

PEMODELAN PENJADWALAN UJIAN

LAPORAN

Diajukan untuk Memenuhi Tugas Mata Kuliah Pemodelan Matematika

oleh

10115041	Aries Yuangga
10115048	Rizki Teguh Kurniawan
10115072	Rizka Desita
10115110	Adnan Radja Maulana



JURUSAN MATEMATIKA
INSTITUT TEKNOLOGI BANDUNG

2018

KATA PENGANTAR

Puji syukur kami ucapkan kepada Tuhan Yang Maha Esa atas berkat rahmat dan karunia-Nya kami dapat menyelesaikan tugas ini tepat pada waktunya dengan hasil yang semaksimal mungkin. Tidak lupa pula kami mengucapkan terima kasih kepada pihak yang turut mendukung yaitu sebagai berikut.

1. Ibu Nuning Nuraini, M. Si., Ph. D selaku koordinator mata kuliah pemodelan matematika.
2. Bapak Fajar Yuliawan, ST., M. Si. yang telah membimbing kami dengan sabar dan membantu menyelesaikan kesulitan saat proses penyelesaian tugas ini.
3. Teman-teman yang turut membantu memberikan informasi mengenai tugas ini.
4. Pihak lainnya yang turut membantu baik itu dalam bentuk material maupun lainnya.

Pembuatan tugas ini memakan usaha dan waktu yang tidak sedikit. Berbagai kesulitan yang kelompok kami temui seperti menyusun kerangka pemodelan, mencari algoritma yang tepat, dan menyelesaikan algoritma yang digunakan namun dapat diselesaikan berkat kerja sama tim yang baik dan dukungan berbagai pihak.

Tugas ini dibuat agar dapat digunakan oleh pihak prodi matematika ITB dalam menjadwalkan ujian yang optimal dengan menggunakan periode ujian seperti beberapa prodi lainnya.

Tentunya banyak kekurangan kami dalam membuat tugas ini. Oleh karena itu, kami mengharapkan kritik dan saran yang membangun agar kami dapat menyelesaikan tugas yang lebih baik lagi di kemudian hari.

Sekian dan terima kasih.

Bandung, 10 Mei 2018

Tim penyusun

DAFTAR ISI

HALAMAN JUDUL	1
KATA PENGANTAR.....	2
DAFTAR ISI	3
ABSTRAK	4
LATAR BELAKANG	5
DATA.....	6
DATA MATA KULIAH	6
DATA RUANG KELAS	7
METODOLOGI	8
ASUMSI	8
TERMINOLOGI	8
Gen dan Kromosom.....	8
Individu	9
DESKRIPSI LANGKAH.....	9
Inisiasi Populasi	9
<i>Fitness Evaluation</i>	9
Perkawinan Silang / <i>Cross Over</i>	9
Mutasi / <i>Mutation</i>	10
FORMULASI	12
DEFINISI MASALAH.....	12
MODEL PERMASALAHAN.....	13
Fungsi Kendala.....	14
Nilai Fitness	14
HASIL DAN PEMBAHASAN	15
SIMPULAN.....	18
DAFTAR PUSTAKA	
LAMPIRAN	

ABSTRAK

Penjadwalan Ujian Jurusan Matematika ITB

Oleh : Aries Yuangga 10115041, Rizki Teguh Kurniawan 10115048, Rizka Desita 10115072, Adnan Radja Maulana 10115110.

Ujian adalah salah satu hal yang penting di dunia perkuliahan. Ujian digunakan setiap universitas untuk menguji kemampuan mahasiswa dalam memahami setiap mata kuliah. Tentulah dengan adanya ujian peran penjadwalan ujian yang optimal sangat dibutuhkan oleh setiap prodi di suatu universitas termasuk Prodi Matematika ITB. Namun kebutuhan setiap prodi tentulah berbeda-beda, sehingga timbulah pertanyaan bagaimanakah penjadwalan ujian yang optimal menurut dosen matematika dan mahasiswa matematika.

Tugas ini bertujuan untuk membuat penjadwalan ujian yang optimal sesuai kebutuhan Prodi Matematika ITB dengan menggunakan algoritma genetik. Definisi optimal yang harus dipenuhi adalah sebagai berikut.

1. Ujian menggunakan ruangan dengan selisih terkecil antara kapasitas ruangan dan banyaknya peserta ujian.
2. Ujian mata kuliah dengan tingkat yang sama diujikan pada hari yang sama dengan maksimal 2 kali ujian.
3. Mata kuliah dengan selisih 1 tingkat tidak dilaksanakan dalam waktu yang bersamaan.

Kata kunci : ujian, optimal, algoritma genetik.

LATAR BELAKANG

Di dunia perkuliahan tidak pernah lepas dari kata 'ujian'. Ujian adalah salah satu hal yang penting di dunia perkuliahan. Ujian digunakan setiap universitas untuk menguji kemampuan mahasiswa dalam memahami setiap mata kuliah. Tentulah dengan adanya ujian peran penjadwalan ujian yang optimal sangat dibutuhkan oleh setiap prodi di suatu universitas termasuk Prodi Matematika ITB.

