

LAPORAN PROYEK

SISTEM PENJADWALAN MATA KULIAH MENGGUNAKAN ALGORITMA GENETIKA



Disusun oleh:

Anggota Kelompok : 12S18001 – Cindy Angelia Siregar
12S18004 – Rosalia Pane
12S18017 – Putri Yohana Panjaitan
12S18030 – Unedo Ignasius Manalu
12S18035 – Angelina Naomi Christina Sinaga
12S18063 – Calvin Adelbert Lumban Gaol

Kode - Mata Kuliah	:	10S3001 – Kecerdasan Buatan
Demo Video	:	https://youtu.be/1Hr_OgTkGGs
GitHub Link	:	https://github.com/angelinasinaga/sistem-penjadwalan-mata-kuliah-menggunakan-algoritma-genetika

Fakultas Informatika dan Teknik Elektro

Institut Teknologi Del

2020

DAFTAR ISI

DAFTAR ISI	2
DAFTAR GAMBAR.....	3
BAB I PENDAHULUAN.....	5
1.1 Latar Belakang	5
1.2 Tujuan.....	6
1.3 Manfaat.....	6
1.4 Ruang Lingkup	6
BAB II LANDASAN TEORI	7
2.1 Jadwal.....	7
2.2 Local Search	7
2.3 Genetic Algorithm.....	8
BAB III DESAIN	10
3.1 Desain Data	10
3.1.1 Desain Data Kelas	10
3.1.2 Desain Data Dosen	10
3.1.3 Desain Data Courses Class	11
3.1.4 Desain Data Room	11
3.1.5 Desain Data Schedule	12
3.2 Prosedur Pengerjaan.....	13
BAB IV HASIL DAN PEMBAHASAN	16
4.1 Hasil	16
4.2 Pembahasan	16
BAB V PENUTUP	17
5.1 Kesimpulan	17
5.2 Saran.....	17
5.3 Pembagian Tugas	17
BAB VI DAFTAR PUSTAKA	19

DAFTAR GAMBAR

Gambar 1. Desain Data Kelas.....	10
Gambar 2. Desain Data Dosen	11
Gambar 3. Desain Data Courses Class.....	11
Gambar 4. Desain Data Room	12
Gambar 5. Desain Data Schedule.....	12
Gambar 6. Library pada Data.....	13
Gambar 7. Data yang diolah	13
Gambar 8. fungsi faculty_member_one_class dan Kelas_member_one_class	14
Gambar 9. Fungsi simulated annealing	14
Gambar 10. Algoritma genetic algorithm	15
Gambar 11. Pemanggilan fungsi main	15
Gambar 12. Output	16

DAFTAR TABEL

Table 1. Pembagian Tugas	17
--------------------------------	----

BAB I

PENDAHULUAN

Bab pendahuluan berisi deskripsi umum dari kajian yang dikerjakan. Deskripsi ini mencakup latar belakang proyek penelitian, pertanyaan proyek penelitian, tujuan penelitian, manfaat proyek penelitian dan ruang lingkup pada proyek penelitian.

1.1 Latar Belakang

Jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Proses belajar mengajar pada sebuah universitas tak lepas dari penyusunan jadwal akademik. Semakin bagus sistem penjadwalan, maka proses belajar mengajar juga akan menjadi semakin optimal dan teratur. Pengaturan jadwal akademik pada universitas merupakan hal yang penting karena dalam pelaksanaannya melibatkan banyak aspek yang menjadi bahan pertimbangan dalam penyusunan, diantaranya adalah jumlah mata kuliah yang akan dijadwalkan, jumlah dosen yang tersedia, jumlah keseluruhan mahasiswa, waktu, ruangan, dan lainnya. Faktor lain yang bisa menjadi kendala dalam penyusunan jadwal akademik ialah, ketidakseimbangan antara dosen yang mengajar dengan mata kuliah yang akan diajarkan (keterbatasan dosen), ruangan kelas yang terbatas, jadwal yang bentrok (jam mengajar ataupun ruangan), waktu yang dimiliki dosen, dan hal lain yang dapat menjadi pemicu kesalahan dalam penyusunan jadwal akademik.

