

ARTICLE

## Penerapan Algoritma Genetika Dalam Pengelompokan Mahasiswa KKN (Studi Kasus: KKN Angkatan XLII Universitas Mercu Buana Yogyakarta)

### *Application Of Genetic Algorithm In Grouping KKN College Students (Case Study: KKN Batch XLII Universitas Mercu Buana Yogyakarta)*

Arinadi Nur Rohmad\* dan Mutaqin Akbar

Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta, Yogyakarta, Indonesia

\*Penulis Korespondensi: arinadi.nur@gmail.com

(Disubmit 23-08-29; Diterima 23-09-13; Dipublikasikan online pada 24-02-05)

#### Abstrak

Kuliah Kerja Nyata-KKN (Program Pengembangan Masyarakat) merupakan kegiatan pengabdian masyarakat oleh mahasiswa dengan pendekatan lintas ilmu dan sektor, pada Universitas Mercu Buana Yogyakarta dikordinasikan oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) yang sekarang bernama Pusat Penelitian Pengabdian Masyarakat dan Kerjasama (P3MK). Dibutuhkan kelompok KKN dengan kriteria seperti kelas, program studi, dan jenis kelamin. Penelitian ini bertujuan untuk memberikan alternative pengelompokan mahasiswa KKN Angkatan XLII UMBY dengan menerapkan algoritma genetika. Representasi kromosom disajikan oleh pendekatan baris index yang dihasilkan secara acak selanjutnya dilakukan inisiasi populasi awal, seleksi elite, reproduksi, dan menentukan kromosom elite dari ranking. Aturan dalam pembagian KKN ini berfungsi sebagai constrain. Solusi terbaik adalah nilai fitness 100% dan mampu memberikan solusi dengan error/penalti rendah. Pada penelitian ini didapati bahwa hasil dari pengelompokan 807 peserta KKN Angkatan XLII UMBY mencapai 95% untuk kelas 22D dengan waktu eksekusi 5 menit 26 detik menggunakan 1 elite, 1500 offspring dan 300 mutant setiap generasi.

**Kata kunci:** Algoritma Genetika; KKN; Pengelompokan; Permutasi

#### Abstract

Kuliah Kerja Nyata-KKN (Community Development Program) is a community service activity by college students with a cross-disciplinary and sectoral approach, at Universitas Mercu Buana Yogyakarta coordinated by the Institute for Research and Community Service (LPPM) which is now called the Center for Community Service Research and Cooperation (P3MK). KKN groups are needed with criteria such as class, study program, and gender. This research aims to provide an alternative grouping of UMBY KKN Batch XLII students by applying genetic algorithms. Chromosome representation is presented by a randomly generated index row approach, then initial population initiation, elite selection, reproduction, and determining elite chromosomes from ranking. The rules in the division of KKN serve as constraints. The best solution is 100% fitness value and is able to provide solutions with low error/penalty. In this research, it was found that the results of grouping 807 KKN participants of Batch XLII UMBY reached 95% for class 22D with an execution time of 5 minutes 26 seconds

using 1 elite, 1500 offspring and 300 mutants every generation.

**KeyWords:** Genetic Algorithm; KKN; Grouping; Permutation

## 1. Pendahuluan

Kuliah Kerja Nyata (KKN) adalah bentuk kegiatan pengabdian kepada masyarakat oleh mahasiswa. Program KKN mengalami perkembangan sebagai tanggapan dari kondisi dinamika masyarakat. Program KKN merupakan tanggapan UMBY (Universitas Mercu Buana Yogyakarta) terhadap kuatnya tekanan globalisasi pada lapisan masyarakat di Indonesia. Perkembangan KKN ditandai dengan adanya perubahan paradigma, yaitu dari paradigma pembangunan (*development*) menjadi pemberdayaan (*empowerment*) [1].

Kegiatan pembelajaran layanan di UMBY, yang dikoordinasikan oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM), melibatkan berbagai pihak di lingkungan UMBY dan mitra kerja dari luar UMBY seperti pemerintah, industri, lembaga usaha, lembaga finansial, dan lembaga mitra pemberdayaan masyarakat. UMBY telah mengembangkan 3 jenis KKN yang berbeda sejak tahun 2019. Pertama, KKN reguler untuk program 12D yang mewajibkan menginap, dengan alokasi terpusat berdasarkan lokasi di dusun-desa-kecamatan tertentu. Kedua, KKN Tematik untuk program 22D yang tidak wajib menginap, dengan alokasi terpusat berdasarkan lokasi tematik seperti pasar atau daerah tertentu. Ketiga, KKN Mandiri untuk 32D yang tidak wajib menginap, namun mahasiswa harus membuat proposal dan surat kesediaan lokasi dari UMKM, kelompok warga, kelompok wisata, kelompok tani, dan lainnya.

Pengelompokan KKN di UMBY selama ini dilakukan secara mandiri oleh mahasiswa, namun terdapat tantangan dalam pembagian kelompok yang ideal. Kriteria ideal untuk pembagian kelompok mencakup kesamaan program kelas, program studi, dan jenis kelamin. Solusi untuk mengoptimalkan pembagian kelompok KKN melibatkan penggunaan Algoritma Genetika (GA) yang menirukan proses evolusi untuk memberikan solusi dengan nilai *fitness* terbaik. GA menggunakan representasi kromosom dengan pendekatan berbasis partisi dari daftar mahasiswa secara acak. Solusi terbaik dihasilkan dengan nilai *fitness* kromosom yang rendah dalam rangka mencapai pembagian kelompok yang lebih sesuai dengan kondisi program KKN.

