

TUGAS AKHIR

**PERENCANAAN JALUR JARINGAN KOMUNIKASI FIBER OPTIK  
MENGUNAKAN ALGORITMA ANT COLONY**

*Di Ajukan Kepada Politeknik Negeri Manado Untuk Memenuhi Salah Satu Persyaratan  
dalam Menyelesaikan Program Studi Diploma IV*

*Jurusan Teknik Elektro*

Oleh :

**LEYDI GRACE LONTAAN**

**NIM. 11 024 080**



Dosen Pembimbing

**FANNY J DORINGIN, ST. MT**  
**NIP. 19670430 199203 1 003**

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI  
POLITEKNIK NEGERI MANADO  
JURUSAN TEKNIK ELEKTRO  
2015**

**LEMBAR PENGESAHAN**

**PERANCANGAN JALUR JARINGAN KOMUNIKASI FIBER OPTIK  
MENGUNAKAN ALGORITMA ANT COLONY**

Oleh

LEYDI GRACE LONTAAN

NIM : 11 024 080

*Tugas Akhir ini telah di terima dan di sahkan senbagai persyaratan untuk  
menyelesaikan Pendidikan **Diploma IV** Teknik Elektro*

*Program Studi **Teknik Informatika***

*Politeknik Negeri Manado*

**Manado, Agustus 2015**

**Ketua Panitia Tugas Akhir,**

**Dosen Pembimbing,**

**Fanny J. Doringin, ST, MT**

**Fanny J. Doringin, ST, MT**

**NIP. 19670430 199203 1 003**

**NIP. 19670430 199203 1 003**

**Mengetahui,**

**Ketua Jurusan Teknik Elektro**

**Ir. Luther Mappadang, MT.**

**NIP. 19610601 199003 1 02**

## ABSTRAK

*Leydi Lontaan, 2015 Perencanaan Jalur Jaringan Komunikasi Fiber Optik Menggunakan Algoritma Ant Colony. Tugas Akhir Jurusan Teknik Elektro. Program Study Teknik Informatika.*

Dimulai pada tahun 2011, PT. Telkom Indonesia mulai menjalankan program untuk penarikan kabel fiber optic. Pada kenyataan dilapangan, penarikan kabel fiber optic dilakukan di pinggiran jalan dengan mengikuti alur jalan. Hal ini jelas lebih memakan waktu dan biaya.

Tujuan dari Tugas Akhir ini adalah untuk aplikasi pencarian jalur terpendek pemasangan kabel fiber optic PT. Telkom Indonesia yang dibangun menggunakan pemograman Visual Basic 2012. Dari hasil pencarian jalur terpendek mampu meminimalisir biaya pemasangan serta waktu pemasangan kabel.

Dalam proses penentuan jalur terpendek dilakukan dengan menggunakan algoritma ant colony. Algoritma ant colony di ambil dari perilaku koloni semut dalam pencarian jalur terpendek dari sarang ke sumber makanan. Terinspirasi dari hal tersebut, dalam Tugas Akhir ini akan diimplementasikan sebuah aplikasi desktop dengan metode ant colony untuk pencarian jalur terpendek pemasangan kabel.

Hasil yang diperoleh dari penelitian ini adalah jalur terpendek yang diperoleh dari pencarian jalur. Jalur terpendek tersebut ditunjukkan dalam bentuk *graph* pada gambar *maps* jl. Politeknik Buha. Juga sebagai bentuk pengembangan dari pemograman Visual Basic 2012.

**Kata Kunci:** *Algoritma Ant Colony, Pemasangan Kabel Fiber Optic, Jalur Terpendek, Visual Studio 2012,*

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yesus yang telah memberikan hikmat serta tuntunan-Nya, sehingga penulis dapat menyelesaikan penulisan Tugas Akhir ini sebagai syarat untuk menyelesaikan pendidikan diploma 4 di Politeknik Negeri Manado.

Dari awal pembuatan Tugas Akhir ini ini sampai telah selesainya laporan pertanggungjawaban ini dibuat, tentunya penulis mendapatkan banyak motivasi dan dorongan positif serta sumbangsih dari berbagai pihak. Oleh karena itu, selayaknya peneliti mengungkapkan rasa terima kasihnya yang tulus, kepada :

1. Bapak Ir. Jemmy J. Rangan, MT. selaku direktur Politeknik Negeri Manado yang telah menyetujui dan membantu hingga terlaksananya kegiatan Tugas Akhirsemester VII program studi D-IV Teknik Informatika jurusan Teknik Elektro tahun akademik 2014/2015.
2. Bapak Ir. Jusuf L. Mappadang, MT. selaku ketua jurusan Teknik Elektro yang telah membantu terlaksananya kegiatan Tugas Akhir Mahasiswa semester VII program studi D-IV Teknik Informatika jurusan Teknik Elektro tahun akademik 2014/2015.
3. Bapak Ir. Nikita Sajangbati, MT. selaku Ketua Program Studi Teknik Informatika.
4. Bapak Fanny Doringin, ST, MT. selaku ketua panitia Tugas Akhir 2015 dan juga sebagai dosen pembimbing dalam penyelesaian Tugas Akhir ini.
5. Kepada Dosen-Dosen Teknik Elektro yang telah membantu memberikan pengarah dan solusi dalam penyusunan Tugas Akhir ini.

6. Mama dan Papa serta adik-adik tercinta yang tanpa hentinya memberikan motivasi dan masukan sehingga segalanya saya jalani dengan baik.
7. Seluruh teman-teman Teknik Informatika angkatan 2011 yang telah banyak memberikan dorongan dan motivasi selama dalam proses pembelajaran dan penyusunan Tugas Akhir ini bisa berjalan dengan baik.
8. Kepada teman-teman dari TORANG-TORANG yang membantu dalam penyusunan skripsi
9. Kepada PT. Telkom Indonesia yang telah membantu memberikan data untuk keperluan penyusunan Program.
10. Kepada saudara Marcelino Grantino Laloan yang senantiasa memberikan motivasi dan dukungan serta membantu dalam penyelesaian Tugas Akhir ini.
11. Untuk Pablo yang menjadi penghibur ketika dalam penyelesaian Tugas Akhir ini.

Tak ada gading yang tak retak demikian pula dengan Tugas Akhir ini belumlah sempurna, sehingga kritik, saran dan masukan yang konstruktif sangat di harapkan untuk penyempurnaan penulisan berikutnya.

Agustus 2015

Penulis,

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>ABSTRAK</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	vii
<b>DAFTAR TABLE</b> .....	x

### **BAB I PENDAHULUAN**

1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3

### **BAB II DAFTAR PUSTAKA**

2.1 Microsoft Visual Studio .....	4
2.2.1 Komponen-Komponen .NET Framework.....	6
2.2 Visual Basic .NET .....	7
2.3 Fiber Optik .....	11
2.2.1 Pemasangan Fiber Optik .....	11
2.2.1 Jenis-Jenis Instalasi .....	12
2.4 Algoritma Semut .....	12
2.5 Algoritma Dijkstra.....	20
2.6 Teori Graf .....	23
2.6.1 Definisi Graf.....	23

2.6.2	Macam-Macam Graf .....	24
2.7	Google Maps .....	27
2.7.1	Google Maps API.....	27
2.8	Roadmap Penelitian.....	29

### **BAB III OBJEK DAN METODOLOGI PENELITIAN**

3.1	Objek Penelitian .....	32
3.2	Analisa Sistem.....	32
3.3	Perancangan Sistem.....	34
3.4	Flowmap.....	35
3.5	Diagram Konteks.....	36
3.6	Data Flow Diagram .....	36
3.7	Perancangan Desain User Interface.....	37
3.7.1	Batasan Implementasi.....	37
3.7.2	Implementasi Antarmuka .....	37
3.7.3	Implementasi Prosedural .....	39

### **BAB IV IMPLEMENTASI DAN TESTING**

4.1	Spesifikasi Hardwere.....	42
4.2	Pengujian Program .....	42

### **BAB V KESIMPULAN DAN SARAN**

5.1	Kesimpulan.....	57
5.2	Saran.....	58

<b>DAFTAR PUSTAKA</b> .....	59
-----------------------------	----

### **LAMPIRAN**

## DAFTAR GAMBAR

Gambar 2.1 Tampilan Awal Visual Studio 2012.....	5
Gambar 2.2 Debugging Program Visual Studio 2012 .....	5
Gambar 2.3 Perjalanan Semut Dari Sarang Ke Sumber Makanan .....	18
Gambar 2.4 Graf Berarah dan berbobot.....	24
Gambar 2.5 Graf Tidak berarah dan berbobot.....	24
Gambar 2.6 Graf Berarah dan Tidak Berbobot.....	25
Gambar 2.7 Graf tidak berarah dan tidak berbobot .....	25
Gambar 2.8 Sinerai Kedekatan .....	27
Gambar 3.1 Diagram Alir Aplikasi Pencarian Jalur.....	35
Gambar 3.2 Diagram Konteks .....	36
Gambar 3.3 Data Flow Diagram.....	37
Gambar 3.4 Menu Utama Aplikasi .....	38
Gambar 3.5 Form Google Map.....	38
Gambar 3.6 Deklarasi Variabel .....	39
Gambar 3.7 Fungsi iterasi.....	39
Gambar 3.8 Fungsi Semut .....	40
Gambar 3.9 Update Pheromone.....	40
Gambar 3.10 Class Road .....	41
Gambar 3.11 Fungsi DrawTour .....	42
Gambar 4.1 Form Utama .....	44
Gambar 4.2 Buka Folder.....	45
Gambar 4.3 Proses Menggambar Titik .....	45



Gambar 4.4 Rute dengan 4 titik ditemukan .....	47
Gambar 4.5 Rute Dengan 6 titik .....	49
Gambar 4.6 Menggambar rute dengan banyak titik .....	49
Gambar 4.7 Proses pencarian rute oleh computer .....	50
Gambar 4.8 Rute Dengan Banyak Titik Ditemukan.....	51
Gambar 4.9 Gambar ketika checkbox tempilkan data diklik.....	52
Gambar 4.10 Tampilkan hasil iterasi .....	53
Gambar 4.11 Form Google Map.....	54
Gambar 4.12 Pencarian jalur 1 .....	55
Gambar 4.13 Pencarian jalur 2 .....	56

## DAFTAR TABEL

Tabel 2.1 Matriks Kedekatan.....	26
Tabel 2.2 Penelitian Terdahulu.....	29
Table 4.1 Keterangan sumbu x dan y pada pencarian dengan 4 titik .....	6
Table 4.2 Keterangan hasil perhitungan 4 titik.....	48
Table 4.3 Keterangan sumbu x dan y pada pencarian dengan 4 titik .....	48
Table 4.2 Keterangan hasil perhitungan 6 titik.....	49

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Fiber optic mempunyai kemampuan membawa lebih banyak informasi dengan lebih akurat dibandingkan dengan kabel tembaga dan kabel coaxial. Fiber optic mendukung data rate yang lebih besar, jarak yang lebih jauh sehingga menjadikannya ideal untuk transmisi serial data digital.

Dimulai dari tahun 2011, PT. Telkom Indonesia mulai menjalankan program untuk memigrasi kabel tembaga ke fiber optic lewat jalur bawah tanah. Hasil dari penentuan jalur terpendek penarikan kabel akan berpengaruh pada biaya pemasangan sehingga menjadi pertimbangan dalam pengambilan keputusan untuk menunjukkan jalur terbaik yang akan di tempuh dalam pemasangan kabel fiber optic. Seperti yang terjadi kebanyakan di lapangan, penarikan kabel fiber optic biasanya hanya di lakukan di pinggiran ruas jalan. Oleh karena hal itu, penulis tertarik untuk membuat aplikasi pencari jalan terpendek atau jalur terbaik.

Dalam pencarian jalur terpendek, ada dua algoritma yang biasa digunakan untuk proses penentuan jalur, yaitu algoritma Dijkstra dan Algoritma Koloni Semut (Ant Colony Algorithm). Dalam penelitian ini, penulis memilih menggunakan algoritma ant colony untuk proses pencarian jalur terpendek.

Sesuai dengan namanya, algoritma semut ini diambil dari perilaku koloni semut dalam pencarian jalur terpendek antara sarang dan sumber makanan. Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat yang menjadi sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah

dilalui. Mengingat prinsip algoritma yang didasarkan pada perilaku koloni semut dalam menemukan jarak perjalanan paling pendek tersebut, algoritma semut sangat tepat untuk digunakan dalam penyelesaian masalah optimasi, salah satunya adalah untuk menentukan jalur terpendek penarikan kabel.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang diatas, penulis merumuskan permasalahan dalam pembuatan skripsi ini, yaitu:

1. Bagaimana mendapatkan rute terbaik untuk sampai ke titik pemasangan dengan menggunakan metode algoritma ant colony?
2. Bagaimana membuat aplikasi untuk mempermudah penentuan rute dengan algoritma ant colony?