Hal yang tidak lupa pula yaitu kebutuhan setiap prodi tentulah berbeda-beda. Terkadang mahasiswa matematika ITB mengeluh dengan jadwal ujian dikarenakan kurang memenuhi standar keinginan mereka. Banyak dari mereka yang menginginkan jadwal ujian yang lebih optimal misalnya rentang waktu ujian yang tidak terlalu berdekatan. Sedangkan dalam pembuatan jadwal itu sendiri memiliki kendala-kendala sebagai berikut.

1. Banyaknya ruangan dengan banyaknya mata kuliah.
2. Kapasitas ruangan dengan banyaknya peserta ujian.
3. Rentang waktu yang terbatas.

Perlu kita ketahui bahwa sulit untuk membuat jadwal ujian yang memenuhi kriteria peserta ujian. Oleh karena itu, hal inilah yang memotivasi kelompok kami untuk membuat jadwal ujian yang optimal yaitu jadwal yang memenuhi standar keinginan dari kedua belah pihak yaitu peserta ujian dan dosen matematika

DATA

Pada tugas ini kami menggunakan 2 buah data yaitu 24 mata kuliah dan 3 ruang kelas. Berikut adalah data yang digunakan dalam pembuatan tugas ini.

DATA MATA KULIAH

No.	Kode Mata Kuliah	Nama Mata Kuliah	Jumlah Peserta	
			K-01	K-02
1	MA2231	Kalkulus Peubah Banyak	56	55
2	MA2251	Matematika Diskrit	110	-
3	MA2252	Pengantar Teori Bilangan	20	-
4	MA2271	Pengantar Persamaan Diferensial	117	-
5	MA2281	Statistika Non Parametrik	96	-
6	MA3011	Karir dalam Matematika	125	-
7	MA3021	Struktur Aljabar	9	-
8	MA3022	Aljabar Linear	73	-
9	MA3041	Pengantar Geometri Diferensial	115	-
10	MA3231	Pengantar Analisis Real	118	-
11	MA3261	Pengantar Matematika Keuangan	21	-
12	MA3272	Metode Optimisasi	68	-
13	MA3281	Statistika Matematika	105	-
14	MA3283	Generalized Linear Models	17	-
15	MA4221	Kapita Selektif Aljabar II	6	-
16	MA4251	Teori Koding	18	-
17	MA4252	Kapita Selektif Matematika Diskrit II	2	-
18	MA4272	Kapita Selektif Mat. Terapan II	34	-

19	MA4281	Analisis Multivariat	44	-
20	MA4282	Kapita Selekta Statistika II	5	-
21	MA4283	Teori Risiko dan Kredibilitas	12	-
22	MA4285	Pengantar Mat. Asuransi Jiwa II	17	-
23	MA3271	Pemodelan Matematika	114	-

Tabel 1. Data mata kuliah

DATA RUANG KELAS

No.	Nama Ruangan	Kapasitas Ruangan
1	Labtek 1 – 9403	130
2	Labtek 1 – 9404	60
3	TVST – 9023	108

Tabel 2. Data ruang kelas

METODOLOGI

ASUMSI

Untuk membatasi cakupan masalah. Kami menggunakan beberapa asumsi-asumsi dalam memodelkan penjadwalan ujian ini. Asumsi-asumsi yang kami gunakan antara lain :

1. Ujian dilaksanakan di luar periode kuliah.
Asumsi ini diterapkan karena jika ujian dilaksanakan pada periode kuliah atau dengan kata lain ujian dilaksanakan pada jadwal dan ruangan kuliah yang sudah ditetapkan maka masalah selesai.
2. Waktu yang disediakan untuk ujian adalah hari senin s/d hari jumat dari pukul 07.00 s/d 17.00
3. Mata ujian dengan selisih tingkat ≤ 1 tidak dilaksanakan bersamaan.
4. Dalam satu tingkat terdapat maksimal 2 ujian dalam satu hari.
5. Seluruh peserta dari satu mata ujian yang sama melaksanakan ujian secara bersamaan.

TERMINOLOGI

Metode yang digunakan adalah metode genetik. Berikut istilah metode genetik yang digunakan dalam pemodelan.

Gen dan Kromosom

Kromosom merupakan subjadwal yang memuat waktu dan ruangan untuk satu mata kuliah.

MA2231	9403	07.00- 09.00
--------	------	-----------------

Tabel 3. Contoh kromosom

Tabel 3 menyatakan bahwa mata ujian MA2231 Kalkulus Peubah Banyak dilaksanakan di ruangan 9403 pada pukul 07.00-09.00.

Sedangkan gen adalah informasi yang menyusun suatu kromosom. Pada contoh diatas, MA2231, 9403, dan 07.00-09.00 adalah gen.

Individu

Individu atau jadwal utuh adalah himpunan kromosom yang memuat jadwal semua mata ujian.

