

Penjadwalan Perkuliahan menggunakan Algoritma Particle Swarm Optimization (PSO)

Ayu Ratna Juwita¹, Adi Rizky Pratama², Triono³

^{1,2} Program Studi Teknik Informatika, Universitas Buana Perjuangan

³ Program Studi Teknik Informatika, STMIK Bina Sarana Global

Email: ¹ayurj@ubpkarawang.ac.id, ²adi.rizky@ubpkarawang.ac.id, ³triono@stmikglobal.ac.id

Abstrak— Pengaturan waktu, ruang, kelas dan matakuliah merupakan kegiatan dari penjadwalan matakuliah. Sistem informasi penjadwalan kuliah di Fakultas Teknik dan Ilmu Komputer Universitas Buana Perjuangan Karawang yang dilakukan saat ini masih secara manual. Penjadwalan perkuliahan dimana sejumlah matakuliah harus dijadwalkan kedalam ruang dan slot waktu tertentu. Untuk mempermudah proses pembuatan penjadwalan perkuliahan maka dibuat secara komputerisasi. Metode pengembangan yang digunakan adalah prototype menggunakan algoritma Particle Swarm Optimization (PSO). Data yang digunakan dalam penelitian ini adalah data perkuliahan di Fakultas Teknik dan Ilmu Komputer pada Prodi Teknik Informatika dan Sistem Informasi. Pengujian dilakukan untuk mendapatkan hasil optimal dari algoritma PSO yang dilakukan menggunakan metode black box dan white box testing. Berdasarkan hasil pengujian yang dilakukan pada algoritma Particle Swarm Optimization (PSO) dengan jumlah partikel yang dibangkitkan mempengaruhi optimasi jadwal yang dihasilkan, semakin banyak partikel yang terlibat maka dapat menghasilkan jadwal dengan rata-rata hasilnya belum optimal.

Kata Kunci— *Paricle Swarm Optomization (PSO), Penjadwalan perkuliahan, Prototype.*

Abstract - Time, space, class and course management are activities of scheduling courses. Information system for scheduling lectures at the Faculty of Engineering and Computer Science, Buana Perjuangan University, Karawang which is currently being done manually. Scheduling lectures where a number of courses must be scheduled into a specific space and time slot. To simplify the process of making lecture scheduling, it is computerized. The development method used is a prototype using the Particle Swarm Optimization (PSO) algorithm. The data used in this study is lecturing data in the Faculty of Engineering and Computer Science at the Informatics Engineering Study Program and Information Systems. The test is carried out to get optimal results from the PSO algorithm which is carried out using the black box and white box testing methods. Based on the results of tests conducted on the Particle Swarm Optimization (PSO) algorithm with the number of particles generated which affects the optimization of the resulting schedule, the more particles involved, it can produce a schedule with average results not yet optimal.

Keywords — *Paricle Swarm Optomization (PSO), lecture scheduling, Prototype.*

I. PENDAHULUAN

Kegiatan penjadwalan matakuliah pada dunia pendidikan disadari perlunya teori dalam materi pembelajaran di kelas dan penjadwalan perkuliahan. Karena secara umum penjadwalan adalah aktifitas penugasan yang tentunya berhubungan dengan beberapa kendala. Sejumlah kejadian dapat menjadi permasalahan saat proses penjadwalan, dimana hal tersebut sering terjadi pada sebagian besar lingkungan pendidikan.

Fakultas Teknik dan Ilmu Komputer Universitas Buana Perjuangan pada setiap semester melakukan penjadwalan. Dimana penjadwalan yang dilakukan masih menggunakan excel untuk mengatur jadwal perkuliahan. Dalam penjadwalan setiap dosen tidak dapat mengajar pada waktu yang bersamaan pada ruang kelas dengan 2 matakuliah yang berbeda. Kegiatan perkuliahan dilakukan dari hari senin sampai dengan hari sabtu, dimana setiap harinya terdapat 6 jadwal perkuliahan. Masalah yang timbul dari penjadwalan ini adalah pembagian waktu mengajar, ruangan dan dosen yang sering terjadi penumpukan jadwal perkuliahan. Data yang digunakan adalah jadwal Fakultas Teknik dan Ilmu Komputer pada tahun 2019-2020 semester ganjil.

Tabel 1 : Sumber : Fakultas Teknik dan Ilmu Komputer universitas buana perjuangan karawang. Program studi S1 teknik informatika dan S1 Sistem informasi.

No	Angkatan / Semester	Kelas	Dosen	Jumlah			
				Mata Kuliah	Mata Kuliah Praktikum	Ruang Kelas	Ruang Praktikum
1	2016 / 7	6	3 0	14	4	20	5
2	2017 / 5	7		14	4		
3	2018 / 3	8		15	4		
4	2019 / 1	8		14	1		

Oleh karena itu kombinasi dari jadwal mata kuliah, mahasiswa, dosen dan ruang kelas menjadi sangat beragam dan rumit sehingga tidak mudah untuk mengukur dan mencari jadwal yang baik. Melihat dari permasalahan diatas, sebuah solusi dengan algoritma *Particle Swarm Optimization* (PSO) untuk penjadwalan perkuliahan.

II. TINJAUAN PUSTAKA

A. Penjadwalan

Penjadwalan merupakan alokasi dari sumber daya terhadap waktu untuk menghasilkan sekumpulan pekerjaan.

