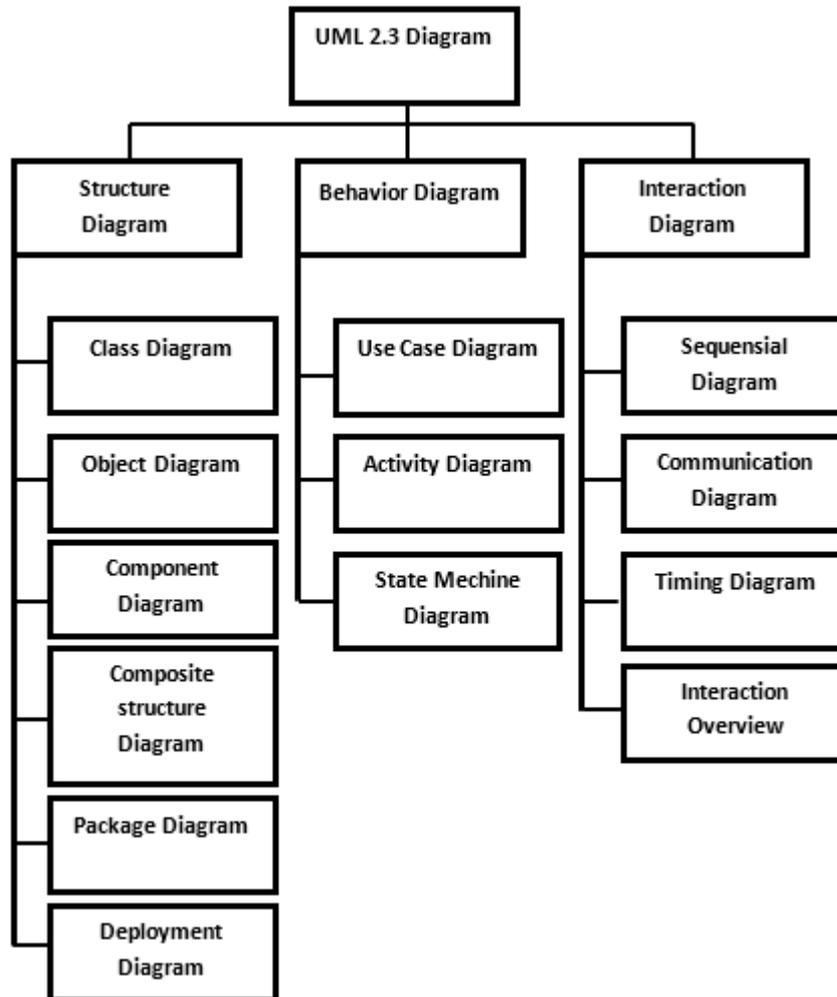
Tabel 3. 1 Aliran Sistem Informasi

3.7.2. UML

“UML merupakan sintesis dari tiga metode analisis dan perancangan berbasis objek serta ditambah keunggulan metode-metode berorientasi objek lainnya (Fision,Shlaer-Mellon, Coad-Yurdon) yang juga disintesakan dalam UML menawarkan pendekatan yang cukup baik serta digunakan cukup luas digunakan di industri perangkat lunak. Secara umum, UML merupakan ‘bahasa’ untuk visualisasi, spesifikasi, konstruksi serta dokumentasi”.

Menurut dalam jurnal (gintoro, andreyus, emilia, 2010: 30), UML (*Unified Modeling Language*) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk memvisualisasi, menspesifikasi, merancang dan mendokumentasi sistem piranti lunak (Booch et al, 1999, p14). UML memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*.

Menurut (A.S & Shalahuddin, 2013: 140)UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar berikut:



Sumber:(A.S & Shalahuddin, 2013: 140)

Gambar 3. 4 Diagram UML

Adapun diagram - diagram yang digunakan pada UML antara lain adalah *class diagram* , *use case diagram* , *activity diagram* , dan *sequence diagram*.

1. Class Diagram

Menurut (Isa & Hartawan, 2017: 141-142), Class adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi).

Kelas memiliki tiga area pokok:

- 1) Nama (Class Name)
- 2) Atribut
- 3) Metode (Operation)

Berikut simbol-simbol yang ada pada diagram Class Diagram:

Tabel 3. 2 Simbol Kelas Diagram

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktu sistem.
2.	Antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	Asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4.	Asosiasi berarah / directed association 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.

Lanjutan Tabel 3.2

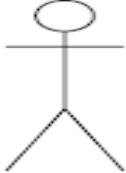
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / dependency 	Relasi antar kelas dengan makna kebergantungan antarkelas.
7.	Agregasi / aggregation 	Relasi antarkelas dengan makna semua-bagian (whole-part).

Sumber:(A.S & Shalahuddin, 2013: 146-147)

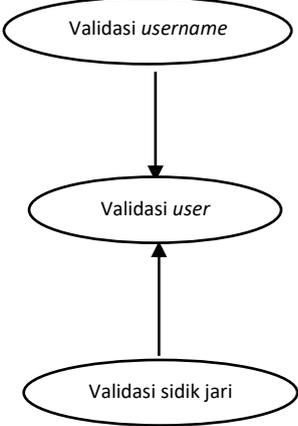
2. Use Case Diagram

Menurut (Isa & Hartawan, 2017: 141), Use Case diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*). Sehingga pembuatan *use case* diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case* diagram mempresentasikan sebuah interaksi antara aktor dengan sistem. Berikut simbol-simbol yang ada pada diagram Use Case Diagram:

Tabel 3. 3 Simbol Use Case Diagram

No.	Simbol	Deskripsi
1.	Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
2.	Aktor / actor  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor itu sendiri adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Asosiasi / association 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.

