

**LAPORAN PENELITIAN**

**PEMODELAN DAN IMPLEMENTASI JASA LAYANAN DISTRIBUSI  
PAKET *ONLINE SHOP* MENGGUNAKAN *FIRST IN FIRST OUT*  
(FIFO) DAN *TRAVELLING SALESEMAN PROBLEM* (TSP)**



Oleh:  
**Ilka Zufria, M.Kom**  
NIP. 198506042015031006

**LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT  
(LP2M)  
UNIVERSITAS ISLAM NEGERI (UIN) SUMATERA UTARA  
MEDAN  
2018**

## LEMBAR PENGESAHAN

1. a. Judul Penelitian : Pemodelan Dan Implementasi Jasa Layanan Distribusi  
: Paket *Online Shop* Menggunakan *First In First Out*  
(FIFO) Dan *Traveling Salesman Problem* (TSP)
  - b. Kluster Penelitian : Penelitian Pembinaan/Peningkatan Kapasitas
  - c. Bidang Keilmuan : Sains dan Teknologi
  - d. Kategori : Kelompok
- 
2. Ketua Peneliti : Ilka Zufria, M.Kom  
ID Peneliti : 010406850110000
  3. Anggota : Triase, ST, M.Kom  
ID Peneliti : 010805830210000
  4. Unit Kerja : Fakultas Sains dan Teknologi – UIN SU Medan
  5. Waktu Penelitian : 5 s.d 6 bulan 2018
  6. Lokasi Penelitian : JNE Express Katamso (Sukasopan) Medan
  7. Biaya Penelitian : Rp. 15.000.000,-  
(Lima Belas Juta Rupiah)

Medan, 30 Oktober 2018

Disahkan oleh Ketua  
Lembaga Penelitian dan Pengabdian  
kepada Masyarakat (LP2M)  
UIN Sumatera Utara Medan

Peneliti,  
Ketua



Prof. Dr. Pagar, M.Ag.  
NIP. 195812311988031016



Ilka Zufria, M.Kom  
NIP. 198506042015031006

## REKOMENDASI

Setelah membaca dan menelaah hasil penelitian yang berjudul “Pemodelan dan Implementasi Jasa Layanan Distribusi Paket *Online Shop* menggunakan *First In First Out*(FIFO) dan *Travelling Salesman Problem* (TSP)”. Yang dilakukan oleh Ilka Zufria, M.Kom maka saya berkesimpulan bahwa hasil penelitian ini dapat diterima sebagai karya tulis berupa hasil penelitian. Demikianlah rekomendasi diberikan kepada yang bersangkutan untuk dapat dipergunakan sebagaimana mestinya.

Medan, 25 Oktober 2018

Konsultan



M. Irwan Padli Nst, MM, M.Kom

NIP. 19750213 200604 1 003

## **SURAT PERNYATAAN BEBAS PLAGIASI**

Yang bertanda tangan di bawah ini :

Nama : Ilka Zufria, M.Kom  
Jabatan : Dosen / Asisten Ahli  
Unit Kerja : Fakultas Sains dan Teknologi  
                  : UIN Sumatera Medan  
Alamat : Jalan IAIN No. 1 Medan

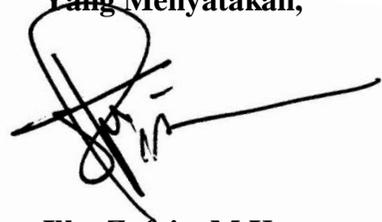
dengan ini menyatakan bahwa :

1. Judul penelitian “ Pemodelan dan Implementasi Jasa Layanan Distribusi Paket *Online Shop* Menggunakan *First In First Out (FIFO)* Dan *Traveling Salesman Problem (TSP)*” merupakan karya orisinal saya.
2. Jika dikemudian hari ditemukan fakta bahwa judul, hasil atau bagian dari laporan penelitian saya merupakan karya orang lain dan/atau plagiasi, maka saya akan bertanggung jawab untuk mengembalikan 100% dana hibah penelitian yang telah saya terima, dan siap mendapatkan sanksi sesuai ketentuan yang berlaku.

Demikian pernyataan ini dibuat untuk digunakan sebagaimana mestinya.

Medan, 30 Oktober 2018

**Yang Menyatakan,**



**Ilka Zufria, M.Kom**

**NIP. 198506042015031006**

## ABSTRAK

*Online shop* dengan jasa layanan distribusi merupakan dua jenis perusahaan yang dalam menjalankan roda usahanya saling bergantung. *Online shop* menyediakan barang yang dibutuhkan konsumen dan jasa layanan distribusi menerima barang yang dikirim oleh online shop untuk didistribusikan kepada konsumen. Dalam hubungan kerja keduanya, agar dapat bersaing dengan perusahaan lain memerlukan sebuah sistem dan strategi yang berkualitas. Metode FIFO merupakan singkatan dari First in first out adalah bagian dari metode antrian dalam pengelolaan data. Metode ini sangat baik digunakan mengatasi antrian paket yang masuk ke perusahaan jasa layanan distribusi dengan cara menentukan barang yang masuk pertama sekali maka yang akan pertama kali diberikan kepada kurir untuk rencana pendistribusian. (*Travelling Salesman Problem*) TSP dapat digunakan untuk menghitung jarak rute terdekat kurir dengan konsumen yang dituju. TSP dapat dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah alamat konsumen terhadap sejumlah n alamat dimana alamat - alamat yang ada hanya dikunjungi sekali oleh kurir. Penggunaan kedua metode ini akan menghasilkan sistem komputerisasi yang menyebabkan perusahaan jasa layanan distribusi dapat mengoptimalkan kinerja pendistribusian paket oleh kurir dari konsumen online shop sehingga dapat meningkatkan keuntungan dalam waktu dan ekonomi.

**Keywords:** *Online Shop, Layanan Distribusi, First In First Out, Travelling Salesman Problem*

## ABSTRACT

*Online shop with distribution services are two types of companies that run their business interdependently. Online shop provides goods needed by consumers and distribution services receive good sent by online shop for distribution to consumers. In their working relationship, in order to compete with other companies requires a quality system and strategy. The FIFO method stands for First in first out is part of the queue method in data management. This method is very good to be used to overcome the queue of packets that enter the distribution service company by determining the items that enter the first time, which will be first given to the courier for the distribution plan. TSP (Traveling Salesman Problem) can be used to calculate the distance of the nearest route of the courier to the intended customer. TSP can be stated as a problem in finding a minimum distance of a customer's address to a number of  $n$  addresses where the addresses are only visited once by the courier. The use of these two methods will produce a computerized system that causes distribution service companies to optimize the performance of package distribution by couriers from online shop consumers so as to increase profits in time and economy.*

**Keywords:** *Online Shop, Distribution Services, First In First Out, Travelling Salesman Problem*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirobbil Alamiin, segala puji bagi Allah SWT. Atas berkat rahmat dan karuniaNya, saya dan tim penelitian dapat menyelesaikan penelitian ini dengan judul “**Pemodelan Dan Implementasi Jasa Layanan Distribusi Paket Online Shop Menggunakan First In First Out (Fifo) Dan Traveling Salesman Problem (Tsp)**”. Sholawat dan salam senantiasa dipanjatkan kepada baginda Muhammad SAW beserta kerabat, sahabat, para pengikutnya sampai akhir zaman, adalah sosok yang telah membawa manusia dan seisi alam dari kegelapan ke cahaya sehingga kita menjadi manusia beriman, berilmu, dan tetap beramal shaleh agar menjadi manusia yang berakhlak mulia.

Penulisan laporan ini bertujuan untuk melengkapi persyaratan luaran penelitian. laporan ini juga diharapkan dapat menambah wawasan ilmu pengetahuan, khususnya bidang sistem informasi dalam instalasi nilai-nilai Islam yang terpadu dalam proses pembelajaran di lingkungan Universitas Islam Negeri Sumatera Utara.

Dalam penulisan laporan ini, saya sangat menyadari bahwa masih banyak kekurangan yang perlu perbaikan dan penyempurnaan, sumbangan pemikiran yang membangun sangat kami harapkan dari rekan-rekan sejawat terutama dari dosen-dosen senior. Semoga laporan penelitian ini dapat diperkaya melalui evaluasi terus menerus. Terimakasih kepada anggota peneliti dan tim penelitian yang sudah fokus dalam penyelesaian laporan ini dan pastinya sangat berperan dalam proses penelitian dari tahap awal hingga akhir. Dan kepada semua pihak yang tidak bisa disebutkan satu persatu, ribuan terimakasih diucapkan atas bantuan yang kami terima selama proses penelitian ini.

Medan, Oktober 2018  
Penulis

**Ilka Zufria, M.Kom**  
NIP. 198506042015031006

## DAFTAR ISI

<b>LEMBAR PENGESAHAN DAN REKOMENDASI.....</b>	<b>ii</b>
<b>SURAT PERNYATAAN BEBAS PLAGIASI .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>ABSTRACT.....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI .....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
A. Latar Belakang .....	1
B. Rumusan Permasalahan .....	3
C. Tujuan Penelitian .....	3
D. Signifikansi .....	4
<b>BAB II KAJIAN PUSTAKA.....</b>	<b>5</b>
A. Pengertian FIFO First in First out .....	5
B. TSP (Travelling Salesman Problem).....	5
C. Algoritma Dijkstra .....	6
D. UML (Unified Modelling Language).....	7
<b>BAB III METODE PENELITIAN.....</b>	<b>17</b>
A. Perencanaan Sistem.....	17
B. Analisa Kebutuhan Sistem dan Informasi.....	18
C. Pemodelan dan Perancangan Sistem.....	18
D. Testing Prototype Sistem .....	19
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>21</b>
A. Analisis Kebutuhan Sistem dan Informasi.....	21
1. Analisis Kebutuhan Data.....	21
2. Analisa Kebutuhan Sistem .....	40
3. Testing dan Implementasi .....	60
<b>BAB V PENUTUP .....</b>	<b>67</b>

A. KESIMPULAN .....	67
B. PENUTUP.....	67
<b>DAFTAR REFERENSI.....</b>	<b>69</b>

## DAFTAR GAMBAR

Gambar 3. 1 Diagram Alur Metode Penelitian .....	20
Gambar 4. 1 Lokasi Konsumen Medan Johor Dari Google Maps.....	25
Gambar 4. 2 Peta Lokasi Penyebaran Alamat Konsumen Wilayah Medan Johor.....	25
Gambar 4. 3 Penyebaran Verteks Konsumen Wilayah Medan Johor.....	26
Gambar 4. 4 Graf Tanpa Bobot Untuk Penyebaran Konsumen Wilayah Medan Johor .....	26
Gambar 4. 5 Graf Berbobot Untuk Wilayah Medan Johor.....	27
Gambar 4. 6 Node Terpilih Iterasi Ke-1 Pada Wilayah Medan Johor.....	28
Gambar 4. 7 Node Terpilih Iterasi Ke-2 Pada Wilayah Medan Johor.....	29
Gambar 4. 8 Node Terpilih Iterasi Ke-3 Pada Wilayah Medan Johor.....	30
Gambar 4. 9 Node Terpilih Iterasi Ke-4 Pada Wilayah Medan Johor.....	30
Gambar 4. 10 Node Terpilih Iterasi Ke-5 Pada Wilayah Medan Johor....	31
Gambar 4. 11 Node Terpilih Iterasi Ke-6 Pada Wilayah Medan Johor....	32
Gambar 4. 12 Node Terpilih Iterasi Ke-7 Pada Wilayah Medan Johor....	32
Gambar 4. 13 Node Terpilih Iterasi Ke-8 Pada Wilayah Medan Johor....	33
Gambar 4. 14 Node Terpilih Iterasi Ke-9 Pada Wilayah Medan Johor....	34
Gambar 4. 15 Node Terpilih Iterasi Ke-10 Pada Wilayah Medan Johor..	34
Gambar 4. 16 Node Terpilih Iterasi Ke-11 Pada Wilayah Medan Johor..	35
Gambar 4. 17 Node Terpilih Iterasi Ke-12 Pada Wilayah Medan Johor..	36
Gambar 4. 18 Node Terpilih Iterasi Ke-13 Pada Wilayah Medan Johor..	36
Gambar 4. 19 Node Terpilih Iterasi Ke-14 Pada Wilayah Medan Johor..	37
Gambar 4. 20 Node Terpilih Iterasi Ke-15 Pada Wilayah Medan Johor..	38
Gambar 4. 21 Node Terpilih Iterasi Ke-16 Pada Wilayah Medan Johor..	38
Gambar 4. 22 Pohon Rentang Minimum Pada Wilayah Medan Johor Menggunakan Algoritma Dijkstra .....	39
Gambar 4. 23 Flowchart Penerapan FIFO .....	40
Gambar 4. 24 Flowchart penerapan algoritma Dijkstra.....	41
Gambar 4. 25 Tabel relasi yang dibangun di dalam Database.....	42
Gambar 4. 26 Rancangan struktur tabel login .....	42
Gambar 4. 27 Rancangan struktur tabel manifest.....	43

Gambar 4. 28 Rancangan struktur tabel Bag on Manifest .....	43
Gambar 4. 29 Rancangan struktur tabel kurir .....	43
Gambar 4. 30 Rancangan struktur tabel warehouse.....	43
Gambar 4. 31 Rancangan struktur tabel Item on Bag .....	44
Gambar 4. 32 Rancangan struktur tabel Item on Bag .....	44
Gambar 4. 33 Use Case Diagram Sistem Layanan Distribusi Paket .....	45
Gambar 4. 34 Activity Diagram Proses Login .....	46
Gambar 4. 35 Activity Diagram Proses Manifest.....	47
Gambar 4. 36 Activity Diagram Proses Bag on Manifest .....	48
Gambar 4. 37 Activity Diagram Proses Item on Bag .....	49
Gambar 4. 38 Activity Diagram Proses Detail consignment Note .....	50
Gambar 4. 39 Activity Diagram Proses JNE Courier Update .....	51
Gambar 4. 40 Activity Diagram Proses JNE Warehouse Courier .....	52
Gambar 4. 41 Activity Diagram Routing Area Tambah.....	53
Gambar 4. 42 Activity Diagram Routing Area Lintasan .....	54
Gambar 4. 43 HIPO Chart Menu Utama .....	55
Gambar 4. 44 Form Input Detail Consignment .....	56
Gambar 4. 45 Form Input Data Courier.....	56
Gambar 4. 46 Form Input Data Bag .....	57
Gambar 4. 47 Form Input Routing Area.....	57
Gambar 4. 48 Output manifest.....	58
Gambar 4. 49 Output courier warehouse .....	58
Gambar 4. 50 Output Bag on Manifest.....	59
Gambar 4. 51 Output detil consignment note .....	59
Gambar 4. 52 Tampilan halaman login .....	60
Gambar 4. 53 Tampilan halaman administrator .....	61
Gambar 4. 54 Tampilan halaman manifest.....	61
Gambar 4. 55 Tampilan halaman Input Bag on manifest .....	62
Gambar 4. 56 Tampilan halaman output Bag on manifest .....	62
Gambar 4. 57 Tampilan list item on bag .....	63
Gambar 4. 58 Tampilan list dan kelola courier warehouse .....	63
Gambar 4. 59 Tampilan kelola Detail Consignment .....	64
Gambar 4. 60 Tampilan kelola Data Kurir .....	64
Gambar 4. 61 Tampilan Routing Area Note to Note .....	65
Gambar 4. 62 Tampilan Routing Area Distance.....	66

Gambar 4. 63 Tampilan Kelola Warehouse..... 66

## DAFTAR TABEL

Tabel 2. 1 Simbol Usecase Diagram.....	10
Tabel 2. 2 Simbol – Simbol Class Diagram.....	13
Tabel 2. 3 Simbol-Simbol Activity Diagram.....	15
Tabel 4. 1 Data Barang JNE Kantor Perwakilan Suka Sopan Medan Untuk Wilayah Medan Johor.....	21
Tabel 4. 2 Data Konsumen JNE Express Wilayah Medan Johor .....	24
Tabel 4. 3 Jarak Antar Titik Untuk Wilayah Medan Johor .....	26
Tabel 4. 4 Hasil Iterasi Ke-1 Pada Wilayah Medan Johor.....	28
Tabel 4. 5 Hasil Iterasi Ke-2 Pada Wilayah Medan Johor.....	29
Tabel 4. 6 Hasil Iterasi Ke-3 Pada Wilayah Medan Johor.....	29
Tabel 4. 7 Hasil Iterasi Ke-4 Pada Wilayah Medan Johor.....	30
Tabel 4. 8 Hasil Iterasi Ke-5 Pada Wilayah Medan Johor.....	31
Tabel 4. 9 Hasil Iterasi Ke-6 Pada Wilayah Medan Johor.....	31
Tabel 4. 10 Hasil Iterasi Ke-7 Pada Wilayah Medan Johor.....	32
Tabel 4. 11 Hasil Iterasi Ke-8 Pada Wilayah Medan Johor.....	33
Tabel 4. 12 Hasil Iterasi Ke-9 Pada Wilayah Medan Johor.....	33
Tabel 4. 13 Hasil Iterasi Ke-10 Pada Wilayah Medan Johor.....	34
Tabel 4. 14 Hasil Iterasi Ke-11 Pada Wilayah Medan Johor.....	35
Tabel 4. 15 Hasil Iterasi Ke-12 Pada Wilayah Medan Johor.....	35
Tabel 4. 16 Hasil Iterasi Ke-13 Pada Wilayah Medan Johor.....	36
Tabel 4. 17 Hasil Iterasi Ke-14 Pada Wilayah Medan Johor.....	37
Tabel 4. 18 Hasil Iterasi Ke-15 Pada Wilayah Medan Johor.....	37
Tabel 4. 19 Hasil Iterasi Ke-16 Pada Wilayah Medan Johor.....	38
Tabel 4. 20 Lintasan Terpendek dari Simpul Awal Ke Simpul Lainnya Pada Wilayah Medan Johor Menggunakan Algoritma Dijkstra .....	39

# BAB I

## PENDAHULUAN

### A. Latar Belakang

*Online shop*, adalah suatu proses pembelian barang atau jasa dari mereka perusahaan yang menjual barang atau jasa melalui *internet* dimana antara penjual dan pembeli tidak pernah bertemu atau melakukan kontak secara fisik yang dimana barang yang diperjualbelikan ditawarkan melalui display dengan gambar yang ada di suatu website atau toko maya. Setelahnya pembeli dapat memilih barang yang diinginkan untuk kemudian melakukan pembayaran kepada penjual melalui rekening bank yang bersangkutan. Setelah proses pembayaran di terima, kewajiban penjual adalah mengirim barang pesanan pembeli ke alamat tujuan..

Jasa layanan distribusi adalah organisasi-organisasi yang saling tergantung yang tercakup dalam proses yang membuat produk atau jasa menjadi tersedia untuk digunakan atau dikonsumsi.

*Online shop* dengan jasa layanan distribusi merupakan dua jenis perusahaan yang dalam menjalankan roda usahanya saling bergantung. *Online shop* menyediakan barang yang dibutuhkan konsumen dan jasa layanan distribusi menerima barang yang dikirim oleh *online shop* untuk didistribusikan kepada konsumen. Dalam hubungan kerja keduanya, agar dapat bersaing dengan perusahaan lain memerlukan sebuah sistem dan strategi yang berkualitas.

Perusahaan jasa layanan distribusi sekarang ini sangat sibuk sekali menerima barang yang dikirim oleh *Online shop* karena masyarakat sekarang sangat minat sekali berbelanja *online*, sehingga barang yang masuk ke gudang jasa layanan distribusi tertumpuk (*overload*). Hal tersebut memungkinkan membuat pegawai yang bekerja di jasa layanan distribusi mengalami *human error* maka barang yang tertumpuk digudang tidak tersortir dan keluar sesuai berdasarkan urutan barang yang telah sampai dahulu. Metode FIFO merupakan singkatan dari *First in first out* adalah bagian dari metode antrian dalam pengelolaan data. Metode ini sangat baik digunakan mengatasi antrian paket yang masuk ke perusahaan jasa layanan distribusi dengan cara menentukan barang yang masuk pertama sekali

maka yang akan pertama kali diberikan kepada kurir untuk rencana pendistribusian.

