

## BAB II

### LANDASAN TEORI

#### 2.1 Kajian Terhadap Penelitian Yang Terkait Sebelumnya

Kajian utama penelitian adalah pengembangan perangkat lunak sistem pemesanan online atau toko online dan masalah optimasi rute perjalanan atau *Travelling Salesman Problem* (TSP). Beberapa penelitian terdahulu terkait dengan kajian penerapan algoritma untuk menyelesaikan permasalahan TSP adalah penelitian yang dilakukan oleh Sayed Fachrurrazi, Oloruntoyin Sefiu Taiwo, Medrio Dwi Aksara Cipta Hasibuan. Berikut ini adalah uraian penelitian mereka :

##### 1. Penelitian oleh Sayed Fachrurrazi (2013)

Latar belakang penelitian adalah menyelesaikan permasalahan PT. Pupuk Iskandar Muda (PT. PIM) dalam mengirimkan barang ke konsumen. PT. PIM mengirimkan pupuk ke wilayah-wilayah di kabupaten Aceh Utara. Banyak tempat harus dikunjungi, dan wilayah operasinya sangat luas. Dibutuhkan cara menentukan rute yang tepat agar pendistribusian pupuk dapat dilakukan dalam waktu yang singkat dan efisien atau dengan kata lain jarak yang terpendek.

Algoritma yang digunakan adalah algoritma genetika. Perangkat lunak dirancang dengan menggunakan DFD, dan diimplementasikan menggunakan pemrograman visual berbasis desktop.

Penelitian menyimpulkan bahwa algoritma genetika dapat diimplementasikan untuk pencarian rute terpendek pendistribusian pupuk Pupuk Iskandar Muda dengan jumlah tujuan sebanyak 21 tempat/kota.

##### 2. Penelitian oleh Oloruntoyin Sefiu Taiwo, Olukehinde Olutosin Mayowa, dan Kolapo Bukola Ruka (2013).

Penelitian Sefiu Taiwo mencoba mengkaji penerapan Algoritma Genetika untuk memecahkan *traveling salesman problem* (TSP).

Algoritma genetika diimplementasikan menggunakan bahasa pemrograman Visual C # dengan alasan teknologi berorientasi objek, dekat ke bahasa alami, dukungannya yang bagus untuk basis data relasional, kecepatan eksekusinya, tidak memerlukan tambahan file untuk implementasinya, dukungan yang baik dari struktur kontrol dan perulangan.

Temuannya adalah :

- a. Algoritma genetika paling efisien dalam waktu komputasi tetapi paling tidak efisien dalam penggunaan memori.
  - b. Algoritma genetika berbeda dari algoritma *nearest neighbourhood heuristic* dimana algoritma genetika mencari rute terpendek sedangkan *nearest neighbourhood heuristic* mencari jalur/path terdekat.
  - c. Algoritma genetika membutuhkan sistem dengan arsitektur paralel untuk implementasi optimalnya.
3. Penelitian oleh Medrio Dwi Aksara Cipta Hasibuan (2015)

Tujuan dari penelitian Medrio adalah untuk menguji kemampuan algoritma genetika dalam pencarian solusi rute mendekati optimal pada pemodelan kegiatan pengangkutan sampah yang dikelola oleh Dinas Kebersihan dan Pertamanan Kota Pekanbaru ke dalam kasus TSP.

Simulasi algoritma genetika dilakukan dengan membuat sebuah perangkat lunak berbasis desktop. Perangkat lunak dirancang menggunakan bahasa pemodelan UML.

Berdasarkan pengujian disimpulkan bahwa Algoritma genetika dapat memberikan solusi pencarian rute optimal pada pemodelan kegiatan pengangkutan sampah Dinas Kebersihan dan Pertamanan Kota Pekanbaru.

