



Available online at : <http://bit.ly/InfoTekJar>

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Implementasi Algoritma *Ant Colony Optimization* (ACO) Pada Pencarian Jalur Terpendek *Automatic Teller Machine* (ATM) Di Kota Palu

Deny Wiria Nugraha, Amriana, Rieska Setiawati

Program Studi Teknik Informatika Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako Palu, Sulawesi Tengah, Indonesia, 94118

KEYWORDS

Android, *Ant Colony Optimization* (ACO), Jalur Terpendek, *Auto Matic Teller Machine* (ATM)

CORRESPONDENCE

Phone: +62 85256192194

E-mail: deny.wiria.nugraha@gmail.com

A B S T R A C T

Penelitian ini bertujuan untuk membangun sebuah sistem yang dapat memberikan informasi lokasi ATM di Kota Palu, dan sekaligus memberikan petunjuk jalur terpendek dengan menggunakan algoritma *Ant Colony Optimization* (ACO) system (ACS), yang merupakan variasi algoritma *Ant Colony Optimization* (ACO), dalam mencari makanan setiap setiap semut akan berusaha mencari jalur terpendek dari sarang ke tempat makanan. Kemudian semut tersebut akan meninggalkan pheromone di jalur yang dilaluinya. Pada proses awal algoritma ini adalah menginisialisasi penggunaan parameter yang tepat sesuai kasus yang akan diselesaikan. Pada proses ini semut akan memilih melakukan eksploitasi atau eksplorasi rute yang akan di kunjungi. Eksploitasi berarti semut hanya akan mengunjungi ruas-ruas simpul yang memiliki pheromone yang tinggi dengan bobot jarak yang kecil sedangkan eksplorasi berarti semut bias saja mengunjungi ruas ruas simpul yang memiliki pheromone yang rendah dengan bobot jarak besar. Proses perhitungan ACO di implementasikan ke dalam sistem yang telah di buat menggunakan aplikasi Android studio versi 3.3 dengan memanfaatkan prosedur algoritma ACO, penelitian ini menggunakan data sebanyak 60 lokasi ATM di kota Palu.

INTRODUCTION

Seiring berjalannya waktu, kebutuhan masyarakat *modern* akan teknologi dengan mobilitas yang tinggi semakin meningkat. Khususnya dalam keperluan manajemen finansial. Segala transaksi tidak luput dari peran mesin *Automatic Teller Machine* (ATM) sebagai pendukung terciptanya peningkatan finansial seseorang atau badan usaha. Dan Memberikan layanan yang terbaik bagi para nasabah merupakan prioritas utama setiap bank, Salah satu layanannya adalah dengan menempatkan mesin ATM di setiap sisi kota yang nantinya dapat memudahkan nasabah dalam melakukan transaksi.

Salah satu permasalahan utama bagi nasabah suatu Bank adalah jalur yang harus mereka tempuh. Hal ini disebabkan oleh banyaknya jumlah ATM dan jalur alternatif yang ada. Sehingga, tidak sedikit waktu dan biaya yang dibutuhkan untuk mencari mesin ATM yang ada serta jalur yang harus ditempuh. Penentuan lokasi dengan Sistem Informasi Geografi (SIG) dapat menggambarkan informasi jarak dan tujuan sesuai dengan keinginan pengguna.

Saat ini Aplikasi android yang tersedia untuk pengguna *Global Positioning System* (GPS) sebagian besar belum memiliki fungsi lengkap. Di aplikasi pencarian mesin ATM terdekat ini misalnya, aplikasi android yang berfungsi untuk membantu kita menemukan mesin ATM terdekat dari posisi pengguna

berdasarkan GPS. Dimana melihat secara garis besar bahwa masyarakat membutuhkan aplikasi yang dapat membantu mereka untuk mencari lokasi berdasarkan posisi mereka saat ini, terutama ketika mereka sedang berpergian ke suatu tempat wisata atau tempat yang belum pernah mereka kunjungi sebelumnya. Saat ini pengguna android di Indonesia terus bertambah dan luas sudah menjadi mayoritas dibandingkan platform lain, serta pengguna android tersebut memiliki ketertarikan terhadap aplikasi pencarian mesin ATM.

Adapun tujuan yang ingin dicapai oleh penulis dalam penelitian ini adalah untuk mengimplementasikan algoritma *Ant Colony Optimization* (ACO) pada pencarian jalur terpendek *Automatic Teller Machine* (ATM) di Kota Palu berbasis android.

Berdasarkan permasalahan yang ada, maka penulis merumuskan masalah dalam penelitian ini, sebagai berikut:

Bagaimana merancang dan membangun aplikasi pencarian rute terpendek lokasi mesin ATM di Kota Palu? Dan Bagaimana algoritma *Ant Colony Optimization* (ACO). dapat diimplementasikan pada pencarian rute terpendek lokasi *Automatic Teller Machine* (ATM) ?

Adapun manfaat dari penelitian ini adalah :

- a) Sisi akademis
 1. Dapat menerapkan dan memperdalam pengetahuan tentang bidang informatika.

2. Dapat dijadikan bahan referensi bagi penelitian-penelitian berikutnya yang memiliki topik yang sama dengan penelitian ini.
- b) Sisi praktis
1. Menghasilkan sebuah aplikasi pencarian jalur terpendek lokasi ATM yang ada di kota palu.
 2. Dapat mempermudah pengguna aplikasi dalam pencarian lokasi ATM di kota palu.

Penelitian yang sebelumnya pernah dilakukan sehubungan dengan pencarian jalur terpendek antara lain adalah sebagai berikut:

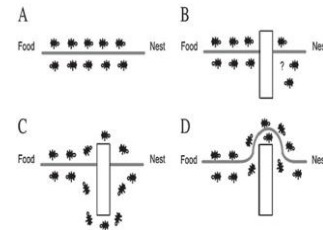
1. Dalam penelitiannya, Verdianto (2014) perancangan sistem penentuan rute terpendek jalur evakuasi tsunami dengan algoritma ant colony. Pada penelitian tersebut menentukan jalur terpendek evakuasi tsunami berbasis desktop. Persamaan penelitian tersebut dengan penelitian yang akan dilakukan oleh penulis yaitu terletak pada penerapan algoritma yang digunakan. Adapun perbedaan penelitian sebelumnya aplikasi yang dibuat berbasis desktop dengan studi kasus penentuan rute terpendek jalur evakuasi tsunami di Kota Medan sedangkan penelitian yang dilakukan penulis berbasis android pencarian rute terpendek ATM di Kota Palu.
2. Dalam penelitiannya, Yuliana (2014) membuat pencarian rute terpendek trans Jogja dengan menggunakan algoritma *Dijkstra*. Penelitian tersebut bertujuan untuk merancang sebuah aplikasi pencarian rute terpendek Trans Jogja berbasis *web* yang dapat memvisualisasikan peta lokasi *shelter* Trans Jogja agar dapat dimanfaatkan oleh para pendatang dan wisatawan yang bingung dalam memilih shelter dan rute yang tepat. Persamaan yang terdapat dalam penelitian yang dilakukan Yuliana (2013) dengan penelitian ini yaitu sama - sama melakukan pencarian rute terpendek sedangkan perbedaan penelitian ini dengan yang dilakukan penulis terdapat pada kasus pencarian lokasi dan jalur terpendek ATM menggunakan algoritma *Ant Colony Optimization (ACO)*
3. Dalam penelitiannya Hartantyo (2014) membuat aplikasi pencarian lokasi rumah sakit dan puskesmas di wilayah Tegal berbasis *Android*. Penelitian ini bertujuan untuk membangun sebuah aplikasi pencarian rumah sakit dan puskesmas di wilayah Tegal untuk membantu masyarakat dan wisatawan dalam mengetahui informasi tentang lokasi rumah sakit dan puskesmas yang ada di wilayah Tegal. Persamaan penelitian ini dengan yang sedang dilakukan penulis terdapat pada lokasi yang ingin dituju sedangkan perbedaannya terdapat pada metode pencarian rute terpendek, dimana penulis menggunakan algoritma *Ant Colony Optimization (ACO)* sedangkan Hartantyo (2014) menggunakan *Google Maps direction*.

