

OPTIMASI PENJADWALAN PERKULIAHAN DI UNIVERSITAS TRIDINANTI PALEMBANG

Ferdian Tomhart¹, Afriyudi², Mutakin Bakti³,
Dosen Universitas Bina Darma^{2,3}, Mahasiswa Universitas Bina Darma²
Jalan Jenderal Ahmad Yani No.12 Palembang
Pos-el : babeyudi@mail.binadarma.ac.id¹, @mail.binadarma.ac.id²,
@mail.binadarma.ac.id³

Abstract: Scheduling of teaching and learning in college is a tricky thing. Problems often arise in the teaching and learning scheduler possibilities professors will administer more than one course that is no, because there is the possibility of the number of courses and number of lecturers are not comparable, so it should be well thought out solutions that lecturers do not administer two different courses on the day and the same hour. Furthermore, it should also be considered so that the availability of class learning activities can be carried out. Based on the above, the author wishes mengoptimasai scheduling classes at University School Guidance and Counseling Especially Tridinanti Palembang. The first goal of the optimization is obtained optimization scheduling scheduling a condition where there is the best combination for a couple of courses and lecturers as a whole, there is no problem on the student timetable clashes, as well as the availability of adequate space and appropriate facilities for all existing courses. So in writing this thesis proposal was appointed a title of "Class Scheduling Optimization at the University of Palembang Tridinanti". Keywords: Optimization, Scheduling, Genetic Algorithm

Abstrak : Penjadwalan kegiatan belajar mengajar dalam suatu kampus adalah hal yang rumit. Permasalahan yang sering timbul dalam penjadwal belajar dan mengajar yaitu kemungkinan-kemungkinan dosen akan mampu lebih dari satu mata kuliah yang ada, sebab ada kemungkinan jumlah mata kuliah dan jumlah dosen tidak sebanding, sehingga harus dipikirkan juga solusi agar dosen tidak mampu dua mata kuliah berbeda pada hari dan jam yang sama. Selain itu, harus dipertimbangkan juga ketersediaan kelas sehingga kegiatan belajar dapat dilaksanakan. Berdasarkan uraian di atas, maka penulis berkeinginan mengoptimasai penjadwalan perkuliahan di Universitas Tridinanti Palembang Khususnya Fakultas FKIP. Diharapkan dengan adanya optimasi penjadwalan akan diperoleh optimasi penjadwalan yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan, tidak ada permasalahan bentrokan jadwal pada sisi mahasiswa, serta ketersediaan ruang yang cukup dan sesuai secara fasilitas untuk seluruh mata kuliah yang ada. Maka dalam penulisan proposal skripsi ini diangkatlah sebuah judul yaitu "**Optimasi Penjadwalan Perkuliahan di Universitas Tridinanti Palembang**".

Kata Kunci : Optimasi, Penjadwalan, Algoritma Genetik

1 PENDAHULUAN

Dengan berkembangnya teknologi informasi diharapkan dapat menjadi media yang paling efektif untuk mencari dan menyebarkan informasi. Salah satunya adalah komputer yang saat ini sudah bukan barang baru dan kini tidak hanya digunakan untuk

kepentingan perkantoran tetapi juga dapat digunakan untuk kepentingan bisnis. Teknologi informasi yang didukung oleh perkembangan perangkat keras (*hardware*) dan perangkat lunak (*software*) secara langsung maupun tidak langsung. Sebagai contohnya yaitu para eksekutif yang banyak menggunakan teknologi

sebagai alat bantu dalam mengambil keputusan.