## **1.3 Batasan Masalah**

Dari latar belakang dan rumusan masalah yang telah di jelaskan sebelumnya, pencarian jalur terpendek dalam penarikan kabel fiber optic dibatasi pada salah satu jenis algoritma yang digunakan dengan metode heuristic, yaitu algoritma semut. Batasan masalah yang diperoleh yaitu:

1. Inputan yang diperlukan untuk program ini berupa titik-titik tujuan
2. Program hanya mencari jalur untuk jaringan yang baru akan dipasang
3. Bobot antar titik yang ditentukan adalah jarak.
4. Keluaran yang dihasilkan dari program ini adalah informasi jalur terbaik untuk sampai ke titik tujuan.
5. Perangkat lunak dibangun menggunakan bahasa pemograman Visual Basic 2012.
6. Lingkup penelitian hanya di daerah politeknik

#### **1.4 Tujuan Penelitian**

Tujuan yang ingin dicapai dari penulisan tugas akhir ini adalah membuat suatu perangkat lunak yang bisa memberikan informasi jalur terpendek dengan algoritma ant colony yang mudah untuk digunakan oleh user.

#### **1.5 Manfaat Penelitian**

Aplikasi ini akan dapat dimanfaatkan untuk:

1. Memberikan informasi mengenai jalur terpendek yang bisa dilalui oleh kabel.
2. Mendapatkan aplikasi desktop yang user friendly.

## **BAB III**

### **OBJEK DAN METODOLOGI PENELITIAN**

#### **3.1 Objek Penelitian**

Objek penelitian dari tugas akhir ini, dilaksanakan di:

Lokasi : Jalur Kabel Di Fiber Optik Di Jl. Politeknik Desa Buha,  
Kec. Mapanget.

Waktu: April – Juni 2015

Yang menjadi objek dari penelitian ini adalah jalur penarikan kabel fiber optic PT. Telkom Indonesia yang dilakukan di daerah Jl. Politeknik, Buka .

#### **3.2 Analisa Sistem**

Analisa system dilakukan untuk mengidentifikasi dan mengevaluasi suatu permasalahan dan hambatan-hambatan yang terjadi pada sebuah system sehingga dapat diusulkan sebuah perbaikan.

Kini komunikasi mengalami perubahan. Sebagai kebutuhan dasar manusia, komunikasi tidak hanya berbentuk suara, tetapi bisa juga gambar dan data. Komunikasi tidak hanya manusia dengan manusia, tetapi manusia dengan mesin atau mesin dengan mesin. Internet menghubungkan semuanya. Yang memiliki konsep untuk meningkatkan kualitas hidup manusia seperti meningkatkan kenyamanan dan kemudahan, efisiensi, keamanan, kesehatan dan lain-lain.

Salah satu realisasi dari internet of thing adalah konsep konsep digital home dengan menggunakan fiber optic. Dan PT. Telkom Indonesia

menghadirkan konsep digital home melalui IndiHome Fiber, Internet Fiber.

Fiber optic memungkinkan kecepatan akses internet sampai dengan 100 Mbps, jauh lebih tinggi dari kabel tembaga yang hanya sampai 5 Mbps. Jaringan fiber optic menghasilkan koneksi yang lebih stabil dan tidak terganggu oleh gelombang elektromagnetik serta lebih aman dari serangan petir.

Dimulai dari tahun 2011 PT. Telkom Indonesia mulai menjalankan proyek migrasi dari kabel tembaga ke fiber optic. Oleh karena itu, sering kita lihat adanya penggalian untuk penarikan kabel fiber optic di pingiran ruas jalan.

Jika dilihat dari efisiensi harga, penarikan kabel di ruas jalan lebih memakan biaya karena jarak yang lebih jauh, dibandingkan jika melakukan pemasangan dengan mencari jarak terdekat.

Maka dari itu diperlukan sebuah system yang dapat melihat jalur terdekat untuk pemasangan kabel fiber optic. Sistem ini memungkinkan pengguna untuk melihat jalur yang lebih efisien dan terdekat sehingga dapat menekan anggaran pemasangan kabel.

### **3.2.1 Analisis Kebutuhan Perangkat Lunak**

Perangkat keras komputer tidak berarti tanpa perangkat lunak begitu juga sebaliknya. Jadi perangkat lunak dan perangkat keras computer saling mendukung satu sama lain. Perangkat keras akan berfungsi jika diberikan instruksi-instruksi kepadanya instruksi tersebut diberikan menggunakan perangkat lunak. Dalam penelitian ini, penyusunan menggunakan perangkat lunak Microsoft Visual Studio 2012.

### **3.2.2 Analisis Kebutuhan Perangkat Lunak**

Penggunaan system computer sebagai alat bantu dalam menyelesaikan tugas-tugas atau pekerjaan sudah bukan hal yang aneh, tapi merupakan suatu keharusan karena banyak kemudahan yang diperoleh.

Perangkat keras computer yang digunakan adalah perangkat keras yang dapat mendukung perangkat lunak yang memiliki kemampuan atau tampilan grafis yang cukup baik. Perangkat keras yang digunakan pada aplikasi ini adalah sebagai berikut:

- a. Laptop Lenovo G40
- b. Processor Intel Inside Core i3
- c. RAM 2 GB
- d. Mouse

### **3.3 Perancangan Sistem**

Perancangan system merupakan bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah melalui tahapan analisis, perancangan system merupakan lanjutan dari analisa system, dimana perancangan dilakukan sebelum pengkodean ke dalam bahasa pemrograman.

#### **3.3.1 Tujuan Perancangan Sistem**

Tujuan perancangan ini merupakan rancangan perangkat lunak yang akan dibuat berdasarkan hasil analisa yang telah dibuat sebelumnya terhadap system yang telah ada. Tujuan lain dari perancangan ini adalah untuk menghasilkan aplikasi atau system pertukaran dokumen elektronik berbasis desktop untuk penentuan jalur terbaik penarikan kabel fiber optic di PT. Telkom Indonesia.



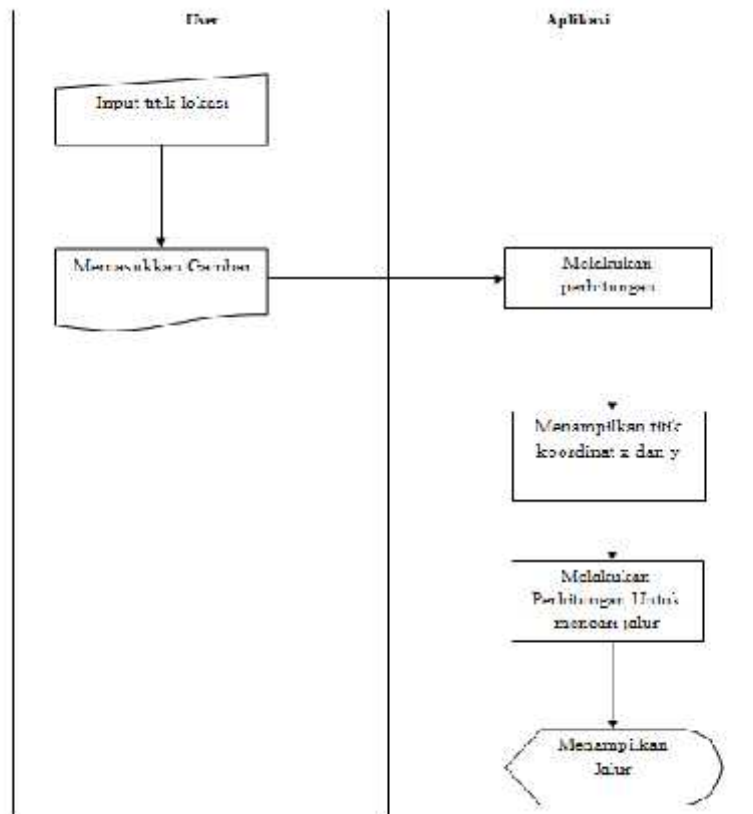
Perancangan aplikasi ini mencakup diagram konteks, data flow diagram (DFD) flowchart, perancangan antar muka dan kebutuhan system sehingga menghasilkan aplikasi yang baik bagi user.

### 3.3.2 Gambaran Umum Aplikasi

Aplikasi yang akan dibuat nantinya mampu melakukan proses pencarian jalur terdekat dari setiap titik tertentu di daerah Jl. Politeknik, Buha. Proses pencarian jalur tersebut dilakukan melalui aplikasi desktop.

### 3.4 Flow Map

Berikut adalah flow map untuk aplikasi penentu jalur:



Gambar 3.1 Diagram Alir Aplikasi Pencarian Jalur

Pada flowmap diatas menunjukan user mengambil gambar dimana gambar yang diambil adalah gambar peta daerah yang akan diproses untuk mencari jalur. Setelah gambar dipilih dan ditampilkan

dalam aplikasi, user menggambar titik yang akan dijadikan sebagai titik tempu untuk mencari jalur. Setelah itu, program akan mencari jalur terbaik untuk dilalui sesuai dengan titik yang telah ditentukan.

### 3.5 Diagram Konteks

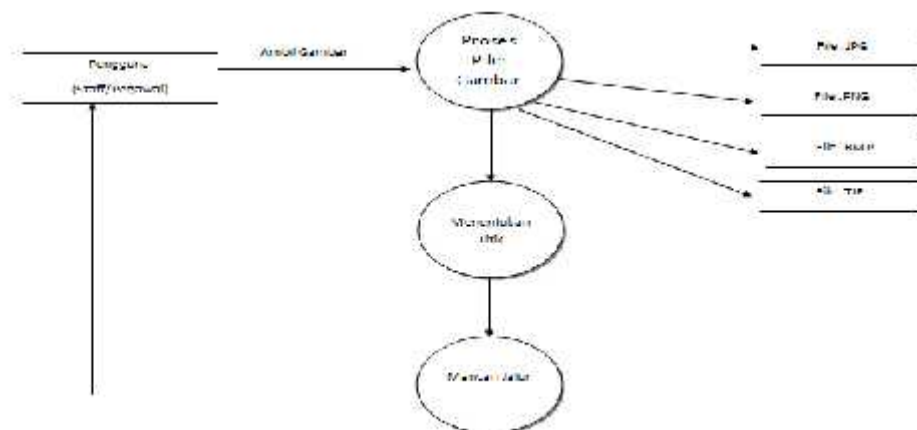
Diagram konteks berfungsi untuk mendefinisikan awal dan akhir dari data yang masuk serta keluaran pada suatu system atau aplikasi. Berikut ini adalah diagram konteks dari aplikasi pencarian jalur terpendek penarikan kabel fiber optic dengan algoritma ant colony.



Gambar 3.2 Diagram Konteks

### 3.6 Data Flow Diagram

DFD adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data system, yang penggunaannya sangat membantu untuk memahami system secara logika terstruktur dan jelas.



Gambar 3.3 Data Flow Diagram

### 3.7 Perancangan Desain User Interface (Antar Muka)

Pada perancangan desain user interface aplikasi penentu jalur terpendek dengan algoritma ant colony ini diharapkan mampu memeberikan user interface yang mudah digunakan oleh pengguna dan setiap fungsi dari interface yang ada mampu dipahami oleh pengguna.

#### 3.7.1 Batasan Implementasi

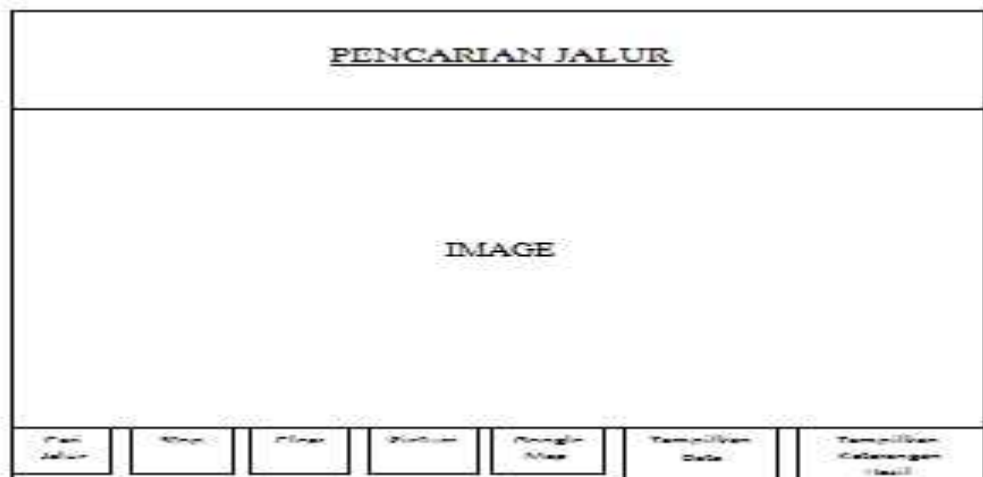
Aplikasi ini adalah untuk menentukan jalur terpendek, dalam hal ini implementasinya hanya pada pencarian jalur terpendek dari graph yang telah dimasukkan oleh pengguna

#### 3.7.2 Implementasi Antar Muka

Halaman ini merupakan halaman utama dari aplikasi penentu jalur terpendek dengan algoritma ant colony.

