

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi informasi, banyak instansi pemerintah kini beralih ke sistem elektronik untuk mengelola berbagai proses administratif, termasuk Surat Perintah Perjalanan Dinas (SPPD). Dalam prakteknya, pengelolaan perjalanan dinas masih dihadapkan pada beberapa tantangan, seperti penjadwalan perjalanan, pengelolaan anggaran, dan pelaporan keuangan. Oleh karena itu, diperlukan peningkatan sistem yang mampu mengoptimalkan proses ini.

Algoritma *Ternary Search* adalah algoritma pencarian yang efisien untuk mencari nilai maksimum atau minimum dalam suatu fungsi yang dapat dihitung. Penerapan algoritma ini pada sistem E-SPPD diharapkan dapat meningkatkan performa pencarian informasi terkait perjalanan dinas.

Melalui implementasi algoritma *Ternary Search*, diharapkan dapat mempercepat proses pencarian informasi terkait perjalanan dinas. Keunggulan algoritma ini dalam menemukan nilai maksimum atau minimum pada fungsi dapat memberikan hasil yang lebih akurat dan efisien.

Kelebihan algoritma *ternary search* adalah pencarian *ternary* dapat menemukan maximal/minimal untuk fungsi unimodal, dimana pencarian biner tidak berlaku, pencarian *Ternary* memiliki kompleksitas waktu $O(2 * \log_3 n)$, yang lebih efisien daripada pencarian linier dan sebanding dengan pencarian biner, dan cocok dengan masalah optimasi. Sedangkan, kelemahan algoritma *ternary search* adalah pencarian *Ternary* hanya berlaku untuk daftar atau larik

yang diurutkan, dan tidak dapat digunakan pada kumpulan data yang tidak berurutan atau non-linier, pencarian *Ternary* membutuhkan lebih banyak waktu untuk menemukan maksimal/minima fungsi monotonik dibandingkan dengan Pencarian Biner. (geeksforgeeks.org, 2023)

Pemilihan Dinas Komunikasi dan Informatika Kota Langsa sebagai lingkungan implementasi skripsi ini dilakukan dengan pertimbangan kebutuhan efisiensi dalam pengelolaan perjalanan dinas di instansi pemerintah tersebut.

Penelitian ini bertujuan untuk meningkatkan efisiensi dan akurasi sistem E-SPPD melalui implementasi algoritma *Ternary Search* pada Dinas Komunikasi dan Informatika Kota Langsa.

Dengan adanya penelitian ini, diharapkan dapat memberikan kontribusi signifikan terhadap pemecahan masalah dalam pengelolaan perjalanan dinas di sektor pemerintahan, khususnya pada Dinas Komunikasi dan Informatika Kota Langsa.

Berdasarkan latar belakang tersebut, penulis membuat judul skripsi dengan judul **“Implementasi Algoritma *Ternary Search* Pada E-SPPD (*Electronic Surat Perintah Perjalanan Dinas*) Di Dinas Komunikasi Dan Informatika Kota Langsa”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang mendasari penulis melakukan penelitian ini, penulis merumuskan beberapa rumusan masalah antara lain sebagai berikut :

1. Bagaimana merancang dan membangun e-SPPD (*electronic surat perintah perjalanan dinas*) di dinas Komunikasi dan Informatika Kota Langsa

berbasis *web* dengan menggunakan PHP MySQL?

2. Bagaimana menerapkan algoritma *ternary search* pada e-SPPD (*electronic* surat perintah perjalanan dinas) di dinas Komunikasi dan Informatika Kota Langsa?

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak melebar dan memudahkan dalam proses penelitian maupun proses perancangan, maka diperlukan batasan masalah sebagai berikut :

1. Aplikasi ini menggunakan data dari pegawai Dinas Komunikasi dan Informatika yang berjumlah 5 orang.
2. Aplikasi e-SPPD (*electronic* surat perintah perjalanan dinas) di Dinas Komunikasi dan Informatika Kota Langsa menggunakan bahasa pemrograman PHP dan *database* nya MySQL, dan XAMPP.
3. Pemodelan data menggunakan UML.