Permasalahan pada pembuatan jadwal mata kuliah di Universitas Tridianti saat ini banyak dosen mengeluh tentang jadwal tabrakan pada hari dan jam yang sama dan juga tidak tersedianya ruang kelas. Bagian tata usaha juga mengeluh dalam pembuatan jadwal tidak cepat, tepat dan kurang optimal, sehingga harus dipikirkan juga solusi agar dosen tidak mendapatkan mata kuliah pada hari dan jam yang sama. Selain itu, harus dipertimbangkan juga ketersediaan kelas sehingga kegiatan belajar dapat dilaksanakan. Sedangkan untuk bagian tata usaha agar dapat membuat jadwal lebih cepat dan optimal perlu dibangun suatu aplikasi penjadwalan.

Berdasarkan uraian di atas, maka penulis berkeinginan mengoptimisasi penjadwalan perkuliahan di universitas tridianti palembang khususnya fakultas keguruan ilmu pendidikan. Diharapkan dengan adanya optimisasi penjadwalan akan diperoleh optimisasi penjadwalan yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan, tidak ada permasalahan bentrokan jadwal pada sisi mahasiswa, serta ketersediaan ruang yang cukup dan sesuai secara fasilitas untuk seluruh mata kuliah yang ada. Maka dalam penulisan skripsi ini diangkatlah sebuah judul yaitu “**Optimasi Penjadwalan Perkuliahan di Universitas Tridianti Palembang**”.

2 METODE, ANALISIS DAN RANCANGAN

2.1 Optimasi Query

Teknik optimasi dapat dilakukan dengan beberapa cara. Terdapat 2 pendekatan optimasi yang umum digunakan sebagaimana diungkapkan oleh (Ermatita, 2010:3), yakni:

1. Heuristik atau *rule-based*

Teknik ini mengaplikasikan aturan heuristik untuk mempercepat proses *query*. Optimasi jenis ini mentransformasikan *query* dengan sejumlah aturan yang akan meningkatkan kinerja eksekusi, yakni:

- a. Melakukan operasi *selection* di awal untuk mereduksi jumlah baris
- b. Melakukan operasi *projection* di awal untuk mereduksi jumlah atribut
- c. Mengkonversikan *query* dengan banyak join menjadi *query* dengan banyak *subquery*.
- d. Melakukan operasi *selection* dan join yang paling kecil keluarannya sebelum operasi lain.

2. *Cost-based*

Teknik ini mengoptimasikan *cost* yang dipergunakan dari beberapa alternatif untuk kemudian dipilih salah satu yang menjadi *cost* terendah. Teknik ini mengoptimalkan urutan *join* terbalik yang dimungkinkan pada relasi-relasi $r1 \rightarrow r2 \rightarrow \dots rn$. Teknik ini dipergunakan untuk mendapatkan pohon *left-deep join* yang akan menghasilkan sebuah relasi sebenarnya pada node sebelah kanan

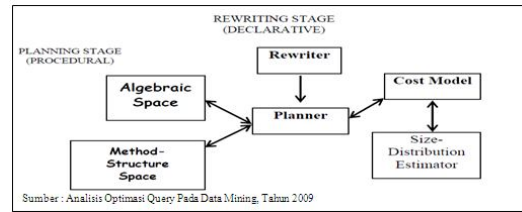
yang bukan hasil dari sebuah *intermediate join*. (Setiawan, 2004:2).

2.2 Query Optimizer

Query optimizer adalah bagian dari *DBMS* yang berfungsi mengoptimasi *query*. Proses yang biasanya terjadi dalam *optimizer* adalah *optimizer* memeriksa semua ekspresi-ekspresi aljabar yang sama yang diberikan *query* dan memilih salah satunya yang memiliki harga taksiran paling rendah. Tugas dari *optimizer* adalah untuk mentransformasikan inisial ekspresi *query* ke dalam sebuah rencana evaluasi yang menghasilkan *record* yang sama. (Ermatita, 2010:3).

Keuntungan dari *optimizer* adalah dapat mengakses semua informasi statistik dari sebuah *database*. Selain itu *optimizer* juga dapat dengan mudah untuk melakukan optimisasi kembali apabila informasi statistik sebuah *database* berubah dan *optimizer* dapat menangani strategi yang berbeda-beda dalam jumlah besar yang tidak mungkin dilakukan oleh manusia.