Barang yang telah diterima kurir selanjutnya harus didistribusikan dengan cepat kepada konsumen *onlin shop* karena apabila tidak diserahkan tepat waktu maka akan membuat konsumen kecewa. Masalah keepatan waktu ini disebabkan jalanan yang macat serta rute jalan yang banyak untuk menuju konsumen. Untuk mengatasi hal tersebut seorang kurir dalam mendistribusikan paket memerlukan metode pendistribusian yang dikenal dengan TSP (*Travelling Salesman Problem*). TSP dapat digunakan untuk menghitung jarak rute terdekat kurir dengan konsumen yang dituju. TSP dapat dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah alamat konsumen terhadap sejumlah  $n$  alamat dimana alamat - alamat yang ada hanya dikunjungi sekali oleh kurir. Diharapkan dengan metode ini kurir tidak kesulitan dalam menentukan tujuan pendistribusian paket. Penggunaan kedua metode ini akan menghasilkan sistem komputerisasi yang menyebabkan perusahaan jasa layanan distribusi dapat mengoptimalkan kinerja pendistribusian paket oleh kurir dari konsumen online shop sehingga dapat meningkatkan keuntungan dalam waktu dan ekonomi.

Penelitian ini dikembangkan berdasarkan dari penelitian terdahulu yang pernah dibuat oleh Oktaviany<sup>1</sup> yang berjudul **Perancangan Sistem Persediaan Barang Menggunakan Metode FIFO Pada PT. Panjunan Sukaraja Sukabumi**. Penelitian tersebut membahas mengenai pemeriksaan persediaan barang dengan menerapkan metode FIFO melalui transaksi keluar masuknya barang, maka diharapkan tidak akan terjadi penumpukan barang yang terlalu lama dalam gudang atau pembelian barang yang terlalu banyak dan tidak sesuai dengan kebutuhan, dengan asumsi barang pada persediaan pertama dibeli akan dijual terlebih dahulu.

Sedangkan penelitian yang dilakukan oleh Maisak<sup>2</sup> yang berjudul **Analisis Dan Perancangan Sistem Informasi Persediaan Barang**

---

<sup>1</sup> Oktapiani, Renny. 2016. *Perancangan Sistem Persediaan Barang Menggunakan Metode FIFO Pada PT. Panjunan Sukaraja Sukabumi*. Konferensi Nasional Ilmu Sosial & Teknologi (KNiST) Maret 2016

<sup>2</sup> Maisak, Despita. 2017. *Analisis Dan Perancangan Sistem Informasi Persediaan Barang Menggunakan Metode FIFO Pada PT. Shukaku*. MEDIASISFO Vol. 11, No. 2, Oktober 2017

**Menggunakan Metode FIFO Pada PT. Shukaku.** Pada penelitian tersebut membahas pencatatan persediaan barang secara manual sering kali proses pencatatan persediaan yang dilakukan mengakibatkan selisih dari data jumlah stok barang dengan jumlah barang fisik yang ada setiap bulannya, mengakibatkan kerugian yang harus ditanggung perusahaan,

Berbeda dengan penelitian di atas yang dilakukan oleh Fitria<sup>3</sup> yang berjudul Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan. Pada penelitian ini algoritma dijkstra dipakai untuk menghitung jarak terdekat dari suatu kota ke kota lainnya pada sumatera bagian selatan.

## **B. Rumusan Permasalahan**

Berdasarkan latar belakang di atas maka dapat dirumuskan permasalahan penelitian sebagai berikut:

- a. Bagaimanakah penggunaan metode FIFO *First in First out*, dalam memecahkan masalah penentuan paket yang akan didistribusikan ke konsumen?
- b. Bagaimanakah *Travelling Salesman Problem* (TSP) dengan metode Dijkstra dapat digunakan oleh kurir untuk mendistribusikan paket ke alamat konsumen dengan hanya sekali kunjungan saja?
- c. Bagaimanakah merancang sistem aplikasi pendistribusian paket yang akan diimplementasikan menggunakan bahasa pemrograman vb.net didukung dengan aplikasi basis data mysql

## **C. Tujuan Penelitian**

Penelitian ini bertujuan untuk:

- a. Mengetahui penggunaan FIFO *First in First out* dalam memecahkan masalah penentuan paket yang akan didistribusikan ke konsumen
- b. Mengetahui *Travelling Salesman Problem* (TSP) dengan metode Dijkstra dapat digunakan oleh kurir untuk mendistribusikan paket ke alamat konsumen dengan hanya sekali kunjungan saja

---

<sup>3</sup> Fitria dan Apri.2013. *Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan*. Jurnal Sistem Informasi (JSI), VOL. 5, NO. 2, Oktober 2013

- c. Merancang sistem pendistribusian paket sehingga memudahkan untuk implementasi sistem

#### **D. Signifikansi**

Penelitian ini dianggap penting karena diharapkan akan memberi kontribusi sebagai berikut:

- a. Tersedianya sistem distribusi berbasis komputer yang dapat membantu perusahaan jasa layanan distribusi paket dalam menentukan paket yang akan didistribusikan
- b. Mempermudah para kurir yang bekerja di perusahaan jasa Layanan distribusi dalam mendistribusikan paket yang akan dikirimkan ke alamat konsumen.
- c. *Outcome* berupa publikasi karya ilmiah hasil penelitian di prosiding international terindex scopus

## BAB II

### KAJIAN PUSTAKA

#### A. Pengertian FIFO First in First out

Dalam menentukan metode FIFO untuk menentukan harga pokok digunakan asumsi atas arus biaya dimana urutan biaya terjadi adalah yang pertama masuk atau diperoleh, yang pertama keluar atau dijual. Dengan demikian persediaan yang tinggal, dianggap akan dinilai dengan menggunakan biaya atau harga pokok (*cost*) yang paling baru. Metode FIFO Mengasumsikan barang dalam persediaan yang pertama dibeli akan dijual atau digunakan terlebih dulu sehingga yang yang tertinggal dalam persediaan akhir adalah yang dibeli atau diproduksi kemudian. Metode FIFO adalah barang yang pertama masuk adalah barang yang akan pertama keluar. ”. <sup>4</sup>(Oktapiani, dkk; 2016)

#### B. TSP (Travelling Salesman Problem)

TSP termasuk kelas NP-*Hard problem* dan tidak dapat diselesaikan secara *Polynomial computation time* dengan algoritma eksak. Bila diselesaikan secara eksak waktu komputasi yang diperlukan akan meningkat secara eksponensial seiring bertambah besarnya masalah. TSP dapat dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah tour tertutup terhadap sejumlah  $n$  kota dimana kota-kota yang ada hanya dikunjungi sekali. TSP direpresentasikan dengan menggunakan sebuah graph lengkap dan berbobot  $G = (V, E)$  dengan  $V$  himpunan vertek yang merepresentasikan himpunan titik - titik, dan  $E$  adalah himpunan dari edge. Setiap edge  $(r,s) \in E$  adalah nilai (jarak)  $d_{rs}$  yang merupakan jarak dari kota  $r$  ke kota  $s$ , dengan  $(r,s) \in V$ . Dalam TSP simetrik (jarak dari kota  $r$  ke titik  $s$  sama dengan jarak dari titik  $s$  ke titik  $r$ ),  $d_{rs} = d_{sr}$  untuk semua edge  $(r,s) \in E$ . Misalkan terdapat  $n$  buah titik maka graph tersebut memiliki

---

<sup>4</sup> Oktapiani, Renny. 2016. *Perancangan Sistem Persediaan Barang Menggunakan Metode FIFO Pada PT. Panjuran Sukaraja Sukabumi*. Konferensi Nasional Ilmu Sosial & Teknologi (KNiST) Maret 2016

(  $n!$  (  $n-2$  !2!) buah edge, sesuai dengan rumus kombinasi, dan juga memiliki  $n-1$  ! 2 buah tour yang mungkin. (<sup>5</sup>Wardhani;2014)

### C. Algoritma Dijkstra

Algoritma Dijkstra, ditemukan oleh ilmuwan komputer asal Belanda Edsger Dijkstra pada tahun 1959, Algoritma dijkstra adalah algoritma pencarian grafik yang memecahkan masalah jalur terpendek dari satu sumber dengan nilai jalur yang dihasilkan tidak negatif, dan menghasilkan pohon jalur terpendek. Algoritma ini sering digunakan dalam pencarian rute. Untuk sumber simpul (node) tertentu dalam grafik, algoritma menghasilkan jalur dengan biaya terendah (yaitu lintasan terpendek) antara *vertex* dan *vertex* lainnya.

Langkah-langkah penentuan lintasan terpendek dari graf G dengan n-buah simpul dengan simpul awal a menggunakan algoritma dijkstra adalah sebagai berikut:

1. Langkah 0 (inisialisasi) :  $s_i = 0$  dan  $d_i = m_{ai}$  untuk  $i = 1, 2, \dots, n$
2. Langkah 1 : isi  $s_a$  dengan 1 dan isi  $d_a$  dengan  $\infty$
3. Langkah 2: untuk setiap  $s_i = 0$  dan  $i=1, 2, \dots, n$ , pilih  $d_j = \min \{d_1, d_2, \dots, d_n\}$  lalu isi  $s_j$  dengan 1 dan perbaharui  $d_i$ , dengan :  $d_i$  (baru) =  $\min \{d_i$  (lama),  $d_j + m_{ji}\}$ . Pada lintasan tambahkan simpul j sebagai simpul terpilih untuk lintasan selanjutnya.
4. Langkah 3: mengulangi langkah 2 sampai  $s_j=1$ , untuk  $j=1, 2, \dots, n$
5. Membuat himpunan simpul berdasarkan urutan yang diperoleh yang merupakan lintasan terpendek dengan bobot  $d_i$ ” (Salaki. T. D, 2011).

#### Pseudokode Dijkstra

*procedure* dijkstra ( $w, a, z, L$ )

$L(a) := 0$

$S := \{ \}$

*for* semua vertex  $x \neq a$  *do*

$L(x) := \infty$

---

<sup>5</sup> Wardhani, Nurmalianti. 2014. *Optimasi Traveling Salesman Problem (TSP) Menggunakan Algoritma Semut*. JURNAL IT STMIK HANDAYANI, Volume 15, Desember 2014

```

T := himpunan semua vertex
while z (T do
begin
pilih v ( T dengan minimum L (v)
T := T - {v}
S := S union {v}
for setiap x(T di samping v
do
L(x) := min {L(x), L (v) + w (v, w)}
end
end dijkstra (Dewi. L. J. E, 2010).

```

#### D. UML (Unified Modelling Language)

UML (*Unified Modelling Language*) adalah sebuah bahasa yang telah menjadi standar dalam industri visualisasi, merancang dan mendokumentasikan sistem peranti lunak (Yuni Sugiarti, 2013:34<sup>6</sup>). Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti *class*, sistem operasi dan jaringan apapun serta ditulis dalam pemrograman apapun.

UML (*Unified Modelling Language*) bisa digunakan untuk :

- Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum.
- Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum.
- Menggambarkan representasi struktur statik.
- Membuat model *behavior* yang menggambarkan kebiasaan atau sifat sebuah sistem
- Menyatakan arsitektur implementasi fisik.
- Menyampaikan atau memperluas *functionality* dengan *stereotypes*.

UML (*Unified Modelling Language*) merupakan salah satu alat bantu yang handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print*. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. UML mempunyai banyak diagram yang dapat mengakomodasi berbagai sudut

---

<sup>6</sup> Yuni Sugiarti, (2013). “Analisis Dan Perancangan UML (*Unified Modeling Language*) Generated VB. 6.” Bandung : Graha Ilmu

pandang dari suatu perangkat lunak yang akan dibangun. Diagram-diagram tersebut digunakan untuk:

1. Mengkomunikasi ide.
2. Melahirkan ide-ide baru dan peluang-peluang baru.
3. Menguji ide dan membuat prediksi memahami struktur dan relasi-relasinya.

### ***1. Usecase Diagram***

Dalam membuat sebuah sistem, langkah awal yang perlu dilakukan adalah menentukan kebutuhan. Kebutuhan yang dimaksud adalah kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional adalah kebutuhan pengguna dan stakeholder yang akan dimiliki oleh sistem, dimana kebutuhan ini akan digunakan oleh pengguna dan stakeholder (Yuni Sigiarti, 2013 :41<sup>6</sup>). Kebutuhan nonfungsional adalah kebutuhan yang memperhatikan hal-hal berikut yaitu performansi, kebutuhan dalam menggunakan sistem, kehandalan sistem, keamanan sistem, keuangan, legalitas, dan operasional (Yuni Sigiarti, 2013 :41<sup>6</sup>).

Kebutuhan fungsional akan digambarkan melalui sebuah diagram yang dinamakan diagram usecase. (Yuni Sigiarti, 2013 :41<sup>6</sup>). UML dengan *usecase* berkaitan dengan interaksi antara sistem dan aktor eksternal. (Jakimi dan M. El Koutbi, 2009<sup>7</sup>). *Usecase* merupakan pemodelan untuk menggambarkan perilaku (*behavior*) sistem yang akan dibuat.

Diagram *usecase* menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Pendekatan *usecase* berbeda dengan metode tradisional. Pendekatan tradisional berfokus pada bagaimana memecahkan persoalan besar menjadi persoalan-persoalan yang lebih kecil, sementara pendekatan *usecase* berfokus pada apa yang pemakai harapkan dari sistem. (Sholiq, 2006 : 61<sup>8</sup>).

Didalam diagram *usecase* terdapat aktor. Langkah-langkah menemukan aktor dalam *usecase* adalah sebagai berikut :

1. Gunakan kata tanya siapa yang akan menggunakan sistem ?
2. Apakah sistem tersebut akan memberikan nilai bagi aktor ?

Jika sudah berhasil menemukan aktor, maka untuk menemukan *usecase* akan lebih mudah dilakukan. Sebuah *usecase* harus mendefinisikan

---

<sup>7</sup> Jakimi dan M. L Khoutbi, (2009). “*An Object-Oriented Approach to UML Scenarios Engineering and Code Generation*”

<sup>8</sup> Sholiq, (2006). “*Pemodelan Sistem Informasi Berorientasi Objek Dengan UML.*” Bandung : Graha Ilmu.hal 61

sebuah pekerjaan, dimana pekerjaan tersebut akan memberikan nilai yang bermanfaat bagi aktor.

Untuk menentukan *usecase* mulailah dari sudut pandang aktor, misalnya dengan bertanya :

1. Informasi apa sajakah yang akan didapatkan aktor dari sistem ?
2. Apakah ada kejadian dari sistem yang perlu diberitahukan ke aktor ?

Sedangkan dari sudut pandang sistem, misalnya dengan pertanyaan sebagai berikut :

1. Apakah ada informasi yang perlu disimpan atau diambil dari sistem ?
2. Apakah ada informasi yang harus dimasukkan oleh aktor ?

Setiap *usecase* harus dijelaskan alur prosesnya melalui sebuah deskripsi *usecase* atau skenario *usecase*. Deskripsi *usecase* berisi :

1. Nama *usecase* yaitu penamaan *usecase* yang menggunakan kata kerja.
2. Deskripsi yaitu penjelasan mengenai tujuan *usecase* dan nilai yang akan didapatkan oleh aktor.

Tidak semua aktor adalah manusia, bisa saja sistem lain yang berinteraksi dengan sistem yang kita buat. Sistem dibangun untuk menyediakan kebutuhan bagi aktor.

1. Kondisi sebelum (*post-condition*) yaitu kondisi-kondisi yang sudah dipenuhi ketika *usecase* sudah dilaksanakan alur dasar (*basic flow*) yaitu alur yang menceritakan jika semua aksi yang dilakukan adalah benar atau proses yang harusnya terjadi.
2. Alur alternatif (*alternatif flow*) yaitu alur yang menceritakan aksi alternatif, yang berbeda dari alur dasar.

Diagram *usecase* adalah sebuah diagram yang menjelaskan apa yang harus dilakukan oleh sistem pada level konseptual sehingga kita akan memahami apakah keputusan yang diambil oleh sistem adalah benar atau tidak. (Yuni Sigiarti, 2013 :45<sup>9</sup>).

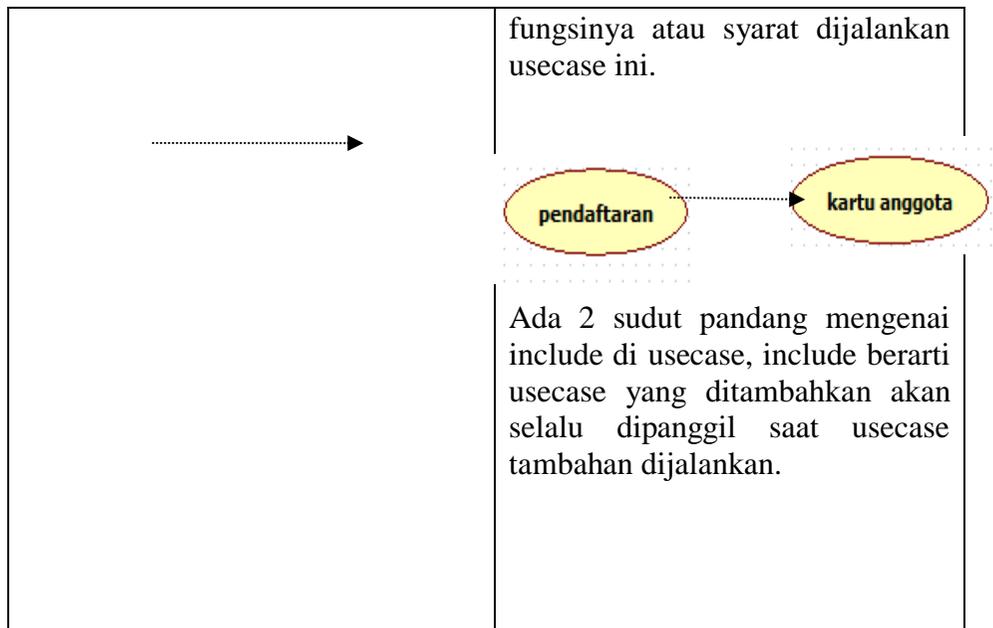
Dari beberapa penjealasan *usecase* dapat ditarik kesimpulan bahwa *usecase* menggambarkan perilaku sistem yang akan dibuat dengan interaksi satu atau lebih aktor dalam sistem yang dibuat. *Usecase* memudahkan dalam hal pemodelan aliran sistem yang akan dibuat.

---

<sup>9</sup> Yuni Sugiarti, (2013). “Analisis Dan Perancangan UML (Unified Modeling Language) Generated VB. 6.” Bandung : Graha Ilmu hal.45

Tabel 2. 1 Simbol Usecase Diagram

Simbol	Deskripsi
	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja.</p>
	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat sendiri. Simbol dari aktor adalah gambar orang, tetapi aktor bukan berarti harus orang. Biasanya diawali dengan menggunakan kata benda.</p>
<p><u>Asosiasi/ association</u></p>	<p>Komunikasi antara aktor dan usecase yang berpartisipasi pada usecase atau usecase memiliki interaksi dengan aktor.</p>
<p>Extend</p>  <p>&lt;&lt;extend&gt;&gt;</p>	<p>Relasi usecase tambahan ke sebuah usecase dimana usecase yang ditambahkan dapat berdiri sendiri.</p>
<p>Include</p> <p>&lt;&lt;include&gt;&gt;</p>	<p>Relasi usecase tambahan ke sebuah usecase dimana usecase yang ditambahkan memerlukan usecase ini untuk menjalankan</p>



## 2. Class Diagram

*Class Diagram* adalah menunjukkan interaksi antara *class* dalam sistem (Sholiq, 2006:13<sup>10</sup>). *Class Diagram* membantu pengembang mendapatkan struktur sistem sebelum menuliskan kode program, membantu untuk memastikan bahwa sistem adalah rancangan terbaik. (Sholiq, 2006 : 102). *Class* adalah sebuah kategori yang membungkus informasi dan perilaku. Secara tradisional, sistem dibangun dengan ide dasar bahwa akan menyimpan informasi pada sisi basis data dan perilaku pengolahannya pada sisi aplikasi.

Hal terpenting sebelum membuat *Class Diagram* meruapakan cara terbaik dalam menemukan *class-class* adalah dimulai dari memperhatikan aliran kejadian (*flow of event*) dari suatu *usecase*. Dengan melakukan penyeleksian kata benda dalam aliran kejadian, dapat ditemukan *class-class* dalam sistem.

