BAB III
METODE PENELITIAN

3.1 Objek Penelitian

Aplikasi penyusunan jadwal (*Ant Colony Timetable)* dapat menyusun jadwal secara automatis dengan dua mode pengguna yang berbeda, yaitu mode pengguna dan mode *adminitsrator*.

Pada mode pengguna, memiliki hak untuk membuka, menutup, dan menyimpan dokumen yang berisi pengaturan untuk penjadwalan. Pengaturan penjadwalan mencakup daftar nama kelas, matakuliah, dosen pengampu matakuliah, dan ruangan yang akan digunakan pada saat perkuliahan. File pengaturan ini, akan disimpan pada format yang sudah ditentukan.

Aplikasi penjadwalan dapat diaengaturan.kses oleh pengguna yang sudah terdaftar sebagai pengajar dalam file pengaturan yang relevan. Jika nama pengguna belum terdaftar, maka tidak akan bisa masuk ke dalam sistem. Secara default, nama pengguna untuk masuk adalah nama depan pengguna yang telah tercatat dalam file pengaturan. Berikut batasan yang mengatur aplikasi penjadwalan ini:

1. Studi kasus pada jurusan teknik informatika fakultas ilmu komputer universitas buana perjuangan karawang.
2. Metode yang dipakai menggunakan *Ant Colony Algorithm.*
3. Kapasitas ruangan.
4. Tidak meliputi matakuliah laboratorium.

Constrain kedua disebut sebagai *constraint lunak*. Kendala ini berperan untuk mengindikasikan tingkat keseimbangan suatu solusi. Dibawah ini terdapat *constrain* lunak yang berlaku dalam proses penjadwalan mata kuliah:

1. Mahasiswa sebaiknya tidak memiliki jadwal kelas melebihi waktu yang telah ditentukan.
2. Mahasiswa tidak diberikan lebih dari dua jadwal kelas secara berutuan dalam 1 hari.

3. Mahasiswa meiliki lebih dari satu jadwal kelas dalam satu hari.

3.2 Jadwal Penelitian

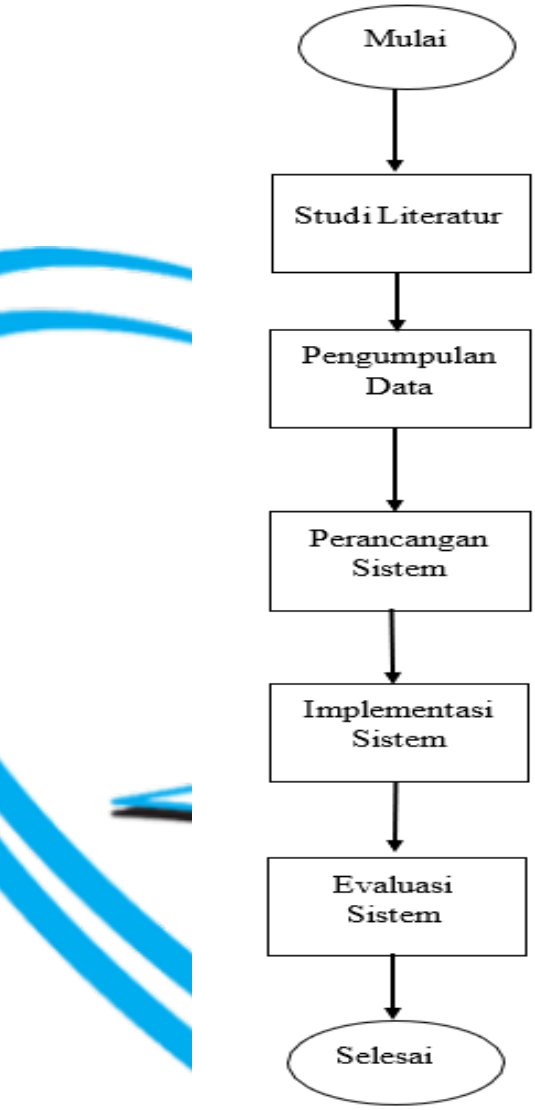
Dibawah ini terdapat tabel yang ebrisi jadwal penelitian. Dapat dilihat pada Tabel 3.1.

