

SKRIPSI

**APLIKASI TRAVELLING SALESMAN PROBLEM DENGAN METODE ANT
COLONY**

Oleh:

**JUNIATI
NIM. 1020000314**



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS POTENSI UTAMA
MEDAN
2014**

ABSTRAK

Permasalahan Travelling Salesman Problem (TSP) banyak terjadi pada perusahaan. Penyelesaian eksak untuk masalah TSP ini mengharuskan perhitungan terhadap semua kemungkinan rute yang dapat diperoleh, kemudian memilih salah satu rute yang terpendek sebagai jalur pilihan. Pada penelitian ini, penulis menggunakan Algoritma Ant Colony dengan bahasa pemrograman VB 2010 untuk menyelesaikan permasalahan pencarian rute terpendek. Ant Colony merupakan kumpulan algoritma-algoritma yang meniru koloni semut dalam mencari makanan melalui rute yang dilaluinya. Tujuan dari penulisan skripsi ini adalah mengetahui penerapan Ant Colony untuk permasalahan TSP. Dari hasil penulisan ini diperoleh bahwa Ant Colony mampu menyelesaikan permasalahan TSP dengan baik.

Kata Kunci : Travelling Salesman Problem (TSP), Ant Colony, VB 2010.

KATA PENGANTAR



Puji dan syukur penulis ucapkan ke hadirat Allah SWT, karena berkat rahmat dan ridhonya maka Pengetikan Skripsi ini dapat diselesaikan. Berdasarkan kurikulum yang telah ditetapkan oleh Program Studi Sistem Informasi STMIK Potensi Utama Medan, bahwa setiap mahasiswa yang akan mengakhiri perkuliahan diwajibkan menyelesaikan seluruh kegiatan termasuk melaksanakan Skripsi.

Skripsi ini tidak akan selesai dengan baik tanpa adanya bantuan dari berbagai pihak. Untuk itu, penulis mengucapkan terima kasih yang setulusnya dan sebesar-besarnya kepada:

1. Ibu Linda Wahyuni, M.Kom selaku Pembimbing I yang telah banyak membantu saya dan membimbing penulis dalam penyusunan skripsi ini.
2. Ibu Ratih Puspasari, M.Kom selaku Pembimbing II yang telah banyak membantu saya dan membimbing penulis dalam penyusunan skripsi ini.
3. Ibu Hj. Nuriandy, BA selaku pembina Yayasan Potensi Utama Medan.
4. Bapak Bob Subhan Riza, S.T selaku Ketua Yayasan Potensi Utama Medan.
5. Ibu Rika Rosnelly, SH, M.Kom, selaku Ketua STMIK Potensi Utama Medan.
6. Ibu Mas Ayoe Elhias, M.Kom selaku Ketua Program Studi Sistem Informasi STMIK Potensi Utama.
7. Ucapan terima kasih dan penghargaan yang tulus untuk kedua orang tua tercinta beserta keluarga yang telah membantu penulis baik moril maupun materil serta doa dan dukungan yang tiada hentinya.

8. Seluruh Keluarga Besar STMIK Potensi Utama Medan dan teman – teman SI-C malam yang telah memberikan bantuan baik informasi dan ilmu kepada penulis.

Penulis menyadari adanya kemungkinan terjadi kekeliruan ataupun kesalahan-kesalahan di dalam penyusunan skripsi ini. Oleh karena itu, penulis sangat mengharapkan saran dan kritik dari pembaca. Semoga skripsi ini dapat membawa manfaat yang sebesar-besarnya khususnya bagi penulis sendiri maupun bagi dunia pendidikan pada umumnya. Atas segala perhatiannya penulis mengucapkan terimakasih.