LANDASAN TEORI

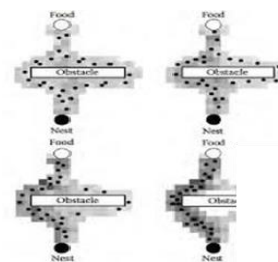
a. Algoritma Ant Colony Optimization (ACO)

Algoritma Ant Colony Algoritma semut diperkenalkan oleh Moysen dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo. Algoritma semut adalah *bioinspired metaheuristic*, mempunyai sekelompok khusus yang berusaha

menyamai karakteristik kelakuan dari serangga sosial, yaitu koloni semut. Kelakuan dari tiap pelaku dalam meniru kelakuan dari semut hidup dan bagaimana mereka berinteraksi satu dengan lainnya agar dapat menemukan sumber makanan dan membawanya ke koloni mereka dengan efisien. Selama berjalan tiap semut mengeluarkan *pheromone*, dimana semut lainnya sensitif dengan *pheromone* tersebut sehingga memberikan harapan untuk mengikuti jejaknya. Lebih atau kurang intensitasnya tergantung pada konsentrasi dari *pheromone*. Adapun contoh pembangunan jejak *pheromone* dalam mencari jalan yang lebih pendek dapat dilihat pada gambar 2.1 dan perubahan konsentrasi *pheromone* dapat dilihat pada gambar 2.2



Gambar 1 pembangunan jejak pheromone
(Sumber : Karjono *et al* 2016)



Gambar 2 perubahan konsentrasi pheromone
(Sumber: Karjono *et al* 2016)

b. Ant Colony system (ACS)

Ant Colony System (ACS) adalah sebuah metodologi yang dihasilkan melalui pengamatan terhadap semut. Pada algoritma ACS, semut berfungsi sebagai agen yang ditugaskan untuk mencari solusi terhadap suatu masalah optimisasi. ACS telah diterapkan dalam berbagai bidang, salah satunya adalah untuk mencari solusi optimal pada *Traveling Salesman Problem (TSP)*. Dengan memberikan sejumlah n titik, TSP dapat didefinisikan sebagai suatu permasalahan dalam menemukan jalur terpendek dengan mengunjungi setiap titik yang ada hanya sekali.

c. Karakteristik Ant Colony System (ACS)

Terdapat tiga karakteristik utama dari ACS, yaitu aturan transisi status, aturan pembaruan pheromone local dan aturan pembaruan pheromone global.

1. Aturan Transisi Status

Aturan transisi status yang berlaku pada ACS adalah sebagai berikut: seekor semut yang ditempatkan pada titik t memilih untuk menuju ke titik v , kemudian diberikan bilangan pecahan acak q dimana $0 \leq q \leq 1$, q_0 adalah sebuah parameter yaitu Probabilitas semut melakukan eksplorasi pada setiap tahapan, dimana ($0 \leq q_0 \leq 1$) dan $p_k(t, v)$ adalah probabilitas dimana semut k memilih untuk bergerak dari titik t ke titik v . Jika $q \leq q_0$ maka

pemilihan titik yang akan dituju menerapkan aturan yang ditunjukkan oleh persamaan (1).

$$\begin{aligned}
 \text{temporary}(t,u) &= [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta, i = 1,2,3 \dots, n \\
 v &= \\
 \max\{[\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta\} &\dots\dots\dots(1)
 \end{aligned}$$

dengan v = titik yang akan dituju sedangkan jika $q > q_0$ digunakan persamaan (2).

$$v = p_k(t,v) = \frac{[\tau(t,v)] \cdot [\eta(t,v)]^\beta}{\sum_{i=1}^n [\tau(t,u_i)] \cdot [\eta(t,u_i)]^\beta} \dots\dots\dots(2)$$

dengan $\eta(t,u_i) = \frac{1}{\text{Jarak}(t,u_i)}$ dimana $\tau(t,u)$ adalah nilai dari jejak *pheromone* pada titik (t,u) , $\eta(t,u)$ adalah fungsi heuristik dimana dipilih sebagai invers jarak antara titik t dan u , β merupakan sebuah parameter yang mempertimbangkan kepentingan relatif dari informasi *heuristic*, yaitu besarnya bobot yang diberikan terhadap parameter informasi heuristik, sehingga solusi yang dihasilkan cenderung berdasarkan nilai fungsi matematis.

2. Aturan Pembaruan *Pheromone* Lokal

Selagi melakukan tur untuk mencari solusi dari TSP, semut mengunjungi ruas-ruas dan mengubah tingkat *pheromone* pada ruas-ruas tersebut dengan menerapkan aturan pembaruan *pheromone* lokal yang ditunjukkan oleh persamaan (3).

$$\begin{aligned}
 \tau(t,v) &\leftarrow (1 - \rho) \cdot \tau(t,v) + \\
 \rho \cdot \Delta\tau(t,v) &\dots\dots\dots(3) \\
 \Delta\tau(t,v) &= \gamma \cdot \max_{z \in J_k(v)} \tau(V,Z)
 \end{aligned}$$

dimana

- ρ = Parameter dengan nilai 0 sampai 1
- $\Delta\tau$ = Perubahan *pheromone*
- γ = Parameter dengan nilai $0 \leq \gamma < 1$

$$\max_{z \in J_k(v)} \tau(V,Z) = \text{Nilai } \textit{pheromone} \text{ tertinggi}$$

diantara titiktitik yang terhubung ke titik awal pencarian adalah sebuah parameter (koefisien evaporasi), yaitu besarnya koefisien penguapan *pheromone*.

3. Aturan Pembaruan *Pheromone* Global

Pada sistem ini, pembaruan *pheromone* secara global hanya dilakukan oleh semut yang membuat tur terpendek sejak permulaan percobaan. Pada akhir sebuah iterasi, setelah semua semut menyelesaikan tur mereka, sejumlah *pheromone* ditaruh pada ruas-ruas yang dilewati oleh seekor semut yang telah menemukan tur terbaik (ruas-ruas yang lain tidak diubah). Tingkat *pheromone* itu diperbarui dengan menerapkan aturan pembaruan *pheromone* global yang ditunjukkan oleh persamaa $\tau(t,v) \leftarrow (1 - \alpha) \cdot \tau(t,v) + \alpha \cdot \Delta\tau(t,v) \dots\dots\dots(4)$

$$\Delta\tau(t,v) = \begin{cases} L_{gb}^{-1} & \text{jika } (t,v) \in \textit{tur_terbaik} \\ 0 & \text{lainnya} \end{cases}$$

Dimana :

$\tau(t,v)$ = Nilai *pheromone* akhir setelah mengalami pembaharuan lokal

L_{gb} = Panjang jalur terpendek pada akhir siklus

α = Parameter dengan nilai antara 0 sampai 1

$\Delta\tau$ = Perubahan *pheromone*

$\Delta\tau(t,v)$ bernilai $1/L_{gb}$ jika ruas (t,v) merupakan bagian dari rute terbaik namun jika sebaliknya $\Delta\tau(t,v) = 0$. α adalah tingkat kepentingan relatif dari *pheromone* atau besarnya bobot yang

diberikan terhadap *pheromone*, sehingga solusi yang dihasilkan cenderung mengikuti sejarah masa lalu dari semut dari perjalanan sebelumnya, dimana nilai parameter α adalah ≥ 0 , dan L_{gb} adalah panjang dari tur terbaik secara global sejak permulaan percobaan. Pembaruan *pheromone* global dimaksudkan untuk memberikan *pheromone* yang lebih banyak pada tur-tur yang lebih pendek. Persamaan (3) menjelaskan bahwa hanya ruas-ruas yang merupakan bagian dari tur terbaik secara global yang akan menerima penambahan *pheromone*.

