

Algoritma Genetika dengan Mutasi Terbatas untuk Penjadwalan Perkuliahan

Sisferi Hikmawan ^{1,*}, Windu Gata ¹

¹ Program Studi Ilmu Komputer; STMIK Nusa Mandiri; Jl. Damai No.8, Warung Jati Barat (Margasatwa), Jakarta Selatan, Telp.(021) 78839513 Fax.(021) 78839421; e-mail: gansisferi@gmail.com, windu@nusamandiri.ac.id

* Korespondensi: e-mail: gansisferi@gmail.com

Submitted: 26/02/2021; Revised: 08/03/2021; Accepted: 11/05/2021; Published: 27/05/2021

Abstract

In University, lecture scheduling is the most important factor in service satisfaction for students. UNISMA Bekasi still uses the manual method in scheduling lectures. Genetic Algorithms can solve scheduling with different constraints. In the proposed Genetic Algorithm, the mutation operator is changed to be a limited individual mutation and a selection feature that is adjusted to the constraints in the problem to be solved. And Genetic Algorithms with limited mutations are proven to have advantages in accommodating the constraints found in UNISMA Bekasi. The result of testing in experiments conducted on curriculum data for the Odd Semester of the Academic Year 2020/2021 using a Genetic Algorithm with mutation_individual_terbatas, namely minimum load = 0 with iterations = 10 and population = 500.

Keywords: Data Mining, Genetic Algorithm, Schedule, Mutation

Abstrak

Dalam perkuliahan, penjadwalan perkuliahan merupakan faktor paling penting dalam kepuasan pelayanan terhadap mahasiswa. UNISMA Bekasi masih menggunakan cara manual dalam penjadwalan perkuliahan. Algoritma Genetika dapat memecahkan penjadwalan dengan constraint berbeda-beda. Pada Algoritma Genetika yang diajukan, dilakukan pengubahan operator mutasi menjadi mutasi individu terbatas dan fitur seleksi yang disesuaikan dengan constraint dalam permasalahan yang ingin dipecahkan. Dan Algoritma Genetika dengan mutasi terbatas terbukti memiliki kelebihan dalam mengakomodir permasalahan constraint yang terdapat di UNISMA Bekasi. Dihasilkan Pengujian dalam percobaan yang dilakukan terhadap data kurikulum untuk Semester Ganjil Tahun Akademik 2020/2021 dengan menggunakan Algoritma Genetika dengan mutasi_individual_terbatas yaitu beban minimum = 0 dengan iterasi = 10 dengan populasi = 500.

Kata kunci: Data Mining, Algoritma Genetika, Mutasi, Jadwal Perkuliahan

1. Pendahuluan

Salah satu proses akademik adalah Penjadwalan Perkuliahan. Penjadwalan merupakan tindakan yang berkaitan dengan mengalokasikan sumberdaya yang dibutuhkan dengan rentang waktu tertentu (Pinedo, 2008). Di Fakultas Teknik UNISMA Bekasi, sesuai Surat Keputusan Rektor memiliki kewenangan menentukan penjadwalan dan tetap berkoordinasi dengan DAPA selaku sentral pelayanan mahasiswa. Terjadi banyak revisi terkait jadwal dikarenakan setiap Fakultas masih menggunakan metode manual untuk pembuatan jadwal perkuliahan.

Salah satu permasalahan dalam penjadwalan adalah bentrok ruangan, dosen dan mahasiswa. Tidak hanya bentrok, namun salah satu kepuasan bagi mahasiswa dalam hal jadwal adalah meminimalkan pergerakan mahasiswa dalam berpindah ruangan kelas. Dan inilah yang disebut sebagai *constraint* atau batasan (Pinedo, 2008) (Shiau, 2011).

Algoritma genetika telah digunakan sebagai solusi dalam menyelesaikan permasalahan penjadwalan, karena dapat menghasilkan jadwal yang hampir optimal (Ahmad et al., 2012). Algoritma genetika juga telah digunakan dalam penjadwalan yang lebih khusus yaitu jadwal perkuliahan (Myori & Hastuti, 2019) (Abdelhalim & El Khayat, 2016).

Menurut Qashlim dan Assidiq (Qashlim & Assidiq, 2016) *Constraint* dalam penjadwalan perkuliahan terbagi menjadi dua, yaitu *soft constraint* dan *hard constraint*. Pembagian tersebut untuk membedakan dimana *hard constraint* (batasan yang sulit) merupakan batasan yang harus di ambil saat bertumbukkan dengan *soft constraint* (batasan yang lunak) (Shiau, 2011). Dalam hal ini, kepuasan pengguna menjadi salah satu optimalisasi dalam proses permodelan algoritma genetika.

Penelitian terkait tentang fungsi objektif dalam penjadwalan masih diperbincangkan (Zhang et al., 2020). Metode yang ditawarkan untuk algoritma ini salah satunya dengan menekankan pada fitur mutasi, seleksi dan juga dalam *fitness function*.

Penelitian ini bertujuan untuk menganalisa algoritma genetika untuk penjadwalan perkuliahan di Fakultas Teknik UNISMA Bekasi dalam menyelesaikan permasalahan *constraint* pada Fakultas tersebut menggunakan kombinasi mutasi terbatas dan seleksi.