3.1 Tabel Jadwal Penelitian

Uraian	Bulan					
	1	2	3	4	5	6
Studi literatur	√					
Pengumpulan data	√	√				
Perancangan sistem			√	√		
Implementasi sistem				√	√	
Evaluasi sistem					√	√

3.3 Prosedur Penelitian

Pada bagian ini, akan dijelaskan tindakan yang dilakukan dalam rangkaian penelitian ini. Informasi terperinci mengenai urutan langkah-langkah tersebut dapat dilihat pada Gambar 3.1.



3.3.1. Studi Literatur

Penelitian ini berfokus pada eksplorasi yang didasarkan pada literatur ilmiah terkait penjadwalan mata kuliah di Lingkungan Universitas serta penerapan algoritma *Ant Colony Optimazaion* dalam penjadwalan waktu terkait.

3.3.2. Pengumpulan Data

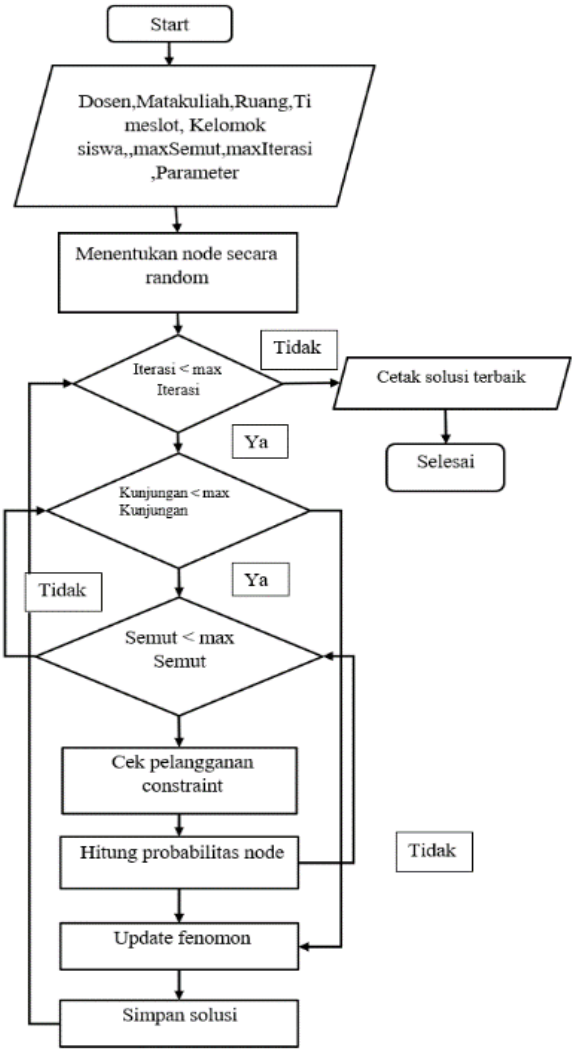
Informasi yang diperlukan dalam rangka penelitian ini, diperolah dari bagian administrasi Fakultas Ilmu Komputer di Universitas Buana Perjuangan Karawang.

3.3.3. Perancangan Sistem

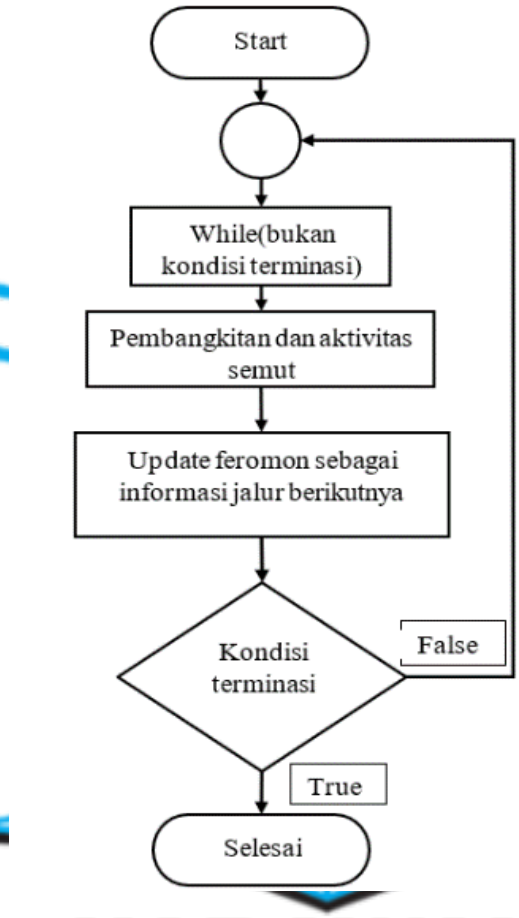
Sistem penjadwalan dirancang untuk digunakan oleh penyusun jadwal di program studi. *Input* data berupa matakuliah, ruang, dan hari kuliah. Perancangan sistem dibuat menggunakan *Use Case* diagram dan *Activity* diagram.