DESKRIPSI LANGKAH

Penelitian ini terdiri dari beberapa tahap, yaitu :

Inisiasi Populasi

Untuk dapat memulai bekerja menggunakan algoritma genetik, diperlukan suatu populasi yang terdiri dari beberapa individu. Selanjutnya memilih 1 individu dari populasi tersebut. Individu yang dipilih ini dibuat sehingga tidak ada satupun mata kuliah yang memiliki jadwal pada waktu yang sama. Kemudian membuat 2 individu dengan mengacak ruangan dari individu tersebut guna membentuk individu baru yang memenuhi kriteria yang diinginkan.

Fitness Evaluation

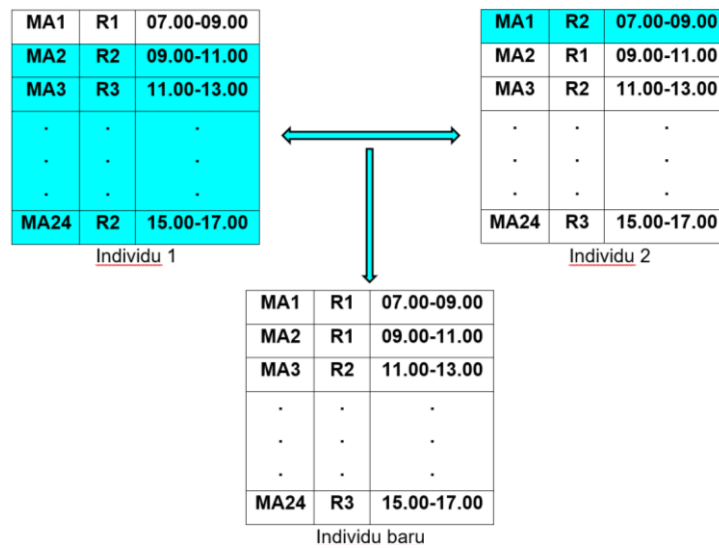
Untuk menilai kelayakan dan efisien atau tidaknya suatu solusi perlu adanya parameter penilaian. Setiap individu dihitung nilai fitnessnya agar dapat dipilih individu-individu terbaik dalam suatu populasi.

Perkawinan Silang / Cross Over

Perkawinan silang antara 2 individu yang didapat dari hasil acak saat inisiasi populasi dengan proses *looping* guna menghasilkan individu baru yang memiliki nilai fitness yang lebih baik atau jadwal ujian yang dicari. Perkawinan silang dapat dilakukan jika memenuhi peluang terjadinya perkawinan silang yaitu 0.5.

Mekanisme dari perkawinan silang yaitu sebagai berikut.

1. Dari 2 individu yang dimiliki, 1 kromosom pada individu 1 akan dipertahankan, namun kromosom 2 sampai kromosom 24 ditukar dengan yang dimiliki individu 2.
2. Selanjutnya, 2 kromosom pada individu 1 akan dipertahankan, namun kromosom 3 sampai kromosom 24 ditukar dengan yang dimiliki individu 2. Begitu seterusnya dan berlaku hal yang sama terhadap individu 2.



Gambar 1. Contoh pembentukan individu baru melalui perkawinan silang (kromosom biru sebagai kromosom yang dibuang)

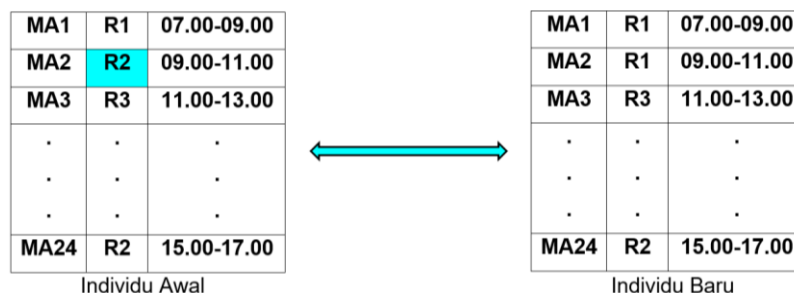
2. Dari individu baru yang terbentuk melalui proses perkawinan silang, kita pilih 1 individu yang memiliki nilai fitness terbaik untuk menggantikan posisi salah satu individu yang didapat dari inisiasi populasi.

Mutasi / Mutation

Mutasi adalah penukaran elemen kromosom pada satu individu. Mutasi terjadi jika peluang terjadinya perkawinan silang tidak terpenuhi dan terjadi di salah satu individu yang dipilih secara acak. Mutasi terbagi menjadi 2, yaitu sebagai berikut.