2. Metode Penelitian

2.1. Pengumpulan Data

Sumber data yang digunakan menggunakan dataset dari Universitas Islam 45 Bekasi, dalam hal ini data akan diambil dari Sistem Informasi Akademik (SIMAK) UNISMA Bekasi dengan menggunakan akun Staff Pengelola Akademik.

Tabel 1. Contoh Data Dosen Pengampu Matakuliah

No	Kode	Nama	NIK	Perkuliahan
1	TKS1621001	Menggambar Bangunan Sipil	45101061998124	REG_A
2	TKS1621002	Kalkulus	45104052015009	REG_A
3	TKS1621003	Fisika Mekanika	45404012016004	REG_A
4	TKS1621004	Statistika dan Probabilitas	45101101994014	REG_A
5	TKS1621005	Statika Statis Tertentu-T	45104032015004	REG_A

Sumber: Hasil Penelitian (2021)

Pengambilan data kurikulum di SIMAK UNISMA Bekasi setelah didapatkan hasilnya adalah seperti pada tabel berikut:

Tabel 2. Contoh Data Dosen Pengampu Matakuliah

No	Kode Matakuliah	Nama Matakuliah	Model Pembelajaran	Kode Semester	Kategori	Penyelenggara	sks
1	TKS2102/T KS162100 2	Kalkulus	Tatap Muka	1	MK Program Studi	Program Studi	3
2	TKS2103/T KS162100 3	Fisika Mekanika	Tatap Muka	1	MK Program Studi	Program Studi	2
3	TKS2104/T KS162100 4	Statistika dan Probabilitas	Tatap Muka	1	MK Program Studi	Program Studi	2
4	TKS2105/T KS162100 5	Statika Statis Tertentu-T	Tatap Muka	1	MK Program Studi	Program Studi	3
5	TKS2106/T KS162100 6	Praktikum Fisika Mekanika	Praktikum	1	MK Program Studi	Program Studi	1

Sumber: Hasil Penelitian (2021)

2.2. Pengolahan Data

Tahap selanjutnya yaitu pengolahan data. Dalam pengolahan data, langkah yang dilakukan adalah dengan cara menghilangkan mata kuliah yang tidak membutuhkan penjadwalan seperti Skripsi dan Kuliah Kerja Nyata. Mata kuliah Skripsi dan Kuliah Kerja Nyata merupakan mata kuliah diluar dari jadwal dan biasanya penentuan waktu serta tanggalnya ditentukan oleh Fakultas.

2.3. Inisialisasi dan Representasi Populasi Awal

Pada algoritma genetika, pembangkitan populasi awal adalah langkah dimana dimunculkannya gen dengan kromosom secara acak untuk nantinya akan diseleksi sebagai orang tua. Untuk pembangkitan populasi awal tetap random namun dibatasi sesuai dengan Jenis Mata Kuliah dan Ruang Induk.

```
def populasi_awal(self):
    populasi = []
    for _ in range(self.n_populasi_awal):
        populasi.append(self.individu())
    self._populasi = populasi
```

Dengan perintah di atas, populasi_awal menyimpan hasil fungsi individu () ke dalam array populasi [] dan dilakukan pengulangan sejumlah n_populasi_awal. N_populasi_awal

adalah nilai untuk menentukan berapa jumlah populasi yang kita inginkan dalam Algoritma Genetika. Untuk isi dari fungsi individu () adalah sebagai berikut

```
def individu(self):
    jumlah_kromosom = len(self.data_dosen)
    individu = []
    for i in range(jumlah_kromosom):
        if (self.data_dosen["Jenismk"][i] != "T"):
            kode_ruangan = self.data_dosen["Jenismk"][i]
            index = self.data_ruang[(self.data_ruang["Kode Ruangan"] == kode_ruangan)].index.to_numpy()
            individu.append(np.random.choice(index))
        else :
            kode_ruangan = self.data_dosen["r_ind"][i]
            index = self.data_ruang[(self.data_ruang["Kode Ruangan"] == kode_ruangan)].index.to_numpy()
            individu.append(np.random.choice(index))
    return np.array(individu)
```

2.4. Implementasi Algoritma Genetika

Constraint di Fakultas Teknik UNISMA Bekasi terdiri dari *hard constraints* dan *soft constraints*. *Hard constraints*: a) Dalam satu plot waktu kuliah, Dosen hanya dapat mengajar satu kali; b) Setiap plot waktu perkuliahan hanya dapat diajarkan oleh satu Dosen; c) Mahasiswa hanya dapat menghadiri satu waktu perkuliahan dalam satu plot waktu; d) Setiap ruang kelas hanya dapat digunakan satu perkuliahan pada waktu yang sama; e) Laboratorium komputer akan digunakan untuk jadwal matakuliah tertentu. *Soft constraints*: a) Dosen dan mahasiswa dapat memilih hari, jam dan ruang yang disukai berdasarkan plot waktu yang tersedia; b) Meminimalkan gerakan mahasiswa untuk melakukan perpindahan ruangan; c) Gerakan dosen untuk perpindahan jadwal hari, jam dan ruang.