Terdapat beberapa jenis GA, termasuk *Generational GA (GGA)*, *Steady-State GA (SSGA)*, *Steady-Generational GA (SGGA)*, dan  $(\mu + \mu)$ -GA, masing-masing memiliki cara kerja yang berbeda dalam menciptakan generasi baru dan menggantikan individu dalam populasi. Pilihan GA yang digunakan tergantung pada masalah yang diselesaikan dan sumber daya yang tersedia. SSGA lebih efisien daripada GGA dalam hal evaluasi fungsi, tetapi GGA memiliki keuntungan mempertahankan keragaman dalam populasi. SGGA menggabungkan keuntungan GGA dan SSGA, sedangkan  $(\mu + \mu)$ -GA menjamin individu terbaik dalam populasi baru tetapi memerlukan evaluasi fungsi yang lebih banyak. Penting untuk mempertimbangkan keseimbangan antara eksplorasi dan eksploitasi dalam memilih GA dan untuk menyeimbangkan eksplorasi dan eksploitasi dalam populasi untuk mencapai solusi konvergen [2].

Pengusulan metode algoritma genetika dalam penelitian ini meninjau pada penelitian sebelumnya yaitu Pemodelan Algoritma Genetika dalam Pengelompokan Siswa Pada Kolaborasi Tim Proyek Perangkat Lunak menggunakan 40 data mahasiswa sebagai data uji, populasi awal 100, dan probabilitas crossover 0.5. Pada hasil generasi akhir menunjukkan bahwa algoritma genetika menunjukkan tingkat keberhasilan 62.5% [3]. Pada penelitian sebelumnya kromosom direpresentasikan sebagai anggota kelompok yang akan digunakan sebagai *parent* dalam proses reproduksi, sebagai pembeda pada penelitian ini kromosom direpresentasikan sebagai keseluruhan data mahasiswa yang dijadikan sebagai *parent* pada proses reproduksi kemudian dipecah menjadi kelompok sesuai dengan jumlah anggota per kelompok.

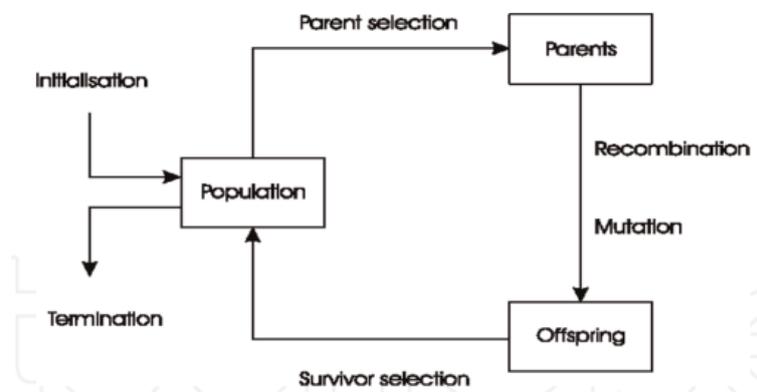
*Pymoo* adalah sebuah *framework* optimisasi multi-objektif yang kuat berbasis algoritma genetika (GA) dalam bahasa pemrograman *Python*. Dengan menggunakan *Pymoo*, pengguna dapat dengan mudah mengimplementasikan dan menjalankan algoritma genetika untuk mencari solusi dengan nilai kecocokan terbaik dalam masalah dengan banyak kriteria yang saling bertentangan. *Pymoo* menyediakan berbagai fitur dan komponen yang kaya, seperti pemodelan matematika yang fleksibel, operator genetik yang dapat disesuaikan, serta metrik evaluasi yang berguna untuk analisis hasil optimisasi. Dengan keunggulannya dalam

optimisasi multi-objektif menggunakan algoritma genetika, *Pymoo* telah menjadi pilihan populer bagi peneliti dan praktisi dalam berbagai bidang, termasuk rekayasa, keuangan, logistik, dan banyak lagi [4].

*Python*, yang pertama kali dirilis pada tahun 1991, telah menjadi bahasa pemrograman yang umum digunakan di berbagai bidang. Banyak pengembang, ilmuwan, dan analis menggunakan *Python* secara luas karena bahasanya mudah dipelajari, didokumentasikan dengan baik, dan dapat diandalkan. *Python* adalah bahasa pemrograman yang kuat dengan fitur-fitur seperti pernyataan *if*, perulangan, generator, objek, dan pemahaman daftar (*list comprehensions*). *Python* juga memiliki ekosistem yang kaya dan aktif dengan banyak *library* yang berguna [5].

## 2. Metode

Algoritma genetika adalah salah satu sub-area dari algoritma evolusioner yang terinspirasi dari konsep evolusi Darwin. Algoritma ini mengimplementasikan konsep seleksi alam, reproduksi, *crossover*, dan mutasi dalam memecahkan masalah optimisasi.



Gambar 1. Flow Algoritma Genetika [6]

Dalam algoritma genetika (Gambar 1), solusi diwakili sebagai kromosom dan dinilai menggunakan fungsi kecocokan. Solusi-solusi optimal dipilih untuk reproduksi, menghasilkan keturunan dengan kombinasi gen dari orangtua, yang kemudian mengalami *crossover* dan mutasi. Algoritma ini mengatur populasi solusi acak. Setiap solusi direpresentasikan sebagai kromosom, dihitung nilai kecocokan (*fitness*) masing-masing kromosom untuk persaingan. Seleksi orangtua awal berdasarkan nilai kecocokan menghasilkan benih-benih untuk anak melalui rekombinasi dan mutasi. Operator genetik ini mempertahankan keragaman. Kromosom orangtua dipilih secara acak untuk persilangan, dan keturunan mengalami mutasi. Eksplorasi melalui rekombinasi dan mutasi mengeksplorasi berbagai wilayah pencarian untuk individu terbaik. Keturunan dinilai untuk kelangsungan hidup. Pada tahap penggantian, anak menggantikan sebagian solusi awal berdasarkan nilai kecocokan. Seleksi alam seiring waktu dan iterasi membimbing menuju optimum global [6].

