

SKRIPSI

**ALGORITMA GENETIKA UNTUK PENYELESAIAN PERMASALAHAN
ALOKASI DOSEN PEMBIMBING TESIS**

***COMPLETION OF THESIS SUPERVISOR ALLOCATION PROBLEM
USING GENETIC ALGORITHM***



**Galuh Kirana Miftahuljannah
16/398507/PA/17468**

**PROGRAM STUDI S1 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2020**

SKRIPSI

ALGORITMA GENETIKA UNTUK PENYELESAIAN PERMASALAHAN ALOKASI DOSEN PEMBIMBING TESIS

COMPLETION OF THESIS SUPERVISOR ALLOCATION PROBLEM USING GENETIC ALGORITHM

Diajukan untuk memenuhi salah satu syarat memperoleh derajat
Sarjana Komputer



Galuh Kirana Miftahuljannah
16/398507/PA/17468

**PROGRAM STUDI S1 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2020**

HALAMAN PENGESAHAN

SKRIPSI

ALGORITMA GENETIKA UNTUK PENYELESAIAN PERMASALAHAN ALOKASI DOSEN PEMBIMBING TESIS

Telah dipersiapkan dan disusun oleh

Galuh Kirana Miftahuljannah
16/398507/PA/17468

Telah dipertahankan di depan tim penguji
pada tanggal 13 Mei, 2020

Susunan tim penguji

Aina Musdholifah, S.Kom., M.Kom., Ph.D
Pembimbing

Retantyo Wardoyo, Drs. M.Sc. Ph.D
Ketua Penguji

Sri Mulyana, Drs, M.Kom
Penguji

*Karya ini dipersembahkan untuk keluarga tersayang,
teman-teman serta rekan MBUGM dan Ilmu Komputer UGM 2016,
serta para pembaca sekalian.*

*“Fokus, disiplin, gerak cepat, percaya dengan orang depan,
dan peka terhadap tepukan.”*

*“Berlatihlah ibarat itu penampilanmu, dan tampil lah ibarat itu latihanmu”
“Tanpamu, display-ku tak berbentuk”*

-MBUGM-

“… でもね、会えたよ！ すてきな天使に
卒業は終わりじゃない
これからも仲間だから…”

-Houkago Tea Time-

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan dibawah ini:

Nama : Galuh Kirana Miftahuljannah
NIM : 16/398507/PA/17468
Tahun terdaftar : 2016
Program Studi : Ilmu Komputer
Fakultas : Matematika dan Ilmu Pengetahuan Alam

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah Skripsi ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 13 Mei 2020



Galuh Kirana Miftahuljannah

PRAKATA

Puji dan syukur kehadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penulis dapat menyusun dan menyelesaikan skripsi berjudul “Algoritma Genetika untuk Penyelesaian Permasalahan Alokasi Dosen Pembimbing Tesis”.

Selama proses pelaksanaan penelitian dan penyusunan tugas akhir, penulis telah mendapatkan banyak bimbingan, bantuan, serta dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, dengan segenap hati, penulis mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Kedua orang tua serta keluarga besar penulis yang selalu memberikan dukungan secara moral maupun materi kepada penulis selama menempuh pendidikan di Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.
2. Bapak Prof. Dr. Triyono, S.U. selaku Dekan di Fakultas Matematika dan Ilmu Pengetahuan Alam.
3. Bapak Dr. Suprapto sebagai Ketua Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Ibu Aina Musdholifah, S.Kom., M.Kom., Ph.D sebagai Dosen Pembimbing Skripsi yang telah membimbing penulis dalam menyelesaikan Skripsi ini.
5. Seluruh dosen Ilmu Komputer Universitas Gadjah Mada yang telah memberikan ilmu dan pengetahuan selama penulis menjalani perkuliahan di Universitas Gadjah Mada.
6. Seluruh anggota Marching Band Universitas Gadjah Mada yang telah memberikan pengalaman, pelajaran, dan kenangan yang berarti sehingga masa perkuliahan penulis menjadi lebih bermakna.
7. Sahabat dan saudara seperjuangan TUNEW yang telah menemani, menyemangati, dan berjuang bersama dari jenjang SMP hingga satu-

persatu menyelesaikan jenjang kuliah dan mendapatkan gelar masing-masing. Semoga ikatan yang dimiliki akan terus mengikat kuat.

8. Fikri Nurqahhari Priambodo selaku rekan yang telah menemani, mendukung, dan menyemangati dari awal perkuliahan.
9. Seluruh Teman KKN-PPM UGM Noborejo 2019 yang telah membahagiakan ketika sedih, menyemangati ketika sedang terpuruk, serta mengurus ketika sakit selama penulis menjalani KKN.
10. Seluruh rekan Ilmu Komputer Angkatan 2016 yang saling membantu dan saling mengingatkan dalam perkuliahan.
11. Pihak-pihak lain yang tidak dapat disebutkan satu-persatu.

Penulis sadar bahwa skripsi ini memiliki banyak kekurangan dan membutuhkan kritik dan saran yang dapat menyempurnakan skripsi ini sehingga dapat berguna dan turut andil dalam membantu perkembangan di bidang teknologi dan informasi. Atas perhatiannya, penulis mengucapkan terima kasih.

Yogyakarta, 13 Mei 2020

Penulis

DAFTAR ISI

| | |
|-----------------------------------------------------|------|
| PRAKATA | i |
| DAFTAR ISI..... | iii |
| DAFTAR GAMBAR | vi |
| DAFTAR TABEL..... | viii |
| DAFTAR LAMPIRAN..... | ix |
| INTISARI..... | xi |
| ABSTRACT..... | xii |
| | |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang Masalah | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan Penelitian..... | 3 |
| 1.5 Manfaat Penelitian..... | 4 |
| 1.6 Metodelogi Penelitian..... | 4 |
| BAB II TINJAUAN PUSTAKA..... | 5 |
| BAB III LANDASAN TEORI..... | 13 |
| 3.1 Permasalahan Alokasi Sumber Daya | 13 |
| 3.2Algoritma Metaheuristik | 13 |
| 3.3 Algoritma Genetika | 15 |
| 3.3.1 Representasi kromosom..... | 16 |
| Model representasi <i>real</i> | 17 |
| Model representasi permutasi | 17 |
| 3.3.2 Evaluasi nilai <i>fitness</i> | 19 |
| 3.4.3 Seleksi <i>parent</i> | 19 |
| 3.3.4 <i>Crossover</i> | 22 |
| 3.3.5 Mutasi | 24 |
| 3.3.6 Update generasi/seleksi <i>survivor</i> | 26 |
| 3.3.7 Kondisi pemberhentian | 27 |
| 3.4 Kromosom <i>Infeasible</i> | 27 |
| BAB IV ANALISIS DAN RANCANGAN..... | 29 |
| 4.1 Analisis Permasalahan..... | 29 |

| | |
|-----------------------------------------------------------------------------------|-----------|
| 4.2 Analisis Data | 30 |
| 4.3 Rancangan Algoritma Genetika | 31 |
| 4.3.1 Penyiapan data | 32 |
| 4.3.2 Perancangan parameter operasi algoritma genetika..... | 34 |
| 4.3.3 Representasi (pengkodean) kromosom..... | 35 |
| 4.3.4 Evaluasi nilai <i>fitness</i> | 38 |
| 4.3.5 Seleksi <i>parent</i> | 40 |
| 4.3.6 <i>Crossover</i> | 41 |
| 4.3.7 Mutasi | 43 |
| 4.3.8 Seleksi <i>survivor</i> | 45 |
| 4.3.9 Kondisi pemberhentian | 45 |
| 4.4 Perancangan Pengujian Sistem..... | 46 |
| 4.5 Evaluasi Kinerja Sistem | 46 |
| BAB V IMPLEMENTASI..... | 48 |
| 5.1 Spesifikasi Software dan Hardware | 48 |
| 5.2 Implementasi Proses..... | 48 |
| 5.2.1 Implementasi proses pembacaan data..... | 48 |
| 5.2.2 Deklarasi parameter pengujian algoritma genetika | 49 |
| 5.2.3 Implementasi <i>generate</i> kromosom..... | 49 |
| 5.2.3 Implementasi pembangkitan populasi | 51 |
| 5.2.4 Implementasi fungsi <i>fitness</i> | 51 |
| 5.2.5 Implementasi seleksi <i>parent</i> | 52 |
| 5.2.6 Implementasi <i>crossover</i> | 54 |
| 5.2.7 Implementasi mutasi | 56 |
| 5.2.8 Implementasi seleksi survivor | 58 |
| 5.2.9 Implementasi <i>cycle</i> algoritma genetika dan kondisi pemberhentian | 59 |
| 5.2.10 Implementasi evaluasi..... | 62 |
| 5.3 <i>Main Function</i> | 62 |
| BAB VI HASIL DAN PEMBAHASAN | 64 |
| 6.1 Pengujian dengan Nilai Parameter Algoritma Genetika | 64 |
| 6.1.1 Pengujian parameter <i>Pc</i> | 64 |
| 6.1.2 Pengujian parameter ukuran populasi | 66 |
| 6.1.3 Pengujian parameter generasi maksimum | 69 |

| | |
|-----------------------------------------------------------------------------|----|
| 6.2 Hasil Pencarian Solusi..... | 71 |
| 6.2.1 Hasil pencarian solusi dataset periode 2018/2019 semester genap | 72 |
| 6.2.2 Hasil pencarian solusi dataset periode 2019/2020 semester genap | 75 |
| 6.3 Evaluasi Kinerja Sistem | 78 |
| 6.3.1 Evaluasi solusi periode 2018/2019 semester genap..... | 78 |
| 6.3.2 Evaluasi solusi periode 2019/2020 semester genap..... | 81 |
| BAB VII PENUTUP | 84 |
| 7.1 Kesimpulan..... | 84 |
| 7.2 Saran | 84 |
| DAFTAR PUSTAKA | 85 |

DAFTAR GAMBAR

| | |
|---------------------------------------------------------------------------------------------------------------------------------|----|
| Gambar 3. 1 : Flowchart Algoritma Genetika (Goldberg et al, 1989) | 16 |
| Gambar 3. 2 : Contoh model kromosom representasi biner..... | 16 |
| Gambar 3. 3 : Contoh model kromosom representasi integer..... | 17 |
| Gambar 3. 4 : Contoh model representasi kromosom bilangan <i>real</i> | 17 |
| Gambar 3. 5 : Contoh model representasi kromosom permutasi | 17 |
| Gambar 3. 6 : Contoh jenis-jenis representasi kromosom 2-dimensi | 18 |
| Gambar 3. 7 : Contoh model representasi <i>tree</i> | 18 |
| Gambar 3. 8 : Ilustrasi <i>roulette wheel selection</i> | 20 |
| Gambar 3. 9 : Ilustrasi Baker SUS ^s | 21 |
| Gambar 3. 10 : Ilustrasi pertukaran informasi genetik pada reproduksi biologi... | 22 |
| Gambar 3. 11 : Contoh proses 1-point crossover kromosom representasi biner (kiri) dan pada representasi integer (kanan)..... | 23 |
| Gambar 3. 12 : Contoh proses N-point crossover..... | 23 |
| Gambar 3. 13 : Contoh proses Uniform crossover..... | 24 |
| Gambar 3. 14 Contoh proses Binary mutation..... | 25 |
| Gambar 3. 15 : Contoh proses mutasi swap | 25 |
| Gambar 3. 16 : Contoh proses random mutation | 26 |
| Gambar 4. 1 Alur rancangan algoritma genetika | 32 |
| Gambar 4. 2c Contoh array bidang riset | 33 |
| Gambar 4. 2b Contoh array mahasiswa | 33 |
| Gambar 4. 2a Contoh array dosen..... | 33 |
| Gambar 4. 3 Contoh array kuota tersedia setiap dosen..... | 33 |
| Gambar 4. 4 Contoh matriks bidang dosen | 33 |
| Gambar 4. 5 Contoh matriks bidang riset pilihan mahasiswa..... | 34 |
| Gambar 4. 6 Contoh kromosom | 36 |
| Gambar 4. 7 Diagram alur proses pengkodean dan pembangkitan kromosom (C _k) | 38 |
| Gambar 4. 8 Contoh pola uniform crossover | 41 |
| Gambar 4. 9 Diagram alur proses mutasi..... | 44 |
| Gambar 5. 1 Kode pembacaan data..... | 49 |
| Gambar 5. 2 Deklarasi variabel global | 49 |
| Gambar 5. 3 Proses generate kromosom | 50 |
| Gambar 5. 4 Fungsi pengecekan constrain | 51 |
| Gambar 5. 5 Kode proses pembangkitan populasi..... | 51 |
| Gambar 5. 6 Kode fungsi fitness | 52 |
| Gambar 5. 7 Sort kromosom berdasarkan nilai fitness | 53 |
| Gambar 5. 8 Kode seleksi parent..... | 53 |
| Gambar 5. 9 Kode pembangkitan pola random | 54 |
| Gambar 5. 10 Proses pengecekan nilai random dengan Pc | 55 |
| Gambar 5. 11a Kode uniform crossover | 55 |
| Gambar 5. 11b Proses perbaikan constrain usulan mahasiswa | 56 |

| | |
|-----------------------------------------------------------------------------------------------------|----|
| Gambar 5. 12 Kode fungsi pengecekan sisa kuota dosen pada kromosom <i>offspring</i> | 56 |
| Gambar 5. 13 Pengecekan <i>infeasible</i> kromosom | 57 |
| Gambar 5. 14 Kode proses mutasi | 58 |
| Gambar 5. 15 Kode proses seleksi <i>survivor</i> | 59 |
| Gambar 5. 16a Kode <i>cycle</i> algoritma genetika jika kondisi pemberhentian tidak terpenuhi | 60 |
| Gambar 5. 17 Kode untuk menampilkan hasil/solusi | 61 |
| Gambar 5. 18 Kode evaluasi hasil akhir/solusi..... | 62 |
| Gambar 5. 19 Kode main function..... | 63 |
| Gambar 6. 1 Grafik nilai <i>fitness</i> solusi pengujian parameter Pc..... | 66 |
| Gambar 6. 2 Grafik durasi pengujian paramter Pc..... | 66 |
| Gambar 6. 3 Grafik durasi pengujian parameter ukuran populasi | 68 |
| Gambar 6. 4 Grafik nilai <i>fitness</i> pengujian parameter ukuran populasi..... | 68 |
| Gambar 6. 5 Grafik durasi pengujian generasi maksimum..... | 70 |
| Gambar 6. 6 Grafik nilai <i>fitness</i> pengujian generasi maksimum | 71 |

DAFTAR TABEL

| | |
|----------------------------------------------------------------------------------------------------------------------------------|----|
| Tabel 2. 1 Tinjauan Pustaka | 9 |
| Tabel 4. 1 Sampel data dosen pembimbing tesis usulan mahasiswa | 29 |
| Tabel 4. 3 Sampel dataset dosen | 30 |
| Tabel 4. 4 Sampel dataset mahasiswa | 31 |
| Tabel 4. 5 Sampel bidang riset tersedia | 31 |
| Tabel 4. 6 Analisis kromosom | 36 |
| Tabel 4. 7 Contoh proses evaluasi nilai <i>fitness</i> | 39 |
| Tabel 4. 8 Contoh seleksi <i>parent</i> dengan Baker's SUS..... | 40 |
| Tabel 4.9 Parameter pengujian..... | 46 |
| Tabel 6. 1 Durasi pengujian parameter Pc | 65 |
| Tabel 6. 2 Nilai <i>fitness</i> pengujian parameter Pc..... | 65 |
| Tabel 6. 3 Durasi pengujian parameter ukuran populasi..... | 67 |
| Tabel 6. 4Nilai <i>fitness</i> pengujian parameter ukuran populasi | 67 |
| Tabel 6. 5 Durasi pengujian parameter generasi maksimum | 69 |
| Tabel 6. 6 Nilai <i>fitness</i> pengujian parameter generasi maksimum..... | 70 |
| Tabel 6. 7 Hasil pengujian paramter algoritma genetika..... | 71 |
| Tabel 6. 8 Hasil pencarian solusi usulan dosen dataset periode 2018/2019 semester genap | 72 |
| Tabel 6. 9 Daftar pasangan mahasiswa dan dosen dengan bidang riset yang cocok dataset periode 2018/2019 semester genap | 74 |
| Tabel 6. 10 Hasil pencarian solusi usulan dosen dataset periode 2019/2020 semester genap | 75 |
| Tabel 6. 11 Daftar pasangan mahasiswa dan dosen dengan bidang riset yang cocok dataset periode 2019/2020 semester genap | 77 |
| Tabel 6. 12 Sampel hasil keputusan rapat periode 2018/2019 semester genap | 79 |
| Tabel 6. 13 Solusi algoritma genetika yang sama dengan hasil keputusan rapat periode 2018/2019 semester genap | 79 |
| Tabel 6. 14 Sisa kuota dosen solusi hasil algoritma genetika periode 2018/2019 semester genap | 79 |
| Tabel 6. 15 Sampel hasil keputusan rapat periode 2019/2020 semester genap | 81 |
| Tabel 6. 16 Solusi algoritma genetika yang sama dengan hasil keputusan rapat periode 2019/2020 semester genap | 81 |
| Tabel 6. 17 Sisa kuota dosen solusi hasil algoritma genetika periode 2018/2019 semester genap | 82 |

DAFTAR LAMPIRAN

| | |
|----------------------------------------------------------------------------|------------|
| LAMPIRAN A | 88 |
| DAFTAR BIDANG MINAT DOSEN PROGRAM MAGISTER DIKE | |
| LAMPIRAN B | 90 |
| TEMA BIDANG RISET DIKE PROGRAM MAGISTER | |
| LAMPIRAN C | 92 |
| DATA DOSEN DAN KUOTA TERSEDIA PERIODE 2018/2019 SEMESTER GENAP | |
| LAMPIRAN D | 93 |
| TOPIK RISET PILIHAN MAHASISWA PERIODE 2018/2019 SEMESTER GENAP | |
| LAMPIRAN E | 95 |
| USULAN DOSEN PEMBIMBING TESIS PERIODE 2018/2019 SEMESTER GENAP | |
| LAMPIRAN F | 98 |
| TOTAL PENGUSUL DOSEN PEMBIMBING TESIS PERIODE 2018/2019 SEMESTER GENAP | |
| LAMPIRAN G | 99 |
| DATA DOSEN DAN KUOTA TERSEDIA PERIODE 2019/2020 SEMESTER GENAP | |
| LAMPIRAN H | 100 |
| TOPIK RISET PILIHAN MAHASISWA PERIODE 2019/2020 SEMESTER GENAP | |
| LAMPIRAN I | 102 |
| USULAN DOSEN PEMBIMBING TESIS PERIODE 2019/2020 SEMESTER GENAP | |
| LAMPIRAN J | 105 |
| TOTAL PENGUSUL DOSEN PEMBIMBING TESIS PERIODE 2019/2020 SEMESTER GENAP | |
| LAMPIRAN K | 106 |
| HASIL RAPAT PENUGASAN DOSEN PEMBIMBING PERIODE 2018/2019 SEMESTER GENAP | |
| LAMPIRAN L | 108 |
| HASIL RAPAT PENUGASAN DOSEN PEMBIMBING PERIODE 2019/2020 PERIODE GENAP | |
| LAMPIRAN M | 110 |
| KROMOSOM SOLUSI PERIODE 2018/2019 SEMESTER GENAP | |
| LAMPIRAN N | 111 |
| KROMOSOM SOLUSI PERIODE 2019/2020 SEMESTER GENAP | |
| LAMPIRAN O | 112 |
| PENGKODEAN KROMOSOM HASIL KEPUTUSAN RAPAT PERIODE 2018/2019 SEMESTER GENAP | |

LAMPIRAN P 113
PENGODEAN KROMOSOM HASIL KEPUTUSAN RAPAT PERIODE
2019/2020 SEMESTER GENAP

INTISARI

ALGORITMA GENETIKA UNTUK PENYELESAIAN PERMASALAHAN ALOKASI DOSEN PEMBIMBING TESIS

Oleh:

Galuh Kirana Miftahuljannah

16/398507/PA/17468

Berdasarkan Dokumen Kurikulum 2017 Program Magister (S2), mahasiswa Departemen Ilmu Komputer dan Elektronika (DIKE) jenjang S2 Universitas Gadjah Mada untuk dapat lulus dan memperoleh gelar M.Cs. harus mampu menyelesaikan tesis sebagai salah satu mata kuliah wajib.

Pada pendaftaran tesis periode 2018/2019 semester genap dan periode 2018/2019 semester genap, terdapat sekitar 54% dosen pembimbing tesis yang diusulkan mahasiswa melebihi kuota dosen yang tersedia. Dalam hasil keputusan rapat penugasan dosen pembimbing kedua periode tersebut, hanya 14% mahasiswa mendapatkan dosen pembimbing dengan bidang minat yang cocok dengan topik tesis yang diambil. Penelitian ini bertujuan untuk menyelesaikan permasalahan alokasi dosen pembimbing tesis yang dapat digunakan sebagai usulan dalam pengambilan keputusan rapat penugasan dosen pembimbing tesis menggunakan algoritma genetika.

Penelitian ini sudah mampu menghasilkan solusi yang valid dengan tingkat kecocokan bidang riset dosen dan mahasiswa mencapai 31% kecocokan untuk usulan penugasan dosen pembimbing pada data periode 2018/2019 semester genap dan mencapai 33% untuk periode 2019/2020 semester genap.

Kata Kunci: Algoritma genetika, metaheuristik, permasalahan alokasi sumber daya

ABSTRACT

COMPLETION OF THESIS SUPERVISOR ALLOCATION PROBLEM USING GENETIC ALGORITHM

By:

Galuh Kirana Miftahuljannah

16/398507/PA/17468

Based on the 2017's Curriculum Documents of the Masters Program (S2), students of the Department of Computer Science and Electronics of Universitas Gadjah Mada, to be able to graduate and obtain M.Cs. degree must be able to complete a thesis as one of the compulsory subjects.

In the thesis registration 2018/2019 period in even semester and 2018/2019 period even semester, there are around 54% of the thesis supervisors proposed by students exceeding the available lecturer quota. In the results of the decision of the both period's lecturer supervisors assignment meeting, only 14% of students received a supervisor with a field of interest that matched the topic of the thesis taken. This study aims to resolve the issue of the allocation of the thesis supervisor which can be used as a suggested solution in the decisions making of the thesis supervisor assignment meeting by using genetic algorithm.

This research has been able to produce a valid solution with a percentage of compatibility in the field of lecturer's research and students reach 31% for the proposed assignment of supervisors in the 2018/2019 period even semester and reach 33% for the 2019/2020 period even semester.

Keywords : Genetic algorithm, metaheuristic, resource allocation problem

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Berdasarkan Dokumen Kurikulum 2017 Program Magister (S2), mahasiswa Departemen Ilmu Komputer dan Elektronika (DIKE) jenjang S2 Universitas Gadjah Mada untuk dapat lulus dan memperoleh gelar M.Cs., mahasiswa harus telah menyelesaikan 44 sks mata kuliah yang terdiri dari 23 sks mata kuliah wajib dan 21 sks mata kuliah pilihan. Tugas akhir yang berupa tesis merupakan salah satu dari mata kuliah wajib yang harus diselesaikan. Mahasiswa yang sedang mengerjakan tesis akan dibimbing oleh setidaknya satu dosen dalam proses pengerjaannya. Mahasiswa akan dibimbing oleh dosen pilihan dengan bidang riset penelitian yang sesuai dengan topik tugas tesis yang dikerjakan.

Permasalahan yang sering terjadi dalam pemilihan dosen pembimbing adalah ketidak merataan antara jumlah mahasiswa yang mengusulkan ke satu dosen tertentu dengan kuota tersedia atau kuota maksimal bimbingan dosen tersebut. Berdasarkan data pendaftaran tesis periode 2018/2019 semester genap dan data pendaftaran tesis periode 2019/2020, terdapat 54% dosen diusulkan melebihi kuota tersedia. Permasalahan lain adalah ketidaksamaan antara topik bidang riset tugas tesis yang diusulkan mahasiswa dengan bidang riset dosen yang diusulkan. Prosedur yang biasa dilakukan setelah mahasiswa melakukan proses pendaftaran tesis adalah dengan melakukan rapat penugasan dosen pembimbing untuk mahasiswa agar tidak terjadi kelebihan kuota dosen dengan mempertimbangkan topik tesis mahasiswa dengan bidang minat/bidang riset dosen pembimbing. Pada hasil keputusan rapat penugasan dosen pembimbing periode 2018/2019 genap dan periode 2019/2020 semester genap, persentase mahasiswa yang mendapatkan dosen dengan bidang minat yang cocok hanya mencapai sekitar 14%, sehingga diperlukan sebuah sistem yang dapat membantu proses pengambilan keputusan dalam rapat tersebut untuk mencapai persentase kecocokan yang lebih tinggi.

Pemilihan dosen pembimbing tesis dapat dikategorikan sebagai permasalahan alokasi sumber daya. Permasalahan alokasi sumber daya adalah proses untuk mengalokasikan sumber daya pada berbagai projek atau unit dengan tujuan memaksimalkan profit atau meminimalkan biaya (Lin dan Gen, 2008). Tujuan akhir dari proses alokasi sumber daya adalah untuk mencari alokasi yang optimal dengan sumber daya yang terbatas untuk tugas atau unit tertentu agar mendapatkan profit yang optimal. Sumber daya dapat berupa sumber daya manusia, barang, modal atau asset yang dapat digunakan untuk mencapai tujuan akhir. Beberapa penelitian sebelumnya tentang permasalahan alokasi sumber daya antara lain: Mencari solusi untuk penempatan kontainer dengan jumlah yang maksimal pada lahan yang terbatas di Pelabuhan Singapura (Chen dkk, 2002), pembagian modal kepada media iklan tertentu secara efisien sehingga mendapatkan untung/profit yang maksimal (Wang dkk, 2018), alokasi mahasiswa dengan jumlah dosen yang terbatas untuk pengawasan projek (Salami dan Mamman, 2016), memberikan satu tugas dari pilihan tugas yang tersedia kepada murid sesuai dengan tingkatan referensi tugas yang diinginkan (Chown dkk, 2018), dan alokasi mahasiswa medis ke rumah sakit yang tersedia untuk melakukan praktik medis sesuai dengan spesialisasi masing-masing mahasiswa (S.N Cresswell, 2016).

Algoritma genetika merupakan salah satu algoritma metaheuristik, yaitu sebuah algoritma untuk menyelesaikan suatu permasalahan untuk mendapatkan solusi yang optimal dan mudah beradaptasi sesuai dengan permasalahan yang akan diselesaikan (Boussaï dkk, 2013). Algoritma genetika diusulkan oleh Holland pada tahun 1975. Holland mengadaptasi konsep evolusi alam dan genetika. Metode ini didasarkan atas prinsip Darwin : “*survival of the fittest*”, yang berarti suatu individu akan bertahan pada generasi selanjutnya apabila memiliki kualitas yang baik. Pada algoritma genetika, individu-individu ini merupakan kumpulan solusi. Calon individu yang akan bertahan sampai generasi selanjutnya dipilih melalui proses seleksi *parent*, *crossover*, mutasi, dan seleksi *survivor*.

Penelitian ini akan fokus untuk menemukan solusi yang optimal untuk permasalahan pemilihan dosen pembimbing tesis untuk mahasiswa S2 DIKE Universitas Gadjah Mada dengan menggunakan algoritma genetika.

1.2 Rumusan Masalah

Dalam proses pemilihan dosen pembimbing tesis oleh mahasiswa terdapat permasalahan yaitu:

1. Adanya ketidakmerataan antara jumlah dosen yang diusulkan oleh mahasiswa dengan kuota tersedia masing-masing dosen.
2. Adanya ketidaksesuaian antara bidang minat/bidang riset dosen yang diusulkan dengan topik tesis mahasiswa.