Fitness dalam Algoritma Genetika menggunakan rumus (Hijriana, 2015) :

$$fitness = \frac{1}{1 + P_{inalti}} \tag{1}$$

Dimana :

$$pinalti = \sum \text{Jumlah beban}$$

Jumlah beban adalah akumulasi bobot dari pelanggaran terhadap setiap *constraint*. Dengan begitu jumlah beban adalah total pelanggaran yang terjadi dalam gen. Maka *fitness* terbaik adalah yang bernilai 1. Namun *fitness* tidak terbatas hanya menggunakan pembagian angka 1, dalam penelitian ini penulis menggunakan fungsi pembebanan.

Pembobotan dari pelanggaran terhadap constraint mengacu pada tabel seperti berikut:

Tabel 3. *Constraint* dan Beban

No.	<i>Constraint</i>	Beban Pelanggaran
C1	Dalam satu plot waktu kuliah, Dosen hanya dapat mengajar satu kali	1
C2	Mahasiswa hanya dapat menghadiri satu waktu perkuliahan dalam satu plot waktu	1

No.	Constraint	Beban Pelanggaran
C3	Setiap ruang kelas hanya dapat digunakan satu perkuliahan pada waktu yang sama	1
C4	Laboratorium komputer akan digunakan untuk jadwal matakuliah tertentu	1

Sumber: Hasil Penelitian (2021)

Dan dihasilkan persamaan fungsi untuk *constraint* beban adalah sebagai berikut:

$$\sum C1 = \sum_{i=\text{duplikasi}}^n NIK_i, t_i, d_i$$

$$\sum C2 = \sum_{i=\text{duplikasi}}^n k_i, t_i, d_i$$

$$\sum C3 = \sum_{i=\text{duplikasi}}^n r_i, t_i, d_i$$

$$\sum C4 = \sum_{i=1}^n jr_i \neq r_i$$

$\sum C1$ = adalah total beban untuk pelanggaran C1, dimana di hitung sejumlah n baris yang terdapat NIK= Id dosen, t=plot waktu, dan d=hari yang sama. Jika terjadi duplikasi setiap baris atau i adalah 1.

$\sum C2$ = adalah total beban untuk pelanggaran C2, dimana di hitung sejumlah n baris yang terdapat k= Kode Kelas, t=plot waktu, dan d=hari yang sama, jika terjadi duplikasi setiap baris atau i adalah 1.

$\sum C3$ = adalah total beban untuk pelanggaran C3, dimana di hitung sejumlah n baris yang terdapat r= Kode Ruangan, t=plot waktu, dan d=hari yang sama, jika terjadi duplikasi setiap baris atau i adalah 1.

$\sum C4$ = adalah total beban untuk pelanggaran C4, dimana di hitung apakah matakuliah Praktek sesuai dengan Kode Ruangan yang ditentukan untuk praktek. Jika berbeda maka setiap baris atau i yang berbeda adalah 1.

Fitness menggunakan formula sebagai berikut:

```
def beban_total(self, individu):
    tabel_kromosom = self.dataframe(individu)
    beban_p = 0
    beban_t = 0

    beban_ruang_waktu_hari = tabel_kromosom.duplicated(['vruangjam', 'hari']).sum()
    beban_dosen_waktu_hari = tabel_kromosom.duplicated(['N I K / N a m a', 'plot jam', 'hari']).sum()
    beban_mahasiswa = tabel_kromosom.duplicated(['kode kelas', 'plot jam', 'hari']).sum()
    for i in range(len(tabel_kromosom)):
        if tabel_kromosom["Jenismk"][i] != "T" :
            if tabel_kromosom["Kode Ruangan"][i] != tabel_kromosom["Jenismk"][i]:
                beban_p = beban_p+0.001
        else :
            if tabel_kromosom["Kode Ruangan"][i] != tabel_kromosom["r_ind"][i]:
```

```
beban_t = beban_t+0.000001
#print(beban_t)
return beban_dosen_waktu_hari + beban_ruang_waktu_hari + beban_mahasiswa + beban_p +
beban_t
```

Permodelan algoritma genetika standar menurut Pinedo (Pinedo, 2008) adalah sebagai berikut:

Langkah 1.

$K=1$

Bangkitkan populasi, dan dihasilkan $P = (gen\ 1... n)$

Langkah 2.

Pilih 2 gen terbaik dari populasi gen, sebagai Parent yaitu $genAyah(K)$ dan $genIbu(K)$

Gandakan $genAyah$ dan $GenIbu$ menjadi $genAnak1(K)$ dan $GenAnak2(K)$

Lakukan kawin silang

Lakukan mutasi

Langkah 3.