1.4 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan oleh penulis antara lain sebagai berikut :

1. Untuk membangun aplikasi e-SPPD (*electronic* surat perintah perjalanan dinas) di Dinas Komunikasi dan Informatika Kota Langsa.
2. Untuk menerapkan algoritma *ternary search* di aplikasi e-SPPD (*electronic* surat perintah perjalanan dinas) di dinas Komunikasi dan Informatika Kota Langsa.

1.5 Manfaat Penelitian

Penulis mengharapkan dari penelitian yang dilakukan dapat

memberikan efek yang positif dan memberikan manfaat.

1.5.1 Manfaat Penelitian Bagi Mahasiswa

1. Menambah wawasan penulis tentang bagaimana mengembangkan sebuah aplikasi berbasis *website* dengan pemrograman PHP dan basis data MySQL.
2. Menciptakan sebuah aplikasi berbasis *web* yang diharapkan dapat membantu dalam mempermudah dalam mengurus dan mendapatkan surat perjalanan dinas di Dinas Komunikasi dan Informatika Kota Langsa.

1.6 Sistematika Penulisan

Untuk mempermudah dalam penyusunan dan memahami skripsi maka penulis menyajikan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini memuat tentang materi-materi pendukung dalam penyusunan skripsi, mulai dari teori-teori yang digunakan, konsep-konsep yang akan diterapkan dalam menyelesaikan permasalahan yang penulis teliti dalam penelitian ini.

BAB III : METODE PENELITIAN

Pada bab ini memuat mengenai metode yang penulis gunakan dalam menyelesaikan rumusan masalah, tahap-tahap mengenai teknik pengolahan data, perancangan aplikasi, dan pembuatan

aplikasi.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini memuat hasil-hasil yang didapat dari penelitian serta melakukan pembahasan atas hasil yang diperoleh. Kesulitan yang ditemukan saat perancangan dan pembuatan aplikasi.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini memuat kesimpulan dan saran penulis atas penelitian yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1 Implementasi

Implementasi adalah suatu kegiatan atau suatu tindakan dari sebuah rencana yang dibuat secara terperinci untuk mencapai suatu tujuan. Implementasi mulai dilakukan apabila seluruh perencanaan sudah dianggap sempurna. Implementasi menurut teori Jones bahwa: *“Those Activities directed toward putting a program into effect”* (Proses mewujudkan program hingga memperlihatkan hasilnya). Jadi Implementasi adalah tindakan yang dilakukan setelah suatu kebijakan ditetapkan. Implementasi merupakan cara agar sebuah kebijakan dapat mencapai tujuannya. (Mulyadi, 2015)

Pengertian implementasi diatas menjelaskan bahwa implementasi itu bukan sekedar aktivitas saja, tetapi juga kegiatan terencana yang dilaksanakan dengan sungguh-sungguh berdasarkan acuan-acuan yang direncanakan dengan sungguh-sungguh. Oleh karena itu implementasi tidak berdiri sendiri tetapi dipengaruhi oleh objek berikutnya yaitu terlaksananya suatu program.

2.2 Algoritma

Algoritma adalah metode atau langkah yang direncanakan secara tersusun dan berurutan untuk menyelesaikan atau memecahkan permasalahan dengan sebuah intruksi atau kegiatan. Pada beberapa sumber buku diperoleh pengertian dari algoritma di antaranya :

1. Algoritma menurut (Kani, 2020, 1.19) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu

masalah untuk menghasilkan suatu output tertentu.

2. Algoritma berasal dari kata algoris dan ritmis yang pertama kali diperkenalkan oleh Abu Ja'far Muhammad Ibn Musa Al Khwarizmi pada 825 M di dalam buku *Al-Jabr Wa-al Muqabla*. Dalam bidang pemrograman, algoritma didefinisikan sebagai metode yang terdiri dari serangkaian langkah yang terstruktur dan sistematis untuk menyelesaikan masalah dengan bantuan komputer (Jando & Nani, 2018, 5).
3. Algoritma menurut (Munir & Lidya, 2016, 5) adalah urutan langkah-langkah untuk menyelesaikan suatu persoalan.

Algoritma merupakan sekumpulan instruksi atau langkah-langkah yang dituliskan secara sistematis dan digunakan untuk menyelesaikan masalah / persoalan logika dan matematika dengan bantuan komputer. (Sismoro, 2005, 29).