1.3 Batasan Masalah

Batasan-batasan dan aturan yang digunakan untuk penelitian adalah sebagai berikut:

1. Fokus untuk permasalahan dan data yang digunakan adalah data pendaftaran tesis Departemen Ilmu Komputer dan Elektronika Universitas Gadjah Mada periode 2018/2019 semester genap dan periode 2019/2020 periode genap.
2. Setiap mahasiswa minimal mengusulkan 1 dosen dan maksimal mengusulkan 3 dosen.
3. Satu dosen dapat memiliki bidang riset penelitian lebih dari satu.
4. Setiap mahasiswa wajib dibimbing minimal 1 dosen dan Setiap mahasiswa dapat dibimbing sampai dengan 2 dosen.
5. Mahasiswa dikatakan mengusulkan dosen pembimbing yang cocok apabila minimal mengusulkan 1 dosen dengan bidang minat yang cocok dengan topik tesis.
6. Kuota tersedia dosen adalah kuota yang tersedia dalam menerima mahasiswa yang baru mendaftar untuk tugas akhir/skripsi pada periode tertentu.

1.4 Tujuan Penelitian

Tujuan dilakukan penelitian ini adalah untuk membentuk sebuah sistem menggunakan algoritma genetika yang dapat memberikan usulan solusi alokasi dosen pembimbing tesis dengan mahasiswa, sehingga tidak terjadi kelebihan beban kuota dosen dan meningkatkan persentase kecocokan topik tesis mahasiswa dengan bidang minat/bidang riset dosen.

1.5 Manfaat Penelitian

Manfaat yang bisa didapatkan dari melakukan penelitian ini adalah hasil yang diperoleh dapat digunakan sebagai rekomendasi untuk membantu proses rapat dalam mengambil keputusan penugasan dosen pembimbing tesis.

1.6 Metodelogi Penelitian

Permasalahan alokasi dosen pembimbing tesis akan diselesaikan menggunakan algoritma genetika.

BAB II

TINJAUAN PUSTAKA

Penelitian yang sudah dilakukan sebelumnya untuk mengatasi permasalahan alokasi adalah dengan menggunakan algoritma metaheuristik. Algoritma metaheuristik adalah algoritma untuk menyelesaikan permasalahan optimalisasi solusi tanpa memerlukan adaptasi yang sulit untuk setiap permasalahan yang akan diselesaikan (Boussaïd dkk, 2013). Istilah heuristik pertama kali diungkapkan oleh Fred Glover pada tahun 1986. yang mengusulkan *tabu search*. Penelitian-penelitian yang berkontribusi pada perkembangan algoritma metaheuristik diantaranya adalah usulan metode *simulated annealing* oleh Kirkpatrick pada tahun 1982. Pada tahun 1988 Koza mendapatkan hak paten untuk *genetic programming* dan berhasil dipublikasikan pada tahun 1992. Pada tahun 1989, David Goldberg mempublikasikan buku yang membahas *genetic algorithm* (algoritma genetika).

Beberapa penelitian yang sudah menggunakan algoritma metaheuristik diantaranya adalah *hybrid tabu search* dengan SWO (*Squeaky Wheel Optimization*) untuk permasalahan alokasi ruang yang bertujuan memaksimalkan alokasi sejumlah barang pada ruang yang terbatas, sehingga penggunaan ruang menjadi lebih efisien. Penelitian ini diaplikasikan untuk alokasi sejumlah kontainer pada lapangan yang terbatas di pelabuhan Singapura (Chen dkk, 2002). Penelitian lain adalah menggunakan *reinforcement learning* untuk alokasi modal kepada berbagai media iklan yang tersedia dengan biayanya masing-masing dengan tujuan untuk mendapatkan profit yang maksimal (Wang dkk, 2018). Terdapat juga alokasi untuk sumber daya manusia. Alokasi sumber daya manusia bertujuan untuk memaksimalkan penyebaran sumber daya manusia yang terbatas yang ada kepada beberapa unit kerja yang tersedia. Berbagai penelitian yang berfokus pada permasalahan alokasi sumber daya manusia diantaranya alokasi penugasan dosen kepada mahasiswa sebagai pengawas penggerjaan projek (Salami dan Mamman, 2016) menggunakan *genetic algorithm*, memberikan satu tugas dari berbagai pilihan tugas yang tersedia kepada sejumlah murid sesuai dengan tingkatan

referensi tugas yang diinginkan (Chown dkk, 2018) menggunakan *simulated annealing*, dan alokasi mahasiswa bidang medis ke berbagai rumah sakit untuk melakukan pelatihan praktik medis sesuai dengan spesialisasi masing-masing mahasiswa (S.N Cresswell, 2016), dimana pada penelitiannya S.N Cresswell mengusulkan sebuah metode *heuristic* baru. Alokasi sumber daya biasanya bertujuan untuk mendapatkan kualitas produk yang baik dari suatu pekerjaan atau proyek yang dilakukan.

Algoritma genetika adalah salah satu dari algoritma metaheuristik yang diadaptasi dari konsep evolusi alam dan genetika (Holland, 1975). Metode ini didasarkan atas prinsip Darwin : “*Survival of the fittest*”, dimana suatu individu akan bertahan pada generasi selanjutnya apabila individu tersebut memiliki kualitas yang lebih baik dibandingkan dengan individu-individu lain. Sivanandam dan Deepa (2008) , menyatakan beberapa perbedaan antara algoritma genetika dengan metode penyelesaian optimalisasi lainnya, antara lain sebagai berikut:

1. Algoritma genetika bekerja dengan pengkodean berbagai set solusi, bukan dengan solusi akhir masalah itu sendiri.
2. Metode optimalisasi biasanya mencari solusi dimulai dari satu titik dan bergerak mendekati solusi akhir. Algoritma genetika selalu beroperasi pada seluruh populasi solusi dan bukan dengan suatu solusi tunggal untuk melakukan pencarian.
3. Algoritma genetika menggunakan fungsi *fitness* untuk evaluasi kualitas solusi, bukan dengan menggunakan *derivative/turunan*, sehingga dapat diterapkan untuk segala jenis permasalahan optimasi kontinu atau diskrit.

Dari sifat-sifat algoritma genetika tersebut, dapat disimpulkan kelebihan penggunaan algoritma genetika, yaitu (Sivanandam dan Deepa, 2008):

1. Setiap kromosom/individu dapat diproses dan dianalisis secara parallel, sehingga durasi penyelesaian akan berbanding terbalik dengan jumlah processor yang digunakan untuk memproses kromosom/individu dalam pencarian solusi

2. Sangat dinamis terhadap perubahan untuk menyesuaikan dengan permasalahan yang akan diselesaikan. Setiap pengkodean untuk representasi kromosom/individu dapat diproses dengan metode yang berbeda-beda agar dapat menggambarkan masalah, batasan, dan solusi dengan baik.

Mahmudy (2006) menggunakan algoritma genetika untuk melakukan optimasi alokasi penugasan dosen ke kelas. Model penugasan (*assignment model*) merupakan salah satu kasus untuk model pemograman linear. Mahmudy menyatakan karakteristik khusus dari model ini cenderung membutuhkan pembatas (*constraint*) dari variabel-variabel sehingga penyelesaian dengan model matematis membutuhkan biaya komputasi yang mahal dan proses perhitungan yang panjang. Untuk mengatasi masalah pada kasus tersebut, Mahmudy mengatasinya dengan algoritma genetika, dimana diperoleh solusi dalam waktu cepat dibandingkan dengan pemograman linear.

Tsai dkk (2015) menyatakan bahwa representasi kromosom dalam bentuk linear tidak mampu untuk merepresentasikan beberapa kasus permasalahan, sementara suatu representasi kromosom harus bisa menggambarkan solusi permasalahan dengan baik dan *valid*. Maka digunakan representasi kromosom 2-dimensi dengan sebuah matriks untuk menyelesaikan permasalahan penjadwalan pesawat terbang. Contoh representasi dimensional lain adalah menggunakan representasi kromosom 2-dimensi untuk menyelesaikan masalah Ising (model fisika dalam termodinamika) (Anderson dkk, 1991) dan menyelesaikan permasalahan dalam pengaturan jadwal karyawan (Dean, 2008).

Salami dan Mamman (2016) menyelesaikan permasalahan alokasi dosen sebagai pengawas pengajaran projek mahasiswa. Pada permasalahan ini, mahasiswa memilih dosen untuk mengawasi projek yang akan dikerjakan, namun jumlah guru yang ada terbatas. Untuk mendapatkan solusi dari permasalahan ini, Salami dan Mamman mengembangkan suatu proses *crossover* dimana seluruh kromosom dalam populasi akan menjadi *parent* dan sepasang *parent* hanya akan menghasilkan satu kromosom *offspring*.

Abido dan Elazouni (2009), menyelesaikan masalah penjadwalan projek-projek. Pada prosesnya, Abido dan Elazouni menyatakan bahwa penyelesaian dengan algoritma genetika dapat menyebabkan terbentuknya kromosom-kromosom *offspring* yang melanggar batasan-batasan (*constraint*) dari representasi kromosom yang sudah ditentukan. Kromosom tersebut dikenal dengan istilah kromosom *infeasible* (tidak layak/cacat). Diusulkan mekasime perbaikan setelah proses *crossover* dengan mengadaptasi metode *conventional free float* (CFF) dan *backward free float* (BFF) dan menggunakan representasi kromosom CPM *network*, yang bertujuan untuk memperbaiki kromosom-kromosom tertentu yang *infeasible*. Metode perbaikan lain yang dilakukan adalah melakukan proses evaluasi nilai *penalty value* pada kromosom-kromosom *offspring*. Hanya kromosom-kromosom dengan *penalty value* tententu akan melakukan proses mutasi (Tsai dkk, 2015).

Perbandingan penelitian-penelitian sebelumnya mengenai permasalahan alokasi dan metode penyelesaiannya ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

| No | Penelitian | Tujuan | Permasalahan | Metode | | | | | |
|----|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-----------------------------|--------|-------------------------------------------------------------|----------------------|----------------------------------|-----------|-----------------|
| | | | | Non-GA | Algoritma Genetika (GA) | | | | |
| | | | | | Representasi Kromosom | Seleksi Parent | Crossover | Mutasi | Update Generasi |
| 1. | Penerapan Algoritma Genetika Pada Optimasi Model Penugasan (Mahmudy, 2006) | Alokasi dosen untuk mengajar pada kelas sesuai bidang minat | Alokasi sumber daya manusia | | Integer | Probabilistik | 1-point | Swap | Elitisme |
| 2. | Two Dimensional Genetic Algorithm and Its Application to Aircraft Scheduling Problem (Tsai dkk,2015) | Mengatur jadwal penerbangan pesawat terbang | Penjadwalan | | Permutasi Integer 2-dimensi (tidak disebutkan) | Substring | Penalty Value Repair dan swap | | Elitisme |
| 3. | A Two Dimensional Genetic Algorithm for the Ising Problem(Anderson dkk, 1991) | Mencari solusi untuk masalah Ising (Termodinami-ka) | Model Fisika | | Bilangan Real 2-dimensi dengan grid vertical dan horizontal | Random dari gen pool | Block Uniform | inversion | Elitisme |

Tabel 2. 1 Tinjauan Pustaka

| | | | | | Biner 2-dimensi | Unbiased Tournament | Vertical 2- point | Mutasi kolom | Elitisme |
|----|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|---------------|-------------------------------------------------------------------|--------------------|------------------------|----------------------|--------------|----------|
| 4. | Staff Scheduling by a Genetic Algorithm with a Two-Dimensional Chromosome Structure (Dean, 2008) | Pembagian jadwal kerja kepada karyawan dengan aturan tertentu. | Penjadwalan | | | | | | |
| 5. | The Yard Allocation Problem (Chen dkk , 2002) | Pengaturan peletakan kontainer pada lahan yang terbatas pada pelabuhan Singapura | Alokasi ruang | Hybrid Tabu Search (TS) dengan Squeaky Wheel Optimizati -on (SWO) | | | | | |
| 6. | An Efficient Budget Allocation Algorithm for Multi-Channel Advertising (Wang dkk, 2018) | Penanaman model pada berbagai media iklan dengan budget yang tersedia untuk mendapatkan untung maksimal. | Alokasi modal | Q-MKCP | | | | | |

Tabel 2. 1 Tinjauan Pustaka

| | | | | | | | | |
|----|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|-----------------------------|-----------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------|----------|
| 7. | A Simulated Annealing Approach to The Student-Project Allocation Problem (Chown, dkk 2018) | Memberikan tugas kepada murid dari tugas-tugas yang tersedia berdasarkan tingkat minat masing-masing murid. | Alokasi sumber daya manusia | Simulated Annealing | | | | |
| 8. | A Genetic Algorithm for Allocating Project Supervisors to Student (Salami and Mamman, 2016) | Menentukan pasangan guru pengawas projek untuk murid dengan aturan tertentu. | Alokasi | Permutasi integer 2 dimensi | Semua kromosom terpilih untuk menjadi parent | Pertukaran dengan membandingkan nilai fitness kedua <i>parent</i> , dan hanya menghasil-kan satu kromosom <i>offspring</i> . | Swap | Elitisme |

Tabel 2. 1 Tinjauan Pustaka

| | | | | | | | | | |
|-----|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|-------------------------|------------------|------------------------------------------------------------------------------------------------|---------------------------------|------------------|--|
| 9. | A Hospital Placements Allocation Problem (C.N Creswell, 2005) | Alokasi mahasiswa bidang medis ke rumah sakit untuk melakukan pelatihan praktik medis sesuai spesialisasi mahasiswa | Penjadwalan dan Alokasi | Heuristic Program -ming | | | | | |
| 10. | Improved Crossover and Mutation Operators for Genetic Algorithm Project Scheduling. (Abido dan Elazouni, 2009) | Melakukan penjadwalan untuk projek-projek dan mengembang-kan sistem untuk mencegah terjadinya kromosom <i>infeasible</i> | Penjadwalan dan <i>infeasible</i> kromosom | <i>CPM network</i> | Tidak disebutkan | Operasi <i>backward path</i> dan <i>forward path</i> dengan mempertimbangkan nilai FF dan BFF) | Membanding-kan nilai FF dan BFF | Tidak disebutkan | |

BAB III

LANDASAN TEORI

3.1 Permasalahan Alokasi Sumber Daya

Permasalahan alokasi sumber daya adalah proses untuk mengalokasikan sumber daya pada berbagai projek atau unit dengan tujuan memaksimalkan profit atau meminimalkan biaya (Lin dan Gen, 2008). Tujuan akhir dari proses alokasi sumber daya adalah untuk mencari alokasi yang optimal dengan sumber daya yang terbatas untuk tugas atau unit tertentu agar mendapatkan profit yang optimal. Sumber daya dapat berupa sumber daya manusia, barang, modal atau asset yang dapat digunakan untuk mencapai tujuan akhir. Tujuan akhir dapat berupa suatu sasaran yang terdorong oleh kebutuhan finansial di masa depan. Solusi terbaik atau solusi yang optimal adalah solusi yang mampu memaksimalkan profit, meminimalkan biaya, atau memperoleh kualitas produk terbaik.

Alokasi dosen pembimbing tesis merupakan salah satu contoh permasalahan alokasi sumber daya. Dosen sebagai sumber daya yang terbatas akan ditugaskan untuk membimbing sejumlah mahasiswa. Setiap mahasiswa akan memilih dan mengusulkan bidang riset untuk tugas tesis. Dosen memiliki minat terhadap bidang riset yang berbeda-beda. Solusi dari permasalahan alokasi dosen pembimbing tugas tesis dikatakan baik apabila tidak ada dosen yang diusulkan mahasiswa melebihi jumlah kuota yang tersedia untuk setiap dosen serta setiap pasangan dosen pembimbing dan mahasiswa memiliki minat terhadap bidang riset yang sama.

3.2 Algoritma Metaheuristik

Dalam Blum dan Roli (2003), Bianchi menyatakan bahwa heuristik adalah algoritma metode pendekatan dasar pencarian dalam *search space* untuk memperoleh sebuah solusi dengan kualitas yang baik. Heuristik umumnya dibagi menjadi dua tipe, yaitu algoritma konstruktif dan algoritma *local search*.

Algoritma konstruktif membangun solusi dengan menggabungkan komponen-komponen dari sebuah solusi, yang ditambahkan secara satu per satu sampai solusinya lengkap. Algoritma *local search* memulai dari satu solusi dan mencoba memperbaiki solusi tersebut dengan memodifikasi beberapa komponennya untuk meningkatkan kualitas dari solusi tersebut. Dibandingkan dengan algoritma optimasi dan metode iterative, heuristik tidak dapat dipastikan dapat menemukan solusi yang optimal dari area global pada beberapa permasalahan.

Algoritma metaheuristik adalah algoritma yang didesain untuk menemukan solusi optimal dalam suatu permasalahan tanpa perlu adaptasi yang sulit untuk setiap permasalahan. Istilah ‘meta’ dari bahasa Yunani menandakan bahwa algoritma ini adalah tingkatan yang lebih tinggi dari metode heuristik. Jika dibandingkan dengan metode heuristik, algoritma metaheuristik dapat diaplikasikan untuk permasalahan yang tidak memiliki solusi khusus dalam pencarian solusi. Algoritma metaheuristik dapat menyelesaikan masalah alokasi untuk mendapatkan profit yang maksimal dengan biaya seminimum mungkin. (Boussaïd dkk, 2013) Contoh dari algoritma metaheuristik adalah *genetic algorithm* (algoritma genetika), *greedy random adaptive search*, *neural network*, *simulated annealing*, *tabu search*, *problem space search*, dan algoritma threshold.

Berbagai algoritma metaheuristic memiliki ciri-ciri yang sama, yaitu :

1. Terinspirasi dari prinsip alam. Beberapa dari algoritma metaheuristik mengadaptasi konsep fisika, biologi, atau etologi.
2. Menggunakan komponen *stochastic* (melibatkan suatu nilai variabel yang *random*).
3. Memiliki beberapa parameter yang perlu disesuaikan dengan permasalahan.

Sebuah algoritma metaheuristic dikatakan berhasil pada suatu optimalisasi solusi permasalahan apabila dapat memberikan keseimbangan antara eksplorasi dan eksploitasi solusi. Eksplorasi diperlukan untuk meningkatkan keberagaman solusi dengan memperluas area pencarian (diversity/keragaman) dan eksploitasi diperlukan untuk mengetahui bagian dari area pencarian yang memiliki solusi-solusi yang baik.

3.3 Algoritma Genetika

Algoritma genetika pertama kali diungkapkan oleh John Holland pada tahun 1975. Algoritma genetika merupakan algoritma pencarian yang mengadaptasi konsep evolusi alam dan genetika. Metode ini didasarkan atas prinsip Darwin : “*survival of the fittest*”, yang berarti suatu individu akan bertahan pada generasi selanjutnya apabila memiliki kualitas yang baik. Pada algoritma genetika, individu-individu ini merupakan kumpulan solusi. Calon individu yang akan bertahan sampai generasi berikutnya dipilih melalui proses seleksi orangtua, crossover, mutasi, dan seleksi generasi. Metode ini menggambarkan proses reproduksi dan memastikan bahwa solusi yang lebih baik tetap ada di generasi selanjutnya dan yang lebih lemah dihilangkan dari generasi selanjutnya, sampai ditemukannya suatu solusi optimal yang memenuhi kondisi tertentu. Algoritma genetika secara umum terdiri dari langkah-langkah berikut :

1. ***Generate kromosom/individu***

Menghasilkan populasi awal yang terdiri dari N buah kromosom/individu sebagai representasi solusi.

2. **Evaluasi nilai *fitness***

Evaluasi setiap solusi dari populasi awal menggunakan fungsi kebugaran (*fitness*)/fungsi objektif.

3. **Seleksi orangtua (*parent*)**

Pilih solusi sebagai orangtua (*parent*) untuk generasi baru berdasarkan probabilitas dari *fitness* atau secara *random*. Solusi dengan *fitness* yang terbaik memiliki kemungkinan lebih tinggi untuk terpilih daripada solusi buruk.

4. **Crossover**

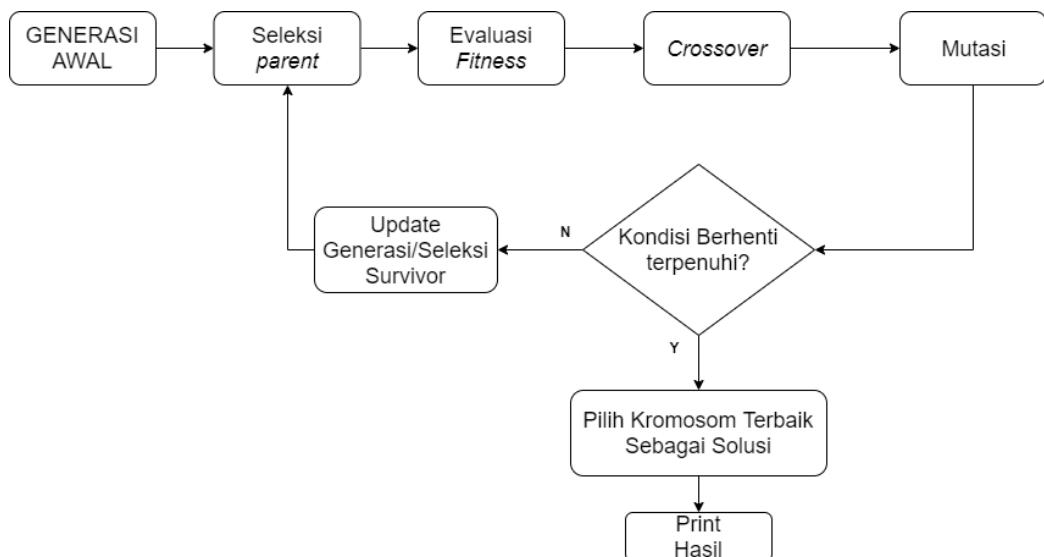
Menggunakan hasil dari tahap 3 untuk menghasilkan keturunan (*offspring*) dengan proses kawin silang (*crossover*).

5. **Mutasi**

Mengubah generasi baru dengan mutasi untuk meningkatkan keberagaman pada populasi atau untuk memenuhi aturan solusi tertentu.

6. Update Generasi/*Seleksi survivor*

Melakukan seleksi kromosom/individu pada populasi berdasarkan nilai *fitness* untuk generasi selanjutnya.



Gambar 3. 1 : Flowchart Algoritma Genetika (Goldberg et al, 1989)

Pada uraian selanjutnya, akan dipaparkan mengenai tahapan-tahapan yang terdapat dalam algoritma genetika, yaitu: representasi kromosom, evaluasi nilai *fitness* kromosom, seleksi *parent*, *crossover*, mutasi, seleksi *survivor* (*replacement*), serta kriteria pemberhentian (*termination process*).

3.3.1 Representasi kromosom

Pada algoritma genetika , setiap kromosom/individu dikodekan sebagai vektor panjang dengan ukuran tertentu yang terdiri dari gen-gen yang memiliki nilai tertentu. Beberapa model representasi kromosom antara lain:

Model representasi biner

Pada model representasi biner, nilai-nilai yang disimpan pada setiap gen merupakan bilangan biner (0 dan 1). Model representasi ini merupakan jenis representasi paling sederhana. Contoh model representasi biner ditunjukkan pada Gambar 3.2.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Gambar 3. 2 : Contoh model kromosom representasi biner

Model representasi integer

Pada model representasi integer, nilai-nilai yang tersimpan pada setiap gennya merupakan bilangan bulat (*integer*). Bilangan bulat pada gen model representasi integer dapat digunakan untuk merepresntasikan suatu urutan, posisi, jumlah, atau kualitas suatu objek. Contoh modeel representasi integer ditunjukkan pada Gambar 3.3.

| | | | | | | | |
|---|---|----|---|---|----|---|---|
| 1 | 4 | 11 | 0 | 5 | 25 | 3 | 3 |
|---|---|----|---|---|----|---|---|

Gambar 3. 3 : Contoh model kromosom representasi integer

Model representasi *real*

Pada model representassi *real*, nilai-nilai yang tersimpan di dalam setiap gennya merupakan bilangan riil (*real*). Contoh model representasi *real* ditunjukkan pada Gambar 3.4.

| | | | | | | | |
|-----|-----|-----|------|------|------|------|-----|
| 1.2 | 4.9 | 1.9 | 0.25 | 0.67 | 2.33 | 3.33 | 2.5 |
|-----|-----|-----|------|------|------|------|-----|

Gambar 3. 4 : Contoh model representasi kromosom bilangan *real*

Model representasi permutasi

Pada model representas, nilai-nilai yang tersimpan dalam setiap gen merupakan bilangan bulat dimana setiap bilangannya merupakan angka unik (tidak berulang). Hal ini dikarenakan setiap gen merepresentasikan 1 solusi. Model ini biasanya digunakan untuk kasus *Travelling Salesman Problem* (TSP). Contoh model representasi permutasi ditunjukkan pada Gambar 3.5.

| | | | | | | | |
|---|---|---|----|----|---|---|----|
| 1 | 4 | 3 | 15 | 25 | 8 | 9 | 12 |
|---|---|---|----|----|---|---|----|

Gambar 3. 5 : Contoh model representasi kromosom permutasi

Suatu kromosom harus bisa merepresentasikan suatu solusi permasalahan dengan baik. Tsai dkk (2015) menyatakan bahwa representasi kromosom dalam bentuk linier tidak mampu untuk merepresentasikan beberapa kasus. Beberapa cara untuk mengatasi ini digunakan representasi kromosom yang tidak linear, antara lain:

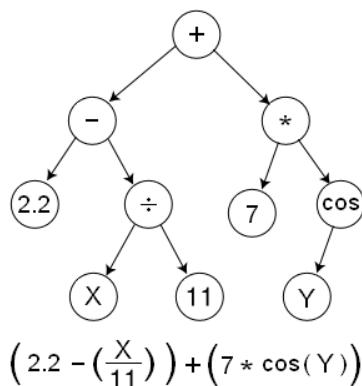
Representasi Kromosom 2-dimensi

Nilai gen untuk setiap representasi kromosom sama dengan representasi kromosom linier (dapat berupa bilangan biner, integer, *real*, atau permutasi). Format 2-dimensi dapat menggunakan *array* 2-dimensi, matriks, atau grid. Contoh model representasi 2-dimensi ditunjukkan pada Gambar 3.6.

| ARRAY | MATRIKS | GRID | | | | | | | | | | | | | | | | | | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---|---|---|---|---|
| <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>2</td><td>3</td><td>4</td></tr> <tr><td>4</td><td>5</td><td>1</td></tr> <tr><td>7</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>2</td><td>4</td></tr> </table> | 1 | 2 | 4 | 2 | 3 | 4 | 4 | 5 | 1 | 7 | 1 | 2 | 3 | 2 | 4 | $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ | <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> </table> | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 2 | 4 | | | | | | | | | | | | | | | | | | | | | |
| 2 | 3 | 4 | | | | | | | | | | | | | | | | | | | | | |
| 4 | 5 | 1 | | | | | | | | | | | | | | | | | | | | | |
| 7 | 1 | 2 | | | | | | | | | | | | | | | | | | | | | |
| 3 | 2 | 4 | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | | | | | | | |

Gambar 3. 6 : Contoh jenis-jenis representasi kromosom 2-dimensi
Representasi Tree.

Representasi pohon (*tree*) digunakan untuk representasi masalah yang batasan-batasan (*constraint*) nya belum diketahui. Model ini memiliki gen berbentuk simbol, nilai, atau karakter. Contoh model representasi *tree* ditunjukkan pada Gambar 3.7



Gambar 3. 7 : Contoh model representasi *tree*
Setiap nilai gen pada representasi kromosom merepresentasikan karakteristik dari solusi. Inisialisasi dilakukan dengan menghasilkan solusi individual secara

acak sesuai dengan aturan untuk menghasilkan populasi awal. Ukuran populasi dapat memuat ratusan hingga ribuan solusi yang memungkinkan, tergantung pada masalah yang ingin diselesaikan. Solusi dapat diarahkan ke area dimana solusi optimal dapat ditemukan.