Set $P = (genAyah(K), genIbu(K), genAnak1(K), genAnak2(K))$

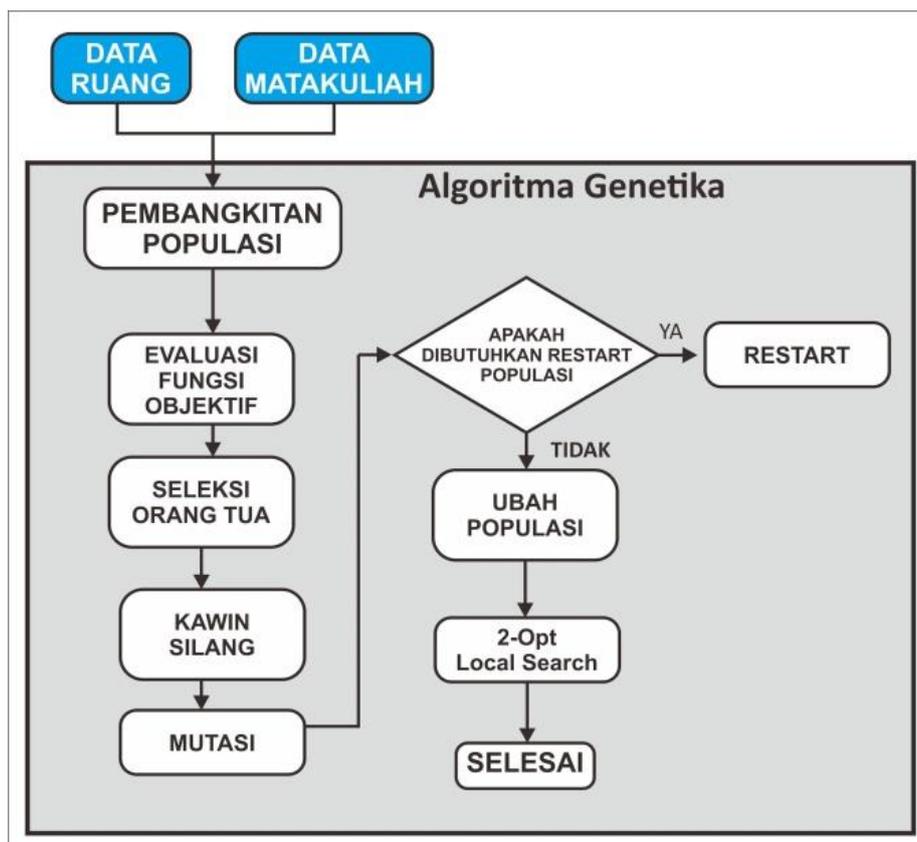
Jika $K = N$ maka STOP

Jika tidak lakukan langkah 2.

Permodelan algoritma genetika menggunakan optimalisasi *Sequence-Dependent Setup Times* untuk mengatasi permasalahan *constraint* ruang berurutan menggunakan formula:

```
Data : populasi, setup, ruang induk , beban , seleksi(.), kawin silang(.), mutasi(.), pmut, restart(.), 2-
optLocalSearch(.)
    Buat populasi;
    Hasil : kromosom= (k1, k2, ....kn )
    while 1 ≤ beban ;
    seleksi →ayah, ibu;
    kawin silang;
    if random() ≤ probabilitas mutasi maka kawin silang;
    jika kriteria untuk restart populasi terpenuhi maka
    populasi, f itness ← restart( populasi, f itness );
    kembali ke proses seleksi
    end
    best ← nilai terbaik ∈ populasi;
print best
```

Dan dapat terlihat dari bagan seperti pada gambar berikut :



Sumber: Hasil Penelitian (2021)

Gambar 1. Bagan Alur Proses Algoritma Genetika *Sequence-Dependent Setup Times*

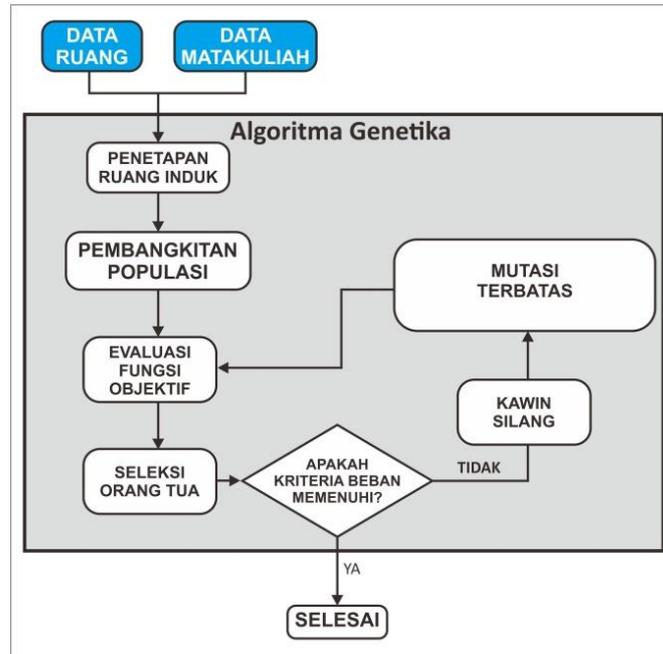
Sedangkan dalam permodelan Algoritma Genetika Mutasi Terbatas yang penulis ajukan adalah penyederhanaan fitur *selection* dan penentuan model mutasi yang digunakan. Serta untuk memungkinkan terjadinya seleksi gen yang dibutuhkan maka proses mutasi menggunakan formula seperti dibawah ini:

```

def mutasi_individu_terbatas(self, individu, n_mutasi):
    for i in range(n_mutasi):
        tabel_kromosom = self.dataframe(individu)
        locus = np.random.randint(len(individu))
        if tabel_kromosom["Jenismk"][locus] == "T":
            kode_kelas = tabel_kromosom["r_ind"][locus]
            index = self.data_ruang[(self.data_ruang["Kode Ruangan"] == kode_kelas)].index.to_numpy()
            individu[locus] = np.random.choice(index)
        else:
            kode_ruangan = tabel_kromosom["Jenismk"][locus]
            index = self.data_ruang[(self.data_ruang["Kode Ruangan"] == kode_ruangan)].index.to_numpy()
            individu[locus] = np.random.choice(index)

    return individu
    
```