## **2.2 Algoritma Genetika**

Algoritma genetika adalah suatu proses optimasi yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah (Haupt & Haupt 2004: 22). Berbeda dengan teknik pencarian konvensional, algoritma genetika dimulai dari himpunan solusi yang pada umumnya dihasilkan secara acak. Himpunan ini disebut populasi sedangkan setiap individu dalam populasi disebut kromosom (merupakan representasi dari solusi) dan yang menempati kromosom disebut gen. Gen biasanya merupakan simbol string. Proses evolusi alamiah yaitu terbentuknya populasi awal secara acak yang terdiri dari individu-individu dengan sifat yang tergantung pada gen-gen 20 dalam kromosomnya. Individu-individu melakukan proses reproduksi untuk melahirkan keturunan. Sifat keturunan dibentuk dari kombinasi sifat kedua induknya. Metode optimasi ini dikembangkan oleh John Holland pada tahun 1960-an dan dipopulerkan oleh mahasiswaanya, David Goldberg pada tahun 1980-an. Dalam bukunya "Adaptation in Natural and Artificial Systems", John Holland menjabarkan dasar-dasar algoritma genetika. Algoritma genetika didasarkan pada proses alamiah yaitu Teori Evolusi Darwin. Dalam algoritma genetika, suatu populasi dari individu bereproduksi kembali berdasarkan nilai fitness (Spears, 1989: 4). Kemunculan algoritma genetika diinspirasi dari teori-teori dalam ilmu biologi, sehingga banyak terdapat istilah dalam biologi yang digunakan dalam algoritma ini. Sesuai dengan namanya, proses-proses yang terjadi dalam algoritma genetika sama dengan apa yang terjadi pada evolusi biologi yaitu seleksi, pindah

silang, dan mutasi. Menurut Zainudin (2014: 11), pencarian dimulai dengan pembangkitan sejumlah “individu” secara acak yang disebut dengan kromosom. Kromosom-kromosom ini merupakan representasi calon penyelesaian yang akan diperiksa nilai yang sebenarnya. Seperti halnya proses evolusi alamiah, kromosom-kromosom akan dinilai tingkat kebugarannya (fitness). Semakin besar nilai fitness, semakin besar pula kemungkinannya untuk dipertahankan ke dalam populasi selanjutnya. Nilai fitness adalah nilai yang menunjukkan nilai ketangguhan kromosom dalam beradaptasi. Kromosom-kromosom yang dibentuk dari kromosom generasi sebelumnya disebut sebagai anak (offspring). Demikian juga dengan 21 kromosom generasi sebelumnya disebut sebagai induk (parents). Proses pembentukan anak dari induknya dilakukan dengan penyilangan (crossover). Dalam algoritma genetika, dikenal operator mutasi (mutation), yaitu operator yang dapat mengubah gen-gen dalam kromosom. Dalam parameter algoritma genetika, terdapat beberapa batasan. Salah satunya adalah ukuran populasi. Dalam algoritma genetika, ukuran populasi setiap generasi adalah tetap. Populasi generasi selanjutnya dibentuk dengan cara menyeleksi kromosom induk dan kromosom anak berdasarkan nilai fitness.

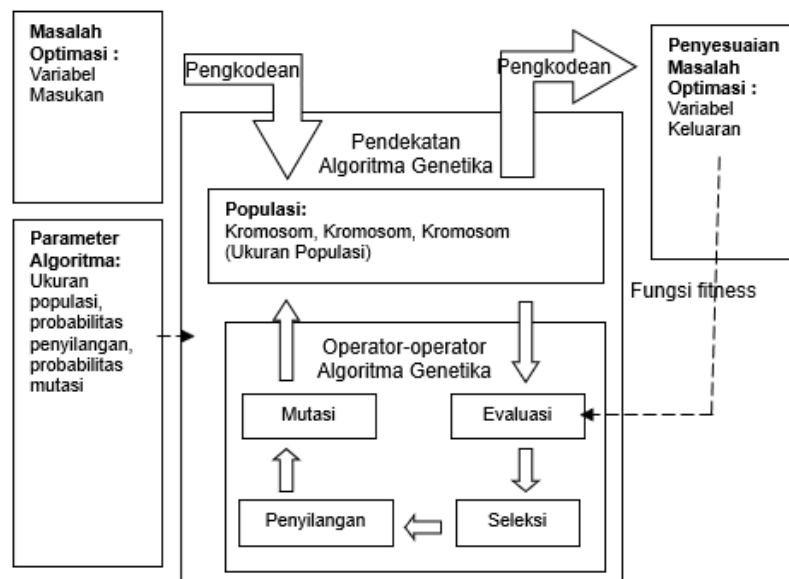
Tabel 2. 1 Pemetaan Proses Alamiah ke Proses Komputasi Algoritma Genetika

<b>Proses Alamiah</b>	<b>Proses Komputasi</b>
Individu	Penyelesaian masalah
Populasi	Himpunan penyelesaian
Fitness/kebugaran	Kualitas penyelesaian

