

OPTIMALISASI PENJADWALAN MATA KULIAH DENGAN KOMBINASI ALGORITMA GENETIK DAN TABU SEARCH

*Erick Fernando, S.Kom, MSI
Dosen tetap STIKOM Dinamika Bangsa Jambi*

Abstrak

Dalam menyusun penjadwalan tentu ada beberapa aspek yang menjadi parameter-parameter penjadwalan yang akan menjadi acuan dalam menilai suatu susunan jadwal perkuliahan yang ideal. Aspek-aspek tersebut diantaranya, tidak boleh ada jadwal mengajar dosen yang bentrok, tidak boleh ada kelas yang bentrok ruangan mata kuliah, tidak boleh ada perkuliahan pada jam larangan perkuliahan, adanya dosen yang memiliki jam-jam tidak bisa mengajar, adanya pembagian ruangan untuk masing-masing mata kuliah, dan lain sebagainya.

Kata kunci; Penjadwalan Mata Kuliah, Algoritma Genetik Dan Tabu Search

PENDAHULUAN

Penjadwalan kegiatan belajar mengajar dalam suatu kampus adalah hal yang rumit. Terdapat berbagai aspek yang berkaitan dalam penjadwalan tersebut yang harus dilibatkan antara lain terdapat jadwal-jadwal di mana dosen yang bersangkutan tidak bisa mengajar, tidak boleh ada jadwal mata kuliah yang bentrok, adanya kemungkinan dosen akan menangani lebih dari satu mata kuliah. Pekerjaan penjadwalan mata kuliah ini akan semakin berat selain jika banyak aspek-aspek perkuliahan yang dilibatkan, juga jika melibatkan banyak kelas per angkatan.

Dalam penyusunan jadwal kuliah ini pun terdapat sangat banyak kemungkinan yang selayaknya dicoba untuk menemukan penjadwalan yang terbaik. Karena itu dibutuhkan metode optimasi yang dapat diterapkan untuk mengerjakan penjadwalan mata kuliah ini. Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah dengan menggunakan pendekatan algoritma genetik dan *tabu search*.

Dengan mengkombinasikan kedua algoritma ini, di mana kemungkinan ditemukannya solusi yang sama berulang-ulang di algoritma genetik dapat dikurangi dengan memanfaatkan *tabu list* di algoritma *tabu search*, dan resiko terjebak di optimum lokal pada *tabu search* yang hanya menggunakan satu solusi tunggal dapat di kurangi dengan parameter *crossover* dan mutasi di algoritma genetik. Diharapkan dengan kombinasi kedua algoritma ini akan diperoleh optimasi penjadwalan yaitu terpilihnya solusi penjadwalan yang terbaik diantara pilihan-pilihan kandidat solusi lainnya, sehingga terbentuk kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan.

LANDASAN TEORI

2.1 ALGORITMA

Kata algoritma diambil dari nama ilmuwan muslim Abu Ja'far Muhammad bin Musa Al-Khwarizmi (780-846 M) yang banyak menghasilkan karya dalam bidang matematika (Fathul Wahid, 2004:1).

Pemecahan sebuah masalah pada hakekatnya adalah menemukan langkah-langkah tertentu yang jika dijalankan efeknya akan memecahkan masalah tersebut. Langkah-langkah itulah yang disebut algoritma. Jadi algoritma adalah urutan langkah-langkah yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah dalam rentang waktu tertentu.

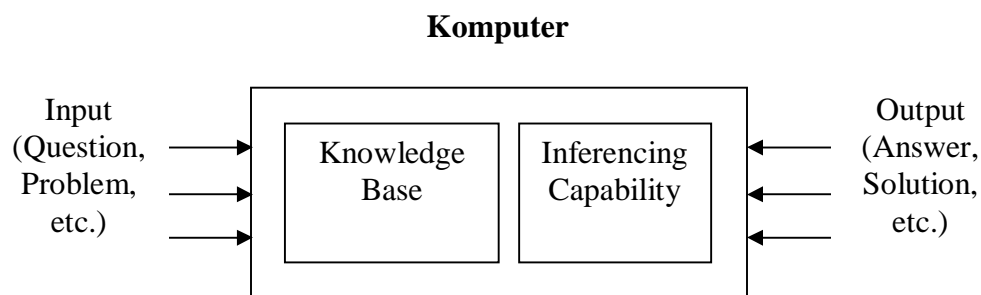
Karakteristik atau syarat algoritma (Fathul Wahid, 2004:3), antara lain:

1. Algoritma harus tidak ambigu (*unambiguous*)
Deskripsi langkah-langkah dalam algoritma harus dan hanya mempunyai tafsiran tunggal.
2. Algoritma harus tepat (*precise*)
Algoritma harus menyatakan urutan langkah-langkahnya. Kapan sebuah langkah x harus dilakukan, harus dinyatakan dengan jelas. Algoritma harus menyatakan dengan jelas kapan berhenti dari sebuah langkah
3. Algoritma harus pasti (*definite*)
Jika serangkaian langkah yang sama dilakukan dua kali maka hasilnya harus selalu sama. Sebagai ilustrasi, jika dua orang koki mengikuti resep yang sama, maka rasa kue yang dihasilkan harus sama. Kalau hasil akhir atau rasa kue tidak sama, pasti ada beberapa perbedaan yang tidak disadari, seperti merek bahan yang berbeda, atau lama memasak yang berbeda, atau ada perbedaan pada sesuatu yang lain.
4. Algoritma harus berhingga (*finite*)
Serangkaian langkah dalam algoritma harus dapat dilaksanakan pada rentang waktu tertentu.