Alur penelitian pengelompokan peserta KKN XLII menggunakan algoritma genetika di Universitas Mercu Buana Yogyakarta sebagai berikut :

### 2.1 Pengumpulan Data

Proses dimulai dengan pengumpulan data publik melalui sumber-sumber yang tersedia, seperti *website* KKN dan observasi langsung terhadap mahasiswa yang terlibat [7]. Tujuannya adalah untuk mendapatkan pemahaman awal mengenai peserta program KKN. Setelah data publik terkumpul, langkah berikutnya adalah melakukan wawancara dengan Pusat Penelitian Pengabdian Masyarakat dan Kerjasama (P3MK) Universitas Mercu Buana Yogyakarta. Hal ini bertujuan untuk mengonfirmasi data publik dan memperoleh informasi tambahan yang tidak terjangkau sebelumnya. Jika ditemukan ketidakvalidan dalam data publik, dilakukan perbaikan untuk menjaga akurasi dan kualitas data [8]. Setelah langkah perbaikan selesai, tahap berikutnya melibatkan permintaan data kredensial yang lebih rinci dan rahasia dari P3MK. Data ini diperlukan untuk mendukung analisis dan pengelompokan mahasiswa KKN dengan menggunakan

algoritma genetika.

### 2.2 Pra-pengolahan Data

Setelah tahap Data Kredensial telah diterima, dilakukan tahap pengecekan guna memastikan kelengkapan data yang diperoleh [9].

Apabila terdapat data yang tidak lengkap, langkah selanjutnya adalah kembali ke tahap sebelumnya. Diperlukan permintaan tambahan atau pengecekan ulang terhadap data kredensial oleh P3MK.

Namun, jika data dinyatakan lengkap, langkah selanjutnya adalah Seleksi Data Relevan. Pada tahap ini, data yang telah lengkap akan dipilah untuk memilih hanya data yang relevan dalam penelitian ini.

### 2.3 Pengolahan Data

Pada tahap pengolahan data, data publik yang telah dikumpulkan dari berbagai sumber seperti *website* KKN dan observasi terhadap mahasiswa peserta KKN digabungkan dengan data kredensial yang bersifat rahasia. Dengan menggabungkan kedua jenis data ini, dapat terbentuk satu set data yang lebih komprehensif dan lengkap. Proses penggabungan ini dilakukan untuk menentukan parameter-parameter yang akan digunakan dalam Algoritma Genetika [10].

#### 2.3.1 Penentuan *Constrain*

*Constrain penalty fitness* adalah metode dalam evolusi genetik untuk mengatasi batasan pada individu dalam Algoritma Genetika (GA) [11]. Batasan tersebut dapat berupa batasan numerik, logika, atau kombinasi keduanya. Dalam GA, individu dinilai berdasarkan *fitness*, tetapi ketika melanggar batasan, konsep *constrain penalty fitness* digunakan dengan mengurangi nilai *fitness* individu yang melanggar atau memberikan penalti. Teknik ini memasukkan faktor penalti pada nilai *fitness* individu yang melanggar batasan, dapat disesuaikan berdasarkan tingkat pelanggaran dan pentingnya batasan tersebut. Penentuan faktor penalti yang tepat penting, karena yang terlalu tinggi menghambat eksplorasi solusi, sementara yang terlalu rendah menghasilkan solusi yang tidak sesuai. Dengan pendekatan ini, GA cenderung menghasilkan individu yang mematuhi batasan dengan baik.

Berikut adalah nilai bobot pada masing-masing aturan ditentukan berdasarkan prioritas dari P3MK dalam pembagian kelompok KKN.

Tabel 1. Daftar *Constrain* dan Bobot *Penalty*

Aturan	Kode	Bobot
Jumlah Prodi	C1	0,4
Jumlah Anggota Per Prodi	C2	0,3
Perbandingan laki – laki dan perempuan	C3	0,2
Jumlah Anggota Kelompok	C4	0,1

C1 - Jumlah prodi dalam 1 kelompok harus minimal 3. Apabila aturan ini tidak terpenuhi maka akan memberikan *penalty* nilai 0,4. Apabila terpenuhi, maka akan memberikan nilai 0.

C2 - Jumlah anggota per prodi dalam 1 kelompok harus minimal 2. Apabila aturan ini tidak terpenuhi maka akan memberikan *penalty* nilai 0,3. Apabila terpenuhi, maka akan memberikan nilai 0.

C3 - Rasio minimal perbandingan jumlah laki-laki dan perempuan ditentukan 30% artinya jika rasio laki-laki atau perempuan di bawah 30% akan diberikan *penalty* nilai 0,2. maka akan memberikan nilai 0. Apabila terpenuhi, maka akan memberikan nilai 0.

C4 - Jumlah anggota per kelompok harus 10. Artinya jika kelompok memiliki jumlah anggota kurang atau lebih dari 10 akan diberikan *penalty* nilai 0,1. Apabila terpenuhi, maka akan memberikan nilai 0.

Pada studi kasus pengelompokan KKN UMBY, Terdapat aturan bahwa mahasiswa dengan tipe kelas 12D, 22D atau 32D tidak boleh dicampur dalam 1 kelompok karena terkait jenis program KKN yang akan diberikan P3MK. Hal tersebut dapat diatasi dengan dilakukan filter data oleh program kemudian eksekusi GA untuk setiap kelas secara terpisah.