Cara kedua adalah dengan menguji objek-objek dalam diagram sekuensial atau diagram kolaborasi. Ada dua cara yang bisa dilakukan berkaitan dengan urutan pendefinisian antar *class-class* dalam diagram *class* dan diagram sekuensial. Kedua cara tersebut adalah

---

<sup>10</sup> Sholiq, (2006). “Pemodelan Sistem Informasi Berorientasi Objek Dengan UML.” Bandung : Graha Ilmu.hal 13 & 102

1. Dengan membuat diagram sekuensial lebih dulu, kemudian melanjutkannya dengan membuat diagram *class*.
2. Dengan menemukan *class-class* dan membuat diagram *class* terlebih dahulu, relasinya untuk membuat diagram sekuensial.

Kedua pendekatan di atas, mempunyai keuntungan dan kerugian. Biasanya, kegiatan dan keputusan desain dilakukan dengan membuat diagram sekuensial terlebih dahulu, kemudian dilanjutkan membuat diagram *class*.

Salah satu keuntungan dengan membuat diagram sekuensial terlebih dulu adalah pengembang perangkat lunak lebih berhati-hati dengan menguji langkah demi langkah objek apa saja yang diperlukan untuk memenuhi fungsionalitas yang disajikan dalam aliran kejadian (*flow of events*), dan memastikan bahwa masing-masing *class* digunakan dalam fungsionalitas. Atau dengan kata lain, membuat diagram sekuensial lebih dulu untuk mendapatkan rancangan yang terbaik, dan tidak dibatasi hanya pada *class-class* yang sudah di defenisikan sebelumnya.

*Class* memiliki tiga area pokok yaitu :

1. Nama
2. Atribut
3. Operasi

Contoh *class* manusia :

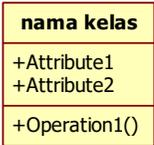
1. Atribut : nama, usia, tanggal lahir.
2. Method / Operasi : berjalan, makan, minum.

Atribut adalah karakteristik data yang dimiliki suatu objek dalam *class* (Yuni Sugiarti, 2013:58<sup>11</sup>). Operasi adalah fungsi atau transformasi yang dapat diaplikasikan ke suatu objek dalam *class* (Yuni Sugiarti, 2013:58). Misalnya suatu objek dalam *class* manusia mungkin memiliki fungsi-fungsi tersenyum, marah, makan, minum, menerima perlakuan tertentu, dan sebagainya.

---

<sup>11</sup> Yuni Sugiarti, (2013). “*Analisis Dan Perancangan UML (Unified Modeling Language) Generated VB. 6.*” Bandung : Graha Ilmu.hal 58

*Tabel 2. 2 Simbol – Simbol Class Diagram*

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas.
Operasi 	Kelas pada struktur sistem.

<p>Antarmuka / interface</p> 	<p>Sama dengan konsep interface dalam pemrograman berorientasi objek.</p>
<p>Asosiasi</p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity.</p>
<p>Asosiasi berarah / directed asosiasi</p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan multiplicity.</p>
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).</p>
<p>Kebergantungan / <i>defedency</i></p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas.</p>
<p>Agregasi</p> 	<p>Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).</p>

### 3. Activity Diagram

*Activity Diagram* adalah sebuah cara untuk memodelkan aliran kerja (*workflow*) dari *usecase* dalam bentuk grafik (Sholiq, 2006:42<sup>12</sup>). Diagram ini menunjukkan langkah-langkah di dalam aliran kerja, titik keputusan dalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas, dan objek-objek yang digunakan dalam aliran kerja.

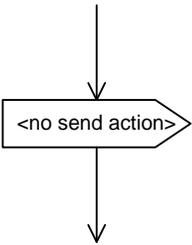
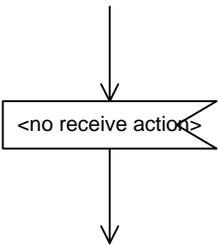
Elemen-elemen yang digunakan dalam *activity diagram* untuk memodelkan aliran kerja melalui *usecase* adalah sebagai berikut :

- a. *Swimlanes*, menunjukkan siapa yang bertanggung jawab melakukan aktivitas dalam suatu diagram.
- b. *Activities*, adalah kegiatan dalam aliran kerja.
- c. *Action*, langka-langkah dalam sebuah aktivitas. Aksi bisa terjadi pada saat memasuki aktivitas, meninggalkan aktivitas, saat di dalam aktivitas atau pada kejadian yang spesifik.
- d. *Object bisnis*, adalah entitas-entitas yang digunakan dalam aliran kerja.
- e. *Transition*, menunjukkan bagaimana aliran kerja itu berjalan dari satu aktivitas ke aktivitas lainnya.
- f. *Decision Point*, menunjukkan di mana sebuah keputusan perlu dibuat dalam aliran kerja.
- g. *Synchronization*, menunjukkan dua atau lebih langkah dalam aliran kerja berjalan secara serentak.
- h. *Start State*, menunjukkan di mana aliran kerja itu dimulai.
- i. *End State*, menunjukkan di mana aliran kerja itu berakhir.

Tabel 2. 3 Simbol-Simbol Activity Diagram

Simbol	Keterangan
	Titik Awal
	Titik Akhir
	Aktivity
	Pilihan untuk pengambilan keputusan

<sup>12</sup> Sholiq, (2006). "Pemodelan Sistem Informasi Berorientasi Objek Dengan UML." Bandung : Graha Ilmu.hal 61

Simbol	Keterangan
	Fork digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir ( <i>flow final</i> ).

## **BAB III**

### **METODE PENELITIAN**

Kerangka kerja dalam sebuah penelitian diperlukan untuk mencapai hasil yang optimal, karena hal itu maka peneliti akan membuat kerangka kerja menggunakan metode *System Development Life Cycle* (SDLC), dimana pada metode ini menggunakan beberapa fase - fase yang digunakan sebagai pemandu dalam menyelesaikan penelitian sehingga tujuan penelitian dapat tercapai.

Adapun kerangka kerja dari metode *System Development Life Cycle* (SDLC) adalah sebagai berikut:

#### **A. Perencanaan Sistem**

##### **a. Ruang Lingkup masalah**

Masalah pokok dalam penelitian adalah seringnya timbul rasa kecewa konsumen online shop terhadap terlambatnya paket yang dipesan dari online shop. Hal tersebut disebabkan perusahaan jasa distribusi paket belum mendistribusikan paket kepada konsumen dengan tepat waktu secara menyeluruh. Masalah tersebut ada disebabkan paket yang masuk ke gudang perusahaan jasa layanan distribusi sangat banyak bahkan overload didukung human error dikarenakan kelelahan dalam menyusun dan memfilter paket yang akan distribusikan sehingga terjadi penumpukan paket di gudang perusahaan. Untuk itu diperlukan metode Fifo untuk mengantri paket yang masuk sehingga paket dapat dikontrol menggunakan media komputerisasi. Tidak hanya sampai disitu saja ternyata paket yang sudah ditangan kurir juga dapat menyebabkan terhambat sampai dengan tepat waktu kepada konsumen. Hal tersebut dikarenakan jalanan yang macet dan rute yang banyak untuk menuju konsumen, sehingga diperlukan metode untuk menyelesaikan masalah tersebut. Metode yang paling tepat dan optimal adalah TSP. TSP dapat memberikan solusi dengan menghitung jarak minimal dari kurir ke konsumen. Apabila kedua masalah tersebut dapat diatasi dengan kedua metode maka penyampaian paket tepat waktu kepada pelanggan dapat diatasi, menghemat waktu pendistribusian dan meningkatkan nilai ekonomi pada perusahaan jasa layanan distribusi paket.

b. Tujuan penelitian

Tujuan penelitian ini adalah Mengetahui penggunaan FIFO *First in First out* dalam memecahkan masalah penentuan paket yang akan didistribusikan ke konsumen, mengetahui *Travelling Salesman Problem* (TSP) dengan metode Dijkstra dapat digunakan oleh kurir untuk mendistribusikan paket ke alamat konsumen dengan hanya sekali kunjungan saja, Merancang sistem pendistribusian paket sehingga memudahkan untuk implementasi sistem menggunakan pemrograman visual Basic. Net dan basis data Mysql

c. Manfaat penelitian

Hasil dari penelitian ini berupa aplikasi komputer yang bermanfaat bagi perusahaan jasa layanan distribusi online shop untuk dapat mengontrol paket yang ada digudang selanjutnya dapat memudahkan kurir untuk menentukan pemilihan paket yang akan dikirim ke rute konsumen dengan tepat

## **B. Analisa Kebutuhan Sistem dan Informasi**

Pada tahap analisa kebutuhan sistem adalah dengan menganalisa kebutuhan informasi fungsi sistem yaitu berupa proses – proses yang terjadi terhadap sistem dan kebutuhan non fungsi yaitu kebutuhan yang harus dipenuhi untuk perilaku sistem. Kebutuhan perilaku berkaitan dengan kebutuhan operasional, dan kinerja sistem. Untuk mendapatkan kebutuhan tersebut diperlukan teknik pengumpulan kebutuhan dengan wawancara, observasi dan analisis dokumen.

## **C. Pemodelan dan Perancangan Sistem**

a. Perancangan Algoritma

Perancangan yang dilakukan adalah dengan menggunakan pemodelan diagram flowchart untuk menunjukkan kerja algoritma fifo dan djikstra terhadap paket. Selanjutnya pemodelan menggunakan UML untuk memodelkan model proses yang berjalan di dalam sistem, dimana model UML yang akan digunakan adalah diagram *uses case, activity dan sequance*

b. Perancangan *Logikal*

Perancangan logikal adalah perancangan basis data berkaitan dengan data yang akan dikelola oleh sistem. Untuk memudahkan dalam merancang basis data maka yang dilakukan

membuat kamus data dari basis data, dan merancang ERD pada data yang berelasi.

c. Perancangan Model Proses

Pada tahap ini adalah perancangan modeling sistem menggunakan alat bantu pemodelan sistem berbasis objek (UML) walaupun secara spesifik diagram model yang dipakai hanya *Use Case Diagram* dan *Activity Diagram*

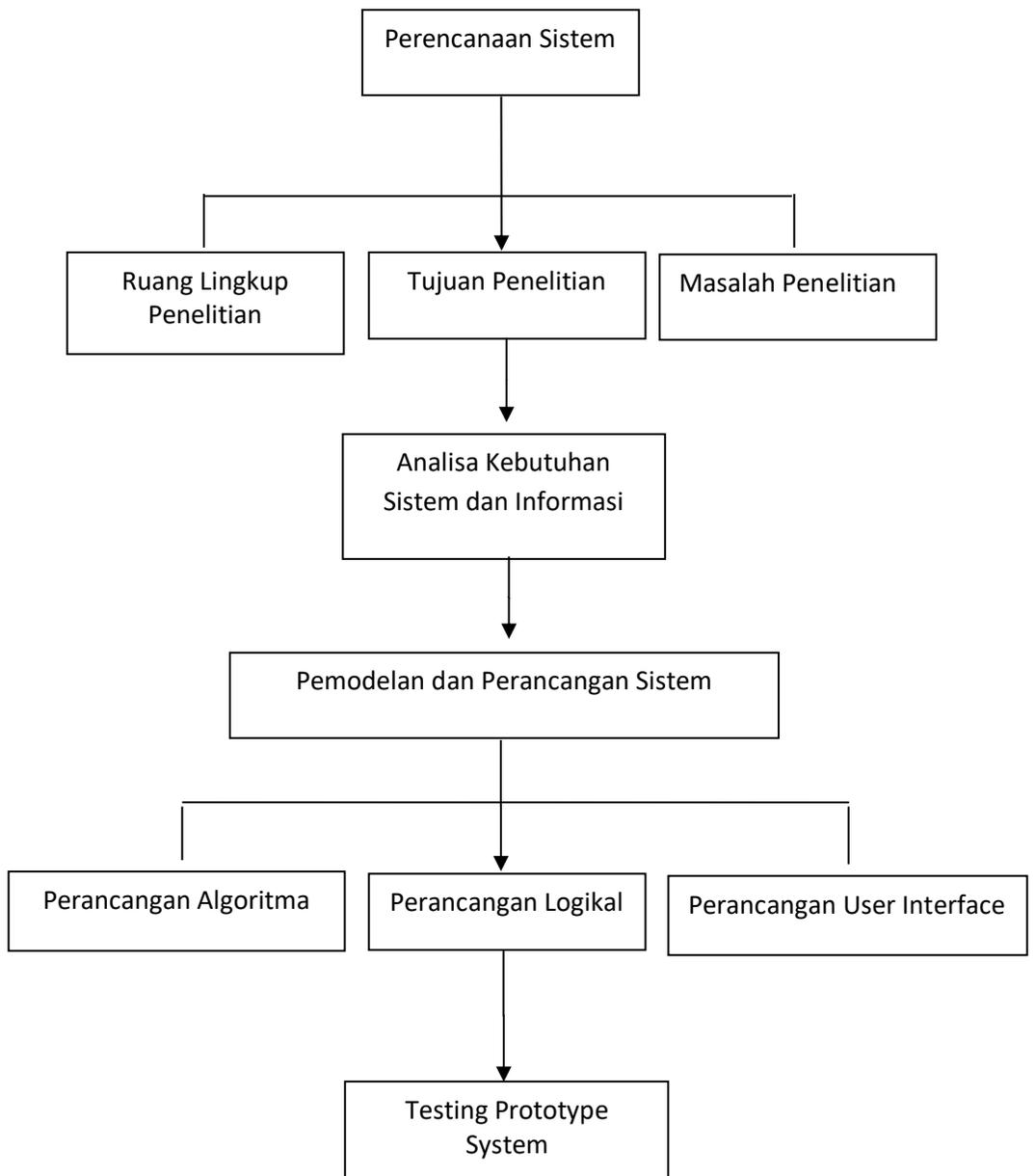
d. Perancangan *User Interface*

Pada tahap perancangan *user interface* dengan membuat rancangan form input sistem, output sistem dan menu – menu utama sistem aplikasi yang menarik untuk memudahkan penggunaan.

**D. Testing Prototype Sistem**

Pada tahap ini dilakukan testing terhadap sistem yang telah diimplementasikan menggunakan bahasa pemrograman visual basic.net dan mysql. Tahapan pengujian dimulai dengan menguji antar muka dan logika pemrograman pada sistem. Hal tersebut dilakukan berulang – ulang hingga sistem dapat berjalan dengan sempurna.

Metode SDLC di atas digambarkan dengan diagram metode penelitian.



*Gambar 3. 1 Diagram Alur Metode Penelitian*

## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Analisis Kebutuhan Sistem dan Informasi

##### 1. Analisis Kebutuhan Data

Pada tahap analisis kebutuhan informasi yang dilakukan adalah menganalisis data yang diperoleh dari hasil pengumpulan data melalui obeservasi lapangan dan wawancara kepada *Development Officer*, Penanggung Jawab Gudang dan Kurir JNE Kantor Perwakilan Suka Sopan Medan. Data yang diperoleh adalah data paket barang yang masuk ke gudang JNE Kantor Perwakilan Suka Sopan Medan. Data yang diambil hanya sample untuk digunakan dalam penerapan antrian paket barang menggunakan algoritma FIFO *First in First out* dan algoritma *Travelling Salesman Problem* (TSP) dengan metode Dijkstra.

Adapun data yang diperoleh dari hasil observasi adalah data paket barang pada kantor JNE Kantor Perwakilan Suka Sopan Medan Untuk Wilayah Medan Johor.

*Tabel 4. 1 Data Barang JNE Kantor Perwakilan Suka Sopan Medan Untuk Wilayah Medan Johor.*

<b>dBag</b>	<b>Date</b>	<b>IdManifest</b>	<b>Origin</b>	<b>Destination</b>	<b>Weight</b>
SY10376611	29-09-2018	MES246723	Surabaya	Medan	10
CK11475511	03-10-2018	MES983912	Jakarta	Medan	10
BD12304421	03-10-2018	MES013251	Bandung	Medan	7
BD12304422	29-09-2018	MES990288	Bandung	Medan	12
CK11475512	02-10-2018	MES347710	Jakarta	Medan	6
SY10376612	03-10-2018	MES899065	Surabaya	Medan	8
SY10376613	29-09-2018	MES770054	Surabaya	Medan	10

<b>BD12304423</b>	<b>05-10-2018</b>	<b>MES992258</b>	<b>Bandung</b>	<b>Medan</b>	<b>11</b>
<b>CK11475513</b>	<b>02-10-2018</b>	<b>MES971360</b>	<b>Jakarta</b>	<b>Medan</b>	<b>6</b>
<b>BD12304424</b>	<b>05-10-2018</b>	<b>MES654441</b>	<b>Bandung</b>	<b>Medan</b>	<b>5</b>
<b>SY10376614</b>	<b>29-09-2018</b>	<b>MES998772</b>	<b>Surabaya</b>	<b>Medan</b>	<b>8</b>
<b>SY10376615</b>	<b>04-10-2018</b>	<b>MES100114</b>	<b>Surabaya</b>	<b>Medan</b>	<b>7</b>
<b>CK11475514</b>	<b>02-10-2018</b>	<b>MES999881</b>	<b>Jakarta</b>	<b>Medan</b>	<b>4</b>
<b>CK11475515</b>	<b>03-10-2018</b>	<b>MES601254</b>	<b>Jakarta</b>	<b>Medan</b>	<b>5</b>
<b>CK11475516</b>	<b>28-09-2018</b>	<b>MES110367</b>	<b>Jakarta</b>	<b>Medan</b>	<b>8</b>

**a. Analisa Metode**

Pada tahapan ini akan dianalisa data paket barang, analisa metode yang digunakan algoritma Fifo *First in First out* dan algoritma dijkstra. Adapun hal yang pertama dilakukan adalah menganalisa data paket. Berikut adalah analisa datanya yaitu :

a) Metode Fifo *First in First out*

Metode Fifo *First in First out* digunakan untuk mengontrol barang yang masuk secara antrian ke gudang JNE. Prosedure Paket barang yang masuk ke gudang dalam bentuk karung – karung yang setiap karung memiliki idbag, dan setiap karung berisi paket barang yang memiliki noresi (informasi detail paket barang konsumen) dari masing – masing konsumen. Idbag karung diinput ke aplikasi gudang oleh petugas administrasi. Idbag yang pertama sekali diinput ke sistem aplikasi maka akan menjadi antrian pertama pada form detail idbag. Sehingga noresi yang ada pada idbag yang menjadi antrian yang pertama sekali akan diproses pertama sekali untuk diinput ke aplikasi, selanjutnya no resi – no resi tersebut akan didistribusikan kepada kurir yang pertama sekali berada pada gudang. Proses tersebut berkelanjutan sesuai dengan antrian, sampai semua paket barang yang masuk digudang

pada tanggal yang sama maka akan didistribusikan juga pada hari yang sama juga. Langkah – langkah metode FIFO (*Firs In First Out*) adalah sebagai berikut :

Operasi – operasi Fifo (*Firs In First Out*) antara lain deklarasi, inisialisasi, kemudian cek kondisi kosong atau penuh, setelah itu membuat fungsi penambahan, pengaksesan dan penghapusan.

1. Algoritma penambahan
  - 1) Mulai
  - 2) Deklarasi daftar paket barang
  - 3) Masukkan data
  - 4) selesai
2. Algoritma Pengaksesan
  - 1) Mulai
  - 2) Cetak Node head paket barang
  - 3) Selesai
3. Algoritma Penghapusan
  - 1) Mulai
  - 2) hapus Node head paket barang
  - 3) Selesai

Penjelasan Algoritma :

Pada algoritma FIFO (*Firs In First Out*) terdapat tiga proses operasi antara lain, penambahan, pengaksesan dan penghapusan. Proses penambahan dilakukan dengan membuat daftar list terhubung, memasukan data, dan selesai. Proses pengaksesan dilakukan dengan mencetak node kepala data paket barang dan selesai. Proses penghapusan dilakukan dengan menghapus node kepala data paket barang dan selesai.