Dan dapat terlihat dari bagan seperti pada gambar berikut :



Sumber: Hasil Penelitian (2021)

Gambar 2. Bagan Alur Proses Algoritma Genetika menggunakan Mutasi Terbatas

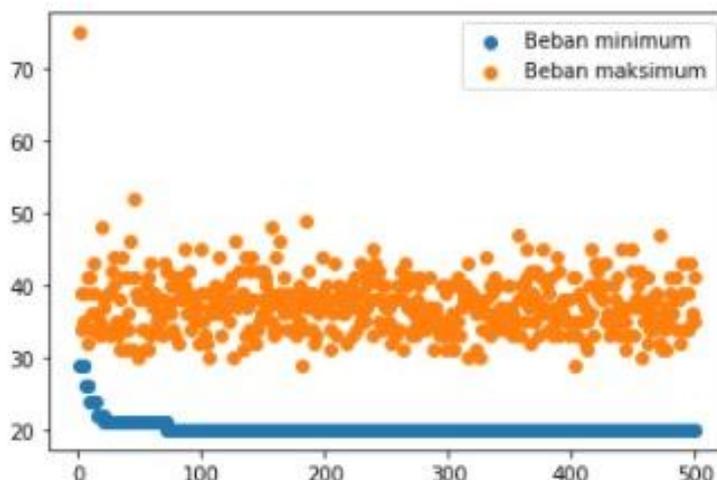
2.5. Evaluasi dan Validasi Hasil Penelitian

Setelah tersimpan iterasi, beban atau *fitness* dan juga konvergensinya maka dilakukan evaluasi dan validasi dari algoritma genetika yang kami ajukan. Evaluasi adalah melihat hasil dengan fungsi objektif yaitu menghitung nilai fungsi apakah membaik atau memburuk, serta melihat konvergensi apakah terjadi penyimpangan atau sebaliknya.

3. Hasil dan Pembahasan

3.1. Algoritma Genetika Pinedo

Pengujian dengan menggunakan Algoritma Genetika standar, dihasilkan dalam iterasi ke 500 seperti pada gambar berikut:



Sumber: Hasil Penelitian (2021)

Gambar 3. Scatter Plot GA Pinedo

3.2. Algoritma Genetika dengan *Sequence-Dependent Setup Times*

Pada pengujian, Algoritma Genetika dengan optimalisasi menurut Levi R. Abreu ini dijalankan dengan menggunakan populasi sebesar 100, 200, 300, dan 500. Pengujian ini dilakukan untuk mengetahui apakah sistem penjadwalan dengan metode Algoritma Genetika ini berjalan sesuai dengan tujuan yang diinginkan yaitu terpenuhinya semua batasan atau constraint yang sebelumnya telah ditetapkan dan perbedaan dari setiap jumlah populasi yang digunakan.

Untuk menggambarkan *soft constraint* “Meminimalkan gerakan mahasiswa untuk melakukan perpindahan ruangan”. Jika ini dimasukkan kedalam fungsi maka akan bertabrakan dengan *constraint* C4. Untuk itulah mengapa diletakkan di *soft constraint*. Perhitungan nilai minimum pergerakan kelas adalah

$$\text{Nilai minimum pergerakan kelas} = \frac{\sum K_{h,kr}}{\sum K} \times 100 \tag{2}$$

Artinya nilai $\sum K_{h,kr}$ adalah total kode kelas dengan hari dan kode ruangan yang sama dibagi dengan $\sum K$ yaitu total kode kelas dikali 100. Nilai ini untuk melihat hubungan jika mutasi terbatas diterapkan terhadap fungsi objektif terhadap *soft constraint* tersebut.

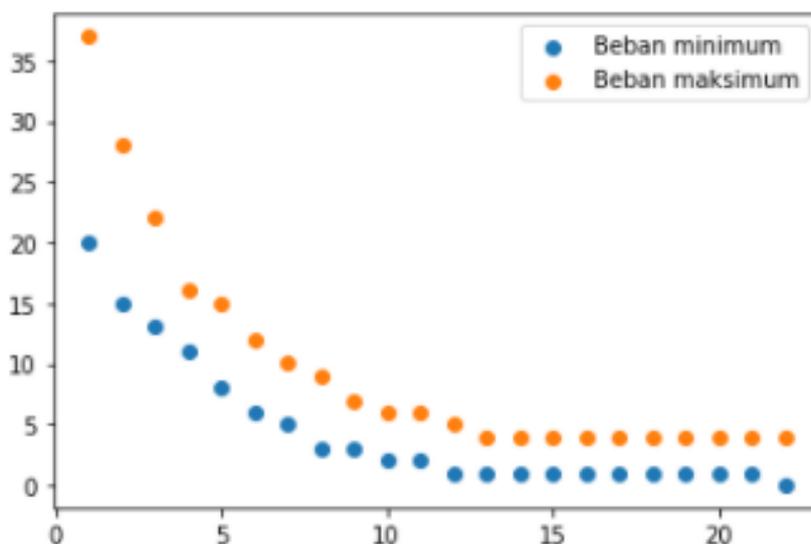
Berikut hasil dari Algoritma Genetika dengan n_keturunan = 100