### 2.3.2 Perumusan *Fitness*

Dalam Algoritma Genetika, langkah penting adalah merumuskan *fitness* untuk mengevaluasi kualitas solusi dalam populasi, terutama saat ada batasan yang harus dipatuhi [12]. Dalam konteks penelitian pengelompokan mahasiswa KKN Universitas Mercu Buana Yogyakarta, digunakan 4 batasan yang diwakili oleh rumus (1) ini untuk menghitung *fitness* individu, di mana *fitness constraint penalty* digunakan sebagai bobot dalam perumusan ini.

$$f(x) = \frac{\sum (\sum_1^4 Fn)}{Qn} \quad (1)$$

Keterangan:

$f(x)$  : Nilai *fitness* dari *mean* total *penalty*

$Fn$  : bobot *constrain*

$Qn$  : Jumlah Kelompok

Sebagai contoh Tabel 2 adalah contoh data 26 peserta KKN yang terdiri dari 15 laki – laki dan 11 perempuan yang akan dibagi ke dalam 3 kelompok. Sehingga kelompok yang diharapkan adalah dalam satu kelompok terdiri dari minimal 3 laki – laki atau perempuan dan kelompok 3 dipastikan kekurangan anggota. Dari

Tabel 2. Contoh Data Mahasiswa

Index	NIM	Prodi	Jenis Kelamin	Kelompok
0	10001	P1	M	1
1	10002	P1	M	1
2	10003	P1	M	1
3	10004	P2	M	1
4	10005	P2	M	1
5	10006	P2	M	1
6	10007	P2	M	1
7	10008	P3	F	1
8	10009	P3	M	2
9	10010	P3	F	2
10	10011	P3	M	2
11	10012	P3	F	2
12	10013	P1	F	2
13	10014	P2	F	2
14	10015	P2	M	2
15	10016	P2	F	2
16	10017	P2	M	2
17	10018	P3	F	2
18	10019	P3	M	2
19	10020	P3	F	2
20	10021	P3	M	3
21	10022	P3	F	3
22	10023	P2	M	3
23	10024	P2	F	3
24	10025	P2	M	3
25	10026	P2	F	3

Tabel 2 dapat dilihat pada kelompok 1 melanggar C3, sedangkan kelompok 2 melanggar C2 dan kelompok 3 melanggar C1, C3, dan C4. Berikut contoh perhitungan untuk memperoleh nilai *fitness* menggunakan

Rumus (1).

$$f(x) = \frac{(0 + 0 + 0.2 + 0) + (0 + 0.3 + 0 + 0) + (0.4 + 0 + 0.2 + 0.1)}{3}$$

$$f(x) = \frac{0.2 + 0.3 + 0.8}{3} = 0.43$$

Jadi, hasil pembagian kelompok pada Tabel 2 memiliki *fitness* 0,43.

### 2.3.3 Penentuan Objektif

Algoritma Genetika (GA) dengan Tujuan Tunggal, atau yang juga dikenal sebagai *Single Objective GA*, adalah metode optimisasi yang menggunakan prinsip-prinsip inspirasi dari teori evolusi alami untuk mencari solusi terbaik untuk masalah dengan satu tujuan atau kriteria.

GA tujuan tunggal melibatkan pencarian solusi terbaik untuk masalah yang didefinisikan dalam bentuk fungsi tujuan tunggal yang harus dicapai. Tujuan umumnya dinyatakan sebagai maksimisasi atau minimisasi fungsi objektif yang diukur dari kumpulan parameter yang dikenal sebagai individu.

Dalam penelitian ini, tujuan yang digunakan adalah minimalisasi [13]. Pilihan ini dilakukan karena metode perumusan *fitness* yang digunakan dalam penelitian ini berbasis *penalty*.

Jika sebuah individu melanggar batasan atau kendala, penalti diberlakukan pada nilai *fitness*. *Penalty* ini akan membuat nilai *fitness* menjadi lebih buruk, sehingga individu-individu yang melanggar batasan akan memiliki nilai *fitness* yang lebih rendah dan kemungkinan lebih rendah untuk dipilih dalam proses seleksi dan reproduksi.

Dengan menggunakan pendekatan ini, GA dengan objektif minimalisasi dan metode perumusan *fitness* berbasis *penalty* dapat menemukan solusi yang memenuhi batasan kendala dan pada saat yang sama mencari solusi yang memiliki nilai fungsi objektif minimal.

## 2.4 Enkripsi Data

Dalam penelitian ini, data di enkripsi menggunakan barisan (*array*) *index* dengan tipe data *integer*. Tujuannya adalah untuk memuat data dengan referensi *index* saat evaluasi, dan dalam kasus ini, tidak boleh mengubah jumlah atau konten data, hanya boleh mengubah urutan data [14].

Misalkan barisan *index* awal yang berisi 0 hingga 25:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]

Dengan menggunakan enkripsi ini, data pada posisi yang ditentukan tetap ada dan hanya mengalami perubahan urutan dalam barisan *index*. Hal ini memungkinkan untuk memuat data dengan referensi *index* yang sesuai saat evaluasi atau penggunaan data.

Untuk membagi kelompok, dapat dilakukan dengan cara memisahkan mereka berdasarkan jumlah anggota per kelompok. Misalnya:

18	19	20	21	22	23	24	25	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Kelompok 1										Kelompok 2										Kelompok 3					

Gambar 2. Pembagian Kelompok

## 2.5 Inisiasi Populasi

Inisiasi populasi dilakukan dengan menggunakan metode iterasi sesuai dengan parameter "*pop\_size*" yang telah ditentukan sebelumnya. Setiap iterasi dalam proses ini bertujuan untuk membuat enkripsi data dengan menghasilkan urutan acak dari indeks data mahasiswa yang akan di proses [15].

## 2.6 Metode Seleksi

Dalam proses seleksi induk menggunakan metode *Elitism*, dilakukan pemilihan satu individu unggul (individu terbaik) dan satu individu non-unggul secara acak. Tujuannya adalah menjaga individu berkualitas

tinggi (unggul) sambil memberi kesempatan pada individu lain dalam populasi untuk berkontribusi pada generasi berikutnya. Ini menciptakan peluang adil bagi individu non-unggul untuk menjadi induk dan turut serta dalam reproduksi populasi, sembari tetap mempertahankan individu unggul guna menjaga kualitas dan ciri-ciri yang diinginkan dalam generasi mendatang [16].

### 2.7 Metode Reproduksi

Dalam penelitian ini, menjaga konsistensi data tanpa mengubah ukuran atau kontennya menjadi fokus utama. Permutasi digunakan untuk mencari solusi dengan metode reproduksi yang terbatas. Pertama, pengembangan operator *crossover* sesuai, seperti *Ordered Crossover (OX)*[17].

