

## OPTIMASI PENJADWALAN PRAKTIKUM MENGGUNAKAN *MODIFIED REAL CODE PARTICLE SWARM OPTIMIZATION* (STUDI KASUS FAKULTAS IMU KOMPUTER UNIVERSITAS BRAWIJAYA)

Brigitta Ayu Kusuma Wardhany<sup>1</sup>, Istiana Rachmi<sup>2</sup>, Nur Firra Hasjidla<sup>3</sup>, Zulianur Khaqiqiyah<sup>4</sup>, Idham Triatmaja<sup>5</sup>, Imam Cholissodin<sup>6</sup>

<sup>1,2,3,4,5,6</sup>Fakultas Ilmu Komputer Universitas Brawijaya

Email: <sup>1</sup>gittawardhany4@gmail.com, <sup>2</sup>istianarahmi@gmail.com, <sup>3</sup>firrapirraa@gmail.com, <sup>4</sup>zulianurhaqq@gmail.com, <sup>5</sup>mahdi3atmaja@gmail.com, <sup>6</sup>imamcs@ub.ac.id

(Naskah masuk: 1 Oktober 2016, diterima untuk diterbitkan: 26 Desember 2016)

### Abstrak

Penjadwalan adalah salah satu proses dalam manajemen waktu yang di atur sedemikian rupa agar kegiatan dapat berjalan dengan lancar. Banyak algoritma yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan. Pada kasus ini penulis menggunakan algoritma *Modified Real Code PSO* (M-RCPSO). Data yang digunakan terdiri dari data dosen, asisten, mahasiswa, ruangan dan waktu praktikum. Dari hasil pengujian *popsize*, pengujian iterasi, pengujian parameter kognitif dan sosial, dan pengujian parameter terbaik, didapatkan nilai rata-rata *fitness* adalah 1. Hal ini menunjukkan bahwa solusi yang didapatkan sudah mendekati optimum.

**Kata kunci:** *modified real code particle swarm optimization, penjadwalan*

### Abstract

*Scheduling is one of the time management process that well regulated so that the activities can run fluently. Many algorithms can be used to solve scheduling problems. In this case, the author uses a Modified Real Code PSO (M-RCPSO) algorithm. The data used consisted of lecturer, assistant, student, room and lab time. From the results of popsize testing, iterative testing, cognitive parameter testing, and the best parameters testing obtained the average fitness value is 1. This matter shows that the solution obtained is already approaching optimal.*

**Keywords:** *modified real code particle swarm optimization, scheduling*

## 1. PENDAHULUAN

Penjadwalan adalah suatu proses yang berkaitan dengan pengaturan waktu. Pengaturan waktu sendiri memiliki arti dimana suatu kegiatan diatur sedemikian mungkin agar dapat berlangsung secara lancar dan tidak terjadi bentrok dengan kegiatan lainnya. Untuk itu diperlukan susunan jadwal kegiatan agar dapat terkoordinir dengan baik. Namun kebanyakan orang masih mengabaikan penyusunan jadwal ini dikarenakan prosesnya yang sangat membingungkan dan memakan waktu lama. Di Indonesia sendiri salah satu permasalahan tentang penjadwalan yang paling sering dihadapi adalah penjadwalan matakuliah yang didalamnya masih dibagi lagi menjadi penjadwalan praktikum dan penjadwalan ujian. Pengaturan waktu terhadap suatu kegiatan merupakan hal yang penting dilakukan agar kegiatan tersebut berlangsung secara lancar. Hal ini sangat diperlukan demi terlaksananya proses belajar mengajar yang efektif bagi sebuah jurusan di universitas tersebut (Puspaningrum dkk., 2013). Dalam penyusunan jadwal kuliah ini tentu diperlukan komponen-komponen yang mendukung agar tercipta

suatu penjadwalan yang baik. Komponen-komponen tersebut bisa juga dikatakan sebagai sumber daya yang harus dipenuhi, antara lain waktu, tempat, orang, dll (Suhartono, 2015). Jika komponen tersebut telah dipenuhi maka penjadwalan tersebut bisa dikatakan baik, karena penjadwalan tersebut bisa diikuti oleh seluruh pihak yang terlibat dalam kegiatan belajar mengajar, baik dosen maupun mahasiswa.

Masalah penjadwalan dalam universitas merupakan salah satu masalah kompleks yang bisa dikaitkan dengan masalah optimasi. Hal ini dikarenakan perlunya suatu perhitungan yang cukup rumit dengan data yang banyak agar didapatkan solusi yang mendekati optimum. Namun kenyataannya proses pencarian solusi pada permasalahan ini memakan waktu komputasi yang lama. Terlebih lagi harus memenuhi tiap komponen yang ada dan komponen tiap universitas hampir berbeda. Komponen tersebut harus saling mendukung satu sama lain agar tercipta suatu penjadwalan yang baik, sebagai contoh kebutuhan mahasiswa dalam perkuliahan sebisa mungkin jangan sampai terkendala hanya karena tidak bisa mengambil

matakuliah wajib dikarenakan jadwal yang bentrok dengan matakuliah lain. Selain itu dari segi dosen, jangan sampai terjadi bentrok dengan matakuliah lain yang diajarkan (Puspaningrum dkk., 2013). Sehingga dibutuhkan suatu batasan dalam permasalahan penjadwalan agar jelas dan mudah untuk menyusunnya. Batasan tersebut terdiri dari *hard constraint*, yaitu batasan yang harus dipenuhi dan *soft constraint* yaitu batasan yang tidak harus dipenuhi namun tetap dijadikan acuan dalam penyusunannya (Mawaddah, 2006). Salah satu permasalahan penjadwalan yang kompleks adalah penjadwalan praktikum, dimana dalam permasalahan ini memiliki komponen yang harus dipenuhi yaitu waktu dosen pengampu, waktu asisten, waktu mahasiswa, jumlah ruang praktikum atau laboratorium (lab), dan masih banyak lagi.