2.2 ARTIFICIAL INTELLIGENCE

2.2.1 Pengertian Artificial Intelligence

Tanpa memiliki kemampuan menalar yang baik, manusia dengan pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Begitu pulalah sebaliknya. Untuk itulah, agar komputer bisa bertindak sebaik manusia, maka perlu dibekali dengan pengetahuan dan kemampuan penalaran. Dengan basis pengetahuan dan kemampuan untuk menarik kesimpulan melalui pengalaman (*inferensi*), maka komputer dapat digunakan sebagai alat bantu dalam memecahkan masalah dan pengambilan keputusan.



Menurut Gambar 2.1 Penerapan Konsep AI dalam Komputer dan dapat dipandang dari berbagai (Sri Kusumadewi, 2003:2)

1. Sudut Kecerdasan buatan akan membuat mesin menjadi ‘cerdas’ (mampu berbuat seperti apa yang dilakukan oleh manusia).
2. Sudut pandang penelitian. Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.
3. Sudut pandang bisnis. Kecerdasan buatan adalah kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalah-masalah bisnis.
4. Sudut pandang pemrograman. Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

2.3 ALGORITMA GENETIK (GA)

2.3.1 Pengertian Algoritma Genetik

Algoritma genetik berangkat dari membangkitkan sejumlah individu sebagai anggota populasi, secara acak atau berdasarkan suatu pengetahuan tertentu. Jumlah individu dalam populasi tersebut selalu tetap selama proses evolusi. Setiap individu memiliki hanya satu kromosom yang merupakan representasi dari solusi dengan panjang yang tetap selama proses evolusi. Artinya, jumlah gen dalam setiap kromosom tidak akan bertambah atau berkurang selama proses evolusi (Suyanto, 2008:116). Beberapa definisi penting dalam algoritma genetik, antara lain (Achmad Basuki, 2003:3) :

- a. Gen, sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang disebut kromosom.
- b. *Allele*, nilai dari gen.
- c. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.

- d. Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
- e. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
- f. Generasi, menyatakan satu-satuan siklus proses evolusi.
- g. Nilai *fitness*, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

2.4 **TABU SEARCH**

Tabu Search (TS) pertama kali diperkenalkan oleh Glover sekitar tahun 1970-an (Suyanto, 2010:135). Glover menyatakan bahwa TS adalah salah satu prosedur *metaheuristik* tingkat tinggi untuk penyelesaian permasalahan optimasi kombinatorial.

Untuk menjaga agar solusi terbaik tidak hilang, TS menyimpan solusi terbaik dan terus mencari berdasarkan solusi terakhir. Selain itu, metode ini mengingat sebagian solusi yang pernah ditemui dan melarang untuk menggunakan solusi yang telah ditelusuri untuk menghindari pengulangan yang sia-sia. TS menggunakan struktur *memory* yang disebut *Tabu List* untuk menyimpan atribut dari sebagian *move* yang telah diterapkan pada iterasi-iterasi sebelumnya. *Tabu List* digunakan untuk menolak solusi-solusi yang memenuhi atribut tertentu agar proses pencarian tidak berulang-ulang pada daerah solusi yang sama dan untuk menuntun proses pencarian menelusuri solusi-solusi yang belum pernah dikunjungi (Suyanto, 2010:136).

Secara umum, algoritma TS dapat dituliskan pada gambar 2.3 berikut.

1. Bangkitkan solusi awal yang layak, misalkan s , secara acak atau menggunakan metode heuristik tertentu.
2. $BiayaOptimum = Biaya(s)$
3. $s^* = s$ { s^* adalah solusi terbaik yang diperoleh}
4. $TabuList = null$
5. Repeat
 - a. $V^* =$ himpunan solusi yang merupakan tetangga dari s yang memenuhi kriteria aspirasi atau tidak berada dalam $TabuList$
 - b. Pilih s' { s' adalah solusi yang memiliki biaya minimum di dalam V^* }
 - c. Simpan *move* yang berlawanan ke dalam $TabuList$, yang mengubah s ke s'
 - d. $s = s'$
if ($Biaya(s) < BiayaOptimum$) then
 $s^* = s$
 $BiayaOptimum = Biaya(s)$

Gambar 2.3 Algoritma *Tabu Search* (Suyanto, 2010:138)

Pada banyak literatur, TS memang efisien untuk menyelesaikan banyak masalah optimasi yang sulit. TS mungkin menemukan optimum global, tetapi TS tidak bisa menjamin ditemukannya optimum global karena banyak parameter yang harus ditentukan secara hati-hati.