Penjadwalan mata kuliah merupakan salah satu masalah yang harus dihadapi di setiap pergantian semester. Penjadwalan akademik pada beberapa universitas masih dibuat dengan sistem manual, yaitu dengan mendata setiap mata kuliah yang tersedia, kemudian mencatat setiap mata kuliah yang menjadi prioritas terhadap kelas tertentu jika ada, kemudian melakukan proses penjadwalan dan membuat laporan hasil jadwal menggunakan Microsoft Excel. Dengan cara seperti ini, akan memakan waktu yang lama untuk mengeluarkan sebuah jadwal yang sempurna (tidak memerlukan perbaikan). Oleh karena itu, perlu dikembangkan suatu sistem penjadwalan mata kuliah yang dapat mengkoordinir berbagai aspek dengan mengimplementasikan algoritma genetika. Pemilihan algoritma tersebut dikarenakan algoritma genetika merupakan bentuk tipe *evolution algorithm* (EA) yang populer karena kemampuannya dalam menyelesaikan berbagai masalah kompleks. Selain itu, algoritma genetika merupakan pencarian secara random namun mengarah pada solusi yang lebih baik.

1.2 Tujuan

Adapun tujuan dari pengerjaan proyek ini adalah untuk melakukan pengimplementasian algoritma genetika untuk memastikan algoritma tersebut dapat menentukan penjadwalan mata kuliah.

1.3 Manfaat

Adapun manfaat dari pengerjaan proyek ini yaitu untuk menghasilkan suatu sistem penjadwalan mata kuliah yang optimal, sehingga mempercepat proses pembuatan penjadwalan mata kuliah yang memenuhi kondisi dimana terjadi kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan, agar terhindarnya kelas yang bentrok, serta ketersediaan ruangan yang cukup dan sesuai secara fasilitas untuk seluruh mata kuliah yang ada.

1.4 Ruang Lingkup

Adapun ruang lingkup proyek ini yaitu:

1. Penggunaan data mata kuliah yang diambil dari CIS (*Campus Information System*) Institut Teknologi Del.
2. Penggunaan data Dosen yang diambil dari CIS (*Campus Information System*) Institut Teknologi Del.
3. Penggunaan data mahasiswa yang diambil dari CIS (*Campus Information System*) Institut Teknologi Del.
4. Variabel yang akan digunakan dalam penentuan jadwal mata kuliah yaitu lokasi, kelas, waktu mulai, waktu selesai.

BAB II

LANDASAN TEORI

Pada bab ini dijelaskan mengenai teori-teori relevan yang berkaitan dengan pengerjaan proyek penelitian yang dilakukan seperti objek kajian, metode, dan algoritma yang digunakan dalam pengerjaan proyek.

2.1 Jadwal

Jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci [1]. Penjadwalan merupakan salah satu kegiatan yang penting dalam terlaksananya proses pembelajaran yang baik bagi sebuah universitas atau perguruan tinggi. Sebuah penjadwalan mata kuliah yang baik merupakan jadwal yang dapat dilakukan oleh semua pihak yang terlibat dalam kegiatan perkuliahan, bukan hanya dosen yang mengajar, tetapi juga bagi mahasiswa yang mengambil mata kuliah tersebut.

Adapun faktor-faktor yang mempengaruhi pembuatan jadwal di Institut Teknologi Del ialah:

1. Lokasi
2. Kelas
3. Waktu mulai
4. Waktu selesai
5. Sumber daya
6. Kode mata kuliah

2.2 Local Search

Local search menguntungkan dalam pengoptimalan murni *object function* untuk menemukan *best state* menurut fungsi pengoptimalan dengan gagasan mempertahankan *current state* dan coba untuk meningkatkannya. Local search disebut perpindahan dari satu keadaan ke keadaan berikutnya yang dapat dilakukan dengan dua strategi, yaitu *Hill-Climbing Search* dan *Genetic Algorithm*. Penggunaan metode *local search* dalam pengerjaan proyek ini dikarenakan tidak perlu pemeliharaan *search tree*, menggunakan sedikit memori, dan sering ditemukan solusi yang cukup baik itu dalam ruang yang besar.