Persoalan optimasi adalah persoalan yang menuntut pencarian solusi optimum (Marwana, 2012). Ada dua metode dalam penyelesaian masalah optimasi, yaitu (1) Metode Konvensional dan (2) Metode Heuristik. Dimana salah satu algoritma dari metode heuristik yang sering digunakan dalam permasalahan optimasi adalah *Particle Swarm Optimization* (PSO). Algoritma PSO berfokus pada penyelesaian masalah optimasi dalam pencarian ruang untuk mendapatkan solusi. Algoritma PSO dapat menyelesaikan masalah penjadwalan praktikum dengan meminimalkan kesenjangan waktu dan memaksimalkan pemanfaatan sumber daya dan *constraint* dalam penggunaan ruangan yang efektif (Mansur dkk., 2014). Pada penelitian sebelumnya yang berjudul *Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi* oleh Mansur dkk., menyatakan bahwa analisa data *resource* dengan memperhatikan *constraint* mampu menghasilkan solusi yang optimal dalam penggunaan ruangan. Selain itu penelitian lain berjudul *Optimasi Penjadwalan Matakuliah Di Jurusan Teknik Informatika PENS Dengan Menggunakan Algoritma Particle Swarm Optimization* (PSO) oleh Dian Ariani dkk. berpendapat bahwa Algoritma PSO dapat digunakan untuk mengoptimasi masalah penjadwalan matakuliah di Jurusan Teknik Informatika PENS. Lalu penelitian dengan judul *Penyelesaian Penjadwalan Kuliah Menggunakan Algoritma Particle Swarm Optimization dan Hybrid Dimension Association Rule* oleh Ling Ria Sukmana Putri dkk., menyatakan bahwa Algoritma PSO dapat diimplementasikan pada permasalahan ini namun masih belum dapat mengatasi pelanggaran *soft constraint* hingga 100%, pernyataan lain bahwa hasil terbaik dalam penggunaan PSO ini dipengaruhi oleh parameter kognitif dan sosial ( $C_1$  dan  $C_2$ ), dimana nilai  $C_1$  harus lebih besar dari nilai  $C_2$ .

Berdasarkan penjelasan sebelumnya di atas, pada penelitian kali ini akan dibahas mengenai penjadwalan praktikum matakuliah Kecerdasan Buatan dengan menggunakan *Modified Real Code*

*Particle Swarm Optimization* pada Fakultas Ilmu Komputer Universitas Brawijaya. Permasalahan yang akan diselesaikan adalah pengelolaan komponen jadwal yang terdiri dari data dosen, data asisten, data mahasiswa, data ruangan, dan waktu praktikum. Lalu akan diselesaikan juga permasalahan dalam pembuatan jadwal praktikum untuk matakuliah Kecerdasan Buatan yang ditujukan bagi mahasiswa. Penjadwalan ini tidak boleh bentrok dengan jadwal asisten dan juga mahasiswa, dimana ruangan dan dosen juga dijadikan panutan dalam penyusunannya. Sehingga diharapkan dengan adanya Optimasi Penjadwalan Praktikum Kecerdasan Buatan dengan M-RCPSO dapat mengatasi permasalahan kompleks dan menghasilkan solusi yang mendekati optimum.

### 1.1 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Optimasi hanya dilakukan untuk penjadwalan praktikum pada matakuliah Kecerdasan Buatan.
2. Metode yang digunakan untuk proses optimasi penjadwalan praktikum adalah *Modified Real Code Particle Swarm Optimization*.
3. Bagian yang di-*Modified* adalah faktor konvergensi ( $\lambda$ ).
4. Dataset yang digunakan terdiri dari data dosen, data asisten, data ruang/kelas, dan data waktu praktikum Kecerdasan Buatan di Fakultas Ilmu Komputer Universitas Brawijaya.

## 2. DASAR TEORI

### 2.1 Penjelasan Dataset

Data yang digunakan dalam penelitian adalah data asisten praktikum dari Laboratorium Komputasi Cerdas (KC) Fakultas Ilmu Komputer Universitas Brawijaya. Dataset yang didapat adalah 13 asisten seperti pada Tabel 1 dimana nantinya masing – masing kelas akan diampu oleh 2 asisten, 7 dosen seperti pada Tabel 2 dimana dosen pertama mengampu 3 kelas, dosen kedua mengampu 2 kelas dan lainnya masing-masing mengampu 1 kelas, 2 ruangan dan 10 kelas yang harus di ajar dimana setiap ruangan hanya dapat digunakan oleh 5 kelas seperti pada Tabel 3. Selain itu tiap ruangan hanya bisa digunakan dalam 5 sesi seperti pada Tabel 4.

Tabel 1. Data Asisten Praktikum

1	M. Syafiq
2	M. Kadafi
3	Sabrina Nurfadilla
4	Diva Kurnianingtyas
5	Nanda Putri
6	Radita Noer Pratiwi
7	Daneswara
8	Andriansyah Yusuf R.
9	Karina
10	Anandhi Tristiaratri
11	Agung Nurjaya Megantara

12	Ardiansyah S.
13	Fathor Rosi

Tabel 2. Data Dosen Pengampu

Dosen Pengampu	Kelas
Satrio Hadi Wijoyo	C
	I
Lailil Muflikhah	D
	E
	F
Ika Kusumaning Putri	J
Mochammad Hannats Hanafi I.	G
M. Ali Fauzi	H
Imam Cholissodin	B
Nurizal Dwi Priandani	A

Tabel 3. Data Kelas Praktikum

	Kelas				
R1	A	B	C	D	E
R2	F	G	H	I	J

Tabel 4. Data Sesi Praktikum

Sesi	Jam
1	07.00 – 08.40
2	08.40 – 10.20
3	10.20 – 12.00
4	12.50 – 14.30
5	14.30 – 16.10

## 2.2 Penjadwalan Praktikum Matakuliah

Penjadwalan dapat didefinisikan sebagai suatu pekerjaan untuk mendistribusikan sejumlah aktivitas yang melibatkan dosen, matakuliah, dan seterusnya, ke dalam satuan waktu (bisa termasuk ruangan) yang terbatas dengan batasan-batasan tertentu. Salah satu jenis penjadwalan adalah penjadwalan praktikum.