Implementasi sistem antrian FIFO (*Firs In First Out*) pada paket barang adalah

Data Pertama

Masukkan idbag : SY10376611

Masukkan Date : 29-09-2018

Masukkan Origin : Surabaya

Masukkan Destination : Medan Johor

Masukkan Weight : 5

Data Kedua

Masukkan idbag : CK11475511

Masukkan Date : 03-10-2018

Masukkan Origin : Jakarta

Masukkan Destination : Medan Johor  
Masukkan Weight : 5

b) Metode Dijkstra

**Data Wilayah Medan Johor**

*Tabel 4. 2 Data Konsumen JNE Express Wilayah Medan Johor*

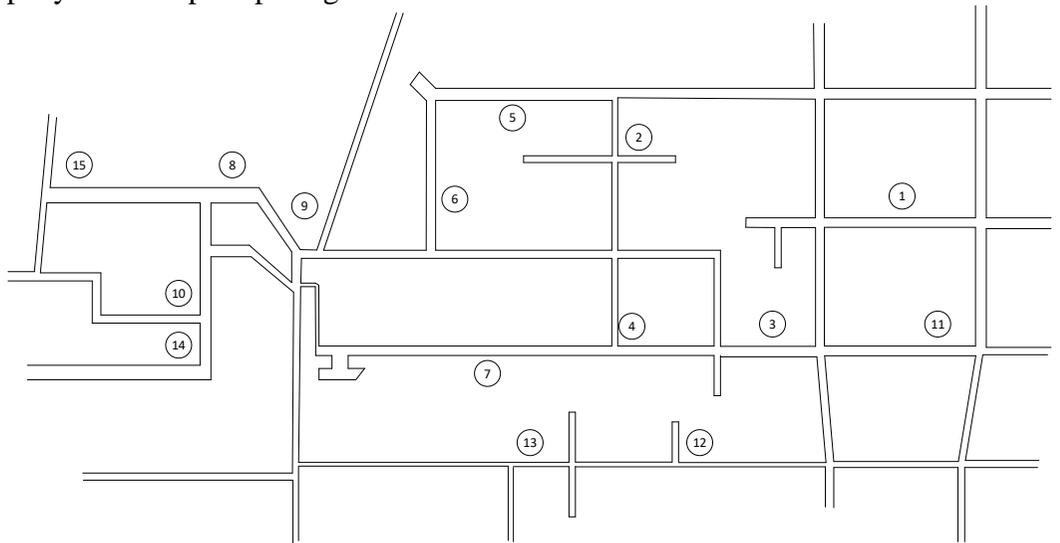
<b>Kode Pada Google Maps</b>	<b>Konsumen</b>	<b>Alamat</b>
11	A	Jl. Suka Suar Medan Johor
3	B	Jl. Suka Suar
4	C	Jl. Suka Suar
7	D	Jl. Suka Suar
2	E	Jl. Suka Menang
5	F	Jl. Suka Terang
6	G	Jl. Suka Maju
12	H	Jl. Suka Eka
13	I	Jl. Suka Eka
9	J	Jl. Suka Tirta
8	K	Gg. Sado
15	L	Gg. Sado
10	M	Jl. Brig Jend. Zein Hamid
14	N	Jl. Brig Jend. Zein Hamid

Pada tabel 4.1 adalah tabel data konsumen, yang berasal dari JNE Express Wilayah Suka Sopan. Peta wilayah penyebaran data konsumen Medan johor dapat seperti pada gambar 4.1 yang di dapat dari google maps.



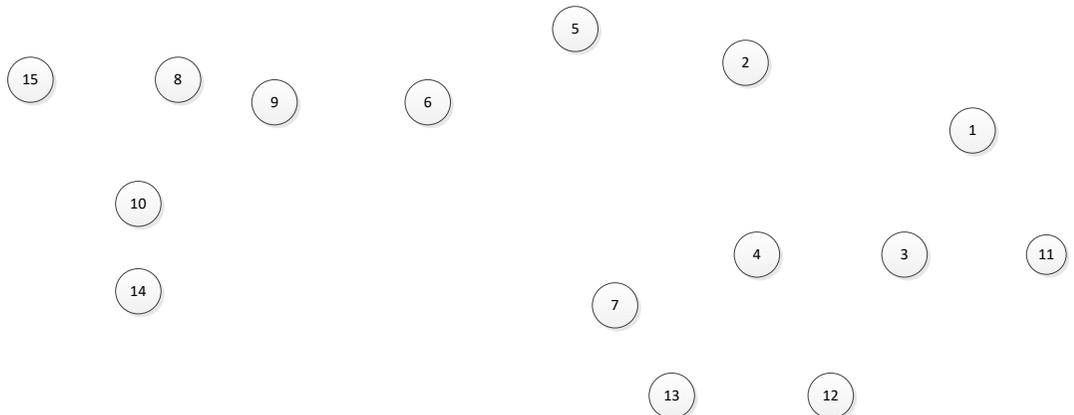
*Gambar 4. 1 Lokasi Konsumen Medan Johor Dari Google Maps*

Berdasarkan peta google maps digambarkan peta wilayah penyebaran seperti pada gambar 4.2.



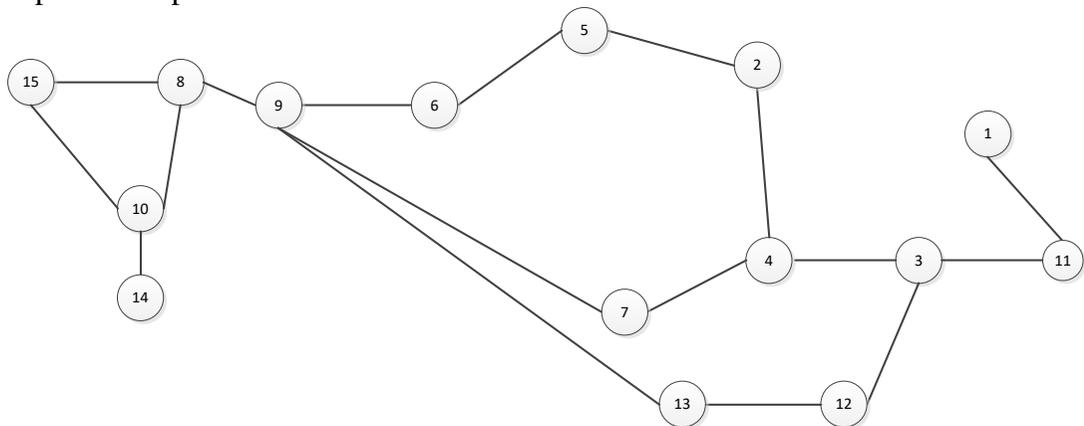
*Gambar 4. 2 Peta Lokasi Penyebaran Alamat Konsumen Wilayah Medan Johor*

Gambar 4.2 adalah gambar peta penyebaran lokasi konsumen JNE Express wilayah Medan Johor dimana letak JNE Express wilayah Suka Sopan sendiri di wakili oleh kode 1. Dari peta penyebaran wilayah 1 di dapatkan vertex penyebarannya seperti pada gambar 4.3 berikut ini.



Gambar 4. 3 Penyebaran Verteks Konsumen Wilayah Medan Johor

Dari gambar 4.3 graf yang dibentuk dengan menghubungkan jarak antar konsumen yang dianggap sebagai sisi. Dimana titik-titik yang mempunyai kode adalah untuk mewakili lokasi konsumen. Graf penyebaran dapat dilihat pada Gambar 4.4 berikut.



Gambar 4. 4 Graf Tanpa Bobot Untuk Penyebaran Konsumen Wilayah Medan Johor

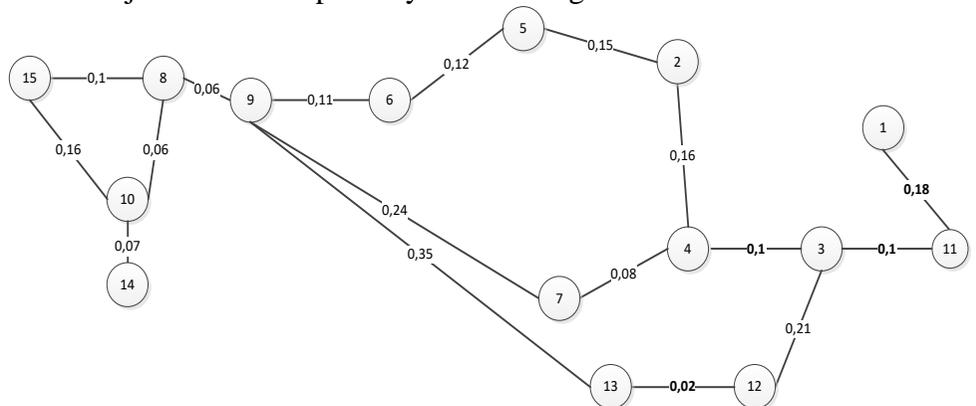
Adapun jarak antar simpul adalah sebagai berikut :

Tabel 4. 3 Jarak Antar Titik Untuk Wilayah Medan Johor

Titik Yang Terhubung	Jarak (Km)
1-11	0,18
11-3	0,1

<b>3-4</b>	0,1
<b>3-12</b>	0,21
<b>12-3</b>	0,02
<b>4-7</b>	0,08
<b>4-2</b>	0,16
<b>2-5</b>	0,15
<b>5-6</b>	0,12
<b>6-9</b>	0,11
<b>9-8</b>	0,06
<b>8-15</b>	0,1
<b>8-10</b>	0,06
<b>10-14</b>	0,07
<b>15-10</b>	0,16
<b>7-9</b>	0,24
<b>13-9</b>	0,35

Dari tabel 4.2 dapat membuat graf berbobot seperti pada gambar 4.3 berikut. Dimana jarak antar simpul dinyatakan sebagai bobot.



*Gambar 4. 5 Graf Berbobot Untuk Wilayah Medan Johor*

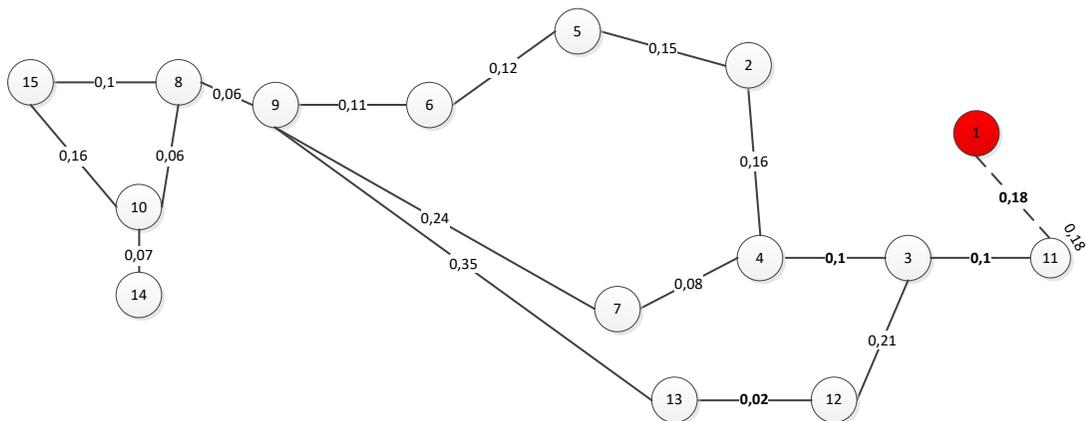
Penyelesaian pencarian lokasi konsumen menggunakan algoritma dijkstra dimulai dengan mencari pohon merentang minimum dan lintasan terpendek untuk konsumen yang berlokasi di Medan Johor. Adapun algoritma dijkstra untuk masalah ini adalah sebagai berikut :

Langkah-langkah penyelesaian algoritma dijkstra :

1. Langkah pertama menentukan status dari node yang belum terpilih dengan menginisialisasikan dengan '0' dan yang sudah terpilih diinisialisasikan dengan '1' di mulai dari node 1.
2. Tentukan bobot dari node yang langsung terhubung dengan node sumber yaitu node 1, seperti node 1 ke node 11 = 1,5. Dan node 2,3,4,5,6,7,8,9,10,12,13,14 diinisialisasikan dengan "-" karena tidak ada lintasan yang terhubung langsung dengan node 1.
3. *Predecessor* (node sumber) adalah 1, karena jarak dihitung dari node 1, sehingga node 1 disebut sebagai *predecessor*.

Tabel 4. 4 Hasil Iterasi Ke-1 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bobot	-	-	-	-	-	-	-	-	-	-	0,18	-	-	-	-
Predecessor	1	-	-	-	-	-	-	-	-	-	1	-	-	-	-

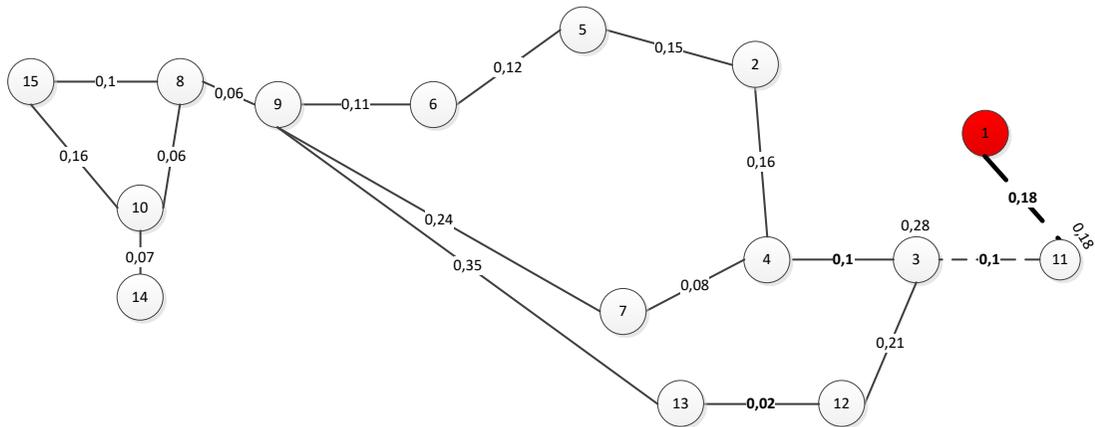


Gambar 4. 6 Node Terpilih Iterasi Ke-1 Pada Wilayah Medan Johor

Dari tabel 4.3 memilih node yang memiliki bobot paling kecil dan statusnya masih '0' yang terhubung langsung dengan node sumber yaitu node 11. Untuk itu node 11 menjadi '1' dan *predecessor*-nya masih tetap 1, dan node yang lainnya *predecessor*-nya masih sama. Jika node 11 sudah terpilih, maka node 3 mempunyai bobot 0,28 dan *predecessor*-nya 11.

Tabel 4. 5 Hasil Iterasi Ke-2 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bobot	-	-	0,28	-	-	-	-	-	-	-	0,18	-	-	-	-
Predecessor	1	-	11	-	-	-	-	-	-	-	1	-	-	-	-

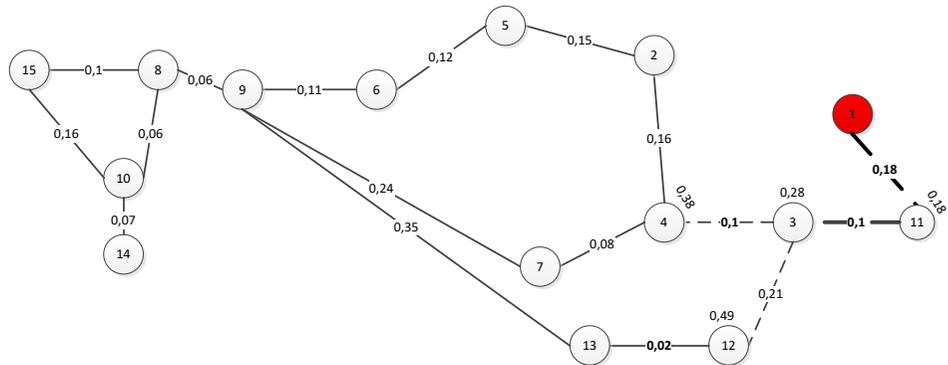


Gambar 4. 7 Node Terpilih Iterasi Ke-2 Pada Wilayah Medan Johor

Dari tabel 4.4 di dapatkan bahwa node 11 memiliki bobot yang paling kecil dan terhubung dengan node 3, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 1. Jika node 3 sudah terpilih, maka node 4 mempunyai bobot 0,38 dan node 12 mempunyai bobot 0,49. Sehingga diperoleh:

Tabel 4. 6 Hasil Iterasi Ke-3 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Bobot	-	-	0,28	0,38	-	-	-	-	-	-	0,18	0,49	-	-	-
Predecessor	1	-	11	3	-	-	-	-	-	-	1	3	-	-	-

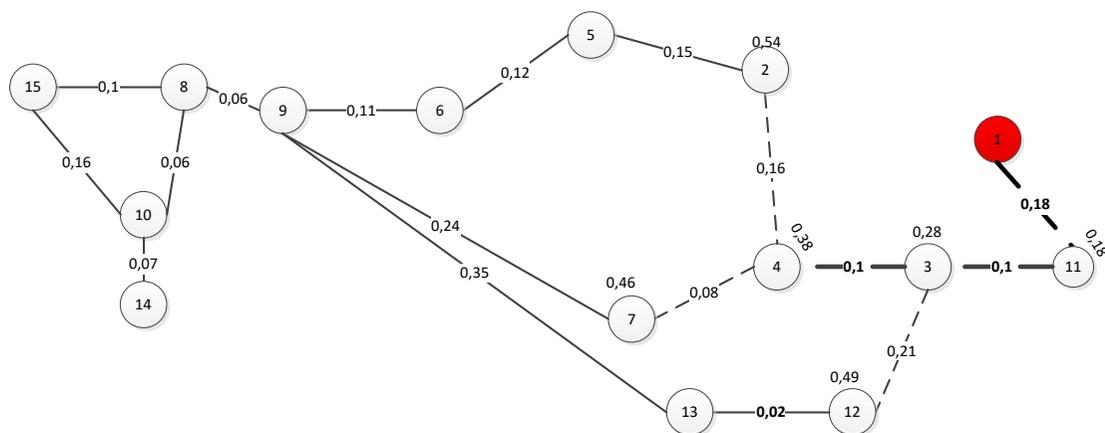


Gambar 4. 8 Node Terpilih Iterasi Ke-3 Pada Wilayah Medan Johor

Dari tabel 4.5 di dapatkan bahwa node 3 memiliki bobot paling kecil dan yang terhubung langsung dengan node 4 dan 12, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 11. jika node sudah dipilih, maka node 2 mempunyai bobot 0,54 dan node 7 bobotnya 0,46, sehingga diperoleh :

Tabel 4. 7 Hasil Iterasi Ke-4 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0
Bobot	-	0,54	0,28	0,38	-	-	0,46	-	-	-	0,18	0,49	-	-	-
Predecessor	1	4	11	3	-	-	4	-	-	-	1	3	-	-	-

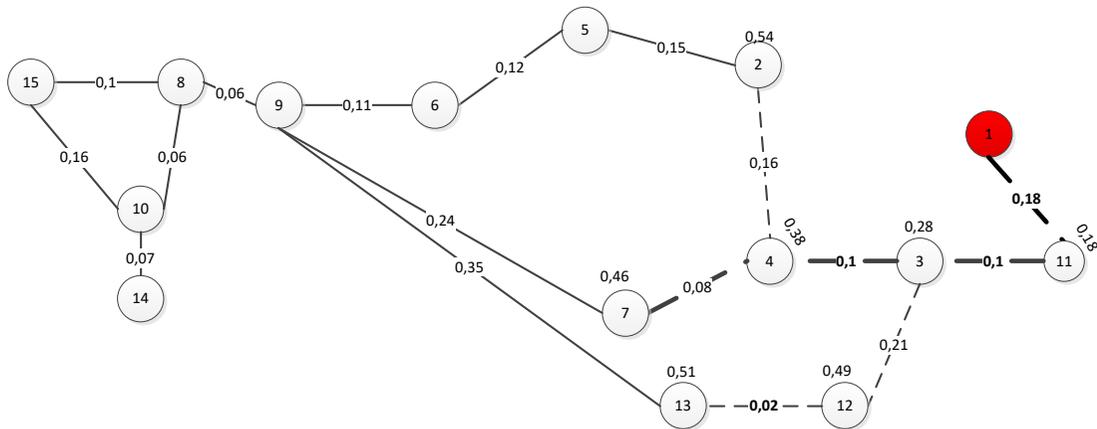


Gambar 4. 9 Node Terpilih Iterasi Ke-4 Pada Wilayah Medan Johor

Dari tabel 4.10 di dapatkan bahwa node 4 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 3. Jika node sudah terpilih, maka node 13 mempunyai bobot 0,51. Sehingga diperoleh :

Tabel 4. 8 Hasil Iterasi Ke-5 Pada Wilayah Medan Johor

Node Status	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0
Bobot	-	0,54	0,28	0,38	-	-	0,46	-	-	-	0,18	0,49	0,51	-	-
Predecessor	1	4	11	3	-	-	4	-	-	-	1	3	12	-	-