##### 3.7.2.1 Halaman Utama

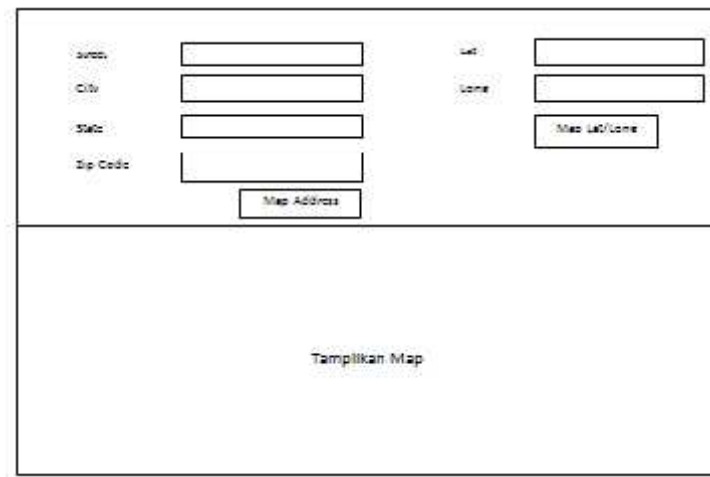
Halaman utama ini merupakan halaman utama dari aplikasi pencarian jalur terpendek dengan algoritma ant colony. Dalam halaman utama ini, terdapat beberapa menu antara lain pilih gambar, tampilkan lebel perhitungan, tampilkan form debug, dan tampilkan form google map.



Gambar 3.4 Menu Utama Aplikasi

### 3.7.2.1 Halaman Google Map

Halaman google map ini digunakan untuk menggambar jalur yang telah didapat dari hasil pencarian di form utama. Halaman ini juga berfungsi untuk melihat peta lokasi yang ingin dilakukan penarikan kabel.



Gambar 3.5 Form Google Map

### 3.7.3 Implementasi Prosedural

Implementasi procedural merupakan implementasi pada pemrograman system. Yang akan dijelaskan disini adalah pemrograman algoritma ant coloni, yang merupakan proses inti dalam perangkat lunak pencarian jalur terpendek dengan algoritma ant coloni.

#### 3.7.3.1 Deklarasi Variabel

Dalam aplikasi ini, variable utama yang di butuhkan untuk pencarian jalur sudah di deklarasikan terlebih dahulu.

```
Public Const  $\alpha$  As Double = -1.5  
Public Const  $\beta$  As Double = 1.5  
Public Const NumberOfAnts As Integer = 100  
Public Const InitialPheromoneValue As Double = 1  
Public Const PheromoneDecayFactor As Double = 0.1  
Public Const MaxIterations As Integer = 20
```

Gambar 3.6 Deklarasi Variabel

### 3.7.3.1 Method FindTour

Method ini berfungsi untuk mencari jalur yang terpendek. Dalam method ini, probabilitas dari masing-masing semut akan dihitung. Berikut implementasinya

#### a. Fungsi Iterasi

```
Do While iterations_without_change < MaxIterations
  iterations -= 1
  Dim ants As New List(Of Ant)(NumberOfAnts)
  Dim ant_success As New Dictionary(Of Ant, Boolean)(NumberOfAnts)

  For i = 1 To NumberOfAnts
    Dim rnd_index = m_Random.Next(Cities.Count)
    ants.Add(New Ant(Cities(rnd_index), ant_pheromone_capacity))
  Next

  last_value = 0
  num_success = 0
```

Gambar 3.7 Fungsi iterasi

#### b. Fungsi Semut

```
For Each ant In ants
  Dim success = ant.SearchTour() Also ant.VisitedCities.Count = #_Cities.Count
  ant_success(ant) = success

  If success Then
    Dim delta = ant.TourValue - best_tour_value

    If Math.Abs(delta) <= 0.01 Then
      iterations_without_change -= 1
    ElseIf delta < 0 Then
      best_tour_value = ant.TourValue
      best_tour = ant.VisitedCities
      iterations_without_change = 0
    ElseIf delta <= -best_tour_value * 0.01 Then
      iterations_without_change -= 1
    Else
      iterations_without_change = 0
    End If

    last_value += ant.TourValue
    num_success += 1
  Else
    iterations_without_change = 0
    number_of_failures += 1
  End If
Next
```

Gambar 3.8 Fungsi Semut

c. Fungsi update pheromone

```
Private Sub UpdatePheromoneLevel(ByVal road As Road)
    Const RemainingPheromoneFactor As Double = 1.0 - PheromoneDecayFactor
    road.PheromoneLevel = road.PheromoneLevel * RemainingPheromoneFactor
End Sub
```

Gambar 3.9 Update Pheromone

d. Class Road

```
<<DebuggerDisplay("Distance: {Distance}, Pheromone Level: {PheromoneLevel}")> _
Public Class Road

    Private Const  $\alpha$  As Double = World. $\alpha$ 
    Private Const  $\beta$  As Double = World. $\beta$ 
    Private m_Distance As Double
    Private m_PheromoneLevel As Double

    Public ReadOnly Property Distance() As Double
    Get
        Return m_Distance
    End Get
End Property

    Public Property PheromoneLevel() As Double
    Get
        Return m_PheromoneLevel
    End Get
    Set(ByVal value As Double)
        m_PheromoneLevel = value
    End Set
End Property

    Public ReadOnly Property WeighadValue() As Double
    Get
        Return Distance ^  $\alpha$  * PheromoneLevel ^  $\beta$ 
    End Get
End Property

    Public Sub New(ByVal distance As Double)
        m_Distance = distance
    End Sub
End Class
```

Gambar 3.10 Class Road

e. Fungsi DrawTour(Menggambar Titik)

```
Using p As New Pen(color, 2)
    Dim i = tour.GetEnumerator()
    i.MoveNext()
    Dim first = m_CityMap(i.Current)
    Dim c1 = first

    Do While i.MoveNext()
        Dim c2 = m_CityMap(i.Current)
        g.DrawLine(p, c1.Location, c2.Location)
        c1 = c2
    Loop

    g.DrawLine(p, c1.Location, first.Location)
End Using
```

Gambar 3.11 Fungsi DrawTour

## **BAB IV**

### **IMPLEMENTASI DAN TESTING**

#### **4.1 Spesifikasi Hardware dan Software**

Program ini diharapkan dapat membantu dalam pencarian jalur terdekat dan mampu memberikan informasi, dan rute yang akurat. Untuk itu program ini dibutuhkan perangkat lunak (Software) dan perangkat keras (hardware) agar berjalan sesuai dengan fungsinya. Adapun perangkat yang diperlukan adalah sebagai berikut:

1. Perangkat Keras
  - Laptop Lenovo G40
  - RAM 2 GB
  - Intel Inside Core i3
2. Perangkat Lunak
  - Visual Studio 2012 (VB .Net)

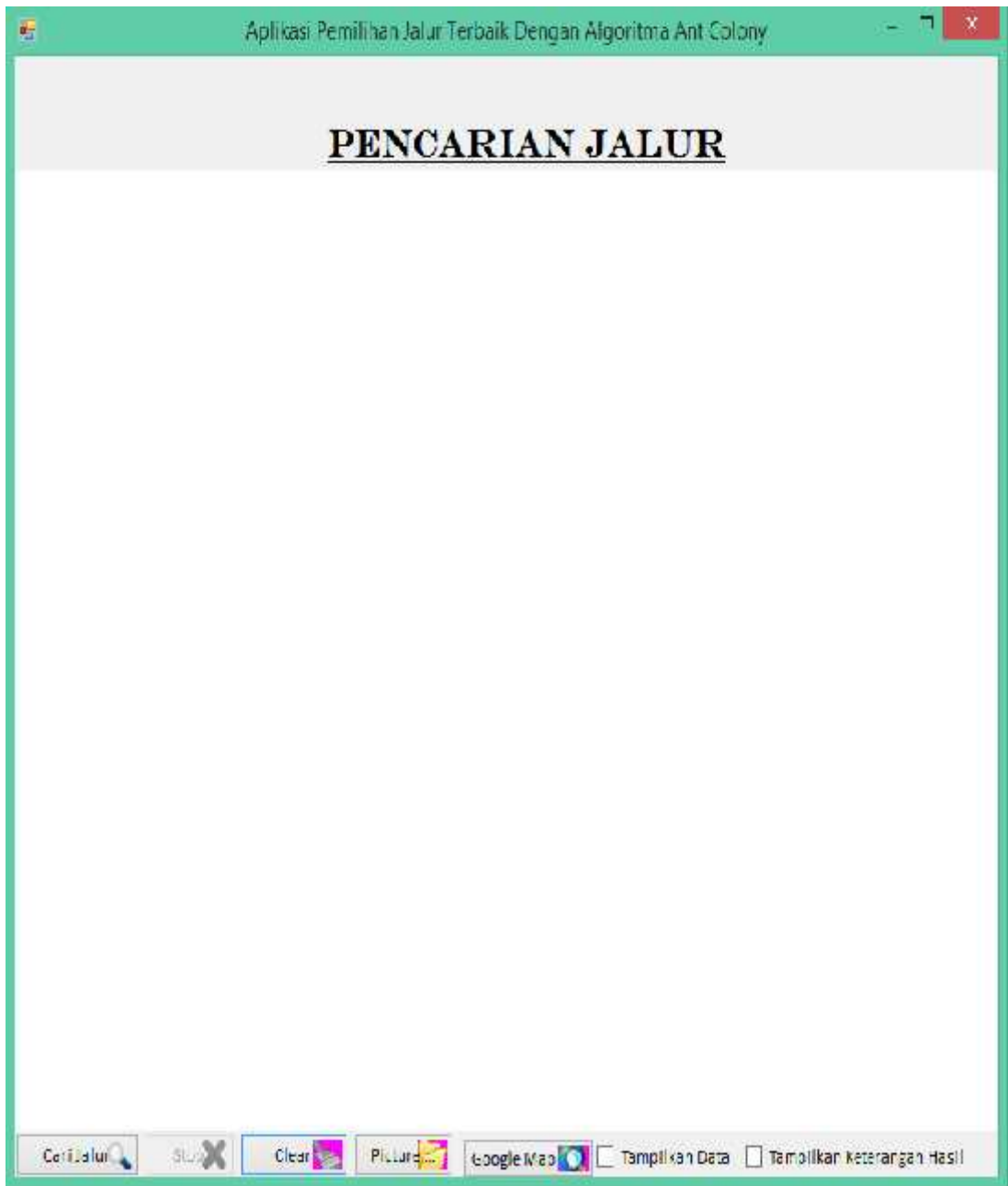
#### **4.2 Pengujian Program**

Pada tahap analisis kinerja perangkat lunak dijelaskan aplikasi pencarian jalur terpendek menggunakan algoritma semut. Pengujian dilakukan dengan kompleks dan diharapkan dapat diketahui kekurangan-kekurangan dari system untuk kemudian diperbaiki sehingga kesalahan dari system dapat diminimalisasi atau bahkan dihilangkan. Pengujian system ini dilakukan untuk mendapatkan hasil yang akurat.

Pengujian system ini dapat dilakukan dengan mengisi form inputan yang telah ditampilkan pada BAB III, yaitu dengan mengambil gambar dan menggambar titik (graph).



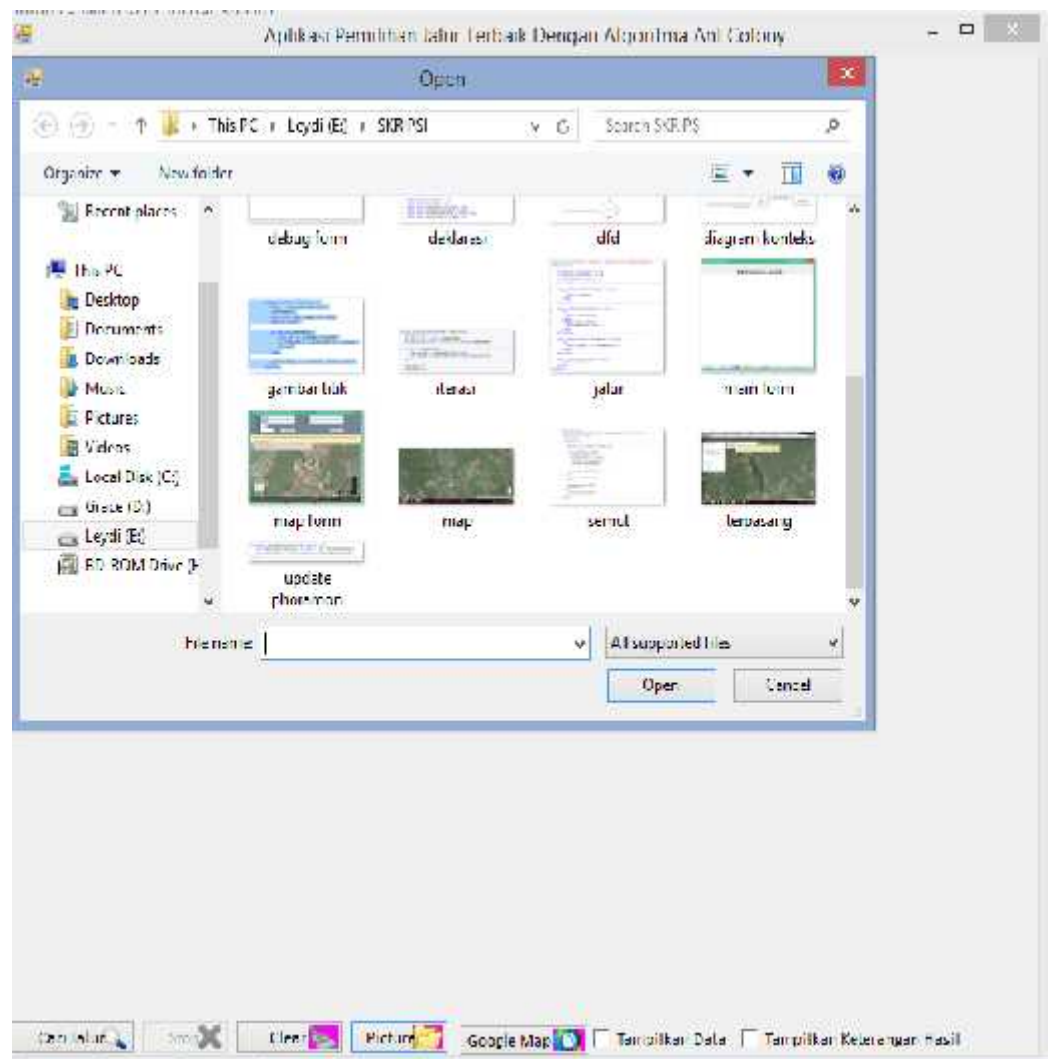
## 1. Form Utama



Gambar 4.1 Form Utama

Form utama adalah eksekusi pertama ketika program ini dijalankan. Pada form ini kita akan melakukan proses pencarian jalur. Berikut adalah proses pencarian jalur:

- a. Klik Tombol untuk membuka Gambar Map.