1. Mutasi Acak / *Random*

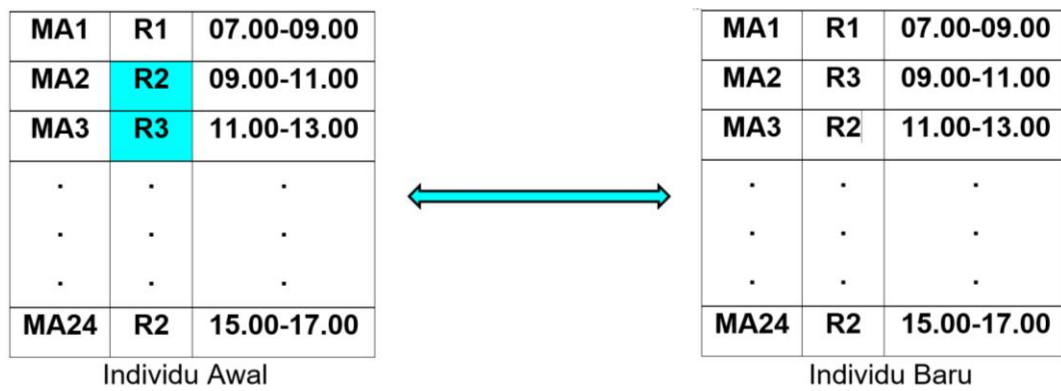
Menukar gen ruangan secara langsung dengan sembarang ruangan yang berbeda.



Gambar 2. Contoh pembentukan individu baru melalui mutasi acak

2. Mutasi Penukaran / Swap

Memilih 2 gen pada kromosom yang akan ditukar, lalu ditukarkanlah kedua gen tersebut.



Gambar 3. Contoh pembentukan individu baru melalui mutasi swap

FORMULASI

DEFINISI MASALAH

Masalah utama dalam membuat penjadwalan ujian adalah sebagai berikut.

1. *Timeslot* / Waktu

Dinotasikan sebagai T .

$$T = \{1, 2, 3, \dots, n\}$$

Keterangan :

$Hari(t)$ menunjukkan hari *timeslot*.

$Jam(t)$ menunjukkan jam *timeslot*.

$$n = 25$$

Berikut matriks T .

T	1	2	3	4	5	6	...	n
$Jam(t)$	1	2	3	4	5	1	...	If $i \bmod 5 = 0, jam(t_i) = 5$ else $jam(t_i) = i \bmod 5$
$Hari(t)$	1	1	1	1	1	2	...	$Hari(t_i) = (i \div 5) + 1$

Tabel 4. Matriks T

$Jam(t)$	1	2	3	4	5
Nilai	07.00 - 09.00	09.00 – 11.00	11.00 – 13.00	13.00 – 15.00	15.00 – 17.00

Tabel 5. Periode waktu yang digunakan

2. Ujian

Dinotasikan sebagai U .

$$U = \{1, 2, 3, \dots, k\}$$

Keterangan :

$Nama(u)$ menunjukkan nama mata kuliah.

$Kode(u)$ menunjukkan kode mata kuliah.

$Tingkat(u)$ menunjukkan tingkat mata kuliah.

$Peserta(u)$ menunjukkan banyaknya peserta mata kuliah.

$$k = 24$$

Berikut matriks U .

u	1	2	3	...	k
$Nama(u)$	Kalkulus Peubah Banyak	Aljabar Linear	Metode Optimisasi	...	Teori Koding
$Kode(u)$	2231	3022	3272	...	4251
$Tingkat(u)$	2	3	3	...	4
$Peserta(u)$	111	73	69	...	18

Tabel 6. Matriks U

3. Ruang

Dinotasikan sebagai R .

$$R = \{1, 2, 3, \dots, m\}$$

Keterangan :

$Ruang(r)$ menunjukkan nama ruangan.

$Kapasitas(r)$ menunjukkan kapasitas ruangan.

$$m = 3$$

Berikut matriks R .

r	1	2	3	...	m
$Ruang(r)$	TVST C	Oktagon 9021	Labtek 1 9403	...	GKU Timur 9234
$Kapasitas(r)$	225	218	130	...	60

Tabel 7. Matriks R

MODEL PERMASALAHAN

Mencari pemetaan dari

$$S : U \rightarrow T$$

$$C : U \rightarrow R$$

Dengan memenuhi beberapa kendala.

Fungsi Kendala

1. Ruang harus bisa mengakomodasi banyaknya peserta yang menempati ruangan tersebut.

$$Kapasitas(C(u)) \geq Peserta(u), \forall u \in U$$

2. Satu ruangan hanya bisa mengakomodasi satu ujian pada satu waktu.

$$[C(a) = C(b)] \wedge [S(a) = S(b)] \Leftrightarrow a = b, \text{ untuk suatu } a, b \in U$$

3. Untuk mata kuliah di tingkat yang sama dan berbeda satu tingkat tidak boleh dilaksanakan di waktu yang sama.

$$|Tingkat(a) - Tingkat(b)| \leq 1 \Rightarrow S(a) \neq S(b), \text{ untuk suatu } a, b \in U$$

4. Dalam satu hari paling banyak terdapat dua ujian dengan tingkat yang sama.

$$|\{a, b \in U \mid Tingkat(a) = k, Hari(S(a)) = Hari(S(b)), a \neq b, Tingkat(b) = k\}| \leq 2, \text{ dengan } k = \{2, 3, 4\}$$

Nilai Fitness

Individu atau jadwal ujian yang baik adalah memenuhi nilai fitness yang terbaik.