Kromosom	Kode/representasi penyelesaian
Gen	Bagian dari representasi penyelesaian
Pertumbuhan	Pendekodean representasi penyelesaian
Penyilangan	Operator genetika
Mutasi	Operator genetika
Seleksi Alam	Menyeleksi penyelesaian masalah (sementara) berdasarkan kualitasnya

### 2.2.1 Prosedur Penerapan Algoritma Genetika

Prosedur penerapan Algoritma Genetika untuk menyelesaikan permasalahan optimasi ditunjukkan Gambar 2.1. Menurut Michalewicz (1998), keberhasilan penggunaan Algoritma Genetika sangat ditentukan oleh penentuan pernyataan masalah ke dalam bentuk titik-titik pencarian yang disebut dengan kromosom, serta pemilihan operator-operator yang digunakan.



Gambar 2. 1 Prosedur penerapan Algoritma Genetika untuk menyelesaikan masalah optimasi

Sumber Michalewicz (1996)

Prosedur penerapan Algoritma Genetika terdiri atas beberapa langkah :

1. Inisialisasi/pembentukan awal populasi
2. Evaluasi populasi.
3. Seleksi populasi yang akan dikenai operator genetika.
4. Proses penyilangan pasangan kromosom tertentu.
5. Proses mutasi kromosom tertentu.
6. Evaluasi populasi baru.
7. Ulangi dari langkah 3 selama syarat berhenti belum terpenuhi.

### 2.2.2 Pengkodean dalam Algoritma Genetika

Pengkodean adalah bagian penting Algoritma Genetika. Pengkodean dilakukan untuk membentuk kromosom sebagai wakil/wujud penyelesaian masalah. Tidak ada aturan khusus mengenai cara pengkodean. Kromosom dapat dirancang sebagai kode-kode tertentu, tetapi dengan persyaratan kromosom harus dapat diproses oleh operator-operator genetika.

Pada masalah optimasi rute perjalanan, biasanya kromosom terdiri atas indeks setiap kota. Jika ada 10 kota maka kromosom terdiri atas 10 gen. Setiap gen terdiri atas bilangan bulat yang merupakan indeks dari kota-kota tersebut. Semua gen sebagai sebuah kromosom harus merepresentasikan rute yang harus ditempuh sales. Misal  $p \in \{Yogyakarta, Denpasar, Kuala Lumpur, Bangkok, Manila, Kairo, Hongkong, Newyork, London, Paris\}$ , maka kromosom dapat dikodekan sebagai kombinasi bilangan 1 sampai 10.

Jika kesepuluh kota diindeks dengan urutan seperti itu, maka kromosom 1|2|3|4|5|6|7|8|9|10 merepresentasikan rute Yogyakarta – Denpasar – Kuala Lumpur – Bangkok – Manila – Kairo – Hongkong – Newyork – London – Paris, kromosom 10|2|3|4|5|6|7|8|9|1 merepresentasikan rute Paris – Denpasar – Kuala Lumpur – Bangkok – Manila –Kairo – Hongkong – Newyork – London – Yogyakarta dan sebagainya.

### 2.2.3 Seleksi dalam Algoritma Genetika

Seleksi merupakan proses dalam Algoritma Genetika untuk memilih kromosom yang tetap bertahan dalam populasi. Kromosom yang terpilih mempunyai kemungkinan untuk dipasangkan dengan kromosom lain atau mengalami proses penyilangan sebanding dengan probabilitas penyilangan yang menghasilkan kromosom anak.

Terdapat beberapa model seleksi dalam Algoritma Genetika, yaitu Seleksi Sebanding dengan Nilai Fitness (*Fitness Proportionate Selection*), Seleksi Peringkat (*Rank Selection*), Seleksi Turnamen, dan Seleksi Elitis. Model-model seleksi tersebut diterapkan dengan anggapan:

- a. Fungsi objektif sudah diketahui.
- b. Penyelesaian yang dicari adalah nilai maksimum.
- c. Nilai penyelesaian adalah positif.
- d. Ukuran kualitas kromosom adalah nilai fitness kromosom.