Proses optimisasi *query* dapat dianggap mempunyai dua tingkatan. Dua tingkatan tersebut adalah : *rewriting* dan *planning*. Hanya ada satu modul pada tingkat pertama yaitu *rewriter*, dimana semua modul-modul lainnya berada pada tingkat kedua. Tahap penulisan dapat disebut sebagai level *declarative*, sedangkan tahap perencanaan dapat juga disebut sebagai *level procedural*.



Gambar 1. Arsitektur *Query Optimizer*

Fungsi-fungsi dari masing-masing modul pada gambar diatas akan dijelaskan secara lebih rinci :

1. Rewriter

Modul ini melakukan transformasi-transformasi untuk sebuah parse *tree* dari *query* yang diberikan dan menghasilkan *query-query* yang sama yang diharapkan lebih efisien.

2. Planner

Modul ini adalah modul utama yang menguji semua rencana-rencana eksekusi *query* yang dihasilkan pada tingkat sebelumnya dan memilih satu dari semua rencana yang termurah, yang akan digunakan untuk menghasilkan jawaban dari *query* yang asli. *Planner* menggunakan *search strategy*, yang memeriksa *space* (tempat) dari rencana-rencana eksekusi. *Space* ini ditentukan oleh dua modul lainnya dari *optimizer*, yaitu *Algebraic Space* dan *Method-Structure Space*. Untuk kebanyakan bagian, dua modul ini dan *search strategy* menentukan harga seperti *running time* dari *optimizer* itu sendiri yang seharusnya serendah mungkin. Rencana-

rencana eksekusi yang diperiksa oleh planner dibandingkan berdasarkan perkiraan-perkiraan harganya dan dipilih yang perkiraan harganya.

2.3 Algoritma Genetik

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Algoritma ini didasari oleh konsep evaluasi biologi, dan dapat memberikan solusi alternatif atas suatu masalah yang hendak diselesaikan. Algoritma genetika menawarkan suatu solusi pemecahan masalah yang terbaik, dengan memanfaatkan metode seleksi, crossover, dan mutasi.

Proses pencarian solusi diawali dengan tahap pembangkitan populasi awal secara acak. Populasi ini terdiri dari kromosom-kromosom. Setiap kromosom merupakan gambaran solusi atas pemecahan masalah. Populasi yang telah dipilih tersebut akan menghasilkan keturunan baru yang sifatnya diharapkan lebih baik dari populasi sebelumnya. Populasi yang baik sifatnya akan memiliki peluang untuk terus dikembangkan agar menghasilkan keturunan populasi yang lebih baik selanjutnya. Dengan demikian, solusi terbaik yang diinginkan dapat dicapai dengan terus mengulang proses pencarian keturunan. (Kurniawan, 2008:3).

2.4 Teknik Penggunaan Algoritma Genetik

Algoritma genetik dimulai dengan sekumpulan set status yang dipilih secara random, yang disebut populasi. Algoritma ini mengkombinasikan dua populasi induk. Setiap

status atau individual direpresentasikan sebagai sebuah string. (Ariwibowo, 2008:2).

1. *Fitness function*

Setiap individual dievaluasi dengan *fitness function*. Sebuah *fitness function* mengembalikan nilai tertinggi untuk individual yang terbaik. Individu akan diurutkan berdasarkan nilai atau disebut dengan *selection*.

2. *Crossover*

Untuk setiap pasang induk, sebuah titik *crossover* akan dipilih secara random dari posisi dalam *string*. Pada gambar titik *crossover* terletak pada indeks ketiga dalam pasangan pertama dan setelah indeks kelima pada pasangan kedua.