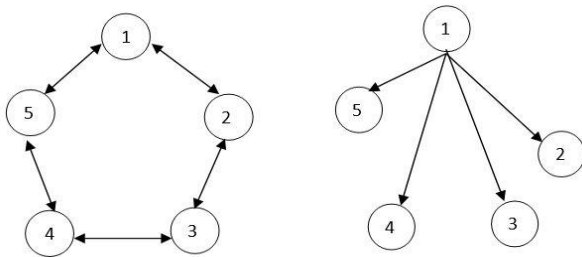
Penjadwalan diperlukan untuk menghasilkan suatu urutan pekerjaan dengan pengalokasian sumber daya yang tepat, seperti mesin yang digunakan, jumlah operator yang bekerja, urutan pengerjaan, dan kebutuhan material. Dengan pengaturan penjadwalan yang efektif dan efisien, perusahaan akan dapat memenuhi order tepat waktu serta kualitas yang telah ditentukan^{[1][2]}.

Beberapa algoritma yang telah digunakan untuk menyelesaikan permasalahan penjadwalan kuliah[3], seperti :

1. Teknik pewarnaan graf : pendekatan yang paling tradisional yang terbukti sangat efisien untuk ukuran masalah penjadwalan yang kecil.
2. Teknik pencarian lokal : dimana perbaikan solusi yang lebih optimum dicari disekitar sebuah solusi yang sudah diketahui sebelumnya. Variasi pencarian lokal yang sering digunakan adalah metode *simulated annealing* dan *tabu search*.
3. Teknik optimasi berdasarkan teori evolusi : yaitu algoritma evolusi, termasuk diantaranya adalah algoritma genetika.

B. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) adalah teknik komputasi berbasis populasi yang dikembangkan oleh James Kennedy dan Russ Eberhart pada tahun 1995. terdapat dua topologi dasar PSO yang digunakan di dalam banyak literature yaitu : *Ring Topology* dan *Star Topology*. Pada *ring topology*, suatu partikel terhubung pada dua partikel lainnya, sehingga memiliki ukuran ketetanggaan (*neighborhood*) sama dengan 3. Sedangkan pada *star topology*, suatu partikel bisa terhubung pada semua partikel lainnya, sehingga disebut *global neighborhood*. Bagaimanapun, ukuran ketetanggaan pada *ring topology* bisa dimodifikasikan^{[4][5]}



Gambar 1. Topologi PSO : *ring topology* (kiri) dan *star topology* (Kanan) (Suyanto, 2010)

Nilai *fitness* terbaik ini dilambangkan dengan *p* dan simpanan *memory*. Sedangkan nilai “terbaik” yang kedua adalah *fitness* terbaik yang dicapai oleh semua partikel dalam topologi ketetanggaan. Indeks *g* digunakan untuk menunjuk partikel dengan *fitness* terbaik tersebut. Jika kita menggunakan topologi ketetanggaan yang berupa *ring topology*, maka secara ini disebut sebagai PSO versi global. Tetapi, jika topologi ketetanggaannya berupa *star topology* maka cara ini disebut PSO versi local.

Setelah menemukan dua nilai “terbaik”, suatu partikel *i* pada posisi X_i , meng-*update* vektor *velocity* dan kemudian meng-*update* posisinya menggunakan persamaan berikut :

$$v_{id} = v_i + \varphi_1 * r * (p_{id} - x_{id}) + \varphi_2 * r * (p_{gd} - x_{id}) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

dimana

i = partikel ke-*i*

d = dimensi ke-*d*

φ_1 = laju belajar (*learning rates*) untuk komponen *cognition* (kecerdasan individu),

φ_2 = adalah laju belajar untuk komponen *social* (hubungan social antar individu).

p = vektor nilai *fitness* terbaik yang dihasilkan sejauh ini.

g = indeks dari partikel dengan *fitness* terbaik didalam topologi ketetanggaan.

r = bilangan acak dalam interval [0,1]

Berdasarkan laju belajarnya, PSO bisa dibedakan ke dalam empat model, yaitu :

1. *Full Model* ($\varphi_1, \varphi_2 > 0$)
2. *Cognition Only* ($\varphi_1 > 0$ dan $\varphi_2 = 0$)
3. *Social Only* ($\varphi_1 = 0$ dan $\varphi_2 > 0$)
4. *Selfless* ($\varphi_1 = 0, \varphi_2 > 0$, dan $g \neq i$).

Velocity partikel pada setiap dimensi dibatasi pada suatu *velocity* maksimum *Vmax*. Nilai batas *Vmax* ditentukan oleh *user*.

C. Black-Box Testing

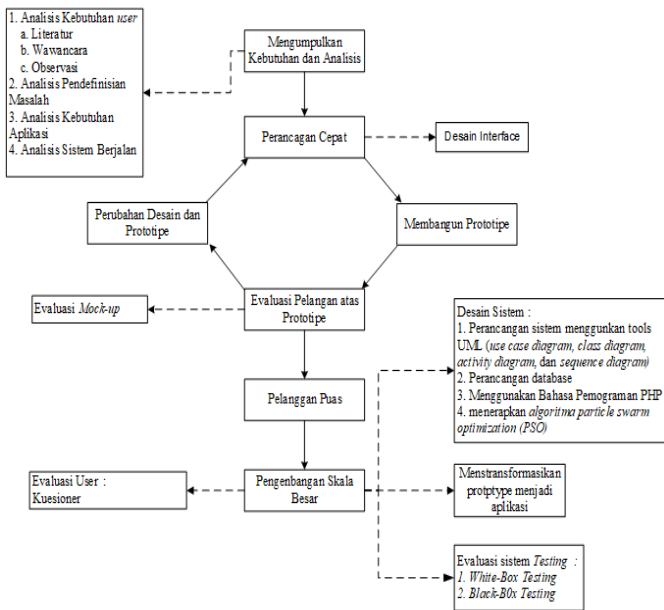
Pengujian *Black-box* berfokus pada persyaratan fungsional yang memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan persyaratan fungsional untuk suatu program^[6]. Roger menegaskan bahwa pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses database eksternal.
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.