Praktikum berasal dari kata *practicus* (Latin) yang secara harfiah memiliki arti “Aktif” atau *pratein* (Yunani) yang memiliki arti “mengerjakan”. Praktikum merupakan salah satu subsystem dari perkuliahan. Kegiatan praktikum memiliki struktur dan jadwal yang terencana. Tujuan praktikum yaitu untuk memberikan kesempatan kepada mahasiswa dalam memperoleh pengalaman nyata dan membuat mahasiswa menguasai suatu keterampilan. beberapa matakuliah tidak hanya dapat dipahami namun dapat lebih mudah diimplementasikan apabila disertai praktikum.

Pada perguruan tinggi terdapat beberapa matakuliah yang memiliki jadwal untuk kegiatan praktikum. Pada kegiatan praktikum beberapa dosen pengampu cenderung dibantu oleh asisten praktikum. Asisten praktikum merupakan mahasiswa yang

tugasnya membimbing atau mendampingi praktikan dalam melakukan kegiatan praktikum. Asisten praktikum diberi tugas oleh dosen pengampu untuk memberikan materi praktik atau implementasi langsung mengenai teori yang telah dijelaskan sebelumnya oleh dosen pengampu.

Fakultas Ilmu Komputer di Universitas Brawijaya merupakan salah satu fakultas yang beberapa matakuliahnya memberikan kegiatan praktikum kepada mahasiswa. Salah satu matakuliah yang memiliki jadwal kegiatan praktikum yaitu Kecerdasan Buatan. Kecerdasan Buatan merupakan salah satu matakuliah yang membutuhkan lebih dari hanya sekedar penyampaian materi dari seorang dosen. Matakuliah tersebut membutuhkan jadwal kegiatan praktikum untuk membuat mahasiswa lebih paham mengenai teori yang disampaikan.

## 2.3 Algoritma Modified Real Code Particle Swarm Optimization

PSO adalah algoritma yang terinspirasi dari perilaku social alam, pergerakan dinamis, dan komunikasi pada serangga, burung, dan ikan. Pada penelitian yang dilakukan oleh Omar S. Soliman dkk. (2014) tentang klasifikasi Diabetes Mellitus menggunakan Modified RCPSO dan LS-SVM. Penerapan M-RCPSO pada penelitian ini digunakan untuk mencari nilai parameter optimal untuk digunakan pada algoritma LS-SVM yaitu nilai  $\gamma$  dan  $\sigma$ . Perbedaan algoritma M-RCPSO dengan PSO pada umumnya terletak pada adanya penambahan rumus untuk menghitung faktor konvergensi. Penelitian tersebut menjelaskan bahwa hasil akurasi yang didapatkan sangat tinggi jika dibandingkan dengan penelitian lain yang mengangkat permasalahan yang sama namun berbeda metode yaitu 97,833% dengan menggunakan 768 data pasien. Pada penelitian ini Modified RCPSO digunakan untuk mencari solusi optimal untuk penjadwalan ruang praktikum Kecerdasan Buatan yang harus memenuhi batasan – batasan yang telah ditentukan.

### 2.3.1 Inisialisasi

Pada proses inisialisasi ini dilakukan inisialisasi terhadap jumlah populasi, posisi awal partikel ( $X(t)$ ), dan kecepatan awal partikel ( $V(t)$ ). Pada penelitian ini, digunakan 5 data jadwal praktikum Kecerdasan Buatan dengan inisialisasi nilai random. Dimana representasi partikelnya diperlukan 2 asisten dalam 1 kelas, yang terdiri dari 10 kelas. Ukuran partikel adalah 10 kelas \* 2 asisten. Setelah itu hasil inisialisasi posisi awal partikel akan dilakukan pembulatan dari bilangan *real* menjadi integer.

### 2.3.2 Proses Repair

Pada penelitian ini proses repair dilakukan dengan cara mengecek asprak yang belum dapat jadwal/kelas (*repair* dilakukan setiap inisialisasi awal dan *update* posisi) dan akan menggantikan asprak

yang mengajar lebih dari 2 kelas. Hal ini dilakukan agar representasi posisi partikel dapat menghasilkan solusi yang mendekati optimum. Proses repair sebenarnya adalah proses tambahan pada M-RCPSO sehingga prosesnya bisa dilakukan ataupun tidak. Namun untuk kasus penjadwalan praktikum ini proses repair sangat diperlukan, hal ini karena jika tidak dilakukan repair maka susah untuk mendapatkan *fitness* yang bernilai 1 sebab representasi partikelnya akan menjadi terlalu liar dalam arti susah untuk mencapai solusi yang optimum. Proses *repair* ini juga mencegah tidak terkontrolnya proses random partikel dalam representasi partikel tersebut. Hal ini akan menyulitkan proses pengujian, karena nilai *fitness*-nya masih ada kemungkinan lebih tinggi dari sebelumnya. Oleh karena itu, dengan adanya proses *repair* ini representasi partikel yang berupa bilangan real kemudian dibulatkan menjadi integer dan merepresentasikan *id* dari asisten akan menjadi tidak terlalu liar, dalam arti mudah untuk memperoleh nilai *fitness* 1 atau solusi yang optimum. Sehingga dalam proses pengujian akan terlihat parameter-parameter terbaiknya dalam mencapai solusi optimum (konvergen).

### 2.3.3 Evaluasi Nilai *Fitness* Tiap Partikel ( $f(X_i)$ )

Seperti pada algoritma evolusi, fungsi objektif mengukur seberapa dekat solusi dengan optimum, contohnya fungsi objektif mengukur performansi atau kualitas partikel. Pada penelitian ini nilai *fitness* tiap partikel dihitung melalui pengecekan *constraint*. Pencarian jadwal yang optimal juga dilakukan dengan memperhatikan batasan – batasan atau *constraints* (*hard & soft*) yang telah ditentukan. *Hard Constraint* merupakan batasan yang harus dipenuhi sedangkan *Soft Constraint* merupakan batasan yang tidak harus dipenuhi namun tetap dijadikan acuan dalam penyusunannya. Berikut dijelaskan secara detail batasan tersebut.