2.3 Ternary Search

Algoritma pencarian *Ternary* adalah algoritma *divide and conquer* yang dapat digunakan untuk menemukan elemen dalam array. Ini mirip dengan pencarian biner dimana kita membagi array menjadi dua bagian. Dalam hal ini, kita membagi array yang diberikan menjadi tiga bagian dan menentukan mana yang memiliki kunci

(elemen yang dicari). Kita dapat membagi array menjadi tiga bagian dengan mengambil mid_1 dan mid_2 yang dapat dihitung seperti yang ditunjukkan dibawah ini. Awalnya, l dan r akan sama dengan 0 dan $n-1$, dimana n adalah panjang array (Kumar & Sharma, 2016).

Berikut adalah cara untuk menentukan pembatas pada masing-masing bagian:

$$mid1 = \frac{l+(r-l)}{3} \quad (2.1)$$

$$mid2 = \frac{r-(r-l)}{3} \quad (2.2)$$

Keterangan:

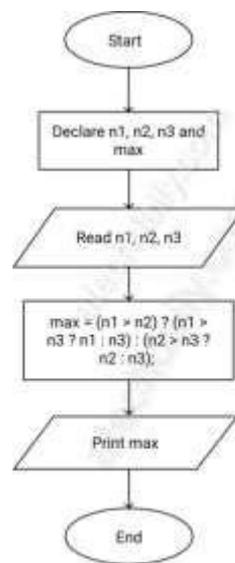
Mid1 = Pembatas pertama

Mid2 = Pembatas kedua

l = array pertama (0)

r = n-1 (= Panjang array)

Untuk lebih jelas dalam memahami, dapat dilihat dari *flowchart* dibawah ini.



Gambar 2.1 *Flowchart* Algoritma Ternary Search

2.4 SPPD

Menurut Aginta Genius dan Febriliyan Samopa (2013) Surat perjalanan dinas atau dikenal dengan SPPD merupakan jenis surat tugas yang dibutuhkan oleh seorang pegawai dalam melaksanakan tugas kerja dan harus pergi ke daerah lain (misalnya ke luar kota). Tugas kerja tersebut dapat berupa studi banding, audit ke perusahaan cabang, rapat kerja nasional maupun pekerjaan-pekerjaan lain yang dilakukan di luar kota (tidak di lingkungan kantor sendiri). Karena

perjalanan kerja ke luar kota membutuhkan biaya tambahan dan itu harus ditanggung oleh perusahaan, maka surat ini dibutuhkan sebagai bukti bahwa seorang pegawai atau karyawan melaksanakan tugas kerja ke luar kota sekaligus juga berfungsi sebagai bukti perintah dari atasan kepada bawahannya untuk melakukan tugas/pekerjaan di luar kota. Hal-hal yang dicantumkan dalam SPPD antara lain:

1. Pejabat atau Atasan yang memberikan perintah.
2. Pegawai atau Karyawan yang diberikan perintah berikut dengan orang-orang yang ikut dalam perjalanan dinas.
3. Maksud dan Tujuan Perjalanan Dinas.
4. Lamanya perjalanan dinas dilakukan.
5. Tempat asal dan tempat tujuan dinas.
6. Pembebanan Anggaran terhadap biaya perjalanan dinas.
7. Tanda tangan pejabat terkait.
8. Keterangan Perjalanan yang diisi oleh pegawai yang melakukan perjalanan dinas.

2.5 Website

Website adalah kumpulan dari halaman-halaman situs yang terdapat dalam sebuah *domain* atau *sub domain* yang berada di dalam *World Wide Web* (WWW) di internet. Alasan seseorang mengunjungi *website* adalah karena konten yang tersedia di *website* tersebut. Contoh *website* adalah *Google.com* dan *Facebook.com*. Penyebaran informasi melalui *website* sangat cepat dan mencakup area yang luas serta tidak dibatasi oleh jarak dan waktu. Oleh sebab itu, *website* merupakan sarana penting untuk mendapatkan dan mengelola

informasi.

2.5.1 Pengertian *Website* Menurut Para Ahli

Terdapat sejumlah pengertian website menurut para ahli sebagai berikut.