d. *Android Studio*

Android Studio merupakan IDE untuk android yang diperkenalkan Mei 2013 di acara *Google I/O developers*, dan dimaksudkan sebagai alternatif dari *Eclipse*. *Android Studio* didasarkan pada *Java IDE* yang disebut *IntelliJ*, semua produk *IntelliJ* berbagi IDE shell yang sama, yang akan kita lihat setelah membuka *Android Studio*. *Android Studio* ini dikemas dengan *shortcut* untuk *keyboard* yang baik sehingga dapat digunakan dengan lebih produktif daripada *Android development* sebelumnya, kecuali untuk *IntelliJ* yang merupakan dasar *Android Studio* (Wolfson, 2013).

e. *Automatic Teller Machine (ATM)*

ATM dalam bahasa Indonesia yaitu *Ajungan Tunai Mandiri* atau dalam bahasa Inggris *Automated Teller Machine* adalah sebuah alat elektronik yang melayani nasabah bank untuk mengambil uang dan mengecek rekening tabungan tanpa perlu dilayani oleh seorang teller. ATM juga bisa digunakan untuk penyimpanan uang, atau cek, transfer uang, atau bahkan membeli pulsa telepon seluler.

METODOLOGI PENELITIAN

Bahan dan Alat Penelitian

1. *Bahan*

Bahan dan alat-alat yang dibutuhkan untuk pembuatan aplikasi pencarian jalur terpendek adalah sebagai berikut :

1. Data ATM Kota Palu yang terdiri dari jenis fasilitas ATM seperti transaksi tunai dan non tunai.
2. Data lokasi ATM di Kota Palu yang terdiri dari alamat lengkap, peta dan titik koordinat.

2. *Alat Penelitian*

Alat yang dibutuhkan pada penelitian ini :

1. *Processor Core i5*, 3.10GHz, monitor dengan resolusi 1366x768 dan RAM 4Gb dan kapasitas *HardDisk* 750Gb serta *smartphone* dengan spesifikasi minimum RAM 2Gb dan layar 5 inch, sedangkan untuk implementasi perangkat lunak yaitu XAMPP minimum *Version 5.6.23*, sistem operasi komputer minimum OS *Windows 7* atau *Mac OS 10.12.0* dan aplikasi web *browser* pada komputer maupun *smartphone*.
2. *Perangkat Lunak*
 - a. *Android Studio*
 - b. *Emulator Nox*

Jenis dan Tipe Penelitian

1. Jenis penelitian yang digunakan adalah kualitatif dengan pendekatan deskriptif. Metode penelitian kualitatif merupakan salah satu jenis penelitian yang sistematis, terencana, terstruktur dengan jelas sejak awal hingga desain penelitiannya (Sugiyono, 2014). Berkaitan dengan penelitian yang penulis lakukan, jenis penelitian ini juga yang penulis terapkan di penelitian.

2. Tipe Penelitian

Penelitian ini menggunakan tipe desain penelitian eksperimental. Tipe desain penelitian eksperimental merupakan tipe desain penelitian dengan menguji secara langsung objek penelitian dengan membandingkan objek yang satu dengan objek yang lainnya dalam suatu penelitian. (Sukriyadi, 2016).

Lokasi Penelitian

Penelitian ini mengambil data, informasi dan lokasi ATM yang ada di Kota Palu pada Kantor PT.Swadharma Sarana Informatika Kota Palu.

Jenis Dan Sumber Data Penelitian

Sumber data adalah segala sesuatu yang dapat memberikan informasi mengenai data. Berdasarkan sumbernya, data dibedakan menjadi dua, yaitu data primer dan data sekunder.

1. Data Primer

Data primer yaitu data yang dibuat oleh peneliti untuk maksud khusus menyelesaikan permasalahan yang sedang ditanganinya. Data dikumpulkan sendiri oleh peneliti langsung dari sumber pertama atau tempat objek penelitian dilakukan.

Data primer di penelitian ini yaitu titik koordinat lokasi ATM, yang di ambil pada masing – masing titik lokasinya dengan cara terjun langsung ke lokasi tersebut dan juga informasi – informasi mengenai fasilitas yang tersedia di tiap mesin ATM tersebut.

2. Data sekunder

Data sekunder yaitu data yang telah dikumpulkan untuk maksud menyelesaikan masalah yang sedang dihadapi. Data ini dapat ditemukan dengan cepat. Dalam penelitian ini yang menjadi sumber data sekunder adalah literatur, artikel, jurnal serta situs di internet yang berhubungan dengan penelitian yang dilakukan. Contohnya seperti pencarian informasi melalui halaman *website* dan jurnal.

Teknik Pengumpulan Data

Metode pengumpulan data yang dilakukan yaitu :

1. Observasi

Observasi adalah metode pengumpulan data yang berdasarkan pengamatan suatu penelitian dalam memahami mencari bukti atau hasil berdasarkan waktu tanpa mempengaruhi daerah suatu pengamatan dengan mencatat dan merekam untuk penemuan data analisis.

2. Studi Literatur

Studi literatur, yaitu dengan menggunakan atau mengumpulkan sumber-sumber tertulis dengan cara membaca, mempelajari dan mencatat hal-hal penting yang berhubungan dengan masalah yang sedang dibahas guna memperoleh gambaran secara teoritis. Sumber-sumber yang dimaksud adalah buku, artikel dan jurnal melalui internet.

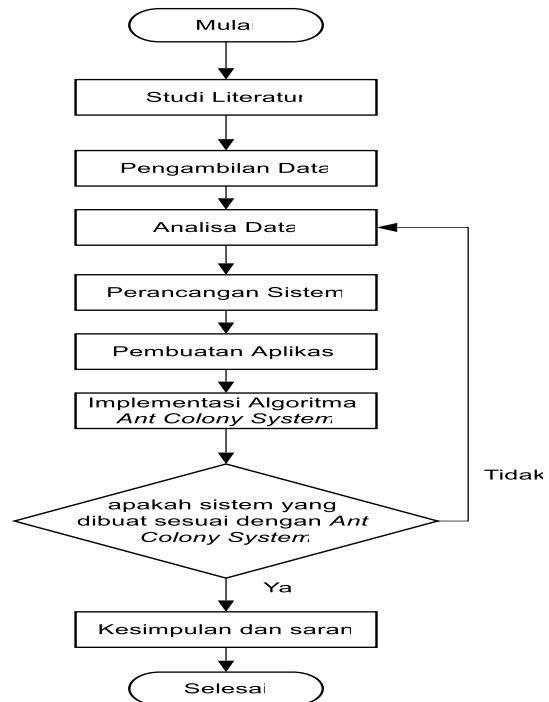
3. Wawancara

Wawancara merupakan percakapan antara dua orang atau lebih dan berlangsung antara narasumber dan pewawancara. Tujuan dari wawancara adalah untuk mendapatkan informasi yang tepat dari narasumber yang

terpercaya. Wawancara dilakukan dengan cara penyampaian sejumlah pertanyaan dari pewawancara kepada narasumber, maka dengan metode wawancara ini pengumpulan data dan informasi data lokasi ATM bisa di dapatkan dengan akurat, narasumber yang diwawancarai seperti pimpinan dan karyawan pada PT. Swadharma Sarana Informatika menanyakan mengenai informasi, data dan lokasi ATM yang ada di Kota Palu.