- *Hard Constraint*
  - a. Asisten hanya boleh mengajar maksimal 2 kelas (Berlebih).
  - b. Asisten tidak boleh mengampu kelas pada dosen yang sama (Bentrok Dosen).
  - c. Dalam 1 kelas tidak boleh diajar oleh 1 asisten (Berbeda).
  - d. Semua asisten harus mendapat jadwal praktikum (AssKosong).
- *Soft Constraint*
  - a. Pasangan asisten harus berbeda tiap kelasnya (Bentrok).

Berdasarkan penjelasan *constraint* sebelumnya pada kasus ini nilai *fitness* diambil dari banyaknya batasan yang dilanggar sehingga semakin kecil jumlah pelanggaran yang dihasilkan maka solusi yang dihasilkan akan semakin baik. Untuk tiap

pelanggaran yang terjadi akan diberikan nilai 1. Agar tidak terjadi nilai *fitness* yang tak terhingga maka jumlah total semua pelanggaran akan ditambahkan 1 yang ditunjukkan pada Persamaan 1.

$$f = \frac{1}{1 + (\sum \text{Berlebih} + \sum \text{BentrokDosen} + \sum \text{Berbeda} + \sum \text{AssKosong})} \quad (1)$$

### 2.3.4 Update *Pbest*

Update *Pbest* dengan mencari nilai *fitness* maksimum tiap partikel dan bandingkan dengan nilai *fitness* maksimum sebelumnya. Jika  $f(X_i) > f(Pbest_k)$ , maka  $f(Pbest_k) = X_i$ . Lalu set nilai  $P_i$  sama dengan dengan lokasi dari nilai *fitness* maksimum  $X_i$ . Dimana  $P_i$  adalah hasil *update* dari tiap *Pbest* (Partikel *Best*).

### 2.3.5 Update *Gbest*

Update *Gbest* (*Global Best*) dengan mengevaluasi keseluruhan *fitness Pbest* dan mengambil yang paling maksimum.

### 2.3.6 Update Kecepatan Partikel

Update kecepatan partikel dengan rumus yang ditunjukkan pada Persamaan 2 dan iterasi,  $t = t + 1$ .

$$V_{id} = \lambda[\omega_{id}V_{id} + C_1r_1(p_{id} - X_{id}) + C_2r_2(p_{gd} - X_{id})] \quad (2)$$

Dimana,

$i$  = indeks partikel ke- $i$

$d$  = indeks dimensi ke- $d$

$\lambda$  = faktor konvergensi

$\omega_{id}$  = bobot inersia partikel ke- $i$  dimensi ke- $d$

$C_1$  = parameter kognitif

$C_2$  = parameter sosial

$r_1$  dan  $r_2$  = angka random dengan range [0,1]

$P_{id}$  = *Pbest* partikel ke- $i$  dimensi ke- $d$

$P_{gd}$  = *Gbest* partikel ke- $g$  dimensi ke- $d$

$X_{id}$  = Posisi partikel ke- $i$  dimensi ke- $d$

Nilai faktor konvergensi ( $\lambda$ ) dihitung dengan menggunakan Persamaan 3.

$$\lambda = \frac{2}{\sqrt{2 - C - \sqrt{C^2 - 4C}}} \quad (3)$$

Dimana, nilai  $C$  adalah hasil penjumlahan dari  $C_1$  dan  $C_2$ . Selanjutnya nilai bobot inersia ( $\omega$ ) dapat dihitung dengan rumus yang ditunjukkan pada Persamaan 4.

$$\omega_{id} = 0.9 - \frac{t}{T_{max}} * 0.5 \quad (4)$$

Dimana,

Setelah itu lakukan perbaikan terhadap hasil *update* kecepatan partikel yang baru sesuai dengan kondisi berikut.

$$v_i = \begin{cases} V_{max} & \text{if } v_i > V_{max} \\ -V_{max} & \text{if } v_i < -V_{max} \end{cases}$$

### 2.3.7 Update Posisi Partikel

Melakukan *update* posisi partikel baru untuk digunakan pada iterasi selanjutnya hingga memenuhi kondisi berhenti yaitu hingga iterasi maksimum terpenuhi. Rumus *update* posisi ditunjukkan pada Persamaan 5.

$$X_{id} = X_{id} + (\omega V_{id}) \quad (5)$$

### 3. PERANCANGAN DAN IMPLEMENTASI

Pada tahap ini akan dibahas mengenai langkah-langkah dan rancangan yang digunakan dalam pembuatan sistem “Optimasi Penjadwalan Praktikum Kecerdasan Buatan Menggunakan *Modified Real Code Particle Swarm Optimization* (Studi Kasus Fakultas Ilmu Komputer Universitas Brawijaya)”, rancangan implementasi *pseudocode* M-RCPSO sebagai berikut:

- Melakukan studi literatur mengenai algoritma *Modified Real Code Particle Swarm Optimization*.
- Menyusun solusi permasalahan dengan algoritma M-RCPSO.
- Menganalisis dan merancang sistem dengan menggunakan hasil pembelajaran pada tahap sebelumnya.
- Implementasi sistem berdasarkan analisis dan perancangan yang dilakukan.
- Melakukan uji coba dan evaluasi.