2.3 Genetic Algorithm

Algoritma genetika adalah suatu algoritma optimasi yang bermanfaat untuk mencari solusi optimasi dalam menyelesaikan suatu permasalahan dengan memaksimalkan atau meminimalkan suatu fungsi. Pada awalnya algoritma genetika memang digunakan sebagai algoritma pencarian parameter-parameter optimal. Namun dalam perkembangannya, algoritma genetika bisa diaplikasikan untuk berbagai masalah lain seperti pembelajaran, peramalan, pemrograman otomatis, dan sebagainya. Pada bidang *soft computing*, algoritma genetika banyak digunakan untuk mendapatkan nilai-nilai parameter yang optimal pada jaringan syaraf tiruan maupun sistem fuzzy.

Algoritma Genetika mulai melakukan pencarian solusi dengan melakukan inisialisasi populasi terlebih dahulu. Selanjutnya dilakukan proses evaluasi dan seleksi terhadap populasi yang terbentuk. Proses seleksi dilakukan berdasarkan fungsi fitness yang telah ditentukan. Individu yang memenuhi nilai fitness yang telah ditentukan akan bertahan hidup dan menentukan generasi berikutnya. Tahap selanjutnya, melalui operator rekombinasi dan mutasi, gen akan mengalami perubahan materi genetik untuk membentuk generasi baru.

Algoritma genetika bekerja dari populasi yang merupakan himpunan solusi yang dihasilkan secara acak. Setiap anggota himpunan yang mempresentasikan satu solusi masalah dinamakan kromosom. Kromosom dalam suatu populasi berevolusi dalam iterasi yang dinamakan generasi, tiap kromosom dievaluasi berdasarkan fungsi evaluasi (*fitness function*). Pada algoritma genetika, fitness biasanya dapat berupa fungsi objektif dari masalah yang akan dioptimasi. Kemudian kromosom-kromosom diseleksi menurut nilai fitness masing-masing, kromosom yang kuat mempunyai kemungkinan tinggi untuk bertahan hidup pada generasi berikutnya, tetapi tidak menutup kemungkinan bagi kromosom yang lemah dapat bertahan hidup. Proses seleksi tersebut kemudian ditentukan oleh kromosom-kromosom baru melalui proses *crossover* dan mutasi dari kromosom yang terpilih.

Algoritma genetika jauh lebih baik dalam memecahkan masalah penjadwalan, optimasi algoritma genetika dapat menciptakan solusi penjadwalan ruang dan waktu. Struktur dasar algoritma genetika terdiri atas beberapa tahapan (Haupt dan Haupt, 2004):

1. Inisialisasi populasi.
2. Evaluasi populasi.
3. Seleksi populasi yang dikenali operator genetika.

4. Proses penyilangan pasangan kromosom tertentu.
5. Proses mutasi kromosom tertentu.
6. Evaluasi populasi baru.
7. Ulangi dari langkah 3 selama syarat berhenti belum terpenuhi.

Menurut Michalewicz (1996), Algoritma Genetika harus memiliki lima komponen berikut :

1. Representasi genetik dari setiap solusi yang mungkin dari suatu permasalahan.
2. Cara pembentukan populasi awal atau inisialisasi populasi.
3. Fungsi evaluasi yang berperan menilai fitness dari solusi yang mungkin.
4. Operator genetik yang mengubah komposisi kromosom.
5. Nilai parameter yang digunakan dalam Algoritma Genetika, meliputi ukuran populasi, nilai probabilitas yang diterapkan dalam operator genetik (seleksi, *crossover*, mutasi).