3. Mutasi

Pada mutasi, tiap lokasi menjadi sasaran mutasi acak, dengan probabilitas independen yang kecil. Sebuah digit dimutasikan pada anak pertama, ketiga, dan keempat. Algoritma genetik mengkombinasikan suatu kecenderungan menaik dengan pengeksplorasian acak di antara *thread* pencarian paralel. Keuntungan utamanya, bila ada, datang dari operasi *crossover*. Namun, secara matematis dapat ditunjukkan bahwa bila posisi dari kode genetik di permutasikan di awal dengan urutan acak, *crossover* tidak memberikan keunggulan. Secara intuisi, keuntungannya didapat dari kemampuan *crossover* untuk menggabungkan blok-blok huruf berukuran besar yang telah

berevolusi secara independen untuk melakukan fungsi yang bermanfaat sehingga dapat menaikkan.

Flowchart program algoritma genetik seperti gambar dibawah ini.



Gambar 2. Flowchat Program Algoritma Genetik

Sedangkan *pseudo code* algoritma genetik seperti dibawah ini.

```

function
GenetikAlgorithm(population, Fitness-FN)
-> an individual
{input berupa population, sebuah
kumpulan individual dan Fitness-FN,
sebuah fungsi yang mengukur fitness
suatu individual}
deklarasi
i,x,y : integer
algoritma
repeat
new_population<-empty set
for i=1 to size(population) do
x<-RandomSelection(population,
Fitness-FN)
y<-RandomSelection(population,
Fitness-FN)
child<-Reproduce(x,y)
if(smallRandomProbability) then
child<-mutate(child)
add child to new_population
population<-new_population
until some individual is fit enough
or the time has elapsed
return the best individual in
population(based on Fitness-FN)
function Reproduce(x,y : parent
individuals)->individual
deklarasi
algoritma
n<-length(x)
c<-random number from 1 to n
return Append(substring(x,1,c),
substring(y,c +1,n))

```

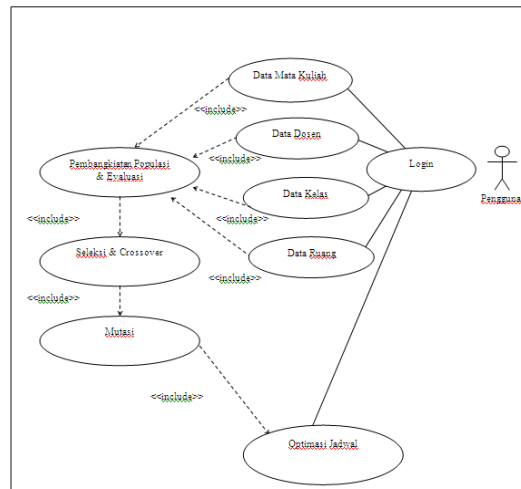
2.5 Analisis

Objek yang diteliti tentang optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang. Diharapkan dengan optimasi penjadwalan perkuliahan pada Universitas Tridinanti Palembang khususnya fakultas keguruan ilmu pendidikan, tidak ada permasalahan bentrokan jadwal pada sisi mahasiswa, serta ketersediaan ruang yang cukup dan sesuai secara fasilitas untuk seluruh mata kuliah yang ada.

2.6 Perancangan

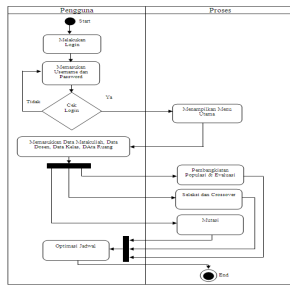
a. Use Case Diagram

Use case Diagram (UCD) menjelaskan apa yang akan dilakukan oleh optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang dan siapa yang berinteraksi dengan sistem. UCD menjadi dokumen kerja dari admin dan user, penjelasan seperti gambar dibawah ini.