Gambar 4.2 Buka Folder

- b. Gambar yang telah dipilih, kemudian diinput ke form. Kemudian masukkan 4 titik dimana 4 titik tersebut adalah titik tujuan untuk pencarian jalur.



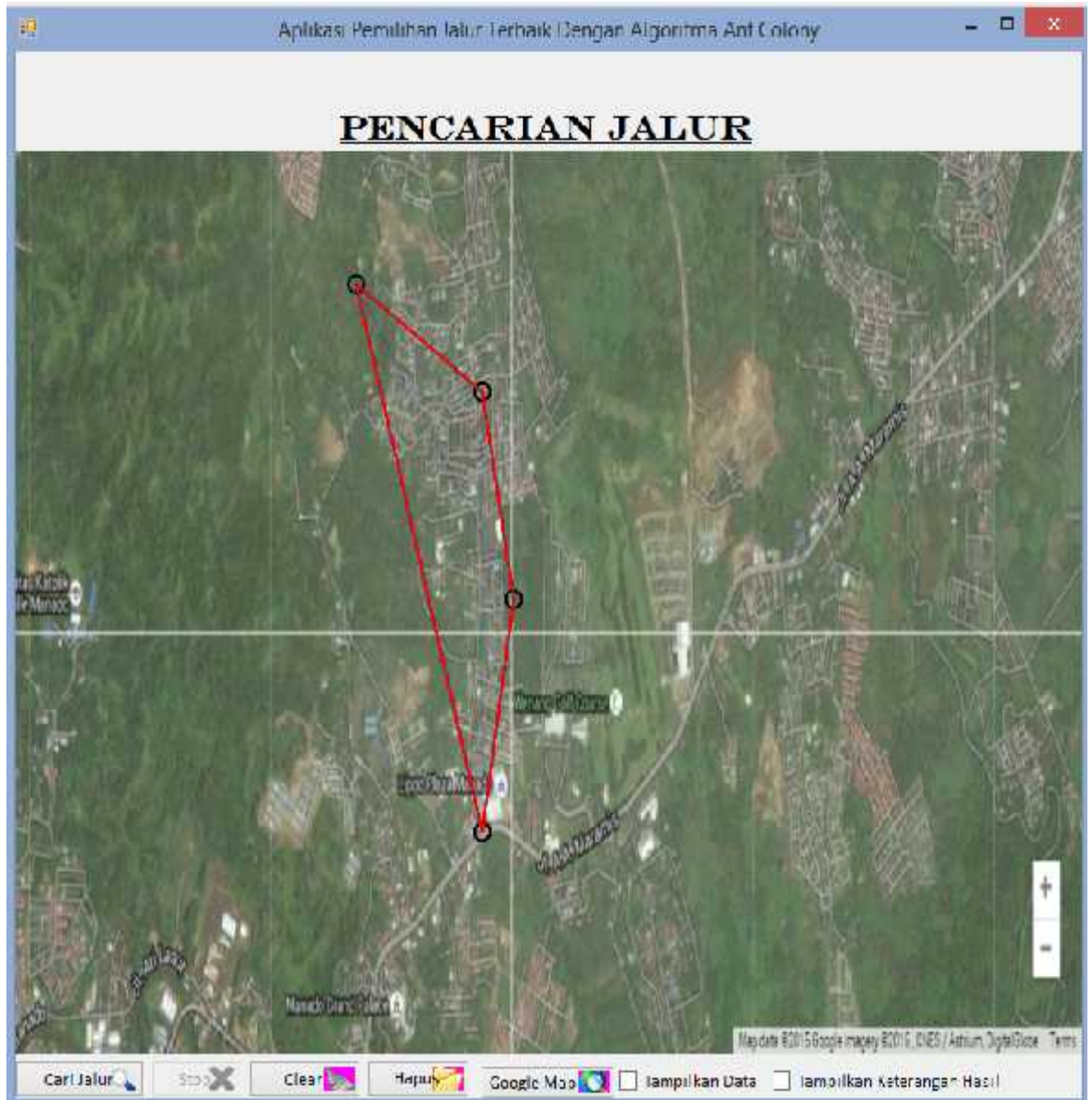
Gambar 4.3 Proses Menggambar Titik

Berikut table keterangan sumbu x dan y dari setiap titik pada pencarian jalur dengan menggunakan 4 titik:

Table 4.1 Keterangan sumbu x dan y pada pencarian dengan 4 titik

Titik	X	Y
A	353	447
B	372	267
C	347	157
D	256	82

Setelah menggambar titik, maka klik button Cari Jalur. Pada saat button di klik, program akan mencari jalur terpendek yang akan dilalui seperti gambar berikut:



Gambar 4.4 Rute dengan 4 titik ditemukan

Table berikut adalah table keterangan hasil perhitungan jumlah jarak dari setiap titik.

Table 4.2 Keterangan hasil perhitungan 4 titik

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>A</b>	-	181	290	378
<b>B</b>	181	-	113	218
<b>C</b>	290	113	-	118
<b>D</b>	378	218	118	-

Bagaimana jika pencarian jalur lebih dari 4 titik?? Berikut implementasi program dengan rute lebih dari 4 titik.

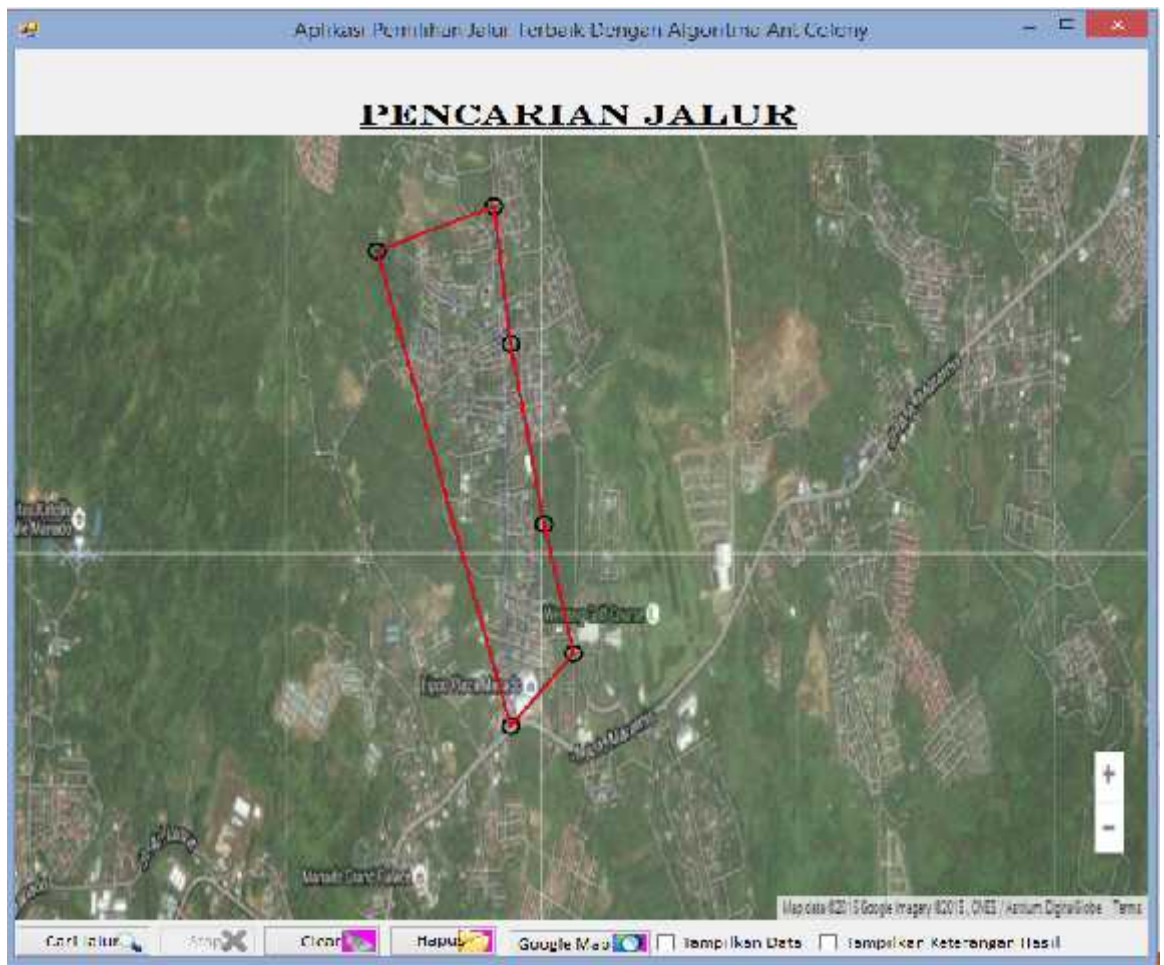
- Pencarian rute dengan 6 titik

Pengujian ini dimaksudkan untuk mencari tahu apakah akan terjadi error ketika mencari program lebih dari 4 titik.

Berikut adalah table keterangan sumbu x dan y pada pencarian jalur dengan 6 titik:

Table 4.3 Keterangan sumbu x dan y pada pencarian dengan 4 titik

Titik	<b>X</b>	<b>Y</b>
A	334	469
B	400	393
C	377	293
D	350	153
E	253	81
F	343	50



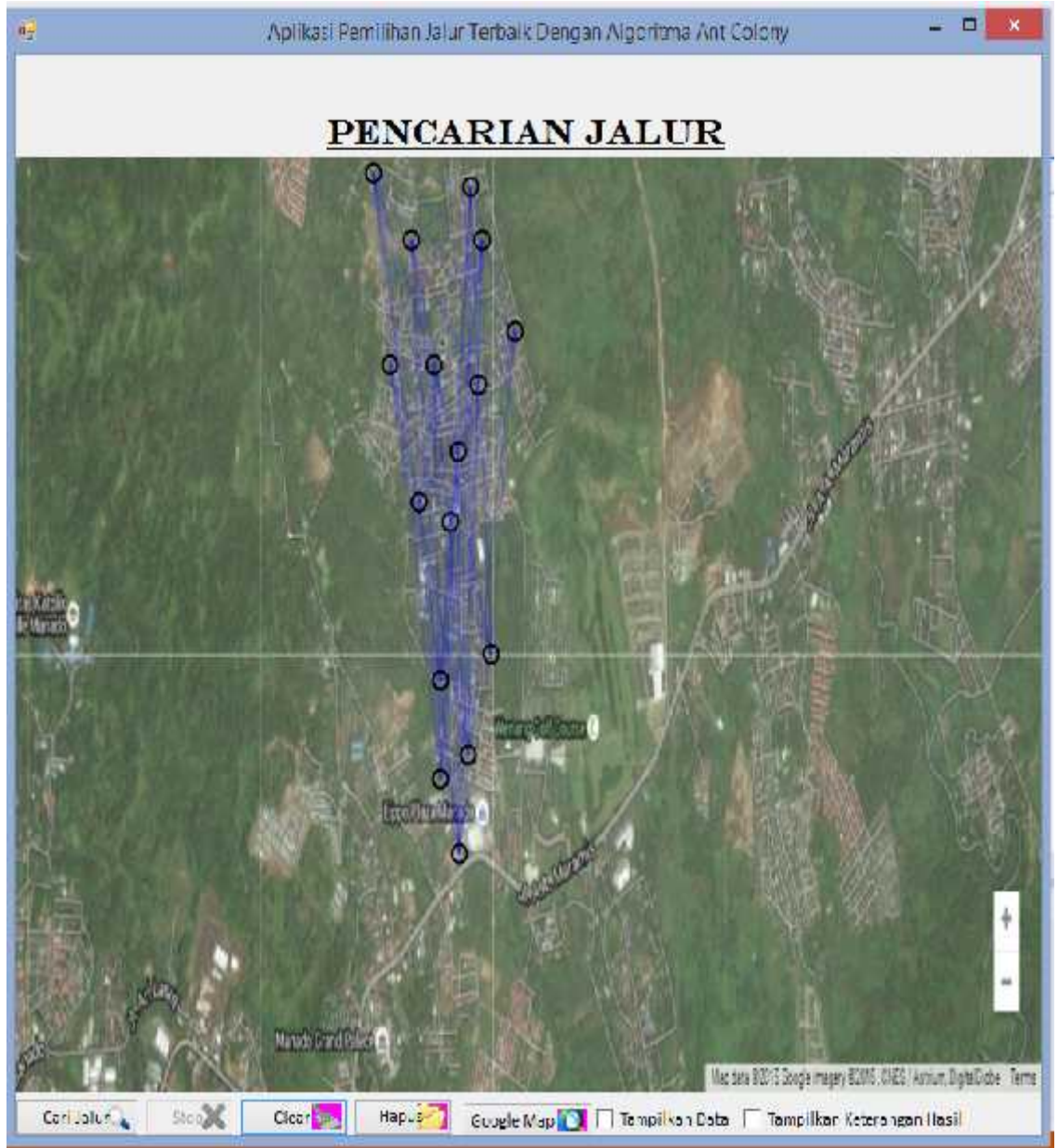
Gambar 4.5 Rute Dengan 6 titik

Table berikut adalah table keterangan hasil perhitungan jumlah jarak dari setiap titik:

Table 4.2 Keterangan hasil perhitungan 6 titik

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>A</b>	-	64	130	237	310	323
<b>B</b>	64	-	84	196	286	286
<b>C</b>	130	84	-	142	205	198
<b>D</b>	323	196	142	-	111	87
<b>E</b>	310	286	205	111	-	85
<b>F</b>	323	286	198	87	85	-

- Pencarian rute dengan memiliki banyak titik



Gambar 4.6 Menggambar rute dengan banyak titik

Gambar diatas menunjukkan ketika titik tujuan yang diinginkan selesai digambar, maka program menunjukkan beberapa alternative jalan yang bisa digunakan untuk sampai ke titik tujuan.

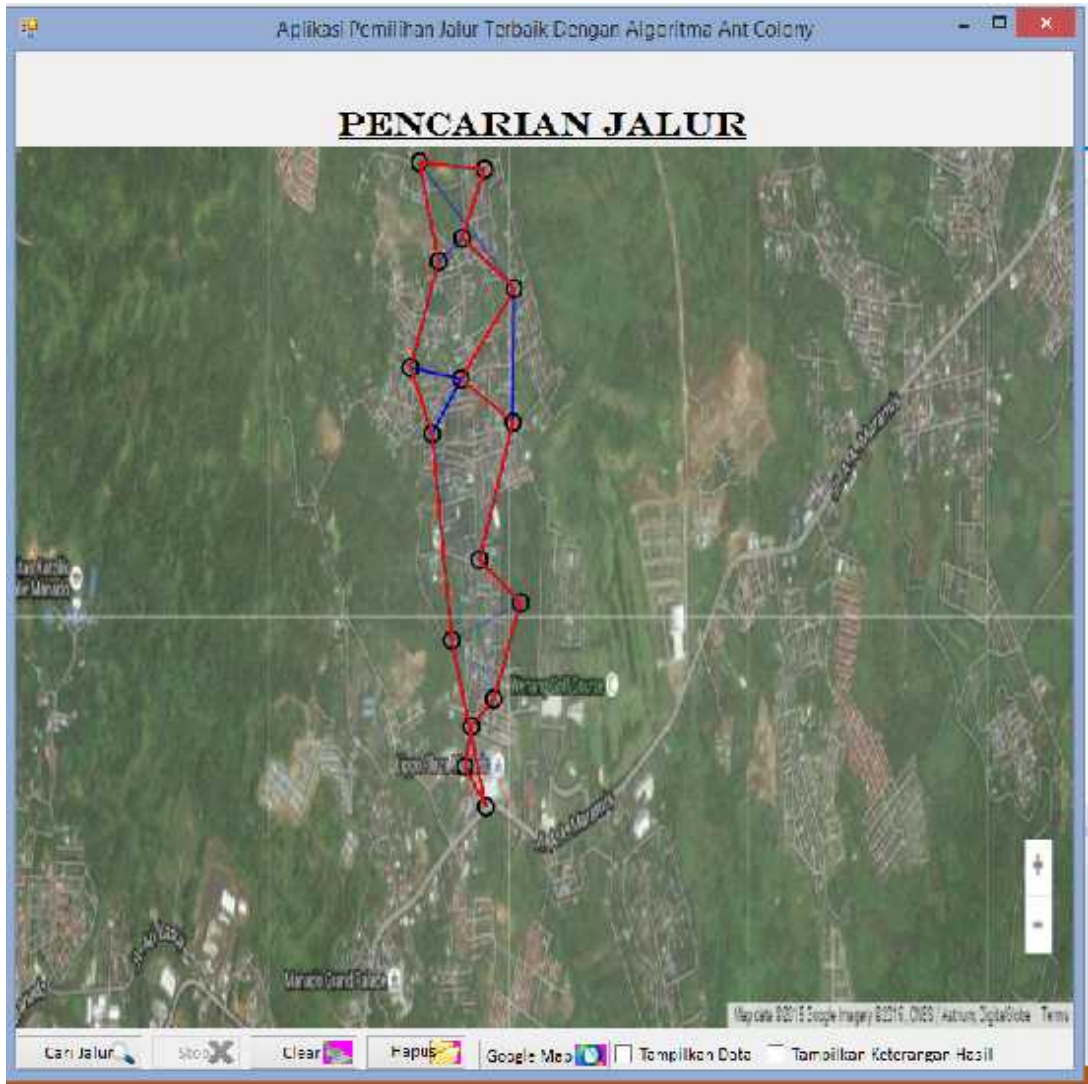
- Proses Pencarian Rute oleh program



Gambar 4.7 Proses pencarian rute oleh computer

Pada gambar diatas menunjukkan proses pencarian jalur terdekat yang dilakukan oleh aplikasi. Dimana jalur terdekat didapat dari hasil perhitungan dengan algoritma ant coloni.





Gambar 4.8 Rute Dengan Banyak Titik Ditemukan

Dari pengujian diatas, diketahui program akan tetap melakukan pencarian jalur dengan banyak titik meskipun akan memakan waktu lama untuk pencarian. Dan juga terjadi error karena ketika jalur telah ditemukan, jalur seharusnya ditunjukkan dengan warna merah sedangkan pada gambar tersebut masih terdapat garis berwarna biru yang menunjukkan bahwa program masih melakukan pencarian jalur.

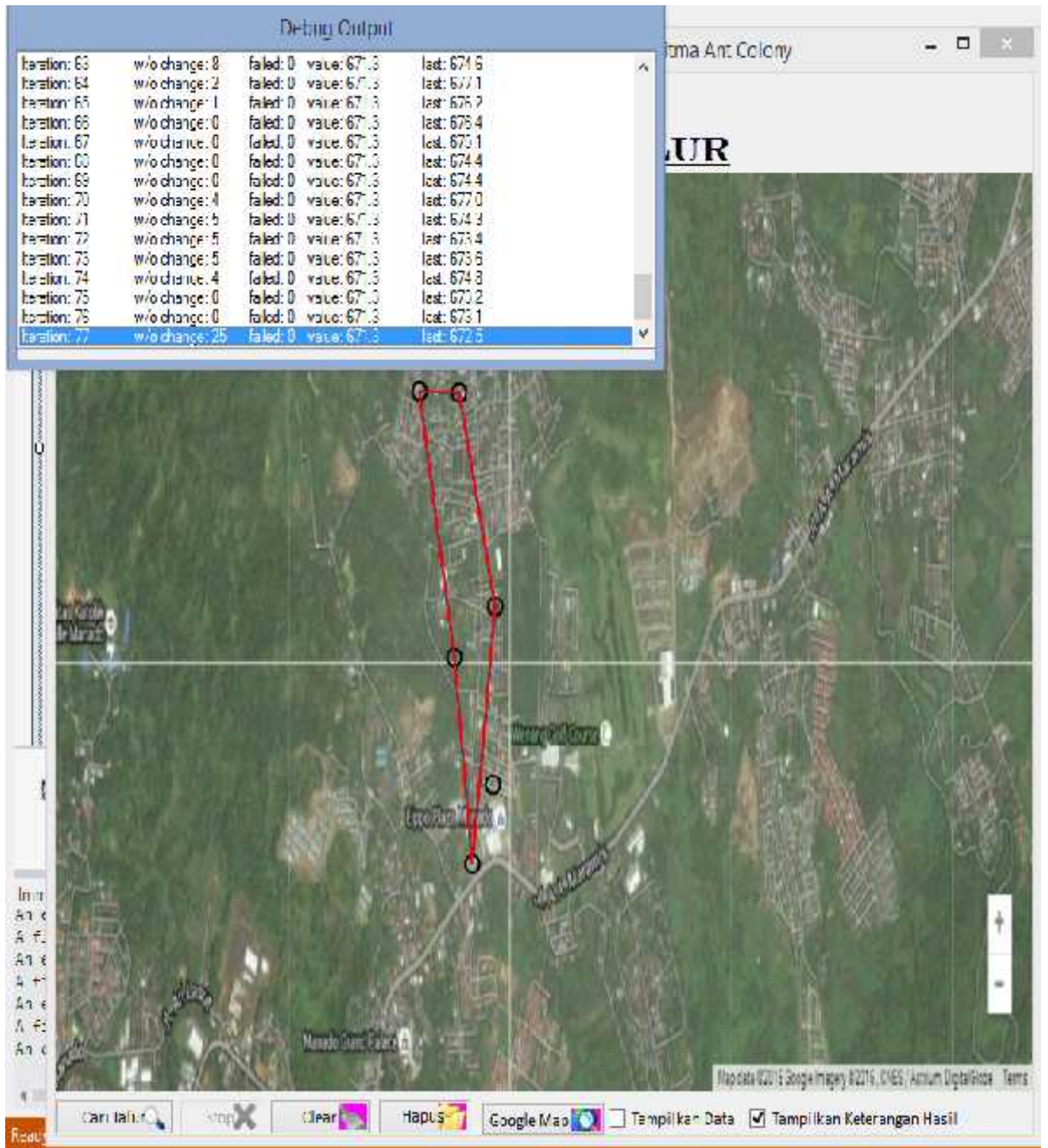
c. Fungsi Checkbox tampilkan data



Gambar 4.9 Gambar ketika checkbox tampilkan data diklik

Ketika checkbox tampilkan data di klik maka akan muncul keterangan titik x dan titik y serta jarak dari masing-masing titik yang akan menjadi acuan untuk pencarian jalur terpendek. Jarak dari masing-masing titik ditunjukkan dengan angka berwarna biru.

d. Fungsi Checkbox tampilkan hasil iterasi



Gambar 4.10 Tampilkan hasil iterasi

Fungsi dari checkbox ini adalah ketika di klik, maka akan menampilkan hasil banyaknya iterasi nilai awal serta nilai akhir yang di dapat.