ANALISIS DAN PERANCANGAN

3.1 Pembentukan Kromosom (Individu)

Satu kromosom (individu) melambangkan satu alternatif solusi susunan jadwal mata kuliah. Satu kromosom terdiri dari banyak gen. Satu gen melambangkan satu pertemuan perkuliahan.

Representasi suatu gen dalam penelitian ini, yaitu:

- a. Hari, jam, dan ruangan sebagai nilai dari gen (*allele*) yang dideklarasikan dengan variabel $N_{i,j}$
- b. Pertemuan dalam kelas sebagai gen yang dideklarasikan dengan variabel G_n

1. Nilai Gen (*Allele*)

Setiap gen akan memiliki satu buah *allele*. *Allele* yang disimbolkan dengan variabel $N_{i,j}$ akan membawa informasi berupa hari, jam dan ruangan tertentu yang mana nilai dari $N_{1,1}$ hingga $N_{30,25}$ tidak akan sama.

Tabel 3.1 Nilai Gen (*Allele*)

| | | Ruangan | | | | | |
|-----|-----------|-----------|-----------|-----------|-----------|------------|------------|
| | | R01 | R02 | R03 | R04 | ... | R25 |
| D1 | Waktu | | | | | | |
| | H1 | $N_{1,1}$ | $N_{1,2}$ | $N_{1,3}$ | $N_{1,4}$ | ... | $N_{1,25}$ |
| | H2 | $N_{2,1}$ | $N_{2,2}$ | $N_{2,3}$ | $N_{2,4}$ | ... | $N_{2,25}$ |
| | H3 | $N_{3,1}$ | $N_{3,2}$ | $N_{3,3}$ | $N_{3,4}$ | ... | $N_{3,25}$ |
| | H4 | $N_{4,1}$ | $N_{4,2}$ | $N_{4,3}$ | $N_{4,4}$ | ... | $N_{4,25}$ |
| H5 | $N_{5,1}$ | $N_{5,2}$ | $N_{5,3}$ | $N_{5,4}$ | ... | $N_{5,25}$ | |
| ... | H1 | ... | ... | ... | ... | ... | ... |
| ... | H2 | ... | ... | ... | ... | ... | ... |

| | | | | | | | |
|-----------|-----------|-------------------|-------------------|-------------------|-------------------|-----|--------------------|
| | H3 | ... | ... | ... | ... | ... | ... |
| | H4 | ... | ... | ... | ... | ... | ... |
| | H5 | ... | ... | ... | ... | ... | ... |
| D6 | H1 | N _{26,1} | N _{26,2} | N _{26,3} | N _{26,4} | ... | N _{26,25} |
| | H2 | N _{27,1} | N _{27,2} | N _{27,3} | N _{27,4} | ... | N _{27,25} |
| | H3 | N _{28,1} | N _{28,2} | N _{28,3} | N _{28,4} | ... | N _{28,25} |
| | H4 | N _{29,1} | N _{29,2} | N _{29,3} | N _{29,4} | ... | N _{29,25} |
| | H5 | N _{30,1} | N _{30,2} | N _{30,3} | N _{30,4} | ... | N _{30,25} |

2. Barisan Gen

Gen dibentuk berdasarkan banyaknya pertemuan perkuliahan dalam satu minggu. Untuk mata kuliah dengan bobot 4 sks, berarti terdapat 2 kali pertemuan dalam satu minggu. Untuk menghitung jumlah gen yang diperlukan, dilakukan dengan cara perhitungan jumlah gen:

$$Len = \sum_{k=1}^{JMK2} KL_k + \sum_{k=1}^{JMK4} KL_k * 2 \dots\dots\dots(4.1)$$

Keterangan:

Len = jumlah gen

KL_k = jumlah kelas yang dibuka untuk satu mata kuliah pada suku ke – *k*

JMK2 = jumlah mata kuliah berbobot 2 sks

JMK4 = jumlah mata kuliah berbobot 4 sks

Berdasarkan tabel 3.1, diketahui bahwa jumlah mata kuliah berbobot 2 sks sebanyak 62 mata kuliah dan yang berbobot 4 sks sebanyak 37 buah. Dengan pertimbangan rata-rata kelas yang dibuka untuk satu mata kuliahnya sebanyak 3 kelas, maka :

| | | | | | | |
|----------------|----------------|----------------|----------------|----------------|-----|------------------|
| <i>Allele</i> | <i>Allele</i> | <i>Allele</i> | <i>Allele</i> | <i>Allele</i> | ... | <i>Allele</i> |
| G ₁ | G ₂ | G ₃ | G ₄ | G ₅ | ... | G ₄₀₈ |

Gambar 3.1 Barisan Gen

Jumlah gen diperkirakan adalah sebanyak 408 buah.

$$Len = (62*3) + (37*3*2)$$

$$Len = 408$$

Setiap gen yang disimbolkan dengan G_k akan membawa informasi berupa kelas, mata kuliah, dosen, semester, sks dan jurusan.