Gambar 3. Use Case Diagram

b. Diagram Activity Pengguna



Gambar 4. Diagram Activity Pengguna

2.6.1 Rancangan Struktur Data

Tabel 3.1 Tabel Mata Kuliah

No	Field	Type	size
1.	Kd_matakul	Varchar	10
2.	Nm_matakul	Varchar	35
3.	SKS	Intger	4

Tabel 3.2 Tabel Dosen

No	Field	Type	Size
1.	Kd_dosen	Varchar	10
2.	Nm_dosen	Varchar	35
3.	Alamat	Varchar	50
4.	Telepon	Varchar	12

Tabel 3.4 Tabel Ruang

No	Field	Type	Size
1.	Kd_ruang	Varchar	10
2.	Ruang	Varchar	15

Tabel 3.5 Tabel Jadwal

No	Field	Type	Size
1.	Kd_dosen	Varchar	10
2.	Kd_matakul	Varchar	15
3.	Hari	Varchar	12
4.	Jam	Varchar	12
5.	Kd_kelas	Varchar	10
6.	Kd_ruang	Varchar	10

Tabel 3.6 Tabel Admin

No	Field	Type	Size
1.	Id_admin	Integer	4

2.	Username	Varchar	25
3.	Password	Varchar	25

2.6.2 Rancangan Antarmuka

a. Rancangan Form Login

Optimasi Penjadwalan
Universitas Trinanti Palembang

Username :

Password :

Copyright@2013

Gambar 3.3 Rancangan Form Login

b. Rancangan Form Menu Utama

Optimasi Penjadwalan
Universitas Trinanti Palembang

Matakuliah

Kelas

Dosen

Ruang

Optimasi Jadwal

Kelas

Copyright@2013

Gambar 3.4 Rancangan Form Menu Utama

c. Rancangan Form Matakuliah

Optimasi Penjadwalan
Universitas Trinanti Palembang

Data Matakuliah

Kode Matakuliah :

Nama Matakuliah :

SKS :

Copyright@2013

Gambar 3.5 Rancangan Form Matakuliah

d. Rancangan Form Dosen

Optimasi Penjadwalan
Universitas Tridanti Palembang

Data Dosen

Kode Dosen : xxxxxxxxxxxx
 Nama Dosen : xxxxxxxxxxxxxxxxxxxxxxxx
 Alamat : xxxxxxxxxxxxxxxxxxxxxxxx
 Telepon : xxxxxxxx

Simpan Cari Hapus Batal

Copyright@2013

Gambar 3.6 Rancangan Form Dosen

e. Rancangan Form Kelas

Optimasi Penjadwalan
Universitas Tridanti Palembang

Data Kelas

Kode Kelas : xxxxxxxxxxxx
 Kelas : xxxxxxxxxxxxxxxxxxxxxxxx

Simpan Cari Hapus Batal

Copyright@2013

Gambar 3.7 Rancangan Form Kelas

f. Rancangan Form Ruang

Optimasi Penjadwalan
Universitas Tridanti Palembang

Data Ruang

Kode Ruang : xxxxxxxxxxxx
 Ruang : xxxxxxxxxxxxxxxxxxxxxxxx

Simpan Cari Hapus Batal

Copyright@2013

Gambar 3.8 Rancangan Form Ruang

g. Rancangan Algoritma Penjadwalan

Optimasi Penjadwalan
Universitas Tridanti Palembang

Populasi Awal [Proses]

Kode Dosen	Nama Dosen	Kode Matakuliah	Matakuliah	Hari	Jam	Kelas	Ruang
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx

Evaluasi [Proses]

Kode Dosen	Nama Dosen	Kode Matakuliah	Matakuliah	Nilai Fitness
xxxx	xxxxx	xxxxx	xxxxx	9999
xxxx	xxxxx	xxxxx	xxxxx	9999

Seleksi [Proses]

Kode Dosen	Kode Matakuliah	Nilai Fitness	Parent	Kromosom
xxxx	xxxxx	9999	xxxxx	xxxxx
xxxx	xxxxx	9999	xxxxx	xxxxx

Crossover [Proses]