Tabel 5. Hasil Algoritma Genetika populasi = 100

n_populasi	Beban < 1 setelah Iterasi ke:	nilai minimum pergerakan kelas
100	22	8 %

Sumber: Hasil Pengolahan Data (2021)

Pada Tabel 5 terlihat iterasi minimal yang dihasilkan adalah 22 dengan nilai minimum pergerakan kelas sebesar 8 %. Untuk *scatter plot* yang terlihat pada Gambar 5.



Sumber: Hasil Penelitian (2021)

Gambar 5. Scatter Plot GA *Sequence-Dependent Setup Times* populasi = 100

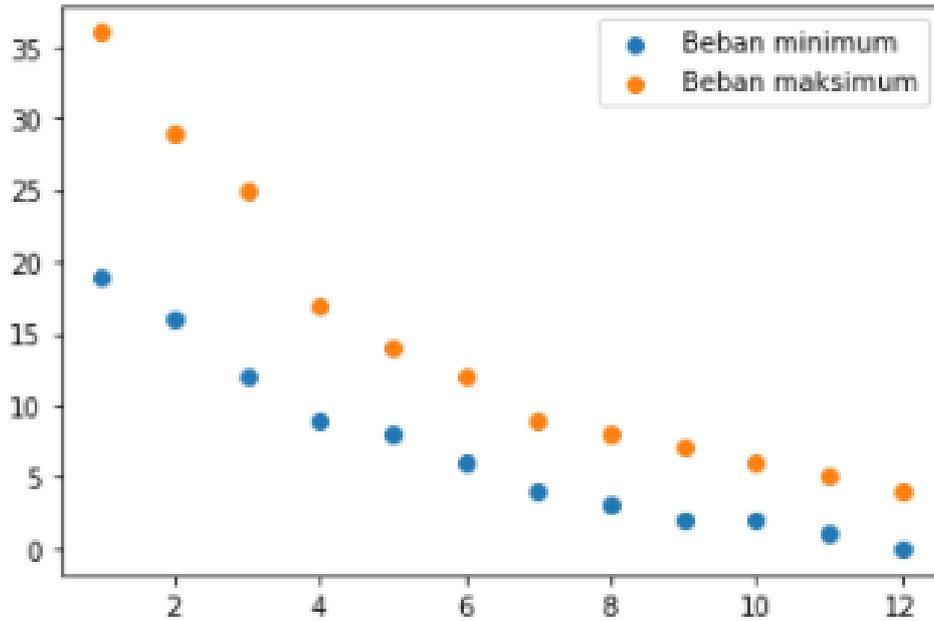
Berikut hasil dari Algoritma Genetika dengan n_keturunan = 200

Tabel 6. Hasil Algoritma Genetika populasi = 200

n_populasi	Beban < 1 setelah Iterasi ke:	nilai minimum pergerakan kelas
200	12	19.5 %

Sumber: Hasil Pengolahan Data (2021)

Pada Tabel 6 terlihat iterasi minimal yang dihasilkan adalah 12 dengan nilai minimum pergerakan kelas sebesar 19.5 %. Untuk *scatter plot* yang terlihat pada gambar berikut:



Sumber: Hasil Penelitian (2021)

Gambar 6. Scatter Plot GA Sequence-Dependent Setup Times populasi = 200

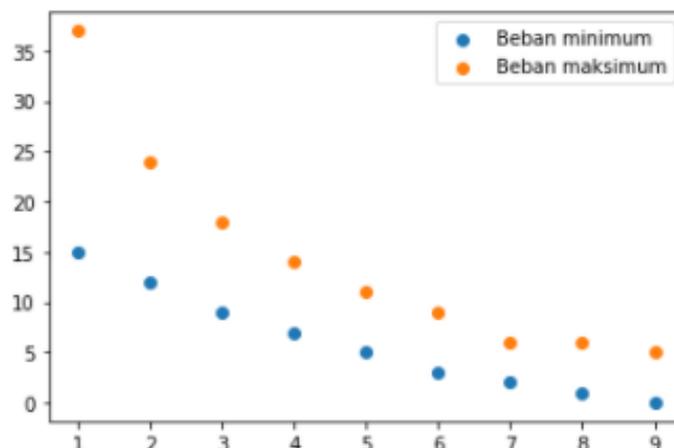
Berikut hasil dari Algoritma Genetika dengan n_keturunan = 300

Tabel 7. Hasil Algoritma Genetika Sequence-Dependent Setup Times populasi = 300

n_populasi	Beban < 1 setelah Iterasi ke:	nilai minimum pergerakan kelas
300	9	14.6 %