Berikut pada Gambar 3 adalah ilustrasi operasi OX pada contoh data mahasiswa Tabel 2 dengan titik potong yang didapat secara *random*. Operasi permutasi OX berhasil membuat kombinasi kromosom unik baru dari mahasiswa dengan *index* 23 yang pada *parent* 1 anggota kelompok 3 kemudian pada *child* berpindah menjadi anggota kelompok 1 dan mahasiswa dengan *index* 22 pada *parent* 2 anggota kelompok 2 kemudian pada *child* berpindah menjadi anggota kelompok 3.

	Random cut point																									
Parent 1	10	25	3	21	7	18	15	2	13	20	24	19	4	1	14	11	16	8	6	0	9	5	17	12	22	23
Parent 2	5	8	20	1	3	0	4	7	19	6	12	25	23	15	10	9	22	16	21	18	14	2	11	24	13	17
	Distribute uniq data in order										Direct distribute to Child															
Child	5	3	7	12	25	23	15	10	13	20	24	19	4	1	14	11	16	8	6	0	9	22	21	18	2	17
	Kelompok 1								Kelompok 2								Kelompok 3									

Gambar 3. Kombinasi OX

Kedua, penggunaan operator mutasi seperti *Swap*, *Insertion*, atau *Inversion*. dilakukan untuk memperkenalkan variasi dalam populasi tanpa merusak ukuran atau konten data, dan hal ini juga relevan dengan penelitian ini.

	Random cut point																									
	5	3	7	12	25	23	15	10	13	20	24	19	4	1	14	11	16	8	6	0	9	22	21	18	2	17
	Inverst Order to Mutant (Off spring)																									
Mutant	5	3	7	12	25	6	8	16	11	14	1	4	19	24	20	13	10	15	23	0	9	22	21	18	2	17
	Kelompok 1								Kelompok 2								Kelompok 3									

Gambar 4. Mutasi *Inversion*

### 2.8 Metode Terminasi

Menghentikan algoritma genetika berdasarkan kriteria tertentu. Pertama, metode berdasarkan jumlah generasi (*n\_gen*) menghentikan algoritma setelah mencapai generasi yang ditentukan, memberi kesempatan pada solusi untuk berkembang. Kedua, metode berdasarkan jumlah evaluasi fungsi tujuan (*n\_eval*) menghentikan algoritma setelah fungsi tujuan dievaluasi sejumlah tertentu, berguna untuk mengontrol penggunaan sumber daya. Ketiga, metode berdasarkan waktu menghentikan algoritma setelah waktu tertentu berlalu, memungkinkan hasil terbaik dalam batasan waktu. Keempat, metode berdasarkan periode menghentikan algoritma setelah periode tertentu, memberikan fleksibilitas pada durasi atau iterasi yang diinginkan sebelum solusi dihentikan [18].

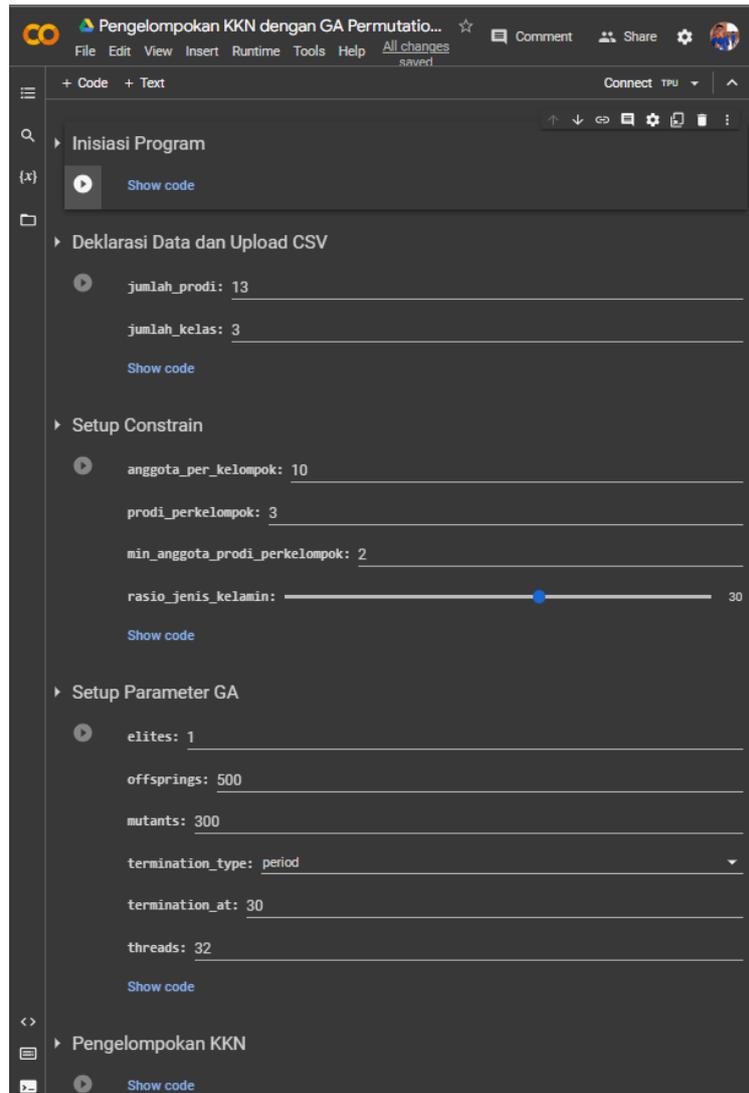
## 3. Hasil

Pengujian dilakukan pada Data Mahasiswa peserta KKN Angkatan XLII dengan total 807 peserta dari kelas 12D, 22D dan 32D. Dalam pengujian ini, peneliti melakukan perbandingan antara tingkat konvergen *fitness* dan waktu eksekusi dari parameter pada Tabel 3 untuk algoritma genetika (GA).

Pengelompokan kelas 12D dengan 607 peserta, 22D dengan 144 peserta dan 32D 56 peserta tidak boleh dicampur maka proses GA dilakukan terpisah.