Berikut nilai fitness yang harus dipenuhi.

$$\min \left(\sum_{i=1}^k (Kapasitas(C(i)) - Peserta(i))^2 \right), \forall i \in U$$

HASIL DAN PEMBAHASAN

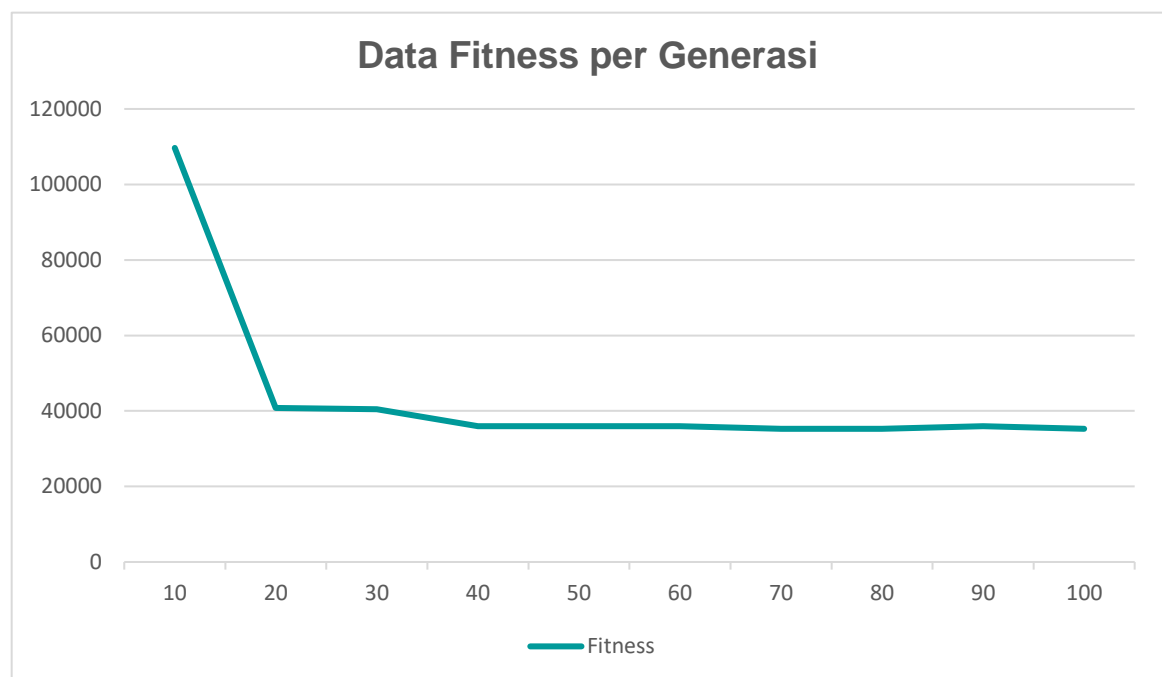
A. Berikut adalah jadwal yang berhasil dibangun oleh program sebanyak 100 generasi

Nama Matakuliah	Kode	Waktu	Hari	Ruangan	Kapasitas	Peserta
Tingkat 2						
Kalkulus Peubah Banyak Genap	201	13.00 - 15.00	Kamis	9403	130	56
Kalkulus Peubah Banyak Ganjil	201	13.00 - 15.00	Kamis	9023	108	56
Matematika Diskrit	202	09.00 - 11.00	Senin	9403	130	110
Pengantar Teori Bilangan	203	09.00 - 11.00	Kamis	9404	60	20
Pengantar Persamaan Diferensial	204	15.00 - 17.00	Senin	9403	130	117
Statistika non Parametrik	205	15.00 - 17.00	Selasa	9023	108	96
Tingkat 3						
Karir dalam Matematika	301	07.00 - 09.00	Jumat	9403	130	125
Struktur Aljabar	302	09.00 - 11.00	Selasa	9404	60	9
Aljabar Linear	303	11.00 - 13.00	Rabu	9023	108	73
Pengantar Geometri Diferensial	304	11.00 - 13.00	Jumat	9403	130	115
Pengantar Analisis Real	305	13.00 - 15.00	Selasa	9403	130	118
Pengantar Matematika Keuangan	306	11.00 - 13.00	Kamis	9404	60	21
Pemodelan Matematika	307	07.00 - 09.00	Rabu	9403	130	114
Metode Optimisasi	308	11.00 - 13.00	Senin	9023	108	69
Statistika Matematika	309	07.00 - 09.00	Senin	9023	108	106
Generalized Linear Models	310	07.00 - 09.00	Kamis	9404	60	17