1.2 Evaluasi *Fitness*

Evaluasi *fitness* dilakukan untuk mengetahui seberapa baik suatu alternatif solusi penjadwalan yang disusun. Semakin tinggi nilai *fitness*-nya maka semakin baik solusi jadwal mata kuliah tersebut. Untuk mengevaluasi nilai *fitness*, diperlukan penetapan parameter-parameter sebagai penentu baik tidaknya suatu alternatif susunan jadwal mata kuliah. Bila parameter-parameter tersebut dilanggar, maka akan dikenai *penalty*.

Fungsi *fitness* yang digunakan termasuk fungsi minimum, yang berarti semakin besar *penalty*-nya, semakin kecil nilai *fitness*.

Parameter-parameter yang didefinisikan dalam penelitian ini, antara lain:

- a. Perkuliahan tidak boleh berlangsung di ruangan yang tidak terdaftar untuk perkuliahan tersebut.
- b. Perkuliahan tidak boleh berlangsung pada jam-jam tertentu.
- c. Kelas perkuliahan tidak boleh bentrok ruangan dengan kelas perkuliahan lainnya.
- d. Dosen tidak boleh memiliki jadwal mengajar yang bentrok.
- e. Perkuliahan tidak boleh diadakan pada jam di mana dosen yang bersangkutan tidak bisa mengajar. Parameter ini berlaku jika ada jam-jam tertentu yang didaftarkan sebagai jam dosen tidak bisa mengajar.
- f. Satu kelas perkuliahan yang memiliki bobot 4 sks (2 kali pertemuan dalam 1 minggu) tidak boleh memiliki jam kuliah yang berjauhan dengan jam kuliah untuk pertemuan berikutnya. Tujuannya adalah menghabiskan 2 kali pertemuan pada suatu kelas dalam waktu sehari dengan jam kuliah yang bersebelahan. Parameter ini berlaku jika suatu kelas terdaftar untuk memiliki jam perkuliahan yang bersebelahan.
- g. Untuk setiap kelas perkuliahan mata kuliah tertentu tidak boleh memiliki jam kuliah yang beririsan dengan mata kuliah lain di kelas perkuliahan itu juga. Parameter ini berlaku jika telah dipilih oleh *user*.

Bila terjadi pelanggaran pada parameter yang termasuk dalam batasan keras, maka akan dikenai *penalty* sebesar 100 point. Untuk parameter pada batasan lunak, nilai *penalty* bersifat relatif. Berikut merupakan tabel nilai *penalty* untuk setiap parameter.

Tabel 3.2 Nilai *Penalty*

| No. | Parameter | <i>Penalty</i> |
|-----|---|----------------|
| 1. | Perkuliahan tidak terdaftar di ruangan kuliah | 100 |
| 2. | Perkuliahan berlangsung pada jam larangan kuliah | 100 |
| 3. | Kelas perkuliahan bentrok dengan kelas lain | 100 |
| 4. | Dosen memiliki jam mengajar yang bentrok | 100 |
| 5. | Perkuliahan pada jam dosen bersangkutan tak bisa mengajar | 0-100 |

| | | |
|----|--|------|
| 6. | Perkuliahan berbobot 4 sks tidak bersebelahan jam kuliahnya dengan pertemuan berikutnya | 0-20 |
| 7. | Jenis kelas tertentu beririsan jam perkuliahannya dengan mata kuliah berbeda yang dikontrak oleh kelas ini juga. | 0-20 |

Perhitungan nilai *fitness* atas suatu kromosom (alternatif solusi jadwal mata kuliah), dilakukan dengan menggunakan rumus pada persamaan 2.1 di mana:

$$a = 0,0000000001$$

$$h = \sum_{k=1}^{JP} B_k * P_k \dots\dots\dots(4.2)$$

Keterangan:

JP = jumlah parameter

B_k = bobot *penalty* pada parameter ke- k

P_k = banyaknya pelanggaran pada parameter ke- k

1.3 Parameter Algoritma

Parameter algoritma merupakan parameter-parameter operasi yang digunakan di dalam kombinasi algoritma genetik dan *tabu search*. Nilai dari masing-masing parameter diatur secara *default* oleh penulis. Namun nilai tersebut masih dapat diubah oleh *user* sesuai kebutuhan, kecuali pada parameter ukuran *tabu list*. Jenis *crossover* yang digunakan adalah *crossover* satu titik, dalam ketetanggaan digunakan metode perpindahan *neighbors* antar tetangga sebelah.