3.3.2 Evaluasi nilai *fitness*

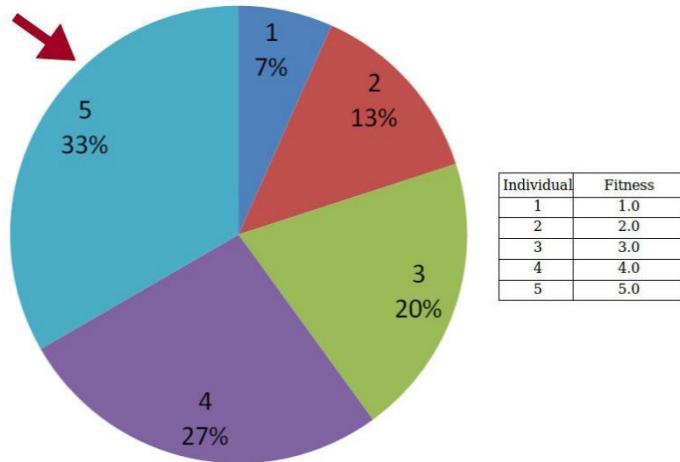
Pada tahap evaluasi individu, setiap kromosom pada populasi dihitung nilai kebugarannya (*fitness*) berdasarkan suatu fungsi *fitness*. Fungsi *fitness* adalah suatu alat ukur pada evaluasi kromosom yang menandakan kualitas suatu kromosom. Fungsi *fitness* dapat berupa fungsi matematika atau fungsi lainnya sesuai dengan permasalahan yang dihadapi. Kromosom dengan kualitas terbaik dapat berupa nilai maksimum atau minimum, tergantung persyaratan dari solusi optimal permasalahan. Proses evaluasi nilai *fitness* pada kromosom dalam suatu populasi akan terus dilakukan sampai kriteria terminasi proses terpenuhi.

3.4.3 Seleksi *parent*

Pada setiap generasi, dipilih sebagian kromosom dari populasi untuk menjadi *parent* untuk menghasilkan keturunan (*offspring*) sebagai calon solusi baru. Metode untuk pemilihan *parent* dapat dilakukan secara *random* dari populasi. Metode pemilihan tertentu menilai kesesuaian setiap solusi dan secara istimewa memilih solusi terbaik (Bajpai dan Kumar, 2010).

Kebanyakan metode dilakukan dengan nilai probabilitas kromosom tersebut akan menjadi *parent* dari melihat nilai *fitness* (*fitness proportionate selection*). Fungsi ini didesain agar sebagian kecil kromosom dari populasi dengan *fitness* yang kurang sesuai tetap terpilih untuk mempertahankan keberagaman yang tinggi dari populasi. Metode seleksi yang sering digunakan yaitu *roulette wheel selection*, Baker SUS's, dan seleksi turnamen.

Roulette Wheel Selection



Gambar 3. 8 : Ilustrasi roulette wheel selection

Tingkatan nilai *fitness* digunakan untuk menentukan probabilitas seleksi terhadap setiap solusi kromosom. Setiap iterasi akan membangkitkan nilai *sampling* secara *random* untuk pemilihan *parent* (diilustrasikan seperti satu pointer pada *roulette wheel*).

Untuk memperoleh nilai peluang *fitness* untuk setiap input, dan merepresentasikan kepada roda roulette (*roulette wheel*) secara matematis adalah sebagai berikut:

$$Fp(z) = \frac{f(z)}{\sum_{k=1}^n f(c(k))} \cdot 100 \quad (3.1)$$

Dimana:

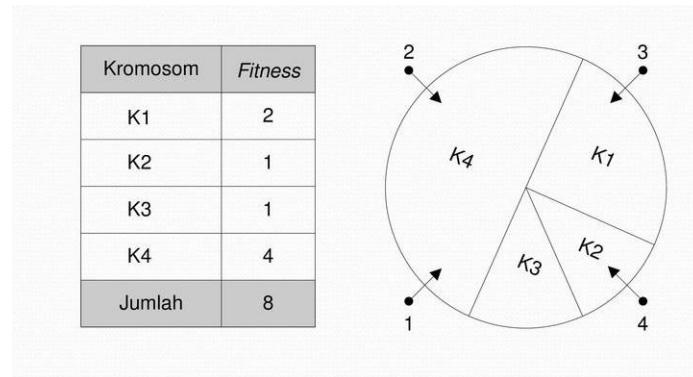
$Fp(z)$: *fitness probability* kromosom z .

$f(z)$: nilai *fitness* pada kromosom z

$\sum f(k)$: Jumlah total nilai *fitness* pada populasi.

Luas area kromosom pada *roulette wheel* merepresentasikan peluang *fitness* kromosom. Kromosom dengan nilai *fitness* yang lebih tinggi akan memiliki peluang yang tinggi untuk terpilih menjadi *parent*.

Baker's SUS (Stochastic Universal Sampling)



Gambar 3. 9 : Ilustrasi Baker SUS^s

Tingkatan nilai *fitness* digunakan untuk menentukan probabilitas seleksi terhadap setiap solusi kromosom. Berbeda dengan *roulette wheel*, pada Baker SUS^s terdapat sejumlah N-pointer hanya menggunakan nilai *sampling* pada iterasi pertama pada interval tertentu. Iterasi berikutnya akan ditambahkan dengan nilai sejumlah 1/N-pointer secara merata.

Luas area kromosom pada bidang merepresentasikan peluang *fitness* kromosom. Kromosom dengan nilai *fitness* yang lebih tinggi akan memiliki peluang yang tinggi untuk terpilih menjadi parent. Untuk memperoleh nilai peluang *fitness* adalah dengan menggunakan Persamaan 3.1.

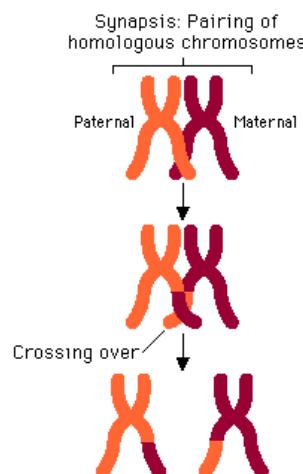
Tournament Selection

Tahapan dalam memperoleh calon *parent* menggunakan *tournament selection* adalah sebagai berikut:

1. Dua kromosom dipilih dari populasi.
2. Nilai *fitness* kedua kromosom dibandingkan. Kromosom dengan nilai *fitness* yang lebih tinggi akan terpilih untuk melakukan reproduksi.
3. Dapat dilakukan operasi secara paralel untuk mempercepat proses seleksi.

3.3.4 Crossover

Langkah selanjutnya setelah melakukan seleksi *parent* yaitu melakukan *crossover* atau perkawinan silang. *Crossover* memilih kromosom induk untuk melakukan reproduksi dan menghasilkan keturunan baru (*offspring*). Prinsip dasar pada operasi ini adalah melakukan pertukaran pada gen-gen yang bersesuaian dari dua induk untuk menghasilkan kromosom/individu baru. Konsep ini mengadaptasi proses pertukaran info genetik yang terjadi pada proses reproduksi yang diilustrakan dengan Gambar 3.10.



Gambar 3. 10 : Ilustrasi pertukaran informasi genetik pada reproduksi biologi.

Pada ilustrasi tersebut, kedua *parent* akan menghasilkan 2 kromosom *offspring*. Proses *crossover* pada algoritma genetika juga akan menghasilkan 2 kromosom *offspring*. Untuk setiap model representasi kromosom, memiliki metode *crossover* yang berbeda-beda untuk menghasilkan kromosom *offspring* untuk menyesuaikan batasan/karakteristik dari setiap kromosom.

Metode crossover untuk model representasi biner dan integer dapat dilakukan hal yang serupa, antara lain:

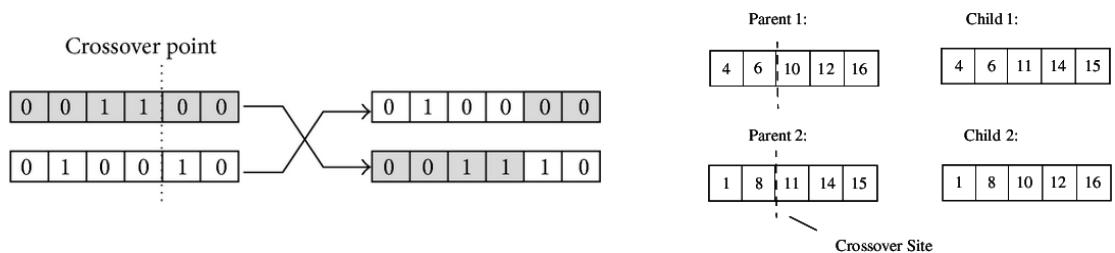
1-point crossover

Tahapan dalam proses *crossover* dengan metode 1-point *crossover* adalah sebagai berikut:

1. Memilih satu titik potong secara acak pada kromosom.

2. *Offspring* pertama memasukkan nilai gen pada bagian kiri titik potong *parent* pertama dan nilai gen pada bagian kanan titik potong *parent* kedua.
3. *Offspring* kedua memasukkan nilai gen pada bagian kiri titik potong *parent* kedua dan nilai gen pada bagian kanan titik potong *parent* pertama.

Contoh proses 1-point crossover ditunjukkan pada Gambar 3.11.



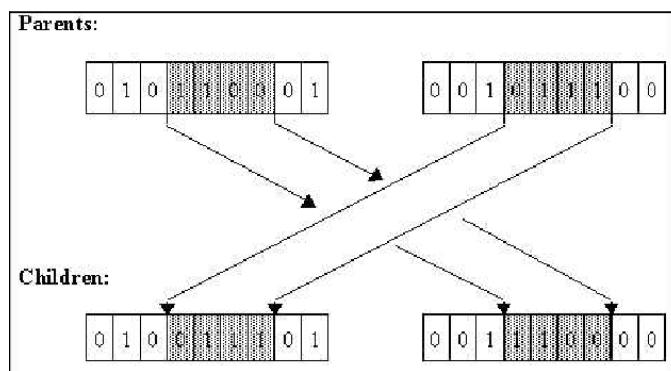
Gambar 3. 11 : Contoh proses 1-point crossover kromosom representasi biner (kiri) dan pada representasi integer (kanan)

N-point crossover

Tahapan proses *crossover* dengan metode N-point crossover adalah sebagai berikut:

1. Memilih sejumlah N titik potong secara acak pada kromosom.
2. *Offspring* pertama mendapatkan daerah ganjil pada *parent* pertama dan daerah genap pada *parent* kedua.
3. *Offspring* kedua mendapatkan daerah ganjil pada *parent* kedua dan daerah genap pada *parent* pertama.

Contoh proses N-point crossover ditunjukkan pada Gambar 3.12



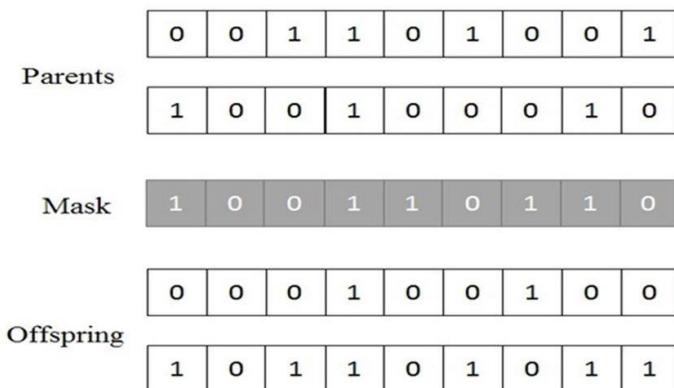
Gambar 3. 12 : Contoh proses N-point crossover

Uniform Crossover

Tahapan proses *crossover* menggunakan metode *uniform crossover* adalah sebagai berikut:

1. Membangkitkan mask nilai biner dengan bit sejumlah panjang kromosom.
2. *Offspring* pertama mendapatkan nilai gen *parent* pertama apabila pointer pada binary mask bernilai 0 dan mendapatkan nilai gen *parent* kedua apabila pointer pada binary mask bernilai 1.
3. *Offspring* kedua mendapatkan nilai gen *parent* pertama apabila pointer pada binary mask bernilai 1 dan mendapatkan nilai gen *parent* kedua apabila pointer pada binary mask bernilai 0.

Contoh *Uniform crossover* ditunjukkan pada Gambar 3.13



Gambar 3. 13 : Contoh proses *Uniform crossover*

Crossover dilakukan berdasarkan nilai ideal probabilitas *crossover* (Pc). Nilai Pc pada setiap permasalahan memiliki nilai ideal yang berbeda karena setiap masalah memiliki *space pencarian* yang berbeda-beda.

3.3.5 Mutasi

Mutasi pada algoritma genetika bekerja dengan mengubah nilai dari suatu individu. *Crossover* menciptakan *offspring* dengan nilai *fitness* yang lebih baik, lalu mutasi akan membantu meragamkan atau mengacak dengan mengeksplor lokasi pencarian. Proses mutasi pada algoritma genetika mencegah solusi terperangkap pada local optima. Mutasi dilakukan pada sebagian kromosom/individu pada

populasi. Kromosom/individu yang melakukan mutasi hanya melibatkan satu kromosom/individu, tidak seperti *crossover* yang melibatkan dua kromosom/individu.

Mutasi berkaitan dengan *mutation rate* (P_m) untuk menentukan banyaknya kromosom *offspring* yang akan melalui proses mutasi. Metode untuk menentukan gen yang bermutasi dapat dilakukan dengan beragam cara. Pada mutasi dapat diberlakukan aturan sedemikian rupa agar mutasi tetap menghasilkan kromosom yang layak (*feasible*) / sesuai aturan dan karakteristik dari kromosom. Beberapa metode mutasi diantaranya sebagai berikut:

Binary Mutation

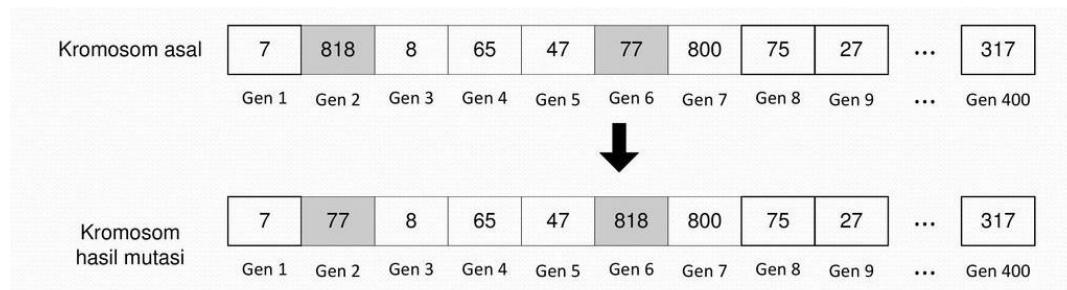
Metode ini digunakan khusus untuk representasi kromosom biner. Proses mutasi dilakukan dengan menukar nilai gen/bit pada posisi tertentu. Apabila nilai gen = 1, maka akan diganti menjadi 0. Sebaliknya, jika nilai gen = 0 akan ditukarkan dengan nilai 1. Contoh mutasi dengan *Binary Mutation* dapat dilihat pada Gambar 3.14.



Gambar 3. 14 Contoh proses *Binary mutation*

Swap Mutation

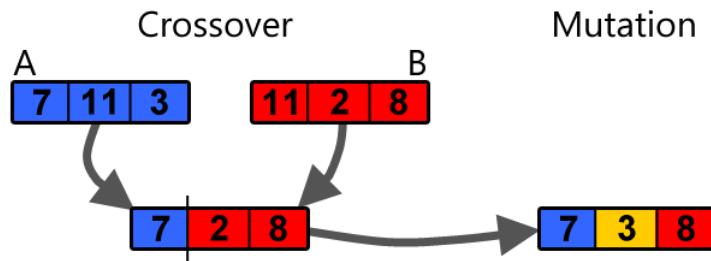
Pada metode ini, mutasi dilakukan dengan mengganti nilai gen pada suatu titik dengan nilai gen pada titik yang lain. Contoh mutasi dengan menggunakan *swap mutation* dapat dilihat pada Gambar 3.15.



Gambar 3. 15 : Contoh proses mutasi swap

Random Mutation

Pada metode ini, perubahan nilai gen dilakukan dengan mengganti nilai pada suatu titik dengan bilangan bulat acak. Contoh *random mutation* dapat dilihat pada Gambar 3.16.



Gambar 3. 16 : Contoh proses *random mutation*

3.3.6 Update generasi/seleksi *survivor*

Setelah terbentuk sejumlah kromosom/individu pada populasi, akan dipilih sebagian dari kromosom/individu tersebut untuk dibawa ke generasi selanjutnya. Beberapa metode yang dilakukan untuk proses seleksi *survivor* adalah sebagai berikut: (Smith dkk, 1998) dan (Chakraborty dan Chaudhuri, 2003)

1. *Generational Genetic Algorithm (GN GA)*

Metode ini menggantikan populasi pada generasi sebelumnya dengan *offspring* yang dihasilkan melalui proses *crossover* dan mutasi. Proses ini akan terus berlanjut hingga kondisi pemberhentian terpenuhi.

2. *Steady State Genetic Algorithm (SS GA)*

Kromosom *offspring* yang dihasilkan dari proses *crossover* dan mutasi ditambahkan ke populasi awal. Dari sejumlah M kromosom pada populasi baru tersebut, akan dipilih sejumlah N kromosom (ukuran populasi), dimana $N < M$ dengan N adalah sejumlah N kromosom dengan nilai *fitness* terbaik. Pada metode ini, kromosom *parent* dan kromosom *offspring* dapat berada pada populasi yang sama di generasi berikutnya, tanpa mempedulikan umur dari setiap kromosom pada setiap generasi.

3. Elitisme

Metode ini melakukan proses seleksi *survivor* dengan menggantikan kromosom *offspring* dengan nilai *fitness* terburuk (O_{worst}) dengan kromosom *parent* dengan nilai *fitness* terbaik (P_{best}). Tahap yang dilakukan adalah sebagai berikut:

- Jika $fitness O_{worst} < fitness P_{best}$: Ganti O_{worst} dengan P_{best}
- Jika $fitness O_{worst} \geq fitness P_{best}$: Seluruh *offspring* terpilih sebagai populasi baru pada generasi selanjutnya

3.3.7 Kondisi pemberhentian

Suatu proses algoritma genetika akan berhenti (*terminated*) dan menghasilkan solusi apabila suatu kondisi yang ingin dicapai sudah terpenuhi. Terdapat beberapa kriteria untuk menghentikan algoritma genetika, antara lain:

1. *Maximum Generation* : Jika telah melakukan evolusi sampai nilai maksimum generasi yang sudah ditentukan.
2. *Specific Time* : Algoritma genetika akan berjalan selama durasi waktu tertentu.
3. *Specific Fitness*: Terdapat suatu individu yang telah memiliki nilai *fitness* objektif yang telah ditentukan.
4. *Unchange fitness value* : Apabila nilai *fitness* tidak berubah setelah sejumlah generasi yang telah ditentukan.

3.4 Kromosom *Infeasible*

Penyelesaian masalah optimasi memiliki batasan-batasan pada proses untuk mendapatkan solusi optimal. Pada algoritma genetika, batasan ini dapat menyebabkan adanya representasi kromosom atau individu yang tidak layak (*infeasible*) setelah melalui proses algoritma genetika. Kromosom yang *infeasible* adalah kromosom yang telah melanggar batasan-batasan (*constraint*) yang sudah ditentukan (Abido dan Elazouni, 2009).

Iima Hitoshi (1995) meneliti efek kromosom atau individu yang tidak layak, namun tidak mengusulkan solusi untuk mengatasi masalah ini. Iima hanya mengeliminasi kromosom atau individu yang tidak *feasible* dari populasi.

Michalewicz (1995) menyarkan untuk tidak membunuh/eliminasi kromosom atau individu yang *infeasible* karena kromosom atau individu bisa memiliki nilai yang baik. Jika dilakukan perbaikan, peforma yang dilakukan pada algoritma genetika juga akan lebih baik daripada membunuh/eliminasi kromosom/individu. Diperlukan suatu mekanisme khusu untuk memperbaiki kromosom yang *infeasible* pada masing-masing kasus.

Sudah banyak penelitian yang mencoba berbagai metode untuk memperbaiki kromosom atau individu yang *infeasible*. Salah satu caranya adalah dengan melakukan mekanisme perbaikan diluar algoritma genetika. Bean(1992) mengusulkan untuk membangkitkan beberapa kunci *random* untuk menghindari hasil yang *infeasible*. Karena proses ini dilakukan diluar algoritma genetika, mekanisme ini menghasilkan turunnya peforma dari proses eksekusi. Solusi lainnya adalah dengan melakukan proses *crossover* dan mutasi sedemikian rupa sehingga kromosom/individu yang dihasilkan pasti selalu *feasible*. Abido dkk(2009) mengembangkan suatu operasi crossover dan mutasi untuk mencegah terjadinya kromosom yang *infeasible*. Ming dkk(2015) mengatasi dengan mendaftarkan 2 jenis nilai fitness, yaitu fitness objektif dan fitness penalti. Fitness objektif merupakan nilai yang ditargetkan untuk kromosom solusi, sedangkan fitness penalti merupakan nilai untuk pelanggaran batasan yang terjadi pada kromosom, kemudian akan dilakukan mekanisme perbaikan dengan mutasi pada masing-masing kromosom yang dengan nilai pinalti yang terpilih

BAB IV

ANALISIS DAN RANCANGAN

4.1 Analisis Permasalahan

Penempatan mahasiswa dengan dosen pembimbing tesis dengan kuota yang terbatas untuk setiap dosennya dan memastikan setiap mahasiswa dan dosen memiliki bidang riset penelitian yang sama merupakan salah satu contoh permasalahan alokasi. Diperlukan sebuah sistem yang bertujuan untuk menyelesaikan masalah alokasi mahasiswa dengan dosen melalui pengolahan dataset yang memuat informasi mahasiswa dan dosen. Tujuan dari sistem adalah untuk memperoleh skema kombinasi pasangan mahasiswa dan dosen yang terbaik dengan memanfaatkan dataset yang ada terhadap batasan-batasan (*constraint*) yang ditentukan.

Tahap awal yang dilakukan adalah memperoleh dataset yang memuat informasi kuota setiap dosen dan bidang riset penelitian dosen dan mahasiswa. Dataset yang masih berbentuk teks akan diproses dan disiapkan menjadi struktur data yang diperlukan pada proses algoritma genetika. Dataset yang memuat informasi dosen pembimbing tesis usulan mahasiswa ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Sampel data dosen pembimbing tesis usulan mahasiswa

| No. | Nama Mahasiswa | Dosen Usulan 1* | Dosen Usulan 2* | Dosen Usulan 3* |
|-----|----------------|------------------|----------------------|-----------------|
| 1 | WIDHI SULISTYO | Nur Rokhman | Mardhani Riasetiawan | Edi Winarko |
| 2 | Ekki Rinaldi | Aina Musdholifah | Wahyono | Edi Winarko |
| 3 | M. TAHIR | Moh. Edi Wibowo | Agus Harjoko | Wahyono |
| ... | ... | ... | ... | ... |

Dari Tabel 4.1, akan dilakukan evaluasi kuota dosen dengan menghitung total pengusul untuk setiap dosen dan dibandingkan dengan kuota yang tersedia. Dari evaluasi kuota tersebut akan terlihat bahwa terdapat dosen yang diusulkan

mahasiswa melebihi kuota tersedia. Evaluasi kuota dosen ditunjukkan pada Tabel 4.2.

Tabel 4. 2 Sampel evaluasi kuota dosen

| No | Nama Dosen | Jumlah Usulan 1 | Jumlah Usulan 2 | Jumlah Usulan 3 | Total Pengusul | Kuota Tersedia |
|----|--------------------|-----------------|-----------------|-----------------|----------------|----------------|
| 1 | Afiahayat | 0 | 2 | 2 | 4 | 1 |
| 2 | Agfianto Eko Putra | 0 | 0 | 0 | 0 | 2 |
| 3 | Agus Harjoko | 5 | 5 | 1 | 11 | 2 |
| 4 | Agus Sihabuddin | 0 | 1 | 0 | 1 | 6 |

4.2 Analisis Data

Data yang akan digunakan berfokus pada data mahasiswa dan dosen S2 Departemen Ilmu Komputer dan Elektronika (DIKE) Universitas Gadjah Mada. Dataset dosen memuat informasi kuota yang tersedia dan bidang riset penelitian setiap dosen. Dataset mahasiswa memuat informasi bidang minat penelitian yang akan diteliti. Dataset bidang minat/bidang riset memuat informasi tema riset yang tersedia. Diperlukan juga dataset hasil keputusan akhir pasangan dosen dan mahasiswa dari hasil rapat diskusi dosen pada periode yang sesuai untuk keperluan evaluasi hasil dari program algoritma genetika. Sampel dataset dosen, mahasiswa , dang bidang riset ditunjukkan pada Tabel 4.3, Tabel 4.4 dan Tabel 4.5.

Tabel 4. 3 Sampel dataset dosen

| No. | Nama Dosen | Kuota Tersedia | Bidang Minat/Bidang Riset |
|-----|--------------------|----------------|---------------------------------|
| 1 | Afiahayati | 1 | Bioinformatics |
| | | | Machine Learning |
| 2 | Agfianto Eko Putra | 2 | Embedded systems |
| | | | DSP |
| 3 | Agus Harjoko | 2 | Computer Vision |
| | | | Pattern Recognition |
| | | | Instrumentation, Sensor Network |
| 4 | Agus Sihabuddin | 6 | Information System |
| | | | Distributed Data Base System |
| | | | Software Engineering |

| | | | |
|-----|--------------|-----|--------------------------------------|
| 5 | Ahmad Ashari | 3 | Distributed System Grid Computing |
| ... | ... | ... | ... |

Tabel 4. 4 Sampel dataset mahasiswa

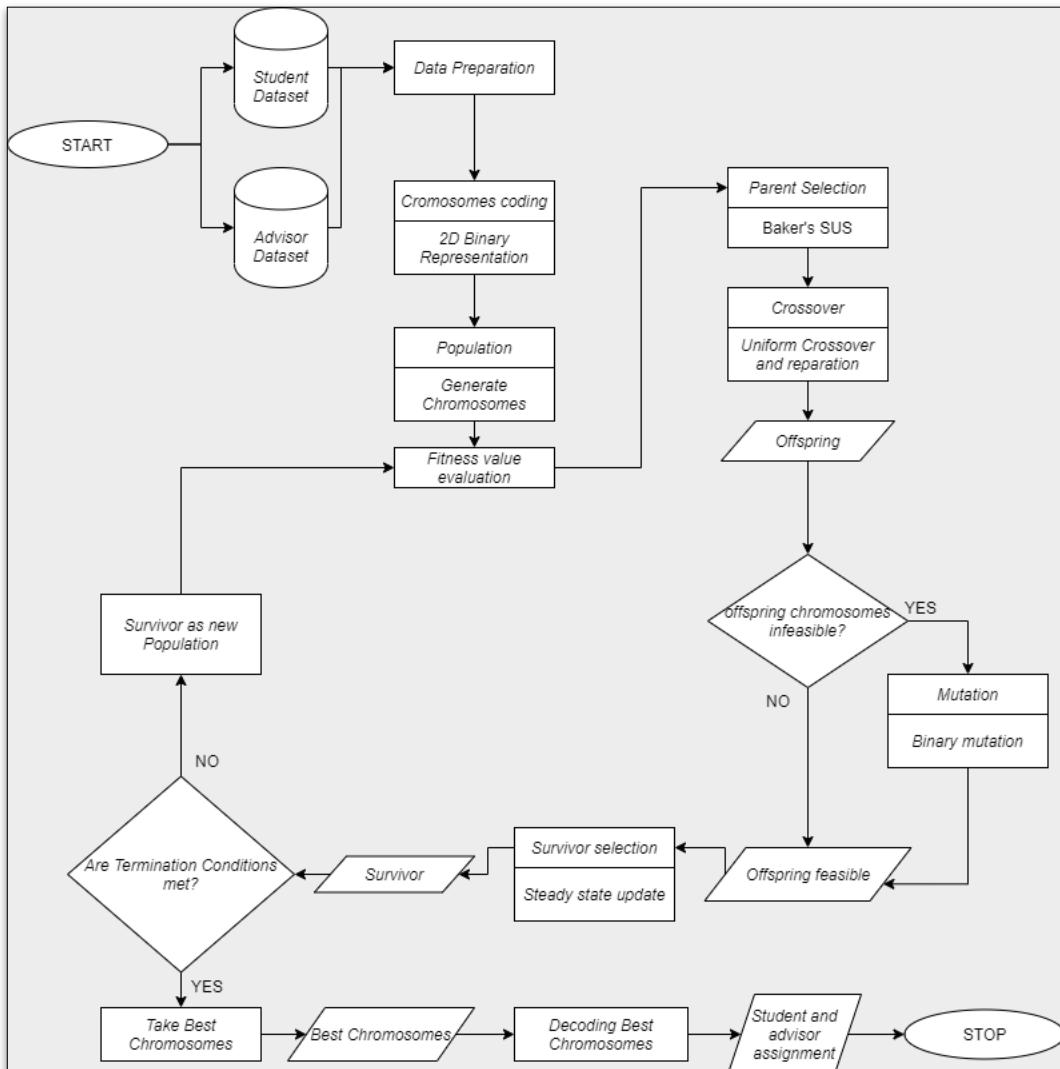
| No | Nama Mahasiswa | Topik Tesis |
|-----|----------------------|-------------------------------------|
| 1 | WIDHI SULISTYO | Big Data Analysis and Visualization |
| 2 | Ekki Rinaldi | Big Data Analysis and Visualization |
| 3 | M. TAHIR | Computer Vision |
| 4 | Ririn Septrisulviani | Algoritma |
| 5 | Ma'ruf Aminudin | Kecerdasan Komputasional |
| ... | ... | ... |

Tabel 4. 5 Sampel bidang riset tersedia

| No. | Bidang Riset |
|-----|-----------------------------------|
| 1 | Lab AK - Algoritma |
| 2 | Lab AK - Approximation Algorithm |
| 3 | Lab AK - GPU & Parallel Computing |
| 4 | Lab AK - Grafika Komputer |
| 5 | Lab AK - Kompresi Data |
| ... | ... |

4.3 Rancangan Algoritma Genetika

Untuk menyelesaikan permasalahan alokasi mahasiswa dan dosen pada tugas tesis dilakukan perancangan algoritma genetika. Proses algoritma genetika yang akan dilakukan ditunjukkan pada Gambar 4.1.