Tahapan Penelitian

Terdapat beberapa tahapan yang perlu dilakukan agar penelitian ini dapat berjalan dengan baik. Tahapan-tahapan tersebut dibuat dalam bentuk flowchart penelitian. Dapat dilihat pada gambar



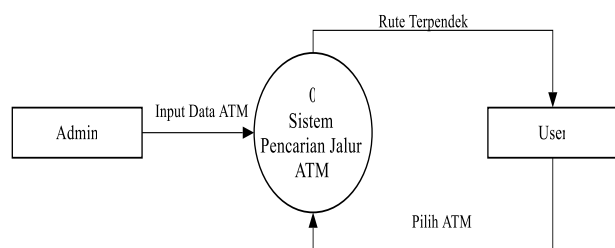
Gambar 3. Flowchart penelitian

HASIL DAN PEMBAHASAN

Analisa Sistem

1. Diagram Konteks

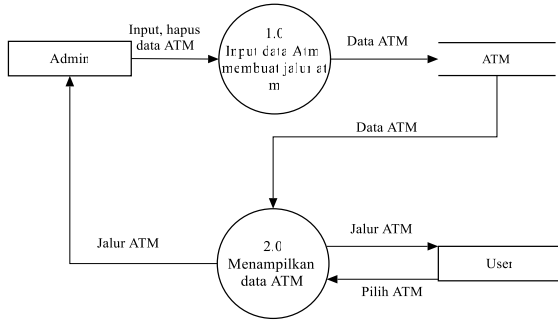
Context diagram menggambarkan hubungan timbal balik dari aktivitas dan reaksi yang diberikan aplikasi (Suryamen dkk, 2016). Skema diagram konteks sistem penentuan jalur terpendek Automatic Teller Machine (ATM) dapat dilihat pada gambar 4.1.



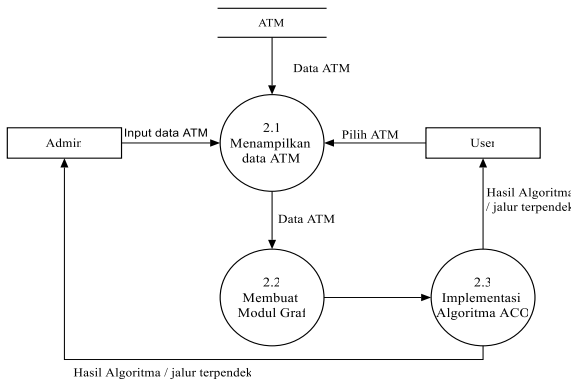
Gambar 4. Context Diagram pencarian jalur terpendek Automatic Teller Machine (ATM) di Kota Palu

1. Data Flow Diagram (DFD)

DFD merupakan salah satu bentuk dokumentasi sistem yang menggambarkan proses dalam sistem itu dan komponen-komponennya, serta arus data atau informasi yang mengalir di antara komponen-komponen tersebut (Adi dan Kristin, 2014). Skema DFD level 1 dan DFD level 2 sistem pencarian jalur terpendek lokasi *Automatic Teller Machine (ATM)* di wilayah kota Palu dapat dilihat masing-masing pada gambar 3.2. dan gambar 3.3.



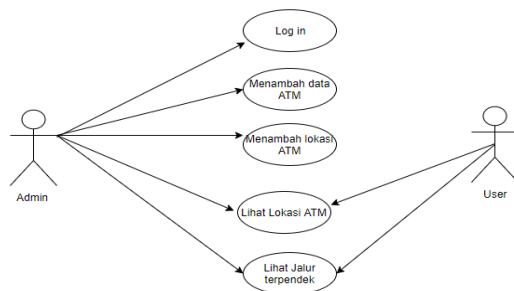
Gambar 5. DFD level 1 pencarian jalur terpendek



Gambar 6. DFD level 2 pencarian jalur terpendek

2. Use Case

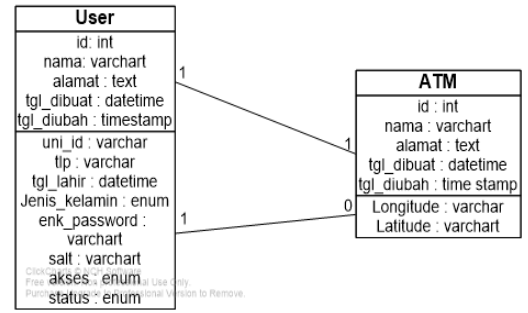
Skema *use case diagram* dalam pencarian jalur terpendek *Automatic Teller Machine (ACO)* dapat dilihat pada gambar 3.4



Gambar 7. Use Case Diagram

3. Class Diagram

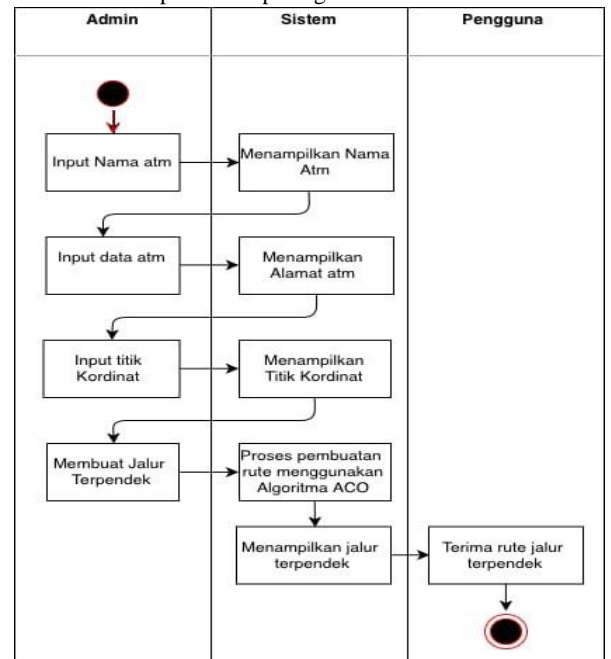
Skema *class diagram* pencarian jalur terpendek dalam pencarian lokasi ATM dapat dilihat pada gambar 3.5.



Gambar 8. Class diagram

4. Activity Diagram

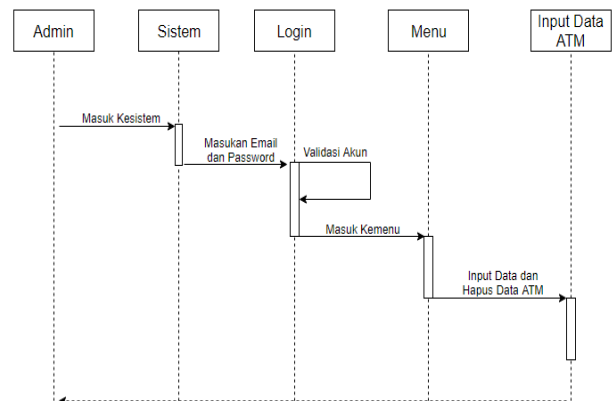
Skema *activity diagram* pencarian jalur terpendek lokasi ATM dapat dilihat pada gambar 3.6.



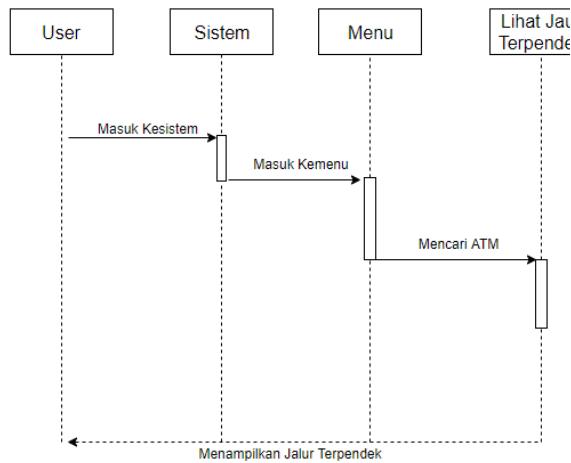
Gambar 9. Activity Diagram

5. Sequence Diagram

Skema *Sequence Diagram* admin dan pencarian jalur terpendek lokasi ATM di kota palu berbasis android dapat di lihat pada gambar 3.7 dan 3.8.