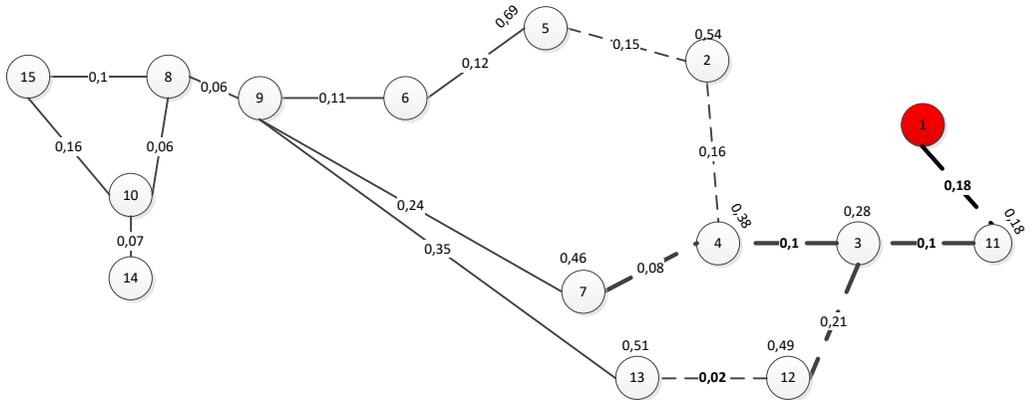
Gambar 4. 10 Node Terpilih Iterasi Ke-5 Pada Wilayah Medan Johor

Dari tabel 4.6 di dapatkan bahwa node 7 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 4. Jika node sudah terpilih, maka node 5 mempunyai bobot 0,69. Sehingga diperoleh :

Tabel 4. 9 Hasil Iterasi Ke-6 Pada Wilayah Medan Johor

Node Status	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0
Bobot	-	0,54	0,28	0,38	0,69	-	0,46	-	-	-	0,18	0,49	0,51	-	-

Predecessor	1	4	11	3	2	-	4	-	-	-	1	3	12	-	-
-------------	---	---	----	---	---	---	---	---	---	---	---	---	----	---	---

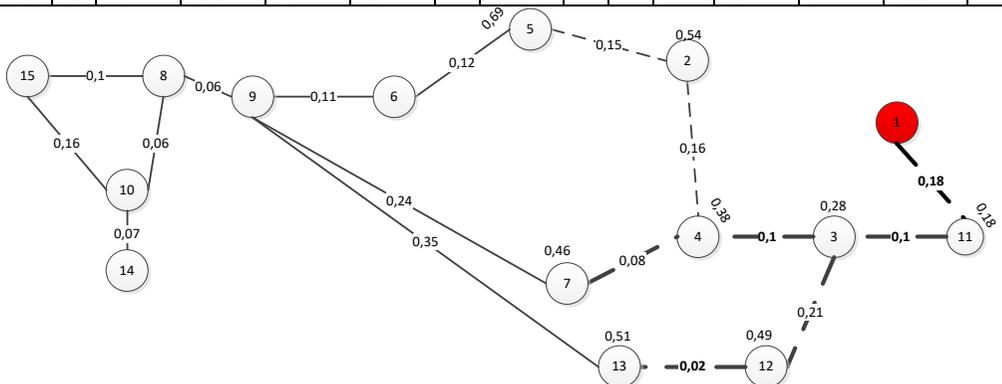


Gambar 4. 11 Node Terpilih Iterasi Ke-6 Pada Wilayah Medan Johor

Dari tabel 4.7 di dapatkan bahwa node 12 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan predecessor-nya adalah 3. Jika node 13 sudah terpilih, maka node mempunyai bobot 0,51. Sehingga diperoleh :

Tabel 4. 10 Hasil Iterasi Ke-7 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	0	1	1	0	0	1	0	0	0	1	1	0	0	0
Bobot	-	0,54	0,28	0,38	0,69	-	0,46	-	-	-	0,18	0,49	0,51	-	-
Predecessor	1	4	11	3	2	-	4	-	-	-	1	3	12	-	-



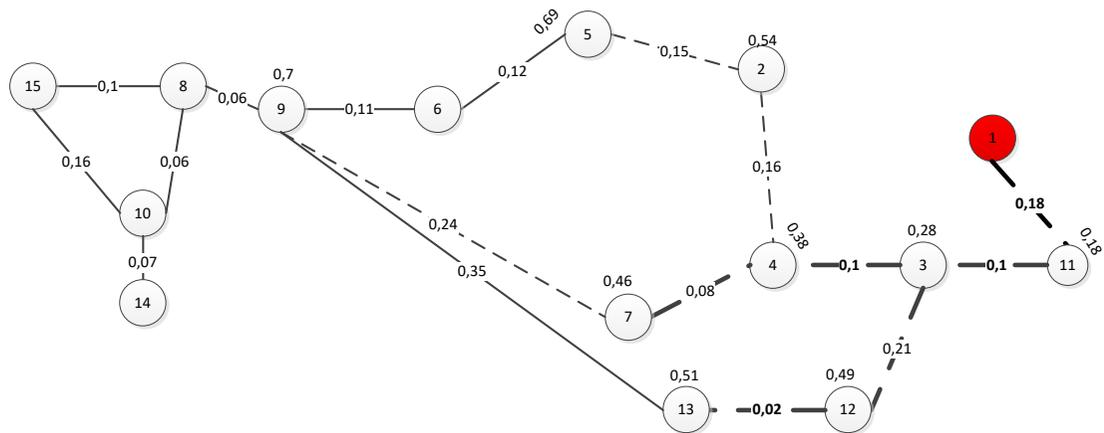
Gambar 4. 12 Node Terpilih Iterasi Ke-7 Pada Wilayah Medan Johor

Dari tabel 4.8 di dapatkan bahwa node 13 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan predecessor-nya adalah

12. Jika node 9 sudah terpilih maka mempunyai bobot 0,7. Sehingga diperoleh :

Tabel 4. 11 Hasil Iterasi Ke-8 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	0	1	1	0	0	1	0	0	0	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	-	0,46	-	0,7	-	0,18	0,49	0,51	-	-
Predecessor	1	4	11	3	2	-	4	-	7	-	1	3	12	-	-

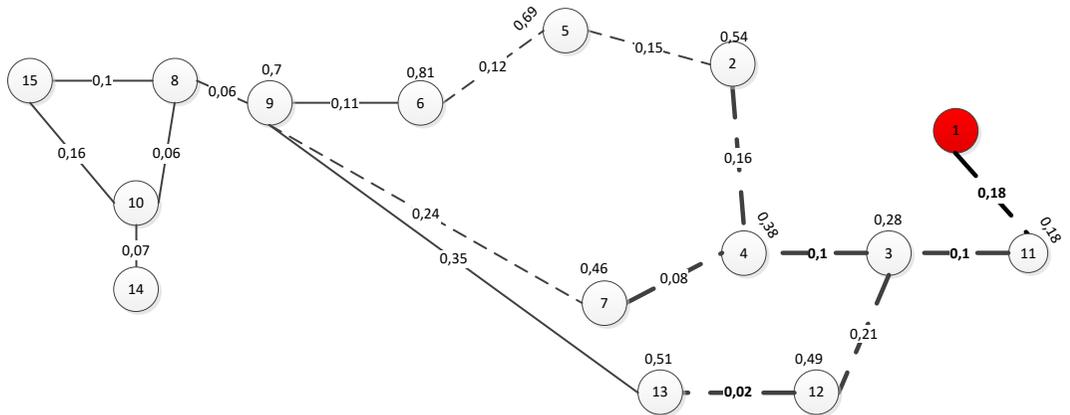


Gambar 4. 13 Node Terpilih Iterasi Ke-8 Pada Wilayah Medan Johor

Dari tabel 4.9 di dapatkan bahwa node 2 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 4. Jika node 6 sudah terpilih maka mempunyai bobot 0,81 Sehingga diperoleh :

Tabel 4. 12 Hasil Iterasi Ke-9 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	0	0	1	0	0	0	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	-	0,7	-	0,18	0,49	0,51	-	-
Predecessor	1	4	11	3	2	2	4	-	7	-	1	3	12	-	-

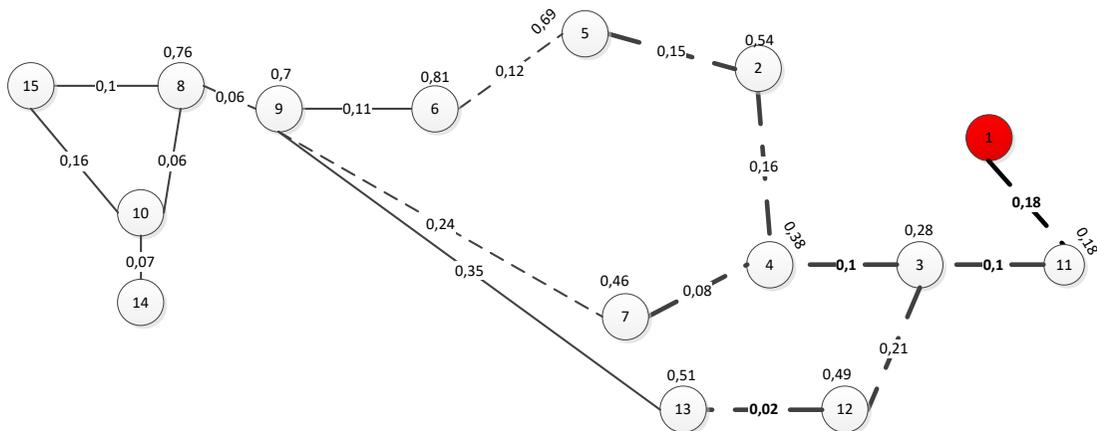


Gambar 4. 14 Node Terpilih Iterasi Ke-9 Pada Wilayah Medan Johor

Dari tabel 4.10 di dapatkan bahwa node 5 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 2. Jika node 8 sudah terpilih maka mempunyai bobot 0,76 Sehingga diperoleh :

Tabel 4. 13 Hasil Iterasi Ke-10 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	0	1	0	0	0	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	-	0,18	0,49	0,51	-	-
Predecessor	1	4	11	3	2	5	4	9	7	-	1	3	12	-	-

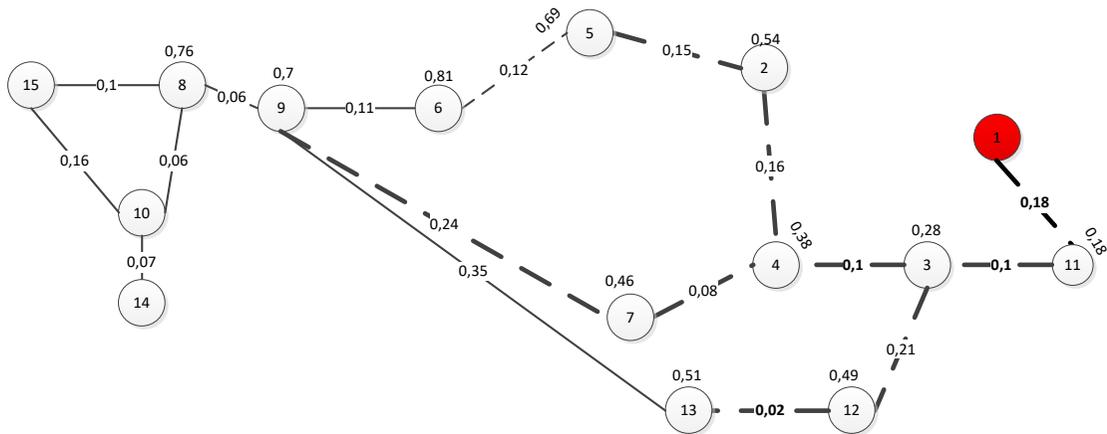


Gambar 4. 15 Node Terpilih Iterasi Ke-10 Pada Wilayah Medan Johor

Dari tabel 4.11 di dapatkan bahwa node 9 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 7. Sehingga diperoleh :

Tabel 4. 14 Hasil Iterasi Ke-11 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	0	1	0	1	0	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	-	0,18	0,49	0,51	-	-
Predecessor	1	4	11	3	2	5	4	9	7	-	1	3	12	-	-

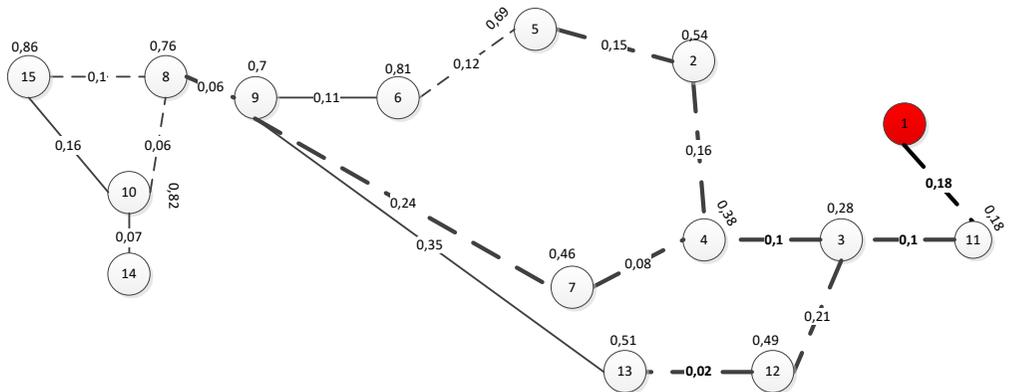


Gambar 4. 16 Node Terpilih Iterasi Ke-11 Pada Wilayah Medan Johor

Dari tabel 4.12 di dapatkan bahwa node 8 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 9. Jika node 8 telah dipilih maka node selanjutnya adalah node 10 dengan bobot 0,82 dan node 15 dengan bobot 0,86. Sehingga diperoleh :

Tabel 4. 15 Hasil Iterasi Ke-12 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	0,82	0,18	0,49	0,51	-	0,86
Predecessor	1	4	11	3	2	5	4	9	7	8	1	3	12	-	8

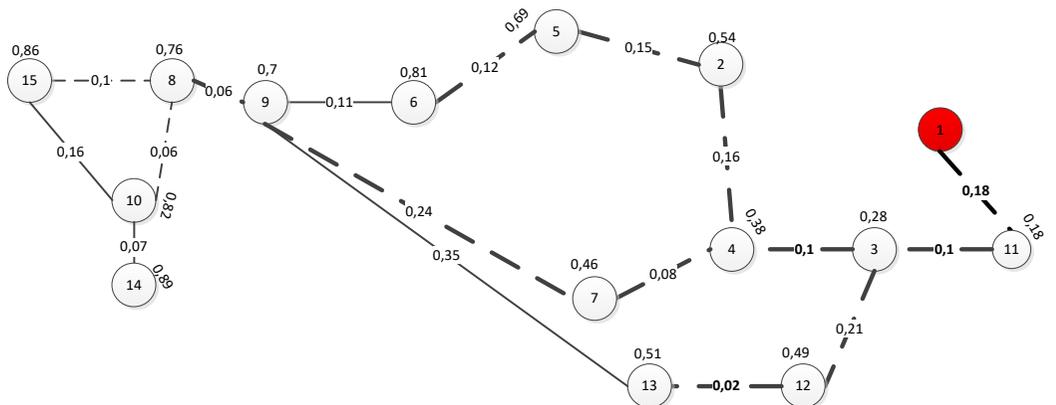


Gambar 4. 17 Node Terpilih Iterasi Ke-12 Pada Wilayah Medan Johor

Dari tabel 4.13 di dapatkan bahwa node 6 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 5. Jika node yang terpilih adalah node 14 maka bobotnya adalah 0,89 Sehingga diperoleh :

Tabel 4. 16 Hasil Iterasi Ke-13 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	0,82	0,18	0,49	0,51	0,89	0,86
Predecessor	1	4	11	3	2	5	4	9	7	8	1	3	12	10	8

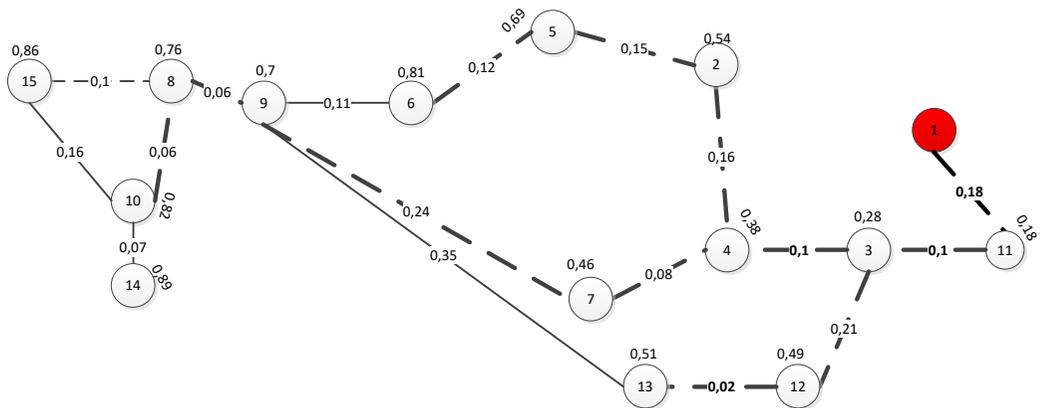


Gambar 4. 18 Node Terpilih Iterasi Ke-13 Pada Wilayah Medan Johor

Dari tabel 4.14 di dapatkan bahwa node 10 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 8. Sehingga diperoleh :

Tabel 4. 17 Hasil Iterasi Ke-14 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	0,82	0,18	0,49	0,51	0,89	0,86
Predecessor	1	4	11	3	2	5	4	9	7	8	1	3	12	10	8

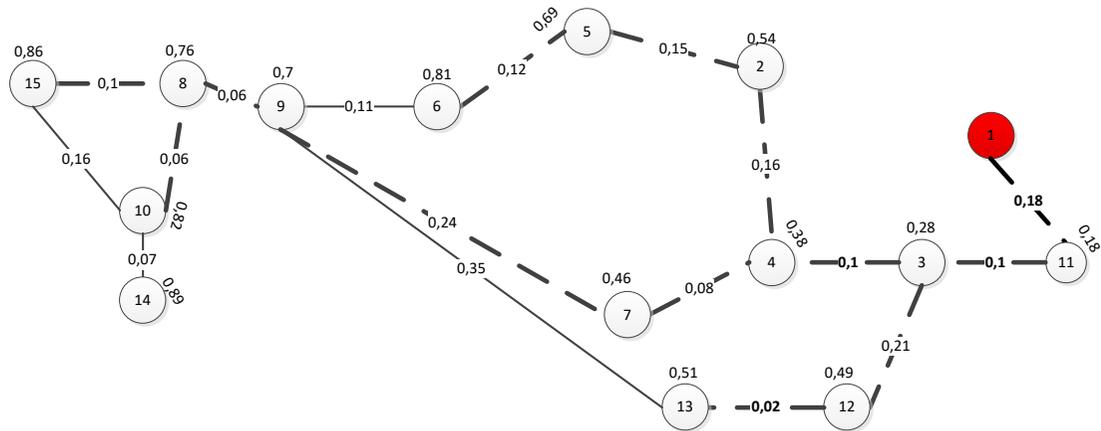


Gambar 4. 19 Node Terpilih Iterasi Ke-14 Pada Wilayah Medan Johor

Dari tabel 4.19 di dapatkan bahwa node 15 memiliki bobot paling kecil, sehingga statusnya akan berubah menjadi '1' dan *predecessor*-nya adalah 8. Sehingga diperoleh :

Tabel 4. 18 Hasil Iterasi Ke-15 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	0,82	0,18	0,49	0,51	0,89	0,86
Predecessor	1	4	11	3	2	5	4	9	7	8	1	3	12	10	8

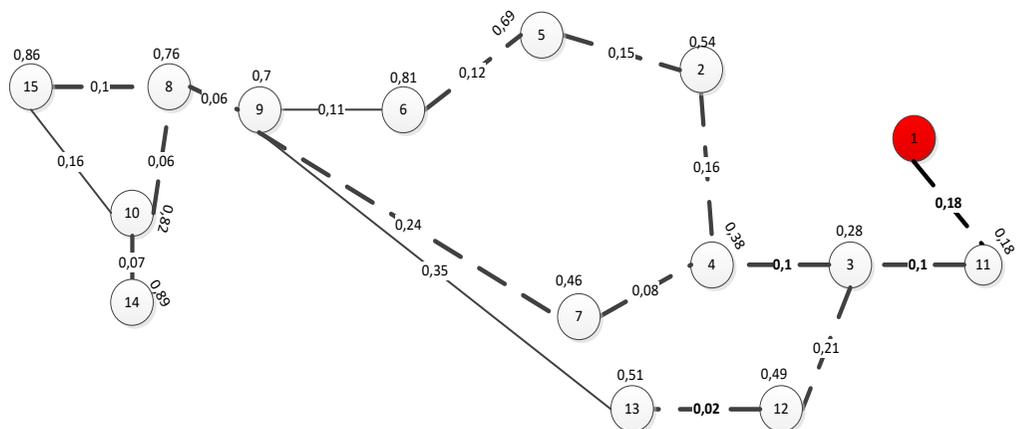


Gambar 4. 20 Node Terpilih Iterasi Ke-15 Pada Wilayah Medan Johor

Semua node telah terpilih dan hanya tinggal node 14 yang belum terpilih, selanjutnya status node 14 akan berubah menjadi '1', *predecessor*-nya 10. Sehingga menjadi:

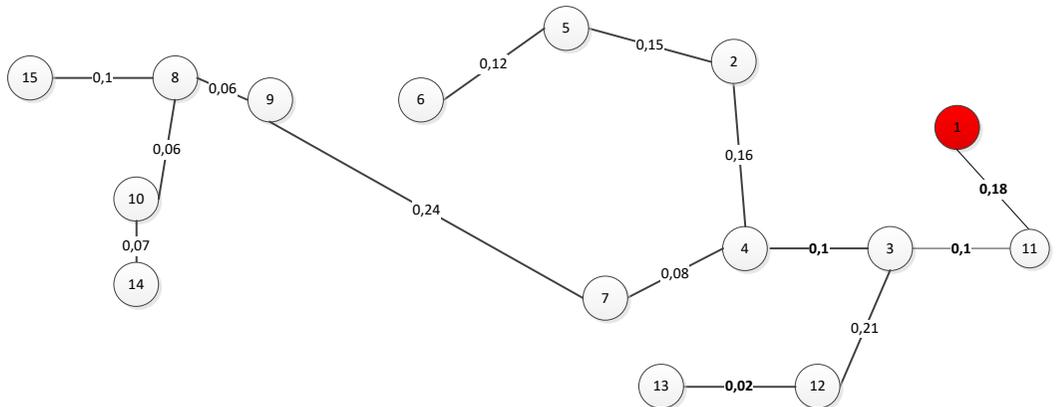
Tabel 4. 19 Hasil Iterasi Ke-16 Pada Wilayah Medan Johor

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Status	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bobot	-	0,54	0,28	0,38	0,69	0,81	0,46	0,76	0,7	0,82	0,18	0,49	0,51	0,89	0,86
Predecessor	1	4	11	3	2	5	4	9	7	8	1	3	12	10	8



Gambar 4. 21 Node Terpilih Iterasi Ke-16 Pada Wilayah Medan Johor

Dari hasil iterasi pada wilayah Medan Johor, maka di dapatkan pohon merentang minimum menggunakan algoritma dijkstra seperti gambar 4.22 berikut.