Gambar 4. 1 Alur rancangan algoritma genetika

4.3.1 Penyiapan data

Dataset dosen, dataset mahasiswa, dan dataset bidang riset perlu diubah atau disiapkan sesuai dengan kebutuhan proses algoritma genetika yang digunakan.

Misal Y adalah jumlah dosen, X jumlah mahasiswa, dan R banyaknya bidang riset penelitian. Dari dataset tersebut akan dibentuk 3 array linear (array dosen, array mahasiswa, dan array bidang riset penelitian dengan panjang masing-masing X, Y , dan R) dimana setiap nomor index merupakan id_dosen, id_mahasiswa, dan

id_bidangRiset, dan setiap elemen-nya merupakan nama dosen, nama mahasiswa, dan nama bidang. Contoh array ditunjukkan pada Gambar 4.2

| | | | | | |
|---------------------------|------------|----------|--------------|-----------------|-----|
| Index = Id_dosen | 0 | 1 | 2 | 3 | ... |
| Array[index] = nama Dosen | Afiahayati | Agfianto | Agus Harjoko | Agus Sihabuddin | ... |

Gambar 4. 2a Contoh array dosen

| | | | | | | |
|-------------------------------|----------------|--------------|----------|----------------------|------------------|-----|
| Index = Id_mahasiswa | 0 | 1 | 2 | 3 | 4 | ... |
| Array[index] = nama mahasiswa | WIDHI SULISTYO | Ekki Rinaldi | M. Tahir | Ririn Septrisulviani | Ma'aruf Aminudin | ... |

Gambar 4. 2b Contoh array mahasiswa

| | | | | | |
|----------------------------|------------------|----------|--------------|------------------|-----|
| Index = Id_bidang | 0 | 1 | 2 | 3 | ... |
| Array[index] = nama bidang | Machine Learning | Robotics | Systems soft | Embedded systems | ... |

Gambar 4. 2c Contoh array bidang riset

Kuota tersedia setiap dosen akan diubah kedalam array linear dengan nomor index sesuai dengan id_dosen dan setiap elemennya berisi nilai kuota yang bersesuaian. Contoh hasil penyiapan kuota dosen kedalam bentuk array ditunjukkan pada Gambar 4.3

| | | | | | | |
|-------------------------------|---|---|---|---|---|-----|
| Index = Id_dosen | 0 | 1 | 2 | 3 | 4 | ... |
| Array[index] = kuota tersedia | 1 | 2 | 2 | 6 | 3 | ... |

Gambar 4. 3 Contoh array kuota tersedia setiap dosen

Dari Tabel 4.3 akan dibentuk matriks bidang dosen (Md) , yaitu matriks berukuran $R*Y$ yang memuat informasi bidang riset penelitian dosen. Elemen dari matriks tersebut merupakan bilangan biner (1 atau 0). Contoh Md ditunjukkan pada Gambar 4.4.

| index bidang = id_bidang | index_dosen = id dosen | | | |
|--------------------------|------------------------|---|---|---|
| | 0 | 1 | 2 | 3 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 | 0 |
| 5 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 | 1 |

Gambar 4. 4 Contoh matriks bidang dosen

Dari contoh Md pada Gambar 4.4, berisikan informasi sebagai berikut:

$$Md_{ij} = \begin{cases} 1, & y_j \text{ minat dengan } r_i \\ 0, & y_j \text{ tidak minat dengan } r_i \end{cases} \quad (4.1)$$

Dimana:

- y_j : dosen dengan id_dosen = j
- r_i : bidang riset dengan id_bidang = i

Dari Tabel 4.4 akan dibentuk matriks bidang mahasiswa (Mm) , yaitu matriks berukuran R*X yang memuat informasi bidang riset pilihan mahasiswa. Elemen dari matriks tersebut merupakan bilangan biner (1 atau 0). Contoh Mm ditunjukkan pada Gambar 4.5.

| index bidang = id_bidang | index_mahasiswa = id mahasiswa | | | | | | | | |
|--------------------------|--------------------------------|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

Gambar 4. 5 Contoh matriks bidang riset pilihan mahasiswa

Dari contoh Mm, berisikan informasi sebagai berikut:

$$Mm_{ij} = \begin{cases} 1, & x_j \text{ minat dengan } r_i \\ 0, & x_j \text{ tidak minat dengan } r_i \end{cases} \quad (4.2)$$

Dimana;

- x_j : Mahasiswa dengan id_mahasiswa = j
- r_i : Bidang riset dengan id_bidang = i

4.3.2 Perancangan parameter operasi algoritma genetika

Tahap awal dari proses perancangan algoritma genetika untuk alokasi dosen pembimbing tesis adalah menentukan parameter yang akan digunakan untuk menyelesaikan permasalahan tersebut. Parameter-parameter operasi algoritma genetika yang digunakan adalah sebagai berikut:

- 1) *Fitness target*: *fitness value* yang menjadi standar *goal* minimum *fitness value* dari solusi/kromosom yang ingin dicapai.
- 2) Generasi maksimum: Banyaknya jumlah generasi maksimum dari proses algoritma genetika. Apabila setelah mencapai generasi maksimum *fitness target* masih belum tercapai, maka proses algoritma akan berhenti dan mengambil kromosom terbaik dari populasi pada generasi terakhir.
- 3) Ukuran populasi : Banyaknya kromosom dalam satu populasi, dimana untuk setiap generasi memiliki ukuran populasi yang sama.
- 4) *Probabilitas crossover* (Pc) : Peluang pasangan *parent* untuk melakukan proses *crossover*. Cara yang umum dilakukan adalah dengan membangkitkan suatu nilai *random* untuk setiap pasangan *parent*. Apabila nilai *random* tersebut dibawah Pc maka tidak terjadi *crossover*.
- 5) *Offspring unvalid value* : Suatu nilai yang mampu menginformasikan apakah kromosom *offspring* yang dihasilkan dari proses *crossover* sudah *feasible* (Tidak melanggar *constraint*). Kromosom *offspring* akan memiliki *unvalid value* > 0 apabila kromosom tersebut *infeasible*. Kromosom yang *infeasible* akan melalui proses mutasi, sedangkan kromosom yang *feasible* tidak akan melalui proses mutase.

4.3.3 Representasi (pengkodean) kromosom

Suatu representasi kromosom pada algoritma genetika harus mampu merepresentasikan dan menjelaskan kemungkinan solusi-solusi pada permasalahan yang akan diselesaikan.

Untuk permasalahan alokasi dosen dan mahasiswa untuk tesis, diujikan dengan representasi kromosom 2D dengan menggunakan matriks berukuran $Y*X$ (jumlah dosen * jumlah mahasiswa).

Notasi, aturan, dan constrain

Notasi-notasi yang akan digunakan dalam proses pengkodean kromosom adalah sebagai berikut:

- P : Populasi yang terdiri dari kumpulan krosomom;
- n : Jumlah kromosom pada P ;
- C_k : Kromosom ke- k pada P , dimana $1 \leq k < n$;
- x_i : id_mahasiswa pada kolom ke- i
- y_j : id_dosen pada baris ke- j
- X : Jumlah kolom dimana $Jumlah_kolom = jumlah_mahasiswa$
- Y : Jumlah baris dimana $Jumlah_baris = jumlah_dosen$
- Mx : Jumlah maksimum usulan dosen untuk setiap mahasiswa (3 usulan)
- mx : Jumlah minimum usulan dosen untuk setiap mahasiswa (1 usulan)
- My : Kuota tersedia setiap dosen
- C_k^{ij} : Gen dengan posisi (i,j) pada kromosom ke- k

Kromosom akan direpresentasikan dengan representasi biner, dimana:

$$C_k^{ij} = \begin{cases} 1, & x_i \text{ mengusulkan } y_j \\ 0, & x_i \text{ tidak mengusulkan } y_j \end{cases} \quad (4.3)$$

Contoh representasi kromosom ditunjukkan pada Gambar 4.6 dan analisis dari kromosom tersebut ditunjukkan pada Tabel 4.6

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Gambar 4. 6 Contoh kromosom

Tabel 4. 6 Analisis kromosom

| id_dosen | index_mahasiswa = id mahasiswa | | | | | | | | | Total Pengusul | Sisa Kuota |
|--------------|--------------------------------|---|---|---|---|---|---|---|---|----------------|------------|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 4 | 5-4 = 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 6 | 9-6 = 3 |
| 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 4-4 = 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 2-2 = 0 |
| Total Usulan | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 2 | 3 | | |

Dari Tabel 4.4. terlihat bahwa tidak ada dosen yang diusulkan melebihi dari kuota yang tersedia, dan tidak ada mahasiswa yang mengusulkan lebih dari jumlah maksimal usulan (M_x) dan setiap mahasiswa dipastikan mendapatkan satu dosen (m_x), dimana pada batasan masalah ini mahasiswa dapat mengusulkan 1 sampai 3 dosen.

Proses pembangkitan kromosom

Proses pengkodean dan pembangkitan kromosom akan dilakukan dengan 2 tahap sebagai berikut:

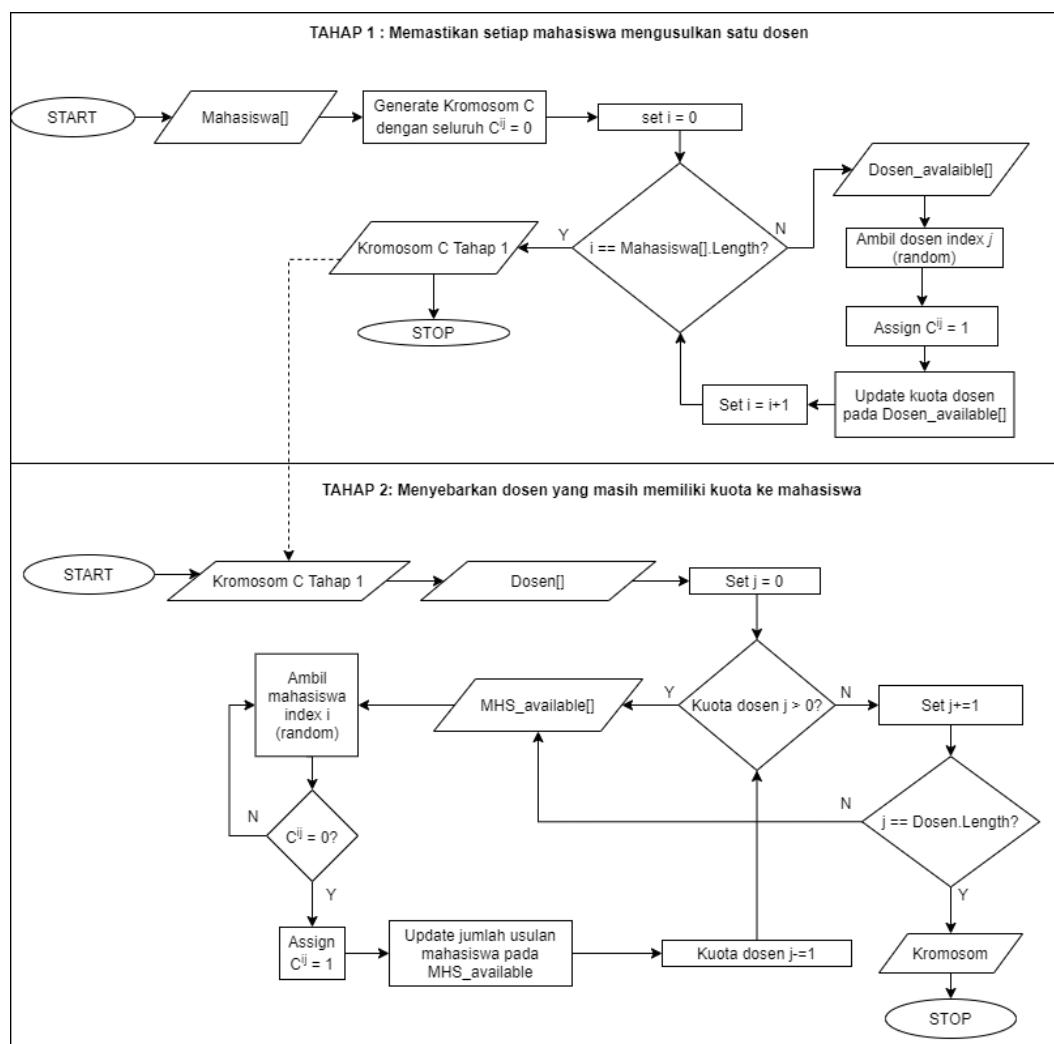
Tahap 1 : Memastikan setiap mahasiswa mengusulkan 1 dosen

1. Deklarasikan kromosom C dengan matriks berukuran $Y*X$, dimana seluruh elemennya memiliki value = 0
2. Deklarasikan suatu array linear (misal `dosen_available`) yang elemennya merupakan `id_dosen` yang kuotanya masih tersedia
3. Pada setiap x_i , generate bilangan random R dengan range((0, `dosen_available`-1)).
4. Simpan value $j = \text{dosen_available}[R]$
5. Set value $C^{ij} = 1$.
6. Set $M_{yj} := 1$. Jika $M_{yj} <= 0$, maka hapus yj dari `dosen available`.

Tahap 2: Menyebarluaskan dosen yang masih memiliki kuota

1. Deklarasikan suatu array linear (misal `mhs_available`) yang elemennya merupakan `id_mahasiswa` yang masih memiliki jumlah dosen usulan $\leq M_x$. Untuk proses deklarasi ini seluruh elemen berisikan seluruh `id_mahasiswa` (karena seluruh mahasiswa masih memiliki 1 usulan dosen).
2. Untuk setiap y_j , Jika y_j merupakan anggota `dosen_available`, maka:
 - 1) Generate bilangan random R dengan range(0,(`mhs_available`-1))
 - 2) Simpan value $i = \text{mhs_available}[R]$
 - 3) Jika $C^{ij} = 0$:
 - a) Set value $C^{ij} = 1$

- b) Set $Mx_i = 1$ dan set $My_j = 1$
c) Jika $Mx_i \leq 0$, hapus x_i dari mhs_available.
4) Ulangi kembali tahap sampai $My_i \leq 0$
- Diagram alur dari proses pembangkitan kromosom, ditunjukkan pada Gambar 4.7.



Gambar 4. 7 Diagram alur proses pengkodean dan pembangkitan kromosom (C_k)

4.3.4 Evaluasi nilai *fitness*

Nilai *fitness* merepresentasikan kualitas suatu kromosom. Pada permasalahan ini, nilai *fitness* suatu kromosom ditentukan berdasarkan persentase mahasiswa

yang mengusulkan setidaknya 1 (satu) dosen dengan bidang riset penelitian yang cocok. Secara matematis dibuat kedalam fungsi *fitness* sebagai berikut:

$$f(fit) = \frac{fit}{X} * 100 \quad (4.4)$$

Dengan *fit* = Jumlah mahasiswa dengan bidang riset penelitian yang cocok dengan dosen yang diusulkan dan *X* adalah jumlah total mahasiswa.

Proses perolehan nilai *fitness* pada kromosom adalah sebagai berikut:

- 1) Deklarasikan *fit* = 0
- 2) Untuk setiap xi dan yj , jika $C^{ij} = 1$
 - o Lakukan iterasi sebanyak jumlah *r* (panjang array bidang):
Jika $Mm_{ki} = Md_{ki} = 1$, set *fit*=1, lalu keluar dari iterasi.
- 3) Set *fitness value* = $f(fit) = \frac{fit}{X} * 100$

Sebagai contoh, akan dihitung nilai *fitness* dari kromosom pada Gambar 4.6 dengan menggunakan matriks bidang pada Gambar 4.4 dan Gambar 4.5. Contoh proses evaluasi nilai *fitness* ditunjukkan pada Tabel 4.7.

Tabel 4. 7 Contoh proses evaluasi nilai *fitness*

| Id_Mahasiswa | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---------------------|-------------|-------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| Usulan1 | C^{01} | C^{10} | C^{22} | C^{30} | C^{41} | C^{51} | C^{61} | C^{71} | C^{80} |
| | $Mm_{00}=1$ | $Mm_{11}=1$ | $Mm_{22}=1$ | $Mm_{73}=1$ | $Mm_{34}=1$ | $Mm_{45}=1$ | $Mm_{46}=1$ | $Mm_{77}=1$ | $Mm_{08}=1$ |
| | $Md_{01}=1$ | $Md_{10}=1$ | $Md_{22}=0$ | $Md_{70}=0$ | $Md_{31}=0$ | $Md_{41}=0$ | $Md_{41}=1$ | $Md_{71}=1$ | $Md_{00}=1$ |
| Usulan 2 | Break | Break | | C_{31} | | | Break | Break | Break |
| | | | | $Mm_{73}=1$ | | | | | |
| | | | | $Md_{71} = 0$ | | | | | |
| Usulan 3 | | | | | | | | | Break |
| | | | | | | | | | |
| | | | | | | | | | |
| <i>fit</i> | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Dari Tabel 4.7, diperoleh $\sum fit = 5$. Maka nilai *fitness* yang dari kromosom pada Gambar 4.6 adalah:

$$f(fit) = \frac{fit}{X} * 100 = \frac{5}{9} * 100 = 55.55 (\%)$$

Dari contoh evaluasi tersebut, dapat disimpulkan bahwa sekitar 56% dari mahasiswa mengusulkan setidaknya satu dosen dengan bidang riset yang sesuai dengan bidang riset penelitian tesis.

4.3.5 Seleksi *parent*

Seleksi *parent* dilakukan untuk memilih kromosom-kromosom tertentu untuk dijadikan calon orangtua untuk menghasilkan kromosom-kromosom *offspring*. Metode Baker's SUS merupakan salah satu metode yang digunakan untuk melakukan seleksi *parent*. Metode ini melakukan proses seleksi dengan memperhatikan nilai *fitness* suatu kromosom.

Setelah dilakukan proses evaluasi nilai *fitness* pada seluruh kromosom di populasi dapat diketahui nilai *fitness* atau kualitas suatu kromosom untuk melewati proses seleksi *parent*. Contoh dari proses seleksi *parent* dengan Baker's SUS diperlihatkan pada Tabel 4.8.

Tabel 4. 8 Contoh seleksi *parent* dengan Baker's SUS

| Kromosom | $fCi(z)$ | Prob(%) | CumProb(%) |
|----------|----------|-------------|-------------|
| C_2 | 3 | 13.04347826 | 13.04347826 |
| C_3 | 3 | 13.04347826 | 26.08695652 |
| C_1 | 5 | 21.73913043 | 47.82608696 |
| C_5 | 5 | 21.73913043 | 69.56521739 |
| C_4 | 7 | 30.43478261 | 100 |
| Total | 23 | 100 | |

Misal, Mating pool = 3, maka N-pointer = 3.

- Iterasi-1:
Pointer-1 = Random[0,1/N] = Random[0,0.33] = 0.2 = 20% → C_3
- Iterasi-2:
Pointer-2 = Pointer-1 + 1/N = 0.2 + 0.33 = 0.53 = 53% → C_5
- Iterasi-3

$$\text{Pointer-3} = \text{Pointer-2} + 1/N = 0.53 + 0.33 = 0.86 = 86\% \rightarrow C_4$$

Maka, kromosom terpilih untuk menjadi parent adalah C_3 , C_4 , dan C_5 .

4.3.6 Crossover

Metode *crossover* (kawin silang) yang diujikan adalah *uniform crossover*, dimana akan dibangkitkan suatu pola acak yang bersesuaian dengan kromosom *parent*. Karena pada permasalahan ini menggunakan representasi kromosom 2D, pola yang dibangkitkan juga akan berbentuk 2D. Setiap pasangan *parent* akan menghasilkan 2 kromosom *offspring*. Pola yang dibangkitkan akan berisi nilai 1 atau 0 dimana:

- *Offspring 1:*

$$C_{\text{offspring1}}^{ij} = \begin{cases} C_{\text{parent1}}^{ij}, & \text{Pola}^{ij} = 0 \\ C_{\text{parent2}}^{ij}, & \text{Pola}^{ij} = 1 \end{cases} \quad (4.5)$$

- *Offspring 2*

$$C_{\text{offspring2}}^{ij} = \begin{cases} C_{\text{parent2}}^{ij}, & \text{Pola}^{ij} = 0 \\ C_{\text{parent1}}^{ij}, & \text{Pola}^{ij} = 1 \end{cases} \quad (4.6)$$

Contoh pola *crossover* yang dibangkitkan ditunjukkan pada Gambar 4.8.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

Gambar 4. 8 Contoh pola *uniform crossover*

Crossover dilakukan berdasarkan nilai probabilitas *crossover* (Pc). Nilai Pc pada setiap permasalahan yang akan diselesaikan dengan algoritma genetika berbeda-beda menyesuaikan setiap permasalahan. Pada penerapannya, untuk setiap pasangan *parent* yang akan melakukan *crossover* mula-mula dibangkitkan sebuah nilai *random* berkisar antara 0.0 hingga 1.0. Apabila nilai dari suatu bilangan

random tersebut lebih kecil atau sama dengan nilai P_c , maka pasangan kromosom *parent* tersebut akan melakukan proses *crossover*. Sebaliknya, apabila nilai bilangan *random* tersebut lebih besar daripada nilai P_c , maka pasangan *parent* tersebut tidak melakukan proses *crossover*.

Pada representasi kromosom yang digunakan untuk permasalahan alokasi pemilihan dosen pembimbing tesis, proses *crossover* dengan metode *uniform crossover* dapat menghasilkan kromosom-kromosom yang *infeasible* (tidak layak). Kromosom dikatakan *infeasible* apabila melanggar batasan-batasan / *constraint* dari permasalahan. Setelah dilakukan proses *crossover*, terdapat peluang kromosom *offspring* yang dihasilkan melanggar aturan M_x (ada mahasiswa yang mengusulkan lebih dari 3 dosen) dan M_y (ada dosen yang diusulkan lebih dari kuota tersedia). Proses perbaikan batasan M_x dilakukan setelah *crossover* dan semua kromosom *offspring* harus melewati proses pengecekan untuk perbaikan. Proses perbaikan batasan M_y dilakukan pada proses mutasi dan belum tentu semua kromosom *offspring* melalui proses perbaikan.

Proses perbaikan batasan M_x adalah sebagai berikut:

1. Untuk setiap x_i , hitung $SUM_usulan = \sum_j^Y C^{ij}$
2. Declare sebuah array kosong (misal *container_value1*)
3. Apabila SUM_usulan pada $x_i \geq M_x$, cek lokasi $C^{ij} = 1$, lalu simpan value j kedalam *container_value1*
4. Hitung $over = SUM_usulan - M_x$
5. Sebanyak $over$ iterasi, lakukan:
 - 1) Generate bilangan *random* R dengan range(0,(panjang *container_value1*-1))
 - 2) Set $k = container_value1[R]$
 - 3) Set $C_{offspring}^{ik} = 0$
 - 4) Hapus element index ke- R dari *container_value1*

Setelah melalui proses perbaikan, dapat dipastikan bahwa tidak ada mahasiswa yang mengusulkan lebih dari M_x (3 dosen).

4.3.7 Mutasi

Mutasi adalah proses mengubah beberapa nilai gen pada sebuah kromosom dengan nilai tertentu yang sesuai dengan representasi kromosom tersebut. Tujuan utama dari proses mutase adalah meningkatkan keragaman (*diversity*) dari kromosom-kromosom pada populasi untuk memperluas area pencarian. Pada permasalahan ini, mutasi juga bertujuan untuk memperbaiki kromosom yang *infeasible* sehingga menjadi kromosom yang *feasible* sesuai dengan batasan-batasan kromosom yang sudah ditentukan.

Berbeda dari proses mutasi pada umumnya yang bergantung pada nilai probabilitas mutase (P_m), pengujian pada permasalahan ini berganti pada *penalty value* untuk suatu kromosom *offspring* melakukan proses mutasi. Fokus batasan yang akan diperbaiki adalah pada batasan M_y (adanya dosen yang diusulkan lebih dari kuota yang tersedia). Apabila ditemukan setidaknya 1 (satu) dosen yang melanggar aturan, maka *penalty value* kromosom bernilai 1. Apabila sama sekali tidak ada dosen yang melanggar maka *penalty value* bernilai 0. Kromosom-kromosom *offspring* yang memiliki *penalty value* = 1 akan melalui proses mutasi.

Proses mutasi yang dilakukan adalah sebagai berikut:

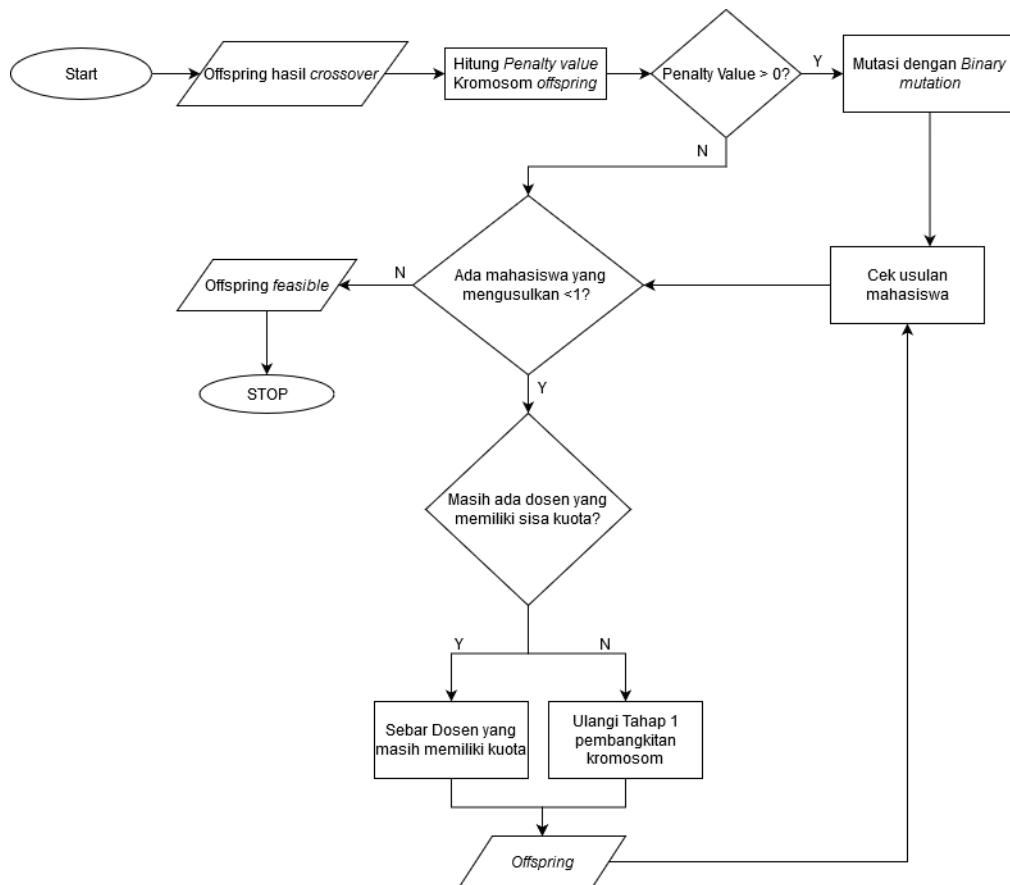
1. Untuk setiap yj , hitung $SUM_pengusul = \sum_i^X C^{ij}$ dan simpan nilai j pada suatu array (misal idx) dimana $\sum_i^X C^{ij} \geq M_{yj}$
2. Deklarasikan sebuah array kosong (misal gen_val1) dan isi dengan nilai i dimana $C^{ij} = 1$.
3. *Generate* nilai random R dari gen_val1
4. Set $C^{Rj} = 0$
5. Set sisa kuota $M_{yj} += 1$
6. Jika sisa kuota $M_{yj} \geq 0$ (dosen tersebut tidak diusulkan lebih dari kuota tersedia), hapus elemen bernilai j dari array idx
7. Selama array idx tidak null, lakukan kembali tahap 3 sampai 7
8. Jika array idx sudah null, perlu dilakukan pengecekan apakah setiap mahasiswa sudah mengusulkan minimal 1 dosen.