### 2.2.3 Penyilangan Dalam Algoritma Genetika

Penyilangan adalah operator Algoritma Genetika untuk menghasilkan kromosom baru yang mewarisi sifat-sifat induknya seperti proses reproduksi dalam kehidupan nyata. Dengan operator penyilangan proses pencarian solusi akan bergerak mencari titik-titik penyelesaian yang berbeda.

Kromosom baru hasil penyilangan disebut sebagai kromosom keturunan. Oleh karena populasi dalam Algoritma Genetika dimodelkan agar berukuran tetap, maka kromosom keturunan harus memungkinkan untuk dimasukkan dalam populasi baru.

Terdapat beberapa cara penggantian ini, yaitu :

- a. Penggantian secara acak (random replacement).
- b. Penggantian sebanding dengan nilai fitness (roulette replacement).
- c. Penggantian berdasarkan peringkatnya jika metode seleksi yang digunakan adalah metode seleksi peringkat (rank replacement).
- d. Penggantian peringkat fitness terburuk (absolute fitness replacement).
- e. Penggantian semua induk dengan semua keturunan jika fitness keturunan lebih besar dari pada fitness induknya (locally elite replacement).
- f. Penggantian kromosom secara acak jika fitness keturunan lebih besar dari pada fitness kromosom yang bersangkutan (random elite replacement).

Jenis-jenis penyilangan :

1. Penyilangan yang melibatkan kode biner : penyilangan n-titik dan penyilangan seragam
2. Penyilangan untuk optimasi kombinatorial : penyilangan berbasis posisi dan penyilangan berbasis urutan

#### 2.2.4 Mutasi dalam Algoritma Genetika

Mutasi adalah operator yang bertujuan mengubah gen-gen tertentu dari sebuah kromosom. Proses mutasi juga dimodelkan seperti yang terjadi di kehidupan nyata. Probabilitas mutasi dari suatu gen biasanya dipilih sangat kecil, persis seperti yang terjadi di alam. Jenis-jenis mutasi :

- a. Mutasi yang melibatkan kode biner
- b. Mutasi untuk optimasi kombinatorial
  - Mutasi berbasis posisi
  - Mutasi berbasis urutan
  - Mutasi campur aduk

##### 1. Syarat Berhenti

Proses pencarian solusi akan berhenti setelah suatu syarat berhenti dipenuhi. Beberapa syarat berhenti yang biasa digunakan yaitu banyaknya generasi, batas waktu komputasi, batas nilai fungsi objektif, batas nilai fungsi fitness, dan terjadinya konvergensi.

##### 2. Penentuan Parameter Algoritma

Parameter algoritma sangat penting dalam menerapkan Algoritma Genetika tetapi tidak mudah untuk menentukannya. Tidak ada aturan pasti menentukan ukuran populasi, probabilitas penyilangan, maupun probabilitas mutasi. Hal ini karena prinsip Algoritma Genetika yang mengandalkan bilangan acak hampir di setiap langkahnya. Bilangan acak yang berbeda akan menyebabkan hasil yang berbeda pula.

Parameter algoritma yang disarankan adalah:

- a. Probabilitas penyilangan cukup besar (berkisar 60% sampai 70%).
- b. Probabilitas mutasi cukup kecil (sebuah gen untuk sebuah kromosom).
- c. Ukuran populasi berkisar antara 50 sampai 500 kromosom.

### **2.3 Google Maps JavaScript API**

Peta dan lokasi adalah topik yang sangat populer dewasa ini, dan Google adalah salah satu permainan utama di area ini. Sebagian besar situs web memiliki halaman kontak dengan menampilkan peta Google. Hal tersebut adalah contoh penggunaan paling sederhana dari Google Maps JavaScript API.

Terdapat alternatif selain Google Maps JavaScript API, yaitu OpenLayers, Leaflet, Bing Maps, MapQuest, dan Here Maps (sebelumnya, Nokia Maps), tetapi hanya Google Maps JavaScript API yang memiliki dukungan terbesar dalam menyajikan peta dasar, citra satelit, dan API itu sendiri.

Google Maps JavaScript API bukan alat gratis untuk menampilkan peta, tetapi batas penggunaan gratisnya cukup untuk sebagian besar pengembang. Ada batas 25.000 beban peta per hari per situs, yang dihitung ketika peta diinisialisasi pada halaman web.

#### **2.3.1 Contoh Pemakaian Peta Google Maps**

Berikut adalah ini contoh untuk membuat peta menggunakan Google Maps JavaScript API.