Dalam pengujian *black-box* terdapat metode pengujian *Graph-Based*, yaitu pengujian perangkat lunak yang dimulai dengan membuat grafik dari objek-objek yang penting dan hubungan objek-objek serta kemudian memikirkan sederetan pengujian yang akan mencakup grafik tersebut sehingga masing-masing objek dan hubungan digunakan dan kesalahan ditemukan^[7].

III. METODE PENELITIAN

Penelitian ini menggunakan metode rekayasa *Software Development Life Cycle* (SDLC) dengan model *prototype*^[8], karena sesuai dengan penelitian yang akan dilakukan dengan tahap penggunaan sistem yang relatif singkat, ruang lingkup yang kecil serta diharapkan sistem yang dibangun merupakan sistem yang inovatif. Dalam pengumpulan data menggunakan metode literatur, wawancara dan observasi. Sedangkan untuk evaluasi *user* menggunakan kuesioner, hasilnya akan dibuat bentuk narasi, sedangkan analisa pengolahan data penjadwalan menggunakan algoritma *Particle Swarm Optimization* (PSO). Adapun tahapan tahapan yang dilakukan pada penggunaan metode *prototype* dapat direpresentasikan seperti pada gambar berikut :



Gambar 2. Skema Penelitian dengan *Prototype*

Dalam penelitian penjadwalan perkuliahan yang akan dirancang, aplikasi yang akan dikembangkan menggunakan objek penelitian di Fakultas Teknik dan Ilmu Komputer Universitas Buana Perjuangan Karawang.

Pengembangan model perangkat lunak ini menggunakan konsep pengembangan SDLC dengan model *prototype* yang memiliki tahap mengumpulkan dan menganalisis kebutuhan, perancangan cepat, membangun *prototype*, evaluasi pelanggan atas *prototype*, perubahan desain dan *prototype*, pelanggan puas dan pengembangan skala besar. Serta menerapkan algoritma *particle swarm optimization* (PSO) sebagai algoritma dalam penjadwalan perkuliahan.

A. Mengumpulkan dan menganalisis Kebutuhan

Tahap analisis disini merupakan tahap mendengarkan kebutuhan user yang nantinya akan menggunakan aplikasi yang dibuat. Kebutuhan yang dianalisis meliputi:

• **Analisis Kebutuhan User**

Pengumpulan kebutuhan dan analisis ini dilakukan dengan tujuan untuk mengetahui dan mengevaluasi sistem berjalan tentang penjadwalan perkuliahan saat ini, sehingga bisa mendapatkan sistem yang lebih baik. Pengumpulan kebutuhan dan analisis dilakukan dengan analisis yang berorientasi pada objek-objek yang berhubungan dengan aplikasi yang akan dirancang. Setelah dilakukan observasi dan wawancara, masalah yang ada saat ini maka didapat hasil analisis sebagai berikut :

1. Ruang kelas hanya tersedia 20 ruangan kelas dengan 5 ruang praktikum.
2. Pada penggunaannya ruang kelas yang dipaksakan karena daya tampung yang tidak sesuai dengan jumlah mahasiswa khususnya kelas perkuliahan malam.
3. Apabila terjadi perubahan jadwal mata kuliah berpengaruh dengan jadwal lainnya, sehingga terjadi bentrok antar ruang kelas dan jam mata kuliah.

• **Analisa Pendefinisian Masalah**

Berdasarkan hasil observasi langsung pada saat proses pembuatan jadwal perkuliahan dan juga melakukan wawancara kepada kepala program studi teknik Informatika dan Sistem Informasi UBP, dan beberapa pihak, ternyata proses pembuatan jadwal perkuliahan mengalami permasalahan yang sering dihadapi sebagai berikut :

1. Sering terjadi ketidakcocokan pada saat proses sinkronisasi jadwal dosen.
2. Tidak adanya pembaruan data dalam pengelolaan ruang kelas, karena proses pendataan persemester.
3. Banyaknya penggunaan ruang kelas yang tidak sesuai dengan jumlah kapasitas mahasiswa.

• **Analisis Kebutuhan Aplikasi**

Analisis kebutuhan Aplikasi dilakukan untuk mengetahui kebutuhan yang dihadapi oleh pengguna aplikasi penjadwalan perkuliahan dalam penelitian ini :

1. **Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional dilakukan untuk mengetahui kebutuhan apa saja yang dibutuhkan oleh pengguna (*user*) untuk aplikasi penjadwalan perkuliahan. Dari hasil wawancara sesuai dengan analisis kebutuhan *user*, maka didapat beberapa kebutuhan pengguna untuk aplikasi penjadwalan perkuliahan sebagai berikut :

- a. Mempermudah dalam menentukan jadwal perkuliahan dengan menggunakan aplikasi yang berbasis web.
- b. mempermudah proses pengolahan data jadwal ruangan kelas, mata kuliah, dan dosen
- c. Aplikasi mampu mengklasifikasikan ruangan mana saja yang siap digunakan dengan kapasitas yang cukup untuk tiap kelas.