- a) Jika terdapat dosen yang masih memiliki sisa kuota, maka proses perbaikan sama dengan Tahap 1 proses *generate* kromosom
- b) Jika seluruh dosen sudah tidak memiliki sisa kuota, maka perbaikan dilakukan dengan menukar (*swap*)/mengoper dosen dari mahasiswa yang mengusulkan lebih dari 1 dosen.
- Secara matematis dituliskan sebagai berikut:

$$C^{ij} = C^{kj}, \quad \sum_j^Y C^{kj} > mx \text{ dan } C^{kj} = 1 \quad (4.7)$$

$$C^{kj} = C^{ij}, \quad \sum_j^Y C^{ij} < mx \text{ dan } C^{ij} = 0 \quad (4.8)$$

Diagram alur proses mutasi ditunjukkan pada Gambar 4. 9.



Gambar 4. 9 Diagram alur proses mutasi

4.3.8 Seleksi *survivor*

Seleksi *survivor* dilakukan untuk memilih sejumlah n kromosom yang akan bertahan pada generasi selanjutnya. Metode seleksi *survivor* yang digunakan pada rancangan algoritma genetika untuk alokasi dosen pembimbing tesis adalah *steady state genetic algorithm* (SS GA). SS GA memilih 2 (dua) kromosom sebagai *parent* berdasarkan metode seleksi tertentu (dimana pada permasalahan alokasi dosen pembimbing tesis menggunakan Baker's SUS), dan *offspring* hasil *crossover*/mutasi ditambahkan ke populasi dan menggantikan kromosom yang memiliki nilai *fitness* yang lebih buruk. Dimisalkan terdapat sejumlah n kromosom pada populasi awal dan dihasilkan sejumlah m *offspring* pada proses *crossover* dan mutasi. Dari $n+m$ kromosom pada populasi, akan ditingkatkan berdasarkan nilai *fitness* dan hanya sejumlah n kromosom yang akan dibawa ke generasi selanjutnya. Metode ini dapat mengakibatkan kromosom *parent* dan *offspring* berada pada populasi yang sama untuk generasi selanjutnya.

4.3.9 Kondisi pemberhentian

Proses yang sudah dilakukan dengan algoritma genetika akan terus berulang sampai tercapainya kriteria untuk berhenti. Terdapat dua kondisi algoritma genetika berhenti, yaitu:

1. Tercapainya *fitness* target
 - Apabila ditemukan satu kromosom dalam suatu populasi pada generasi tertentu dengan nilai *fitness* terbaik sudah mencapai nilai *fitness* target, maka proses algoritma genetika akan berhenti dan menyimpan kromosom terbaik tersebut sebagai solusi/hasil akhir.
 - Nilai *fitness* target yang ingin dicapai adalah 80%, dalam kata lain apabila suatu kromosom memiliki nilai $fitness \geq 80\%$, maka minimal 80% dari jumlah mahasiswa mengusulkan dosen dengan bidang riset penelitian yang cocok dengan bidang riset tesisnya.
2. Mencapai generasi maksimum

- Apabila nilai *fitness* kromosom terbaik dalam populasi pada setiap generasi masih tidak ada yang mencapai *fitness* target, maka proses algoritma genetika akan langsung berhenti apabila generasi sudah mencapai generasi maksimum. Kromosom dengan nilai *fitness* terbaik pada generasi terakhir akan disimpan sebagai solusi.

4.4 Perancangan Pengujian Sistem

Setelah sistem diimplementasikan, dilakukan fase pengujian terhadap peforma sistem algoritma genetika. Pengujian ini dilakukan untuk mencari parameter algoritma genetika yang tepat dalam menyelesaikan permasalahan alokasi dosen dengan mahasiswa untuk mendapatkan kromosom/solusi yang optimal. Perubahan nilai parameter dilakukan pada ukuran populasi, probabilitas *crossover* (Pc) dan generasi maksimum. Nilai-nilai parameter yang akan diujikan ditunjukkan pada Tabel 4.9.

Tabel 4.9 Parameter pengujian

| No | Parameter | Nilai parameter yang diujikan |
|----|-------------------------------|-------------------------------|
| 1 | Probabilitas <i>crossover</i> | 0,6; 0,7; 0,8; 0,9. |
| 2 | Ukuran populasi | 10, 25, 50. |
| 3 | Generasi maksimum | 100, 1000, 1500 |

4.5 Evaluasi Kinerja Sistem

Dari hasil yang diperoleh, diperlukan evaluasi apakah sistem algoritma genetika yang sudah diimplementasikan mampu menyelesaikan permasalahan alokasi mahasiswa dengan dosen pembimbing tesis. Kromosom solusi akan dibandingkan dengan dataset hasil keputusan akhir mahasiswa dan dosen pembimbing tesis. Dataset hasil keputusan akan disiapkan kedalam bentuk kromosom untuk dilihat nilai *fitness* dari data tersebut. Nilai *fitness* dataset dan nilai *fitness* kromosom solusi akan dibandingkan, dan akan terlihat apakah penyelesaian dengan algoritma genetika lebih baik dibandingkan dengan data hasil keputusan rapat dosen. Evaluasi lain adalah memastikan bahwa kromosom solusi adalah solusi

yang valid dengan memperhatikan apakah setiap dosen tidak ada yang diusulkan mahasiswa lebih dari kuota yang tersedia dan setiap mahasiswa tidak ada yang mengusulkan lebih dari 3 dosen.

BAB V

IMPLEMENTASI

5.1 Spesifikasi Software dan Hardware

Setelah Menyusun analisis dan perancangan sistem, tahap selanjutnya adalah implementasi untuk membangun sistem. Spesifikasi perangkat yang digunakan untuk membangun sistem algoritma genetika untuk permasalahan pemilihan dosen pembimbing tesis adalah sebagai berikut:

1. *Processor* Intel Core i7-7700HQ CPU @ 2.80GHz (8 CPUs), ~2.8GHz
2. *Memory (RAM)* 8 GB
3. *Hardisk Drive* 1 TB
4. Sistem Operasi Windows 10
5. Bahasa pemograman : Python 3.8.2
6. IDE : Visual Studio Code

5.2 Implementasi Proses

Analisis dan rancangan sistem yang telah disusun pada bab sebelumnya akan diimplementasikan ke dalam bentuk program. Pada bab ini akan dijelaskan proses, tahapan, dan dokumentasi (*source code*) dalam pembuatan program sistem algoritma genetika untuk menyelesaikan permasalahan pemilihan dosen pembimbing skripsi, mulai dari awal proses sampai mendapatkan solusi akhir permasalahan.

5.2.1 Implementasi proses pembacaan data

Data yang diperlukan pada sistem akan disimpan dalam format .csv (*comma separated value*). Pada tahap awal, file dataset yang diperlukan diimpor. Sesuai dengan Tabel 4.1, Tabel 4.2, Tabel 4.3, data yang diimpor adalah data yang menyimpan informasi nama dosen, nama mahasiswa pendaftar tesis, bidang topik penelitian yang tersedia, bidang riset dosen dan mahasiswa, dan kuota sisa setiap dosen. Untuk proses evaluasi, diperlukan juga untuk mengimpor file hasil keputusan rapat dosen pembimbing tesis. Setiap file yang diimpor menyesuaikan

dengan periode pendaftaran tesis. Kode program untuk proses pembacaan data ditunjukkan pada Gambar 5.1

```

6   # Import csv file
7   Dosen_csv = open(
8     "E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\FIX\\nama_dosen.csv", "r")
9   mhs_csv = open(
10    "E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\FIX\\List_MHS2019.csv", "r")
11   bidang_csv = open("E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\bidang.csv", "r")
12   matrix_bidangDosen_csv = open(
13     "E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\FIX\\matrix_bidangDosen.csv",
14   matrix_bidangMHS_csv = open(
15     "E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\FIX\\Bidang_2019.csv", "r")
16   kuota_csv = open(
17     "E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\FIX\\Kuota_2019.csv", "r")
18   kromosom_dp_csv = open(
19     "E:\\KULIYAH\\PROPOSAL_SKRIPSI\\SKRIPSI\\FIX\\DP_2019.csv", "r")
20
21   csvReader_dosen = csv.reader(Dosen_csv)
22   csvReader_mhs = csv.reader(mhs_csv)
23   csvReader_matrixBidangMHS = csv.reader(matrix_bidangMHS_csv)
24   csvReader_bidang = csv.reader(bidang_csv)
25   csvReader_matrixBidangDosen = csv.reader(matrix_bidangDosen_csv)
26   csvReader_kuotaDosen = csv.reader(kuota_csv)
27   csvReader_DP = csv.reader(kromosom_dp_csv)

```

Gambar 5. 1 Kode pembacaan data

5.2.2 Deklarasi parameter pengujian algoritma genetika

Parameter pengujian algoritma genetika akan dideklarasikan di awal kode program sebagai variabel global. Deklarasi parameter sebagai variabel global akan mempermudah tahap pengujian parameter-parameter pada Tabel 4.6, karena kode akan menjadi fleksibel terhadap perubahan nilai variabel . Deklarasi parameter pengujian ditunjukkan pada Gambar 5.2.

```

29   # Global Variabel
30   maxUsulan = 3
31   minUsulan = 1
32   popCount = 10
33   matingPool = int(1/2 * popCount)
34   Pc = 0.5
35   maxGen = 3
36   fitnessTarget = 80

```

Gambar 5. 2 Deklarasi variabel global

5.2.3 Implementasi *generate* kromosom

Kode untuk *generate* 1(satu) kromosom ditunjukkan pada Gambar 5.3.

```

99  def generate_kromosom():
100     kuotaTmp = []
101     for i in range(dosen.__len__()):
102         kuotaTmp.append(kuotaDosen[i])
103
104     kromosom = [[0 for x in range(mhs.__len__())]
105                 for y in range(dosen.__len__())]
106
107     # Memastikan Setiap Mahasiswa mendapatkan satu dosen
108     dosenAvailable = []
109     for y in range(dosen.__len__()):
110         if kuotaDosen[y] > 0:
111             dosenAvailable.append(y)
112
113     for x in range(mhs.__len__()):
114         r = random.randint(0, (dosenAvailable.__len__()-1))
115         randomDosen = dosenAvailable[r]
116         kromosom[randomDosen][x] = 1
117         kuotaTmp[randomDosen] -= 1
118         if kuotaTmp[randomDosen] == 0:
119             dosenAvailable.pop(r)
120
121     # Menyebarkan dosen sisanya yang masih ada kuota
122     MHS_available = []
123     usulanCount = usulanCheck(kromosom)
124     for x in range(mhs.__len__()):
125         if usulanCount[x] < maxUsulan:
126             MHS_available.append(x)
127
128     for y in range(dosenAvailable.__len__()):
129         id_dosen = dosenAvailable[y]
130         while(kuotaTmp[id_dosen] > 0):
131             r = random.randint(0, (MHS_available.__len__()-1))
132             random_MHS = MHS_available[r]
133             if(kromosom[id_dosen][random_MHS] == 0):
134                 kromosom[id_dosen][random_MHS] = 1
135                 usulanCount[random_MHS] += 1
136                 kuotaTmp[id_dosen] -= 1
137                 if usulanCount[random_MHS] >= maxUsulan:
138                     del MHS_available[r]
139
140     return kromosom

```

Gambar 5. 3 Proses *generate kromosom*

Untuk setiap proses pembangkitan kromosom, diperlukan pengecekan jumlah usulan dosen setiap mahasiswa (*usulanCount()*) dan sisa kuota dosen (*kuotaCheck()*). Untuk kedua fungsi fungsi tersebut ditunjukkan pada Gambar 5.4.

```

73     def kuotaCheck(kromosom): # Control Constrain kuota Dosen
74         count = []
75         for y in range(dosen.__len__()):
76             count.append(0)
77             for x in range(mhs.__len__()):
78                 count[y] += kromosom[y][x]
79         return count
80
81
82     def usulanCheck(kromosom): # Control Constrain dosen usulan untuk mahasiswa
83         count = []
84         for x in range(mhs.__len__()):
85             count.append(0)
86             for y in range(dosen.__len__()):
87                 count[x] += kromosom[y][x]
88         return count

```

Gambar 5. 4 Fungsi pengecekan *constrain*

5.2.3 Implementasi pembangkitan populasi

Pada proses pembangkitan populasi, akan di *generate* sejumlah n kromosom pada Populasi. Fungsi populasi akan membutuhkan variabel nilai n (*popCount*) untuk melakukan pembangkitan kromosom sebanyak n kali. Kromosom yang sudah terbentuk akan disimpan kedalam sebuah array populasi (*pop*). Implementasi pembangkitan populasi ditunjukkan pada Gambar 5.4.

```

159     def Populasi(): # Generate Populasi
160         pop = []
161         for i in range(popCount):
162             kromosom = generate_kromosom()
163             pop.append(kromosom)
164             print(f"Kromosom-{i} Generated")
165
166         return pop

```

Gambar 5. 5 Kode proses pembangkitan populasi

5.2.4 Implementasi fungsi *fitness*

Setelah populasi terbentuk, diperlukan evaluasi nilai *fitness* untuk setiap kromosom dalam populasi tersebut. Fungsi *fitness* membutuhkan *matrix_bidangMHS* dan *matrix_bidangDosen* yang sudah diimpor . Fungsi *fitness*

akan mengembalikan sebuah value *float*. Implementasi fungsi *fitness* dari persamaan 4.1 ditunjukkan pada Gambar 5.6.

```

142 def fitnessFunc(kromosom): # Evaluasi Fitness
143     fitness_val = 0
144     for x in range(mhs.__len__()):
145         for y in range(dosen.__len__()):
146             if kromosom[y][x] == 1:
147                 for r in range(bidang.__len__()):
148                     if matrix_bidangMHS[r][x] == '1' and matrix_bidangDosen[r][y] == '1':
149                         fitness_val += 1
150                         break
151                     else:
152                         continue
153                     break
154                 else:
155                     continue
156             return (fitness_val/mhs.__len__())*100

```

Gambar 5. 6 Kode fungsi *fitness*

5.2.5 Implementasi seleksi *parent*

Metode yang digunakan untuk seleksi *parent* adalah metode Baker's SUS. Karena metode ini memperhatikan nilai *fitness* kromosom pada proses seleksinya, maka diperlukan untuk mengurutkan nilai *fitness* dari kromosom pada populasi. Kromosom akan diurutkan dari yang memiliki nilai *fitness* terbesar (terbaik) sampai dengan nilai *fitness* terkecil (terburuk). Algoritma *sort* yang digunakan adalah algoritma *quicksort*. Proses mengurutkan kromosom berdasarkan dengan nilai *fitness* ditunjukkan pada Gambar 5.7.

```

169 def quickSort(sequence): # Algoritma buat sorting fitness
170     length = len(sequence)
171     if length <= 1:
172         return sequence
173     smaller, equal, larger = [], [], []
174     pivot = sequence[randint(0, len(sequence)-1)]
175     for val in sequence:
176         if val < pivot:
177             smaller.append(val)
178         elif val == pivot:
179             equal.append(val)
180         else:
181             larger.append(val)
182     return quickSort(larger)+equal+quickSort(smaller)
183
184
185 def sortFitness(fitness): # Sort Fitness Value
186     sortedFitness = []
187     sortedFitness = quickSort(fitness)
188     return sortedFitness

```

```

185 def sortFitness(fitness): # Sort Fitness Value
186     sortedFitness = []
187     sortedFitness = quickSort(fitness)
188     return sortedFitness
189

```

Gambar 5. 7 *Sort* kromosom berdasarkan nilai *fitness*

Setelah kromosom diurutkan, seleksi *parent* sudah bisa dilakukan. Untuk setiap *parent* yang terpilih, akan disimpan dalam sebuah array *parent*. Implementasi dari seleksi *parent* ditunjukkan pada Gambar 5.8.

```

210 def parentSel(kromosom_order, sortedFitness): # Seleksi Parent
211     # Hitung Fitness Probability
212     fitnessSum = 0
213     for val in sortedFitness:
214         fitnessSum += val
215     fitnessProb = []
216     for val in sortedFitness:
217         fitprob = (val/fitnessSum)
218         fitnessProb.append(fitprob)
219     print(f"Fitness Prob:{fitnessProb}")
220     # Hitung cumulative Probability
221     cumProb = [0 for i in range(fitnessProb.__len__())]
222     for i in range(cumProb.__len__()):
223         if i == 0:
224             cumProb[i] = fitnessProb[i]
225         else:
226             cumProb[i] = cumProb[i-1]+fitnessProb[i]
227     print(f"CumProb: {cumProb}")
228     # Iterasi Seleksi Parent
229     print(f"Mating Pool = {matingPool}")
230     N_Pointer = matingPool
231     parent = []
232     pointer = [0 for i in range(N_Pointer)]
233     for i in range(matingPool):
234         if i == 0:
235             pointer[i] = (random.uniform(0, 1/N_Pointer))
236             for cp in range(cumProb.__len__()):
237                 if cumProb[cp] >= pointer[i] and cumProb[cp] not in parent:
238                     parent.append(kromosom_order[cp])
239                     break
240                 else:
241                     pointer[i] = pointer[i-1]+(1/N_Pointer)
242                     for c in range(cumProb.__len__()):
243                         if cumProb[c] >= pointer[i] and cumProb[c] not in parent:
244                             parent.append(kromosom_order[c])
245                             break
246     print(f"Pointer: {pointer}")
247     return parent

```

Gambar 5. 8 Kode seleksi *parent*

5.2.6 Implementasi *crossover*

Metode *crossover* yang digunakan adalah metode *uniform crossover*, dimana diperlukan pembangkitan pola *random* yang sesuai dengan representasi kromosom. Pola akan dibangkitkan untuk setiap generasi. Pola untuk setiap pasangan *parent* pada proses *crossover* sama pada suatu generasi. Implementasi untuk pembangkitan pola ditunjukkan pada Gambar 5.9.

```

257     def generate_Pattern():
258         pattern = [[0 for x in range(mhs.__len__())]
259                    for y in range(dosen.__len__())]
260         for y in range(dosen.__len__()):
261             for x in range(mhs.__len__()):
262                 if kuotaDosen[y] > 0:
263                     r = random.randint(0, 1)
264                     pattern[y][x] = r
265         return pattern
266

```

Gambar 5. 9 Kode pembangkitan pola *random*

Sebelum dilakukan proses *crossover*, perlu dibangkitkan suatu nilai *random* untuk setiap pasangan *parent* dan nilai tersebut dibandingkan dengan nilai *Pc*. Apabila nilai *random* untuk suatu pasangan *parent* lebih kecil dibandingkan nilai *Pc*, maka pasangan *parent* tersebut melakukan proses *crossover* dan *offspring* yang dihasilkan akan dimasukkan kedalam array *offspring*. Proses pengecekan tersebut ditunjukkan pada Gambar 5.10.

Apabila suatu pasangan *parent* lolos untuk pengecekan nilai *Pc*, maka kedua *parent* tersebut akan dibawa ke fungsi *crossover* (*do_crossover()*) sebagai parameter. Selanjutnya adalah melakukan pengecekan dan perbaikan *constraint* usulan mahasiswa (*Mx*) terhadap kromosom *offspring* yang dihasilkan. Proses *crossover* ditunjukkan pada Gambar 5.11.

```

312     def CrossOver(parent, pop, pattern):
313         print(f"PC = {Pc}")
314         print(f"Pola Crossover:")
315         print(f"\n{print_kromosom(pattern)}")
316         Offspring = []
317         for i in range(len(parent)):
318             for j in range((i+1), len(parent)):
319                 r = random.random() # Random Float antara 0 dan 1
320                 print(f"Random Number Generated: {r}")
321                 if(r <= Pc):
322                     _parent = []
323                     _parent.append(parent[i])
324                     _parent.append(parent[j])
325                     print(f"{parent[i]} and {parent[j]} OK")
326                     pKromosom = getParent_krom(pop, _parent)
327                     # Un-Comment ini buat nampilin Kromosom Parent
328                     ofKrom = do_crossOver(pKromosom, pattern)
329                     for krom in ofKrom:
330                         Offspring.append(krom)
331                 else:
332                     print(f"{parent[i]} and {parent[j]} NOT OK")
333
334     return Offspring

```

Gambar 5. 10 Proses pengecekan nilai *random* dengan *Pc*

```

286     # Versi Uniform Crossover, random uniform array per generasi
287     def do_crossOver(pKromosom, pattern):
288         # DeclareParent
289         p1 = pKromosom[0]
290         p2 = pKromosom[1]
291         # Declare Offspring
292         o1 = [[0 for x in range(mhs.__len__())]
293               for y in range(dosen.__len__())]
294         o2 = [[0 for x in range(mhs.__len__())]
295               for y in range(dosen.__len__())]
296         ofKrom = []
297         for y in range(dosen.__len__()):
298             for x in range(mhs.__len__()):
299                 # Offspring 1: 0 -> dari parent 1, 1-> dari parent 2
300                 # Offspring 2: 0 -> dari parent 2, 1-> dari parent 1
301                 if pattern[y][x] == 0:
302                     o1[y][x] = p1[y][x]
303                     o2[y][x] = p2[y][x]
304                 elif pattern[y][x] == 1:
305                     o1[y][x] = p2[y][x]
306                     o2[y][x] = p1[y][x]
307         ofKrom.append(repair_usulan(o1))
308         ofKrom.append(repair_usulan(o2))
309
310     return ofKrom

```

Gambar 5. 11a Kode *uniform crossover*

```

268 # Cek apakah ada mahasiswa yang mendapatkan dosen lebih dari maxUsulan
269 def repair_usulan(kromosom_offspring):
270     check = usulanCheck(kromosom_offspring)
271     for x in range(mhs.__len__()):
272         value1_cont = [] # container gen (y,x) Value 1
273         if check[x] > maxUsulan: # Cek Kromosom 1
274             for y in range(dosen.__len__()):
275                 if kromosom_offspring[y][x] == 1:
276                     value1_cont.append(y)
277             over = check[x] - maxUsulan # berapa lebihnya
278             for _ in range(over):
279                 r = random.randint(0, (value1_cont.__len__()-1))
280                 k = value1_cont[r]
281                 kromosom_offspring[k][x] = 0
282                 del value1_cont[r]
283     return kromosom_offspring
284

```

Gambar 5. 11b Proses perbaikan *constrain* usulan mahasiswa

5.2.7 Implementasi mutasi

Sebuah kromosom *offspring* akan melakukan proses mutasi apabila terdapat dosen yang diusulkan mahasiswa melebihi dari kuota tersedia dosen tersebut. Sesuai dengan alur mutasi yang dijelaskan pada Gambar 4.9, sebelum dilakukan proses mutasi, diperlukan sebuah fungsi untuk mengecek kuota tersedia dari setiap dosen pada kromosom *offspring*. Kode fungsi dari proses pengecekan ini ditunjukkan pada Gambar 5.12.

```

337 def check_kuota_offspring(offspring, kromNum): # Proses Mutasi
338     selisih = [0 for i in range(dosen.__len__())]
339     off_Kuota = kuotaCheck(offspring[kromNum])
340     # Kuota Maksimal Setiap Dosen
341     for i in range(kuotaDosen.__len__()):
342         selisih[i] = kuotaDosen[i] - off_Kuota[i]
343     return selisih
344

```

Gambar 5. 12 Kode fungsi pengecekan sisa kuota dosen pada kromosom *offspring*

Dari kode tersebut, sisa kuota dosen akan bernilai negatif apabila dosen tersebut diusulkan lebih dari kuota yang tersedia. Apabila pada kromosom *offspring* setidaknya ditemukan 1 dosen yang memiliki sisa kuota negatif, maka kromosom tersebut dianggap *infeasible* dan akan melalui proses mutasi. Kode untuk pengecekan apakah suatu kromosom *infeasible* ditunjukkan pada Gambar 5.13 dan kode untuk proses mutasi ditunjukkan pada Gambar 5.14.

```

393 def mutasi(offspring, kromNum):
394     unValid = 0
395     Idx = [] # Container Index dimana Row pada Kromosom dilanggar
396     sisaKuota = check_kuota_offspring(offspring, kromNum)
397     of_after_mutasi = [
398         [0 for x in range(mhs.__len__())] for y in range(dosen.__len__())
399     print(f"Sisa Kuota dosen Kromosom-{kromNum} : {sisaKuota}")
400     for i in range(sisaKuota.__len__()):
401         if sisaKuota[i] < 0:
402             unValid = 1
403             Idx.append(i)
404     if unValid > 0:
405         print(f"Kromosom Offspring-{kromNum} is Infeasible!! Must Mutate!!")
406         of_after_mutasi = do_mutasi(Idx, offspring[kromNum], sisaKuota)
407     elif unValid == 0:
408         print(f"Kromosom Offspring-{kromNum} is feasible!")
409         of_after_mutasi = offspring[kromNum]
410     return of_after_mutasi

```

Gambar 5. 13 Pengecekan *infeasible* kromosom

```

346 def do_mutasi(Idx, ofKrom, sisakuota):
347     m_krom = ofKrom
348     col_to_mutate = 0
349     print(Idx)
350     if len(Idx) > 0:
351         for val in Idx:
352             # Memilih Lokasi random-j pada Gen[Idx][x] dengan value gen = 1
353             gen_val1 = [] # Kontainer Gen dengan Value=1
354             for x in range(mhs.__len__()):
355                 if m_krom[val][x] == 1:
356                     # Simpan Informasi index x kedalam container
357                     gen_val1.append(x)
358             # Pilih lokasi random yang tersedia pada kontainer
359             col_to_mutate = gen_val1[random.randint(0, (gen_val1.__len__()-1))]
360             m_krom[val][col_to_mutate] = 0
361             sisakuota[val] += 1 # Tampungannya jadi semakin positif
362             # Jika sudah positif Remove dari list
363             if sisakuota[val] >= 0:
364                 Idx.remove(val)
365             # Selama sisa Kuota pada Kromosom ini ada yang masih negatif:
366             do_mutasi(Idx, m_krom, sisakuota)
367     # Memastikan setiap mahasiswa minimal dapat satu dosen
368     elif len(Idx) == 0:
369         # print(f'sisa Kuota = {sisakuota}')
370         usulan_sum = usulanCheck(m_krom)
371         MHS_to_repair = []
372         dosenAvailable = []
373         for x in range(mhs.__len__()):
374             if usulan_sum[x] < minUsulan:
375                 MHS_to_repair.append(x)
376         # print(f"MHS to Repair: {MHS_to_repair}")
377         for i in range(sisakuota.__len__()):
378             if sisakuota[i] > 0:
379                 dosenAvailable.append(i)
380         # print(f"Dosen Available: {dosenAvailable}")

```

```

380     for val in MHS_to_repair:
381         if dosenAvailable.__len__() > 0:
382             y = random.choice(dosenAvailable)
383             # print(f"Random Dosen = {y}")
384             # print(f"Gen {val},{y} = 1")
385             m_krom[y][val] = 1
386             sisaKuota[y] -= 1
387             if sisaKuota[y] <= 0:
388                 dosenAvailable.remove(y)
389             # Kalau misalnya gaada dosen yang available,
390             # mahasiswa yg mengusulkan lebih dari satu oper dosen
391         else:
392             usulan_lebih_dari1 = []
393             dos1 = []
394             c = usulanCheck(m_krom)
395             for i in range(c.__len__()):
396                 if c[i] > minusUsulan:
397                     usulan_lebih_dari1.append(i)
398             rx = random.choice(usulan_lebih_dari1)
399             for y in range(dosen.__len__()):
400                 if m_krom[y][rx] == 1:
401                     dos1.append(y)
402             ry = random.choice(dos1)
403             m_krom[ry][rx] = 0
404             m_krom[ry][val] = 1
405
406     return m_krom
407 # Proses Mutasi Selesai