BAB III

DESAIN

3.1 Desain Data

Data yang diolah untuk mendapatkan jadwal terbagi dalam beberapa class, yakni:

1. Kelas
2. Dosen
3. CoursesClass
4. Room
5. Schedule

3.1.1 Desain Data Kelas

Desain pada *class* Kelas terdiri dari 'name' dan 'size'. 'Name' untuk nama kelas dan 'size' untuk berapa jumlah anggota kelas tersebut. Dalam python diimplementasikan seperti berikut:

```
1 class Kelas:
2     kelas = None
3
4     def __init__(self, name, size):
5         self.name = name
6         self.size = size
7
8     @staticmethod
9     def find(name):
10        for i in range(len(Kelas.kelas)):
11            if Kelas.kelas[i].name == name:
12                return i
13        return -1
14
15    def __repr__(self):
16        return "Kelas: " + self.name
17
```

Gambar 1. Desain Data Kelas

3.1.2 Desain Data Dosen

Desain data pada class 'dosen' terdiri dari 'name', yang mana 'name' itu merupakan nama dari dosen. Berikut implementasi class dosen:

```

19 class Dosen:
20     dosen = None
21
22     def __init__(self, name):
23         self.name = name
24
25     @staticmethod
26     def find(name):
27         for i in range(len(Dosen.dosen)):
28             if Dosen.dosen[i].name == name:
29                 return i
30         return -1
31
32     def __repr__(self):
33         return "Dosen Pengampu: " + self.name
34

```

Gambar 2. Desain Data Dosen

3.1.3 Desain Data Courses Class

Desain data pada class 'CoursesClass' adalah 'code' dan 'is_lab'. 'code' ini merupakan kode dari matakuliah, dan 'is_lab' adalah menunjukkan bahwa matakuliah (*courses*) tersebut menggunakan ruangan laboratorium. Berikut implementasi class CoursesClass:

```

36 class CourseClass:
37     classes = None
38
39     def __init__(self, code, is_lab=False):
40         self.code = code
41         self.is_lab = is_lab
42
43     @staticmethod
44     def find(code):
45         for i in range(len(CourseClass.classes)):
46             if CourseClass.classes[i].code == code:
47                 return i
48         return -1
49
50     def __repr__(self):
51         return "Kode Matakuliah: " + self.code
52

```

Gambar 3. Desain Data Courses Class

3.1.4 Desain Data Room

Desain data pada class 'room' adalah 'name', 'size', dan 'is_lab'. 'Name' merupakan nama dari ruangan, 'size' merupakan ukuran(kapasitas) dari ruangan tersebut, dan 'is_lab'

menunjukkan bahwa ruangan tersebut merupakan laboratorium. Berikut implementasi class room:

```
54 class Room:
55     rooms = None
56
57     def __init__(self, name, size, is_lab=False):
58         self.name = name
59         self.size = size
60         self.is_lab = is_lab
61
62     @staticmethod
63     def find(name):
64         for i in range(len(Room.rooms)):
65             if Room.rooms[i].name == name:
66                 return i
67         return -1
68
69     def __repr__(self):
70         return "Ruangan: " + self.name
71
```

Gambar 4. Desain Data Room

3.1.5 Desain Data Schedule

Desain data pada class schedule adalah 'start', 'end', dan 'day'. 'start' merupakan jam memulai sesi perkuliahan, 'end' untuk jam mengakhiri sesi perkuliahan, 'day' merupakan hari-hari perkuliahan. Berikut implementasi class schedule:

```
class Schedule:
    schedules = None

    def __init__(self, start, end, day, is_lab_slot=False):
        self.start = start
        self.end = end
        self.day = day
        self.is_lab_slot = is_lab_slot

    def __repr__(self):
        return "Pukul :" + self.start + "-" + self.end + " Day: " + self.day
```

Gambar 5. Desain Data Schedule

3.2 Prosedur Pengerjaan

Tahapan-tahapan yang dalam pengerjaan pembuatan sistem penjadwalan sebagai berikut:

1. Import semua class yang sudah diimplementasikan ke dalam file python (main.py) dan beberapa *library* yang penting seperti:

```
1 import random, copy
2 from Classes import *
3 from math import ceil, log2
4 import math
```