3.3.4. Implementasi Sistem

Dalam penelitian ini, bahasa pemograman java digunakan oleh penulis untuk mengimplementasikan algoritma *Ant Colony*. tahapan dalam algoritma seperti *flowchart* berikut:



Gambar 3.2 *Flowchart* Algoritma Semut



Gambar 3.3 *Flowchart* Algoritma Semut

Pada Gambar 3.2 dan Gambar 3.3 menjelaskan tentang operasi sistem yang akan dibuat oleh penulis. Proses dimulai dengan langkah inout data kegiatan mata kuliah seperti dosen, matakuliah, ruangan dan paramater yang dibutuhkan untuk menggunakan algoritma. Selanjutnya, pengecekan pada iterasi, kunjungan dan semut yang sedang berjalan melebihi batas maksimal yang telah diinput sebelumnya. Setelah dilakukan pengecekan, akan ada pembatasan pada proses penjadwalan.

Sebelum memulai pembahasan mengenai langkah-langkah algoritma, terdapat beberapa konstanta atau parameter yang perlu diatur:

Alpha dan Beta adalah faktor yang digunakan dlaam perhitungan taueta. Semakin besar nilai alpha, maka nilai taueta juga semakin besar. Semakin besar nilai beta, maka nilai taueta menjadi semakin kecil.

private static final int alpha = 3;
private static final int beta = 2;

Rho dan Q adalah koefisien yang digunakan dalam perhitungan feromon. Semakin besar nilai rho, maka nilai feromon akan semakin kecil. Semakin besar nilai Q, maka nilai feromon akan semakin besar.

private static final double rho = 0.01;

private static final double Q = 2.0;

Jumlah titik yang harus dilewati harus ditentukan. Dalam kasus ini, diasumsikan terdapat 8 titik yang perlu dilewati.

private static final int jumlahTitik = 8;

Jumlah semut yang akan melakukan pencarian jalur juga perlu ditentukan. Setiap semut akan melalui semua titik yang telah disebutkan sebelumnya. Dalam kasus ini, diasumsikan terdapat 10 semut yang akan melakukan pencarian jalur.

Const jumlahSemut As Integer = 10

Tentukan jumlah iterasi yang akan digunakan dalam perhitungan. Setiap iterasi akan diulang oleh setiap semut untuk melakukan pencarian jalur. Dalam situasi ini, diasumsikan bahwa jumlah iterasi yang akan dijalankan adalah sebanyak 100 kali.

Const iterasi As Integer = 100

Langkah-langkah penggunaan algoritma ini adalah:

1. Tentukan Array Jarak untuk setiap titik yang tersedia, Sesuai asumsi diatas,jarak untuk tiap titik akan dihitung secara acak dengan angka 1 sampai 8

Dim Jarak(jumlahTitik - 1)() As Integer For i As Integer = 0 To Jarak.Length - 1

Jarak(i) = New Integer(jumlahTitik - 1) {}

Next i

For i As Integer = 0 To jumlahTitik - 1

For j As Integer = i + 1 To jumlahTitik - 1 Dim d As Integer = random.Next(1, 8) Jarak(i)(j) = d

Jarak(j)(i) = d

Next j

Next i

2. Tentukan Array Semut yang digunakan untuk melakukan pencarian jalur Setiap semut pada Array Semut akan memiliki array Jejak untuk mencatat jalur- jalur yang dilewati, Beri nilai jejak acak sebagai jejak awal mula dari setiap semut

Dim Semut(jumlahSemut - 1)() As Integer

For k As Integer = 0 To jumlahSemut - 1

Dim titikAwal As Integer = random.Next(0, jumlahTitik)

Dim Jejak(jumlahTitik - 1) As Integer

For i As Integer = 0 To jumlahTitik - 1 Jejak(i) = i

Next i

For i As Integer = 0 To jumlahTitik - 1

Dim r As Integer = random.Next(i, jumlahTitik)

Dim tmp As Integer = Jejak(r)

Jejak(r) = Jejak(i) Jejak(i) = tmp

Next i

Dim idx As Integer = 0

For i As Integer = 0 To Jejak.Length - 1 If Jejak(i) = titikAwal Then

idx = i

End If

Next i

Dim temp As Integer = Jejak(0) Jejak(0) = Jejak(idx) Jejak(idx) = temp

Semut(k) = Jejak

*Next k*y3.