*Gambar 4. 22 Pohon Rentang Minimum Pada Wilayah Medan Johor Menggunakan Algoritma Dijkstra*

Total bobot pada pohon rentang minimum yang di dapat adalah 0,89 Km. Dan berikut adalah tabel lintasan terpendek dari simpul awal ke simpul lainnya dari hasil penyelesaian menggunakan algoritma djikstra:

*Tabel 4. 20 Lintasan Terpendek dari Simpul Awal Ke Simpul Lainnya Pada Wilayah Medan Johor Menggunakan Algoritma Dijkstra*

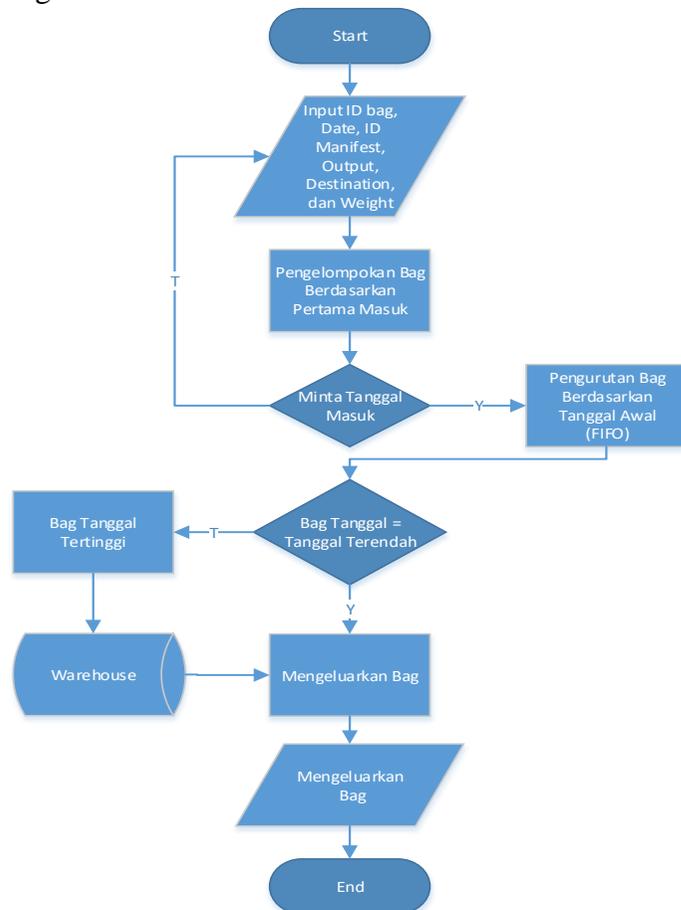
Simpul Tujuan	Lintasan	Jarak (Km)
1-2	1-11-3-4-2	0,54
1-3	1-11-3	0,28
1-4	1-11-3-4	0,38
1-5	1-11-3-4-2-5	0,69
1-6	1-11-3-4-2-5-6	0,81
1-7	1-11-3-4-7	0,46
1-8	1-11-3-4-7-9-8	0,76
1-9	1-11-3-4-7-9	0,7
1-10	1-11-3-4-7-9-8-10	0,82

1-11	1-11	0,18
1-12	1-11-3-12	0,49
1-13	1-11-3-12-13	0,51
1-14	1-11-3-4-7-9-8-10-14	0,89
1-15	1-11-3-4-7-9-8-15	0,86

## 2. Analisa Kebutuhan Sistem

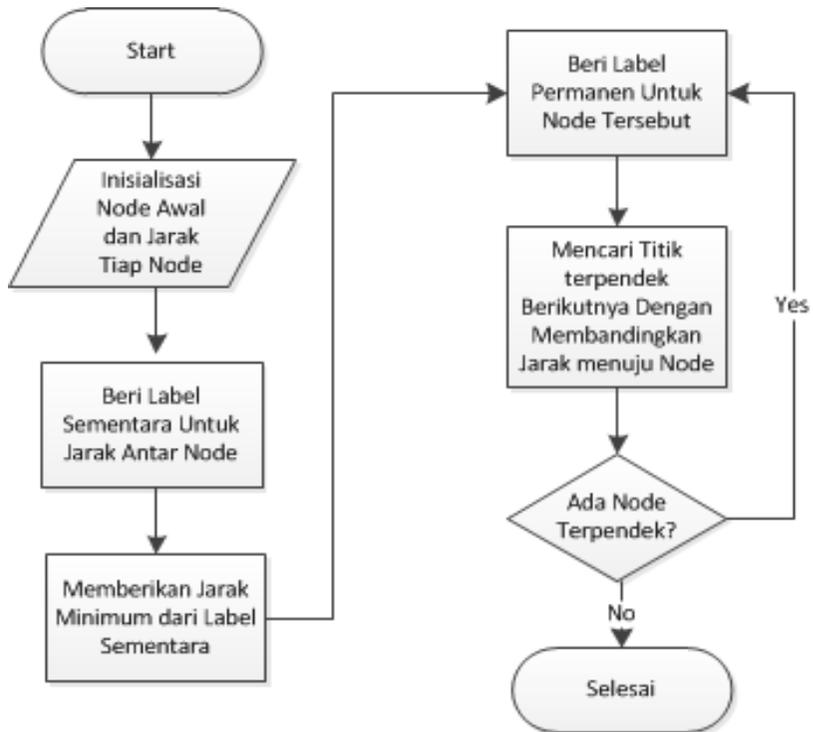
### a) Perancangan Algoritma

#### 1) Algoritma FIFO



Gambar 4. 23 Flowchart Penerapan FIFO

## 2) Algoritma Dijkstra

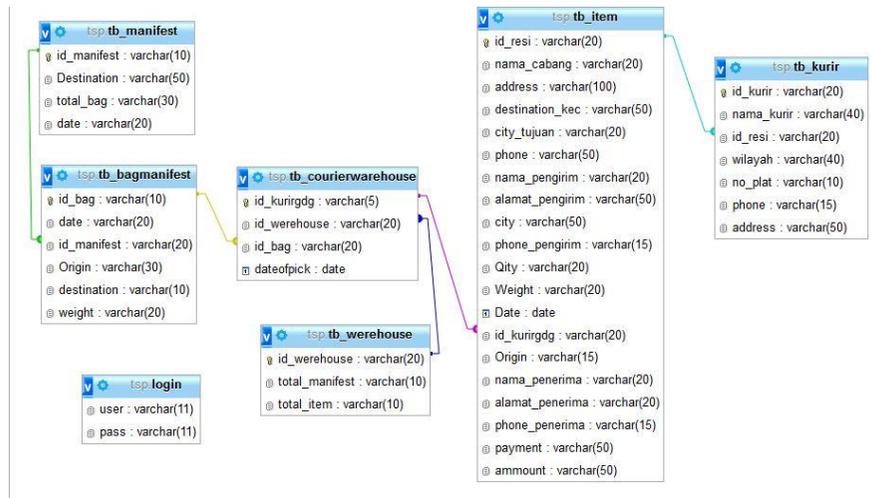


Gambar 4. 24 Flowchart penerapan algoritma Dijkstra

### b) Perancangan Logikal

#### 1) Rancangan Entity Relationship Model (Database)

Entity Relational Model ini adalah gambaran rancangan database dimana terdapat enam buah tabel yang berelasi dan satu tabel login yang berdiri sendiri. Model relasinya terlihat pada gambar 4.24



Gambar 4. 25 Tabel relasi yang dibangun di dalam Database

## 2) Rancangan Tabel (Kamus Data)

Dari Entity Relational Model (ERM) di atas maka, setiap tabel dibentuk struktur yang membangunnya diantaranya adalah field yang dibentuk, tipe data, index field dan size data yang disediakan dari ke tujuh tabel yang dirancang dan direlasikan atau biasa disebut juga dengan kamus data. Tabel-tabel tersebut diantaranya:

### a) Table login

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra Tindakan
1	user	varchar(11)	latin1_swedish_ci	Tidak	Tidak ada	Ubah, Hapus, Jelajahi nilai distingtif, Kunci Utama, Unik, Indeks, Spasial, Teks penuh
2	pass	varchar(11)	latin1_swedish_ci	Tidak	Tidak ada	Ubah, Hapus, Jelajahi nilai distingtif, Kunci Utama, Unik, Indeks, Spasial, Teks penuh

Gambar 4. 26 Rancangan struktur tabel login

## b) Table Manifest

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
<input type="checkbox"/>	1 <b>id_manifest</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 <b>Destination</b>	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 <b>total_bag</b>	varchar(30)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	4 <b>date</b>	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya

Gambar 4. 27 Rancangan struktur tabel manifest

## c) Table Bag On Manifest

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
<input type="checkbox"/>	1 <b>id_bag</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 <b>date</b>	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 <b>id_manifest</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	4 <b>Origin</b>	varchar(30)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	5 <b>destination</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	6 <b>weight</b>	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya

Gambar 4. 28 Rancangan struktur tabel Bag on Manifest

## d) Table Kurir

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
<input type="checkbox"/>	1 <b>id_kurir</b>	char(5)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 <b>nama_kurir</b>	varchar(40)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 <b>wilayah</b>	varchar(40)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	4 <b>no_plat</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	5 <b>phone</b>	varchar(15)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	6 <b>address</b>	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya

Gambar 4. 29 Rancangan struktur tabel kurir

## e) Table Warehouse

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
<input type="checkbox"/>	1 <b>id_warehouse</b>	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 <b>total_manifest</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 <b>total_item</b>	varchar(10)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya

Gambar 4. 30 Rancangan struktur tabel warehouse

## f) Table Item On Bag

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
<input type="checkbox"/>	1 id_resi		latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	2 nama_cabang	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	3 address	varchar(100)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	4 destination_kec	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	5 city_tujuan	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	6 phone	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	7 nama_pengirim	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	8 alamat_pengirim	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	9 city	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	10 phone_pengirim	varchar(15)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	11 Qity	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	12 Weight	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	13 Date	date		Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	14 Origin	varchar(15)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	15 nama_penerima	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	16 alamat_penerima	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	17 phone_penerima	varchar(15)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	18 payment	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya
<input type="checkbox"/>	19 ammount	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Lainnya

Gambar 4. 31 Rancangan struktur tabel Item on Bag

## g) Table Courier Warehouse

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
<input type="checkbox"/>	1 id_kurirgdp	varchar(5)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	2 id_bag	varchar(8)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya
<input type="checkbox"/>	3 dateofpick	date		Tidak	Tidak ada			Ubah Hapus Jelajahi nilai distingtif Kunci Utama Unik Indeks Lainnya

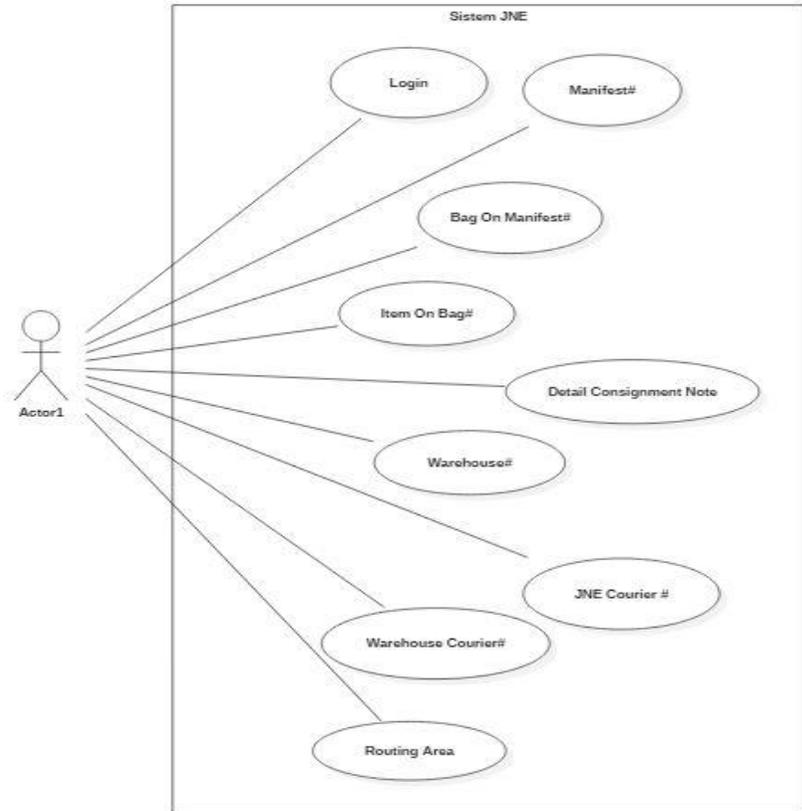
Gambar 4. 32 Rancangan struktur tabel Item on Bag

## c) Pemodelan Proses Sistem

Untuk pemodelan proses sistem digunakan alat bantu pemodelan sistem berorientasi objek, yaitu UML (*Unified Modelling Language*). UML merupakan suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek

Adapun diagram UML yang dipakai dalam memodelkan sistem ini diantaranya:

## 1) Use Case Diagram



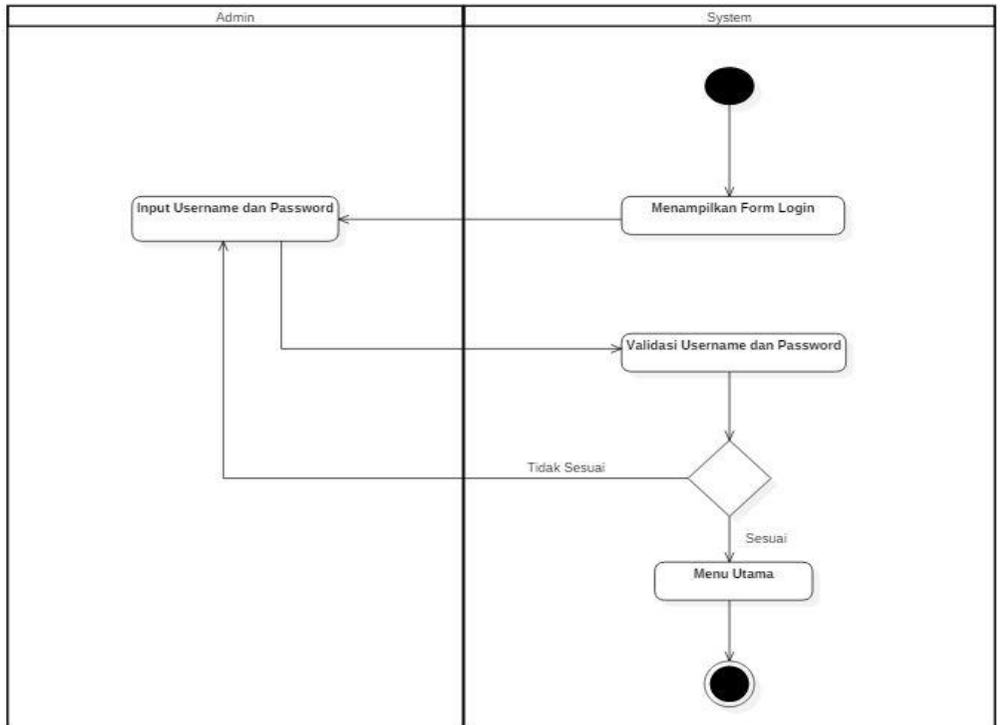
*Gambar 4. 33 Use Case Diagram Sistem Layanan Distribusi Paket*

Pada gambar di atas actor dapat mengakses setiap case diantaranya: warehouse, bag on manifest, courier, routing area dan detail consignment note

## 2) Activity Diagram

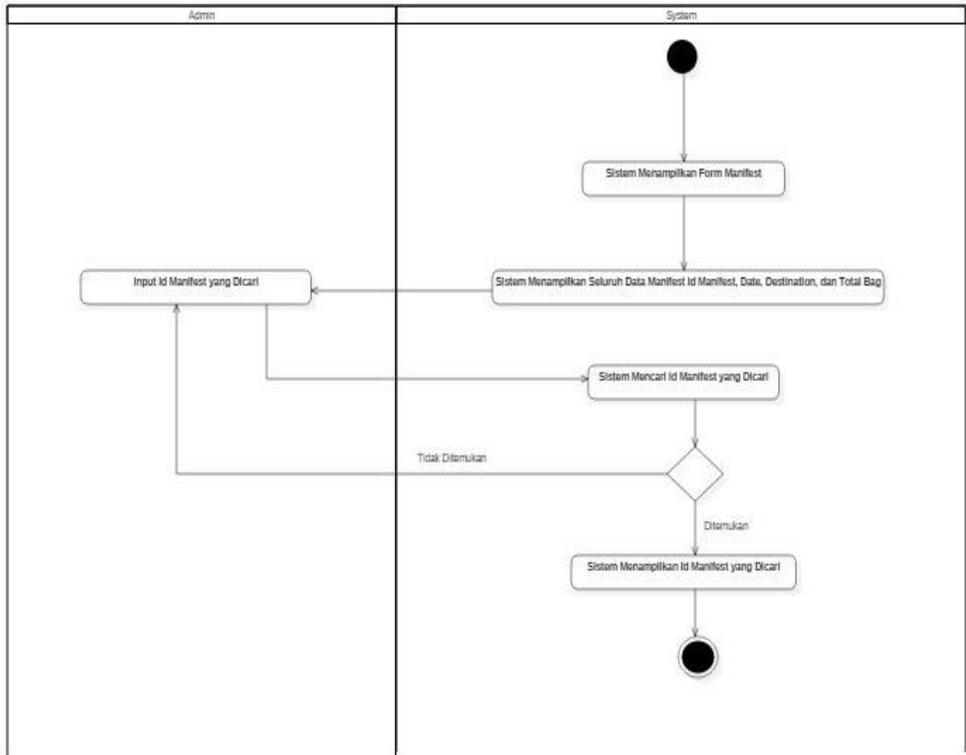
Activity diagram ini menggambarkan bagaimana aktifitas dan perilaku pengguna saat mengakses sistem dan mengelola data serta menghasilkan keputusan atau keluaran yang diinginkan