Gambar 6. Library pada Data

2. Membuat data- data yang akan diolah. Data-data tersebut sesuai dengan beberapa class yang terlebih dahulu dibuat. Contohnya:

```
5 Kelas.kelas = [Kelas("11SI1", 20), Kelas("11SI2", 22), Kelas("12SI1", 30),
6                 Kelas("12SI2", 30), Kelas("13SI1", 28), Kelas("13SI2", 25)]
7
8 Dosen.dosen = [Dosen("Mr. Mario Simaremare"), Dosen("Mr. Tennyson"), Dosen("Mrs. Parmonangan"),
9                Dosen("Mrs. Junita"), Dosen("Mr. Samuel"), Dosen("Mr. Humasak")]
10
11 CourseClass.classes = [CourseClass("KUS1002"), CourseClass("MAS1101"), CourseClass("KUS1002"), CourseClass("TIS1101"),
12                        CourseClass("KUS1001"), CourseClass("FIS1103"), CourseClass("12SI101"),
13                        CourseClass("TIS3001")]
14
15 Room.rooms = [Room("GD512", 40), Room("GD513", 40), Room("GD514", 40),
16               Room("GD522", 40, is_lab=True), Room("GD523", 40, is_lab=True)]
17
18 Schedule.schedules = [Schedule("08:15", "10:00", "Mon"), Schedule("10:15", "12:00", "Mon"),
19                        Schedule("13:15", "15:00", "Mon"), Schedule("15:15", "17:00", "Mon"),
20                        Schedule("08:15", "10:00", "Tue"), Schedule("10:15", "12:00", "Tue"),
21                        Schedule("13:15", "15:00", "Tue"), Schedule("15:15", "17:00", "Tue"),
22                        Schedule("08:15", "10:00", "Wed"), Schedule("10:15", "12:00", "Wed"),
23                        Schedule("13:15", "15:00", "Wed"), Schedule("15:15", "17:00", "Wed"),
24                        Schedule("08:15", "10:00", "Thu"), Schedule("10:15", "12:00", "Thu"),
25                        Schedule("13:15", "15:00", "Thu"), Schedule("15:15", "17:00", "Thu"),]
```

Gambar 7. Data yang diolah

3. Kemudian membuat fungsi `faculty_member_one_class` dan `Kelas_member_one_class`. Fungsi ini digunakan agar dosen dan kelas tidak masuk ke dalam schedule yang sama (waktu yang sama). Berikut implementasinya:

```

30 | # checks that a class is teaches by dosen only one course at a time.
31 | def faculty_member_one_class(chromosome):
32 |     scores = 0
33 |     for i in range(len(chromosome) - 1): # select one cpg pair
34 |         clash = False
35 |         for j in range(i + 1, len(chromosome)): # check it with all other cpg pairs
36 |             if slot_clash(chromosome[i], chromosome[j])\
37 |                 and dosen_bits(chromosome[i]) == dosen_bits(chromosome[j]):
38 |                 clash = True
39 |             if not clash:
40 |                 scores = scores + 1
41 |     return scores
42 |
43 |
44 | # check that a kelas member takes only one class at a time.
45 | def kelas_member_one_class(chromosomes):
46 |     scores = 0
47 |
48 |     for i in range(len(chromosomes) - 1):
49 |         clash = False
50 |         for j in range(i + 1, len(chromosomes)):
51 |             if slot_clash(chromosomes[i], chromosomes[j]) and\
52 |                 kelas_bits(chromosomes[i]) == kelas_bits(chromosomes[j]):
53 |                 clash = True
54 |                 break
55 |             if not clash:
56 |                 scores = scores + 1
57 |     return scores
58 |