2. **Analisis Pengguna Sistem**

Analisis pengguna sistem dilakukan untuk mencari pengguna yang tepat terhadap aplikasi yang akan dibuat. Hasil analisis yang didapat dari klasifikasi pengguna sistem aplikasi ini hanya dapat digunakan oleh satu *user* yang akan mengelola data penjadwalan perkuliahan.

IV. HASIL DAN PEMBAHASAN

Dalam penelitian penjadwalan perkuliahan yang dibuat menggunakan objek penelitian di Fakultas Teknik dan Ilmu Komputer Universitas Buana Perjuangan Karawang. Metode rekayasa perangkat lunak dengan konsep pengembangan SDLC yang digunakan adalah model *prototype* dengan tahapan mengumpulkan dan menganalisis kebutuhan, perancangan cepat, membangun *prototype*, evaluasi pelanggan atas *prototype*, perubahan desain dan *prototype*, pelanggan puas dan pengembangan skala besar.

A. Mengumpulkan dan menganalisis Kebutuhan

Tahap analisis disini merupakan tahap mendengarkan kebutuhan user yang nantinya akan menggunakan aplikasi yang dibuat. Kebutuhan yang dianalisis meliputi:

- **Analisis Kebutuhan User**

Pengumpulan kebutuhan dan analisis ini dilakukan dengan tujuan untuk mengetahui dan mengevaluasi sistem berjalan tentang penjadwalan perkuliahan saat ini, sehingga bisa mendapatkan sistem yang lebih baik. Pengumpulan kebutuhan dan analisis dilakukan dengan analisis yang berorientasi pada objek-objek yang berhubungan dengan aplikasi yang akan dirancang. Setelah dilakukan observasi dan wawancara, masalah yang ada saat ini maka didapat hasil analisis sebagai berikut :

1. Ruang kelas hanya tersedia 20 ruangan dengan jumlah 2 prodi.
2. Pada penggunaannya ruang kelas yang dipaksakan karena daya tampung yang tidak sesuai dengan jumlah mahasiswa khususnya kelas perkuliahan malam.
3. Apabila terjadi perubahan jadwal mata kuliah berpengaruh dengan jadwal lainnya, sehingga terjadi bentrok antar ruang kelas dan jam mata kuliah.

- **Analisa Pendefinisian Masalah**

Berdasarkan hasil observasi langsung pada saat proses pembuatan jadwal perkuliahan dan juga melakukan wawancara kepada kepala program studi Informatika dan Sistem Informasi UBP, dan beberapa pihak, ternyata proses pembuatan jadwal perkuliahan mengalami permasalahan yang sering dihadapi sebagai berikut :

1. Sering terjadi ketidakcocokan pada saat proses sinkronisasi jadwal dosen..
2. Tidak adanya pembaruan data dalam pengelolaan ruang kelas, karena proses pendataan persemester.
3. Banyaknya penggunaan ruang kelas yang tidak sesuai dengan jumlah kapasitas mahasiswa.

- **Analisis Kebutuhan Aplikasi**

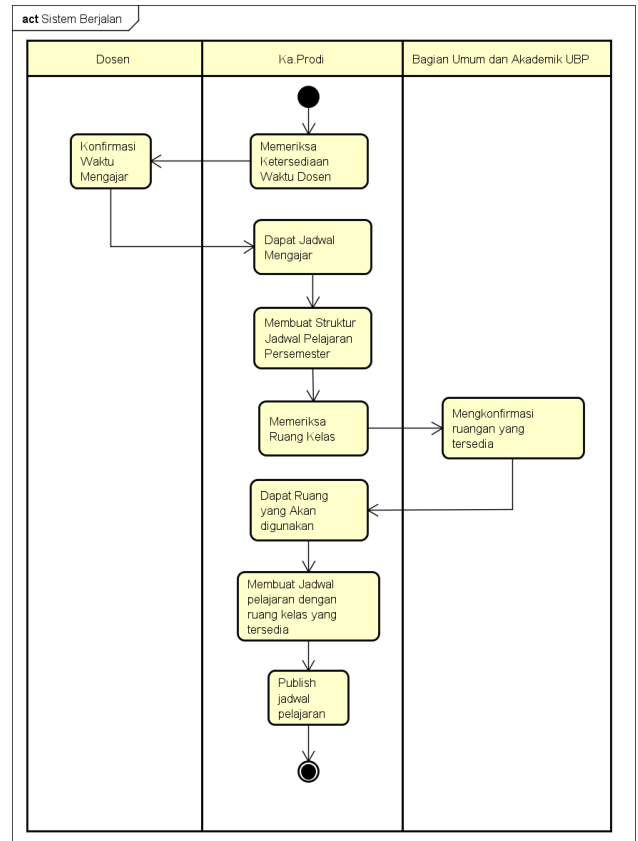
Analisis kebutuhan Aplikasi dilakukan untuk mengetahui kebutuhan yang dihadapi oleh pengguna aplikasi penjadwalan perkuliahan dalam penelitian ini :

1. **Analisis Kebutuhan Fungsional**
 Analisis kebutuhan fungsional dilakukan untuk mengetahui kebutuhan apa saja yang dibutuhkan oleh pengguna (*user*) untuk aplikasi penjadwalan perkuliahan. Dari hasil wawancara sesuai dengan analisis kebutuhan *user*, maka didapat beberapa kebutuhan pengguna untuk aplikasi penjadwalan perkuliahan sebagai berikut :
 - a. Mempermudah dalam menentukan jadwal perkuliahan dengan menggunakan aplikasi yang berbasis web.
 - b. mempermudah proses pengolahan data jadwal ruangan kelas, mata kuliah, dan dosen
 - c. aplikasi mampu mengklasifikasikan ruangan mana saja yang siap digunakan dengan kapasitas yang cukup untuk tiap kelas.