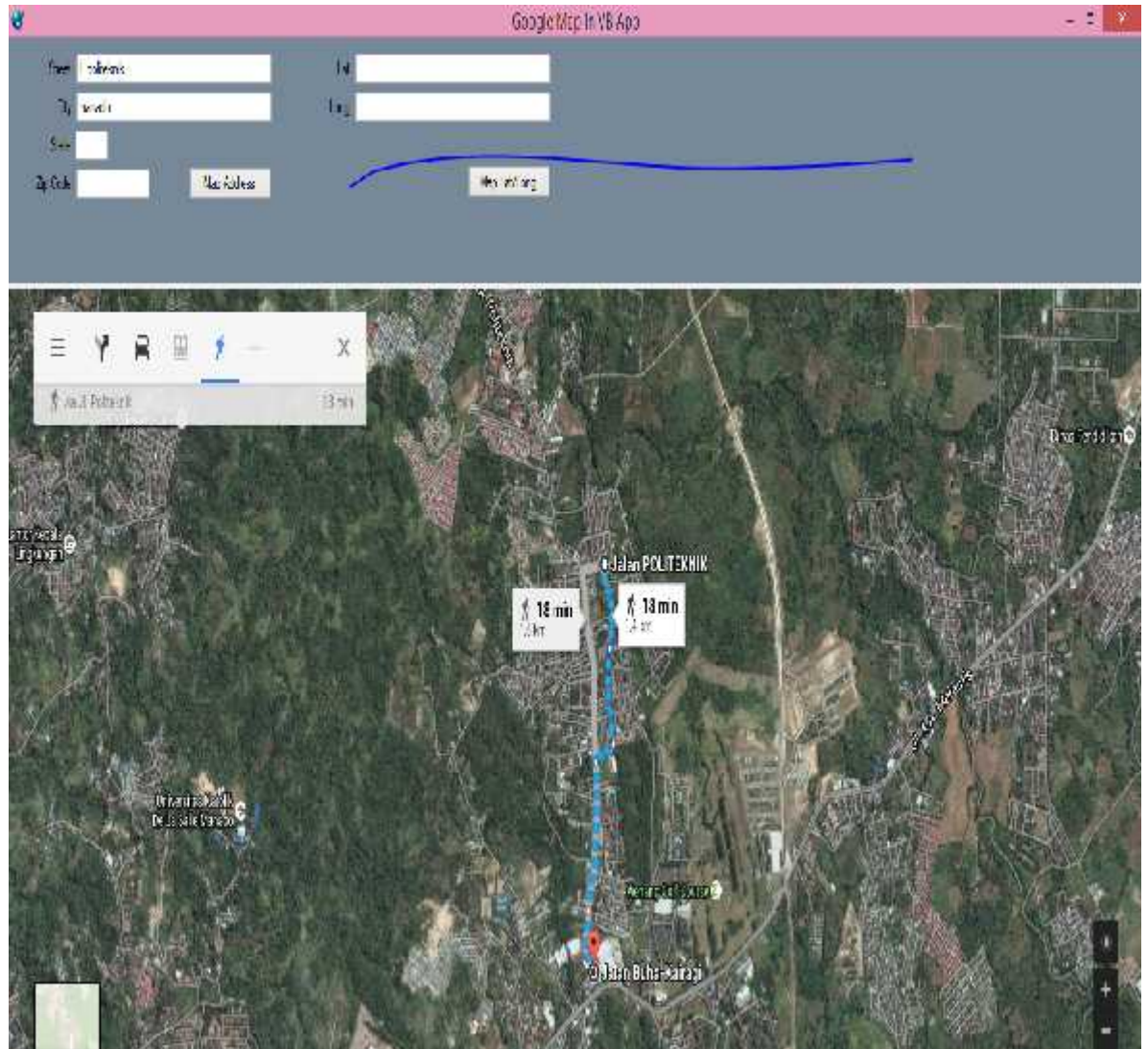
## 2. Form Google Map



Gambar 4.11 Form Google Map

Pada form ini, pengguna bisa melihat lebih dekat lokasi dari jalur terpendek yang telah ditentukan oleh aplikasi algoritma semut pada form sebelumnya. Pada form ini, pengguna bisa mencari lokasi berdasarkan longitude dan latitude lokasi tersebut.

Berikut adalah gambar pencarian jalur menggunakan map dengan menentukan 2 titik tujuan.



Gambar 4.12 Pencarian jalur 1

Gambar diatas adalah gambar pencarian jalur dengan menggunakan Google Maps API melalui program VB.Net yang telah dibuat. Diketahui bahwa aplikasi menemukan 2 jalur untuk menuju ke titik tujuan dimana jalur pertama berjarak 1.5 Km dan jalur ke dua 1.4 Km jadi aplikasi memilih jalur atau rute ke 2 karena memiliki jarak yang lebih pendek dibandingkan jalur 1.



Gambar 4.13 Pencarian jalur 2

Pada pencarian jalur yang ke 2, menggunakan teknik yang sama dengan pencarian jalur 1 namun titik tujuan berbeda. Diketahui aplikasi menunjukkan beberapa jalur yang bisa ditempuh untuk sampai ke titik tujuan serta jarak dari masing-masing jalur yang akan ditempuh. Jalur efektif yang diberikan oleh aplikasi ditunjukkan dengan garis putus-putus berwarna biru.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Dari hasil pembuatan aplikasi penentuan jalur terpendek penarikan kabel fiber optic diatas, diperoleh kesimpulan sebagai berikut:

- Aplikasi ini dibangun dengan menggunakan pemograman Visual Basic 2012 untuk melakukan pencarian jalur terpendek dengan menggunakan algoritma ant colony.
- Proses pencarian jalur dengan metode ini, membutuhkan waktu yang lebih lama untuk menemukan jalur terpendek.
- Jalur terpendek yang di dapat dalam pencarian dengan aplikasi ini di tunjukkan dengan garis merah dari titik satu ke titik yang lain.
- Untuk pencarian jalur dalam *maps* menggunakan 2 titik dan jalur terpendek yang didapat ditunjukkan oleh garis putus-putus berwarna biru.
- Jarak dalam aplikasi ini dinyatakan dalam bentuk Meter dan Kilometer.

#### **4.2 Saran**

Aplikasi ini dibuat untuk memantu proses pencarian jalur terpendek pemasangan kabel fiber optic dimana jalur terpendek didapa dari perhitungan jalur dari titik satu ke titik yang lain. Dimana titik-titik tersebut dihubungkan oleh garis. Oleh karena keterbatasan yang dimiliki oleh aplikasi ini, dimana aplikasi ini hanya untuk pemasangan jalur baru, penulis berharap system ini terus dikembangkan kedepannya seperti menambah fitur untuk proses penyambungan kabel yang sudah ada serta menambah fitur untuk menampilkan jalur-jalur yang telah terpasang. Besar harapan penulis agar

aplikasi Pencarian Jalur Terpendek Penarikan Kabel Fiber Optik Menggunakan Algoritma Semut ini dapat dikembangkan menjadi lebih sempurna lagi dan bisa menjadi suatu produk yang dapat membantu petugas/pegawai PT. Telkom Indonesia dalam penarikan kabel fiber optic.



## ▪ CLASS ANT

Public Class Ant

```
Private Shared ReadOnly m_Random As New Random()  
Private m_Pheromones As Double  
Private ReadOnly m_VisitedCities As List(Of City)  
Private m_CurrentPosition As City  
Private m_TourValue As Double  
  
Private Shared ReadOnly Property Random() As Random  
    Get  
        Return m_Random  
    End Get  
End Property  
  
Public ReadOnly Property VisitedCities() As IList(Of City)  
    Get  
        Return m_VisitedCities  
    End Get  
End Property  
  
Public Property CurrentPosition() As City  
    Get  
        Return m_CurrentPosition  
    End Get  
    Private Set(ByVal value As City)  
        m_CurrentPosition = value  
    End Set  
End Property  
  
Public Property Pheromones() As Double  
    Get  
        Return m_Pheromones  
    End Get  
    Set(ByVal value As Double)  
        m_Pheromones = value  
    End Set  
End Property  
  
Public Property TourValue() As Double  
    Get  
        Return m_TourValue  
    End Get  
    Private Set(ByVal value As Double)  
        m_TourValue = value  
    End Set  
End Property  
  
Public Sub New(ByVal start_position As City, ByVal pheromones As Double)  
    m_CurrentPosition = start_position  
    m_Pheromones = pheromones  
    m_VisitedCities = New List(Of City)()  
    VisitedCities.Add(start_position)
```

```

End Sub

Public Function SearchTour() As Boolean
    TourValue = 0.0

    Do While TravelOn()
    Loop

    Dim closing_road = CurrentPosition.Roads(VisitedCities.First())
    If closing_road IsNot Nothing Then
        TourValue += closing_road.Distance
        Return True
    End If

    Return False
End Function

Private Function TravelOn() As Boolean
    Dim next_city = GetNextCity()
    If next_city Is Nothing Then
        Return False
    End If

    CurrentPosition = next_city
    TourValue += VisitedCities.Last().Roads(CurrentPosition).Distance
    VisitedCities.Add(CurrentPosition)
    Return True
End Function

Private Function GetNextCity() As City
    Dim city_weights As New Dictionary(Of City, Double)()
    Dim sum_of_weights = 0.0

    For Each city In CurrentPosition.NeighbourCities
        If Not VisitedCities.Contains(city) Then
            Dim weight = CurrentPosition.Roads(city).WeighedValue
            city_weights.Add(city, weight)
            sum_of_weights += weight
        End If
    Next

    Dim rnd = Random.NextDouble()
    Dim sum = 0.0

    For Each pair In city_weights
        sum += pair.Value / sum_of_weights
        If sum >= rnd Then
            Return pair.Key
        End If
    Next

    Return Nothing
End Function
End Class

```

## ▪ CLASS CITY

Imports System.Diagnostics

```
<DebuggerDisplay("{Name}")> _  
Public Class City
```

```
    Private ReadOnly m_Name As String  
    Private ReadOnly m_Roads As Dictionary(Of City, Road)
```

```
    Public ReadOnly Property Name() As String  
        Get  
            Return m_Name  
        End Get  
    End Property
```

```
    Public ReadOnly Property NeighbourCities() As IEnumerable(Of City)  
        Get  
            Return m_Roads.Keys  
        End Get  
    End Property
```

```
    Public ReadOnly Property Roads(ByVal [to] As City) As Road  
        Get  
            Dim ret As Road = Nothing  
            m_Roads.TryGetValue([to], ret)  
            Return ret  
        End Get  
    End Property
```

```
    Public Sub New(ByVal name As String)  
        m_Name = name  
        m_Roads = New Dictionary(Of City, Road)()  
    End Sub
```

```
    Friend Sub AddRoad(ByVal road As Road, ByVal other_city As City)  
        m_Roads.Add(other_city, road)  
    End Sub
```

```
End Class
```

## ▪ CLASS ROAD

Imports System.Diagnostics

```
<DebuggerDisplay("Distance: {Distance}, Pheromone level: {PheromoneLevel}")> _  
Public Class Road
```

```
    Private Const  $\alpha$  As Double = World. $\alpha$   
    Private Const  $\beta$  As Double = World. $\beta$   
    Private m_Distance As Double  
    Private m_PheromoneLevel As Double
```

```
    Public ReadOnly Property Distance() As Double  
        Get
```

```

        Return m_Distance
    End Get
End Property

Public Property PheromoneLevel() As Double
    Get
        Return m_PheromoneLevel
    End Get
    Set(ByVal value As Double)
        m_PheromoneLevel = value
    End Set
End Property

Public ReadOnly Property WeighedValue() As Double
    Get
        Return Distance ^  $\alpha$  * PheromoneLevel ^  $\beta$ 
    End Get
End Property

Public Sub New(ByVal distance As Double)
    m_Distance = distance
End Sub
End Class

```

#### ▪ CLASS UPDATEEVENTARGG

```

Public Class UpdateEventArgs
    Inherits EventArgs

    Private ReadOnly m_CurrentIteration As Integer
    Private ReadOnly m_SuccessfulIterations As Integer
    Private ReadOnly m_Failures As Integer
    Private ReadOnly m_CurrentBestValue As Double
    Private ReadOnly m_LastValue As Double
    Private ReadOnly m_BestTour As IEnumerable(Of City)

    Public ReadOnly Property CurrentIteration() As Integer
        Get
            Return m_CurrentIteration
        End Get
    End Property

    Public ReadOnly Property SuccessfulIterations() As Integer
        Get
            Return m_SuccessfulIterations
        End Get
    End Property

    Public ReadOnly Property Failures() As Integer
        Get
            Return m_Failures
        End Get
    End Property

```

```

Public ReadOnly Property CurrentBestValue() As Double
    Get
        Return m_CurrentBestValue
    End Get
End Property

Public ReadOnly Property LastValue() As Double
    Get
        Return m_LastValue
    End Get
End Property

Public ReadOnly Property BestTour() As IEnumerable(Of City)
    Get
        Return m_BestTour
    End Get
End Property

Public Sub New( _
    ByVal current_iteration As Integer, _
    ByVal successful_iterations As Integer, _
    ByVal failures As Integer, _
    ByVal current_best_value As Double, _
    ByVal last_value As Double, _
    ByVal best_tour As IEnumerable(Of City) _
)
    m_CurrentIteration = current_iteration
    m_SuccessfulIterations = successful_iterations
    m_Failures = failures
    m_CurrentBestValue = current_best_value
    m_LastValue = last_value
    m_BestTour = best_tour
End Sub

Public Overrides Function ToString() As String
    Return String.Format( _
        "Iteration: {1}{0}w/o change: {2}{0}failed: {3}{0}value:
{4:f1}{0}last: {5:f1}", _
        ControlChars.Tab, CurrentIteration, SuccessfulIterations,
Failures, CurrentBestValue, LastValue _
    )
End Function
End Class

```

## ▪ CLASS WORLD

```

Public Class World

    Public Const  $\alpha$  As Double = -1.5
    Public Const  $\beta$  As Double = 1.5
    Public Const NumberOfAnts As Integer = 100
    Public Const InitialPheromoneValue As Double = 1
    Public Const PheromoneDecayFactor As Double = 0.1