```

Gambar 8. fungsi *faculty_member_one_class* dan *Kelas_member_one_class*

4. Kemudian dilakukan simulasi dengan menentukan α , T dan T_{\min} sebelum memasuki fungsi *genetic algortihm*. Menentukan α , T dan T_{\min} menggunakan fungsi simulated annealing. Berikut implementasinya:

```

294 | def simulated_annealing():
295 |     alpha = 0.9
296 |     T = 1.0
297 |     T_min = 0.00001
298 |
299 |     convert_input_to_bin()
300 |     population = init_population(1) # as simulated annealing is a single-state method
301 |     old_cost = cost(population[0])
302 |
303 |     for __n in range(500):
304 |         new_solution = swm(population[0])
305 |         new_solution = ssn(population[0])
306 |         new_cost = cost(new_solution[0])
307 |         ap = acceptance_probability(old_cost, new_cost, T)
308 |         if ap > random.random():
309 |             population = new_solution
310 |             old_cost = new_cost
311 |             T = T * alpha
312 |         # print(population)
313 |         # print("Cost of altered solution: ", cost(population[0]))
314 |         print("\nScore: ", evaluate(population[0]))
315 |         print("\n----- Schedule -----")
316 |         for lec in population[0]:
317 |             print_chromosome(lec)

```

Gambar 9. Fungsi *simulated annealing*

5. Kemudian melakukan algoritma genetic algorithm dengan fungsi sebagai berikut:

```
def genetic_algorithm():
    generation = 0
    convert_input_to_bin()
    population = init_population(3)

    print("\n----- Genetic Algorithm ----- \n")
    while True:

        # if termination criteria are satisfied, stop.
        if evaluate(max(population, key=evaluate)) == 1 or generation == 500:
            print("Generations:", generation)
            print("Best Chromosome fitness value", evaluate(max(population, key=evaluate)))
            print("Best Chromosome: ", max(population, key=evaluate))
            for lec in max(population, key=evaluate):
                print_chromosome(lec)
            break

        # Otherwise continue
        else:
            for _c in range(len(population)):
                crossover(population)
                selection(population, 5)
                mutate(population[_c])
            generation = generation + 1
```

Gambar 10. Algoritma genetic algorithm

6. Untuk menemukan hasilnya, perlu pemanggilan setiap fungsi pada fungsi main. Fungsi main sebagai berikut:

```
47 def main():
48     random.seed()
49     genetic_algorithm()
50     simulated_annealing()
51
52     main()
```