### a) Activity Diagram Proses Login



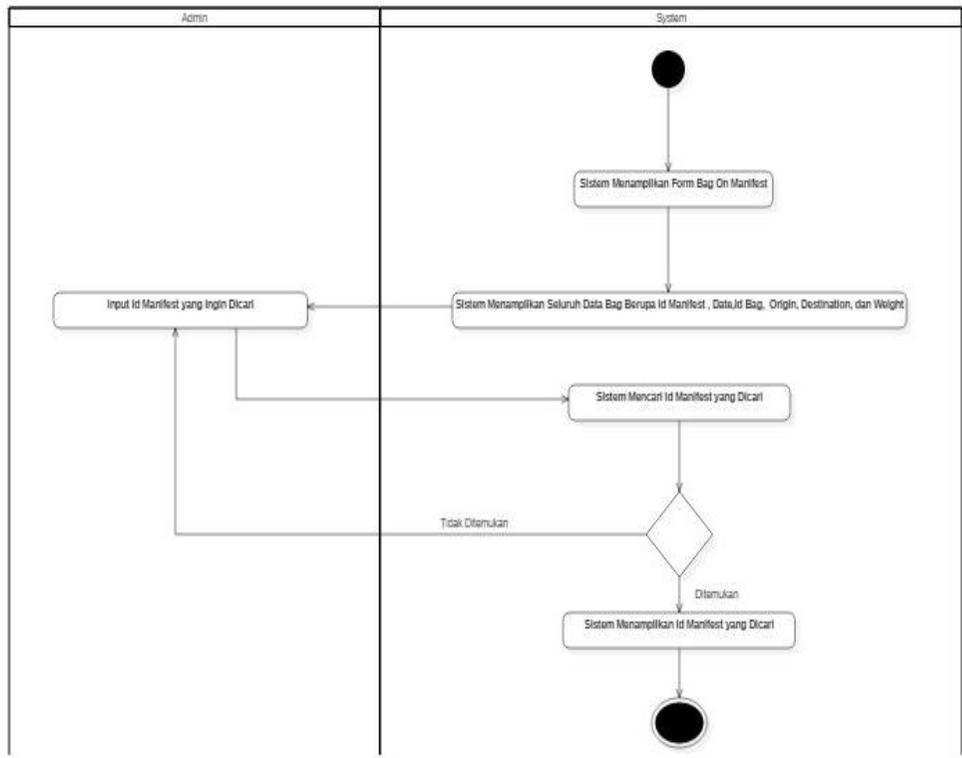
*Gambar 4. 34 Activity Diagram Proses Login*

b) Activity Diagram Proses Manifest



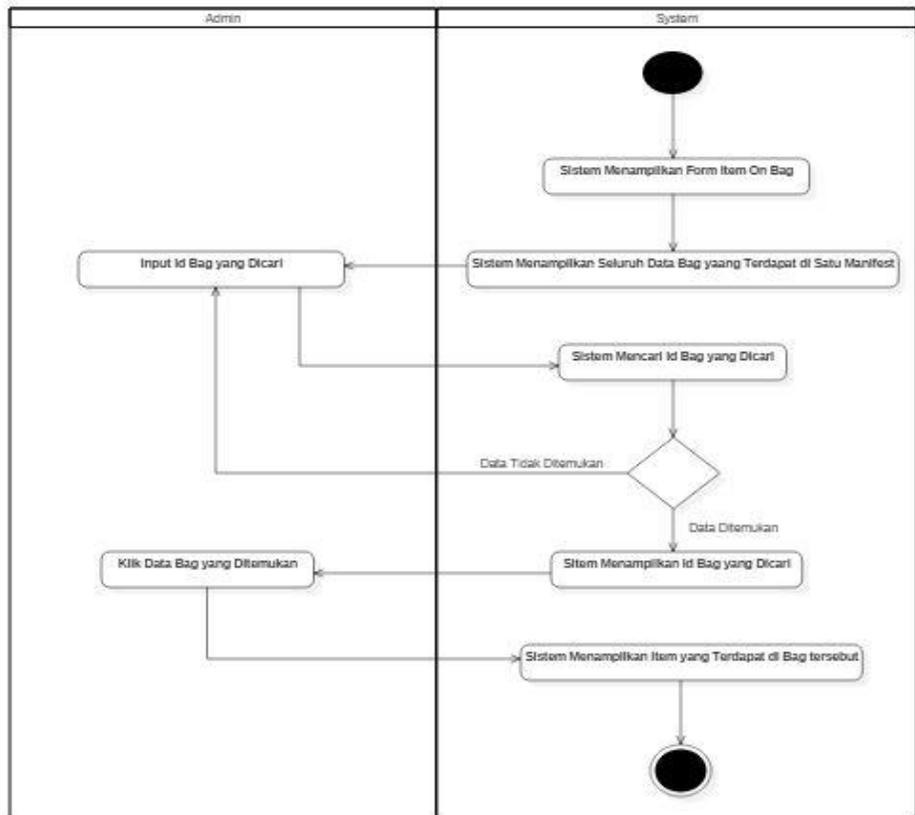
Gambar 4. 35 Activity Diagram Proses Manifest

c) Activity Diagram Proses Bag On Manifest



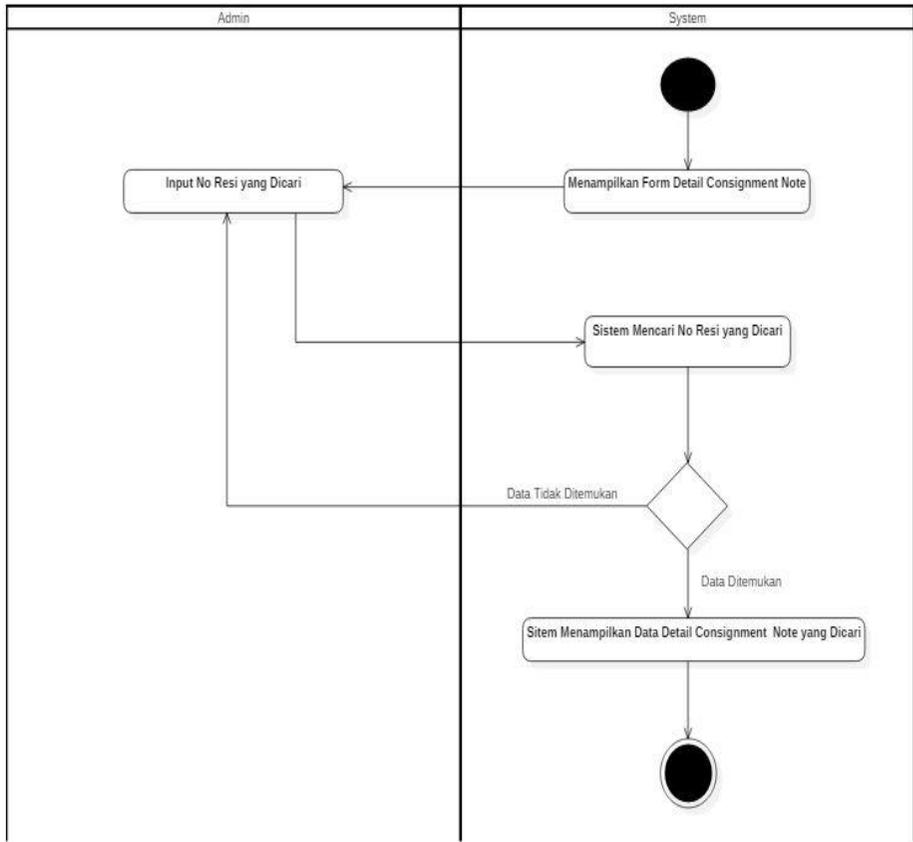
Gambar 4. 36 Activity Diagram Proses Bag on Manifest

d) Activity Diagram Item On Bag



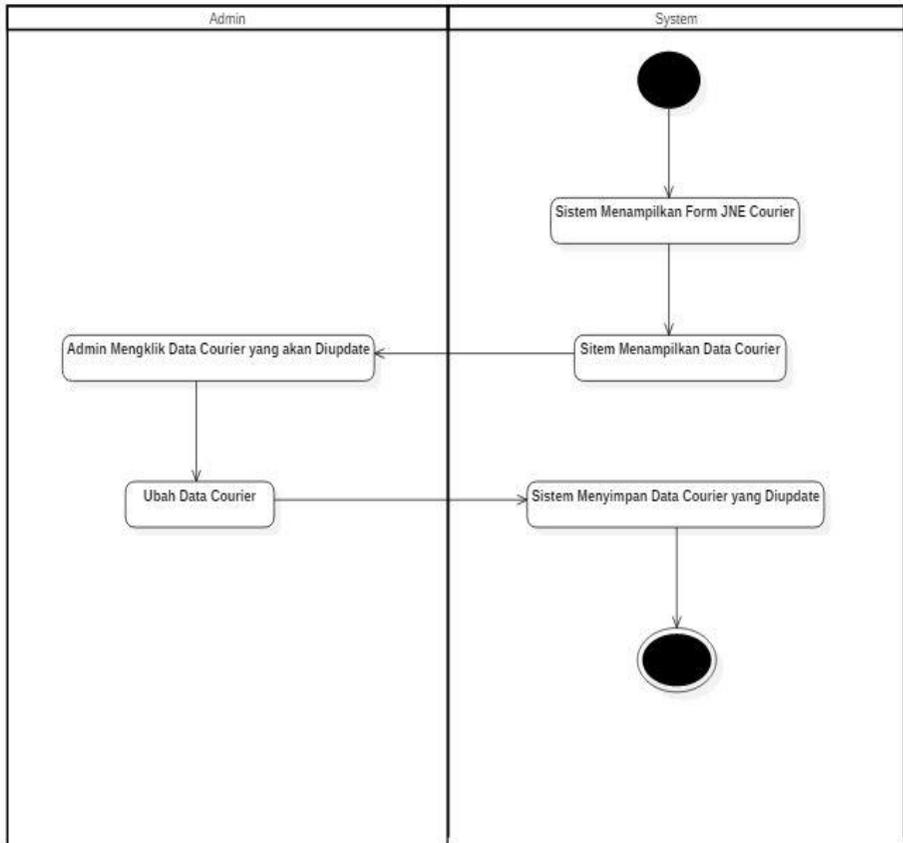
Gambar 4. 37 Activity Diagram Proses Item on Bag

e) Activity Diagram Detail consignment Note



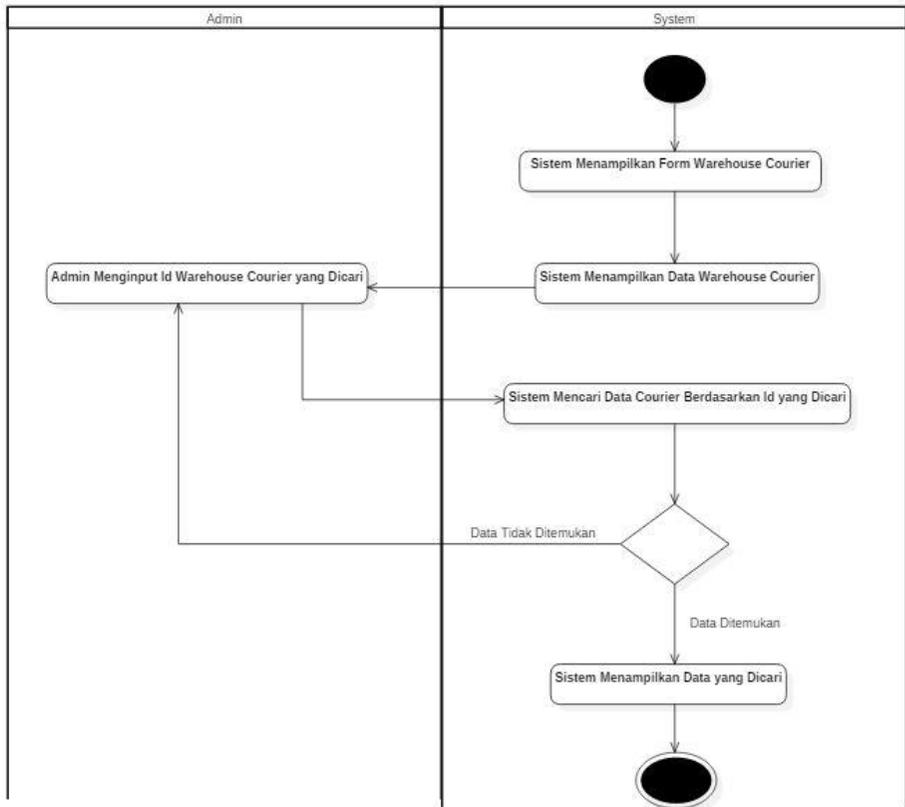
Gambar 4. 38 Activity Diagram Proses Detail consignment Note

f) Activity Diagram JNE Courier Update



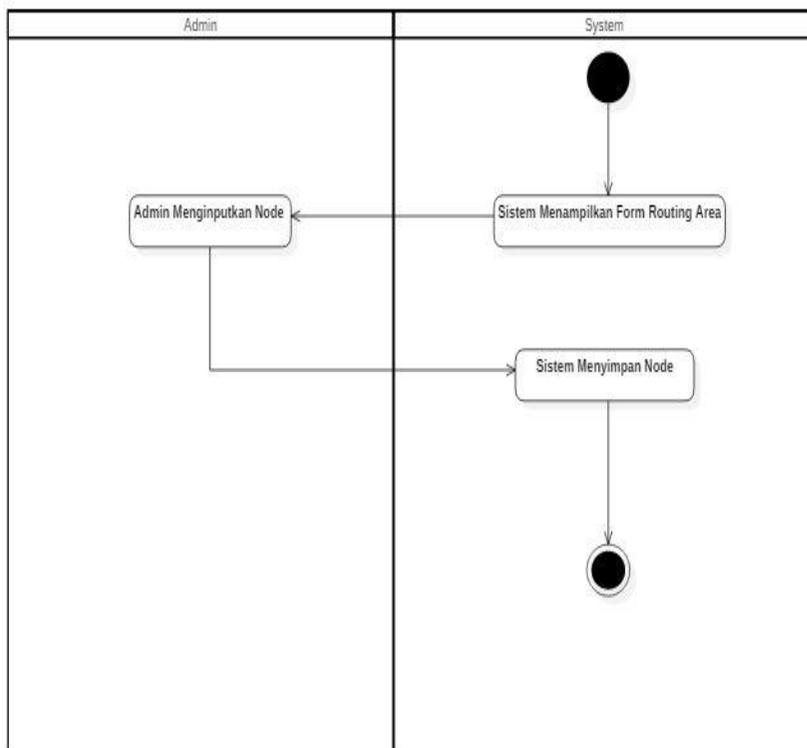
*Gambar 4. 39 Activity Diagram Proses JNE Courier Update*

g) Activity Diagram JNE Warehouse Courier



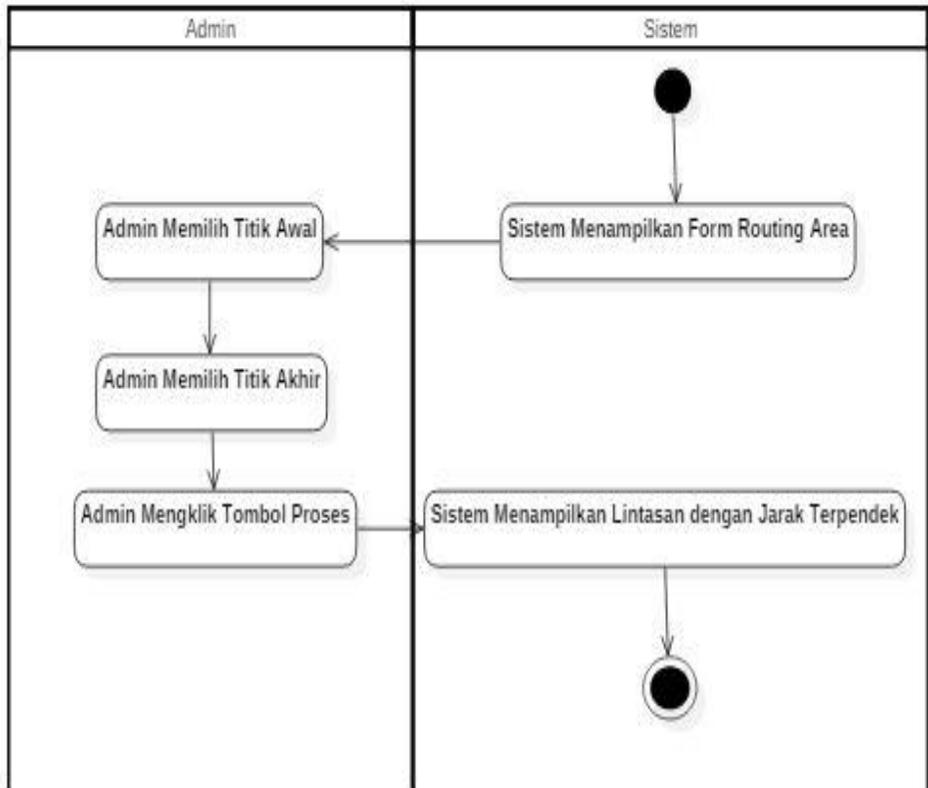
Gambar 4. 40 Activity Diagram Proses JNE Warehouse Courier

### h) Activity Diagram Routing Area Tambah



*Gambar 4. 41 Activity Diagram Routing Area Tambah*

i) Activity Diagram Routing Area Lintasan

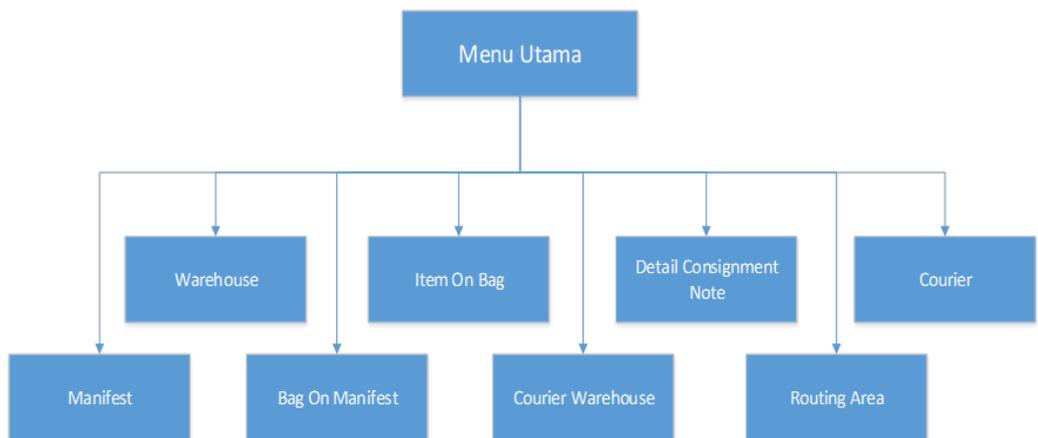


Gambar 4. 42 Activity Diagram Routing Area Lintasan

**d) Perancangan *User Interface***

**1) Struktur menu pada HIPO Chart**

HIPO Chart adalah bagan untuk menentukan urutan menu pada user interface pengguna. Menu di sini diturunkan berdasarkan setiap operasi yang disediakan oleh sistem(aplikasi)



*Gambar 4. 43 HIPO Chart Menu Utama*

**2) Rancangan Input**

**a) Form Input Detail Consignment**

Rancangan form consignment ini berguna untuk menerima dan merekap barang yang masuk yang sudah terdata jelas identitas lengkap pengirim dan tujuan pakatnya. Form ini sebagai proses awal dimana barang masuk ke Gudang.