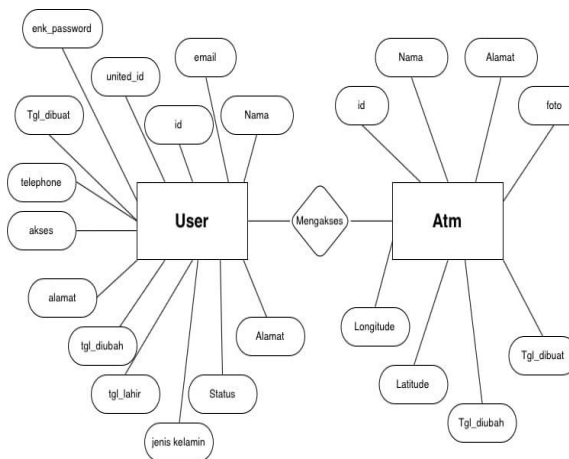
Gambar 10. Squence diagram admin pencarian jalur terpendek lokasi ATM



Gambar 11. Sequence diagram user pencarian jalur terpendek lokasi ATM.

6. Entity Relationship Diagram (ERD)

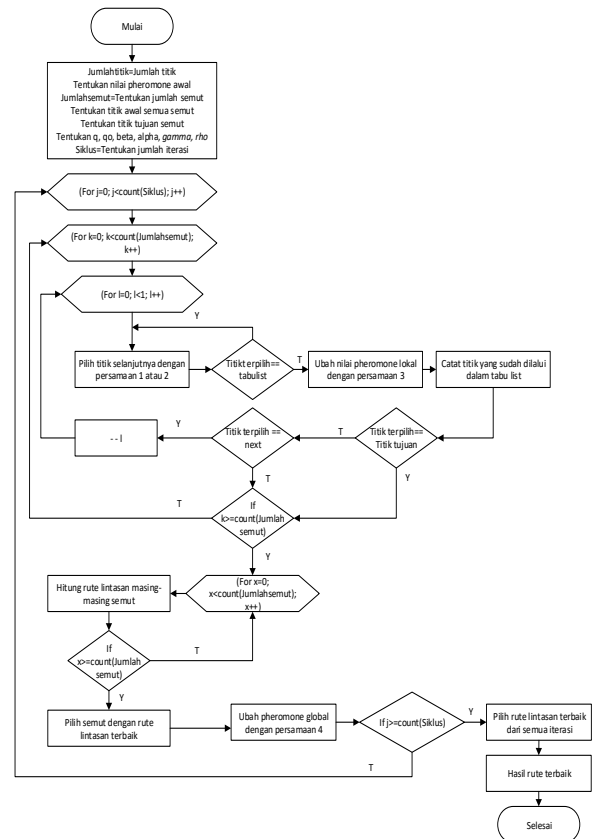
Skema Entity Relationship Diagram (ERD) pencarian jalur terpendek lokasi ATM di kota palu dapat dilihat pada gambar 3.9.



Gambar 12. Entity Relationship Diagram (ERD) sistem pencarian jalur terpendek pencarian lokasi ATM

Implementasi Algoritma

1. Flowchart Algoritma Ant Colony Optimization (ACO)



Gambar 13. Flowchart Algoritma ACO

1. Pseudocode algoritma Ant Colony System (ACS)

Berikut ini adalah pseudocode algoritma ant colony system untuk pencarian jalur terpendek :

1. Fase Inisialisasi
 Jumlah semut = m , jumlah iterasi = max , pheromone , nilai q dan q0 , beta, alpha(α), rho(ρ) dan gamma(γ) .
2. Perulangan iterasi
 (For i=0; i<count(n); i++)
3. Perulangan semut
 (For j=0; j<count(max); j++)
4. Perulangan titik
 (For k=0; k<count(m); k++)
5. Pilih titik menggunakan aturan transisi status menurut persamaan 1 atau 2
6. Update pheromone dengan pembaruan pheromone lokal menurut persamaan 3
7. Simpan titik yang dikunjungi dalam tabu list
8. End for

1. Source code Algoritma Ant Colony Optimization (ACO)

```

}Aco Tsp.java
package com.rieska.atmapp.aco;

import java.io.BufferedWriter;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.Writer;

public class AcoTsp {

    static boolean termination_condition()
    {
        return ((InOut.n_tours >= InOut.max_tours) && (Timer.elapsed_time() >= InOut.max_tim
        (Ants.best_so_far_ant.tour_length <= InOut.optimal));
    }

    static void construct_solutions()
    {
        int k; /* counter variable */
        int step; /* counter of the number of construction steps */

        // TRACE ( System.out.println("construct solutions for all ants\n"); );

        /* Mark all cities as unvisited */
        for (k = 0; k < Ants.n_ant; k++) {
            Ants.empty_memory(Ants.ant[k]);
        }

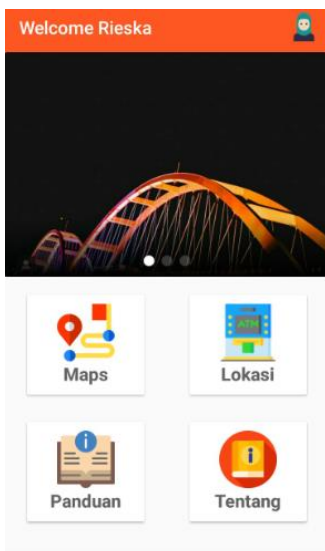
        step = 0;
        /* Place the ants on same initial city */
        for (k = 0; k < Ants.n_ant; k++)
            Ants.place_ant(Ants.ant[k], step);

        while (step < Tsp.n - 1) {
            step++;
            for (k = 0; k < Ants.n_ant; k++) {
                Ants.neighbour_choose_and_move_to_next(Ants.ant[k], step);
                if (Ants.acs_flag)
                    Ants.local_acs_pheromone_update(Ants.ant[k], step);
            }
        }
    }
}
    
```

Gambar 14. Source code Algoritma ACO

Implementasi Sistem

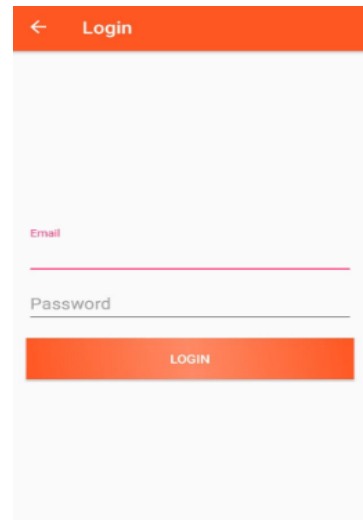
1. Implementasi Input
a. Form Menu Utama



Gambar 15. Form Menu Utama

a. Form Login Admin

Pada form ini admin dapat melakukan login dengan mengisi username yang telah terdaftar agar admin dapat melihat titik kerusakan pada peta. Berikut adalah tampilan form login admin dapat di lihat pada gambar



Gambar 16 Log in admin

b. Form penambahan data ATM

Form ini berfungsi agar admin dapat melakukan penambahan data ATM dengan mengisi nama ATM, alamat ATM, Latitude, Longitude dan foto ATM dan dapat tersimpan di sistem apat dilihat pada gambar

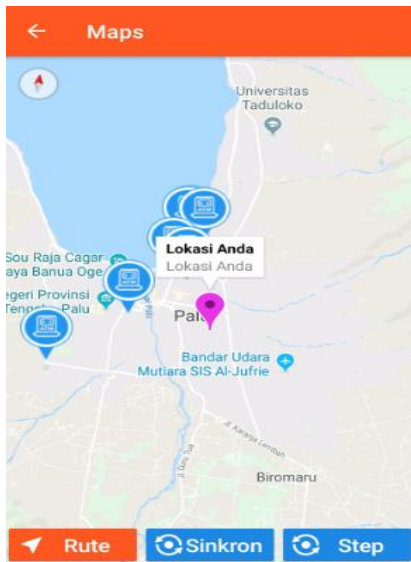


Gambar 17 tambah data ATM

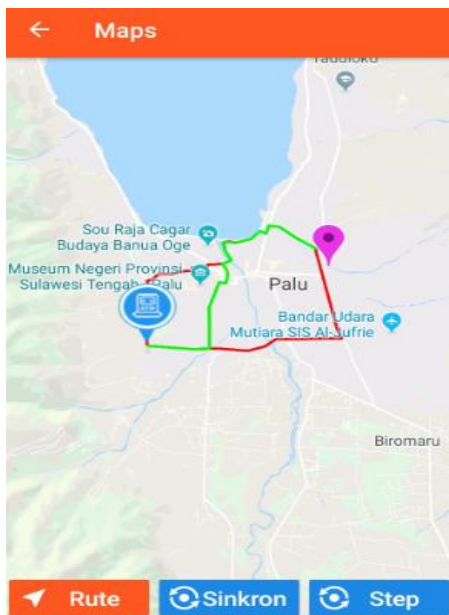
2. Implementasi Output

a. Form Maps

Form ini adalah menampilkan lokasi awal pengguna dan menampilkan seluruh lokasi ATM setelah itu pengguna dapat memilih satu ATM yang akan di tuju lalu sistem akan menampilkan jalur terpendek dari lokasi titik awal pengguna.