Kode Dosen	Nama Dosen	Kode Matakuliah	Matakuliah	Hari	Jam	Kelas	Ruang
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx

Mutasi [Proses]

Kode Dosen	Nama Dosen	Kode Matakuliah	Matakuliah	Hari	Jam	Kelas	Ruang
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx

[Optimasi Jadwal]

Gambar 3.9 Rancangan Algoritma Penjadwalan

h. Rancangan Form Optimasi Jadwal

Optimasi Penjadwalan
Universitas Tridanti Palembang

Optimasi Jadwal

Kode Dosen	Nama Dosen	Kode Matakuliah	Matakuliah	Hari	Jam	Kelas	Ruang
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx
xxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx	xxxxx

Copyright@2013

Gambar 3.10 Rancangan Form Optimasi Jadwal

2.7 Algoritma Penjadwalan

Algoritma penjadwalan pada optimasi penjadwalan perkuliahan di Universitas Tridanti Palembang berdasarkan struktur data yaitu :

```
SELECT kd_dosen, kd_matakul, hari, jam, kd_ruang from
jadwal,matakul,ruang,dosen where
jadwal.kd_matakul=matakul.kd_jadwal and jadwal.kd_ruang=kd_ruang
```

3. HASIL

3.1. Form Login

form login merupakan tampilan pertama ketika optimasi penjadwalan perkuliahan di Universitas Tridanti Palembang ditampilkan, tampilannya seperti dibawah ini.



Gambar 25. Form Login

3.2. Form Menu Utama

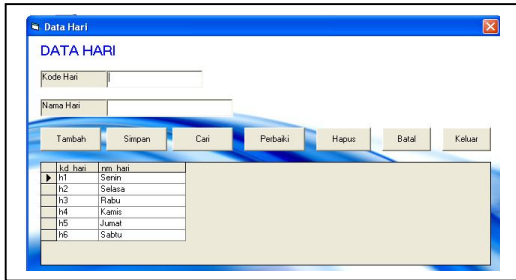
Form menu utama merupakan form induk dari optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.2 Form Menu Utama

3.3. Form Hari

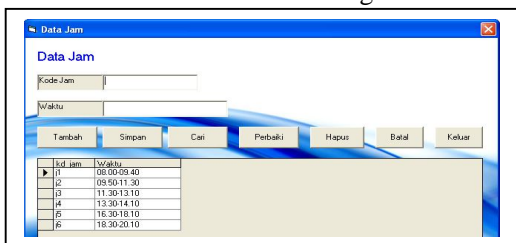
Form hari merupakan form informasi hari pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.3 Form Hari

3.4. Form Jam

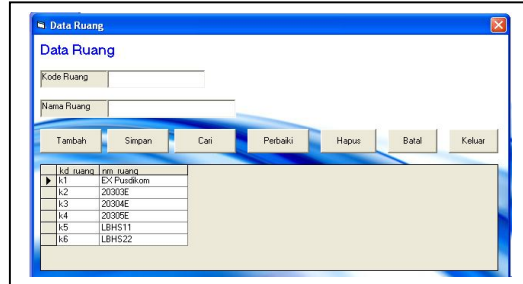
Form jam merupakan form informasi jam pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 28. Halaman Poliklinik

3.5. Form Ruang

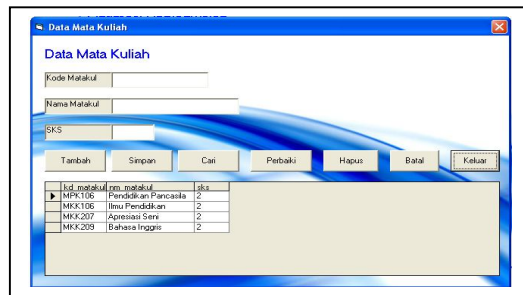
Form ruang merupakan form informasi ruang pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.5 Form Ruang