Cari Jejak Terbaik dan Jarak Terpendek dari setiap jejak awal pada semua semut

3. Penjelasan lebih detail tentang fungsi ini dapat dilihat pada penjelasan skrip dibawah ini

Dim JejakTerbaik() As Integer = CariJejakTerbaik(Semut, Jarak)

Dim JarakTerpendek As Double = cariJarakTerpendek(JejakTerbaik, Jarak)

Gunakan Fungsi ini untuk mencari Jejak Terbaik yang memiliki Jarak Terpendek jejak terbaik adalah jejak yang memiliki jarak terpendek, Perhitungan mengenai jarak terpendek dilakukan oleh fungsi cari Jarak Terpendek

Private Function cariJarakTerpendek(ByVal Jejak() As Integer, ByVal Jarak()() As Integer) As Double

Dim hasil As Double = 0.0

For i As Integer = 0 To Jejak.Length - 2 hasil += Jarak(Jejak(i))(Jejak(i + 1))

Next i

Return hasil

End Function

Gunakan Fungsi ini untuk mencari jarak terpendek pada setiap jejak. Jarak terpendek dihitung dari jumlah Jarak yang ditempuh pada Array Jejak

Private Function CariJejakTerbaik(ByVal Semut()() As Integer, ByVal Jarak()() As Integer) As Integer()

Dim JarakTerpendek As Double = cariJarakTerpendek(Semut(0), Jarak)

Dim idxJarakTerpendek As Integer = 0 For k As Integer = 1 To Semut.Length - 1

Dim totalJarak As Double = cariJarakTerpendek(Semut(k), Jarak)

If totalJarak < JarakTerpendek Then JarakTerpendek = totalJarak idxJarakTerpendek = k

End If

Next k

Dim jumlahTitik As Integer = Semut(0).Length

Dim hasilJejakTerbaik(jumlahTitik - 1) As Integer Semut(idxJarakTerpendek).CopyTo(hasilJejakTerbaik, 0) Return hasilJejakTerbaik

End Function

4. Tentukan Array feromon digunakan untuk perhitungan pencarian titik berikutnya

   Beri nilai awal feromon dengan nilai yang sangat rendah sekali, yaitu 0.01

   Dim Feromon(jumlahTitik - 1)() As DoubleFor i As Integer = 0 To jumlahTitik-1

   Feromon(i) = New Double(jumlahTitik - 1) {}

   Next i

   For i As Integer = 0 To Feromon.Length - 1

   For j As Integer = 0 To Feromon(i).Length - 1

   Feromon(i)(j) = 0.01

   Next j

   Next i

5. Lakukan proses pencarian jalur sebanyak n iterasi (poin 5a - 5c)

   5a. Lakukan proses pencarian jejak baru pada setiap semut penjelasan lebih detailtentang fungsi ini dapat dilihat pada penjelasan skrip dibawah ini

   UpdateSemut(Semut, Feromon, Jarak, alpha, beta)

5. Lakukan perulangan pada setiap semut

   Lakukan pencarian jejak baru ke semua titik-titik lain secara acak

6. Untuk semut dengan index k, yang berada pada titik titikX, hitung probabilitasuntuk berpindah ke semua titik tujuan

   Dim taueta(jumlahTitik - 1) As Double Dim jumlahTaueta As Double = 0.0

   For j As Integer = 0 To taueta.Length – 1

   If j = titikX Then

   taueta(j) = 0.0 'Peluang untuk berpindah ke titik diri sendiri adalah 0

   ElseIf titikTerpakai(j) = True Then

taueta(j) = 0.0 'Peluang untuk berpindah ke titik titikTerpakai adalah 0

Else

*taueta(j) = Math.Pow(feromon(titikX)(j), alpha) * Math.Pow((1.0 / Jarak(titikX)(j)), beta)*

If taueta(j) < 0.0001 Then

taueta(j) = 0.0001

*ElseIf taueta(j) > (Double.MaxValue / (jumlahTitik * 100)) Then*

*taueta(j) = Double.MaxValue / (jumlahTitik * 100)*

End If

End If

jumlahTaueta += taueta(j)