- *Website* adalah kumpulan halaman web yang saling terhubung dan seluruh file saling terkait. Web terdiri dari page atau halaman dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas dengan halaman-halaman terkait berada di bawahnya. Biasanya, setiap halaman di bawah *homepage* (child page) berisi *hyperlink* ke halaman lain dalam web (Gregorius, 2000).
- *Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan link dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik antara *page* yang disimpan dalam *server* yang sama maupun *server* di seluruh dunia. Halaman dapat diakses dan dibaca melalui browser seperti *Google Chrome*, *Mozilla Firefox*, dan lainnya (Hakim Lukmanul, 2004).
- *Website* adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara, dan animasi sehingga menarik untuk dikunjungi (Sholechul Azis, 2013).

2.5.2 Jenis-Jenis *Website*

Berdasarkan buku CMM *Website Interaktif MCMS Joomla (CMS)*, jenis

website dibagi berdasarkan sifat, tujuan, dan bahasa pemrograman.

Jenis-jenis *website* berdasarkan sifatnya adalah:

- *Website* dinamis, yaitu sebuah *website* yang menyediakan konten atau isi yang selalu berubah setiap saat. Contoh *website* dinamis adalah media berita daring.
- *Website* statis, merupakan *website* yang kontennya sangat jarang diubah. Misalnya, profil organisasi dan sebagainya.

Berdasarkan tujuannya, jenis *website* dibedakan sebagai berikut:

- *Personal website*, yaitu situs web yang berisi informasi pribadi seseorang. *Corporate web*, merupakan *website* yang dimiliki perusahaan.
- *Portal website*, yaitu *website* yang memiliki banyak layanan, seperti layanan berita, email, dan jasa-jasa lainnya.
- *Forum website*, yaitu sebuah situs web yang bertujuan sebagai sarana diskusi pengunjunnya. Selain itu, terdapat *website* pemerintah, e-banking, e-payment, e-procurement dan sebagainya.

Ditinjau dari segi bahasa pemrograman yang digunakan, jenis *website* dibagi menjadi dua, yaitu:

- *Server side*, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung pada tersedianya *server*, seperti PHP, ASP, dan sebagainya. Jika tidak ada *server*, *website* yang dibangun menggunakan bahasa pemrograman tidak dapat berfungsi sebagaimana mestinya.
- *Client side*, yaitu *website* yang tidak membutuhkan *server* dalam

menjalankannya, cukup diakses melalui *browser*.

2.6 PHP

PHP adalah bahasa pelengkap HTML yang dapat digunakan untuk membuat aplikasi dinamis yang memungkinkan manipulasi dan pemrosesan data. *Syntax* apa pun yang diberikan sepenuhnya dijalankan di *server* sementara hanya hasilnya yang dikirim ke *browser*. Kemudian itu adalah bahasa *scripting* yang ditempatkan di *server* dan diproses oleh *server*. Hasilnya dikirim ke klien tempat pengguna menggunakan *browser*. PHP merupakan sebuah perangkat lunak *Open Source*. (Tim EMS, 2012 : 61)

2.7 XAMPP

XAMPP merupakan *software server apache* dimana dalam XAMPP yang telah tersedia *database server* seperti MySQL dan PHP programming. XAMPP memiliki keunggulan yaitu cukup mudah dioperasikan, tidak memerlukan biaya serta mendukung instalasi pada *Windows* dan *Linux*. Keuntungan lain yang didapatkan adalah hanya dengan melakukan instalasi cukup satu kali kemudian didalamnya tersedia MySQL, *apache web server*, *database server* PHP. (Mawaddah & Fauzi, 2018)

2.8 MySQL

MySQL merupakan suatu program *database server* dimana perangkat tersebut mampu untuk digunakan sebagai transaksi menerima dan mengirim dengan waktu yang singkat pengguna dengan jumlah yang banyak sesuai standar SQL (*Structured Query Language*) yaitu bahasa pemrograman *database*. MySQL dapat diakses berdasarkan privileg (hak *user*) secara bersamaan.

(Parulian, 2017).