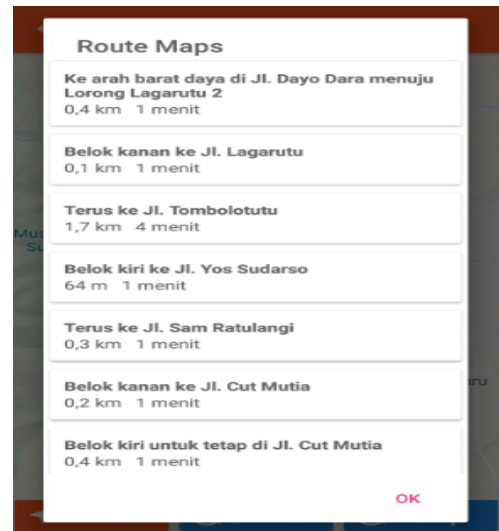
Gambar 18. Tampilan lokasi titik awal pengguna dan lokasi ATM



Gambar 19. Jalur terpendek lokasi ATM

b. Form Step

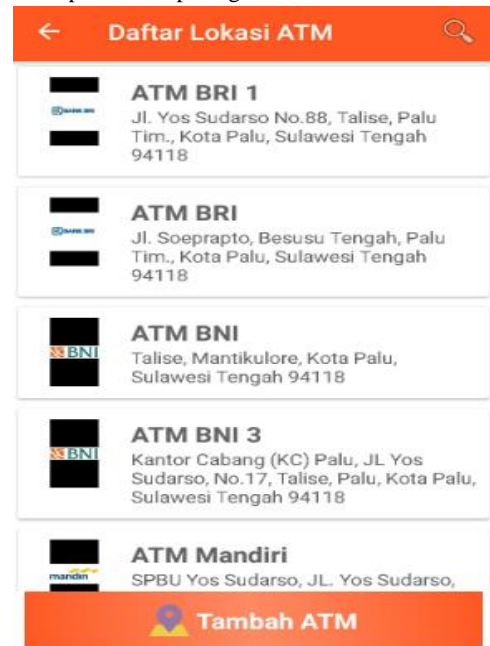
Form ini berfungsi mengarahkan pengguna berdasarkan jarak dan menampilkan waktu tempuh dapat dilihat pada gambar.



Gambar 20. Step panduan menuju lokasi ATM

c. Form Lokasi

Form Lokasi berfungsi untuk melihat semua lokasi ATM dan jalur terpendek serta panduan jalan menuju lokasi tersebut dapat di lihat pada gambar.



Gambar 21. Lokasi ATM dan alamat ATM

d. Form Panduan

Form Panduan berfungsi untuk melihat panduan aplikasi dapat di lihat pada gambar.



Gambar 22. Panduan penggunaan Aplikasi

e. *Form* Tentang

Pada *form* ini terdapat informasi yang berisi tentang informasi aplikasi serta profil mahasiswa pembuat aplikasi pencarian jalur terpendek lokasi ATM berbasis android menggunakan algoritma *Ant Colony Optimization* Adapun tampilan *form* sebagai berikut :






Gambar 23. *Form* Tentang

Pengujian Sistem


1. Pengujian system menggunakan *blackbox testing*.


Hasil pengujian fungsi sistem dengan menggunakan *blackbox testing* adalah dapat dilihat pada table 3.1.

TABEL I
TABEL PENGUJIAN FUNGSI SISTEM

| No | Fungsi Yang Diuji | Yang diharapkan | Pengamatan | Hasil Pengujian |
|----|--|--|---|-----------------|
| 1. | Tombol admin <i>Test case :</i>  | Pada saat menekan tombol admin akan menampilkan email dan password yang telah di registrasi di database. Hasil pengujian :  | Dapat menambahkan dan menghapus data ATM berfungsi dengan baik. Sesuai dengan yang diharapkan | Baik |
| 2. | Tombol maps <i>Test case :</i>  | Pada saat menekan tombol maps akan muncul titik lokasi awal pengguna dan semua lokasi ATM yang telah dimasukan ke sistem. Hasil pengujian : | Dapat menampilkan lokasi ATM dan titik lokasi pengguna. | Baik |

TABEL II
TABEL PENGUJIAN FUNGSI SISTEM (LANJUTAN 2)

| No | Fungsi Yang Diuji | Yang diharapkan | Pengamatan | Hasil Pengujian |
|----|---|---|------------------------------------|-----------------|
| 3. | Tombol marker lokasi pengguna <i>Test case :</i>  | Pada saat menekan tombol marker lokasi pengguna akan menampilkan lokasi pengguna dimana dia berada. | Dapat menampilkan lokasi pengguna. | Baik |

| | | | | | | | | | | | | |
|----|--|--|-------------------------------------|------|--|-------|--------|--------|--------|-------|---|-------|
| 4. | Tombol marker nama ATM Test cast :  | Pada saat menekan tombol marker ATM akan menampilkan seluruh nama ATM. Hasil pengujian : | Dapat menampilkan Semua lokasi ATM. | Baik | 2 | - | 79.01 | - | 127.65 | - | - | 70.83 |
| | | | | | 3 | - | - | 127.65 | - | 75.77 | - | - |
| | | | | | 4 | - | 126.91 | - | 75.77 | - | - | - |
| | | | | | 5 | 64.35 | - | - | - | - | - | - |
| | | | | | 6 | - | - | 70.83 | - | - | - | - |
| | | | | | Pada <i>graph</i> 0123456 akan dilakukan pencarian jalur terpendek dari titik 0 menuju titik 6 menggunakan algoritma ACO dengan tahapan penyelesaian sebagai berikut | | | | | | | |

TABEL III
TABEL PENGUJIAN FUNGSI SISTEM (LANJUTAN)

| No | Fungsi Yang Diuji | Yang diharapkan | Pengamatan | Hasil Pengujian |
|----|--|---|---|-----------------|
| 9. | Tombol open folder . Test case :  | Pada saat menekan tombol open folder akan menampilkan dialog penyimpanan (galeri) . | Dapat menampilkan galeri yang ada di handphone. | Baik |

1. Inisialisasi parameter

$q_0 = 0.5$

$\beta = 2$

$\rho = 0.1$

$\gamma = 0.1$

$\alpha = 0.1$

jumlah semut = 10

jumlah iterasi (siklus semut) = 2

2. Pembuatan tabel *pheromone* awal

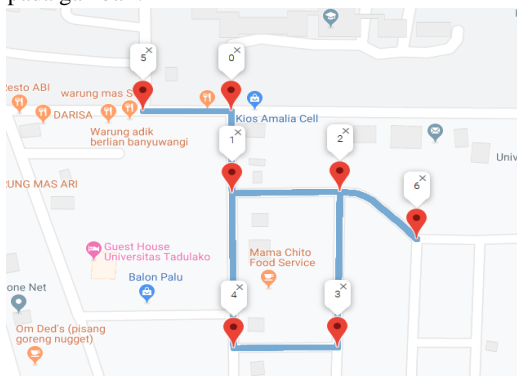
Tabel *pheromone* awal dapat dilihat pada tabel 3.5.