Medan, 22 Agustus 2014

Penulis,

Juniati

DAFTAR ISI

ABSTRAK

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
DAFTAR LAMPIRAN.....	ix

BAB I PENDAHULUAN..... 1

I.1. Latar Belakang.....	1
I.2. Ruang Lingkup Permasalahan	2
I.2.1. Identifikasi Masalah	2
I.2.2. Rumusan Masalah.....	2
I.2.3. Batasan Masalah	3
I.3. Tujuan dan Manfaat.....	3
I.3.1. Tujuan.....	3
I.3.2. Manfaat.....	3
I.4. Metodologi Penelitian.....	4
I.4.1. Analisa tentang sistem yang ada.....	5
I.4.2. Perbandingan sistem	7
I.5. Lokasi Penelitian	8
I.6. Sistematika Penulisan	8

BAB II TINJAUAN PUSTAKA 10

II.1. Sistem	10
II.1.1. Informasi	10
II.1.2. . Sistem Informasi	10
II.2. Sistem Pakar	10
II.2.1. Kelebihan Sistem Pakar	11
II.2.2. Elemen Manusia pada Sistem Pakar	12

II.2.3. Struktur Sistem Pakar	12
II.2.4. Karakteristik Sistem pakar	14
II.3. Metode Ant Colony	14
II.4. Visual Basic 2010.....	16
II.5. <i>Unifed Modelling Language</i>	16
II.5.1. <i>Use Case Diagram</i>	17
II.5.2. <i>Sequence Diagram</i>	18
II.5.3. <i>Class Diagram</i>	19
II.5.4. <i>Acivity Diagram</i>	20
BAB III ANALISIS DAN PERANCANGAN	22
III.1. Analisis Sistem yang Berjalan	22
III.1.1. Analisa <i>Input</i>	22
III.1.2. Analisa <i>Proses</i>	22
III.1.3. Analisa <i>Output</i>	24
III.2. Evaluasi Sistem yang Berjalan.....	24
III.3. Desain Sistem.....	25
III.3.1. Algoritma Ant Colony	25
III.3.1.1 Cara kerja semut menemukan rute.....	26
III.3.2. Desain Sistem Secara Global	32
III.3.2.1. <i>Usecase Diagram</i>	32
III.3.2.2. <i>Activity Diagram</i>	33
III.3.2.3. <i>Class Diagram</i>	35
III.3.2.4. <i>Sequence Diagram</i>	35
III.3.2. Desain Sistem Secara Detail	36
BAB IV HASIL DAN UJI COBA.....	39
IV.1. Tampilan Hasil	39
IV.2. Pembahasan	43
IV.3. Kelebihan dan Kelemahan sistem yang akan dirancang.....	43

BAB V	KESIMPULAN DAN SARAN	45
	V.1. Kesimpulan	45
	V.2. Saran.....	46

DAFTAR PUSTAKA

LAMPIRAN

LISTING PROGRAM

DAFTAR GAMBAR

Gambar I.1. Prosedur Perancangan	6
Gambar II.1. <i>Usecase</i> Diagram.....	18
Gambar II.2. <i>Sequence</i> Diagram.....	19
Gambar II.3. <i>Class</i> Diagram	19
Gambar III.4. <i>Activity</i> Diagram.....	21
Gambar III.1. <i>FOD</i> Sistem.....	23
Gambar III.2. Perjalanan Semut Dari Sarang ke Sumber Makanan	26
Gambar III.3. Graf G Berbobot.....	28
Gambar III.4. <i>Use Case</i> Diagram	32
Gambar III.5. <i>Activity</i> Diagram Input Peta	33
Gambar III.6. <i>Activity</i> Diagram Input Lokasi	33
Gambar III.7. <i>Activity</i> Diagram Mencari Solusi	34
Gambar III.8. <i>Activity</i> Diagram Membaca Data Dan Solusi	34
Gambar III.9. <i>Class</i> Diagram.....	35
Gambar III.10. <i>Sequence</i> Diagram.....	36
Gambar III.11. <i>Form</i> Utama	37
Gambar III.12. <i>Form</i> Debug Output.....	38
Gambar III.13. Desain <i>Form</i> Menu Utama.....	38
Gambar IV.1. Tampilan <i>Form</i> Utama.....	39
Gambar IV.2. Tampilan Peta Konsumen Di Medan.....	40