1. Buat file kosong baru bernama map.html dan masukkan blok kode berikut ke dalamnya. Blok ini diperlukan untuk setiap aplikasi yang menggunakan

Google Maps JavaScript API. Siapkan dan masukkan kunci API Google Maps JavaScript ke URL dalam kode.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<!-- Include Google Maps JS API -->
```

```
<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=INSERT_YOUR_MAP_API_KEY_HERE&sensor=false"></script>
```

2. Bagian berikut diperlukan untuk menempatkan peta di mana diperlukan.

Dibagian `<head>`, tambahkan kode styling HTML untuk membuat peta dengan lebar 800 px dan 500 px tinggi dengan elemen `<style>` sebagai berikut:

```
<style type="text/css">
```

```
    #mapDiv { width: 800px; height: 500px; }
```

```
</style>
```

3. Tambahkan baris JavaScript berikut ke kode untuk dijalankan dengan Google Maps JavaScript API. Jangan lupa untuk mendefinisikan objek peta di luar fungsi untuk mengaksesnya dari setiap bagian kode.

```
<!-- Map creation is here -->
```

```

<script type="text/javascript">

//Defining map as a global variable to access

//from other functions var map;

function initMap() {

//Enabling new cartography and themes

google.maps.visualRefresh = true;

//Setting starting options of map

var mapOptions = {

center: new google.maps.LatLng(39.9078, 32.8252),

zoom: 10, };

//Getting map DOM element

var mapElement = document.getElementById('mapDiv');

//Creating a map with DOM element which is

//just obtained

map = new google.maps.Map(mapElement, mapOptions); }

```

4. Tambahkan baris berikut untuk menyelesaikan kode. Bagian ini mendefinisikan tag <html> di mana peta akan ditambahkan dan kapan untuk menginisialisasi peta.