Tingkat 4						
Kapita Selekt Aljabar II	405	07.00 - 09.00	Selasa	9404	60	6
Teori Koding	406	15.00 - 17.00	Rabu	9404	60	18
Kapita Selekt Matematika Diskrit II	407	15.00 - 17.00	Kamis	9404	60	2
Kapita Selekt Matematika Terapan II	408	15.00 - 17.00	Senin	9404	60	34
Analisis Multivariat	409	09.00 - 11.00	Jumat	9404	60	44
Kapita Selekt Statistika II	410	13.00 - 15.00	Kamis	9404	60	5
Teori Risiko dan Kredibilitas	411	13.00 - 15.00	Senin	9404	60	12
Pengantar Matematika Asuransi Jiwa II	412	15.00 - 17.00	Selasa	9404	60	18

Tabel 8. Output jadwal ujian

B. Dalam grafik berikut diperoleh nilai fitness yang konvergenk



Grafik 1. Data fitness per generasi

Grafik ini diperoleh dengan memetakan nilai generasi dengan nilai fitness yang berkorespondensi. Terlihat bahwa nilai fitness konvergen pada

suatu nilai. Hal ini menunjukkan bahwa semakin tinggi nilai generasi akan diperoleh solusi yang semakin optimal. Batas atas dan batas bawah nilai fitness diperoleh dari variasi kapasitas ruangan. Jika ingin diperoleh nilai fitness yang sangat baik yaitu mendekati nol maka diperlukan ruangan dengan kapasitas mendekati banyaknya peserta ujian.

Generasi	Fitness
10	109671
20	40764
30	40457
40	35933
50	35933
60	35933
70	35273
80	35273
90	35933
100	35273

Tabel 9. Data fitness

SIMPULAN

Pada penelitian ini telah dibuat model genetik untuk masalah penjadwalan ujian. Algoritma genetik dapat digunakan sebagai alternatif solusi untuk menyelesaikan masalah penjadwalan ujian. Jadwal ujian diperoleh dari individu yang memiliki nilai fitness terbaik.

Pada kasus penjadwalan ujian nilai probabilitas perkawinan silang yang digunakan adalah 0.5.

DAFTAR PUSTAKA

<https://www.scribd.com/doc/240711741/Makalah-TTKI>

<https://www.komentarmu.com/contoh-abstrak/>

<https://akademik.itb.ac.id/app/mahasiswa:10115072+2017-2/kelas/jadwal/kuliah/list>

<https://ditsp.itb.ac.id/daftar-data/#>

<http://wayanfm.lecture.ub.ac.id/files/2014/03/200607-Kursor-Nia-Wayan-GA-Jadwal-Ujian.pdf>

<https://ferko.fer.hr/people/Marko.Cupic/files/2009-422047.iti2009.pdf>

LAMPIRAN

Algoritma genetik untuk penjadwalan ujian dengan menggunakan matlab 2017.

```
clear
clc
kuliah = xlsread('Datareal.xlsx', 1);
ruangan = xlsread('Datareal.xlsx', 3);
timeslot = xlsread('datareal.xlsx', 2);

%Generate Population%
banru = 3;
krom = zeros(24,5);

for i = 1 : 24

    %kode%
    krom(i,1) = kuliah(i,1);

    %tingkat%
    krom(i,2) = kuliah(i,2);

    %timeslot%
    %cek kelas paralel%
    j = 1;
    ada = 0;
    while j < i && ada == 0
        if krom(i,1) == krom(j,1)
            krom(i,3) = krom(j,3);
            krom(i,4) = krom(j,4);
            ada = 1;
        else
            j = j + 1;
        end
    end
    %jika belum ada jadwal%
    if ada == 0
        ujiansehari = 1;
        bentrok = 1;
        utem = 1;
        while bentrok == 1 || ujiansehari == 1 || utem == 1
            %random timeslot%
            krom(i,3) = randi(24);
            %hari timeslot%
            if mod(krom(i,3), 5) == 0
                krom(i,4) = (krom(i,3) / 5);
            else
                krom(i,4) = floor(krom(i,3)/5) + 1;
            end

            j = 1;
            stop = 0;
            while stop == 0 && j < i
```

```

        if abs(krom(i,2) - krom(j,2)) < 2
            if krom(i,3) == krom(j,3)
                stop = 1;
            else
                j = j + 1;
            end
        else
            j = j + 1;
            stop = 0;
        end
    end
    if stop == 1
        bentrok = 1;
    else
        bentrok = 0;
    end

    j = 1;
    n = 0;
    while n < 2 && j < i
        if krom(i,4) == krom(j,4)
            if krom(i,2) == krom(j,2)
                n = n + 1;
            end
        end
        j = j + 1;
    end
    if n < 2
        ujiansehari = 0;
    else
        ujiansehari = 1;
    end

    %ujian dalam timeslot dibatasi hanya 4 ujian dalam 1 timeslot%
    n = 0;
    j = 1;
    while n < 4 && j < i
        if krom(i,3) == krom(j,3)
            n = n + 1;
        end
        j = j + 1;
    end

    if n < 5
        utem = 0;
    else
        utem = 1;
    end
end
end
end

%Generate Room%

%Matriks Ruangan%

parent = zeros(3,24,2);
child = zeros(3,24,1);
fites = zeros(2,1);
i = 1;