Next j

Dim probabilitas(jumlahTitik - 1) As Double

For j As Integer = 0 To probabilitas.Length – 1

probabilitas(j) = taueta(j) / jumlahTaueta

Next j

5a1b. Untuk semut dengan index k, yang berada pada titik titikX, tentukan titik acak berikutnya

Dim NilaiKumulatif(probabilitas.Length) As Double For j As Integer = 0 To probabilitas.Length - 1

NilaiKumulatif(j + 1) = NilaiKumulatif(j) + probabilitas(j)

Next j

Dim p As Double = random.NextDouble()

For j As Integer = 0 To NilaiKumulatif.Length - 2 If p >= NilaiKumulatif(j) AndAlso p <

NilaiKumulatif(j + 1) Then

titikSelanjutnya = j Exit For

End If

Next j

5b. Lakukan proses perubahan nilai feromon pada masing-masing jarak antar titik Penjelasan lebih detail tentang fungsi ini dapat dilihat pada penjelasan skrip dibawah ini

UpdateFeromon(Feromon, Semut, Jarak, rho, Q)

5b1. Update semua nilai feromon untuk setiap jalur cari jarak terpendek pada jalur tersebut Jika titik-titiknya bersebelah5n dan jaraknya bersebelahan, maka nilai feromon akan semakin besar

For i As Integer = 0 To feromon.Length - 1

For j As Integer = i + 1 To feromon(i).Length - 1 For k As Integer = 0 To Semut.Length - 1

Dim jarakTerpendek As Double = cariJarakTerpendek(Semut(k), Jarak)

*Dim faktorPengecil As Double = (1.0 - rho) * feromon(i)(j)*

Dim faktorPembesar As Double = 0.0 If isBersebelahan(i, j, Semut(k)) =

True Then

faktorPembesar = (Q /

jarakTerpendek)

End If

feromon(i)(j) = faktorPengecil +

faktorPembesar

If feromon(i)(j) < 0.0001 Then feromon(i)(j) = 0.0001

ElseIf feromon(i)(j) > 100000.0 Then feromon(i)(j) = 100000.0

End If

feromon(j)(i) = feromon(i)(j)

Next k

Next j

Next i

5c. Cari Jejak Terbaik dan Jarak Terpendek dari jejak semut yang telah mengalami perubahan, Apabila jarak terpendek ternyata lebih baik (rendah) daripada jarak terpendek terbaik, maka ambil nilai jejak nya sebagai jejak terbaik.

Dim JejakTerbaikSkrg() As Integer = CariJejakTerbaik(Semut, Jarak)

Dim JarakTerpendekSkrg As Double =
cariJarakTerpendek(JejakTerbaikSkrg, Jarak)

If JarakTerpendekSkrg < JarakTerpendek Then JarakTerpendek = JarakTerpendekSkrg JejakTerbaik = JejakTerbaikSkrg

Console.WriteLine("Jarak Terpendek terbaru " & JarakTerpendek.ToString("F1") & " ditemukan pada iterasi " & n)

End If

3.3.5. Evaluasi Sistem

Pengujian ini dilakukan dengan tujuan menilai sistem aplikasi secara menyeluruh melalui beberapa aspek berikut:

1. Persyaratan:

   Pengujian mempertimbangkan aspek kebutuhan yang berkaitan dengan penjadwalan kuliah, termasuk informasi tentang dosen, mata kuliah, semester, ruang, hari, dan waktu.
2. Utilitas:

   Awalnya, tujuan utama dari aplikasi ini adalah memberikan solusi penjadwalan kuliah yang optimal secara otomatis dengan menggunakan metode Ant Colony. Pengujian dilakukan untuk menilai sejauh mana aplikasi ini memenuhi tujuan utilitas tersebut.
3. Kinerja:

   Pengujian juga mencakup evaluasi kinerja sistem. Harapannya adalah agar sistem dapat melakukan penjadwalan otomatis dengan mencari solusi yang cepat. Namun, setelah analisis, ditemukan bahwa waktu yang diperlukan cukup lama karena jumlah data yang sangat besar.
4. Dokumentasi:

   Selama pengujian, aspek dokumentasi juga menjadi perhatian. Hal ini bertujuan agar pengguna aplikasi dapat dengan mudah memahami dan menggunakan sistem ini. Dokumentasi tersedia dalam berbagai format, seperti file xls dan doc, untuk mempermudah penggunaan.