```
google.maps.event.addDomListener(window, 'load',initMap);
```

```
</script>

</head>

<body>

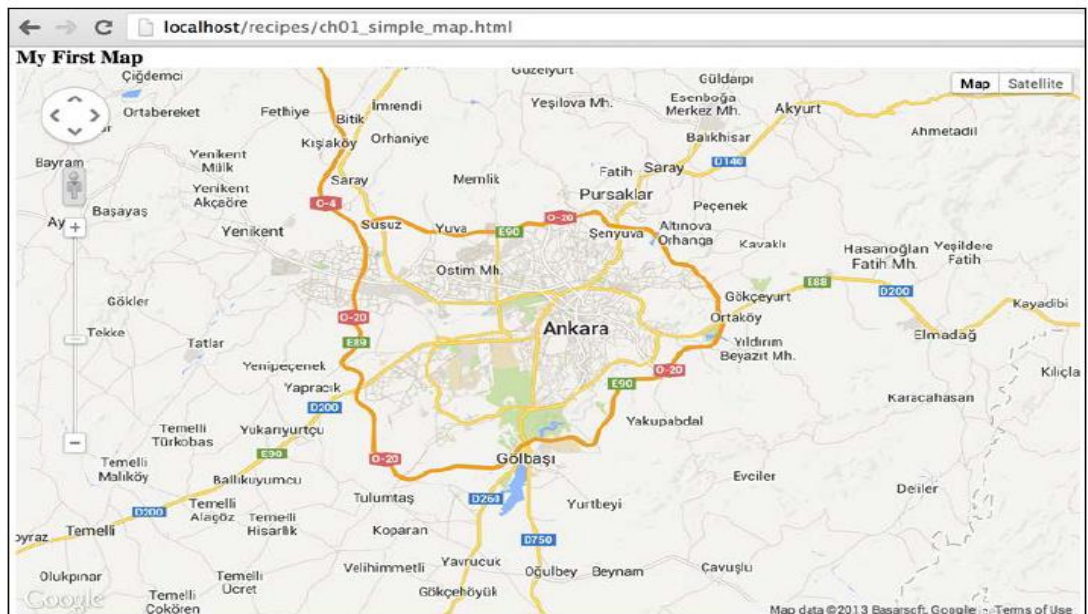
<b>My First Map </b>

<div id="mapDiv"></div>

</body>

</html>
```

5. Masukkan URL server lokal Anda, tempat file map.html Anda disimpan, di favorit Anda browser dan lihat hasilnya. Anda akan melihat peta dengan kontrol navigasi di sudut kiri atas dan kontrol peta dasar di sudut kanan atas.



Gambar 2. 2 Tampilan google maps

( sumber google maps Api )

### 2.3.2 Peta Dasar

Peta dasar adalah salah satu bagian terpenting dari proses pemetaan API. Mendasarkan peta menunjukkan jalan, gambar satelit, atau medan, yang dapat digunakan untuk situasi yang berbeda. Misalnya, peta jalan bisa cocok untuk menunjukkan lokasi bisnis Anda, tetapi citra satelit tidak.

Google Maps JavaScript API memiliki empat berbeda yaitu ROADMAP, SATELLITE, HYBRID, dan TERRAIN. Semua peta dasar ini dapat dilihat pada *screenshot* berikut di mana mereka dapat dibandingkan satu sama lain.



Gambar 2. 3 Tampilan tipe peta dasar (base maps) peta google

( sumber google maps Api )

### 2.3.3 Services atau Layanan Google Maps API

Google Maps JavaScript API juga menawarkan beberapa layanan. Layanan ini menambahkan fungsi penting yang sangat membedakan Google Maps dari pesaingnya. Keandalan dan kualitas data membuat *services* lebih dihargai. *Services* memungkinkan aplikasi yang memanfaatkan Google Maps memiliki tambahan fungsionalitas. Bagian terbaik dari layanan ini adalah, karena mereka adalah bagian dari Google Maps JavaScript API, mereka sepenuhnya kompatibel dengan kelas dan objek dari API.

Beberapa jenis layanan Google Maps JavaScript API adalah *geocoding*, *direction*, *elevation*, *distance matrix*, dan *Street View*.

1. *Geocoding* : untuk mencari koordinat sebuah alamat.

Menemukan alamat atau tempat di peta selalu menjadi tugas yang membosankan, dan Google Maps JavaScript API memudahkan tugas ini dengan layanan *geocoding*. *Geocoding*, dalam yang paling sederhana definisi, adalah untuk mengasosiasikan koordinat geografis dengan informasi alamat, baik itu hanya nama jalan, rincian nomor bangunan dan kode pos, atau hanya nama lokalitas.

Proses *geocoding* dapat dibalik dan diberi *nama reverse geocoding*, yaitu proses konversi koordinat ke alamat yang dapat dibaca manusia. Misalnya ketika pengguna melakukan klik pada peta, google akan menemukan alamat di mana pengguna melakukan klik dan menampilkan alamat tersebut ke pengguna.

2. *Elevation* : mengembalikan nilai positif medan/tanah relatif terhadap permukaan laut. *Elevation* juga memberikan informasi tentang kedalaman dasar samudra dengan nilai negatif
3. *Distance matrix* : membuat matriks jarak untuk yang lokasi diberikan Google Maps JavaScript API membawa beberapa properti yang menarik dan sangat membantu, salah satunya adalah *Distance Matrix Services*. Dengan menggunakan layanan ini, kita dapat menghitung waktu perjalanan dan jarak antara lokasi asal dan lokasi tujuan. Mode perjalanan dapat kita pilih yaitu mengemudi, berjalan, atau bersepeda. Perlu dicatat bahwa output layanan ini tidak dapat dipetakan ke peta dasar. Kita dapat memiliki informasi tentang waktu dan durasi perjalanan, tetapi untuk petunjuknya, kita harus menggunakan layanan *direction*.
4. *Direction* atau petunjuk arah atau rute : memberi pengembang kemampuan untuk menghasilkan arah secara terprogram antara lokasi pilihan mereka dengan berbagai opsi.

#### **2.4 Sistem Informasi.**

Sistem informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah darimana informasi tersebut bias didapatkan?. Informasi dapat diperoleh dari sistem informasi (*information systems*) atau disebut juga dengan *processing system* atau *information processing systems* atau *information-generating systems*.

Menurut Alter dalam Buku Kadir (2014:9) mendefinisikan bahwa, “Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi”.

Menurut Tata Sutabri (2012 : 22) informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan

## **2.5 Pengertian e-commerce**

Menurut Kotler & Armstrong (2012) E-commerce adalah saluran online yang dapat dijangkau seseorang melalui komputer, yang digunakan oleh pebisnis dalam melakukan aktifitas bisnisnya dan digunakan konsumen untuk mendapatkan informasi dengan menggunakan bantuan komputer yang dalam prosesnya diawali dengan memberi jasa informasi pada konsumen dalam penentuan pilihan. Menurut Wong (2010) e-commerce adalah proses jual beli dan memasarkan barang serta jasa melalui sistem elektronik, seperti radio, televisi dan jaringan komputer atau internet. Maka dapat disimpulkan bahwa e-commerce merupakan kumpulan dinamis antara teknologi, aplikasi dan proses bisnis yang menghubungkan perusahaan dan konsumen serta komunitas tertentu dimana pertukaran barang antara pengecer dan konsumen dari berbagai komoditi dalam skala luas dan suatu transaksi elektronik, dan dalam proses pengiriman barang dari pengecer menggunakan transportasi dari suatu wilayah ke wilayah lain hingga sampai ke tangan konsumen dan hubungan yang terjadi adalah hubungan yang saling menguntungkan kedua belah pihak.

## 2.6 Pengertian Internet

Internet adalah suatu jaringan komunikasi yang menghubungkan satu media elektronik dengan media yang lainnya. Standar teknologi pendukung yang dipakai secara global adalah *Transmission Control Protocol* atau *Internet Protocol Suite* (disingkat sebagai istilah TCP/IP). TCP/IP ini merupakan protokol pertukaran paket (dalam istilah asingnya *Switching Communication Protocol*) yang bisa digunakan untuk miliaran lebih pengguna yang ada di dunia. Sementara itu, istilah “*internetworking*” berarti cara/prosesnya dalam menghubungkan rangkaian internet beserta aturannya yang telah disebutkan sebelumnya.

## 2.7 PHP

Menurut Supono dan Putratama (2016:6) mengemukakan bahwa: Paket *Web Server* adalah sebuah perangkat lunak server yang berfungsi untuk menerima permintaan dalam bentuk situs web melalui HTTP atau HTTPS dari klien itu, yang di kenal sebagai *browser web* dan mengirimkan kembali (reaksi) hasil dalam bentuk situs yang biasanya merupakan dokumen HTML.

Mode operasi dasar *server web* antara lain adalah menunggu untuk koneksi dari *client web* dan setiap permintaan, menempatkan koresponden dokumen pada system file-nya dan mengirimkannya ke browser jika muncul permintaan untuk halaman statis (Simarmata, 2010:69). *Web server* merupakan sebuah perangkat lunak *server* yang berfungsi menerima permintaan dari klien yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman – halaman *web* melalui protocol HTTP atau HTTPS dan bertugas mengelola halaman – halaman *web* dan dokumen – dokumen lainnya (Solichin,2016:6). Maka dari itu,

*web server* merupakan perangkat lunak yang dapat melakukan permintaan pemanggilan alamat melalui *web browser*, dimana *web server* mengirimkan kembali informasi yang diminta melalui *hypertext transfer protocol* (HTTP).

## **2.8 MySQL**

Menurut Raharjo (2011), “MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user”.

## **2.9 Apache HTTP Server**

Menurut Hashemian (2015), Apache adalah salah satu jenis web server yang dapat dijalankan di berbagai sistem operasi, seperti Microsoft Windows, Linux, Unix, Novell Netware serta platform lainnya yang digunakan untuk melayani dan melakukan pengaturan fasilitas web menggunakan sebuah protokol yang dikenal dengan HTTP (*Hypertext Transfer Protocol*). Nama Apache sendiri dipilih sebagai penghormatan terhadap suku Indian Apache yang menggunakan keterampilan dan strategis yang luar biasa dalam peperangan.

Sama seperti web server pada umumnya Apache Web Server juga berfungsi sebagai memperoleh berkas sesuai request pembaca dari web browser. Setelahnya maka Apache akan memproses berkas tersebut dan memberikan output yang diinginkan oleh pembaca. Output yang dikeluarkan sebelumnya tersimpan pada database website. Server Apache memungkinkan pengguna untuk menjalankan berbagai skrip dan aplikasi didalamnya. Dengan begitu, setiap web yang dihosting ke server Apache menjadi dinamis, konten didukung oleh standar HTTP saat ini.

## 2.10 Bootstrap

Bootstrap adalah sebuah library framework CSS yang dibuat khusus untuk bagian pengembangan front-end website. Bootstrap merupakan salah satu framework HTML, CSS dan javascript yang paling populer di kalangan web developer. Pada saat ini hampir semua web developer telah menggunakan bootstrap untuk membuat tampilan front-end menjadi lebih mudah dan sangat cepat. Karena Anda hanya perlu menambahkan *class-class* tertentu untuk misalnya membuat tombol, grid, navigasi dan lainnya.