#### 3.1 Implementasi *Pseudocode* M-RCPSO

Berdasarkan perancangan perangkat lunak yang telah dirancang sebelumnya, maka akan diuraikan mengenai implementasi *pseudocode* algoritma M-RCPSO sesuai dengan perancangan yang telah dibuat yang ditunjukkan pada Gambar 1. Sistem diimplementasikan menggunakan bahasa pemrograman PHP.

```

1 begin
2   t = 0
3   inialisasi posisi partikel( $X_{i,j}^t$ ),
4   kecepatan( $V_{i,j}^t$ ),  $Pbest_{i,j}^t = X_{i,j}^t$ , hitung
5   fitness tiap partikel, dan  $Gbest_{g,j}^t$ 
6   do
7     t = t + 1
8     hitung faktor konvergensi ( $\lambda$ )
9     update kecepatan  $v_{i,j}(t)$ 
10    update posisi  $x_{i,j}(t)$ 
11    hitung fitness tiap partikel
12    update  $Pbest_{i,j}(t)$  dan  $Gbest_{g,j}(t)$ 
13  while (bukan kondisi berhenti)
14  end
15

```

Gambar 1. *Pseudocode* Algoritma M-RCPSO

Penjelasan dari Gambar 1:

1. Baris 2 merupakan inialisasi iterasi awal
2. Baris 3-5 merupakan proses inialisasi posisi awal partikel, kecepatan awal partikel,  $Pbest$  awal partikel, evaluasi  $fitness$  tiap partikel, dan  $Gbest$  awal partikel
3. Baris 6 melakukan perulangan
4. Baris 7 inialisasi iterasi selanjutnya
5. Baris 8-12 merupakan proses perhitungan yang ada pada algoritma M-RCPSO
6. Baris 13 bernilai benar akan melanjutkan perulangan dan bernilai salah akan keluar dari perulangan

### 3.2 Implementasi Antarmuka

Implementasi antarmuka berupa tampilan halaman hasil proses M-RCPSO dengan jumlah populasi partikel 5, maksimal iterasi 2, dan nilai  $C_1$  dan  $C_2$  berturut-turut adalah 1. Berikut hasil implementasinya yang ditunjukkan pada Gambar 2 yang merupakan hasil iterasi 0, Gambar 3 yang merupakan hasil iterasi 1, dan Gambar 4 yang merupakan hasil dari keseluruhan iterasi.

```

ITERASI [0]
Kecepatan X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20
V1(0) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
V2(0) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
V3(0) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
V4(0) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
V5(0) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Partikel[1]: 1.56 || 3.51 || 12.09 || 10.14 || 5.98 || 11.7 || 10.53 || 2.34 || 8.45 || 1.17 || 5.72 || 9.62 || 1.69 || 2.08 || 9.36
|| 11.7 || 6.11 || 11.83 || 8.06 || 10.79 ||
Hasil pembulatan => Partikel[1]: 2 | 4 | 12 | 10 | 6 | 12 | 11 | 2 | 8 | 1 | 6 | 10 | 2 | 2 | 9 | 12 | 6 | 12 | 8 | 11 |
Hasil repair => Partikel[1]: 7 | 4 | 3 | 10 | 3 | 5 | 11 | 5 | 8 | 1 | 6 | 10 | 2 | 2 | 9 | 12 | 6 | 12 | 8 | 11 |
C1) Asisten yg belum dapat jadwal = 1
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 2
C4) Berbeda = 1
Fitness P1 = 0.2

Partikel[2]: 11.83 || 1.43 || 0.65 || 8.45 || 1.95 || 6.89 || 0.78 || 5.59 || 3.9 || 8.19 || 5.98 || 7.67 || 2.08 || 2.73 || 2.86 ||
0.13 || 11.05 || 0.13 || 3.12 || 2.6 ||
Hasil pembulatan => Partikel[2]: 12 | 1 | 1 | 8 | 2 | 7 | 1 | 6 | 4 | 8 | 6 | 8 | 2 | 3 | 3 | 1 | 11 | 1 | 3 | 3 |
Hasil repair => Partikel[2]: 12 | 10 | 9 | 5 | 2 | 7 | 10 | 6 | 4 | 8 | 6 | 8 | 2 | 5 | 9 | 1 | 11 | 1 | 3 | 3 |
C1) Asisten yg belum dapat jadwal = 1
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 1
C4) Berbeda = 1
Fitness P2 = 0.25

Partikel[3]: 1.04 || 8.06 || 8.84 || 3.12 || 12.35 || 10.4 || 2.08 || 4.55 || 3.64 || 3.25 || 9.36 || 6.5 || 3.64 || 7.8 || 6.37 ||
1.56 || 4.42 || 7.28 || 0.26 || 4.81 ||
Hasil pembulatan => Partikel[3]: 1 | 8 | 9 | 3 | 12 | 10 | 2 | 5 | 4 | 3 | 9 | 7 | 4 | 8 | 6 | 2 | 4 | 7 | 1 | 5 |
Hasil repair => Partikel[3]: 1 | 8 | 9 | 3 | 12 | 10 | 2 | 5 | 11 | 3 | 9 | 7 | 4 | 8 | 6 | 2 | 4 | 7 | 1 | 5 |
C1) Asisten yg belum dapat jadwal = 1
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 0
C4) Berbeda = 0
Fitness P3 = 0.5

Partikel[4]: 13 || 10.01 || 9.23 || 3.64 || 10.79 || 12.09 || 1.82 || 1.56 || 1.17 || 11.57 || 5.2 || 2.73 || 1.82 || 0.65 || 7.28 ||
6.76 || 6.76 || 4.42 || 2.99 || 6.24 ||
Hasil pembulatan => Partikel[4]: 13 | 10 | 9 | 4 | 11 | 12 | 8 | 2 | 2 | 1 | 12 | 5 | 3 | 2 | 1 | 7 | 7 | 4 | 3 | 6 |
Hasil repair => Partikel[4]: 13 | 10 | 9 | 4 | 11 | 12 | 8 | 2 | 1 | 12 | 5 | 3 | 2 | 1 | 8 | 7 | 7 | 4 | 3 | 6 |
C1) Asisten yg belum dapat jadwal = 0
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 0
C4) Berbeda = 0
Fitness P4 = 1

Partikel[5]: 10.66 || 4.81 || 7.28 || 0.78 || 3.64 || 11.44 || 6.11 || 6.11 || 4.81 || 4.94 || 10.53 || 12.87 || 6.89 || 7.8 || 9.1 ||
9.75 || 3.25 || 5.85 || 12.87 || 0.26 ||
Hasil pembulatan => Partikel[5]: 11 | 5 | 7 | 1 | 4 | 11 | 6 | 6 | 5 | 5 | 11 | 13 | 7 | 8 | 9 | 10 | 3 | 6 | 13 | 1 |
Hasil repair => Partikel[5]: 12 | 2 | 7 | 1 | 4 | 11 | 2 | 6 | 5 | 5 | 11 | 13 | 7 | 8 | 9 | 10 | 3 | 6 | 13 | 1 |
C1) Asisten yg belum dapat jadwal = 0
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 1
C4) Berbeda = 1
Fitness P5 = 0.3333333333333333

Posisi X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
X1(0) 7 4 3 10 3 5 11 5 8 1 6 10 2 2 9 12 6 12 8 11 0.2
X2(0) 12 10 9 5 2 7 10 6 4 8 6 8 2 5 9 1 11 1 3 3 0.25
X3(0) 1 8 9 3 12 10 2 5 11 3 9 7 4 8 6 2 4 7 1 5 0.5
X4(0) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1
X5(0) 12 2 7 1 4 11 2 6 5 5 11 13 7 8 9 10 3 6 13 1 0.3333333333333333

Pbest X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
Pbest1(0) 7 4 3 10 3 5 11 5 8 1 6 10 2 2 9 12 6 12 8 11 0.2
Pbest2(0) 12 10 9 5 2 7 10 6 4 8 6 8 2 5 9 1 11 1 3 3 0.25
Pbest3(0) 1 8 9 3 12 10 2 5 11 3 9 7 4 8 6 2 4 7 1 5 0.5
Pbest4(0) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1
Pbest5(0) 12 2 7 1 4 11 2 6 5 5 11 13 7 8 9 10 3 6 13 1 0.3333333333333333

Gbest(0) X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
P(4) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1