```

Gambar 5. 14 Kode proses mutasi

5.2.8 Implementasi seleksi survivor

Setelah dilakukan proses mutasi, seluruh kromosom sudah feasible dan mampu untuk dilanjutkan pada proses seleksi survivor. Kromosom populasi awal akan digabungkan dengan kromosom offspring yang dihasilkan (disimpan dalam sebuah array survivor). Selanjutnya kromosom dalam array survivor akan diurutkan dari kromosom dengan nilai fitness terbesar sampai kromosom dengan nilai fitness terburuk dengan algoritma quicksort. Akan diambil sebanyak n kromosom terbaik dari array survivor ($n <$ panjang array survivor) sebagai populasi baru untuk generasi selanjutnya. Proses seleksi survivor ditunjukkan pada Gambar 5.15.

```

455  def survivor(pop, offspring):
456      _s = []
457      _survivor = []
● 458  for krom in pop:
459      |   _s.append(krom)
460  for krom in offspring:
461      |   _s.append(krom)
462      _survivor_fitness = []
463  for krom in _s:
464      |   _survivor_fitness.append(fitnessFunc(krom))
465      sorted_fitness = sortFitness(_survivor_fitness)
466  sorted_kromosom = sort_kromosom_by_fitness(
467      |   sorted_fitness, _survivor_fitness)
468  for kromNum in sorted_kromosom:
469      |   _survivor.append(_s[kromNum])
470  return _survivor
471
472
473  def elitisme(_survivor, popCount):
474      nextGen = []
475  for i in range(popCount):
476      |   nextGen.append(_survivor[i])
477  return nextGen

```

Gambar 5. 15 Kode proses seleksi *survivor*

5.2.9 Implementasi *cycle* algoritma genetika dan kondisi pemberhentian

Pada Gambar 4.1, proses algoritma genetika akan terus terulang sampai kondisi pemberhentian terpenuhi. Terdapat 2 kondisi pemberhentian, pertama ketika nilai *fitness* terbaik dari suatu generasi sudah mencapai atau melebihi *fitness* target , kedua ketika nilai *fitness* target masih belum terpenuhi, namun banyaknya generasi sudah mencapai generasi maksimal. *Cycle* algoritma genetika akan diimplementasikan dengan metode *recursive* yang membutuhkan parameter populasi baru (kromosom yang lolos tahap seleksi *survivor*), nilai *fitness* dari kromosom pada populasi baru, dan *counter* generasi untuk mengecek sudah banyak generasi yang dihasilkan. Kode *Cycle* algoritma genetika ditunjukkan pada Gambar 5.16a.

```

530  def GeneticAlgorithm(pop, kromFitness, genCount): # Algoritma Genetika
531      if genCount > 1:
532          |   print(f"\nGeneration: {genCount}")
533
534      # sort Kromosom berdasarkan fitness value dari yang paling kecil
535      fitness = kromFitness
536      sorted_fitness = sortFitness(fitness)
537      kromosom_order = sort_kromosom_by_fitness(sorted_fitness, fitness)
538      print_kromosom_by_fitness(kromosom_order, sorted_fitness)
539

```

```

540     # Proses Parent Selection:
541     print(f"\nSelecting Parent....")
542     parent = parentSel(kromosom_order, sorted_fitness)
543     print("Chosen Kromosom as Parent:")
544     for val in parent:
545         print(f"Kromosom-{val}")
546         # print Kromosom Parent
547         # print_kromosom(pop[val])
548
549     # Proses CrossOver
550     print(f"\nCrossOver.....")
551     pattern = generate_Pattern()
552     Offspring = CrossOver(parent, pop, pattern)
553     print(f"CrossOver Finished! {Offspring.__len__()} offspring(s) produced")
554     # for krom in Offspring:
555     #     print_kromosom(krom)
556
557     # Proses Mutasi
558     print(f"\nMutation.....")
559     of_after_mutasi = []
560     for i in range(Offspring.__len__()):
561         print(f"Tumpungan Dosen Kromosom-{i} : {kuotaCheck(Offspring[i])}")
562         of_after_mutasi.append(mutasi(Offspring, i))
563     # Uncomment buat ngecek Tumpungan Setiap Dosen Setelah Mutasi
564     print(f"\n")
565     for i in range(of_after_mutasi.__len__()):
566         dosenMHS = usulanCheck(of_after_mutasi[i])
567         print(
568             f"Jumlah Usulan Dosen per Mahasiswa setelah Mutasi Kromosom-{i} : {dosenMHS}")
569         print(
570             f"Sisa Kuota Dosen setelah Mutasi: {check_kuota_offspring(of_after_mutasi, i)}")
571
572     # Hitung Fitness dari Offspring
573     print(f"\nOffspring fitness Value:")
574     offFitness = []
575     for krom in range(of_after_mutasi.__len__()):
576         offFitness.append(fitnessFunc(of_after_mutasi[krom]))
577     sorted_of_fitness = sortFitness(offFitness)
578     kromosom_of_order = sort_kromosom_by_fitness(sorted_of_fitness, offFitness)
579     print_kromosom_by_fitness(kromosom_of_order, sorted_of_fitness)
580
581     # Seleksi Survivor
582
583     print(f"\nSeleksi survivor.....")
584     # _survivor = survivor(of_after_mutasi, pop, parent, fitness, offFitness)
585     _survivor = survivor(pop, of_after_mutasi)
586     NextGen = elitisme(_survivor, popCount)
587     NextGen_Fitness = []
588     bestKromosom = 0
589     for val in NextGen:
590         NextGen_Fitness.append(fitnessFunc(val))
591     print(f"Next Generation: ")
592     for i in range(NextGen.__len__()):
593         print(f"Kromosom-{i} = {NextGen_Fitness[i]} (%)")
594     bestFitness = max(NextGen_Fitness)
595     print(f"Best Fitness = {bestFitness}")
596     if genCount < maxGen:
597         GeneticAlgorithm(NextGen, NextGen_Fitness, genCount+1)

```

Gambar 5. 16a Kode *cycle* algoritma genetika jika kondisi pemberhentian tidak terpenuhi

Apabila salah satu dari kondisi pemberhentian terpenuhi, maka *cycle* algoritma genetika akan berhenti dan menampilkan hasil/solusi. Lanjutan kode dari Gambar 5.16a sebagai kondisi pemberhentian ditunjukkan pada Gambar 5.16b dan fungsi untuk memunculkan hasil/solusi ditunjukkan pada Gambar 5.17.

```

581     elif bestFitness >= fitnessTarget:
582         print(f"Fitness Target Reached!")
583         bestKromosom = NextGen[NextGen_Fitness.index(bestFitness)]
584         print_result(bestFitness, bestKromosom)
585         Evaluasi(bestKromosom)
586     elif genCount >= maxGen:
587         print(f"Max Generation Reached!")
588         bestKromosom = NextGen[NextGen_Fitness.index(bestFitness)]
589         print_result(bestFitness, bestKromosom)
590         Evaluasi(bestKromosom)

```

Gambar 5. 16b Kode kondisi pemberhentian *cycle* algoritma genetika

```

496 ✓ def print_resultv2(result):
497     print(f"\nResult:")
498     print_kromosom(result)
499     # Print Mahasiswa dengan Dosen Usulannya
500 ✓     for x in range(mhs.__len__()):
501         print(f"\nUsulan Dosen untuk Mahasiswa: {mhs[x]}:")
502 ✓         for r in range(bidang.__len__()):
503 ✓             if matrix_bidangMHS[r][x] == "1":
504                 print(f"({bidang[r]})")
505 ✓             for y in range(dosen.__len__()):
506 ✓                 if result[y][x] == 1:
507                     print(f"{dosen[y]}")
508     print(f"\n")
509     # print dosen dan mahasiswa yang cocok
510     fitnessFunc_result(result)
511     # Print Sisa Kuota Dosen:
512     sisa_kuota = [0 for i in range(dosen.__len__())]
513     tampungan = kuotaCheck(result)
514 ✓     for i in range(dosen.__len__()):
515         sisa_kuota[i] = kuotaDosen[i] - tampungan[i]
516     print(f"\nSisa Kuota Dosen:{sisa_kuota} ")

```

Gambar 5. 17 Kode untuk menampilkan hasil/solusi

5.2.10 Implementasi evaluasi

Setelah didapatkan hasil akhir/solusi dari proses algoritma genetika, perlu dilakukan evaluasi untuk melihat performa dari sistem algoritma genetika. Hasil

```

498  def Evaluasi(bestKromosom):
499      # Evaluasi
500      print(f"Perbandingan dengan Hasil Keputusan Rapat:")
501      print(f"\n Nilai Fitness: {fitnessFunc(DP)}")
502      print_kromosom(DP)
503      sum = 0
504      for y in range(dosen.__len__()):
505          for x in range(mhs.__len__()):
506              if bestKromosom[y][x] == 1 and DP[y][x] == 1:
507                  sum += 1
508
509      acc = (sum/mhs.__len__())*100
510      print(f"Akurasi: {acc}(% MATCH!!")
```

Gambar 5. 18 Kode evaluasi hasil akhir/solusi

akhir/solusi akan dibandingkan dengan data hasil keputusan rapat dosen pembimbing tesis. Data yang sudah diubah kedalam bentuk kromosom dan diimpor pada kode akan dievaluasi nilai *fitness*nya. Setelah dievaluasi, akan terlihat apakah solusi dari algoritma genetika lebih baik atau lebih buruk daripada hasil keputusan rapat. Kromosom solusi juga akan dicek seberapa mirip dengan data hasil keputusan rapat. Kode untuk proses evaluasi ditunjukkan pada Gambar 5.18.

5.3 Main Function

Setiap proses dan tahap algoritma genetika yang sudah diimplementasikan dalam bentuk kode program akan dijalankan pada *main function* yang ditunjukkan pada Gambar 5.19.

```

610  def main():
611
612      genCount = 1
613      # Hitung total kuota dosen
614      Print_kuota()
615      total_kuota = 0
616      pop = []
617      for val in kuotaDosen:
618          total_kuota += val
619      print(f"Total Kuota:{total_kuota}")
620      print(f"Jumlah Mahasiswa pendaftar:{mhs.__len__()}")
621      if (total_kuota < mhs.__len__()):
622          print("Pendaftar Skripsi Lebih Banyak daripada kuota total Dosen!!")
```

```
623 elif (total_kuota > mhs.__len__()):  
624     print(f"\nGeneration: {genCount}")  
625     # Generate Kromosom dalam Populasi  
626     startTime = time.time()  
627     print(f"\nGenerating Kromosom.....")  
628     pop = Populasi()  
629     generate_kromosom_time = time.time() - startTime  
630     print(f"Generate Kromosom Time: {generate_kromosom_time}")  
631     # Hitung Fitness Value  
632     print(f"\nCalculating Fitness Value...")  
633     fitness = []  
634     for i in range(popCount):  
635         fitness_val = fitnessFunc(pop[i])  
636         fitness.append(fitness_val)  
637     # print(f"Fitness Kromosom-{i} = {fitness_val}%")  
638     GeneticAlgorithm(pop, fitness, genCount)  
639     total_time = time.time() - startTime  
640     print(f"Total Time: {total_time}")  
641  
642  
643 if __name__ == "__main__":  
644     # Program Utama  
645     main()
```

Gambar 5. 19 Kode main function

Dari Gambar 5.19, sistem algoritma genetika untuk alokasi dosen pembimbing tesis tidak akan jalan apabila jumlah kuota tersedia dari dosen lebih sedikit daripada jumlah mahasiswa pendaftar tesis.

BAB VI

HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan proses, hasil, dan peforma pengujian untuk sistem algoritma genetika yang dikembangkan . Pengujian dilakukan untuk 2 dataset, yaitu dataset pendaftaran tesis periode 2018/2019 dan dataset pendaftaran tesis periode 2019/2020.

6.1 Pengujian dengan Nilai Parameter Algoritma Genetika

Tahap awal yang dilakukan yaitu pengujian dengan menggunakan variasi parameter-parameter algoritma genetika, yaitu nilai probabilitas *crossover* (Pc), ukuran populasi, dan nilai generasi maksimal. Pengujian parameter dilakukan untuk mengetahui nilai parameter-parameter yang baik untuk pencarian solusi. Setiap parameter akan diujikan pada program yang sudah dikembangkan untuk masing-masing dataset dan dilakukan percobaan sebanyak 10 kali.

6.1.1 Pengujian parameter Pc

Pengujian ini dilakuakn untuk mengetahui pengaruh probabilitas *crossover* (Pc) terhadap solusi yang diperoleh. Sebagaimana tertulis pada Tabel 4.5, parameter Pc yang diujikan adalah 0,6 ; 0,7; 0,8; dan 0,9 . Sementara Pc akan diujikan pada ukuran populasi 10 kromosom dan 10 generasi maksimal. Hasil yang diperoleh akan dihitung rata-rata dan durasi waktu sistem dalam menjalankan algoritma. Durasi pengujian ditunjukkan pada Gambar 6.1 dan nilai *fitness* solusi ditunjukkan pada Gambar 6.2.

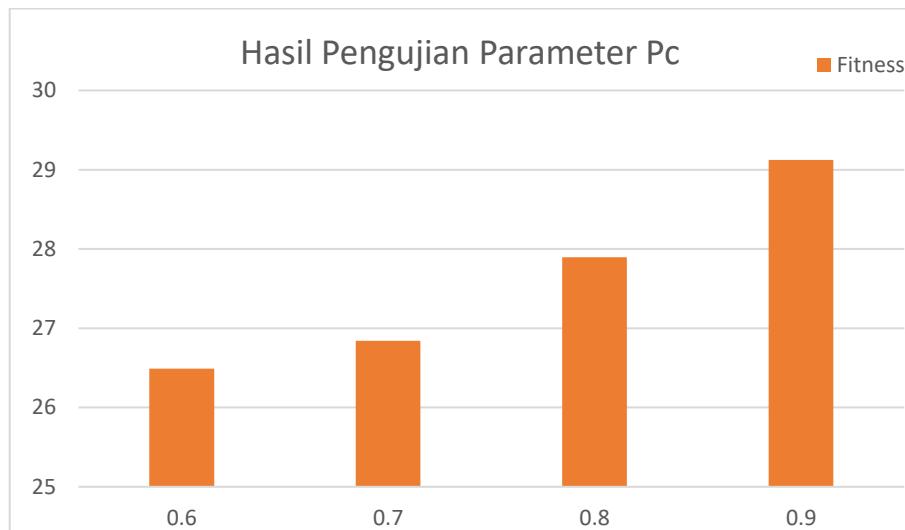
Tabel 6. 1 Durasi pengujian parameter Pc

| Percobaan ke- | Durasi (detik) | | | |
|----------------------|-----------------------|-------------|-------------|-------------|
| | PC | 0.6 | 0.7 | 0.8 |
| 1 | 2.764710665 | 3.229804993 | 3.772608995 | 4.297745466 |
| 2 | 2.542061806 | 2.676974058 | 3.096548557 | 3.650126219 |
| 3 | 2.405750275 | 2.428834677 | 3.387766838 | 3.286896229 |
| 4 | 2.667007446 | 2.487168789 | 3.365776062 | 3.526962519 |
| 5 | 2.799104691 | 2.535722017 | 3.138920784 | 3.593335152 |
| 6 | 2.393543959 | 2.623030186 | 3.393528461 | 3.634650469 |
| 7 | 2.565675974 | 2.599478006 | 3.390169859 | 3.439420938 |
| 8 | 2.65069294 | 2.437003136 | 3.278598547 | 3.433617115 |
| 9 | 2.318190575 | 2.742924213 | 3.452619791 | 3.655783892 |
| 10 | 2.37699461 | 2.199640274 | 3.395653486 | 3.364763021 |
| Rata-Rata | 2.548373294 | 2.596058035 | 3.36721914 | 3.588330102 |

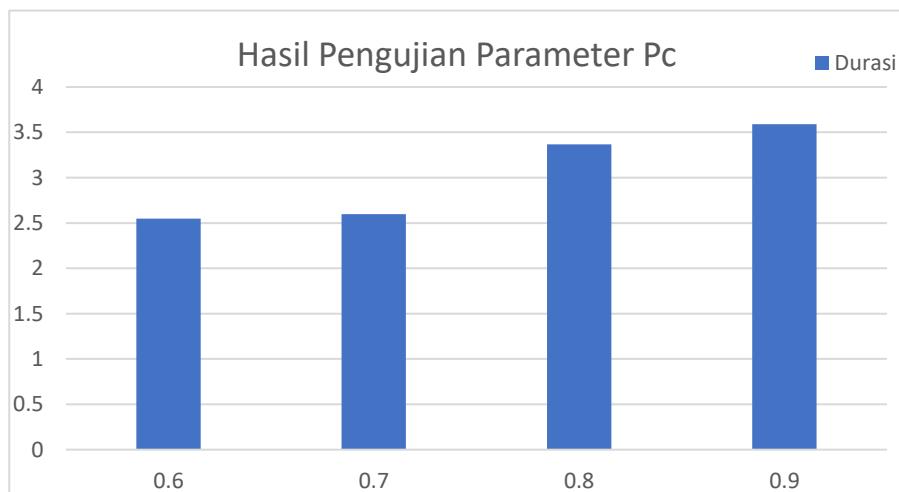
Tabel 6. 2 Nilai *fitness* pengujian parameter Pc

| Percobaan ke- | fitness value Hasil | | | |
|----------------------|----------------------------|-------------|-------------|-------------|
| | PC | 0.6 | 0.7 | 0.8 |
| 1 | 24.56140351 | 28.07017544 | 26.31578947 | 31.57894737 |
| 2 | 28.07017544 | 29.8245614 | 29.8245614 | 26.31578947 |
| 3 | 29.8245614 | 22.80701754 | 26.31578947 | 26.31578947 |
| 4 | 28.07017544 | 22.80701754 | 24.56140351 | 26.31578947 |
| 5 | 24.56140351 | 33.33333333 | 31.57894737 | 33.33333333 |
| 6 | 24.56140351 | 22.80701754 | 28.07017544 | 26.31578947 |
| 7 | 28.07017544 | 29.8245614 | 26.31578947 | 28.07017544 |
| 8 | 26.31578947 | 24.56140351 | 26.31578947 | 26.31578947 |
| 9 | 26.31578947 | 33.33333333 | 33.33333333 | 33.33333333 |
| 10 | 24.56140351 | 21.05263158 | 26.31578947 | 33.33333333 |
| Rata-Rata | 26.49122807 | 26.84210526 | 27.89473684 | 29.12280702 |

Grafik untuk hasil pengujian parameter Pc ditunjukkan pada Gambar 6.1 dan Gambar 6.2



Gambar 6. 1 Grafik nilai *fitness* solusi pengujian parameter P_c



Gambar 6. 2 Grafik durasi pengujian paramter P_c

Dari hasil pengujian, rata-rata durasi tercepat diperoleh pada $P_c = 0.6$, yaitu sekitar 2,5 detik. Jika dibandingkan dengan parameter yang lain, pada $P_c = 0.6$ diperoleh rata-rata nilai *fitness* terkecil, tetapi selisih dengan rata-rata nilai *fitness* parameter P_c yang lain dianggap tidak signifikan. Maka untuk tahap pengujian selanjutnya akan digunakan parameter $P_c = 0.6$.

6.1.2 Pengujian parameter ukuran populasi

Pengujian ukuran populasi dilakukan untuk melihat pengaruh jumlah anggota pada populasi terhadap nilai *fitness* solusi dan durasi untuk pencarian solusi. Sesuai dengan Tabel 4.5, masing-masing jumlah anggota pada populasi yang akan diuji

adalah 10, 25, dan 50 anggota. Berdasarkan pengujian Pc yang sudah dilakukan, maka Pc yang digunakan adalah 0,6. Jumlah maksimal generasi yang digunakan adalah 10 generasi. Hasil yang diperoleh akan dihitung rata-rata durasi pengujian dan rata-rata nilai *fitness* solusi. Hasil dari durasi pengujian ukuran populasi ditunjukkan pada Tabel 6.3 dan nilai *fitness* solusi ditunjukkan pada Tabel 6.4.

Tabel 6. 3 Durasi pengujian parameter ukuran populasi

| Percobaan ke- | Durasi (detik) | | |
|---------------|-----------------|-------------|-------------|
| | Ukuran Populasi | | |
| | 10 | 25 | 50 |
| 1 | 2.657889843 | 14.71939111 | 55.71197271 |
| 2 | 2.569436073 | 13.00936961 | 52.39019728 |
| 3 | 3.011395931 | 12.97320008 | 50.32319689 |
| 4 | 3.006711245 | 13.57219577 | 51.01452208 |
| 5 | 2.725593328 | 14.42011118 | 53.80744529 |
| 6 | 2.702562094 | 13.31655264 | 57.53849721 |
| 7 | 2.590945244 | 15.85565877 | 53.56279874 |
| 8 | 2.777313471 | 12.47440696 | 53.8067112 |
| 9 | 3.067092419 | 12.32085657 | 57.80953193 |
| 10 | 2.719545126 | 13.26796269 | 60.82411218 |
| Rata-Rata | 2.782848477 | 13.59297054 | 54.67889855 |

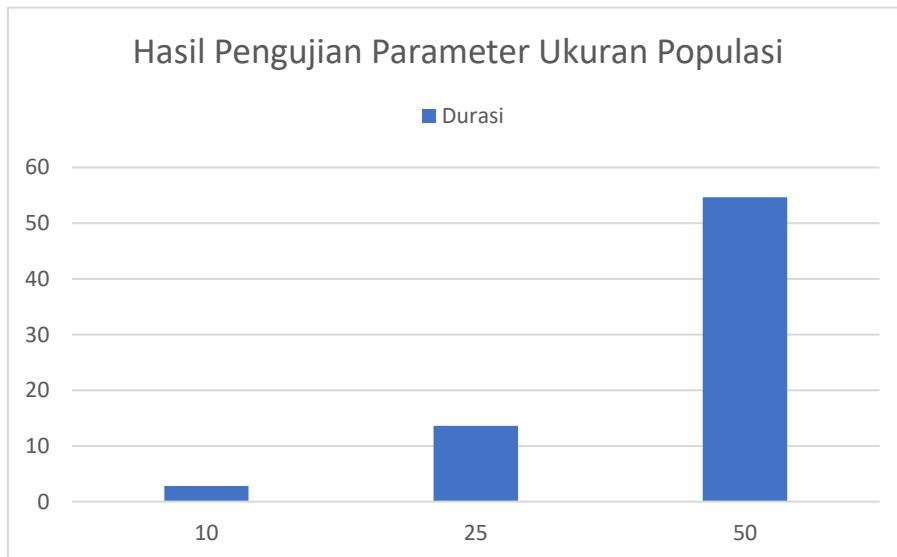
Tabel 6. 4Nilai *fitness* pengujian parameter ukuran populasi

| Percobaan ke- | nilai <i>fitness</i> | | |
|---------------|----------------------|-------------|-------------|
| | Ukuran Populasi | | |
| | 10 | 25 | 50 |
| 1 | 22.80701754 | 38.59649123 | 50.87719298 |
| 2 | 22.80701754 | 36.84210526 | 47.36842105 |
| 3 | 31.57894737 | 35.0877193 | 45.61403509 |
| 4 | 26.31578947 | 36.84210526 | 49.12280702 |
| 5 | 28.07017544 | 42.10526316 | 49.12280702 |
| 6 | 24.56140351 | 35.0877193 | 47.36842105 |
| 7 | 24.56140351 | 36.84210526 | 47.36842105 |
| 8 | 24.56140351 | 40.35087719 | 50.87719298 |
| 9 | 28.07017544 | 40.35087719 | 47.36842105 |
| 10 | 28.07017544 | 42.10526316 | 45.61403509 |
| Rata-Rata | 26.14035088 | 38.42105263 | 48.07017544 |

Grafik hasil pengujian parameter ukuran populasi ditunjukkan pada Gambar 6.3 dan Gambar 6.4.



Gambar 6. 4 Grafik nilai *fitness* pengujian parameter ukuran populasi



Gambar 6. 3 Grafik durasi pengujian parameter ukuran populasi

Dari hasil pengujian, pada ukuran populasi 50 anggota didapatkan rata-rata nilai *fitness* terbaik, tetapi jika dibandingkan dengan nilai *fitness* ukuran populasi 25 anggota, selisih rata-rata nilai *fitness* tidak signifikan, tetapi selisih durasi ukuran populasi tersebut mencapai sekitar 40 detik. Selisih nilai *fitness* ukuran populasi 25

anggota dengan ukuran populasi 10 anggota juga tidak signifikan, tetapi pada ukuran populasi 10 anggota rata-rata durasi lebih cepat. Maka pada pengujian selanjutnya akan digunakan ukuran populasi 10 anggota.

6.1.3 Pengujian parameter generasi maksimum

Pengujian generasi maksimum dilakukan untuk mengetahui banyaknya generasi maksimal yang diperlukan apabila nilai *fitness* dari kromosom tidak mencapai *fitness* target. Nilai *fitness* target adalah 80. Sesuai Tabel 4.5, banyaknya generasi maksimal yang akan diujikan adalah 100, 1000, dan 1500 generasi. Berdasarkan pengujian yang sudah dilakukan untuk parameter Pc dan ukuran populasi, nilai Pc yang digunakan adalah 0,6 dan ukuran populasi sebanyak 10 anggota. Hasil yang diperoleh akan dihitung rata-rata durasi pengujian dan rata-rata nilai *fitness* solusi. Durasi dan nilai *fitness* solusi untuk pengujian parameter generasi maksimum ditunjukkan pada Tabel 6.5 dan Tabel 6.6.