```

```

while i < 3
    j = 1;
    while j < 25
        parent(1,j,i) = kuliah(j,3);
        parent(2,j,i) = randi(banru);
        parent(3,j,i) = ruangan(parent(2,j,i),2);
        j = j + 1;
    end
    sum = 0;
    for k = 1 : 24
        if (parent(3,k,i) - parent(1,k,i) < 0)
            sum = sum + (parent(1,k,i))^2;
        else
            m = 1;
            bentrok = 0;
            while m < k && bentrok == 0
                if (parent(2,m,i) == parent(2,k,i)) && (krom(m,3) ==
krom(k,3))
                    bentrok = 1;
                else
                    bentrok = 0;
                end
                m = m + 1;
            end
            if bentrok == 1
                sum = sum + parent(3,k,i)^3;
            else
                sum = sum + (parent(3,k,i) - parent(1,k,i))^2;
            end
        end
    end
    fites(i,1) = sum;
    i = i + 1;
end
temppar = zeros(3,24);
if fites(1,1) > fites(2,1)
    temppar(:, :) = parent(:, :, 1);
    parent(:, :, 1) = parent(:, :, 2);
    parent(:, :, 2) = temppar(:, :);
    fitemp = fites(1,1);
    fites(1,1) = fites(2,1);
    fites(2,1) = fitemp;
end

%crossover and mutation
generasi = 100;
count = 1;
fitness = 99999999;
x = 0.5;
y = 0.3;
z = 0.5;
anakbaik = zeros(3,24,1);
while (count < generasi)
    j = 1;
    while j <= 46
        prob = rand;
        if prob > x
            %crossover%
            if j <= 22
                for l = 1 : j
                    child(:,l,1) = parent(:,l,1);

```

```

        end
        for k = j + 1 : 24
            child(:,k,1) = parent(:,k,2);
        end
    else if j > 23
        for l = 1 : j - 24
            child(:,l,1) = parent(:,l,2);
        end
        for k = j - 23 : 24
            child(:,k,1) = parent(:,k,1);
        end
    end
    end
    anak = 1;
    %end Crossover
else
    %mutasi random%
    if prob > y
        prob = rand;
        idx = randi(24);
        idx1 = randi(2);
        child(:, :, 1) = parent(:, :, idx1);
        child(2, idx, 1) = randi(banru);
        child(3, idx, 1) = ruangan(child(2, idx, 1), 2);
        anak = 1;
    else
        %mutasi swap%
        if prob > y/2
            idx = randi(24);
            idx1 = randi(2);
            hm = mod(j, 24);
            if hm == 0
                hm = 24;
            end
            child(:, :, 1) = parent(:, :, idx1);
            child(2, idx, 1) = parent(2, hm, idx1);
            child(3, idx, 1) = parent(3, hm, idx1);
            child(2, hm, 1) = parent(2, idx, idx1);
            child(3, hm, 1) = parent(3, idx, idx1);
            anak = 1;
        else
            anak = 0;
        end
    end
end

if anak == 1
    %fitness evaluation anak%
    sum = 0;
    for k = 1 : 24
        if (child(3, k, 1) - child(1, k, 1) < 0)
            sum = sum + child(3, k, 1)^2;
        else
            m = 1;
            bentrok = 0;
            while m < k && bentrok == 0
                if (child(2, m, 1) == child(2, k, 1)) && (krom(m, 3) ==
krom(k, 3))
                    bentrok = 1;
                else
                    bentrok = 0;
                end
            end
        end
    end
end

```

```

        end
        m = m + 1;
    end
    if bentrok == 1
        sum = sum + child(3,k,1)^3;
    else
        sum = sum + (child(3,k,1) - child(1,k,1))^2;
    end
end
end
fit = sum;
if fit < fitness
    fitness = fit;
    anakbaik(:, :, 1) = child(:, :, 1);
end
end
j = j + 1;
end
%anak jadi parent
if fitness < fites(2,1)
    parent(:, :, 2) = anakbaik(:, :, 1);
    fites(2,1) = fitness;
end
%haruslah parent 1 < parent 2%
temppar = zeros(3,24);
if fites(1,1) > fites(2,1)
    temppar(:, :) = parent(:, :, 1);
    parent(:, :, 1) = parent(:, :, 2);
    parent(:, :, 2) = temppar(:, :);
    fitemp = fites(1,1);
    fites(1,1) = fites(2,1);
    fites(2,1) = fitemp;
end
count = count + 1;
end

krom(:,5) = parent(2, :, 1);
%Output All%
tingkat2 = zeros(6,7);

tingkat3 = zeros(10,7);

tingkat4 = zeros(8,7);

j = 1;
k = 1;
l = 1;
for i = 1 : 24
    if kuliah(i,1) < 300
        tingkat2(j,1) = kuliah(i,1);
        tingkat2(j,2) = ruangan(parent(2,i,1),1);
        tingkat2(j,3) = krom(i,5);
        tingkat2(j,4) = kuliah(i,3);
        tingkat2(j,5) = ruangan(krom(i,5),2);
        tingkat2(j,6) = mod(krom(i,3),5);
        tingkat2(j,7) = krom(i,4);
        j = j + 1;
    else if kuliah(i,1) < 400
        tingkat3(k,1) = kuliah(i,1);
        tingkat3(k,2) = ruangan(parent(2,i,1),1);