Tabel 3. Parameter Pengujian

Parameter	Pengujian 1	Pengujian 2	Pengujian 3
<i>elites</i>	100	1	1
<i>offsprings</i>	1500	500	1500
<i>mutants</i>	1500	300	300
<i>termination_type</i>	<i>period</i>	<i>period</i>	<i>period</i>
<i>terminationtable_at</i>	30	30	30
<i>threads</i>	32	32	32



Gambar 5. Tampilan Program di Google Colab

Peneliti mengamati bagaimana *fitness* berkembang seiring berjalannya waktu eksekusi, sambil memperhatikan waktu yang diperlukan mencapai tingkat konvergensi. Dengan menganalisis hasil perbandingan ini, peneliti dapat mengevaluasi efektivitas dan performa parameter-parameter yang digunakan dalam mencapai solusi dengan *fitness* tertinggi.

Dari kombinasi parameter pada Tabel 3 kepada kelas 12D, 22D, dan 32D hasilnya dapat dilihat pada Tabel 4.

Hanya memberi 1 individu *elite* mengakibatkan strategi *elite* selection selalu memilih individu dengan nilai paling tinggi untuk *crossover* dengan non elite secara *random*. Berikut dapat dilihat Tabel 5 adalah contoh hasil pengelompokan kelas 12D pada pengujian 3.

Dapat dilihat pada kelompok 2 tidak melanggar *constrain* sedangkan pada kelompok 1 prodi P4 dan P2

Tabel 4. Contoh Data Mahasiswa

Kelas	Pengujian	<i>n_gen</i>	<i>n_eval</i>	<i>Fitness%</i>	<i>Time</i>
12D	1	71	213100	74.91	12:45:18 AM
12D	2	188	150401	84.59	12:31:57 AM
12D	3	88	158401	86.56	12:33:32 AM
22D	1	62	186100	90	12:10:47 AM
22D	2	46	36801	90	12:02:02 AM
22D	3	53	95401	95.33	12:05:26 AM
32D	1	31	93100	85	12:02:25 AM
32D	2	32	25601	85	0:00:36
32D	3	31	55801	85	12:01:22 AM

Tabel 5. Contoh Hasil Pengelompokan

Nim	Jenis Kelamin	Prodi	Kelas	Kelompok
191110050	M	P4	12D	1
18052579	F	P2	12D	1
191110014	M	P4	12D	1
191110010	M	P4	12D	1
190710018	F	P1	12D	1
190610003	F	P3	12D	1
191110113	M	P4	12D	1
191110094	F	P4	12D	1
190510155	F	P2	12D	1
190810131	M	P5	12D	1
190710286	M	P1	12D	2
190210113	M	P8	12D	2
190710171	M	P1	12D	2
190510098	M	P2	12D	2
190710039	F	P1	12D	2
17021101	M	P8	12D	2
190210116	F	P8	12D	2
18071327	M	P1	12D	2
190710205	F	P1	12D	2
190510051	M	P2	12D	2

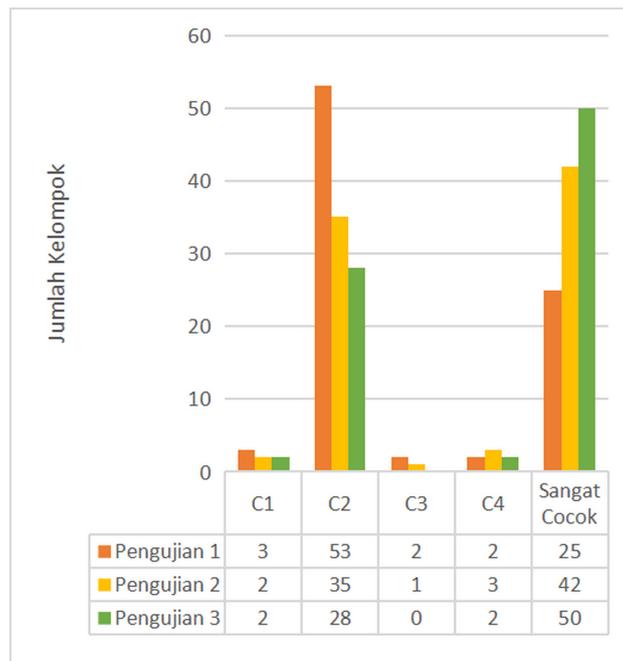
memiliki jumlah anggota memenuhi *constrain* tetapi P1, P3, dan P5 hanya memiliki 1 anggota di mana hal tersebut kurang ideal untuk kelompok KKN sehingga mendapat penalti dan mengurangi nilai *fitness*.

#### 4. Pembahasan

Strategi elit seleksi dengan hanya 1 individu elite menyebabkan pilihan individu tertinggi untuk *crossover*. Pada kelompok 2, tidak ada pelanggaran *constrain*, namun pada kelompok 1, beberapa program studi memiliki anggota di bawah *constrain*, mengurangi nilai *fitness*. Perubahan nilai *offspring* dan *mutant* memengaruhi penggunaan *memory* atau waktu eksekusi. Dapat disimpulkan bahwa populasi yang besar menggunakan lebih banyak *memory* dan berpotensi kehabisan *memory* sebelum konvergensi, sedangkan populasi yang kecil dapat menghambat penemuan gen terbaik karena kurangnya *offspring* dari *gen elite*.

Berikut pada Gambar 6 adalah rekap pengelompokan pada pengujian 1, 2 dan 3.