```

Gambar 2. Antarmuka Hasil Proses M-RCPSO Iterasi 0

```

ITERASI [1]
Kecepatan X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20
V1(1) -1.2 3.6 2.8 -0.4 0.4 2.4 0.4 1.2 -2.4 0 2.4 -2 0.4 0 -3.2 -1.6 0.4 -2 -1.6 -3.2
V2(1) -3.2 1.2 0.4 1.6 0.8 1.6 0.8 0.8 -0.8 -2.8 2.4 -1.2 0.4 -1.2 -3.2 2.8 -1.6 2.4 0.4 0
V3(1) 1.2 2 0.4 2.4 -3.2 0.4 4 1.2 -3.6 -0.8 1.2 -0.8 -0.4 -2.4 -2 2.4 1.2 0 1.2 -0.8
V4(1) -3.6 1.2 0.4 2 -2.8 -0.4 1.6 2.4 0.4 -4.4 2.8 0.8 0.4 0.4 -2.8 0.4 0 1.2 0.4 -1.2
V5(1) -3.2 4.4 1.2 3.2 0 0 4 0.8 -1.2 -1.6 0.4 -3.2 -1.6 -2.4 -3.2 -0.8 1.6 0.4 -3.6 0.8

Hasil repair => Partikel[1]: 9 | 4 | 5 | 4 | 3 | 7 | 11 | 9 | 12 | 1 | 5 | 8 | 2 | 2 | 13 | 10 | 6 | 10 | 6 | 8 |
C1) Asisten yg belum dapat jadwal = 0
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 2
C4) Berbeda = 1
Fitness P1 = 0.25

Hasil repair => Partikel[2]: 12 | 11 | 10 | 1 | 1 | 9 | 11 | 7 | 10 | 5 | 8 | 7 | 2 | 4 | 6 | 4 | 9 | 12 | 3 | 3 |
C1) Asisten yg belum dapat jadwal = 1
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 2
C4) Berbeda = 1
Fitness P2 = 0.2

Hasil repair => Partikel[3]: 8 | 8 | 9 | 5 | 9 | 10 | 1 | 3 | 7 | 2 | 10 | 6 | 1 | 6 | 3 | 4 | 5 | 7 | 2 | 4 |
C1) Asisten yg belum dapat jadwal = 3
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 2
C4) Berbeda = 1
Fitness P3 = 0.14285714285714

Hasil repair => Partikel[4]: 9 | 11 | 9 | 6 | 13 | 12 | 10 | 4 | 1 | 8 | 8 | 4 | 2 | 1 | 13 | 7 | 7 | 5 | 3 | 5 |
C1) Asisten yg belum dapat jadwal = 0
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 1
C4) Berbeda = 0
Fitness P4 = 0.5

Hasil repair => Partikel[5]: 12 | 11 | 8 | 1 | 4 | 11 | 12 | 7 | 4 | 3 | 11 | 10 | 5 | 13 | 6 | 9 | 5 | 6 | 9 | 2 |
C1) Asisten yg belum dapat jadwal = 0
C2) Asisten yg kelebihan jadwal = 0
C3) Bntrok Dosen = 1
C4) Berbeda = 0
Fitness P5 = 0.5

Posisi X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
X1(1) 9 4 5 4 3 7 11 9 12 1 5 8 2 2 13 10 6 10 6 8 0.25
X2(1) 12 11 10 1 1 9 11 7 10 5 8 7 2 4 6 4 9 12 3 3 0.2
X3(1) 8 8 9 5 9 10 1 3 7 2 10 6 1 6 3 4 5 7 2 4 0.14285714285714
X4(1) 9 11 9 6 13 12 10 4 1 8 8 4 2 1 13 7 7 5 3 5 0.5
X5(1) 12 1 8 1 4 11 12 7 4 3 11 10 5 13 6 9 5 6 9 2 0.5

Pbest X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
Pbest1(1) 9 4 5 4 3 7 11 9 12 1 5 8 2 2 13 10 6 10 6 8 0.25
Pbest2(1) 12 10 9 5 2 7 10 6 4 8 6 8 2 5 9 1 11 1 3 3 0.25
Pbest3(1) 1 8 9 3 12 10 2 5 11 3 9 7 4 8 6 2 4 7 1 5 0.5
Pbest4(1) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1
Pbest5(1) 12 1 8 1 4 11 12 7 4 3 11 10 5 13 6 9 5 6 9 2 0.5

Gbest(1) X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
P(4) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1
    
```