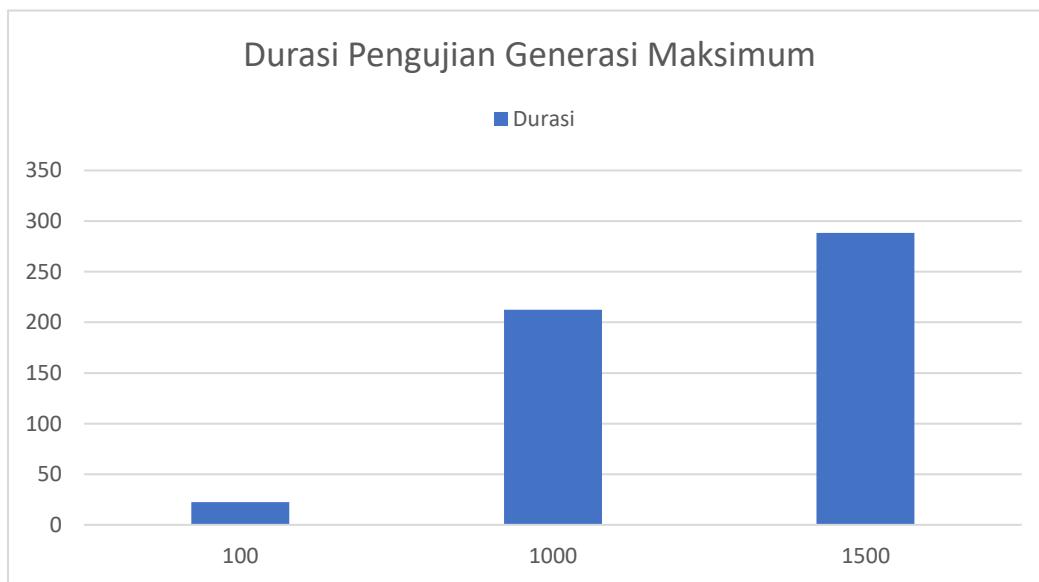
Tabel 6. 5 Durasi pengujian parameter generasi maksimum

| Percobaan ke- | Durasi (detik) | | |
|----------------------|-----------------------|--------------------|--------------------|
| | 100 | 1000 | 1500 |
| 1 | 24.09728551 | 189.2703946 | 298.2290804 |
| 2 | 23.76967597 | 204.7533071 | 289.0124176 |
| 3 | 28.17936087 | 232.1586072 | 290.5969274 |
| 4 | 23.05888367 | 228.0236328 | 281.5549848 |
| 5 | 20.79526782 | 252.3755713 | 285.2938132 |
| 6 | 22.60435414 | 186.9471569 | 302.1561735 |
| 7 | 21.3151269 | 223.2503753 | 290.0607333 |
| 8 | 18.5150311 | 191.9367137 | 293.0216272 |
| 9 | 20.47898912 | 206.9439368 | 308.7619772 |
| 10 | 21.91682982 | 208.7443154 | 243.9421728 |
| Rata-Rata | 22.47308049 | 212.4404011 | 288.2629907 |

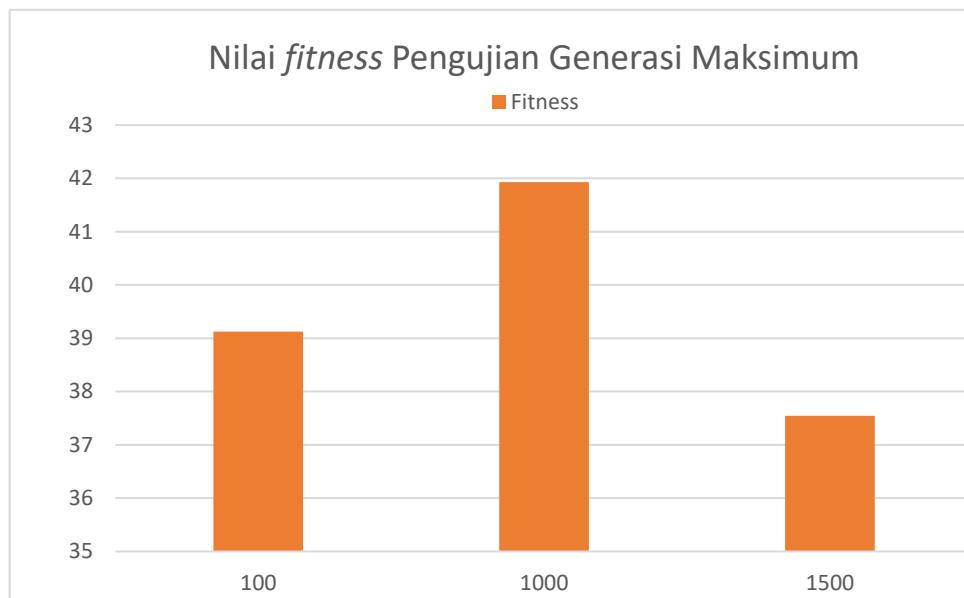
Tabel 6. 6 Nilai *fitness* pengujian parameter generasi maksimum

| Percobaan ke- | Nilai <i>fitness</i> | | |
|----------------------|-----------------------------|-------------|-------------|
| | 100 | 1000 | 1500 |
| 1 | 43.85964912 | 43.85964912 | 33.33333333 |
| 2 | 38.59649123 | 45.61403509 | 38.59649123 |
| 3 | 38.59649123 | 40.35087719 | 40.35087719 |
| 4 | 40.35087719 | 40.35087719 | 36.84210526 |
| 5 | 36.84210526 | 45.61403509 | 43.85964912 |
| 6 | 38.59649123 | 38.59649123 | 42.10526316 |
| 7 | 36.84210526 | 33.33333333 | 35.0877193 |
| 8 | 49.12280702 | 40.35087719 | 36.84210526 |
| 9 | 36.84210526 | 45.61403509 | 35.0877193 |
| 10 | 31.57894737 | 45.61403509 | 33.33333333 |
| Rata-Rata | 39.12280702 | 41.92982456 | 37.54385965 |

Grafik rata-rata durasi pengujian dan rata-rata nilai *fitness* pengujian generasi maksimum ditunjukkan pada Gambar 6.5 dan Gambar 6.6.



Gambar 6. 5 Grafik durasi pengujian generasi maksimum



Gambar 6. 6 Grafik nilai *fitness* pengujian generasi maksimum

Dari pengujian generasi maksimum, didapatkan rata-rata nilai *fitness* solusi terbaik pada generasi maksimum sejumlah 1000 generasi. Maka untuk pengujian selanjutnya, digunakan 1000 generasi sebagai jumlah generasi maksimum.

Dari hasil pengujian parameter-parameter algoritma genetika, maka parameter yang digunakan untuk pencarian solusi ditunjukkan pada Tabel 6.7.

Tabel 6. 7 Hasil pengujian paramter algotitma genetika

| Parameter | Nilai |
|-------------------|-------|
| Pc | 0.6 |
| Ukuran Populasi | 10 |
| Generasi Maksimum | 1000 |

6.2 Hasil Pencarian Solusi

Pencarian solusi dilakukan pada dua dataset, yaitu dataset pendaftaran tesis periode 2018/2019 semester genap dan dataset periode 2019/2020 semester genap dengan menggunakan parameter pada Tabel 6.7.

Setelah data diproses pada sistem algoritma genetika yang telah dibuat, diperoleh kromosom solusi. Kromosom solusi akan melalui proses *decoding* untuk menampilkan hasil penyelesaian dengan algoritma genetika.

6.2.1 Hasil pencarian solusi dataset periode 2018/2019 semester genap

Hasil pencarian solusi pada dataset periode 2018/2019 semester genap ditunjukkan pada Tabel 6.8. Solusi diperoleh dengan durasi sekitar 172 detik. Nilai *fitness* yang diperoleh untuk solusi pada dataset ini adalah sekitar 31%, yang berarti 31% dari 42 mahasiswa mendapatkan usulan dosen dengan bidang riset/bidang minat yang cocok dengan topik tesis yang diambil. Daftar pasangan mahasiswa dan dosen tersebut ditunjukkan pada Tabel 6.9.

Tabel 6. 8 Hasil pencarian solusi usulan dosen dataset periode 2018/2019 semester genap

| No | Nama Mahasiswa | Usulan 1 | Usulan 2 | Usulan 3 |
|----|---------------------------------------------------------------------------------|-------------------|-----------------|----------------|
| 1 | Abdul Aziz (Cloud Computing) | Danang Lelono | | |
| 2 | Abraham Koroh (Computer Vision) | Andi Dharmawan | Sigit Priyanta | |
| 3 | Akhmad Jayadi (Robotics) | Danang Lelono | Wahyono | |
| 4 | Andy Supriyadi (Big data analysis and visualization) | Sri Hartati | | |
| 5 | Arief Ardiansyah (Computer Vision) | Andi Dharmawan | Azhari | |
| 6 | Astrid Noviana Paradhita (Data models, databases, and information retrieval) | Faizal Makhrus | | |
| 7 | Canggih Gelar Setyo Adhi (Pattern Recognition) | Agus Harjoko | Danang Lelono | |
| 8 | Caraka Wedhatama (Big data analysis and visualization) | Anny Kartika Sari | Sri Hartati | |
| 9 | Danur Wijayanto (Cloud Computing) | Ahmad Ashari | Andi Dharmawan | |
| 10 | Deska Rizka Amalia (Enterprise systems) | Suprapto | | |
| 11 | Dimas Elang Setyoko (Computer Vision) | Aina Musdholifah | Yohanes Suyanto | |
| 12 | Dino Dwi Jayanto (Big data analysis and visualization) | Danang Lelono | Raden Sumiharto | |
| 13 | Ekki Rinaldi (Big data analysis and visualization) | Anny Kartika Sari | Suprapto | |
| 14 | Faisal Dharma Adhinata (Pattern Recognition) | Agus Harjoko | Faizal Makhrus | Khabib Mustofa |

| | | | | |
|----|----------------------------------------------------------------------------------|------------------------|------------------------|-----------------|
| 15 | Febi Elvira Messe (Intelligent Agent) | Ahmad Ashari | Nur Rokhman | |
| 16 | Fredy Aga Nugroho (Sains Komputasional) | Aina Musdholifah | Yohanes Suyanto | |
| 17 | I Nyoman Prayana Trisna (NLP) | Sigit Priyanta | | |
| 18 | Jaka Persada Sembiring (Robotics) | Afiahayati | Raden Sumiharto | Suprapto |
| 19 | Jarot Achid Alvian (Computer Vision) | Sigit Priyanta | | |
| 20 | Kurnia Febrianti (Sistem Pendukung Keputusan) | Faizal Makhrus | Yohanes Suyanto | |
| 21 | Luthfil Khairi (Big data analysis and visualization) | Aina Musdholifah | Wahyono | |
| 22 | M. Tahir (Computer Vision) | Tri Kuntoro Priyambodo | | |
| 23 | Ma'ruf Aminudin (Kecerdasan Komputasional) | Aina Musdholifah | Azhari | Yohanes Suyanto |
| 24 | muhammad anshar (Data models, databases, and information retrieval) | Faizal Makhrus | Tri Kuntoro Priyambodo | |
| 25 | Muhammad Auzan (Robotics) | Danang Lelono | Moh.Edi Wibowo | |
| 26 | Muhammad Fajar B (Intelligent Agent) | Agfianto Eko Putra | | |
| 27 | Muhammad Ikhsan (Computer Vision) | Agfianto Eko Putra | Reza M.I.Pulungan | Yunita Sari |
| 28 | Nicolaus Euclides Wahyu Nugroho (Kecerdasan Komputasional) | Aina Musdholifah | | |
| 29 | Nirwana Hendrastuty (Big data analysis and visualization) | Tri Kuntoro Priyambodo | | |
| 30 | Rahmad Mulya (Business process management and modeling) | Andi Dharmawan | | |
| 31 | Rian Junianto Suryo Prayoga (Kecerdasan Komputasional) | Aina Musdholifah | Azhari | Moh.Edi Wibowo |
| 32 | Riszki Wijayatun Pratiwi (Big data analysis and visualization) | Danang Lelono | | |
| 33 | Rory Florensa (Computer Vision) | Tri Kuntoro Priyambodo | | |
| 34 | Sarah Lintang Sariwening (Software models, architectures, and patterns) | Agus Sihabuddin | | |
| 35 | Saryanto (Computer Vision) | Agus Sihabuddin | Andi Dharmawan | |
| 36 | Shabrina (Sistem Pakar dan Penalaran Komputer) | Yunita Sari | | |
| 37 | Susilo Romadhon (Sistem Pendukung Keputusan) | Agus Sihabuddin | Ahmad Ashari | |

| | | | | |
|----|-------------------------------------------------------------------------|-------------------|----------------|--|
| 38 | Taufan Bagus Dwi Putra Aditama (Big data analysis and visualization) | Edi Winarko | Moh.Edi Wibowo | |
| 39 | Wendel Herman Selsily (Software models, architectures, and patterns) | Reza M.I.Pulungan | | |
| 40 | WIDHI SULISTYO (Big data analysis and visualization) | Agus Sihabuddin | Danang Lelono | |
| 41 | Wiranti kusuma hapsari (Sistem Pendukung Keputusan) | Sri Hartati | | |
| 42 | Zahra Khaerunnisa (Big data analysis and visualization) | Azhari | | |

Tabel 6. 9 Daftar pasangan mahasiswa dan dosen dengan bidang riset yang cocok dataset periode 2018/2019 semester genap

| No | Nama Mahasiswa | Nama Dosen | Topik Thesis/Bidang Riset |
|----|--------------------------------|------------------|---------------------------------------------------|
| 1 | Akhmad Jayadi | Danang Lelono | Robotics |
| 2 | Astrid Noviana Paradhita | Faizal Makhrus | Data models, databases, and information retrieval |
| 3 | Canggih Gelar Setyo Adhi | Agus Harjoko | Pattern Recognition |
| 4 | Danur Wijayanto | Ahmad Ashari | Cloud Computing |
| 5 | Faisal Dharma Adhinata | Agus Harjoko | Pattern Recognition |
| 6 | I Nyoman Prayana | Sigit Priyanta | NLP |
| 7 | Ma'ruf Aminudin | Aina Musdholifah | Kecerdasan Komputasional |
| 8 | muhammad anshar | Faizal Makhrus | Data models, databases, and information retrieval |
| 9 | Muhammad Auzan | Danang Lelono | Robotics |
| 10 | Nicolaus Euclides | Aina Musdholifah | Kecerdasan Komputasional |
| 11 | Rian Junianto Suryo Prayoga | Aina Musdholifah | Kecerdasan Komputasional |
| 12 | Taufan Bagus Dwi Putra Aditama | Edi Winarko | Big data analysis and visualization |
| 13 | Wiranti kusuma hapsari | Sri Hartati | Sistem Pendukung Keputusan |

6.2.2 Hasil pencarian solusi dataset periode 2019/2020 semester genap

Hasil pencarian solusi pada dataset periode 2019/2020 semester genap ditunjukkan pada Tabel 6.10. Solusi diperoleh dengan durasi sekitar 191 detik. Nilai *fitness* yang diperoleh untuk solusi pada dataset ini adalah sekitar 33%, yang berarti 33% dari 57 mahasiswa mendapatkan usulan dosen dengan bidang riset/bidang minat yang cocok dengan topik tesis yang diambil. Daftar pasangan mahasiswa dan dosen tersebut ditunjukkan pada Tabel 6.11.

Tabel 6. 10 Hasil pencarian solusi usulan dosen dataset periode 2019/2020 semester genap

| No. | Nama Mahasiswa | Usulan 1 | Usulan 2 | Usulan 3 |
|-----|-------------------------------------------------------------|----------------------|------------------------|------------------|
| 1 | Abdur Rasyid (Big data analysis and visualization) | Aina Musdholifah | Reza M.I.Pulungan | |
| 2 | Ahmad Muyassar Ibrahim (Sains Komputasional) | Yunita Sari | | |
| 3 | Ahmad Subhan Yazid (Pembelajaran Mesin) | Azhari | | |
| 4 | Ahmad Widardi (Image Processing) | Faizal Makhrus | Retantyo Wardoyo | |
| 5 | Anang Prasetyo (Big data analysis and visualization) | Moh.Edi Wibowo | Reza M.I.Pulungan | |
| 6 | Anggun Prasanti (NLP) | Reza M.I.Pulungan | Tri Kuntoro Priyambodo | |
| 7 | Antonius Adi Nugroho (Image Processing) | Yohanes Suyanto | | |
| 8 | Ardi Yusri Hilmi (Web, mobile, and ubiquitous computing) | Khabib Mustofa | | |
| 9 | Ari Wahyono (Big data analysis and visualization) | Reza M.I.Pulungan | | |
| 10 | Azis Kurniawan (Image Processing) | Khabib Mustofa | Raden Sumiharto | |
| 11 | Bayu Wibisana (Pembelajaran Mesin) | Andi Dharmawan | Sigit Priyanta | |
| 12 | Bryan Andi Gerrardo (Sains Komputasional) | Sigit Priyanta | Suprapto | |
| 13 | Budi Ihsan Daulay (Pembelajaran Mesin) | Aina Musdholifah | Andi Dharmawan | Retantyo Wardoyo |
| 14 | Chrisantonius (Enterprise systems) | Afiahayati | Andi Dharmawan | |
| 15 | Curie habiba (Robotics) | Azhari | Danang Lelono | |
| 16 | Deffa Rahadiyan (Intelligent Agent) | Mardhani Riasetiawan | | |
| 17 | Fachreza (Pembelajaran Mesin) | Anny Kartika Sari | | |
| 18 | Farchan hakim raswa (Computer Vision) | Danang Lelono | | |

| | | | | |
|----|---------------------------------------------------------------------------------------|----------------------|------------------------|-------------|
| 19 | Fathurrochman Habibie (Bioinformatika) | Agus Sihabuddin | Nur Rokhman | |
| 20 | Febry Dwi Putra (Signal Processing) | Afiahayati | Aina Musdholifah | |
| 21 | Firnanda Akmal Subarkah (Data models, databases, and information retrieval) | Ahmad Ashari | | |
| 22 | Hari Surrisyad (Pembelajaran Mesin) | Edi Winarko | | |
| 23 | HENDRI KURNIAWAN PRAKOSA (Data models, databases, and information retrieval) | Andi Dharmawan | Faizal Makhrus | Yunita Sari |
| 24 | Hutomo Khairinas (Software models, architectures, and patterns) | Wahyono | | |
| 25 | I Putu Budhi Darma Purwanta (Image Processing) | Ahmad Ashari | | |
| 26 | Inneke Cynthia Kusumawardhani (Sains Komputasional) | Mardhani Riasetiawan | Reza M.I.Pulungan | |
| 27 | Kadek Gemilang Santiyuda (NULL) | Agfianto Eko Putra | | |
| 28 | Krisnawan Hartanto (Data models, databases, and information retrieval) | Faizal Makhrus | Yohanes Suyanto | |
| 29 | Marvina Pamularsih (Big data analysis and visualization) | Faizal Makhrus | | |
| 30 | Muhammad Fatih Darmawan (Data models, databases, and information retrieval) | Faizal Makhrus | | |
| 31 | Muhammad Irfan Indriyan (Pembelajaran Mesin) | Faizal Makhrus | Sigit Priyanta | Suprapto |
| 32 | Muhammad Mubarok (Data models, databases, and information retrieval) | Agus Sihabuddin | Danang Lelono | |
| 33 | Muhammad Pajar Kharisma Putra (Kecerdasan Komputasional) | Aina Musdholifah | Nur Rokhman | |
| 34 | Muhammad Zha'farudin Pudya W (Pembelajaran Mesin) | Suprapto | | |
| 35 | Mukarramah Thuhury F. (NLP) | Agus Harjoko | Aina Musdholifah | Yunita Sari |
| 36 | Mukta Hikam (Data models, databases, and information retrieval) | Danang Lelono | Yohanes Suyanto | |
| 37 | Mulkiah (Image Processing) | Reza M.I.Pulungan | Tri Kuntoro Priyambodo | |
| 38 | Nadifa Sophia (NULL) | Andi Dharmawan | Danang Lelono | |
| 39 | Oky Antoro (Pembelajaran Mesin) | Agfianto Eko Putra | Anny Kartika Sari | Edi Winarko |
| 40 | Panji Bintoro (Computer Vision) | Reza M.I.Pulungan | Yohanes Suyanto | |

| | | | | |
|----|------------------------------------------------------------------------|----------------------|------------------------|-------------|
| 41 | Prabowo Wahyu Sudarno (Intelligent Environment) | Moh.Edi Wibowo | Nur Rokhman | |
| 42 | Prasinta Setyo Wati (Image Processing) | Agus Harjoko | Suprapto | |
| 43 | Rafika Reza Fitriana (Kecerdasan Komputasional) | Sri Hartati | Yunita Sari | |
| 44 | Rahmat Budiarsa (NULL) | Faizal Makhrus | Tri Kuntoro Priyambodo | |
| 45 | Reza Pahlevi Sofyan (Big data analysis and visualization) | Mardhani Riasetiawan | Reza M.I.Pulungan | |
| 46 | Sensa Gudya Sauma Syahra (NLP) | Mardhani Riasetiawan | Yunita Sari | |
| 47 | Septa Bagas Kara (Big data analysis and visualization) | Edi Winarko | | |
| 48 | Shaifudin Zuhdi (Metode Numerik) | Ahmad Ashari | Faizal Makhrus | |
| 49 | Sukrisman (Sistem Pendukung Keputusan) | Sri Hartati | | |
| 50 | Teguh puji widianto (NLP) | Agus Harjoko | Mardhani Riasetiawan | Yunita Sari |
| 51 | Wahyu Cahyo Utomo (Big data analysis and visualization) | Agfianto Eko Putra | | |
| 52 | Wanda Yusuf Alvian (Kecerdasan Komputasional) | Afiahayati | | |
| 53 | WILIA SATRIA (Big data analysis and visualization) | Agus Harjoko | Edi Winarko | |
| 54 | Winarto Saputro (Data models, databases, and information retrieval) | Andi Dharmawan | Mardhani Riasetiawan | |
| 55 | Yuliana (Sains Komputasional) | Yunita Sari | | |
| 56 | Yusril Ihza (Image Processing) | Anny Kartika Sari | Nur Rokhman | |
| 57 | Zandy Yudha Perwira (Intelligent Agent) | Suprapto | | |

Tabel 6. 11 Daftar pasangan mahasiswa dan dosen dengan bidang riset yang cocok dataset periode 2019/2020 semester genap

| No | Nama Mahasiswa | Nama Dosen | Topik Tesis/Bidang Riset |
|----|--------------------------|----------------|---------------------------------------------------|
| 1 | Ahmad Subhan Yazid | Azhari | Pembelajaran Mesin |
| 2 | Ardi Yusri Hilmi | Khabib Mustofa | Web, mobile, and ubiquitous computing |
| 3 | Curie habiba | Danang Lelono | Robotics |
| 4 | Hari Surrisyad | Edi Winarko | Pembelajaran Mesin |
| 5 | HENDRI KURNIAWAN PRAKOSA | Faizal Makhrus | Data models, databases, and information retrieval |

| | | | |
|----|-------------------------------|------------------|---------------------------------------------------|
| 6 | Krisnawan Hartanto | Faizal Makhrus | Data models, databases, and information retrieval |
| 7 | Muhammad Fatih Darmawan | Faizal Makhrus | Data models, databases, and information retrieval |
| 8 | Muhammad Mubarok | Agus Sihabuddin | Data models, databases, and information retrieval |
| 9 | Muhammad Pajar Kharisma Putra | Aina Musdholifah | Kecerdasan Komputasional |
| 10 | Mukarramah Thuhury F. | Yunita Sari | NLP |
| 11 | Oky Antoro | Edi Winarko | Pembelajaran Mesin |
| 12 | Prasinta Setyo Wati | Agus Harjoko | Image Processing |
| 13 | Rafika Reza Fitriana | Sri Hartati | Kecerdasan Komputasional |
| 14 | Sensa Gudya Sauma Syahra | Yunita Sari | NLP |
| 15 | Septa Bagas Kara | Edi Winarko | Big data analysis and visualization |
| 16 | Shaifudin Zuhdi | Faizal Makhrus | Metode Numerik |
| 17 | Sukrisman | Sri Hartati | Sistem Pendukung Keputusan |
| 18 | Teguh puji widianto | Yunita Sari | NLP |
| 19 | WILIA SATRIA | Edi Winarko | Big data analysis and visualization |

6.3 Evaluasi Kinerja Sistem

Dari solusi setiap dataset yang sudah didapatkan, akan dibandingkan dengan dataset hasil keputusan rapat dosen pembimbing dengan periode yang sesuai. Dataset hasil keputusan rapat akan diubah kedalam bentuk kromosom dan dihitung nilai *fitness*nya. Akan dilihat juga berapa persentase kecocokan antara solusi dengan dataset hasil rapat. Evaluasi lain yang harus diperhatikan adalah memastikan tidak ada dosen yang diusulkan lebih dari kuota tersedia.

6.3.1 Evaluasi solusi periode 2018/2019 semester genap

Hasil keputusan rapat dosen pembimbing pada periode 2018/2019 semester genap setelah direpresentasikan kedalam bentuk kromosom menghasilkan nilai *fitness* sekitar 14.3%. Solusi hasil algoritma genetika memperoleh sekitar 5% kecocokan dengan hasil keputusan rapat. Sampel dari hasil keputusan rapat penugasan dosen pembimbing periode 2018/2019 semester ditunjukkan pada Tabel

6.12. Hasil solusi algoritma genetika yang sama dengan hasil keputusan rapat ditunjukkan pada Tabel 6.13.

Tabel 6. 12 Sampel hasil keputusan rapat periode 2018/2019 semester genap

| No | Nama Mahasiswa | Dosen Pembimbing |
|-----|------------------|------------------------|
| 1 | Abdul Aziz | Tri Kuntoro Priyambodo |
| 2 | Abraham Koroh | Moh. Edi Wibowo |
| 3 | Akhmad Jayadi | Andi Dharmawan |
| 4 | Andy Supriyadi | Anny Kartika Sari |
| 5 | Arief Ardiansyah | Wahyono |
| ... | | |

Tabel 6. 13 Solusi algoritma genetika yang sama dengan hasil keputusan rapat periode 2018/2019 semester genap

| No | Nama Mahasiswa | Nama Dosen |
|----|-----------------------------|------------------|
| 1 | Rian Junianto Suryo Prayoga | Aina Musdholifah |
| 2 | Muhammad Auzan | Danang Lelono |

Setelah membandingkan nilai fitness solusi pada Tabel 6.8 dengan nilai fitness hasil rapat pada Tabel 6.12, solusi yang diperoleh menggunakan algoritma genetika lebih baik daripada hasil keputusan rapat, tetapi solusi algoritma genetika masih memiliki nilai kecocokan yang kecil (<10%) dengan hasil keputusan rapat.

Evaluasi lain yang dilakukan untuk solusi hasil algoritma genetika adalah memastikan bahwa tidak ada dosen yang mengalami kelebihan beban kuota pada solusi usulan dosen. Evaluasi untuk sisa kuota dosen dari solusi hasil algoritma genetika ditunjukkan pada Tabel 6.14.

Tabel 6. 14 Sisa kuota dosen solusi hasil algoritma genetika periode 2018/2019 semester genap

| No | Nama Dosen | Kuota Awal | Total Pengusul | Sisa Kuota (Kuota awal-total pengusul) |
|----|--------------------|------------|----------------|----------------------------------------|
| 1 | Afiahayati | 1 | 1 | 0 |
| 2 | Agfianto Eko Putra | 2 | 2 | 0 |

| | | | | |
|----|------------------------|---|---|---|
| 3 | Agus Harjoko | 2 | 2 | 0 |
| 4 | Agus Sihabuddin | 6 | 4 | 2 |
| 5 | Ahmad Ashari | 3 | 3 | 0 |
| 6 | Aina Musdholifah | 6 | 6 | 0 |
| 7 | Andi Dharmawan | 8 | 5 | 3 |
| 8 | Anny Kartika Sari | 3 | 2 | 1 |
| 9 | Azhari | 5 | 4 | 1 |
| 10 | Danang Lelono | 7 | 7 | 0 |
| 11 | Edi Winarko | 1 | 1 | 0 |
| 12 | Faizal Makhrus | 5 | 4 | 1 |
| 13 | Khabib Mustofa | 1 | 1 | 0 |
| 14 | Mardhani Riasetiawan | 0 | 0 | 0 |
| 15 | Moh.Edi Wibowo | 6 | 3 | 3 |
| 16 | Nur Rokhman | 1 | 1 | 0 |
| 17 | Raden Sumiharto | 2 | 2 | 0 |
| 18 | Reza M.I.Pulungan | 2 | 2 | 0 |
| 19 | Retantyo Wardoyo | 0 | 0 | 0 |
| 20 | Sigit Priyanta | 3 | 3 | 0 |
| 21 | Sri Hartati | 3 | 3 | 0 |
| 22 | Suprapto | 3 | 3 | 0 |
| 23 | Tri Kuntoro Priyambodo | 4 | 4 | 0 |
| 24 | Wahyono | 3 | 2 | 1 |
| 25 | Yohanes Suyanto | 6 | 4 | 2 |
| 26 | Yunita Sari | 2 | 2 | 0 |

Tabel 6.14 menunjukkan sudah tidak ada dosen yang mengalami kelebihan beban kuota pada solusi Tabel 6.8. Evaluasi sisa kuota dosen tersebut membuktikan

bahwa solusi yang diperoleh menggunakan algoritma genetika sudah menghasilkan solusi yang *feasible*, sehingga dapat dikatakan bahwa solusi Tabel 6.8 adalah solusi yang valid.

6.3.2 Evaluasi solusi periode 2019/2020 semester genap

Hasil keputusan rapat dosen pembimbing pada periode 2019/2020 semester genap setelah direpresentasikan kedalam bentuk kromosom menghasilkan nilai *fitness* sekitar 14%. Solusi hasil algoritma genetika memperoleh sekitar 7% kecocokan dengan hasil keputusan rapat. Sampel dari hasil keputusan rapat penugasan dosen pembimbing periode 2019/2020 semester ditunjukkan pada Tabel 6.15. Hasil solusi algoritma genetika yang sama dengan hasil keputusan rapat ditunjukkan pada Tabel 6.16.