Gambar IV.3. Tampilan <i>Input</i> Konsumen Di Medan	41
Gambar IV.4. Tampilan Jarak Konsumen Di Medan	41
Gambar IV.5. Tampilan Halaman Rute Terpendek Di Medan.....	42
Gambar IV.6. Tampilan Halaman <i>Form Output</i> Di Medan.....	42

DAFTAR TABEL

Tabel III.1.	Matrik Nilai dari d_{ij} (jarak)	28
Tabel III.2.	Visibilitas $n = \frac{1}{d_{ij}}$	29
Tabel III.3.	Probabilitas kota untuk dikunjungi	31
Tabel III.4.	Panjang Jalur Ant Colony	32

DAFTAR LAMPIRAN

- Lampiran-1 Listing Program
- Lampiran-2 Surat Pengajuan Judul Skripsi
- Lampiran-3 Formulir Pendaftaran Judul Skripsi
- Lampiran-4 Surat Pernyataan Kesiadaan Pembimbing Skripsi I
- Lampiran-5 Surat Pernyataan Kesiadaan Pembimbing Skripsi II
- Lampiran-6 Surat Pernyataan Kesiadaan Pembimbing Riset
- Lampiran-7 Formulir Pendaftaran Seminar Skripsi
- Lampiran-8 Berita Acara Seminar Skripsi
- Lampiran-9. Formulir Pendaftaran Sidang Skripsi
- Lampiran-10 Surat Permohonan Izin Melakukan Riset dari Perusahaan
- Lampiran-11 Surat Keterangan Selesai Riset dari Perusahaan

DAFTAR PUSTAKA

Andioffset ,2010 , *Visual Basic 2010* ,Wahana Komputer, Semarang

A.S, Rosa, Shalahuddin, M. 2011. *Rekayasa Perangkat Lunak (Terstruktur dan berorientasi objek)*, Modula, Bandung.

Endah Damayanti, Supeno Mardi SN dan Moch.Hariadi, 2011. *Pencarian Jalur Terpendek Pada Pemodelan Pergerakan Agen Cerdas Dengan Algoritma Ant Colony System*. JSM Vol 2 No 2 Juni 2011

Rindra Yusianto dan Budi Setyo Utomo, 2009. *Rancang Bangun Software Simulasi Pendukung Keputusan Dengan Menggunakan Algoritma Ant Colony Sistem Pada Kasus Traveling Salesman Problem*. *Techno Science Vol 3 No.1 Mei 2009*

Priyantohidayatullah , 2014 , *Visual Basic.Net*, Informatika Bandung

Yuliyani Siyamtining Tyas dan Widodo Prijodiprodjo, 2013. *Aplikasi Pencarian Rute Terbaik dengan Metode Ant Colony Optimazation (ACO)*. *IJCCS*, Vol.7, No.1, January 2013

Listing Program

MainForm

```
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("At least 4 cities are needed.", 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()
    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 = "Clear pic" Then
        m_Map.BackgroundImage = Nothing
        LoadBackgroundButton.Text = "Picture ..."
    Else
        If OpenFileDialog.ShowDialog() = Windows.Forms.DialogResult.OK Then
            m_Map.BackgroundImage = Bitmap.FromFile(OpenFileDialog.FileName)
            LoadBackgroundButton.Text = "Clear pic"
        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 BeeColonyTSP.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 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) Handles MapPicture.Click
```

```
End Sub
End Class
```

Button start

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

    If m_Map.CityCount < 4 Then
        MsgBox("At least 4 cities are needed.", 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
```

Button Stop

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

Button Clear

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

Button Picture

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

```
End Sub
```

ShowLabels

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

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  
End 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 BeeColonyTSP.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

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 BeeColonyTSP.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 BeeColonyTSP.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
            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
            If world Is Nothing Then
                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

```

```

        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
    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

    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

        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 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 BeeColonyTSP.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

```