```

```

Public Const MaxIterations As Integer = 20

Public Event Update(ByVal sender As World, ByVal e As UpdateEventArgs)

Private Shared ReadOnly m_Random As New Random()
Private ReadOnly m_Cities As List(Of City)
Private ReadOnly m_Roads As List(Of Road)
Private ReadOnly m_WorstTourValue As Double
Private ReadOnly m_WeightingFactor As Double

Public ReadOnly Property Cities() As IList(Of City)
    Get
        Return m_Cities
    End Get
End Property

Public ReadOnly Property Roads() As IEnumerable(Of Road)
    Get
        Return m_Roads
    End Get
End Property

Public Sub New(ByVal prototype As WorldBuilder)
    Me.New(prototype.Cities, prototype.Roads)
End Sub

Public Sub New(ByVal cities As IEnumerable(Of City), ByVal roads As
IEnumerable(Of Road))
    m_Cities = New List(Of City)(cities)
    m_Roads = New List(Of Road)(roads)
    m_WorstTourValue = Aggregate road In roads Into Sum(road.Distance)
    m_WeightingFactor = cities.Count / roads.Count
End Sub

Public Function FindTour() As IEnumerable(Of City)
    Dim best_tour As IList(Of City) = Nothing
    Dim best_tour_value = m_WorstTourValue
    Dim iterations = 0
    Dim iterations_without_change = 0
    Dim number_of_failures = 0
    Dim last_value As Double
    Dim num_success As Integer

    For Each road In Roads
        road.PheromoneLevel = InitialPheromoneValue
    Next

    Dim ant_pheromone_capacity = 0.2
    Dim overall_decay_value = PheromoneDecayFactor * InitialPheromoneValue
    * m_Roads.Count

    Do While iterations_without_change < MaxIterations
        iterations += 1
        Dim ants As New List(Of Ant)(NumberOfAnts)
        Dim ant_success As New Dictionary(Of Ant, Boolean)(NumberOfAnts)

```

```

For i = 1 To NumberOfAnts
    Dim rnd_index = m_Random.Next(Cities.Count)
    ants.Add(New Ant(Cities(rnd_index), ant_pheromone_capacity))
Next

last_value = 0
num_success = 0

For Each ant In ants
    Dim success = ant.SearchTour() AndAlso ant.VisitedCities.Count
= m_Cities.Count
    ant_success(ant) = success

    If success Then
        ' We use a delta to compensate mathematical instabilities
in
        ' floating point operations.
        Dim delta = ant.TourValue - best_tour_value

        If Math.Abs(delta) <= 0.01 Then
            iterations_without_change += 1
        ElseIf delta < 0 Then
            best_tour_value = ant.TourValue
            best_tour = ant.VisitedCities
            iterations_without_change = 0
        ElseIf delta <= best_tour_value * 0.01 Then

            iterations_without_change += 1
        Else
            iterations_without_change = 0
        End If

        last_value += ant.TourValue
        num_success += 1
    Else
        iterations_without_change = 0
        number_of_failures += 1
    End If
Next

last_value /= num_success

For Each road In Roads

    Dim road_is_in_best_tour = False

    If best_tour IsNot Nothing AndAlso best_tour.Count > 0 Then
        Dim first = best_tour(0)

        For i = 1 To best_tour.Count - 1
            If first.Roads(best_tour(i)) Is road Then
                road_is_in_best_tour = True
                Exit For
            End If
        Next
    End If
Next

```

```

        End If

        first = best_tour(i)
    Next

    If best_tour(best_tour.Count - 1).Roads(best_tour(0)) Is
road Then
        road_is_in_best_tour = True
    End If
End If

If Not road_is_in_best_tour Then
    UpdatePheromoneLevel(road)
End If
Next

Dim individual_pheromone_level = overall_decay_value

For Each annotated_ant In ant_success
    If annotated_ant.Value Then
        annotated_ant.Key.Pheromones = individual_pheromone_level
        Dim cities = annotated_ant.Key.VisitedCities
        Dim tour_bonus = TourPheromoneBonus(annotated_ant.Key)
        For i = 1 To cities.Count - 1
            Dim road = cities(i - 1).Roads(cities(i))
            road.PheromoneLevel += tour_bonus
        Next

        Dim last_road = cities(cities.Count - 1).Roads(cities(0))
        last_road.PheromoneLevel += tour_bonus
    End If
Next

RaiseUpdate( _
    New UpdateEventArgs( _
        iterations, iterations_without_change, number_of_failures,
-
        best_tour_value, last_value, best_tour _
    ) _
)
Loop

Return best_tour
End Function

Private Sub RaiseUpdate(ByVal args As UpdateEventArgs)
    RaiseEvent Update(Me, args)
End Sub

Private Sub UpdatePheromoneLevel(ByVal road As Road)
    Const RemainingPheromoneFactor As Double = 1.0 - PheromoneDecayFactor
    road.PheromoneLevel = road.PheromoneLevel * RemainingPheromoneFactor
End Sub

Private Function TourPheromoneBonus(ByVal ant As Ant) As Double

```



```

        ' We penalize long tours and try to get the worst possible tour
        ' to yield 0 pheromone.
        ' The square tries to "stretch" the range of possible bonuses.
        Return (ant.Pheromones * (m_WorstTourValue / ant.TourValue - 1)) ^ 2
    End Function

```

End Class

## ▪ CLASS WORLDBUILDER

Public Class WorldBuilder

```

    Private ReadOnly m_Cities As New Dictionary(Of String, City)()
    Private ReadOnly m_Roads As New List(Of Road)()

```

```

    Friend ReadOnly Property Cities() As IEnumerable(Of City)

```

```

        Get
            Return m_Cities.Values
        End Get

```

End Property

```

    Friend ReadOnly Property Roads() As IEnumerable(Of Road)

```

```

        Get
            Return m_Roads
        End Get

```

End Property

```

    Public Function AddCity(ByVal name As String) As City

```

```

        Dim city As New City(name)
        m_Cities.Add(name, city)
        Return city

```

End Function

```

    Public Sub AddCity(ByVal city As City)

```

```

        m_Cities.Add(city.Name, city)

```

End Sub

```

    Public Sub AddCities(ByVal names As IEnumerable(Of String))

```

```

        For Each name In names
            AddCity(name)

```

```

        Next

```

End Sub

```

    Public Sub AddCities(ByVal ParamArray names As String())

```

```

        For Each name In names
            AddCity(name)

```

```

        Next

```

End Sub

```

    Public Sub AddCities(ByVal cities As IEnumerable(Of City))

```

```

        For Each city In cities
            AddCity(city)

```

```

        Next

```

End Sub

```

Public Sub AddCities(ByVal ParamArray cities As City())
    For Each city In cities
        AddCity(city)
    Next
End Sub

Public Function AddRoad(ByVal distance As Double, ByVal from As City,
ByVal [to] As City) As Road
    Dim road As New Road(distance)
    from.AddRoad(road, [to])
    [to].AddRoad(road, from)
    m_Roads.Add(road)
    Return road
End Function

Public Function AddRoad(ByVal distance As Double, ByVal from As String,
ByVal [to] As String) As Road
    Dim from_city As City = Nothing

    If Not m_Cities.TryGetValue(from, from_city) Then
        from_city = AddCity(from)
    End If

    Dim to_city As City = Nothing
    If Not m_Cities.TryGetValue([to], to_city) Then
        to_city = AddCity([to])
    End If

    Return AddRoad(distance, from_city, to_city)
End Function

End Class

```

- **CLASS CITY**

```

<DebuggerDisplay("City {Name}")> _
Public Class City

    Public Const Radius As Integer = 6

    Private ReadOnly m_Location As Point
    Private ReadOnly m_Name As String
    Private m_TspCity As AntColonyTSP.City

    Public ReadOnly Property Location() As Point
        Get
            Return m_Location
        End Get
    End Property

    Public ReadOnly Property Name() As String
        Get
            Return m_Name
        End Get
    End Property

```

```

        End Get
    End Property

    Public Property TspCity() As AntColonyTSP.City
        Get
            Return m_TspCity
        End Get
        Set(ByVal value As AntColonyTSP.City)
            m_TspCity = value
        End Set
    End Property

    Public Sub New(ByVal location As Point, ByVal name As String, ByVal
tsp_city As AntColonyTSP.City)
        m_Location = location
        m_Name = name
        m_TspCity = tsp_city
    End Sub

    Public Shared Function Distance(ByVal pt1 As Point, ByVal pt2 As Point) As
Double
        Return Math.Sqrt((pt1.X - pt2.X) ^ 2 + (pt1.Y - pt2.Y) ^ 2)
    End Function

    Public Shared Function Distance(ByVal line As KeyValuePair(Of Point,
Point)) As Double
        Return Distance(line.Key, line.Value)
    End Function

End Class

```

## ▪ FORM DEBUGWINDOW

```

Public Class DebugWindow

    Public Shared ReadOnly Property Instance() As DebugWindow
        Get
            Static _instance As New DebugWindow()
            Return _instance
        End Get
    End Property

    Public Sub AddItem(ByVal text As String)
        If DebugList.Items.Count > 1000 Then
            DebugList.Items.RemoveAt(0)
        End If
        DebugList.Items.Add(text)
        DebugList.SetSelected(DebugList.Items.Count - 1, True)
    End Sub

    Public Sub Clear()
        DebugList.Items.Clear()
    End Sub

```

```

Private Sub DebugWindow_FormClosing(ByVal sender As Object, ByVal e As
FormClosingEventArgs) Handles Me.FormClosing
    If e.CloseReason = CloseReason.UserClosing Then
        e.Cancel = True
    End If
End Sub

Private Sub DebugWindow_Load(ByVal sender As Object, ByVal e As EventArgs)
Handles Me.Load
    DebugList.UseCustomTabOffsets = True
End Sub

Private Sub DebugList_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles DebugList.SelectedIndexChanged

End Sub
End Class

```

## ▪ FORM FILTERLOADHELPER

Option Strict Off

Imports System.Runtime.CompilerServices

Module FilterLoadHelper

```

Public Function SupportedPictureFilters() As String
    ' Load file filters
    Dim supportedExtensions As Object() = { _
        New With {.Key = "*.bmp", .Value = "Windows Bitmap"}, _
        New With {.Key = "*.gif", .Value = "Graphics Interchange Format"},
-
        New With {.Key = "*.png", .Value = "Portable Network Graphic"}, _
        New With {.Key = "*.jpg;*.jpeg", .Value = "JPEG file"}, _
        New With {.Key = "*.tif;*.tiff", .Value = "TIFF file"} _
    }

    Dim extensions = From x In supportedExtensions Select
DirectCast(x.Key, String)
    Dim descriptions = From x In supportedExtensions Select
DirectCast(x.Value, String)
    Dim pairs = From x In supportedExtensions _
        Select DirectCast(x.Value, String) & " (" &
DirectCast(x.Key, String) & ")"|" & DirectCast(x.Key, String)

    Dim ext As New System.Text.StringBuilder()

    ext.Append("All supported files|")
    ext.Append(String.Join(";", extensions.ToArray()))
    ext.Append("|"c)
    ext.Append(String.Join("|", pairs.ToArray()))
    Return ext.ToString()
End Function

```

End Module

## ▪ CLASS MAP REPAIR

Public Partial Class Map

```
Private Structure Pair(Of T1, T2)
    Public ReadOnly First As T1
    Public ReadOnly Second As T2

    Public Sub New(ByVal first As T1, ByVal second As T2)
        Me.First = first
        Me.Second = second
    End Sub
End Structure
```

End Class

## ▪ CLASS MAP

Public Class Map

```
Private WithEvents m_Display As PictureBox
Private m_Bitmap As Bitmap
Private m_BackgroundPicture As Image
Private ReadOnly m_Cities As New List(Of City)()
Private ReadOnly m_Roads As New List(Of Pair(Of City, City))()
Private m_CityMap As New Dictionary(Of AntColonyTSP.City, City)
Private m_RoadMap As Dictionary(Of Road, Pair(Of City, City))
Private m_ShowLabels As Boolean

Public Sub New(ByVal display As PictureBox)
    m_Display = display
    m_Bitmap = New Bitmap(display.Width, display.Height)
End Sub

Public Property ShowLabels() As Boolean
    Get
        Return m_ShowLabels
    End Get
    Set(ByVal value As Boolean)
        m_ShowLabels = value
        Redraw()
    End Set
End Property

Public Property BackgroundPicture() As Image
    Get
        Return m_BackgroundPicture
    End Get
    Set(ByVal value As Image)
        m_BackgroundPicture = value
        Redraw()
    End Set
End Property
```

```

        End Set
    End Property

    Public ReadOnly Property CityCount() As Integer
        Get
            Return m_Cities.Count
        End Get
    End Property

    Public Sub AddCity(ByVal location As Point)
        Dim city As New City(location, NameFromLocation(location), Nothing)

        For Each c In m_Cities
            m_Roads.Add(New Pair(Of City, City)(city, c))
        Next

        m_Cities.Add(city)
        Redraw()
    End Sub

    Public Function FindCity(ByVal location As Point) As City
        Return m_Cities.Find(Function(c) City.Distance(location, c.Location)
<= City.Radius * 2)
    End Function

    Public Sub RemoveCity(ByVal location As Point)
        Dim city = FindCity(location)
        m_Cities.Remove(city)

        m_Roads.RemoveAll(Function(road) road.First Is city OrElse road.Second
Is city)
        Redraw()
    End Sub

    Public Sub Clear()
        m_Cities.Clear()
        m_Roads.Clear()
        Redraw()
    End Sub

    Public Function ConstructTsp() As World
        Dim wb As New WorldBuilder()

        m_CityMap.Clear()
        m_CityMap = New Dictionary(Of AntColonyTSP.City, City)(m_Cities.Count)

        For Each c In m_Cities
            c.TspCity = wb.AddCity(c.Name)
            m_CityMap.Add(c.TspCity, c)
        Next

        m_RoadMap = New Dictionary(Of Road, Pair(Of City,
City))(CInt(m_Cities.Count ^ 2))

        For Each road In m_Roads