2. Analisis Pengguna Sistem

Analisis pengguna sistem dilakukan untuk mencari pengguna yang tepat terhadap aplikasi yang akan dibuat. Hasil analisis yang didapat dari klasifikasi pengguna sistem aplikasi ini hanya dapat digunakan oleh satu *user* yang akan mengelola data penjadwalan perkuliahan.

- **Analisis Sistem Berjalan**



Gambar 3. Activity Diagram Berjalan Penjadwalan Kuliah

B. Perancangan Sistem

1. Use Case Diagram

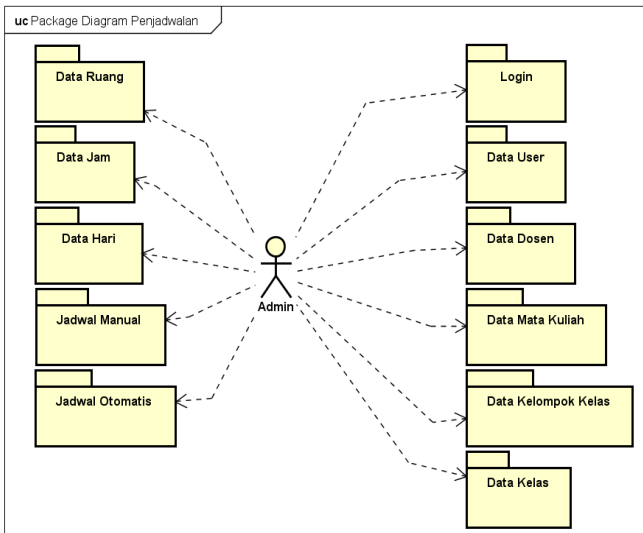
Use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada 2 hal utama pada *use case* yaitu:

- **Aktor**

Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.

- **Use Case Diagram**

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut ini adalah diagram *use case* dari aplikasi penjadwalan perkuliahan.



Gambar 4. Package diagram Sistem Penjadwalan

Berikut adalah penjelasan dari *Package diagram* dari gambar 3 yaitu:

- Definisi Aktor
Berikut ini adalah deskripsi pendefinisian aktor pada aplikasi penjadwalan perkuliahan menggunakan Algoritma *Particle Swarm Optimization (PSO)*.

Tabel 2. Definisi Aktor

No	Aktor	Deskripsi
1	Ka. Prodi	Orang yang bertugas untuk mengelola data <i>user</i> , mengelola data dosen, mengelola data kelas, mengelola data ruang, mengelola data mata kuliah, mengelola data hari, mengelola data jam, mengelola data jadwal manual, dan mengelola data penjadwalan.

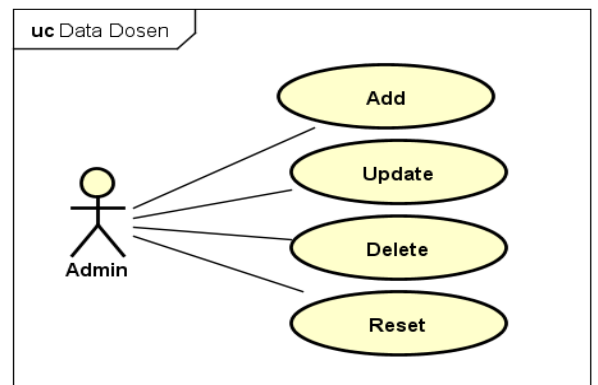
- Definisi *Package Diagram*
Berikut ini adalah deskripsi pendefinisian *Package Diagram* pada aplikasi penjadwalan perkuliahan menggunakan Algoritma *Particle Swarm Optimization (PSO)*.

Tabel 3. Definisi *Package Diagram*

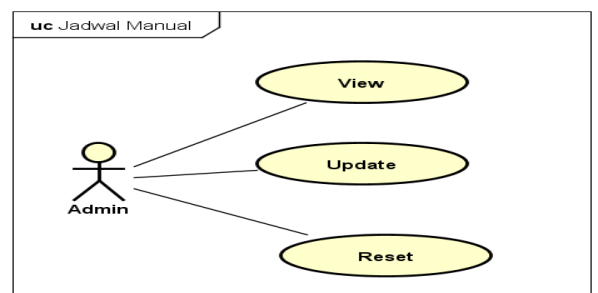
No	Use Case	Deskripsi
1	<i>Login</i>	Merupakan proses untuk mengecek hak akses siapa yang berhak mengakses proses dalam memasukan data penjadwalan. <i>Login</i> wajib untuk fungsi-fungsi yang berkaitan dengan akses pengubah ke basis data.
2	Mengelola Data <i>User</i>	Merupakan proses untuk mengelola data <i>user</i> sehingga dapat mengetahui langsung data nama-nama <i>user</i> yang bisa menggunakan aplikasi ini.