Gambar 4. 44 Form Input Detail Consignment

b) Rancangan Form Master Courier

Form ini difungsikan untuk mengelola data master kurir dimana banyak kurir yang masuk dan keluar sehingga rotasi pegawai (kurir) ini sering terjadi sehingga perlu pengelolaan pada database aplikasi

	id_kurir	nama_kurir	wilayah	no_plat	phone

Gambar 4. 45 Form Input Data Courier

c) Rancangan Form Input Bag

Rancangan form ini difungsikan untuk menginput data bag (paket barang yang terkelompok) yang masuk dari perusahaan pengirim (sumber barang) ke gudang. Bag bisa saja berisikan puluhan hingga ratusan paket di dalamnya



ID Manifest

	Id_manifest	date	Destination	Total_bag

Gambar 4. 48 Output manifest

b) Rancangan output courier warehouse

Tampilan rancangan output pengelolaan laporan kurir atau list kurir yang terdaftar yang nantinya akan diupdate berdasarkan kurir yang masuk atau keluar.

ID Courier

	Id_kurirgdg	Nama_kurir	Id_bag	dateofpick	phone	Address

Gambar 4. 49 Output courier warehouse

c) Rancangan output *Bag on Manifest*

Rancangan untuk pengelolaan laporan bag yang masuk dimana di dalam *bag* inilah paket-paket dikelompokkan dan memberikan informasi darimana asal dan tujuannya.

ID Manifest

	Id_manifest	date	Destination	Origin

Gambar 4. 50 Output Bag on Manifest

d) Rancangan output *detil consignment note*

Rancangan laporan ini lah yang utama sekali dan dari laporan ini juga yang akan dikelola secara FIFO, sehingga bag yang pertama datang, maka akan segera dikelompokkan dan didistribusikan ke penerimanya

ID Bag

List Of Item

	Id Resi	Nama Cabang	Address	Destination_kec	City_tujuan	Phone

Gambar 4. 51 Output *detil consignment note*

### 3. Testing dan Implementasi

#### a) Kebutuhan perangkat lunak

Adapun kebutuhan perangkat lunak dalam menerapkan aplikasi jasa layanan distribusi paket online shop ini diantaranya adalah:

- 1) Bahasa Pemrograman Visual Basic 2010
- 2) Crystal Report
- 3) XAMPP versi 7.3.1

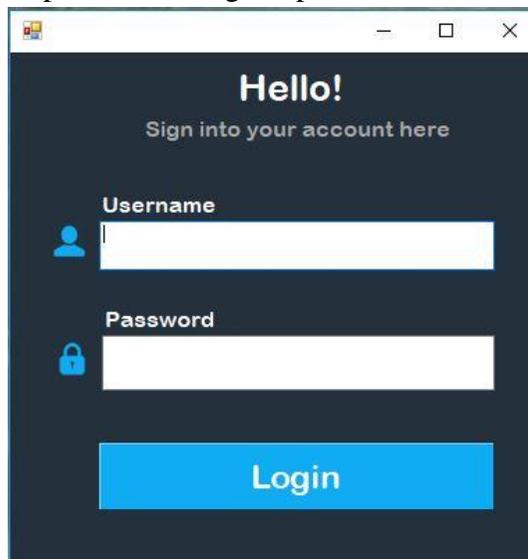
#### b) Kebutuhan perangkat keras

Untuk kebutuhan akan perangkat keras untuk menerapkan dan menguji aplikasi ini agar dapat *running well* diantaranya kebutuhan akan

- 1) PC dengan spesifikasi:
  - a) Processor minimal Core i3
  - b) RAM minimal 2GB
  - c) Hardisk 500GB
  - d) Perangkat *networking*
  - e) Alat cetak Printer

#### c) Implementasi

- 1) Implementasi Login Aplikasi



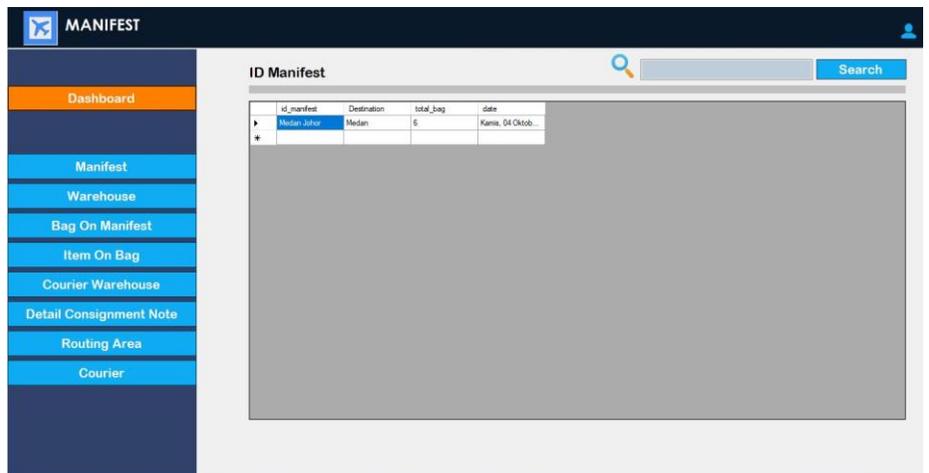
Gambar 4. 52 Tampilan halaman login

## 2) Halaman Dashboard Administrator



Gambar 4. 53 Tampilan halaman administrator

## 3) Tampilan Halaman Manifest



Gambar 4. 54 Tampilan halaman manifest

#### 4) Implementasi Input Bag on manifest

The screenshot shows a web application interface for 'Bag On Manifest'. On the left is a vertical navigation menu with items: Dashboard, Manifest, Warehouse, Bag On Manifest, Item On Bag, Courier Warehouse, Detail Consignment Note, Routing Area, and Courier. The main area contains a form with the following fields: ID Bag (text input), Date (calendar dropdown showing 'Minggu, 28 Oktober 2018'), ID Manifest (text input), Origin (text input), Destination (text input), and Weight (text input). At the bottom of the form are two buttons: 'Save' and 'Lihat Data'. A search bar is located at the top right of the main area.

Gambar 4. 55 Tampilan halaman Input Bag on manifest

#### 5) Implementai Tampilan output Bag on manifest

The screenshot shows the 'Bag On Manifest' output view. It features a table with the following data:

id_bag	tanggal	id_manifest	Origin	destination	weight
5Y10376611	29/09/2018	Medan Johor	Surabaya	Medan	4.00
CK11475512	02/10/2018	Medan Johor	Jakarta	Medan	2.00

Below the table is a large grey rectangular area, likely a placeholder for additional content or a scrollable list. The interface includes a search bar and a 'Search' button at the top right.

Gambar 4. 56 Tampilan halaman output Bag on manifest

## 6) Implementasi Proses *Item on Bag*

id_resi	nama_cabang	address	destination_jec	city_sajian	phone	nama_pemgim	alamat_pemgim	city	phone
1092111256764	JNE Lamongan	Jl.KH Ahmad Da...	Medan Johor		0322-77127-73	Toko Family Fash	Jl.Dokter Wahidi...	Lamongan	032232
1092111256785	JNE Jember	Jl.Sumatra No. 67	Medan Johor	Medan	0331-3238-66	Biru Daun	Jl.Anggrek No.8	Jember	08564
1092111256783	JNE Surabaya	Ruko Surya Inti P...	Medan Johor	Medan	031-86738-85	Baju Poko Surab	Jl.Tanah Merah ...	Surabaya	08122
1092111256771	JNE Aguna Sura...	Jl.Aguna No. 151	Medan Johor	Medan	061-1299-133	Meehoes Store	Jl.Gatot Subroto ...	Surabaya	08135
1092111256778	JNE Semarang	Jl.Sultan Agung ...	Medan Johor	Medan	024-85040-13	Indo Fashion	Jl.Tlogosan Raya...	Semarang	02467
1092111256772	JNE Srengaya/BOG	Jl.Srengaya No.30	Medan Johor	Medan	081-1239-191	Dieno Bandung	Jl.Kemang No.38	Bandung	08226
1092111256774	JNE Beger	Jl.Salah Iskandar...	Medan Johor	Medan	0251-86563-41	Kandang Unchal	Jl.Cerama Ungu...	Buger	02516
1092111256779	JNE Magelang	Jl.Gatot Subroto...	Medan Johor	Medan	0293-55260-26	Hebu Store	Jl.Tentara Pelajar	Magelang	02933
1092111256781	JNE Boyolali	Jl.Pahlawan No. 26	Medan Johor	Medan	0276-3218-10	An-Naas Busana	Jl.Pandaranan N...	Boyolali	08129
1092111256777	JNE Solo	Jl.Laksda Adi Su...	Medan Johor	Medan	0271-7212-56	Klaben	Jl.Mayor Sunaryo...	Solo	08157
1092111256782	JNE Yogyakarta	Jl.Songomen No...	Medan Johor	Medan	0274-4142-92	Sakola Joga	Jl.Kaptan Piere T...	Yogyakarta	02743
1092111256773	JNE Jakarta	Jl.Tomang Raya ...	Medan Johor	Medan	021-5665262-63	Groar Dress	Jl.Danau Indah ...	Jakarta	08708
1092111256776	JNE Banjar	Komp. Ruko Ter...	Medan Johor	Medan	0265-70319-90	Nefa Collection	Jl.Birgeman. Iea. S...	Banjar	08112
1092111256775	JNE Bandung	Jl.Pemata Kawal...	Medan Johor	Medan	022-7335454-55	Groar Kaos Distr	Jl.Ciwatno No.11	Bandung	08111

Gambar 4. 57 Tampilan list item on bag

## 7) Impementasi Kelola Courier Warehouse

id_kurirgud	nama_kurr	id_bag	dateofpick	id_warehouse
KG001	Adipati Syaputra	BD12304421	Kems, 04 Oktob...	JHR19912
KG002	Badi Sunaryo	CK11475511	Kems, 04 Oktob...	JHR19912

Gambar 4. 58 Tampilan list dan kelola courier warehouse

## 8) Implementasi Kelola *Detail Consignment*

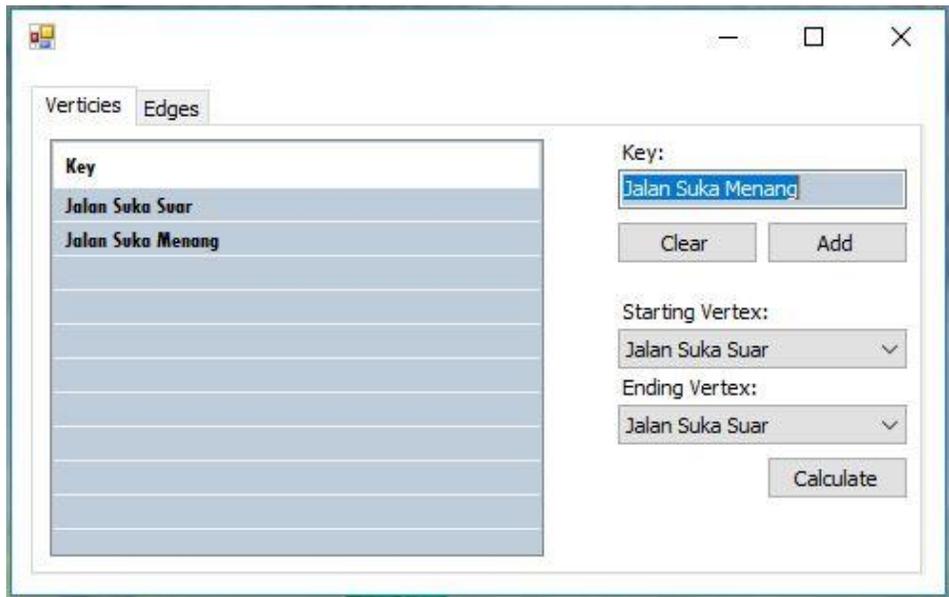
Gambar 4. 59 Tampilan kelola *Detail Consignment*

## 9) Implementasi Data Courier

id_kurir	nama_kurir	id_resi	wilayah	no
KJ001	Firmanyah Adh S...	109211256771	Medan Johor	BK
KJ001	Firmanyah Adh S...	109211256783	Medan Johor	BK
KJ001	Firmanyah Adh S...	109211256782	Medan Johor	BK
KJ001	Firmanyah Adh S...	109211256781	Medan Johor	BK
KJ001	Firmanyah Adh S...	109211256779	Medan Johor	BK
KJ001	Firmanyah Adh S...	109211256780	Medan Johor	BK
KJ001	Firmanyah Adh S...	1102111796779	Medan Johor	BK

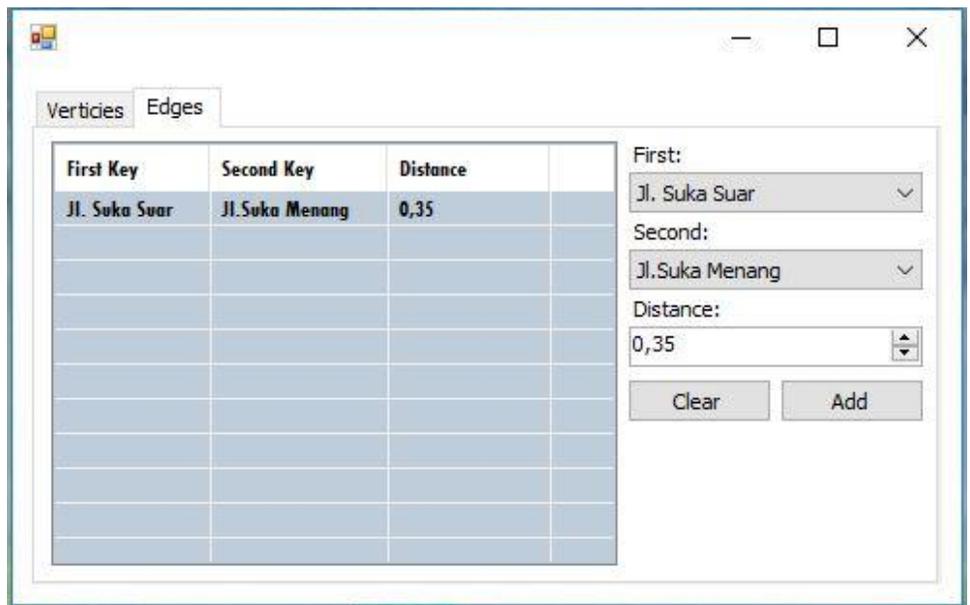
Gambar 4. 60 Tampilan kelola *Data Kurir*

## 10) Implementasi Routing Area (*Vertices*)



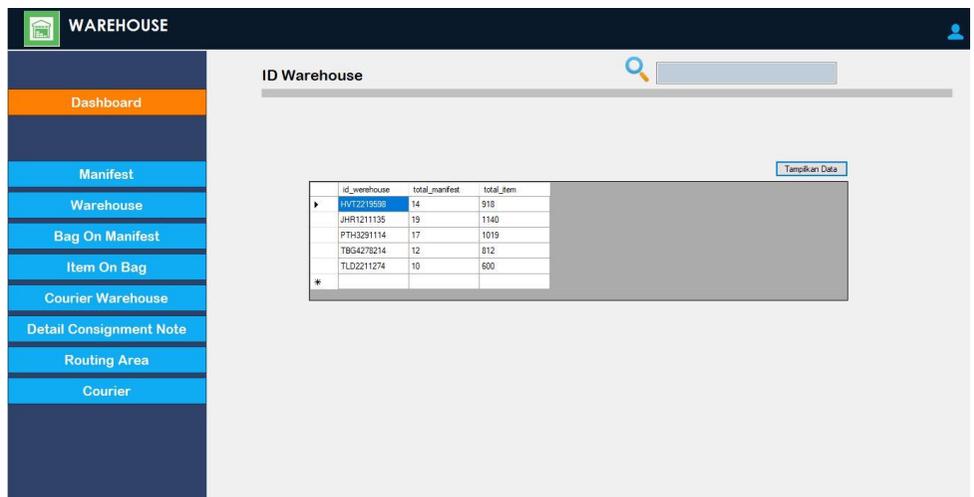
*Gambar 4. 61 Tampilan Routing Area Note to Note*

## 11) Implementasi Routing Area (*Edges*)



Gambar 4. 62 Tampilan Routing Area Distance

## 12) Impementasi Kelola Warehouse



Gambar 4. 63 Tampilan Kelola Warehouse

## BAB V

### PENUTUP

#### A. Kesimpulan

Berdasarkan penelitian dan pembahasan yang ketua dan tim penelitian lakukan, dapat ditarik kesimpulan sebagai berikut :

1. Metode *First In First Out* (FIFO) sangat diperlukan oleh pihak jasa layanan distribusi paket dalam pengelolaan barang yang masuk dan barang yang keluar pada bagian gudang(*warehouse*) dikarenakan jumlah barang yang sangat multi dan banyak dalam bag yang sudah diterima. Biasanya *bag* yang masuk sifatnya gelondongan sehingga perlu proses identifikasi barang mana yang harus didahulukan dalam distribusi kepada konsumen tujuannya
2. Setelah barang teridentifikasi maka, barang yang akan didistribusikan merupakan tanggung jawab kurir. Dan dengan penyelesaian masalah kurir dengan TSP (*Travelling Salesman Problem*) maka kurir akan dipermudah dalam identifikasi ke tujuan mana paket akan dikirimkan terlebih dahulu, dan dari setiap node(tujuan) paket yang satu ke yang lainnya
3. Aplikasi ini dibangun dengan baik dengan memanfaatkan pemrograman visual basic.net 2010 yang dilengkapi dengan GUI yang user friendly dan ditunjang dengan pengelolaan database menggunakan MySQL (Maria db) yang memanfaatkan package XAMPP versi terbaru.

#### B. Penutup

Dengan selesainya penelitian ini dilaksanakan hingga tahap akhir dan laporan dihasilkan, maka:

1. Aplikasi jasa layanan distribusi paket *online shop* ini dirasakan sangat bermanfaat bagi perusahaan dalam hal ini PT.JNE Express terutama dalam menyelesaikan masalah penentuan jarak tempuh dan pemilihan jalur terbaik dalam proses pengiriman paket yang dilakukan oleh kurir.
2. Penulis menyadari masih perlunya penyempurnaan dalam aplikasi yang dirancang ini dan aplikasi benar-benar bisa

- dipakai oleh perusahaan JNE dalam menyelesaikan masalah mereka terhadap layanan pendistribusian paket *online shop*
3. Penelitian ini layak untuk tetap dikembangkan dengan memanfaatkan metode-metode baru lainnya dalam menghasilkan luaran yang lebih maksimal
  4. Aplikasi jasa layanan distribusi paket *online shop* saat ini masih berbentuk aplikasi desktop namun juga nantinya harus bisa dikembangkan dalam layanan secara *online* dan bisa bermanfaat di seluruh kantor cabang dan gudang perusahaan yang tersedia

## DAFTAR REFERENSI

- [1] Oktapiani, Renny. 2016. *Perancangan Sistem Persediaan Barang Menggunakan Metode FIFO Pada PT. Panjunan Sukaraja Sukabumi*. Konferensi Nasional Ilmu Sosial & Teknologi (KNiST) Maret 2016
- [2] Maisak, Despita. 2017. *Analisis Dan Perancangan Sistem Informasi Persediaan Barang Menggunakan Metode FIFO Pada PT. Shukaku*. MEDIASISFO Vol. 11, No. 2, Oktober 2017
- [3] Fitria dan Apri. 2013. *Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan*. Jurnal Sistem Informasi (JSI), VOL. 5, NO. 2, Oktober 2013
- [4] Wardhani, Nurmaliyanti. 2014. *Optimasi Traveling Salesman Problem (TSP) Menggunakan Algoritma Semut*. JURNAL IT STMIK HANDAYANI, Volume 15, Desember 2014
- [5] Yuni Sugiarti, (2013). *“Analisis Dan Perancangan UML (Unified Modeling Language) Generated VB. 6.”* Bandung : Graha Ilmu
- [6] Jakimi dan M. L Khoutbi, (2009). *“An Object-Oriented Approach to UML Scenarios Engineering and Code Generation*
- [7] Sholiq, (2006). *“Pemodelan Sistem Informasi Berorientasi Objek Dengan UML.”* Bandung : Graha Ilmu. hal 61
- [8] Edwin, Jemmy. 2012. *Model Antrian FIFO (First – In First-out) Pada Pelayanan Mahasiswa Fakultas Teknik Universitas Jandabra Berbasis Multimed*i. Jurnal Teknik Vol. 2 No. 2 / Oktober 2012
- [9] Puspita, dkk. 2016. *Penerapan Teori Antrian Pada Pelayanan Teller Bank X Kantor Cabang Pembantu Putri Sentra Niaga*. JURNAL GAUSSIAN, Volume 6, Nomor 1, Tahun 2016