```

```

        m_RoadMap.Add( _
            wb.AddRoad( _
                City.Distance(road.First.Location, road.Second.Location),
                road.First.TspCity, road.Second.TspCity _
            ), road _
        )
    Next

    Return New World(wb)
End Function

Private Shared Function NameFromLocation(ByVal location As Point) As
String
    Return location.ToString()
End Function

Public Sub DrawBestTour(ByVal tour As IEnumerable(Of AntColonyTSP.City))
    If tour Is Nothing Then
        Return
    End If

    Using g = Graphics.FromImage(m_Bitmap)
        DrawTour(tour, g, Color.Red)
    End Using
    m_Display.Invalidate()
End Sub

Public Sub Redraw(Optional ByVal world As World = Nothing, Optional ByVal
e As UpdateEventArgs = Nothing)
    Using g = Graphics.FromImage(m_Bitmap)
        g.SmoothingMode = Drawing2D.SmoothingMode.HighQuality
        g.TextRenderingHint =
Drawing.Text.TextRenderingHint.AntiAliasGridFit
        '
        ' Draw the background picture.
        '
        If m_BackgroundPicture Is Nothing Then
            g.Clear(Color.White)
        Else
            g.DrawImage(m_BackgroundPicture, 0, 0, m_Bitmap.Width,
m_Bitmap.Height)
        End If
        '
        ' Draw the roads.
        '
        If world Is Nothing Then
            ' We're not currently running the heuristic: Draw standard
roads.
            Using p As New Pen(Color.FromArgb(26, Color.Blue), 2)
                For Each road In m_Roads

```

```

        g.DrawLine(p, road.First.Location,
road.Second.Location)
        Next
    End Using
Else
    ' While running the heuristic, we don't draw the usual roads.
    ' Instead, we draw the roads according to their pheromone
level.

    Dim sum_of_pheromones = Aggregate road In world.Roads Into
Sum(road.PheromoneLevel)
    Dim factor = 255 * world.Roads.Count

    For Each road In world.Roads
        Dim line = m_RoadMap(road)
        Dim alpha = Math.Min(Math.Max(CInt(road.PheromoneLevel /
sum_of_pheromones * factor), 0), 255)
        Using p As New Pen(Color.FromArgb(alpha, Color.Blue), 2)
            g.DrawLine(p, line.First.Location,
line.Second.Location)
        End Using
    Next
End If

'
' Draw the cities.
'

Using p As New Pen(Color.Black, City.Radius / 3)
    For Each c In m_Cities
        g.DrawEllipse(p, c.Location.X - City.Radius, c.Location.Y
- City.Radius, City.Radius * 2, City.Radius * 2)
    Next
End Using

'
' Draw the labels.
'

If ShowLabels Then
    Using b As New SolidBrush(Color.FromArgb(128, Color.Blue))
        For Each road In m_Roads
            Dim text_pos As _
                New PointF( _
                    (road.First.Location.X +
road.Second.Location.X) / 2.0F, _
                    (road.First.Location.Y +
road.Second.Location.Y) / 2.0F _
                )
            g.DrawString( _
                City.Distance(road.First.Location,
road.Second.Location).ToString("0"), _
                m_Display.Font, b, text_pos _
            )
        Next
    End Using
End If

```



```

        End Using

        Using b As New SolidBrush(Color.Black)
            For Each c In m_Cities
                Dim text_pos = c.Location
                text_pos.Offset(City.Radius + 5, 0)
                g.DrawString(c.Name,
m_Display.Font, b, text_pos)
            Next
        End Using
    End If

    '
    ' If we're running the heuristic, draw the currently best tour.
    '

    If e IsNot Nothing Then
        If e.BestTour IsNot Nothing Then
            DrawTour(e.BestTour, g, Color.FromArgb(192,
Color.DarkGreen))
        End If
    End If
End Using

    m_Display.Invalidate()
End Sub

Private Sub DrawTour(ByVal tour As IEnumerable(Of AntColonyTSP.City),
ByVal g As Graphics, ByVal color As Color)
    Using p As New Pen(color, 2)
        Dim i = tour.GetEnumerator()
        i.MoveNext()
        Dim first = m_CityMap(i.Current)
        Dim c1 = first

        Do While i.MoveNext()
            Dim c2 = m_CityMap(i.Current)
            g.DrawLine(p, c1.Location, c2.Location)
            c1 = c2
        Loop

        g.DrawLine(p, c1.Location, first.Location)
    End Using
End Sub

Private Sub Display_Resize(ByVal sender As Object, ByVal e As EventArgs)
Handles m_Display.Resize
    m_Bitmap = New Bitmap(m_Display.Width, m_Display.Height)
    Redraw()
End Sub

Private Sub Display_Paint(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles m_Display.Paint
    e.Graphics.DrawImageUnscaled(m_Bitmap, 0, 0)
End Sub

```

End Class

## ▪ FORM 1

Imports System.Text

Public Class Form1

Private Sub btnMapIt\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMapIt.Click

Try

```
Dim street As String = String.Empty
Dim city As String = String.Empty
Dim state As String = String.Empty
Dim zip As String = String.Empty
```

```
Dim queryAddress As New StringBuilder()
queryAddress.Append("http://maps.google.com/maps?q=")
```

```
' build street part of query string
If txtStreet.Text <> String.Empty Then
    street = txtStreet.Text.Replace(" ", "+")
    queryAddress.Append(street + "," & "+")
End If
```

```
' build city part of query string
If txtCity.Text <> String.Empty Then
    city = txtCity.Text.Replace(" ", "+")
    queryAddress.Append(city + "," & "+")
End If
```

```
' build state part of query string
If txtState.Text <> String.Empty Then
    state = txtState.Text.Replace(" ", "+")
    queryAddress.Append(state + "," & "+")
End If
```

```
' build zip code part of query string
If txtZipCode.Text <> String.Empty Then
    zip = txtZipCode.Text.ToString()
    queryAddress.Append(zip)
End If
```

```
' pass the url with the query string to web browser control
webBrowser1.Navigate(queryAddress.ToString())
```

Catch ex As Exception

```
MessageBox.Show(ex.Message.ToString(), "Unable to Retrieve Map")
```

```

        End Try
    End Sub

    Private Sub btnMapLatLong_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMapLatLong.Click

        If txtLat.Text = String.Empty Or txtLong.Text = String.Empty Then
Data")
            MessageBox.Show("Supply a latitude and longitude value.", "Missing
            End If

        Try
            Dim lat As String = String.Empty
            Dim lon As String = String.Empty

            Dim queryAddress As New StringBuilder()
            queryAddress.Append("http://maps.google.com/maps?q=")

            ' build latitude part of query string
            If txtLat.Text <> String.Empty Then
                lat = txtLat.Text
                queryAddress.Append(lat + "%2C")
            End If

            ' build longitude part of query string
            If txtLong.Text <> String.Empty Then
                lon = txtLong.Text
                queryAddress.Append(lon)
            End If

            webBrowser1.Navigate(queryAddress.ToString())

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error")

        End Try

    End Sub

    Private Sub webBrowser1_DocumentCompleted(sender As Object, e As
WebBrowserDocumentCompletedEventArgs) Handles webBrowser1.DocumentCompleted

    End Sub
End Class

```

- **MAIN FORM**

```

Imports PtPair = System.Collections.Generic.KeyValuePair(Of
System.Drawing.Point, System.Drawing.Point)

Public Class MainForm

    Private m_Map As Map
    Private m_TspThread As Threading.Thread

    Private Sub MapPicture_MouseUp(ByVal sender As Object, ByVal e As
MouseEventArgs) Handles MapPicture.MouseUp
        Select Case e.Button
            Case Windows.Forms.MouseButtons.Left
                If m_Map.FindCity(e.Location) Is Nothing Then
                    m_Map.AddCity(e.Location)
                End If
            Case Windows.Forms.MouseButtons.Right
                m_Map.RemoveCity(e.Location)
        End Select
    End Sub

    Private Sub StartButton_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles StartButton.Click
        StopTsp()

        If m_Map.CityCount < 4 Then
            MsgBox("Masukan 4 Titik Tujuan", MsgBoxStyle.Information, "Error")
            Return
        End If

        StopButton.Enabled = True
        StartButton.Enabled = False
        m_TspThread = New Threading.Thread(AddressOf StartTsp)
        m_TspThread.IsBackground = True
        m_TspThread.Start()
    End Sub

    Private Sub StopButton_Click(ByVal sender As Object, ByVal e As EventArgs)
Handles StopButton.Click
        StopTsp()
    End Sub

    Private Sub ClearButton_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles ClearButton.Click
        StopTsp()
        m_Map.Clear()
    End Sub

    Private Sub ShowLabelsCheck_CheckedChanged(ByVal sender As Object, ByVal e
As EventArgs) Handles ShowLabelsCheck.CheckedChanged
        m_Map.ShowLabels = ShowLabelsCheck.Checked
    End Sub

    Private Sub MainForm_Load(ByVal sender As Object, ByVal e As EventArgs)
Handles Me.Load
        OpenFileDialog.Filter = SupportedPictureFilters()
    End Sub

```

```

        StopButton.Enabled = False
        m_Map = New Map(MapPicture)
    End Sub

    Private Sub MainForm_FormClosing(ByVal sender As Object, ByVal e As
    FormClosingEventArgs) Handles Me.FormClosing
        StopTsp()
    End Sub

    Private Sub ShowDebugCheck_CheckedChanged(ByVal sender As Object, ByVal e
    As EventArgs) Handles ShowDebugCheck.CheckedChanged
        DebugWindow.Instance.Visible = ShowDebugCheck.Checked
    End Sub

    Private Sub LoadBackgroundButton_Click(ByVal sender As Object, ByVal e As
    EventArgs) Handles LoadBackgroundButton.Click
        If LoadBackgroundButton.Text = "Hapus Gambar" Then
            m_Map.BackgroundImage = Nothing
            LoadBackgroundButton.Text = "Pilih Gambar..."
        Else
            If OpenFileDialog.ShowDialog() = Windows.Forms.DialogResult.OK
            Then
                m_Map.BackgroundImage =
                Bitmap.FromFile(OpenFileDialog.FileName)
                LoadBackgroundButton.Text = "Hapus Gambar"
            End If
        End If
    End Sub

    Private Sub StartTsp()
        Invoke(New Action(AddressOf DebugWindow.Instance.Clear))

        Dim w = m_Map.ConstructTsp()
        AddHandler w.Update, AddressOf World_Update
        Dim best_tour = w.FindTour()
        Invoke(New Action(Of IEnumerable(Of AntColonyTSP.City))(AddressOf
        m_Map.DrawBestTour), best_tour)
        Invoke(New Action(AddressOf StopTsp))
    End Sub

    Private Sub StopTsp()
        StopButton.Enabled = False
        If m_TspThread IsNot Nothing AndAlso m_TspThread.IsAlive Then
            m_TspThread.Abort()
        End If
        StartButton.Enabled = True
    End Sub

    Private Sub World_Update(ByVal sender As World, ByVal e As
    UpdateEventArgs)
        If InvokeRequired Then
            Invoke(New Action(Of World, UpdateEventArgs)(AddressOf
            World_Update), sender, e)
            Threading.Thread.Sleep(100)
        Return
    End Sub

```

```
        End If

        m_Map.Redraw(sender, e)
        DebugWindow.Instance.AddItem(e.ToString())
    End Sub

    Private Sub MapPicture_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.Show()
    End Sub
End Class
```

▪ **GAMBAR MAPS JL.POLITEKNIK**



## DAFTAR PUSTAKA

**Alfred**, *Pengertian Fiber Optik*,

<https://alfredoeblog.wordpress.com/2014/01/16/pengertian-fiber-optik/> (Diakses 2 April 2015)

**Dani Satrio Kintoko**, 2013. *Pengertian Graf*.  
<http://danysatriokintoko.blogspot.com/2013/02/teori-dasar-graf.html> (Diakses 3 Agustus 2015)

**Hindriyanto Dwi Purnomo**, Ph.D, 2014. *Cara Mudah Belajar Metode Heuristik*. Penerbit Gava Media. Yogyakarta.

**Imran Zulmi Pratama**, 2011. *Membantu itu indah: Microsoft Visual Basic .Net*.  
<http://imranzulmi.blogspot.com/2011/04/microsoft-visual-basic-net.html> (diakses Juli 2015)

**Mursids**, *Algoritma Koloni Semut*, <http://mursids.blogspot.com/2009/12/algoritma-koloni-semut-aco.html> (Diakses 28 Maret 2015)

**PIP**, 2015. *Algoritma ACO (Ant Colony Optimization)*. <http://piptools.net/algoritma-aco-ant-colony-optimization/> (Diakses Juli 2015)

**PIP**, 2015. *Algoritma Dijkstra*. <http://piptools.net/algoritma-dijkstra/> (Diakses 1 Agustus 2015)

**Rizky**, S, 2013. *Visual Basic .Net Tutuorial*. [www.bukanSembarang.info](http://www.bukanSembarang.info) , (Diakses 21 Juni 2015)

**Rismon Hasiholan Sianipar**. 2015. *Kasus Dan Penyelesaian VB.Net*. Penerbit Informatika. Bandung.