Tabel 3.3 Nilai Parameter Algoritma

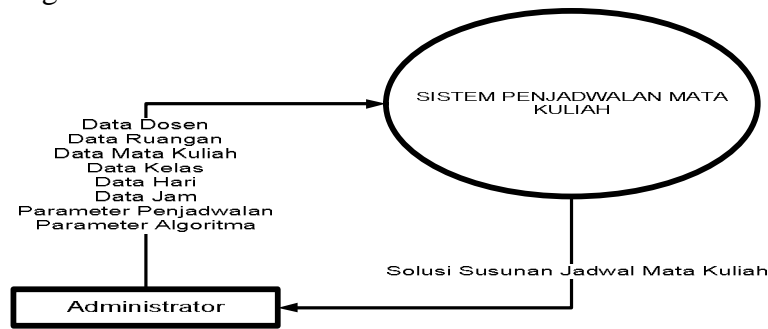
| Parameter | Nilai |
|-------------------------------|-------|
| Ukuran Populasi | 50 |
| Probabilitas <i>Crossover</i> | 0,7 |
| Probabilitas Mutasi | 0,03 |
| Jumlah Generasi | 20 |
| Ukuran <i>tabu list</i> | 1 |

3.4. ANALISIS KEBUTUHAN PERANGKAT LUNAK

1.3.1 Analisis Proses Perangkat Lunak

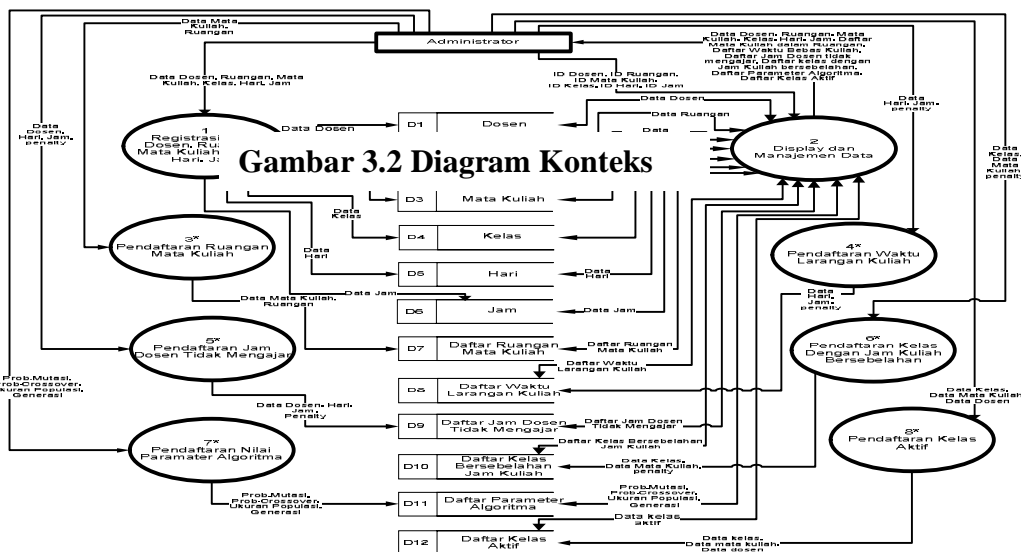
Untuk menjelaskan proses aliran data pada pengaplikasian kombinasi algoritma genetik dan *tabu search* untuk penjadwalan mata kuliah, penulis menggunakan DFD (*Data Flow Diagram*) sebagai modelnya.

1. Diagram Konteks



Pada gambar 4.7 Diagram Konteks menggambarkan diagram konteks SISTEM PENJADWALAN MATA KULIAH. Hanya terdapat 1 entitas eksternal, yaitu administrator yang mengatur manajemen data dan parameter penjadwalan.

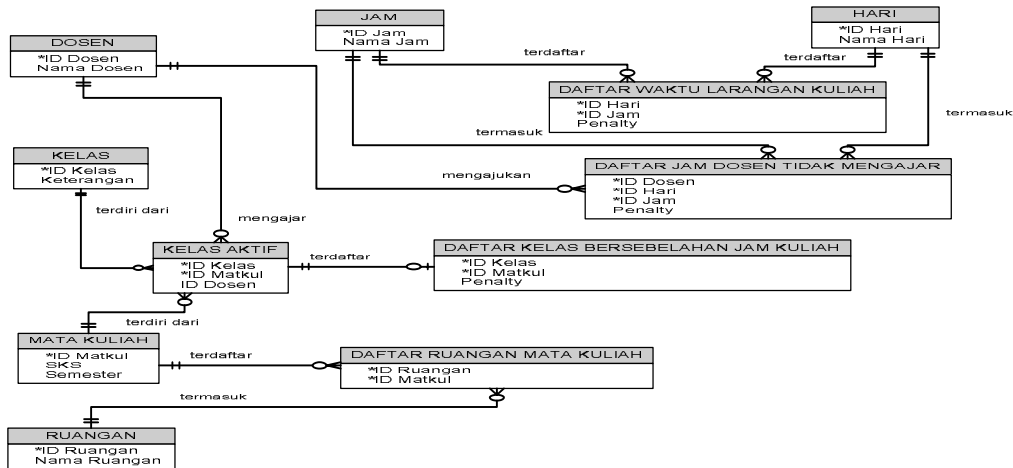
2. DFD Level 0



Gambar 3.3 DFD Level 0

1.3.2 Analisis Kebutuhan Data

Untuk menjelaskan kebutuhan data dari perangkat lunak yang dirancang, akan dimodelkan dalam bentuk ERD (*Entity Relationship Diagram*) berikut ini.