Gambar 3. Antarmuka Hasil Proses M-RCPSTO Iterasi 1

```

-----Gbest dari seluruh iterasi-----
Gbest X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 Fitness
Gbest(0) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1
Gbest(1) 13 10 9 4 11 12 8 2 1 12 5 3 2 1 8 7 7 4 3 6 1
    
```

Gambar 4. Hasil Gbest Keseluruhan Iterasi

## 4. PENGUJIAN DAN ANALISIS

### 4.1 Pengujian Ukuran Populasi (*PopSize*)

Pada pengujian ukuran populasi (*PopSize*) dilakukan untuk mengetahui ukuran populasi yang dapat menghasilkan solusi paling optimal pada kasus penjadwalan praktikum matakuliah Kecerdasan Buatan Fakultas Ilmu Komputer. Pengujian populasi ini akan dilakukan sebanyak 5 kali dengan jumlah iterasi sebanyak 5. Ukuran populasi yang akan diuji merupakan nilai kelipatan 5 sampai dengan 50 dan kombinasi  $C_1$  dan  $C_2$  adalah 1. Setelah diketahui nilai *fitness*nya lalu dilakukan rata-rata terhadap nilai *fitness* tersebut, dari situ dapat diketahui nilai rata-rata *fitness* terbaik berdasarkan pengujian populasi ini adalah 1 dan didapatkan mulai dari pengujian dengan banyak populasi 10 sampai dengan 50, dimana semakin besar nilai *fitness* (mendekati 1) maka semakin mendekati solusi optimum. Berikut hasil

dari pengujian populasi (*PopSize*) yang ditunjukkan pada Tabel 5.

Tabel 5. Hasil Pengujian Ukuran Populasi (*PopSize*)

Banyak <i>Popsize</i>	Nilai <i>Fitness</i> Percobaan ke-					Rata-rata nilai <i>Fitness</i>
	1	2	3	4	5	
5	0,5	1	1	1	1	0,9
10	1	1	1	1	1	1
15	1	1	1	1	1	1
20	1	1	1	1	1	1
25	1	1	1	1	1	1
30	1	1	1	1	1	1
35	1	1	1	1	1	1
40	1	1	1	1	1	1
45	1	1	1	1	1	1
50	1	1	1	1	1	1

### 4.2 Pengujian Banyaknya Iterasi

Uji coba banyak iterasi pada Tabel 6 akan dilakukan 5 kali dengan menggunakan kombinasi parameter kognitif dan sosial  $C_1$  dan  $C_2$  dan banyaknya *popsize* yang digunakan adalah *popsize* terbaik dari pengujian ukuran populasi yang sebelumnya telah dilakukan yaitu 10. Banyaknya iterasi yang akan diuji merupakan nilai kelipatan 10 sampai dengan 100. Setelah diketahui nilai *fitness*nya lalu dilakukan rata-rata terhadap nilai *fitness* tersebut, dari hal itu dapat diketahui nilai rata-rata *fitness* terbaik berdasarkan pengujian iterasi ini adalah 1 dan didapatkan mulai dari pengujian dengan banyak iterasi 10.

Tabel 6. Hasil Pengujian Banyaknya Iterasi

Banyak Iterasi	Nilai <i>Fitness</i> Percobaan ke-					Rata-rata nilai <i>Fitness</i>
	1	2	3	4	5	
10	1	1	1	1	1	1
20	1	1	1	1	1	1
30	1	1	1	1	1	1
40	1	1	1	1	1	1
50	1	1	1	1	1	1
60	1	1	1	1	1	1
70	1	1	1	1	1	1
80	1	1	1	1	1	1
90	1	1	1	1	1	1
100	1	1	1	1	1	1

**4.3 Pengujian Kombinasi Parameter Kognitif dan Sosial (C<sub>1</sub> dan C<sub>2</sub>)**

Pengujian kombinasi C<sub>1</sub> dan C<sub>2</sub> yang dilakukan bertujuan untuk mendapatkan kombinasi C<sub>1</sub> dan C<sub>2</sub> yang dapat menghasilkan nilai *fitness* paling besar. Uji coba dilakukan sebanyak 5 kali dengan kenaikan kombinasi probabilitasnya sebanyak 0.2 dalam rentang 2.5-0,5 untuk C<sub>1</sub> dan rentang 0,5-2,5 untuk C<sub>2</sub>. Dimana C<sub>1</sub> dan C<sub>2</sub> dicari dengan menggunakan rumus *time variant* yang ditunjukkan pada Persamaan 6 dan Persamaan 7. Pengujian ini dilakukan menggunakan jumlah *popsize* terbaik hasil pengujian *popsize* yaitu 10 dan jumlah iterasi hasil pengujian iterasi sebelumnya yaitu 10. Hasil pengujian kombinasi C<sub>1</sub> dan C<sub>2</sub> terbaik terdapat pada kombinasi C<sub>1</sub> mulai dari 2.3 sampai 0.5 dan C<sub>2</sub> mulai dari 0.7 sampai 2.5 pada permasalahan penjadwalan asisten praktikum matakuliah Kecerdasan Buatan. Berikut hasil dari pengujian kombinasi C<sub>1</sub> dan C<sub>2</sub> yang ditunjukkan pada Tabel 7.

$$C_1 = (C_{1f} - C_{1i}) \frac{t}{t_{max}} + C_{1i} \tag{6}$$

$$C_2 = (C_{2f} - C_{2i}) \frac{t}{t_{max}} + C_{2i} \tag{7}$$

Tabel 7. Hasil Pengujian Parameter Kognitif dan Sosial (C<sub>1</sub> dan C<sub>2</sub>)