2.9 *Unified Modelling Language (UML)*

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan grafis yang digunakan sebagai standar untuk memodelkan sistem dengan metodologi pemodelan berorientasi objek (Mulyani, 2016). Sedangkan menurut (Mujilan, 2017) UML merupakan diagram sistem menggunakan orientasi objek (*object oriented*) dalam analisis dan perancangan sistem yang saat ini menjadi standar dalam berbagai tipe solusi perangkat lunak dalam pengembangan sistem.

Beberapa fungsi dan kegunaan dari UML yaitu (Mulyani, 2016:) :

1. *Visualizing*, yaitu sebagai alat komunikasi konseptual model antara tim pengembang sistem (sistem analis dengan programmer).
2. *Specifying*, yaitu sebagai tools yang digunakan untuk memodelkan sistem secara tepat dan jelas.
3. *Constructing*, yaitu UML sebagai bahasa grafis mampu melakukan mapping dan konseptual model kedalam bahasa pemrograman.
4. *Documenting*, yaitu UML digunakan sebagai tools untuk melakukan dokumentasi teknis sebuah sistem.

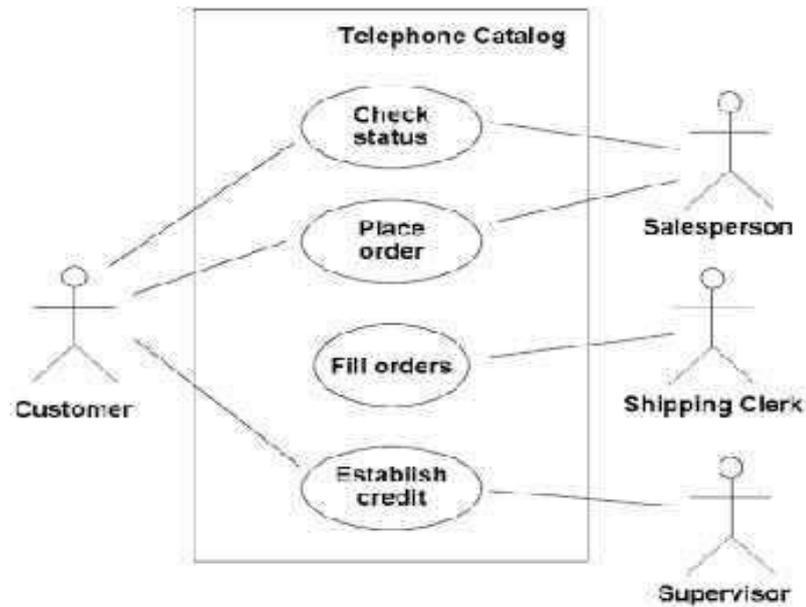
Terdapat beberapa diagram *Unified Modeling Language (UML)* yang sering digunakan dalam pengembangan sistem menurut (Mulyani, 2016) yaitu :

1. *Use Case Model*

Use case model merupakan kumpulan diagram dan *text* yang saling bekerja sama untuk mendokumentasikan bagaimana *user* (aktor) berinteraksi dengan sistem. *Use case model* terdiri dari beberapa diagram yaitu :

a. *Use Case Diagram*

Use case diagram yaitu diagram yang menggambarkan dan merepresentasikan aktor, *user cases* dan *dependencies* suatu proyek dimana tujuan dari diagram ini adalah untuk menjelaskan konsep hubungan antara sistem dengan dunia luar.



Gambar 2.2 Contoh *Use Case Diagram*

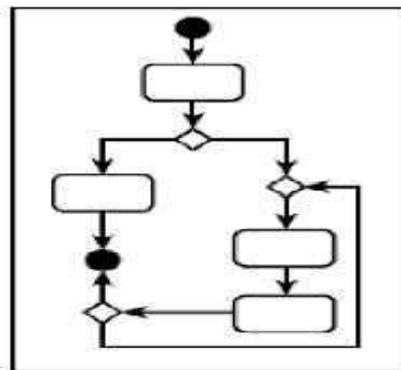
b. *Use Case Narrative*

Use case narrative yaitu deskripsi yang menjelaskan *use case diagram*. Pada *use case diagram* sistem hanya digambarkan secara sederhana menggunakan simbol *use case* yang berhubungan (*relationship*) dengan aktor, sehingga terkadang diperlukan deskripsi yang menjelaskan dari proses tersebut.