Gambar 3.4 ERD Penjadwalan Mata Kuliah

1.4 RANCANGAN OUTPUT

Rancangan *output* dari kombinasi algoritma genetik dan *tabu search* dalam optimalisasi penjadwalan mata kuliah adalah berupa suatu susunan penjadwalan yang ditampilkan dalam bentuk seperti berikut.

| | | Ruang1 | Ruang2 | Ruang3 | Ruang4 | Ruang5 | ... | n |
|-------|------|--------|--------|--------|--------|--------|-----|----|
| Hari1 | Jam1 | MK | MK | MK | MK | MK | ... | MK |
| | Jam2 | MK | MK | MK | MK | MK | ... | MK |
| | Jam3 | MK | MK | MK | MK | MK | ... | MK |
| | ... | MK | MK | MK | MK | MK | ... | MK |
| Hari2 | Jam1 | MK | MK | MK | MK | MK | ... | MK |
| | Jam2 | MK | MK | MK | MK | MK | ... | MK |
| | Jam3 | MK | MK | MK | MK | MK | ... | MK |
| | ... | MK | MK | MK | MK | MK | ... | MK |
| ... | Jam1 | MK | MK | MK | MK | MK | ... | MK |
| | Jam2 | MK | MK | MK | MK | MK | ... | MK |
| | Jam3 | MK | MK | MK | MK | MK | ... | MK |
| | ... | MK | MK | MK | MK | MK | ... | MK |

Gambar 3.3 Rancangan Output Susunan Jadwal Mata Kuliah

Ket: MK = Penempatan Kandidat Mata Kuliah

1.5 RANCANGAN INPUT

Untuk dapat menghasilkan *output* yang sesuai dengan kebutuhan, tentunya harus memiliki rancangan *input* yang berfungsi untuk memanipulasi data atau informasi. Rancangan *input* dari aplikasi kombinasi algoritma genetik dan *tabu search* untuk penjadwalan mata kuliah adalah sebagai berikut:

1.5.1 Rancangan Input Parameter Algoritma

Merupakan rancangan tampilan *input* parameter-parameter algoritma, di mana pada form ini dapat dilakukan proses *setting* ukuran populasi, probabilitas *crossover*, probabilitas mutasi, jumlah generasi dan ukuran *tabu list*, melalui form ini juga *user* akan memulai optimasi.

Gambar 3.4 Rancangan *Input* Parameter Algoritma

1.6 RANCANGAN STRUKTUR DATA

Dalam perancangan aplikasi kombinasi algoritma genetik dan *tabu search* untuk penjadwalan mata kuliah ini, digunakan *database* sebagai media penyimpanan data. Adapun rancangan tabel-tabel yang akan digunakan antara lain:

- a. Tabel *Master* Dosen

Primary Key : ID_Dosen

Tabel 3.5 Rancangan Tabel *Master* Dosen

| Nama Field | Tipe | Panjang | Keterangan |
|------------|---------|---------|------------|
| ID_Dosen | Varchar | 4 | Id Dosen |
| Nama_Dosen | Varchar | 50 | Nama Dosen |

- b. Tabel *Master* Ruangan

Primary Key : ID_Ruangan

Tabel 3.6 Rancangan Tabel *Master* Ruangan

| Nama Field | Tipe | Panjang | Keterangan |
|--------------|---------|---------|--------------|
| ID_Ruangan | Varchar | 3 | ID Ruangan |
| Nama_Ruangan | Varchar | 30 | Nama ruangan |

- c. Tabel *Master* Mata Kuliah

Primary Key : ID_Matakuliah

Tabel 3.7 Rancangan Tabel *Master* Mata Kuliah

| Nama Field | Tipe | Panjang | Keterangan |
|-----------------|---------|---------|------------------|
| ID_Matakuliah | Varchar | 8 | Id Mata kuliah |
| Nama_Matakuliah | Varchar | 50 | Nama Mata kuliah |
| Semester | Int | 15 | Semester |
| Bobot_SKS | Int | 15 | Bobot sks |
| Jurusan | Varchar | 2 | Jurusan |

- d. Tabel *Master* Kelas
Primary Key : ID_Kelas

Tabel 3.8 Rancangan Tabel *Master* Kelas

| Nama Field | Tipe | Panjang | Keterangan |
|------------|---------|---------|------------|
| ID_Kelas | Varchar | 5 | Id kelas |
| Keterangan | Varchar | 255 | Keterangan |

- e. Tabel *Master* Hari
Primary Key : ID_Hari

Tabel 3.9 Rancangan Tabel *Master* Hari

| Nama Field | Tipe | Panjang | Keterangan |
|------------|---------|---------|------------|
| ID_Hari | Varchar | 2 | Id hari |
| Nama_Hari | Varchar | 10 | Nama hari |