Sumber: Hasil Penelitian (2021)

Pada Tabel 7 terlihat iterasi minimal yang dihasilkan adalah 9 dengan nilai minimum pergerakan kelas sebesar 14.6 %. Untuk *scatter plot* yang terlihat pada Gambar berikut:



Sumber: Hasil Penelitian (2021)

Gambar 7. Scatter Plot GA Levi R. Abreu populasi = 300

3.3. Algoritma Genetika dengan Mutasi Terbatas

Pada pengujian, Algoritma Genetika dengan mutasi terbatas ini dijalankan dengan menggunakan populasi sebesar 100, 200, 300, dan 500. Pengujian ini untuk mengetahui apakah sistem penjadwalan dengan metode Algoritma Genetika mutasi terbatas ini berjalan sesuai dengan tujuan yang diinginkan yaitu terpenuhinya semua batasan atau *constraint* yang sebelumnya telah ditetapkan dan perbedaan dari setiap jumlah populasi yang digunakan.

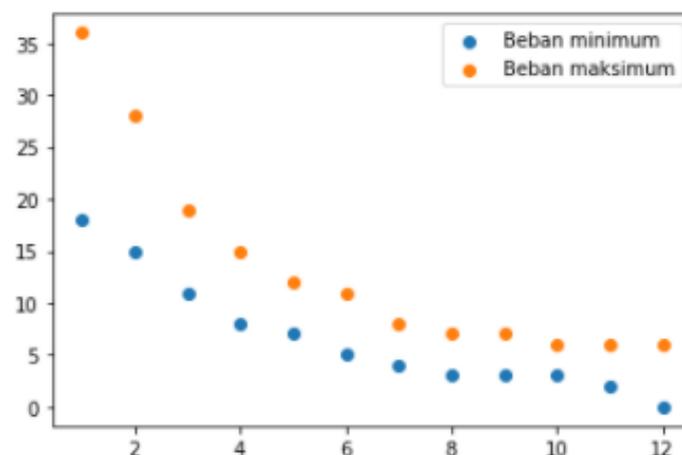
Berikut hasil dari Algoritma Genetika dengan n_keturunan = 100

Tabel 9. Hasil Algoritma Genetika dengan Mutasi Terbatas, populasi = 100

n_populasi	Beban = 0 setelah Iterasi ke:	nilai minimum pergerakan kelas
100	12	20.3 %

Sumber: Hasil Penelitian (2021)

Pada Tabel 9 terlihat iterasi minimal yang dihasilkan adalah 12 dengan nilai minimum pergerakan kelas sebesar 20.3 %. Untuk *scatter plot* yang terlihat pada Gambar 9 terlihat sedikit terjadinya perubahan jarak beban maksimum dan minimum.



Sumber: Hasil Penelitian (2021)

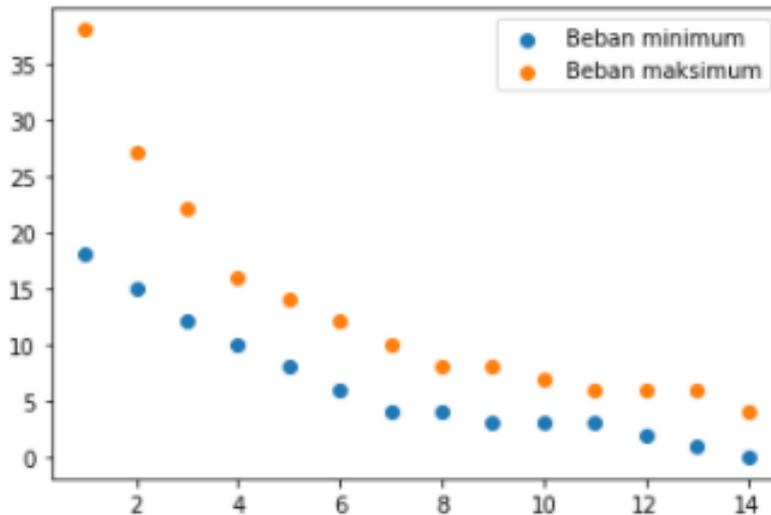
Gambar 9. Scatter Plot GA Mutasi Terbatas Populasi = 100

Setelah itu Algoritma Genetika dijalankan dengan keturunan = 200. Dan dihasilkan yang terlihat pada Tabel 10 yaitu iterasi minimal yang didapatkan adalah 14 iterasi dengan nilai minimum pergerakan kelas adalah 6.6 %.

Tabel 10. Hasil Algoritma Genetika populasi = 200

n_populasi	Beban = 0 setelah iterasi ke:	nilai minimum pergerakan kelas
200	14	6.6 %

Sumber: Hasil Penelitian (2021)



Sumber: Hasil Penelitian (2021)

Gambar 10. Scatter Plot Beban Maksimum Dan Minimum Populasi = 200

Terlihat Scatter Plot pada Gambar 10 terlihat konsistensi jarak antara Beban Maksimum dan Beban Minimum mulai terlihat. Variasi mutasi gen menjadi semakin berkurang dikarenakan Mutasi Terbatas mulai mendapat sedikit pilihan.