## 2.11 jQuery

Menurut Bekti (2015), jQuery adalah salah satu jenis web server yang dapat dijalankan di berbagai sistem operasi, seperti Microsoft Windows, Linux, Unix, Novell Netware serta platform lainnya yang digunakan untuk melayani dan melakukan pengaturan fasilitas web menggunakan sebuah protokol yang dikenal dengan HTTP (*Hypertext Transfer Protocol*). Nama Apache sendiri dipilih sebagai penghormatan terhadap suku Indian Apache yang menggunakan keterampilan dan strategis yang luar biasa dalam peperangan.

## 2.12 UML (Unified Modelling Language)


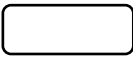
UML merupakan singkatan dari “*Unified Modelling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar



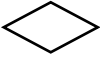
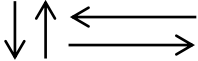
pada visualisasi, perancangan dan juga pendokumentasian sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan blue print software.

### 2.12.1 Activity Diagram

Menurut Sukamto dan Shalahuddin (2014:161) diagram aktivitas atau activity diagram adalah menggambarkan aliran kerja atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan oleh aktor. Nugroho (2010:62) berpendapat bahwa diagram aktivitas atau activity diagram merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. Dengan demikian diagram aktivitas atau activity diagram adalah menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol yang digunakan yaitu :

Tabel 2. 2 Komponen Activity Diagram

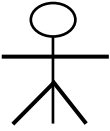
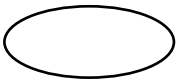

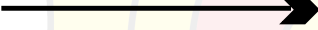


NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing - masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi

3		<i>Initial Node</i>	Bagaimana objek di bentuk atau diawali
4		<i>Activity</i> <i>Final Code</i>	Bagaimana objek di bentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus di ambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

### 2.12.2 Use Case Diagram

Menurut Sukamto dan Shalahuddin (2014:155) berpendapat use case diagram merupakan pemodelan untuk kelakuan (behavior) system informasi yang akan dibuat, use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan yang berhak menggunakan fungsi-fungsi tersebut. Sedangkan menurut Pratama (2014:48) menjelaskan use case diagram merupakan aliran kegiatan dan proses bisnis yang dilakukan oleh pengguna (aktor).


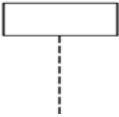
Tabel 2. 3 Komponen Use Case Diagram


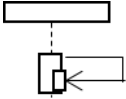
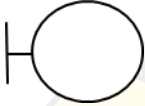

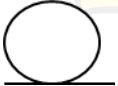
Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi
	Use Case : Abstraksi dan interaksi antara sistem dan actor
	Association : Abstraksi dari penghubung antara actor dengan use case
	Generalization / Generalisasi : Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan use case
<<include>> 	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
<<extend>> 	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

### 2.12.3 Sequence Diagram

Pada jenis pemrograman berbasis objek (object oriented) misalkan dengan bahasa Java, digunakan pemodelan UML. Pada sequence diagram menggambarkan aliran pengiriman pesan yang terjadi di aplikasi, sebagai bentuk interaksi dengan pengguna (user) menurut Pratama (2014:48). Sedangkan Sukamto dan Shalahuddin (2014:165) berpendapat sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Dengan demikian sequence diagram adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object.

Tabel 2. 4 Komponen Sequence Diagram

Simbol	Keterangan
 Aktor	Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary system</i>
 Lifeline	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)

 <p>Message</p>	<p>Menunjukkan aliran informasi atau kendali transaksi antar elemen</p>
 <p><i>Sel-Message</i></p>	<p>Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i>. Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i></p>
 <p>Aktor</p>	<p><i>Boundary</i> adalah objek stereotip yang memodelkan batasan sistem. Biasanya layar <i>user interface</i></p>
 <p><i>Control</i></p>	<p><i>Control</i> adalah objek stereotip yang mengontrol atau mengatur entitas</p>
 <p><i>Entity</i></p>	<p>Merupakan table pada <i>database</i> yang merupakan model penyimpanan data yang menangkap dan menyimpan informasi dalam system</p>