- f. Tabel *Master* Jam
Primary Key : ID_Jam

Tabel 3.10 Rancangan Tabel *Master* Jam

| Nama Field | Tipe | Panjang | Keterangan |
|------------|---------|---------|------------|
| ID_Jam | Varchar | 2 | Id jam |
| Nama_Jam | Varchar | 50 | Nama jam |

- g. Tabel Daftar Kelas Mata Kuliah Aktif
Primary Key : ID_Kelas + ID_Matakuliah

Tabel 3.11 Rancangan Tabel Daftar Kelas Mata Kuliah Aktif

| Nama Field | Tipe | Panjang | Keterangan |
|---------------|---------|---------|-------------------|
| ID_Kelas | Varchar | 5 | Id kelas |
| ID_Matakuliah | Varchar | 8 | Id matakuliah |
| ID_Dosen | Varchar | 4 | Id dosen pengajar |

- h. Tabel Daftar Ruang Mata Kuliah
Primary Key : ID_Ruangan + ID_Matakuliah

Tabel 3.12 Rancangan Tabel Daftar Ruang Mata Kuliah

| Nama Field | Tipe | Panjang | Keterangan |
|---------------|---------|---------|---------------|
| ID_Ruangan | Varchar | 3 | Id ruangan |
| ID_Matakuliah | Varchar | 8 | Id matakuliah |

- i. Tabel Jam Dosen Tidak Bisa Mengajar
Primary Key : ID_Dosen + ID_Hari + ID_Jam

Tabel 3.13 Rancangan Tabel Jam Dosen Tidak Mengajar

| Nama Field | Tipe | Panjang | Keterangan |
|------------|---------|---------|----------------------|
| ID_Dosen | Varchar | 4 | Id dosen |
| ID_Hari | Varchar | 2 | Id hari |
| ID_Jam | Varchar | 2 | Id jam |
| Penalty | Int | 15 | Bobot <i>penalty</i> |

- j. Tabel Jam Larangan Kuliah
Primary Key : ID_Hari + ID_Jam

Tabel 3.14 Rancangan Tabel Jam Larangan Kuliah

| Nama Field | Tipe | Panjang | Keterangan |
|------------|---------|---------|----------------------|
| ID_Hari | Varchar | 2 | Id hari |
| ID_Jam | Varchar | 2 | Id jam |
| Penalty | Int | 15 | Bobot <i>penalty</i> |

- k. Tabel Kelas Bersebelahan Jam Kuliah
Primary Key : ID_Kelas + ID_Matakuliah

Tabel 3.15 Rancangan Tabel Daftar Ruang Mata Kuliah

| Nama Field | Tipe | Panjang | Keterangan |
|---------------|---------|---------|----------------------|
| ID_Kelas | Varchar | 5 | Id kelas |
| ID_Matakuliah | Varchar | 8 | Id mata kuliah |
| Penalty | Int | 15 | Bobot <i>penalty</i> |

- l. Tabel Parameter Algoritma
Primary Key : ID_Parameter

Tabel 3.16 Rancangan Tabel Parameter Algoritma

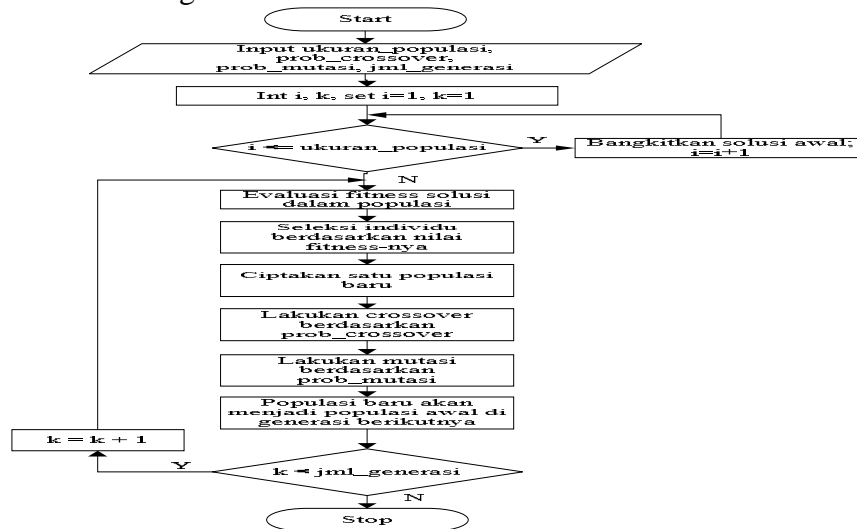
| Nama Field | Tipe | Panjang | Keterangan |
|--------------|---------|---------|--------------|
| ID_Parameter | Varchar | 3 | Id parameter |

| | | | |
|------------|---------|---|----------------------|
| Keterangan | Varchar | 8 | Keterangan parameter |
| Nilai | Single | 8 | Nilai parameter |

1.7 RANCANGAN *FLOWCHART*

Perancangan aplikasi penjadwalan mata kuliah ini berdasarkan dari kombinasi algoritma genetik dan *tabu search*. Untuk memperlihatkan proses-proses dasar alur logika dari algoritma, penulis menggunakan *flowchart* sebagai modelnya.