TABEL V
TABEL PHEROMONE AWAL

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|--------|--------|--------|--------|--------|--------|--------|
| 0 | - | 0.0001 | - | - | - | 0.0001 | - |
| 1 | 0.0001 | - | 0.0001 | - | 0.0001 | - | - |
| 2 | - | 0.0001 | - | 0.0001 | - | - | 0.0001 |
| 3 | - | - | 0.0001 | - | 0.0001 | - | - |
| 4 | - | 0.0001 | - | 0.0001 | - | - | - |
| 5 | 0.0001 | - | - | - | - | - | - |
| 6 | - | - | 0.0001 | - | - | - | - |

2. Pengujian Algoritma *Ant Colony Optimization* (ACO)

Pada pengujian ini akan diinisialisasi parameter-parameter yang menjadi bagian penting untuk algoritma ACO dalam menemukan jalur terpendek menuju lokasi tujuan. Pengujian Implementasi algoritma *Ant Colony Optimization* dapat dijelaskan sebagai berikut dengan contoh kasus *graph* 0123456 pada peta yang dapat dilihat pada gambar .



Gambar 24. *Graph* 0123456 pada peta

Dari *graph* 0123456 terbentuk tabel bobot jarak yang dapat dilihat pada tabel .

TABEL IV
TABEL BOBOT JARAK

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|-------|-------|-------|---|--------|-------|---|
| 0 | - | 67,72 | - | - | - | 64,35 | - |
| 1 | 67,72 | - | 79.01 | - | 126.91 | - | - |

3. Pembuatan tabel *invers* jarak

Tabel *invers* jarak dapat dilihat pada tabel 3.6.

TABEL VI
TABEL INVERS JARAK

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|-------|-------|-------|-------|-------|-------|-------|
| 0 | - | 0.015 | - | - | - | 0.016 | - |
| 1 | 0.015 | - | 0.013 | - | 0.008 | - | - |
| 2 | - | 0.013 | - | 0.008 | - | - | 0.014 |
| 3 | - | - | 0.008 | - | 0.013 | - | - |
| 4 | - | 0.013 | - | 0.008 | - | - | - |
| 5 | 0.016 | - | - | - | - | - | - |
| 6 | - | - | 0.014 | - | - | - | - |

4. Proses algoritma meliputi transisi status, pembaruan *pheromone* lokal dan pembaruan *pheromone* global.

Hasil proses perhitungan algoritma melalui tahapan-tahapan yang dimulai dari proses transisi status dimana semut akan memilih titik kunjungannya, kemudian proses pembaruan *pheromone* lokal yang dilakukan untuk mengurangi ketertarikan ruas simpul yang sudah dilewati oleh semut. Selanjutnya proses pembaruan *pheromone*

global yang dilakukan setelah satu siklus berakhir untuk menambahkan *pheromone* pada rute semut terbaik. Berikut adalah hasil dari setiap iterasi :

a. Hasil iterasi 1

- Semut 1 : hasil rute yaitu 0-1-4-3-2-6 dengan jarak 468,88
- Semut 2 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 3 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 4 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 5 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 6 : hasil rute yaitu 0-1-2-6 dengan jarak 217.56
- Semut 7 : hasil rute yaitu 0-1-2-3-4 dengan jarak 350.15
- Semut 8 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 9 : hasil rute yaitu 0-1-4-3-2-6 dengan jarak 468,88
- Semut 10 : hasil rute yaitu 0-1-2-6 dengan jarak 217.56.

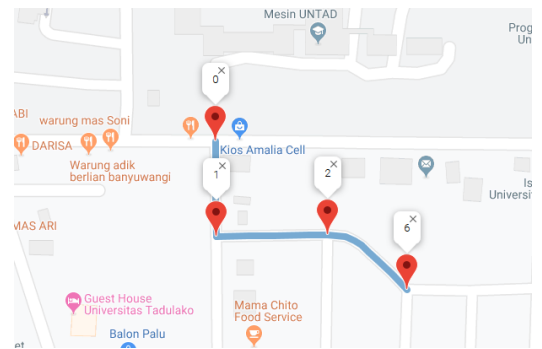
Dari hasil perolehan rute, terdapat 4 semut yang berhasil mencapai tujuan dari titik 0 ke titik 6. Semut tersebut adalah semut 1,6,9 dan 10. Selanjutnya bobot jarak keempat semut tersebut dibandingkan dan diambil nilai paling minimum untuk dijadikan sebagai semut terbaik dari satu iterasi. Hasil terbaik dari iterasi 1 adalah semut ke 6 dengan rute melewati titik 0-1-2-6 dan bobot jarak 217.56.

b. Hasil iterasi 2

- Semut 1 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 2 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 3 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 4 : hasil rute yaitu 0-1-2-6 dengan jarak 217.56
- Semut 5 : hasil rute yaitu 0-1-2-6 dengan jarak 217.56
- Semut 6 : hasil rute yaitu 0-1-4-3-2-6 dengan jarak 468,88
- Semut 7 : hasil rute yaitu 0-1-2-3-4 dengan jarak 350.15
- Semut 8 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 9 : hasil rute yaitu 0-5 dengan jarak 64.35
- Semut 10 : hasil rute yaitu 0-1-4-3-2-6 dengan jarak 468,88

Dari hasil perolehan rute, terdapat 4 semut yang berhasil mencapai tujuan dari titik 0 ke titik 6. Semut tersebut adalah semut 4,5,6 dan 10. Selanjutnya bobot jarak keempat semut tersebut dibandingkan dan diambil nilai paling minimum untuk dijadikan sebagai semut terbaik dari satu iterasi. Hasil terbaik dari iterasi 1 adalah semut ke 4 dengan rute melewati titik 0-1-2-6 dan bobot jarak 217.56.

Setelah proses kedua iterasi telah selesai maka dilanjutkan dengan proses untuk membandingkan nilai rute terbaik yang dihasilkan oleh masing-masing iterasi dengan mengambil nilai bobot jarak paling minimum. Dari hasil perbandingan, iterasi 1 dan iterasi 2 mendapatkan nilai jarak yang sama, oleh sebab itu rute terbaik yang akan dipilih bisa saja iterasi 1 maupun 2. Pada contoh kasus ini, rute terbaik yang akan dipilih adalah rute dari hasil iterasi 1 yaitu rute 0-1-2-6 dengan bobot jarak 217.56.



Gambar 25. Hasil Rute Terpendek

Pembahasan

Dari pengujian sistem, terdapat dua pengujian yang dapat dijadikan acuan hasil penelitian ini. Pengujian pertama yaitu pengujian fungsional sistem menggunakan *blackbox testing* yang membuktikan bahwa fungsional sistem aplikasi dapat berjalan dengan baik. Berdasarkan pengujian, aplikasi dapat menampilkan menu utama dan tombol-tombol aplikasi seperti *maps*, lokasi, panduan, tentang, admin, simpan data, serta search pencarian lokasi ATM dapat berfungsi sesuai dengan yang diharapkan. Pada perhitungan ACO di implementasikan ke dalam sistem yang telah di buat menggunakan aplikasi android studio versi 3.3 dengan memanfaatkan prosedur algoritma ACO, penelitian ini menggunakan data sebanyak 60 lokasi ATM di kota Palu

Pengujian kedua yaitu pengujian implementasi algoritma ACO yang dilakukan pada contoh kasus satu titik lokasi pengguna dan satu titik lokasi ATM. Hasil dari pengujian ini yaitu rute terpendek dari lokasi pengguna menuju ATM dapat dengan mudah didapatkan dan ditampilkan pada aplikasi.