Gambar 11. Pemanggilan fungsi main

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Pada output program terdapat atribut berupa pukul, ruangan, kelas, kode matakuliah, dan dosen pengampu telah tersusun dengan baik. Ouput berupa dataset sudah di kombinasikan dengan baik menggunakan algoritma genetik. Gambar dibawah ini adalah hasil dari program yang dikombinasikan menggunakan Genetic Algorithm.

```
C:\Windows\System32\cmd.exe
D:\Materi Kuliah\Semester 5\CERTAN\proyek\sistem-penjadwalan-mata-kuliah-menggunakan-algoritma-genetika-main\py main.py

----- Genetic Algorithm -----
Generations: 13
Best Chromosome fitness value 1.0
Best Chromosome: ['000000100110010', '1010001000111001', '1000110001101010', '0111010b10011001', '0101000b11100001', '0010010010000001', '1000110100101001', '0100100b111010001', '0000000b10101000', '1010000b10001001', '1000100110000001', '1111011001010010', '0101001001101001', '0010010001100010', '1000110110001000', '0b10100b10110010']

Pukul :13:15-15:00 Day: thu Ruangan: GD514 Kelas: 13S11 Kode Matakuliah: KUS1002 Dosen Pengampu: Mr. Mario Simaremare
Pukul :15:15-17:00 Day: tue Ruangan: GD513 Kelas: 13S11 Kode Matakuliah: FIS1103 Dosen Pengampu: Mr. Mario Simaremare
Pukul :10:15-12:00 Day: thu Ruangan: GD514 Kelas: 11S11 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :15:15-17:00 Day: Mon Ruangan: GD513 Kelas: 11S12 Kode Matakuliah: TIS1101 Dosen Pengampu: Mr. Humasak
Pukul :08:15-10:00 Day: thu Ruangan: GD513 Kelas: 11S12 Kode Matakuliah: KUS2002 Dosen Pengampu: Mr. Samuel
Pukul :08:15-10:00 Day: Mon Ruangan: GD513 Kelas: 11S12 Kode Matakuliah: MAS1101 Dosen Pengampu: Mr. Tennyov
Pukul :10:15-12:00 Day: tue Ruangan: GD513 Kelas: 12S11 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :13:15-15:00 Day: wed Ruangan: GD513 Kelas: 11S12 Kode Matakuliah: KUS2002 Dosen Pengampu: Mrs. Parmonangan
Pukul :10:15-12:00 Day: tue Ruangan: GD512 Kelas: 11S12 Kode Matakuliah: KUS1002 Dosen Pengampu: Mr. Mario Simaremare
Pukul :10:15-12:00 Day: Mon Ruangan: GD513 Kelas: 11S12 Kode Matakuliah: FIS1103 Dosen Pengampu: Mr. Mario Simaremare
Pukul :08:15-10:00 Day: wed Ruangan: GD513 Kelas: 11S12 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :13:15-15:00 Day: wed Ruangan: GD514 Kelas: 13S11 Kode Matakuliah: TIS3001 Dosen Pengampu: Mr. Humasak
Pukul :10:15-12:00 Day: thu Ruangan: GD513 Kelas: 13S11 Kode Matakuliah: KUS2002 Dosen Pengampu: Mr. Samuel
Pukul :08:15-10:00 Day: thu Ruangan: GD514 Kelas: 11S11 Kode Matakuliah: MAS1101 Dosen Pengampu: Mr. Tennyov
Pukul :10:15-12:00 Day: Mon Ruangan: GD512 Kelas: 12S12 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :13:15-15:00 Day: thu Ruangan: GD514 Kelas: 11S12 Kode Matakuliah: MAS1101 Dosen Pengampu: Mrs. Parmonangan

----- Schedule -----
Pukul :15:15-17:00 Day: thu Ruangan: GD512 Kelas: 13S11 Kode Matakuliah: KUS1002 Dosen Pengampu: Mr. Mario Simaremare
Pukul :10:15-12:00 Day: Mon Ruangan: GD522 Kelas: 13S11 Kode Matakuliah: FIS1103 Dosen Pengampu: Mr. Mario Simaremare
Pukul :15:15-17:00 Day: wed Ruangan: GD522 Kelas: 11S11 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :08:15-10:00 Day: thu Ruangan: GD522 Kelas: 11S12 Kode Matakuliah: TIS1101 Dosen Pengampu: Mr. Humasak
Pukul :10:15-12:00 Day: thu Ruangan: GD522 Kelas: 11S12 Kode Matakuliah: KUS2002 Dosen Pengampu: Mr. Samuel
Pukul :08:15-10:00 Day: thu Ruangan: GD512 Kelas: 11S12 Kode Matakuliah: MAS1101 Dosen Pengampu: Mr. Tennyov
Pukul :08:15-10:00 Day: thu Ruangan: GD523 Kelas: 12S11 Kode Matakuliah: KUS2002 Dosen Pengampu: Mrs. Parmonangan
Pukul :15:15-17:00 Day: Mon Ruangan: GD512 Kelas: 11S12 Kode Matakuliah: KUS1002 Dosen Pengampu: Mr. Mario Simaremare
Pukul :15:15-17:00 Day: tue Ruangan: GD514 Kelas: 11S12 Kode Matakuliah: FIS1103 Dosen Pengampu: Mr. Mario Simaremare
Pukul :10:15-12:00 Day: tue Ruangan: GD514 Kelas: 11S12 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :08:15-10:00 Day: wed Ruangan: GD523 Kelas: 13S11 Kode Matakuliah: TIS3001 Dosen Pengampu: Mr. Humasak
Pukul :13:15-15:00 Day: wed Ruangan: GD514 Kelas: 13S11 Kode Matakuliah: KUS2002 Dosen Pengampu: Mr. Samuel
Pukul :15:15-17:00 Day: Mon Ruangan: GD512 Kelas: 11S11 Kode Matakuliah: MAS1101 Dosen Pengampu: Mr. Tennyov
Pukul :15:15-17:00 Day: thu Ruangan: GD522 Kelas: 12S12 Kode Matakuliah: KUS1001 Dosen Pengampu: Mrs. Junita
Pukul :08:15-10:00 Day: Mon Ruangan: GD514 Kelas: 11S12 Kode Matakuliah: MAS1101 Dosen Pengampu: Mrs. Parmonangan
Score: 0.9247311827956989
```