Gambar 2.3 Contoh *Use Case Narrative*

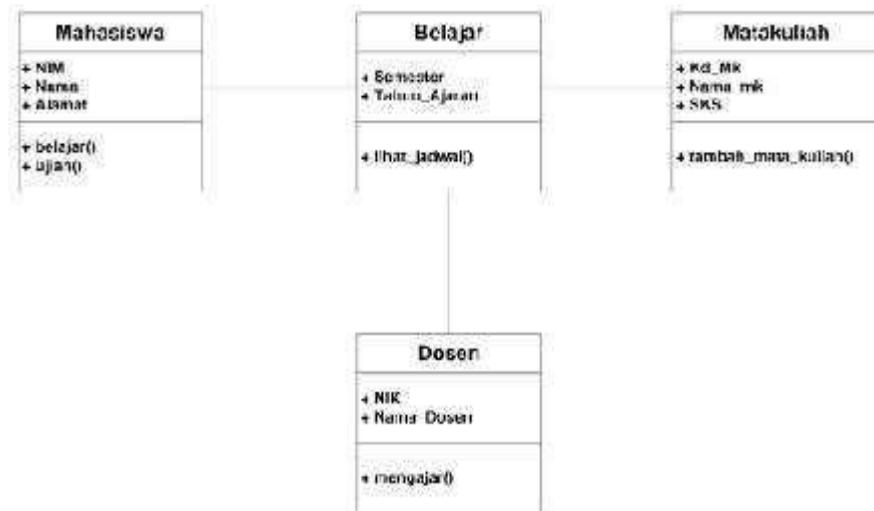
c. *Use Case Skenario*

Use case scenario yaitu pemecahan kemungkinan logika pada *use case diagram*.

Gambar 2.4 Contoh *Use Case Skenario*

2. *Class Diagram*

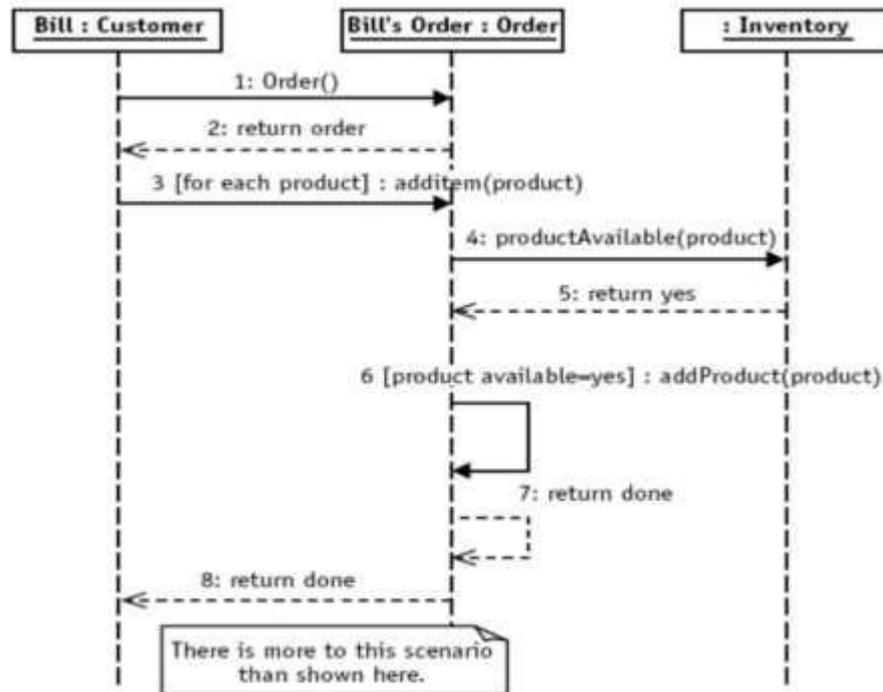
Class diagram adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan antar masing-masing kelas. Selain itu *class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan *property* dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.



Gambar 2.5 Contoh *Class Diagram*

3. *Activity Diagram*

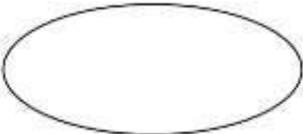
Activity diagram adalah diagram *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan alur aktivitas dari satu proses. *Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya, dengan kata lain diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus diikuti.