3.6. Form Mata Kuliah

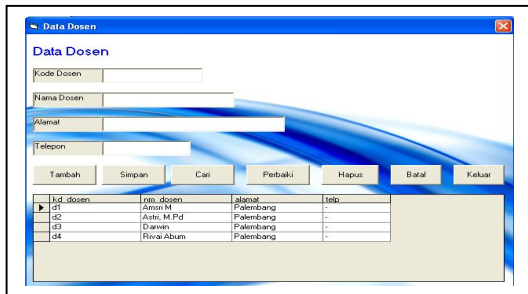
Form matakuliah merupakan form informasi matakuliah pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.6 Form Matakuliah

3.7. Form Dosen

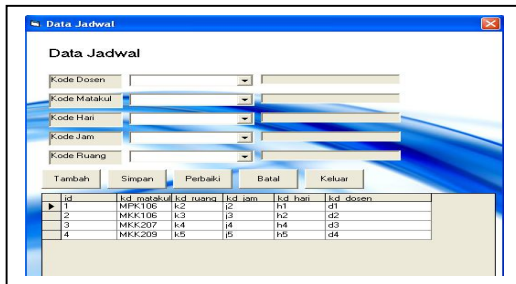
Form dosen merupakan form informasi dosen pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.7 Form Dosen

3.8. Form Jadwal

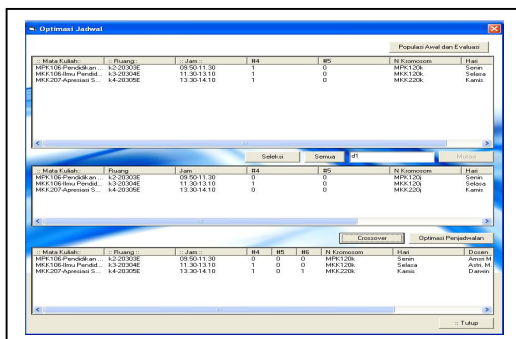
Form jadwal penjadwalan merupakan form informasi jadwal pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.8 Form Jadwal

3.9. Form Proses Jadwal

Form proses optimasi jadwal merupakan form informasi optimasi jadwal pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.9 Form Proses Optimasi Jadwal

3.10. Populasi Awal dan Evaluasi

```
try{
String sql = "select count(har_id) from hari";
ResultSet rs = kon.state.executeQuery(sql);
rs.next();
hariMax = Integer.parseInt(rs.getString(1));
sql = "select count(wkt_id) from waktu";
rs = kon.state2.executeQuery(sql);
rs.next();
jamMax = Integer.parseInt(rs.getString(1));
sql = "select * from ruang order by ru_id";
rs = kon.state2.executeQuery(sql);
rs.last();
int n = rs.getRow();
ruang = new String[n];
int j = 0;
rs.beforeFirst();
while(rs.next()){
ruang[j] = rs.getString(1);
j++;
}
}
catch(Exception e){
System.err.println("Eror ambil data : "+e);
}
}
```

3.11. Seleksi

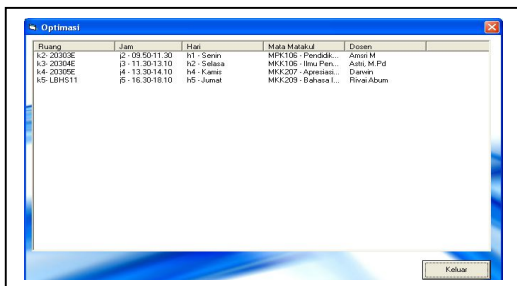
```
private int ChooseParent(){
double jmlhFit = sumFitnes();
double i = (Math.random() * (jmlhFit));
double s = 0;
int k = 0;
while (s < i){
s += Indv[k].fitnes;
k++;
}
if(k == PopulationSize){
k -= 1;
}
return k;
}
```