```



```

        tingkat3(k,3) = krom(i,5);
        tingkat3(k,4) = kuliah(i,3);
        tingkat3(k,5) = ruangan(krom(i,5),2);
        tingkat3(k,6) = mod(krom(i,3),5);
        tingkat3(k,7) = krom(i,4);
        k = k + 1;
    else
        tingkat4(l,1) = kuliah(i,1);
        tingkat4(l,2) = ruangan(parent(2,i,1),1);
        tingkat4(l,3) = krom(i,5);
        tingkat4(l,4) = kuliah(i,3);
        tingkat4(l,5) = ruangan(krom(i,5),2);
        tingkat4(l,6) = mod(krom(i,3),5);
        tingkat4(l,7) = krom(i,4);
        l = l + 1;
    end
end
end
filename = 'outputjadwal.xlsx';
sheet = 1;
xlRange = 'C38';
xlswrite(filename,tingkat2(:,6),sheet,xlRange);
xlRange = 'D38';
xlswrite(filename,tingkat2(:,7),sheet,xlRange);
xlRange = 'E38';
xlswrite(filename,tingkat2(:,2),sheet,xlRange);
xlRange = 'F38';
xlswrite(filename,tingkat2(:,5),sheet,xlRange);
xlRange = 'G38';
xlswrite(filename,tingkat2(:,4),sheet,xlRange);

xlRange = 'B46';
xlswrite(filename,tingkat3(:,6),sheet,xlRange);
xlRange = 'C46';
xlswrite(filename,tingkat3(:,7),sheet,xlRange);
xlRange = 'E45';
xlswrite(filename,tingkat3(:,2),sheet,xlRange);
xlRange = 'F45';
xlswrite(filename,tingkat3(:,5),sheet,xlRange);
xlRange = 'G45';
xlswrite(filename,tingkat3(:,4),sheet,xlRange);

xlRange = 'B58';
xlswrite(filename,tingkat4(:,6),sheet,xlRange);
xlRange = 'C58';
xlswrite(filename,tingkat4(:,7),sheet,xlRange);
xlRange = 'E57';
xlswrite(filename,tingkat4(:,2),sheet,xlRange);
xlRange = 'F57';
xlswrite(filename,tingkat4(:,5),sheet,xlRange);
xlRange = 'G57';
xlswrite(filename,tingkat4(:,4),sheet,xlRange);

%Output 9403%
japrem = zeros(5,5);
for i = 1 : 24
    if i ~= 24 && i ~= 23 && krom(i,5) == 1
        tss = mod(krom(i,3), 5);
        if tss == 0
            tss = 5;
        end
    end
end

```

```

        japrem(tss,krom(i,4)) = krom(i,1);
    else if i == 24 && krom(i,5) == 1
        tss = 5;
        japrem(tss,krom(i,4)) = krom(i,1);
    else if i == 23 && krom(i,5) == 1
        tss = 4;
        japrem(tss,krom(i,4)) = krom(i,1);
    end
end
end
end
sheet = 2;
xlRange = 'B16';
xlswrite(filename,japrem,sheet,xlRange);

%Output 9404%
japrem = zeros(5,5);
for i = 1 : 24
    if i ~= 24 && i ~= 23 && krom(i,5) == 2
        tss = mod(krom(i,3), 5);
        if tss == 0
            tss = 5;
        end
        japrem(tss,krom(i,4)) = krom(i,1);
    else if i == 24 && krom(i,5) == 2
        tss = 5;
        japrem(tss,krom(i,4)) = krom(i,1);
    else if i == 23 && krom(i,5) == 2
        tss = 4;
        japrem(tss,krom(i,4)) = krom(i,1);
    end
end
end
end
sheet = 3;
xlRange = 'B16';
xlswrite(filename,japrem,sheet,xlRange);

%Output 9023%
japrem = zeros(5,5);
for i = 1 : 24
    if i ~= 24 && i ~= 23 && krom(i,5) == 3
        tss = mod(krom(i,3), 5);
        if tss == 0
            tss = 5;
        end
        japrem(tss,krom(i,4)) = krom(i,1);
    else if i == 24 && krom(i,5) == 3
        tss = 5;
        japrem(tss,krom(i,4)) = krom(i,1);
    else if i == 23 && krom(i,5) == 3
        tss = 4;
        japrem(tss,krom(i,4)) = krom(i,1);
    end
end
end
end
sheet = 4;
xlRange = 'B16';
xlswrite(filename,japrem,sheet,xlRange);

```