Gambar 12. Output

4.2 Pembahasan

Untuk table yang pertama, merupakan dataset yang menggunakan genetic algorithm dan belum ditentukan alpha, T, T-min. Kemudian setelah dilakukan fungsi genetic algorithm, akan dilakukan fungsi simulated annealing() dimana pada fungsi tersebut sudah ditentukan alpha, T, T-min yang berfungsi untuk menentukan alpha dan fitness dari hasil genetika fungsi genetic algorithm. Jika dilihat pada output, hasilnya akan berbeda diantara fungsi genetic algorithm dan fungsi simulated annealing.

Output dihasilkan merupakan pilihan terbaik dari fitness, hasil genetika dari genetic algorithm. Pada proses proyek ini, semua hasil genetika/fitness itu tidak ditunjukkan semua, melainkan hasil yang terbaik dilihat dari score yang mendekati angka satu(1).

BAB V

PENUTUP

Adapun bagian penutup dalam pengerjaan proyek ini terdiri dari kesimpulan dan saran seperti:

5.1 Kesimpulan

Berdasarkan hasil pembahasan penelitian ini pada BAB IV, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini menghasilkan Sistem Informasi Pendjadwalan Kuliah yang sudah terotomatisasi oleh sistem sehingga memudahkan pihak akademik dan dosen dalam membuat jadwal kuliah.
2. Dengan menggunakan sistem ini dapat meminimalisit angka kesalahan dan efisiensi waktu dalam pembuatan jadwal kuliah.
3. Penelitian ini menghasilkan sistem yang memudahkan mahasiswa untuk mendapatkan informasi jadwal kuliah apabila terjadi perubahan jadwal.
4. Dengan adanya sistem ini, dosen ataupun BAAK akan lebih mudah untuk menginput waktu kesiapan mengajarkan sendiri karena jadwal dosen yang sering berubah-ubah.

5.2 Saran

Berdasarkan hasil pengembangan proyek yang telah dilakukan, diperoleh beberapa saran sebagai perbaikan dalam pengembangan sistem penjadwalan akademik ini di kemudian hari, yaitu: Saat hendak menyusun jadwal menggunakan sistem, user perlu memasukkan beberapa inputan berkali kali sehingga, untuk kedepannya dapat dikembangkan dengan meminimalisir fitur untuk memasukkan inputan sehingga lebih efektif saat digunakan.

5.3 Pembagian Tugas

Adapun pembagian tugas dalam pengerjaan proyek ini ialah sebagai berikut:

Table 1. Pembagian Tugas

No.	NIM	Tugas
1	12S18001	Proposal, Laporan Akhir
2	12S18004	Proposal, Laporan Akhir
3	12S18017	Proposal, Laporan Akhir

4	12S18030	Proposal, Laporan Akhir, <i>code</i> program
5	12S18035	Proposal, Laporan Akhir
6	12S18063	Proposal, Laporan Akhir

BAB VI

DAFTAR PUSTAKA

- [1] Kamus Besar Bahasa Indonesia, Pusat Pembinaan dan Pengembangan Bahasa, Kamus Besar Bahasa Indonesia. Jakarta: Depdikbud, 1990.
- [2] Carr, “An introduction to genetic algorithm,” 2014.
- [3] Carr, “An introduction to genetic algorithm,” 2014.
- [4] L. Budin, D. Jakobović, and M. Golub, “Genetic algorithms in real-time imprecise computing,” *J. Comput. Inf. Technol.*, vol. 8, no. 3, pp. 249–257, 2000.