3.12. Crossover

```
private kromosom Crossover(int p1, int p2){
    int k,l;
    kromosom c=new kromosom(GenSize);
    int cut = (int)(Math.random()*GenSize);
    for (k = 0; k <= cut; k++){
        c.idDosen[k] = Indv[p1].idDosen[k];
        c.idHari[k] = Indv[p1].idHari[k];
        c.idJam[k] = Indv[p1].idJam[k];
        c.idKelas[k] = Indv[p1].idKelas[k];
        c.idMatkul[k] = Indv[p1].idMatkul[k];
        c.jumlah[k] = Indv[p1].jumlah[k];
        c.nmKelas[k] = Indv[p1].nmKelas[k];
        c.sks[k] = Indv[p1].sks[k];
        c.smstr[k] = Indv[p1].smstr[k];
        c.idRuang[k] = Indv[p1].idRuang[k];
    }
    for (l = cut + 1; l < GenSize; l++){
        c.idDosen[l] = Indv[p2].idDosen[l];
        c.idHari[l] = Indv[p2].idHari[l];
        c.idJam[l] = Indv[p2].idJam[l];
        c.idKelas[l] = Indv[p2].idKelas[l];
        c.idMatkul[l] = Indv[p2].idMatkul[l];
        c.jumlah[l] = Indv[p2].jumlah[l];
        c.nmKelas[l] = Indv[p2].nmKelas[l];
        c.sks[l] = Indv[p2].sks[l];
        c.smstr[l] = Indv[p2].smstr[l];
        c.idRuang[l] = Indv[p2].idRuang[l];
    }
    return c;
}
```

3.13. Form Hasil Optimasi

Form hasil optimasi jadwal merupakan form informasi optimasi jadwal pada optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.



Gambar 4.10 Form Hasil Optimasi Jadwal

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Berdasarkan dari penelitian yang telah dilaksanakan dan sudah diuraikan dalam optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang dengan algoritma genetik, maka penulis dapat menarik kesimpulan sebagai berikut :

1. Penelitian ini menghasilkan optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang.
2. Penelitian ini menghasilkan optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang dibangun menggunakan bahasa pemrograman *microsoft visual basic 6.0* dan *database MySQL*.

4.2. Saran

Saran dari optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang ini yaitu:

1. Diharapkan optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang dengan algoritma genetik r ini dapat dimanfaatkan secara optimal.
2. Seiring dengan kemajuan ilmu pengetahuan dan teknologi, maka tidak menutup kemungkinan optimasi penjadwalan perkuliahan di Universitas Tridinanti Palembang yang telah dibangun nantinya dapat dikembangkan lagi dengan bahasa pemrograman berbasis *web*.

DAFTAR RUJUKAN

- Al bahra, Bin Ladjamudin, 2005, *Analisis dan Desain Sistem Informasi*, Graha Ilmu: Yogyakarta.
- Cahyadi, Ana Nur & Erlina Idola Gadis, 2012, *Jurnal: Sistem Informasi Pengolahan Data Pasien Rawat Inap Puskesmas Pakis Baru Nawangan, Seruni FTI UNSA*.
- Nyowidanas, 2010, *Definisi Rawat Jalan*. diakses 1 Oktober 2012, dari <http://nyowidanaskm.wordpress.com/2010/08/03/definisi-rawat-jalan/>
- Pressman, Roger S. 2007. *Web EGINEERING: A Practitioner's Approach*. Mc Graw Hill: Americas.
- Riswanto, Eko. 2007. *Analisa dan Perancangan Sistem Informasi*. STMIK El Rahma: Yogyakarta.
- Shalahuddin, M dan Rosa. 2011. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Modula: Bandung.
- Sofwan, Akhmad. 2006. *Belajar MySQL dengan Phpmyadmin*. Ilmu komputer.com.
- Teguh.2001. *Pengantar PHP*. Team Training SMK-TI.
- Yulianto, dkk. 2009. *Analisa dan Perancangan*. Politeknik Telkom: Bandung.