Kombinasi		Nilai <i>fitness</i> percobaan generasi ke-					Rata-rata <i>fitness</i>
C <sub>1</sub>	C <sub>2</sub>	1	2	3	4	5	
2.3	0.7	1	1	1	1	1	1
2.1	0.9	1	1	1	1	1	1
1.9	1.1	1	1	1	1	1	1
1.7	1.3	1	1	1	1	1	1
1.5	1.5	1	1	1	1	1	1
1.3	1.7	1	1	1	1	1	1
1.1	1.9	1	1	1	1	1	1
0.9	2.1	1	1	1	1	1	1
0.7	2.3	1	1	1	1	1	1
0.5	2.5	1	1	1	1	1	1

**4.4 Pengujian Parameter Terbaik**

Uji coba ini menggunakan parameter terbaik yang telah diperoleh dari pengujian sebelumnya yaitu jumlah *popsize* sebanyak 10, jumlah iterasi sebanyak 10 hal ini dikarenakan pada iterasi pertama *fitness* yang didapatkan telah optimum hingga iterasi maksimum tercapai, nilai C<sub>1</sub> = 2.3 dan C<sub>2</sub> = 0.7 untuk mendapatkan hasil penjadwalan praktikum matakuliah Kecerdasan Buatan seperti pada Tabel 8.

Tabel 8. Hasil Pengujian Parameter Terbaik

R1	Lailil		Lailil		Lailil		Satrio		Satrio	
	Ardi	Agung	Anan	Fathor	Syafiq	Sabrina	Kadafi	Danes	Nanda	Syafiq
R2	Ika			Hannats		Ali		Imam		Nurizal
	Radita	Karina	Sabrina	Danes	Radita	Andri	Diva	Kadafi	Andri	Anan

**4.5 Analisis Hasil Pengujian**

Berdasarkan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem telah memberikan solusi yang optimal dengan besar *fitness* 1 dimana *fitness* 1 menandakan bahwa solusi yang dihasilkan telah memenuhi syarat yang telah ditentukan, namun dalam hal ini sistem mencapai global optimal dimana solusi optimal didapatkan secara cepat pada iterasi – iterasi awal. Hal ini disebabkan oleh masih kecilnya ruang lingkup pada permasalahan penjadwalan ini dan juga karena adanya proses *repair* yang diterapkan pada sistem.

**5. KESIMPULAN DAN SARAN**

**5.1 Kesimpulan**

Berdasarkan hasil uji coba parameter algoritma Optimasi Penjadwalan Praktikum Kecerdasan Buatan Menggunakan *Modified Real Code Particle Swarm Optimization* dengan Studi Kasus Fakultas Ilmu Komputer Universitas Brawijaya, terdapat beberapa kesimpulan yaitu:

1. Untuk mengukur solusi dari permasalahan optimasi M-RCPSO ini dilakukan proses *repair*, dimana cek asisten yang jadwalnya > 2 dan di-*repair* dengan asisten yang belum mendapat jadwal.
2. Dari hasil pengujian ukuran populasi sebanyak 10 dengan nilai rata-rata *fitness* tertingginya yaitu 1, banyaknya iterasi yaitu 10 dengan nilai rata-rata *fitness* tertingginya 1, kombinasi C<sub>1</sub> dan C<sub>2</sub> sebesar 2.3 sampai 0.5 dan C<sub>2</sub> mulai dari 0.7 sampai 2.5 dengan nilai rata-rata *fitness* tertingginya 1 dan *fitness* terbaik hal ini menunjukkan bahwa solusi yang didapatkan sudah mendekati optimum.

**5.2 Saran**

Berdasarkan beberapa kesimpulan yang telah didapatkan, sistem optimasi penjadwalan praktikum Kecerdasan Buatan dengan M-RCPSO memiliki beberapa kekurangan sehingga dibu-tuhkan beberapa saran untuk pengembangan sistem ke penelitian selanjutnya antara lain:

1. Ruang lingkup dalam optimasi penjadwalan ini masih pada tingkat jurusan, sehingga dapat diperbesar sampai ketinggian universitas.
2. Proses pengujian diharapkan dapat menguji ke semua aspek variabel yang ada dalam algoritma sehingga dapat mengetahui tingkat *fitness* secara akurat dan lebih relevan.
3. Optimasi penjadwalan ini bisa lebih berkembang dengan menambahkan lebih banyak *constraint*.

4. Perlu dilakukan pencegahan konvergensi dini dengan menggunakan algoritma lain seperti *random injection*.

## 6. DAFTAR PUSTAKA

- ARIANI, DIAN, dkk., 2011. Optimasi Penjadwalan Mata Kuliah Di Jurusan Teknik Informatika PENS Dengan Menggunakan Algoritma Particle Swarm Optimization (PSO). pp. 1-11.
- MANSUR, dkk., 2014. Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi. *Jurnal Sistem Informasi Bisnis*, pp. 11-19.
- MAWADDAH, NIA KURNIA. 2006. Optimasi Penjadwalan Ujian Menggunakan Algoritma Genetika. Vol 2, No 2, Hal 1-8. Universitas Brawijaya Malang.
- PUSPANINGRUM, WIGA AYU, dkk. 2013. Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. *Jurnal Teknik Pomits*. Vol 2, No 1, Hal 127-131. Institut Teknologi Sepuluh November.
- PUTRI, LING RIA SUKMANA, dkk., 2013. Penyelesaian Penjadwalan Kuliah Menggunakan Algoritma Particle Swarm Optimization and Hybrid Dimension Association Rule.
- SOLIMAN, OMAR S., dkk., 2014. Classification of Diabetes Mellitus using Modified Particle Swarm Optimization and Least Squares Support Vector Machine. *International Journal of Computer and Technology (IJCTT)*, vol. 8, no. 1, Februari 2014, pp. 38-44.
- SUHARTONO, ENTOT. 2015. Optimasi Penjadwalan Mata Kuliah dengan Algoritma Genetika (Studi Kasus di AMIK JTC Semarang). No 2, September 2015, Hal 132-146. AMIK JTC Semarang.