Gambar 2.6 Contoh *Sequence Diagram*

2.10 Daftar Simbol Diagram

a. Daftar Simbol *Use Case Diagram*

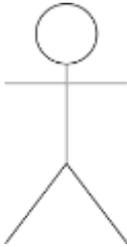
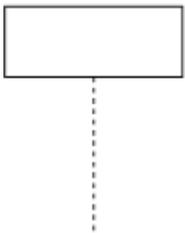
Tabel 2.1 Tabel Daftar Simbol *Use Case Diagram*(Maharani, 2018)

Simbol	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran ketika berinteraksi dengan sistem usulan.
	Use Case	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem, dan mewakili sebagian besar sistem secara fungsional.

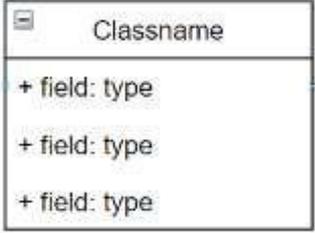
	Sistem	Menggambarkan ruang lingkup sistem.
	Asosiasi	Menghubungkan aktor dengan use case yang berinteraksi.
	Ekstend	Relasi yang menggambarkan bahwa sebuah use case (sub use case) bisa berdiri sendiri atau bisa berjalan tanpa menjalankan main use case terlebih dahulu.
	Include	Relasi yang menggambarkan bahwa sebuah use case (sub use case) harus menjalankan use case lain terlebih dahulu sebelum menjalankan fungsinya.

b. Daftar Simbol *Sequence Diagram*

Tabel 2.2 Tabel Daftar Simbol *Sequence Diagram*(Maharani, 2018)

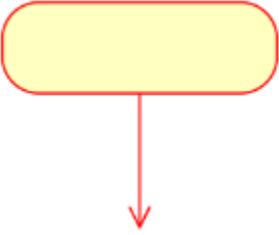
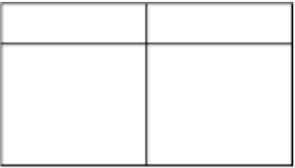
Simbol	Nama	Keterangan
	Actor	Menspesifikasikan himpunan peran ketika berinteraksi dengan sistem usulan
	Object Lifeline	Menyatakan hidup uatu object dalam basis waktu
	Activation	Menyatakan object dalam keadaan aktif dan berinteraksi
	Message	Pesan antar object, dan menggambarkan urutan kejadian
	Message return	Menyatakan arah kembali antara urutan kejadian

c. Daftar Simbol *Class Diagram*Tabel 2.3 Tabel Daftar Simbol *Class Diagram* (Maharani, 2018)

Simbol	Nama	Keterangan
	Class	<i>Class diagram</i> ini terdiri dari nama kelas, atribut kelas, dan metode / <i>operation</i> (fungsi yang dimiliki suatu kelas)
	Asosiasi	Menyatakan hubungan statis antar <i>class</i> , dan di simbolkan dengan garis tegas saja.
	Agregasi	Hubungan yang menyatakan terdiri atas, dimana <i>class</i> yang satu merupakan bagian dari <i>class</i> lain, namun kedua <i>class</i> ini dapat berdiri sendiri.
	Komposisi	Bentuk khusus dari agragasi dimana <i>class</i> yang menjadi bagian, baru dapat dibuat setelah <i>class</i> yang menjadi <i>whole</i> dibuat.

d. Daftar Simbol Activity Diagram

Tabel 2.4 Tabel Daftar Simbol *Activity Diagram* (Maharani, 2018)

Simbol	Nama	Keterangan
	Start Poin	Merupakan awal penelusuran. Sebuah activity diagram selalu dimulai dengan start poin
	End Point	Merupakan akhir dari penelusuran. Sebuah activity diagram selalu diakhiri dengan End Point
	Activities	Activity menggambarkan proses, disisi dengan kata kerja atau merupakan state dari sistem yang mencerminkan eksekusi dari suatu aksi.
	Swimline Style	Sebuah cara untuk mengelompokan activity berdasarkan actor. Actor bisa ditulis dengan nama actor.