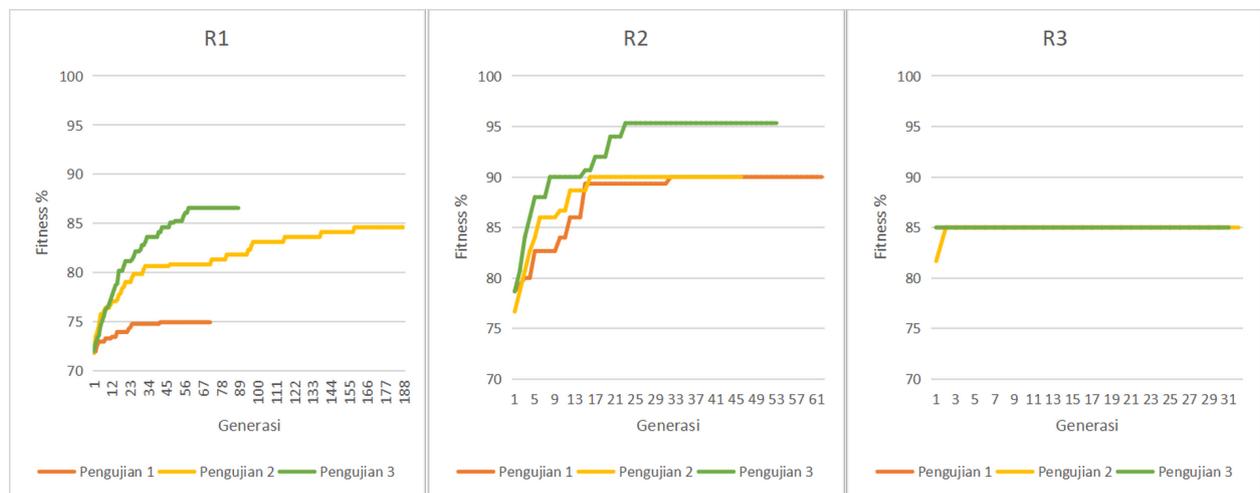
Sebagian besar kelompok melanggar *constrain 2* (C2) kekurangan anggota per prodi, yaitu minimal setiap anggota per prodi adalah 2 mahasiswa. Jika *value constrain* C2 diubah menjadi minimal anggota per prodi



Gambar 6. Rekap Pengujian

1 dan variabel C1 diubah menjadi minimal prodi per kelompok 5 atau lebih maka hasil *fitness* berpotensi lebih baik dengan hasil kelompok KKN yang lebih homogen.

Berdasarkan pengujian, berikut pada Gambar 7 adalah visualisasi dari pengelompokan KKN kelas 12D, 22D, dan 32D.

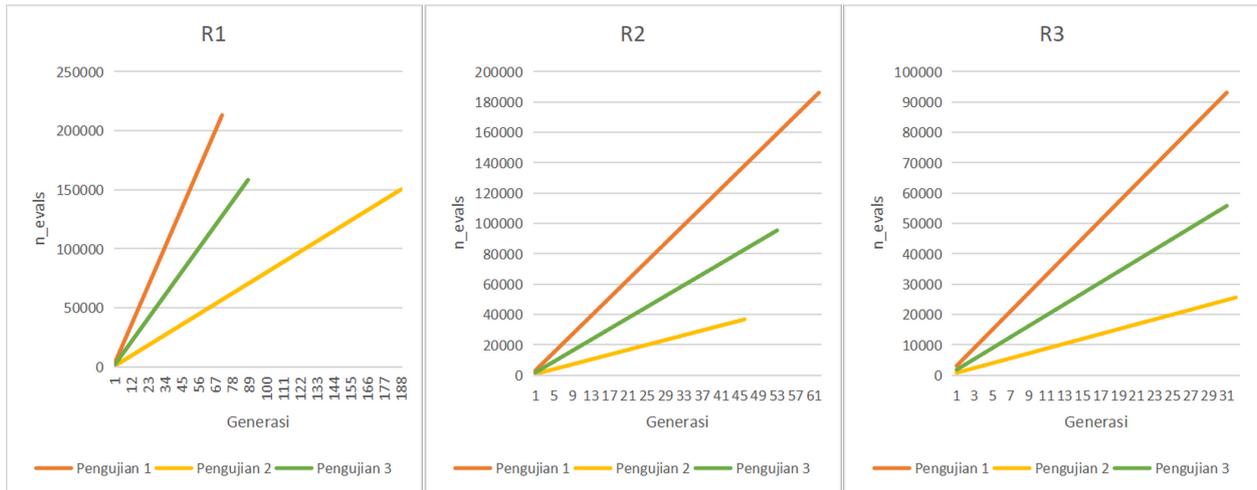


Gambar 7. Visualisasi Hasil *Fitness*

Didapati bahwa pengujian 3 memiliki hasil terbaik karena kecepatan konvergen dengan *fitness* yang tinggi mencapai 95% untuk kelas 22D. Tetapi dari semua kombinasi parameter pengujian 32D mengalami stagnasi dari generasi 1 berada di 85% karena sedikitnya peserta.

Pada kasus 32D walaupun terlihat stagnasi namun perubahan nilai *offspring* dan *mutant* berpengaruh pada penggunaan *memory* atau waktu eksekusi. Berikut pada Gambar 8 adalah visualisasi *n\_evals* yang dapat menggambarkan penggunaan *memory* dan waktu eksekusi dari pengujian.

Dari sini dapat disimpulkan bahwa nilai populasi yang tinggi menggunakan banyak *memory* untuk melakukan evaluasi dan berpotensi untuk kehabisan *memory* sebelum iterasi selesai menyatakan solusi konvergen. Namun, kekurangan populasi akan berpotensi menghambat proses penemuan gen terbaik berikutnya karena kekurangan *offspring* dari *gen elite*.