3	Mengelola Data Dosen	Merupakan proses untuk mengelola data dosen yang akan menjadi pengajar saat perkuliahan.
4	Mengelola Data Kelas	Merupakan proses untuk mengelola data kelas yang akan mengikuti perkuliahan.
5	Mengelola Data Ruang	Merupakan proses untuk mengelola data ruangan yang akan menjadi tempat perkuliahan
6	Mengelola Data Mata Kuliah	Merupakan proses untuk mengelola data matakuliah yang akan menjadi bahan ajar.
7	Mengelola Data Hari	Merupakan proses untuk mengelola data hari yang tersedia dalam seminggu.
8	Mengelola Data Jam	Merupakan proses untuk mengelola data jam perkuliahan yang tersedia dalam sehari.
9	Mengelola Data Jadwal Manual	Merupakan proses untuk mengelola data jadwal perkuliahan sehingga dosen bisa memesan jadwal terlebih dahulu sebelum jadwal perkuliahan dibuat.
11	Mengelola Data Jadwal Otomatis	Merupakan proses untuk mengelola data penjadwalan dengan menggunakan Algoritma <i>Particle Swarm optimization (PSO)</i> .

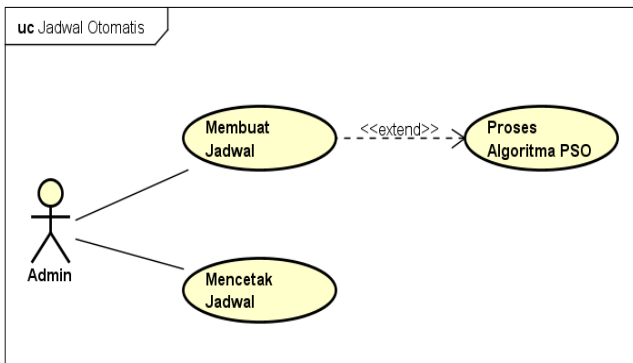
- Use Case Diagram



Gambar 5. Use case Diagram Dosen untuk Penjadwalan Kuliah



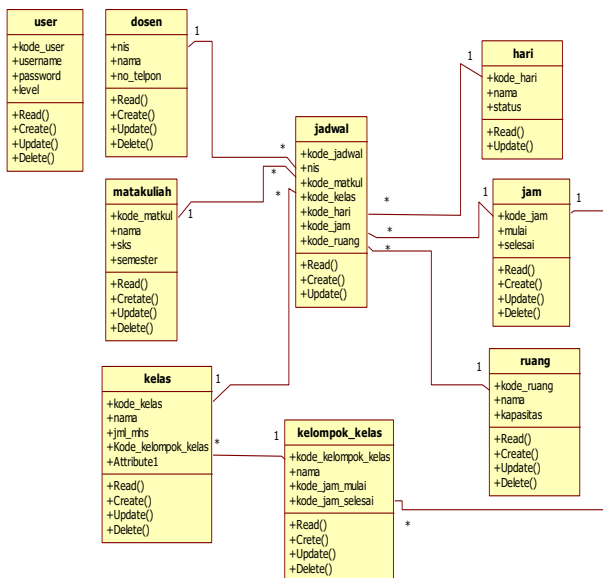
Gambar 6. Use case Diagram Jadwal Manual untuk Penjadwalan Kuliah



Gambar 7. Use case Diagram Jadwal Otomatis untuk Penjadwalan Kuliah

C. Perancangan Database Sistem

Perancangan database dilakukan setelah perancangan sistem, perancangan database yang dibuat terdiri dari class diagram dan struktur data. Database dalam aplikasi ini diimplementasikan menggunakan MySQL. Perancangan database ini meliputi pembentukan class diagram yang berikutnya akan diimplementasikan dalam bentuk tabel-tabel dengan keterkaitan atau keterhubungan diantara tabel tersebut. Berikut ini adalah class diagram untuk menggambarkan database pada aplikasi yang telah dibuat sebagai berikut :



Gambar 8. Class Diagram

Pada struktur database ini, semua jenis data yang terlihat dalam proses yang terjadi pada gambar 8.

D. Mentranformasikan Prototype Menjadi Aplikasi

Setelah melakukan beberapa tahap atas prototipe dan user atas desain mock-up interface, desain sistem, dan perancangan database yang telah dibuat maka akan

menstrasformasikan prototype menjadi sebuah aplikasi sebagai berikut:

- Implementasi desain program

Pada tahapan ini dilakukan proses menerjemahkan desain ke dalam implementasi kode program komputer. Pembuatan kode program tersebut menggunakan bahasa pemrograman php dengan menggunakan Notepad++ kemudian untuk databasenya yaitu dengan MySQL. Implementasi desain program didapat dari desain interface atau tampilan dari aplikasi yang telah dibuat, berikut ini beberapa hasil implementasi dari desain interface atau desain antarmuka yang telah dibuat:



Gambar 9. Menu login

Dimana user harus memasukan username dan password dan klik tombol login maka akan masuk ke halaman utama penjadwalan perkuliahan.



Gambar 10. Menu Penjadwalan dengan PSO

Merupakan menu penjadwalan untuk membuat dan mencetak hasil penjadwalan perkuliahan menggunakan algoritma Particle Swarm Optimizaton (PSO).