Algoritma ACO merupakan algoritma yang memiliki proses yang cukup panjang untuk menyelesaikan masalah rute terpendek, namun dengan pengimplementasian algoritma pada sistem menggunakan bahasa pemrograman serta *device* yang mendukung maka proses penyelesaian algoritma ini dapat dilakukan dengan waktu yang singkat.

Pada prosesnya, algoritma ACO memiliki karakteristik yang harus dilakukan untuk mendapatkan hasil rute terpendek. Karakteristik-karakteristik tersebut yang pertama adalah proses transisi status. Proses transisi status adalah proses dimana semut akan memilih titik kunjungan selanjutnya menggunakan cara *eksploitasi* atau *eksplorasi* jalur. Karakteristik yang kedua adalah proses pembaruan *pheromone* lokal dimana proses ini akan terjadi ketika semut telah memilih titik kunjungannya. Proses ini akan mengatur penguapan *pheromone* untuk mengurangi ketertarikan ruas simpul yang telah dilewati oleh semut. Karakteristik yang ketiga adalah proses pembaruan *pheromone* global. Proses pembaruan *pheromone* global merupakan proses yang terjadi pada semut dengan hasil rute terbaik pada satu siklus (iterasi). Semut ini akan mengalami penambahan *pheromone* pada ruas-ruas yang telah dilewati. Hal ini dapat membantu semut-semut lainnya untuk membangun tur yang lebih baik pada siklus semut (iterasi) selanjutnya. Karakteristik algoritma ini akan berulang secara terus menerus hingga tetapan siklus semut (iterasi) berakhir.

Pada penelitian ini, sistem aplikasi memiliki tahapan-tahapan proses pengolahan informasi yang dilakukan oleh

sistem untuk memberikan hasil akhir yang dapat dikenali oleh pengguna. Tahapan pertama yaitu tahapan yang dilakukan oleh administrator aplikasi yakni penginputan lokasi-lokasi ATM. Penginputan lokasi ATM pada aplikasi meliputi informasi-informasi penting ATM yaitu nama ATM, *latitude*, *longitude* serta foto ATM. Informasi *latitude* dan *longitude* ATM akan digunakan untuk proses pencarian rute terpendek dari lokasi *user*.

Tahapan selanjutnya yaitu tahapan yang dilakukan oleh pengguna yang akan melakukan pencarian rute terpendek menuju lokasi-lokasi ATM. Pengguna akan melakukan input lokasi tujuan kemudian program akan segera otomatis mendapatkan posisi pengguna melalui GPS. Setelah mendapatkan koordinat posisi pengguna dan koordinat tujuan, selanjutnya adalah mencari koordinat jalan terdekat. Setiap jalan mempunyai titik-titik koordinat dan pada tahap ini program akan mencari koordinat jalan terdekat dari posisi pengguna berada dan koordinat jalan terdekat dari posisi ATM. Koordinat jalan terdekat dari pengguna akan menjadi titik awal mulainya pencarian rute, sedangkan koordinat jalan terdekat dari lokasi ATM akan menjadi titik tujuan pencarian rute.

Kemudian tahapan selanjutnya adalah proses pencarian rute terpendek. Sistem akan melakukan pencarian rute terpendek menggunakan algoritma ACO hingga selesai. Hasil dari pencarian rute akan ditampilkan pada *form* aplikasi dengan menampilkan rute meliputi informasi jarak dan waktu serta nama jalan yang akan dilewati oleh pengguna. Hasil pencarian jalur terpendek pada aplikasi ini dapat memandu pengguna menuju lokasi ATM terdekat dengan cara yang mudah serta waktu yang relatif lebih singkat.

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan pengujian dan analisis sistem implementasi algoritma *Ant Colony Optimization (ACO)* pencarian jalur terpendek lokasi *Automatic Teller Machine (ATM)* di Kota Palu Berbasis Android maka dapat di simpulkan sebagai berikut :

1. Dengan menggunakan Aplikasi Android Studio versi 3.3 penulis berhasil mengimplementasikan algoritma *Ant Colony Optimization (ACO)* untuk pencarian jalur terpendek lokasi ATM di kota palu berbasis *Android*.
2. Sistem penentuan rute menggunakan algoritma *Ant Colony Optimization (ACO)* dapat menghasilkan rute yang optimal untuk pencarian jalur terpendek lokasi ATM
3. Proses pembuatan rute yang optimal pada algoritma ACS dilakukan dengan melalui penentuan nilai *pheromone* awal dan nilai parameter q_0 , β , q , ρ , α , jumlah semut serta jumlah siklus semut atau iterasi kemudian masuk pada proses algoritma ACS yang meliputi transisi status, update *pheromone* lokal dan update *pheromone* global.

Proses perhitungan algoritma ACO di implementasikan ke dalam sistem yang telah di buat menggunakan aplikasi Android studio versi 3.3 dengan memanfaatkan prosedur pseudocode Algoritma *Ant Colony Optimization (ACO)*.

Pada perhitungan ACO di implementasikan ke dalam sistem yang telah di buat menggunakan aplikasi android studio versi

3.3 dengan memanfaatkan prosedur algoritma ACO, penelitian ini menggunakan data sebanyak 60 lokasi ATM di kota Palu

Saran

Diharapkan pada penelitian sistem pencarian jalur terpendek berbasis *android* dengan menggunakan algoritma *Ant colony optimization (ACO)* selanjutnya yaitu dengan lebih memperluas *platform* yang tersedia sehingga bukan hanya untuk *android* saja tetapi untuk *iOS*.

DAFTAR PUSTAKA

- [1] Adi, S., dan Kristin, D. M. 2014. Strukturisasi Entity Relationship Diagram dan Data Flow Diagram Berbasis Business Event Driven , *ComTech Vol. 5 No*, Binus University, Jakarta Barat.
- [2] Berry. J. 2016. "Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada Sistem Informasi Geografis Pemetaan Stasiun Pengisian Bahan Bakar Umum". Universitas Padang. Padang.
- [3] Edi.S. 2014. "Jenis-jenis Graf dan Graf Bipartisi". STIMIK GI MDP. Palembang.
- [4] Indra. B.M. 2016 "Ant Colony Optimazation Dalam Penyelesaian Traveling Salesmen Problem Menggunakan Matlab". Universitas Negeri Semarang. Semarang.
- [5] Karjono. Et al. 2016 "Algoritma Ant Colony Optimazation (ACO) untuk menyelesaikan traveling saleman Problem (TSP)". Universitas Diponegoro. Semarang.
- [6] Murya. Y. "Pemograman Android Black Box". Jasakom. Jakarta.
- [7] Hartantyo. T. P. "Rancang Bangun Pencarian Lokasi Rumah Sakit dan Puskesmas Di Wilayah Tegal Berbasis Android". Fakultas Ilmu Komputer Univeritas Dian Nuswantoro. Semarang.
- [8] Hengki, Suprawiro. S. "Analisis dan Perancangan Sistem Informasi Inventory Sparepart Kapal Berbasis Web". STIMIK Atma Luhur. Bangka Belitung.
- [9] Poernomo. B, Frydow. N. O. "Sistem Informasi Geografis Pencarian Rute Terpendek Kampus Di Kota Malang Dengan Menggunakan Algoritma Ant Colony". STIMIK ASIA. Malang.
- [10] Rossa. A.S, Shalahuddin. M. 2014 "Model Pengembangan Perangkat Lunak SLDC Waterfall". Bandung.
- [11] Sugiyono. 2014. "Tipe Penelitian Data Sekunder". UNPAS. Pasudan.
- [12] Suryamen, H., Aminuddin, I., dan Akbar, F. 2017. Perancangan Sistem Informasi Geografis Lapangan Futsal Kota Padang Berbasis Web, *TEKNOSI, Vol. 02, No. 01*, Universitas Andalas Limau Manis Padang, Padang.
- [13] Verdianto. E. 2014. "Perancangan Sistem Penentuan Rute Terpendek Jalur Evakuasi Tsunami Dengan Algoritma Ant Colony". Universita Sumatera Utara.