Gambar 8. Visualisasi  $n\_evals$ 

## 5. Simpulan

Kuliah Kerja Nyata-KKN (Program Pengembangan Masyarakat) merupakan kegiatan pengabdian masyarakat oleh mahasiswa dengan pendekatan lintas ilmu dan sektor, pada Universitas Mercu Buana Yogyakarta dikoordinasikan oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) yang sekarang bernama Pusat Penelitian Pengabdian Masyarakat dan Kerjasama (P3MK). Artikel ini menyajikan penerapan algoritma genetika dalam pengelompokan mahasiswa KKN dengan metode permutasi digunakan untuk menciptakan kombinasi baru kemudian dibagi menjadi kelompok dengan jumlah anggota yang telah ditentukan. Dibutuhkan kelompok KKN dengan kriteria seperti kelas, program studi, dan jenis kelamin. Berdasarkan penelitian yang dilakukan pada 807 data mahasiswa peserta KKN Angkatan XLII dari kelas 12D, 22D, dan 32D dengan parameter pengujian ke 3, algoritma genetika menghasilkan *fitness* 95% pada 22D, 86.56% pada 12D dan 85% pada 32D dengan total waktu eksekusi 40 menit 20 detik sehingga dapat di jadikan alternatif untuk melakukan pengelompokan angkatan berikutnya. Didapati bahwa sebanyak 28 Kelompok yang dibentuk dari pengujian 3 melanggar *constrain 2* (C2), Program mengalami kesulitan dalam membentuk kelompok dengan anggota per prodi minimal 2. Penelitian selanjutnya diharapkan dapat melanjutkan untuk prototipe dikembangkan hingga memiliki antarmuka yang lebih *user-friendly* dan menyesuaikan parameter untuk mendapatkan hasil yang optimal.

## Pustaka

- [1] H. Hermanto, "Kuliah Kerja Nyata Mahasiswa Its Sebagai Media Pemberdayaan Masyarakat Siap Menghadapi Era Revolusi Industry 4.0," *Widya Laksana*, vol. 10, no. 1, pp. 1–6, 2021, doi: 10.23887/jwl.v10i1.23220.
- [2] A. Jenkins, V. Gupta, A. Myrick, and M. Lenoir, "Variations of Genetic Algorithms," *CoRR*, vol. abs/1911.00490, 2019, doi: 10.48550/arXiv.1911.00490.
- [3] D. W. Utomo, D. Kurniawan, and R. R. Sani, "Pemodelan Algoritma Genetika dalam Pengelompokan Siswa Pada Kolaborasi Tim Proyek Perangkat Lunak," in *Seri Prosiding Seminar Nasional Dinamika Informatika*, 2020.
- [4] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020, doi: 10.1109/ACCESS.2020.2990567.
- [5] Y. Babuji et al., "Parsl: Pervasive parallel programming in Python," in *HPDC 2019- Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing*, Association for Computing Machinery, Inc, Jun. 2019, pp. 25–36. doi: 10.1145/3307681.3325400.
- [6] S. Ventura, J. Luna, and J. Moyano, *Genetic Algorithms*. IntechOpen, 2022. doi: 10.5772/intechopen.94664.

- [7] M. Ichsan, L. Rusdiana, S. Sam'ani, F. Haris, and M. H. Qamaruzzaman, "Perancangan e-business UKM Kerajinan Khas Daerah Kalimantan Tengah Di Kota Palangka Raya," *Jurnal Sains Komputer Dan Teknologi Informasi*, vol. 4, no. 2, pp. 38–42, 2022, doi: 10.33084/jsakti.v4i2.3340.
- [8] S. C. Nurzanah, S. Alam, and T. I. Hermanto, "Analisis Association Rule Untuk Identifikasi Pola Gejala Penyakit Hipertensi Menggunakan Algoritma Apriori (Studi Kasus: Klinik Rafina Medical Center)," *JIKO (Jurnal Informatika dan Komputer)*, vol. 5, no. 2, pp. 132–141, 2022, doi: 10.33387/jiko.v5i2.4792.
- [9] D. R. Agustian and B. A. Darmawan, "Analisis Clustering Demam Berdarah Dengue Dengan Algoritma K-Medoids (Studi Kasus Kabupaten Karawang)," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 1, pp. 18–26, 2022, doi: <http://dx.doi.org/10.26798/jiko.v6i1.504>.
- [10] N. A. Pratama, "Aplikasi Sistem Informasi Data Petani Dengan Optical Character Recognition Berbasis Android," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 1, p. 92, Feb. 2022, doi: 10.26798/jiko.v6i1.503.
- [11] A. E. S. Setyadji et al., "CAUSES OF INEFFECTIVE IMPLEMENTATION OF IT GOVERNANCE IN RISK MANAGEMENT: A SYSTEMATIC LITERATURE REVIEW," *JIKO (Jurnal Informatika dan Komputer)*, vol. 6, no. 2, 2023, doi: 10.33387/jiko.v6i2.6182.
- [12] A. F. Gad, "PyGAD: An Intuitive Genetic Algorithm Python Library," *CoRR*, vol. abs/2106.06158, 2021, doi: 10.48550/arXiv.2106.06158.
- [13] S. N. Kapita, "Aplikasi Jaringan Syaraf Tiruan Kohonen Self Organizing Map (K-SOM) pada Data Mutu Sekolah," *JIKO (Jurnal Informatika Dan Komputer)*, vol. 3, no. 1, pp. 56–61, 2020, doi: 10.33387/jiko.v3i1.1737.
- [14] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.
- [15] N. Shanmugasundaram, K. Sushita, S. P. Kumar, and E. N. Ganesh, "Genetic algorithm-based road network design for optimising the vehicle travel distance," *International Journal of Vehicle Information and Communication Systems*, vol. 4, no. 4, pp. 344–354, 2019, doi: 10.1504/IJVICS.2019.103931.
- [16] B. V Natesha and R. M. R. Guddeti, "Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment," *Journal of Network and Computer Applications*, vol. 178, p. 102972, 2021, doi: 10.1016/j.jnca.2020.102972.
- [17] A. Arram and M. Ayob, "A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems," *Comput Ind Eng*, vol. 133, pp. 267–274, 2019, doi: 10.1016/j.cie.2019.05.012.
- [18] S. Stajkowski, D. Kumar, P. Samui, H. Bonakdari, and B. Gharabaghi, "Genetic-Algorithm-Optimized Sequential Model for Water Temperature Prediction," *Sustainability*, vol. 12, no. 13, 2020, doi: 10.3390/su12135374.