- Implemtasi Algoritma Particle Swarm Optimizaton (PSO)

Secara singkat, pseudo code algoritma PSO dapat dituliskan seperti pada gambar 2.3. Pada saat inisialisasi, nilai-nilai pada vektor velocity dibangkitkan secara acak dalam interval [-Vmax, Vmax], di mana Vmax adalah nilai maksimum yang dapat diberikan v_{id} .

```

for setiap partikel
  Inisialisasi partikel
end
repeat
  for setiap partikel
    Hitunglah nilai fitness
    if nilai fitness baru lebih baik dari pada
    nilai
      fitness lama
    end
  end
  pilih partikel dengan nilai fitness terbaik
  diantara semua
  partikel tetangganta dan simpan nilai
  fitness terbaik
  tersebut
  for setiap partikel
    Hitung velocity partikel menggunakan
    rumus pertama
    diatas
    update posisi partikel menggunakan
    rumus dua di atas
  end
until (KriteriaBerhenti = true)
    
```

Untuk menggambarkan hubungan suatu partikel dengan partikel yang lainnya dengan menggunakan topologi *star* karena suatu partikel bisa terhubung pada semua partikel pada semua partikel lainnya sehingga disebut dengan global *neighborhood* dan dapat digambarkan dengan topologi *star* untuk proses kinerja algoritma *particle swarm optimization* dalam implementasi untuk *update velocity*. Untuk penentuan laju belajar menggunakan *full model* ($\phi_1, \phi_2 > 0$), dengan nilai *range* $w = 0.4 - 1.4$, $c_1 = 1.5 - 2.0$ dan $c_2 = 2.0 - 2.5$

Contoh perhitungan Algoritma :

Nilai X=posisi V=kecepatan $w=0,4$ $c_1=1,5$ $c_2=2$
 $X_{max} = 7$
 $X_{min} = 0$

x = posisi

Ruang	hari	Jam
R1	senin	08.00 - 10.00
R2	selasa	10.00 – 12.00

Posisi

X	Posisi
0	R1, SENIN, 08.:10
1	R1, SENIN 10:12
2	R1, SELASA, 08:10
3	R1, SELASA, 10:12
4	R2, SENIN, 08:10
5	R2, SENIN, 10:12
6	R2, SELASA, 08:10
7	R2, SELASA, 10:12

Tabel partikel

Kelas	Matkul	Dosen
IF18A	ADBO	Ayu Ratna
IF18B	ADBO	Ayu Ratna

Menentukan partikel

Partikel 1: ADBO, Ayu Ratna, IF18A = $1 + \text{RANDBETWEEN}(0,7-1)$ persamaan 2.3
 Partikel 2: ADBO, Ayu Ratna, IF18B = $1 + \text{RANDBETWEEN}(0,7-1)$ persamaan 2.4

Mencari kecepatan misalnya :

V=kecepatan	
3	ADBO, AYU RATNA, IF18A, R2, SELASA, 08.00:10.00
2	ADBO, AYU RATNA, IF18B, R1, SELASA, 10.00:12.00
2	ADBO, AYU RATNA, IF18A, R2, SENIN, 08.00:10.00
1	ADBO, AYU RATNA, IF18B, R1, , 10.00:12.00
1	ADBO, AYU RATNA, IF18A, R2, SELASA, 08.00:10.00
0	ADBO, AYU RATNA, IF18B, R1, SELASA, 10.00:12.00

Perhitungan menggunakan rumus persamaan 1
 $v = 0.4 * 6 * 1.5 * \text{RANDBETWEEN}(0,1) * (3-6) * 2 * \text{RANDBETWEEN}(0,1) * (3-6) = 0$

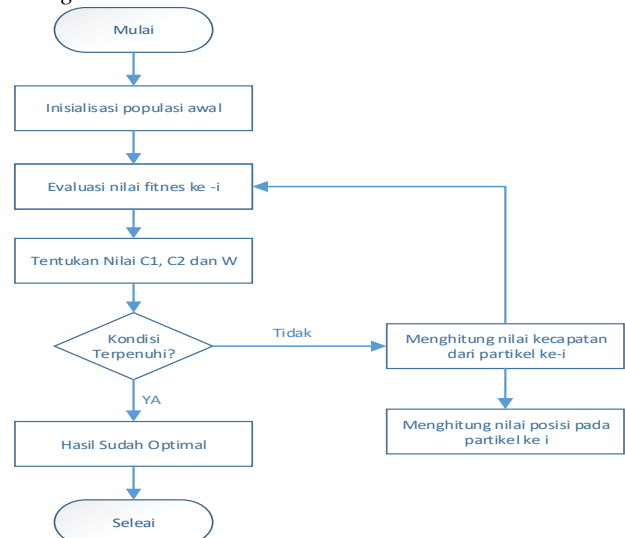
Menggunakan rumus persamaan 2

$$X = 7 + 0 = 7$$

$$X = \text{MOD}((7),7)$$

$$X = 7$$

- Implementasi Desain Program Hasil Generate Algoritma PSO



Gambar 11. Alur Sistem Algoritma PSO

No.	Jadwal	Fitness	X	V
0	31	0	162	500
1	33	0	456	398
2	46	0	105	332
3	48	0	239	105
4	76	0	63	97
5	78	0	190	61
6	271	0	189	180
7	273	0	84	77
8	324	0	42	136
9	325	0	176	179
10	339	0	140	325
11	340	0	57	172
12	361	0	444	205
13	362	0	227	154
14	363	0	94	320
15	364	0	150	406
16	366	0	206	100
17	367	0	143	384
18	396	0	78	23
19	397	0	238	-9
20	398	0	163	-48