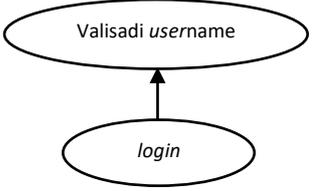
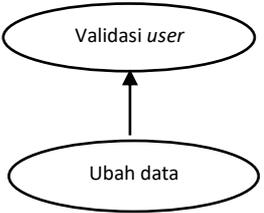
Lanjutan Tabel 3.3

4.	<p>Ekstensi / extend</p> <p><<extend>></p> 	<p>Relasi use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, misal</p> <div style="text-align: center;">  <pre> graph TD A([Validasi user]) --> B([Validasi username]) C([Validasi sidik jari]) --> A </pre> </div> <p><<extend>></p> <p><<extend>></p> <p>arah panah mengarah pada use case yang ditambahkan; biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.</p>
----	--	--

Lanjutan Tabel 3.3

5.	Generalisasi / generalization	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> <div data-bbox="858 734 1102 1048" style="text-align: center;"> <pre> graph TD A([Ubah data]) --> B([Mengelola]) C([Hapus data]) --> B </pre> </div> <p>arah panah mengarah pada use case yang menjadi generalisasi (umum).</p>
6.	Menggunakan / include/ uses	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di use case:</p>

Lanjutan Tabel 3.3

		<p>1) Include berarti use case yang ditambahkan akan selalu dipanggil saat use case</p> <p>2) tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT login([login]) --> validasi([Validasi username]) </pre>
		<p>3) Include berarti use case tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT ubah([Ubah data]) --> validasi([Validasi user]) </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber:(A.S & Shalahuddin, 2013: 156-158)

3. *Activity Diagram*

Menurut (Isa & Hartawan, 2017: 141-142), *Activity diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat jugadigunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya, atau dari aktifitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*. Berikut simbol-simbol yang ada pada diagram *Activity Diagram*:

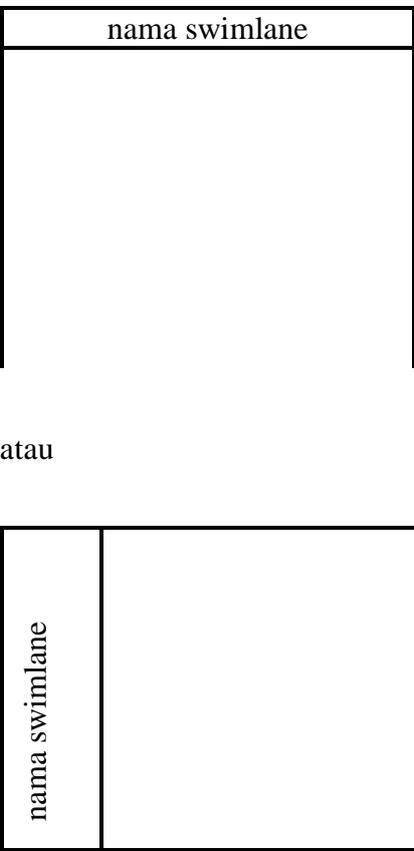
Tabel 3. 4 Simbol Activity Diagram

No.	Simbol	Deskripsi
1.	Status awal 	Staus awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

Lanjutan Tabel 3.4

2.	Aktivitas  aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Lanjutan Tabel 3.4

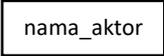
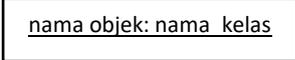
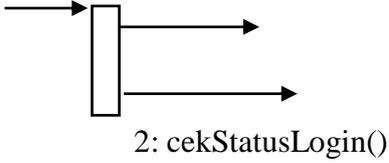
6.	<p><i>Swimlane</i></p>  <p>atau</p>	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.</p>
----	---	---

Sumber:(A.S & Shalahuddin, 2013: 162-163)

4. *Sequence Diagram*

Menurut (Isa & Hartawan, 2017: 141), *Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. *Berikut simbol-simbol yang ada pada diagram Sequence Diagram:*

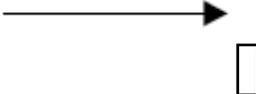
Tabel 3.5 Simbol Squence Diagram

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>Tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar 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 di awal frase nama aktor.</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> 

Lanjutan Tabel 3.5

		<p>1: login()</p> <p>3: open()</p> <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login()</p> <p>Aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe create</p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe call</p> <p>1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p> <p>-----</p> <p>1: nama_metode()</p>

Lanjutan Tabel 3.5

		Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	<p>Pesan tipe <i>send</i></p> <p>1:masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	<p>Pesan tipe <i>return</i></p> <p>keluaran</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	<p>Pesan tipe <i>destroy</i></p> <p><<destroy>></p> 	Menyatakan suatu objek mengakhiri hidup yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber:(A.S & Shalahuddin, 2013: 165-167)