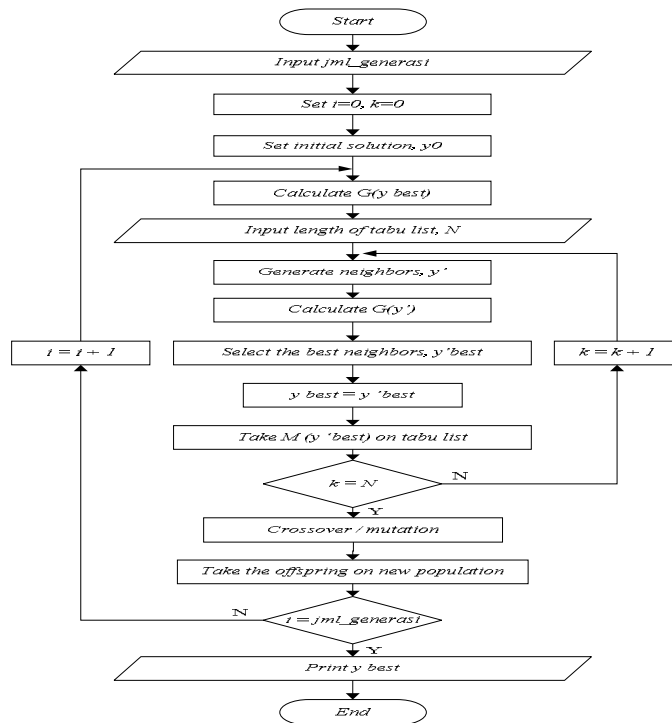
1. *Flowchart* Algoritma Genetik



Gambar 3.5 *Flowchart* Algoritma Genetik

Flowchart pada gambar 3.5 merupakan proses dasar dari Algoritma Genetik. Proses diawali dari penginputan parameter algoritma, kemudian pembentukan populasi awal hingga melalui proses seleksi, *crossover*, dan mutasi, kemudian dilanjutkan lagi untuk generasi berikutnya.

2. *Flowchart* Kombinasi Algoritma Genetik dan *Tabu Search*



Gambar 3.6 *Flowchart* Kombinasi Algoritma Genetik dan *Tabu Search*

Flowchart pada gambar 3.6 merupakan gambaran proses dasar dari kombinasi algoritma genetik dan *tabu search*. Proses diawali dari penginputan parameter algoritma, kemudian pembentukan 1 solusi awal yang akan dibangkitkan kandidat-kandidat solusinya menggunakan metode ketetangaan hingga sebanyak ukuran 1 populasi, lalu melalui kandidat-kandidat solusi tersebut akan melalui proses *crossover* dan mutasi, hingga terpilihnya satu kandidat solusi terbaik yang kemudian untuk dilanjutkan lagi prosesnya pada generasi berikutnya.

DAFTAR PUSTAKA

- Achmad Basuki, 2003, *Algoritma Genetika Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning*, http://budi.blog.undip.ac.id/files/2009/06/algoritma_genetika.pdf, Desember 2010.
-, 2003, *Strategi Menggunakan Algoritma Genetika*, <http://www2.eepis-its.edu/~basuki/lecture/StrategiAlgoritmaGenetika.pdf>, Desember 2010.
- Adi Nugroho, 2005, *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Bandung : Informatika Bandung.

- Anharku, 2009, *Flowchart*, <http://ilmukomputer.org/2009/06/14/flowchart/>, Desember 2010.
- Aries Syamsuddin, 2004, *Pengenalan Algoritma Genetik*, <http://ilmukomputer.org/2006/08/25/pengenalan-algoritma-genetik/>, Desember 2010.
- Dennis, Alan., Wixom B. Haley., Roth Roberta M., 2010, *System Analysis and Design*. 4th ed. Hoboken : John Wiley & Sons.
- Eko Budi P., 2008, *Perancangan dan Analisis Algoritma*. Yogyakarta : Graha Ilmu.
- Fathul Wahid, 2004, *Dasar-Dasar Algoritma dan Pemrograman*. Yogyakarta : Andi Publisher.
- Fena Yuanita, 2006, *Analisa perbandingan algoritma kombinasi genetic & tabu search dengan metode Ant Colony pada permasalahan penjadwalan flowshop Thesis Skripsi S-1 Universitas Kristen Petra*, http://dewey.petra.ac.id/dts_res_detail.php?knokat=3424, November 2010.
- Jogiyanto, 2005, *Analisis dan Disain Sistem Informasi: pendekatan terstruktur teori dan praktek aplikasi bisnis*. Edisi III. Yogyakarta : Andi.
- Leong, Marlon., 2006, *Dari Programmer untuk Programmer Visual Basic*. Yogyakarta : Andi.
- Novandry Widyastuti, Astika Ratnawati, Rahma N. Cahyani, 2008, *Optimasi Penjadwalan Kegiatan Belajar Mengajar Dengan Algoritma Genetik*.