Gambar 12. Hasil Generate PSO dengan data 16 Jadwal

No.	Jadwal	Fitness	X	V
0	31	0	162	144
1	33	0	456	103
2	46	0	105	482
3	48	0	239	249
4	76	0	63	468
5	78	0	189	401
6	271	0	49	149
7	273	0	84	307
8	324	0	42	409
9	325	0	176	421
10	339	0	140	289
11	340	0	57	268
12	361	0	444	327
13	362	0	227	39
14	363	0	233	139
15	364	0	150	154
16	366	0	206	155
17	367	0	143	492
18	396	0	78	266
19	397	0	238	393
20	398	0	163	366
21	399	0	235	78
22	400	0	340	377
23	401	0	489	198
24	402	0	298	311
25	403	0	130	70
26	404	0	368	490
27	405	0	67	230
28	406	0	453	102
29	407	0	312	252
30	408	0	81	133
31	409	0	103	27
32	410	0	159	190
33	411	0	228	474
34	412	0	292	210
35	413	1	188	0
36	414	0	360	139
37	415	0	58	386
38	416	0	226	1
39	417	0	129	35
40	418	0	374	2
41	419	0	492	25

Gambar 12. Hasil Generate PSO dengan data 35 Kelas

- Hasil Perbandingan jadwal sebelum dan setelah menggunakan PSO.

Tabel 4. Hasil Perbandingan Jadwal

Angkatan Prodi	Total Jadwal	Total Jadwal Sebelum PSO	Total Jadwal dengan PSO
IF 16	40	26	30
IF 17	35	30	32
IF 18	48	38	40
IF 19	42	38	40
SI 16	20	14	14
SI 17	14	10	14
SI 18	16	14	16
SI 19	14	11	14

Dari hasil tabel diatas pengaturan jadwal menggunakan PSO lebih baik dibandingkan tidak menggunakan PSO.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilaksanakan maka terdapat beberapa kesimpulan sebagai berikut :

1. Merancang suatu aplikasi penjadwalan perkuliahan di Fakultas Teknik dan Ilmu Komputer UBP, menggunakan algoritma *Particle Swarm Optimization* (PSO) dilakukan dengan menggunakan metode *prototype* dan UML sebagai perancangan sistem.
2. Aplikasi penjadwalan diimplementasikan dengan algoritma *Particle Swarm Optimization*, untuk masalah penjadwalan perkuliahan.
3. Beberapa percobaan dapat menghasilkan nilai fitness minimum yang diharapkan, yaitu 0 pada iterasi yang cukup besar, namun pada beberapa percobaan tidak dapat menghasilkan nilai fitness 0 pada iterasi maksimal yang ditentukan dikarenakan masih ada beberapa jadwal belum dapat dioptimasi.

B. Saran

Berdasarkan kesimpulan di atas, maka ada beberapa saran sebagai referensi untuk penelitian selanjutnya sebagai berikut :

1. Aplikasi ini menggunakan algoritma *Particle Swarm Optimization* (PSO) dalam proses pembuatan penjadwalan perkuliahan, diharapkan pada penelitian selanjutnya bisa menggunakan algoritma yang lain yang mempunyai kemampuan lebih baik dari pada algoritma *Particle Swarm Optimization* (PSO).
2. Nilai konstrain dapat ditambah dan disesuaikan dengan kebutuhan dan persyaratan pembuatan jadwal yang berlaku.
3. Aplikasi ini tidak berhubungan dengan mata kuliah praktikum, sehingga diharapkan bisa mengatur jadwal perkuliahan dengan jadwal praktikum.
4. Aplikasi ini tidak bisa membuat jadwal yang terlalu banyak, sehingga diharapkan pada penelitian

selanjutnya bisa melakukan pembuat jadwal perkuliahan dengan jumlah kelas yang sesuai.

DAFTAR PUSTAKA

- [1] J. Sistem, D. A. N. Informatika, I. M. B. Adnyana, and I. K. Wijayana, "Rancang Bangun Sistem Penjadwalan Sidang Skripsi Menggunakan Algoritma Genetika," pp. 38–47.
- [2] I. Permana, F. N. Salisah, P. Studi, S. Informasi, and B. K. Literatur, "Sistem Penjadwalan Otomatis Tempat Khut Bah Jum 'at," vol. 3, no. 1, pp. 59–64, 2017.
- [3] D. Haryadi and A. Jamal, "Preferensi Dosen Pada Proses Penjadwalan Kuliah Menggunakan Algoritma Genetik Studi Kasus: Universitas Al Azhar Indonesia," *J. Al-Azhar Indones. Seri Sains Dan Teknol.*, vol. 3, no. 2, p. 91, 2017.
- [4] Suyanto, *Algoritma Optimasi Deterministik atau Probabilitik*. Yogyakarta: Graha Ilmu, 2010.
- [5] D. W. Nugraha, A. Y. E. Dodu, and M. F. B. Paloloang, "Sistem Penjadwalan Perkuliahan Menggunakan Algoritma Particle Swarm Optimization Pada Universitas Tadulako Lecture Scheduling System Using Particle Swarm Optimization Algorithm At Department Of Information Technology Faculty Of Engineering Tadulako Univer," vol. 2, no. 1, pp. 1–11, 2019.
- [6] Rosa A S and M. Shalahuddin, *Rekayasa Perangkat Lunak terstruktur dan Berorientasi Objek Edisi Revisi*. Bandung: Informatika Bandung, 2018.
- [7] P. Roger S, *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 – Buku 1*. Yogyakarta: Andi Offset, 2012.
- [8] Jenner Simarmata, *Rekayasa Perangkat Lunak*. Yogyakarta: Andi, 2010.