Setelah itu Algoritma Genetika dijalankan dengan keturunan = 300. Dan dihasilkan yang terlihat pada Tabel 11 yaitu iterasi minimal yang didapatkan adalah 10 iterasi dengan nilai minimum pergerakan kelas adalah 14.75%.

Tabel 11. Hasil Algoritma Genetika populasi = 300

n_populasi	Beban = 0 setelah iterasi ke:	nilai minimum pergerakan kelas
300	10	6.6%

Sumber: Hasil Penelitian (2021)

Setelah itu Algoritma Genetika dijalankan dengan keturunan = 500. Dan dihasilkan yang terlihat pada Tabel 12 yaitu iterasi minimal yang didapatkan adalah 10 iterasi dengan nilai minimum pergerakan kelas adalah 11.4%.

Tabel 12. Hasil Algoritma Genetika populasi = 500

n_populasi	Beban = 0 setelah iterasi ke:	nilai pergerakan kelas minimum
500	10	11.4%

Sumber: Hasil Penelitian (2021)

Nilai minimum pergerakan kelas tidak akan tercapai 100% dikarenakan matakuliah praktek lebih banyak. Hal ini normal karena data yang diuji menggunakan program studi Teknik Mesin S1 yang terdapat banyak matakuliah di ruang Praktek. Namun dengan adanya ruang induk hasilnya adalah mahasiswa sudah memiliki ruang teori yang diprioritaskan.

Tabel 13. Rata-rata Hasil Percobaan

n_populasi	GA. Sequence-Dependent Setup		GA Mutasi Terbatas	
	Times Iterasi	Nilai beban = 0.00140 nilai pergerakan kelas (%)	Nilai beban = 0 Iterasi	Nilai beban = 0 nilai pergerakan kelas (%)
100	22	8	12	20.3
200	12	19.5	14	6.6
300	9	14.6	10	6.6
500	12	17.1	10	11.4

Sumber: Hasil Penelitian (2021)

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa Algoritma Genetika dengan mutasi individu terbatas dapat membantu dalam pembuatan jadwal perkuliahan dengan *hard constraint* dan *soft constraint* secara proporsional. Pembatasan mutasi juga sangat diperlukan untuk menangani kasus *constraint* dengan penjadwalan perkuliahan yang memiliki kondisi pemilihan ruang, jam dan hari yang terbatas. Pengujian dalam percobaan yang dilakukan terhadap data kurikulum untuk Semester Ganjil Tahun Akademik 2020/2021 dengan menggunakan Algoritma Genetika dengan Mutasi Terbatas yang menghasilkan beban minimum = 0 dengan iterasi = 10 dengan populasi sebanyak 500. Kelemahan dari metode ini adalah adanya proses pengklasifikasian terhadap kelas dan untuk kasus yang lebih diinginkan keteracakan akan membutuhkan lebih banyak iterasi.

Daftar Pustaka

- Abdelhalim, E. A., & El Khayat, G. A. (2016). A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA). *Alexandria Engineering Journal*, 55(2), 1395–1409. <https://doi.org/10.1016/j.aej.2016.02.017>
- Ahmad, S. G., Munir, E. U., & Nisar, W. (2012). PEGA: A Performance Effective Genetic Algorithm for task scheduling in heterogeneous systems. *Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications, HPCC-2012 - 9th IEEE International Conference on Embedded Software and Systems, ICCESS-*

- 2012, 1082–1087. <https://doi.org/10.1109/HPCC.2012.158>
- Aristoteles, Wardiyanto, & Dwiastuti, A. (2015). Evaluasi Kinerja Genetic Algorithm (GA) dengan Strategi Perbaikan Kromosom Studi Kasus: Knapsack Problem 1. *Jurnal Komputasi*, 3(2), 162–168.
- Hijriana, N. (2015). PENERAPAN METODE ALGORITMA GENETIKA UNTUK PERMASALAHAN PENJADWALAN PERAWAT (Nurse Scheduling Problem). *Info Teknik*, 16(1), 61–74.
- Myori, D. E., & Hastuti, H. (2019). Kombinasi Logika Fuzzy dan Algoritma Genetika untuk Masalah Penjadwalan Perkuliahan. *Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung (SENER 2018)*, 284–292. <http://senter.ee.uinsgd.ac.id/repositori/index.php/prosiding/article/view/senter2018p31>
- Pinedo, M. L. (2008). Scheduling: Theory, algorithms, and systems. In *Scheduling: Theory, Algorithms, and Systems*. <https://doi.org/10.1007/978-0-387-78935-4>
- Qashlim, A., & Assiddiq, M. (2016). Penerapan Algoritma Genetika untuk Sistem Penjadwalan Kuliah. *Jurnal Ilmiah Ilmu Komputer*, 2(1), 1–6.
- Shiau, D. F. (2011). A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences. *Expert Systems with Applications*, 38(1), 235–248. <https://doi.org/10.1016/j.eswa.2010.06.051>
- Zhang, G., Hu, Y., Sun, J., & Zhang, W. (2020). An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. *Swarm and Evolutionary Computation*, 54(February), 100664. <https://doi.org/10.1016/j.swevo.2020.100664>