Tabel 6. 15 Sampel hasil keputusan rapat periode 2019/2020 semester genap

| No | Nama Mahasiswa | Dosen Pembimbing |
|-----|------------------------|-----------------------|
| 1 | Abdur Rasyid | Agus Sihabuddin |
| 2 | Ahmad Muyassar Ibrahim | Mardhani Ria Setiawan |
| 3 | Ahmad Subhan Yazid | Edi Winarko |
| 4 | Ahmad Widardi | Wahyono |
| 5 | Anang Prasetyo | Edi Winarko |
| ... | ... | ... |

Tabel 6. 16 Solusi algoritma genetika yang sama dengan hasil keputusan rapat periode 2019/2020 semester genap

| No | Nama Mahasiswa | Nama Dosen |
|----|-------------------------------|----------------------|
| 1 | Septa Bagas Kara | Edi Winarko |
| 2 | Inneke Cynthia Kusumawardhani | Mardhani Riasetiawan |
| 3 | Yohanes Suyanto | Yohanes Suyanto |
| 4 | Yunita Sari | Yunita Sar |

Setelah membandingkan nilai *fitness* solusi pada Tabel 6.10 dengan nilai *fitness* hasil rapat pada Tabel 6.15, solusi yang diperoleh menggunakan algoritma

genetika lebih baik daripada hasil keputusan rapat, tetapi solusi algoritma genetika masih memiliki nilai kecocokan yang kecil (<10%) dengan hasil keputusan rapat.

Evaluasi lain yang dilakukan untuk solusi hasil algoritma genetika adalah memastikan bahwa tidak ada dosen yang mengalami kelebihan beban kuota pada solusi usulan dosen. Evaluasi untuk sisa kuota dosen dari solusi hasil algoritma genetika ditunjukkan pada Tabel 6.17.

Tabel 6. 17 Sisa kuota dosen solusi hasil algoritma genetika periode 2018/2019 semester genap

| No | Nama Dosen | Kuota awal | Total Pengusul | Sisa Kuota (Kuota awal-total pengusul) |
|----|----------------------|------------|----------------|----------------------------------------|
| 1 | Afiahayati | 3 | 3 | 0 |
| 2 | Agfianto Eko Putra | 3 | 3 | 0 |
| 3 | Agus Harjoko | 4 | 4 | 0 |
| 4 | Agus Sihabuddin | 3 | 2 | 1 |
| 5 | Ahmad Ashari | 3 | 3 | 0 |
| 6 | Aina Musdholifah | 5 | 5 | 0 |
| 7 | Andi Dharmawan | 7 | 6 | 1 |
| 8 | Anny Kartika Sari | 3 | 3 | 0 |
| 9 | Azhari | 2 | 2 | 0 |
| 10 | Danang Lelono | 5 | 5 | 0 |
| 11 | Edi Winarko | 4 | 4 | 0 |
| 12 | Faizal Makhrus | 8 | 8 | 0 |
| 13 | Khabib Mustofa | 2 | 2 | 0 |
| 14 | Mardhani Riasetiawan | 6 | 6 | 0 |
| 15 | Moh.Edi Wibowo | 3 | 2 | 1 |
| 16 | Nur Rokhman | 4 | 4 | 0 |
| 17 | Raden Sumiharto | 1 | 1 | 0 |

| | | | | |
|----|------------------------|---|---|---|
| 18 | Reza M.I.Pulungan | 8 | 8 | 0 |
| 19 | Retantyo Wardoyo | 2 | 2 | 0 |
| 20 | Sigit Priyanta | 3 | 3 | 0 |
| 21 | Sri Hartati | 2 | 2 | 0 |
| 22 | Suprapto | 5 | 5 | 0 |
| 23 | Tri Kuntoro Priyambodo | 3 | 3 | 0 |
| 24 | Wahyono | 1 | 1 | 0 |
| 25 | Yohanes Suyanto | 4 | 4 | 0 |
| 26 | Yunita Sari | 7 | 7 | 0 |

Tabel 6.17 menunjukkan sudah tidak ada dosen yang mengalami kelebihan beban kuota pada solusi Tabel 6.8. Evaluasi sisa kuota dosen tersebut membuktikan bahwa solusi yang diperoleh menggunakan algoritma genetika sudah menghasilkan solusi yang *feasible*, sehingga dapat dikatakan bahwa solusi Tabel 6.8 adalah solusi yang valid.

BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Solusi yang diperoleh menggunakan algoritma genetika sudah mampu menghasilkan alokasi dosen pembimbing dan mahasiswa yang lebih baik daripada hasil keputusan rapat penugasan dosen pembimbing.
2. Solusi dari sistem algoritma genetika masih belum mampu mencapai target *fitness* atau target kecocokan topik tesis mahasiswa dengan dosen pembimbing (target : 80%). Hal tersebut dikarenakan ketidakseimbangan jumlah antara topik tesis tertentu yang diusulkan mahasiswa dengan jumlah dosen yang tersedia dengan bidang minat/bidang riset yang cocok.
3. Perolehan solusi dengan algoritma genetika sudah mampu memberikan solusi alokasi dosen pembimbing tesis dalam durasi yang cepat (Kurang dari 5 menit).
4. Akurasi atau kecocokan hasil algoritma genetika dengan hasil keputusan rapat masih dibawah 10%.

7.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan pada penelitian selanjutnya:

1. Melakukan perbaharuan terhadap data bidang minat/bidang riset dosen. Perbaharuan data tersebut bertujuan untuk memastikan bahwa data yang ada adalah data terbaru dan lengkap sehingga dalam proses pencarian solusi dapat dicapai nilai *fitness* yang lebih baik.
2. Mempertimbangkan dosen pembimbing yang diusulkan mahasiswa pada masa pendaftaran tesis sebagai salah satu *constrain* dalam rancangan sistem algoritma genetika.

DAFTAR PUSTAKA

- Abido, M. A. and Elazouni, A. (2009) ‘Improved crossover and mutation operators for genetic- algorithm project scheduling’, 2009 IEEE Congress on Evolutionary Computation, CEC 2009, pp. 1865–1872. doi: 10.1109/CEC.2009.4983168.
- Anderson, C. a., Jones, K. F. and Ryan, J. (1991) ‘A two-dimensional genetic algorithm for the Ising problem’, Complex Systems, 5, pp. 327–333. Available at: <http://www.complex-systems.com/pdf/05-3-4.pdf>.
- Bajpai, P. and Kumar, M. (2010) ‘Genetic algorithm—an approach to solve global optimization problems’, Indian Journal of computer science and engineering, 1(3), pp. 199–206.
- Blum, C. and Roli, A. (2003) ‘Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison’, ACM Computing Surveys, 35(3), pp. 268–308. doi: 10.1145/937503.937505.
- Boussaïd, I., Lepagnot, J. and Siarry, P. (2013) ‘A survey on optimization metaheuristics’, Information Sciences, 237(February), pp. 82–117. doi: 10.1016/j.ins.2013.02.041.
- Chakraborty, B. and Chaudhuri, P. (2003) ‘On the use of genetic algorithm with elitism in robust and nonparametric multivariate analysis’, Austrian Journal of Statistics, 32(1), pp. 13–27. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.5252&rep=rep1&type=pdf>.
- Chen, P., Fu, Z. and Lim, A. (2002) ‘The yard allocation problem’, Proceedings of the National Conference on Artificial Intelligence, pp. 3–8.
- Chown, A. H., Cook, C. J. and Wilding, N. B. (2018) ‘A simulated annealing approach to the student-project allocation problem’, American Journal of Physics, 86(9), pp. 701–708. doi: 10.1119/1.5045331.
- Dean, J. S. (2008) ‘Staff scheduling by a genetic algorithm with a two-dimensional chromosome structure’, 7th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2008, pp. 1–15.
- Holland, J. H. (1975) ‘Genetic Algorithms-John H. Holland’, pp. 1–4. Available at: <http://www.econ.iastate.edu/tesfatsi/holland.GAIntro.htm>.
- Lin, C. M. and Gen, M. (2008) ‘Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm’, Expert Systems with Applications, 34(4), pp. 2480–2490. doi: 10.1016/j.eswa.2007.04.016.
- Mahmudy, W. F. (2006) ‘Penerapan algoritma genetika pada optimasi model penugasan’, Natural, 10 No.3(January 2006), pp. 197–207.
- S.N Cresswell, T. . Mcc. (2016) ‘A Hospital Placement Allocation Problem’, Technological Forecasting & Social Change, 104(December 2005), pp. 1–15.
- Salami, H. O. and Mamman, E. Y. (2016) ‘A genetic algorithm for allocating project supervisors to students’, International Journal of Intelligent Systems and Applications, 8(10), pp. 51–59. doi: 10.5815/ijisa.2016.10.06.
- Sivanandam, S. N. and Deepa, S. N. (2008) Introduction to genetic algorithms, Introduction to Genetic Algorithms. doi: 10.1007/978-3-540-73190-0.

- Smith, G. D. et al. (1998) 'Generational and Steady-State Genetic Algorithms for Generator Maintenance Scheduling Problems', *Artificial Neural Nets and Genetic Algorithms*, pp. 259–263. doi: 10.1007/978-3-7091-6492-1_57.
- Tsai, M. W., Hong, T. P. and Lin, W. T. (2015) 'A two-dimensional genetic algorithm and its application to aircraft scheduling problem', *Mathematical Problems in Engineering*, 2015. doi: 10.1155/2015/906305.
- Wang, X., Li, P. and Hawbani, A. (2018) 'An Efficient Budget Allocation Algorithm for Multi-Channel Advertising', *Proceedings - International Conference on Pattern Recognition*, 2018-Augus(August), pp. 886–891. doi: 10.1109/ICPR.2018.8545777.
- Abido, M. A. and Elazouni, A. (2009) 'Improved crossover and mutation operators for genetic- algorithm project scheduling', *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 1865–1872. doi: 10.1109/CEC.2009.4983168.
- Anderson, C. a., Jones, K. F. and Ryan, J. (1991) 'A two-dimensional genetic algorithm for the Ising problem', *Complex Systems*, 5, pp. 327–333. Available at: <http://www.complex-systems.com/pdf/05-3-4.pdf>.
- Bajpai, P. and Kumar, M. (2010) 'Genetic algorithm—an approach to solve global optimization problems', *Indian Journal of computer science and engineering*, 1(3), pp. 199–206.
- Blum, C. and Roli, A. (2003) 'Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison', *ACM Computing Surveys*, 35(3), pp. 268–308. doi: 10.1145/937503.937505.
- Boussaïd, I., Lepagnot, J. and Siarry, P. (2013) 'A survey on optimization metaheuristics', *Information Sciences*, 237(February), pp. 82–117. doi: 10.1016/j.ins.2013.02.041.
- Chakraborty, B. and Chaudhuri, P. (2003) 'On the use of genetic algorithm with elitism in robust and nonparametric multivariate analysis', *Austrian Journal of Statistics*, 32(1), pp. 13–27. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.90.5252&rep=rep1&type=pdf>.
- Chen, P., Fu, Z. and Lim, A. (2002) 'The yard allocation problem', *Proceedings of the National Conference on Artificial Intelligence*, pp. 3–8.
- Chown, A. H., Cook, C. J. and Wilding, N. B. (2018) 'A simulated annealing approach to the student-project allocation problem', *American Journal of Physics*, 86(9), pp. 701–708. doi: 10.1119/1.5045331.
- Dean, J. S. (2008) 'Staff scheduling by a genetic algorithm with a two-dimensional chromosome structure', *7th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2008*, pp. 1–15.
- Holland, J. H. (1975) 'Genetic Algorithms-John H. Holland', pp. 1–4. Available at: <http://www.econ.iastate.edu/tesfatsi/holland.GAIntro.htm>.
- Lin, C. M. and Gen, M. (2008) 'Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm', *Expert Systems with Applications*, 34(4), pp. 2480–2490. doi: 10.1016/j.eswa.2007.04.016.
- Mahmudy, W. F. (2006) 'Penerapan algoritma genetika pada optimasi model

- penugasan’, *Natural*, 10 No.3(January 2006), pp. 197–207.
- S.N Cresswell, T. . Mcc. (2016) ‘A Hospital Placement Allocation Problem’, *Technological Forecasting & Social Change*, 104(December 2005), pp. 1–15.
- Salami, H. O. and Mamman, E. Y. (2016) ‘A genetic algorithm for allocating project supervisors to students’, *International Journal of Intelligent Systems and Applications*, 8(10), pp. 51–59. doi: 10.5815/ijisa.2016.10.06.
- Sivanandam, S. N. and Deepa, S. N. (2008) *Introduction to genetic algorithms, Introduction to Genetic Algorithms*. doi: 10.1007/978-3-540-73190-0.
- Smith, G. D. et al. (1998) ‘Generational and Steady-State Genetic Algorithms for Generator Maintenance Scheduling Problems’, *Artificial Neural Nets and Genetic Algorithms*, pp. 259–263. doi: 10.1007/978-3-7091-6492-1_57.
- Tsai, M. W., Hong, T. P. and Lin, W. T. (2015) ‘A two-dimensional genetic algorithm and its application to aircraft scheduling problem’, *Mathematical Problems in Engineering*, 2015. doi: 10.1155/2015/906305.
- Wang, X., Li, P. and Hawbani, A. (2018) ‘An Efficient Budget Allocation Algorithm for Multi-Channel Advertising’, *Proceedings - International Conference on Pattern Recognition*, 2018-Augus(August), pp. 886–891. doi: 10.1109/ICPR.2018.8545777.

LAMPIRAN A

DAFTAR BIDANG MINAT DOSEN PROGRAM MAGISTER DIKE

| No. | Nama Dosen | Lab | Bidang Minat/Bidang Riset |
|-----|-------------------------------------------|-----------------------------|---------------------------------|
| 1 | Afiahayati, S.Kom, M.Cs., Ph.D | Lab Sistem Cerdas | Bioinformatics |
| | | | Machine Learning |
| 2 | Agfianto Eko Putra, M.Si., M.Kom., Dr | Anggota Lab SKJ | Embedded systems |
| | | | DSP |
| 3 | Agus Harjoko, Drs., Dr. | Lab ELINS | Computer Vision |
| | | | Pattern Recognition |
| | | | Instrumentation, Sensor Network |
| 4 | Agus Sihabuddin, S.Si., M.Kom., Dr. | Lab RPLD | Information System |
| | | | Distributed Data Base System |
| | | | Software Engineering |
| 6 | Ahmad Ashari, M.Ikom., Dr.-techn. | Lab SKJ | Distributed System |
| | | | Grid Computing |
| 7 | Aina Musdholifah, S.Kom., M.Kom., Ph.D | Lab Sistem Cerdas | Genetic Algorithm |
| | | | Fuzzy Logic |
| 8 | Andi Dharmawan, S.Si., M.Cs., Dr. | Lab Elins | UAV |
| | | | Control System |
| | | | Robotics |
| 9 | Anny Kartika Sari, S.Si., M.Sc., Ph.D | Lab Algoritma dan Komputasi | Discrete Structure |
| | | | Ontology |
| 10 | Azhari, Dr., MT. | Lab RPLD | Community Detection |
| | | | Question Answer System |
| | | | Machine Translation |
| 11 | Danang Lelono, S.Si, M.T., Dr. | Lab ELINS | Electronics |
| | | | Control |
| | | | E-Nose |
| 12 | Edi Winarko, Drs., M.Sc. Ph.D. | Lab RPLD | Data Mining |
| | | | Machine Learning |
| | | | big data |
| 13 | Faizal Makhrus,S.Kom., M.Sc., Ph.D | Lab Algoritma dan Komputasi | Simulation |
| | | | Numerical Method |
| | | | Distributed Database |
| 14 | Khabib Mustofa, S.Si., M.Kom., Dr.tech. | Lab RPLD | Semantic Web |
| 15 | Mardhani Riasetiawan, SE., Akt., MT., Dr. | Lab SKJ | Cloud Computing |

| | | | |
|----|------------------------------------------|-----------------------------|---------------------------------------------|
| 16 | Moh. Edi Wibowo,S.Kom., M.Kom, Ph.D. | Lab Algoritma dan Komputasi | Multimedia Analysis |
| 17 | Raden Sumiharto,S.Si.,M.Kom., Dr. | Lab ELINS | Instrumentasi |
| 18 | Reza M.I.Pulungan, S.Si., M.Sc., Dr.-Ing | Lab Algoritma dan Komputasi | Formal Method |
| | | | Stochastic Analysis |
| | | | Software Verification and Validation |
| 19 | Retantyo Wardoyo, Drs., M.Sc.,Ph.D. | Lab Algoritma dan Komputasi | Computation Theory |
| | | | Sains Manajemen |
| 20 | Sigit Priyanta, S.Si., M.Kom., Dr. | Lab RPLD | Text Mining |
| | | | Natural Language Computation |
| 21 | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Lab Sistem Cerdas | Artificial and Computational Intelligence |
| | | | Decision Support System |
| 22 | Suprapto, Drs., M.I.Kom. Dr. | Lab Algoritma dan Komputasi | Computational Logic |
| | | | Algorithm Analysis and Design |
| 23 | Tri Kuntoro Priyambodo, M.Sc.,Dr. | Lab SKJ | Satellite System Communication |
| | | | Precision Agriculture and real-time systems |
| 24 | Wahyono, S.Kom, Ph.D. | Lab Algoritma dan Komputasi | Image processing |
| | | | Graphics Computing |
| 25 | Yohanes Suyanto, M.Ikom., Dr. | Lab SKJ | Network Programming |
| 50 | Yunita Sari, S.Kom., M.Sc., Ph.D. | Lab Sistem Cerdas | Natural Language Processing |

LAMPIRAN B

TEMA BIDANG RISET DIKE PROGRAM MAGISTER

| No | Tema Riset |
|----|--------------------------------------------------------------------------|
| 1 | Lab AK - Algoritma |
| 2 | Lab AK - Approximation Algorithm |
| 3 | Lab AK - GPU & Parallel Computing |
| 4 | Lab AK - Grafika Komputer |
| 5 | Lab AK - Kompresi Data |
| 6 | Lab AK - Kriptologi |
| 7 | Lab AK - Metode Formal |
| 8 | Lab AK - Metode Numerik |
| 9 | Lab AK - Pemrograman Parallel |
| 10 | Lab AK - Pengolahan Citra |
| 11 | Lab AK - Sains Komputasional |
| 12 | Lab AK - Verifikasi dan Validasi |
| 13 | Lab Elins - Embedded system |
| 14 | Lab Elins - Image Processing |
| 15 | Lab Elins - Industrial Instrumentation |
| 16 | Lab Elins - Pattern Recognition |
| 17 | Lab Elins - Robotics |
| 18 | Lab Elins - Sensor Network, IOT |
| 19 | Lab Elins - Sensor Tranduser |
| 20 | Lab Elins - Signal Processing |
| 21 | Lab Elins - Telecommunication |
| 22 | Lab RPLD - Big data analysis and visualization |
| 23 | Lab RPLD - Business process management and modeling |
| 24 | Lab RPLD - Code analysis |
| 25 | Lab RPLD - Data models, databases, and information retrieval |
| 26 | Lab RPLD - e-Government |
| 27 | Lab RPLD - Enterprise systems |
| 28 | Lab RPLD - Gamification |
| 29 | Lab RPLD - Human computer interactions |
| 30 | Lab RPLD - IT Project Management |
| 31 | Lab RPLD - Software engineering processes, methods, practices, and tools |
| 32 | Lab RPLD - Software models, architectures, and patterns |
| 33 | Lab RPLD - software quality assurance |
| 34 | Lab RPLD - User interfaces and experiences |
| 35 | Lab RPLD - Virtual/augmented reality and immersive technologies |

| | |
|----|--------------------------------------------------|
| 36 | Lab RPLD - Web, mobile, and ubiquitous computing |
| 37 | Lab SC - AI |
| 38 | Lab SC - Bioinformatika |
| 39 | Lab SC - Computer Vision |
| 40 | Lab SC - Intelligent Agent |
| 41 | Lab SC - Kecerdasan Komputasional |
| 42 | Lab SC - Logika Fuzzy |
| 43 | Lab SC - NLP |
| 44 | Lab SC - Pembelajaran Mesin |
| 45 | Lab SC - Sistem Pakar dan Penalaran Komputer |
| 46 | Lab SC - Sistem Pendukung Keputusan |
| 47 | Lab SKJ - Big Data Architecture |
| 48 | Lab SKJ - Big Data Processing |
| 49 | Lab SKJ - Cloud Computing |
| 50 | Lab SKJ - Cyber Physical System |
| 51 | Lab SKJ - Cyber Security |
| 52 | Lab SKJ - Distributed Network & System |
| 53 | Lab SKJ - High Speed Network & Management |
| 54 | Lab SKJ - Infrastructure Data Center |
| 55 | Lab SKJ - Intelligent Environment |
| 56 | Lab SKJ - Operating System & Architecture |
| 57 | Lab SKJ - Security and Preservation |
| 58 | Lab SKJ - Smart City Infrastructures |

LAMPIRAN C

DATA DOSEN DAN KUOTA TERSEDIA PERIODE 2018/2019 SEMESTER GENAP

| No. | Nama Dosen | Kuota Tersedia |
|-----|-------------------------------------------|----------------|
| 1 | Afiahayati, S.Kom, M.Cs., Ph.D | 1 |
| 2 | Agfianto Eko Putra, M.Si., M.Kom., Dr | 2 |
| 3 | Agus Harjoko, Drs., Dr. | 2 |
| 4 | Agus Sihabuddin, S.Si., M.Kom., Dr. | 6 |
| 5 | Ahmad Ashari, M.Ikom., Dr.-techn. | 3 |
| 6 | Aina Musdholifah, S.Kom., M.Kom., Ph.D | 6 |
| 7 | Andi Dharmawan, S.Si., M.Cs., Dr. | 8 |
| 8 | Anny Kartika Sari, S.Si., M.Sc., Ph.D | 3 |
| 9 | Azhari, Dr., MT. | 5 |
| 10 | Danang Lelono, S.Si, M.T., Dr. | 7 |
| 11 | Edi Winarko, Drs., M.Sc.Ph.D. | 1 |
| 12 | Faizal Makhrus,S.Kom., M.Sc., Ph.D | 5 |
| 13 | Khabib Mustofa, S.Si., M.Kom., Dr.tech. | 1 |
| 14 | Mardhani Riasetiawan, SE., Akt., MT., Dr. | 0 |
| 15 | Moh.Edi Wibowo,S.Kom., M.Kom,Ph.D. | 6 |
| 16 | Nur Rokhman, S.Si., M.Kom., Dr. | 1 |
| 17 | Raden Sumiharto,S.Si.,M.Kom., Dr. | 2 |
| 18 | Reza M.I.Pulungan, S.Si., M.Sc., Dr.-Ing | 2 |
| 19 | Retantyo Wardoyo, Drs., M.Sc.,Ph.D. | 0 |
| 20 | Sigit Priyanta, S.Si., M.Kom., Dr. | 3 |
| 21 | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | 3 |
| 22 | Suprapto, Drs., M.I.Kom. Dr. | 3 |
| 23 | Tri Kuntoro Priyambodo, M.Sc.,Dr. | 4 |
| 24 | Wahyono, S.Kom,Ph.D. | 3 |
| 25 | Yohanes Suyanto, M.Ikom., Dr. | 6 |
| 26 | Yunita Sari, S.Kom., M.Sc.,Ph.D. | 2 |

LAMPIRAN D

TOPIK RISET PILIHAN MAHASISWA PERIODE 2018/2019 SEMESTER GENAP

| No | Nama Mahasiswa | Lab | Topik Tesis |
|----|---------------------------------|-------------------------|--------------------------------------------------|
| 1 | WIDHI SULISTYO | RPLD | Big Data Analysis and Visualization |
| 2 | Ekki Rinaldi | RPLD | Big Data Analysis and Visualization |
| 3 | M. TAHIR | Sistem Cerdas | Computer Vision |
| 4 | Ririn Septrisulviani | Algoritma dan Komputasi | Algoritma |
| 5 | Ma'ruf Aminudin | Sistem Cerdas | Kecerdasan Komputasional |
| 6 | Abdul Aziz | SKJ | Cloud Computing |
| 7 | I Nyoman Prayana Trisna | Sistem Cerdas | NLP |
| 8 | Sarah Lintang Sariwening | RPLD | Software Model, Architecture and Pattern |
| 9 | Astrid Noviana Paradhita | RPLD | Data Model, Databases, and Information Retrieval |
| 10 | FEBI ELVIRA MESSE | Sistem Cerdas | Intelligent Agent |
| 11 | Danur Wijayanto | SKJ | Cloud Computing |
| 12 | Canggih Gelar Setyo Adhi | ELINS | Pattern Recogniton |
| 13 | Faisal Dharma Adhinata | ELINS | Pattern Recogniton |
| 14 | Taufan Bagus Dwi Putra Aditama | RPLD | Big Data Analysis and Visualization |
| 15 | Shabrina | Sistem Cerdas | Sistem Pakar dan penalaran komputer |
| 16 | Luthfil Khairi | RPLD | Big Data Analysis and Visualization |
| 17 | Riszki Wijayatun Pratiwi | RPLD | Big Data Analysis and Visualization |
| 18 | Wendel Herman Selsily | RPLD | Web, Mobile, and upiquitous Computing |
| 19 | Nirwana Hendrasutty | RPLD | Big Data Analysis and Visualization |
| 20 | Muhammad Ikhsan | Sistem Cerdas | Computer Vision |
| 21 | Susilo Romadhon | Sistem Cerdas | Sistem Pendukung Keputusan |
| 22 | Nicolaus Euclides Wahyu Nugroho | Sistem Cerdas | Kecerdasan Komputasional |
| 23 | Muhammad Auzan | ELINS | Robotics |
| 24 | Rian Junianto Suryo Prayoga | Sistem Cerdas | Kecerdasan Komputasional |
| 25 | Dino Dwi Jayanto | RPLD | Big Data Analysis and Visualization |
| 26 | Deska Rizka Amalia | RPLD | Enterprise System |
| 27 | Kurnia Febrianti | Sistem Cerdas | Sistem Pendukung Keputusan |
| 28 | WIDHI SULISTYO | RPLD | Big Data Analysis and Visualization |
| 29 | SARYANTO | Sistem Cerdas | Computer Vision |

| | | | |
|----|------------------------|-------------------------|--------------------------------------------------|
| 30 | Stephani Dian Angelina | Sistem Cerdas | Pembelajaran Mesin |
| 31 | Akhmad Jayadi | ELINS | Robotics |
| 32 | Abraham Koroh | Sistem Cerdas | Computer Vision |
| 33 | Rory Florensa | Sistem Cerdas | Computer Vision |
| 34 | Rahmad Mulya | RPLD | Bussiness Proses Management and Modeling |
| 35 | Andy Supriyadi | RPLD | Big Data Analysis and Visualization |
| 36 | muhammad anshar | RPLD | Data Model, Databases, and Information Retrieval |
| 37 | wiranti kusuma hapsari | Sistem Cerdas | Sistem Pendukung Keputusan |
| 38 | Caraka Wedhatama | RPLD | Big Data Analysis and Visualization |
| 39 | Muhammad Fajar B | Sistem Cerdas | Intelligent Agent |
| 40 | Fajar Dwi Mawan | Algoritma dan Komputasi | Algoritma |
| 41 | Dimas Elang Setyoko | Sistem Cerdas | Computer Vision |
| 42 | Jaka Persada Sembiring | ELINS | Robotics |
| 43 | Jarot Achid Alvian | Sistem Cerdas | Computer Vision |
| 44 | Arief Ardiansyah | Sistem Cerdas | Computer Vision |
| 45 | Zahra Khaerunnisa | RPLD | Big Data Analysis and Visualization |
| 46 | Fredy Aga Nugroho | Algoritma dan Komputasi | Sains Komputational |

LAMPIRAN E

USULAN DOSEN PEMBIMBING TESIS PERIODE 2018/2019 SEMESTER GENAP

| No. | Nama Mahasiswa | Dosen Usulan 1* | Dosen Usulan 2* | Dosen Usulan 3* |
|-----|--------------------------------|---------------------------------------------|---------------------------------------------|---------------------------------------------|
| 1 | WIDHI SULISTYO | Nur Rokhman, S.Si., M.Kom., Dr. | Mardhani Riasetiawan, M.T., Dr. | Edi Winarko, Drs., M.Sc., Ph.D. |
| 2 | Ekki Rinaldi | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Wahyono, S.Kom., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. |
| 3 | M. TAHIR | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 4 | Ririn Septrisulviani | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing |
| 5 | Ma'ruf Aminudin | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 6 | Abdul Aziz | Mardhani Riasetiawan, M.T., Dr. | Tri Kuntoro Priyambodo, Drs., M.Sc., Dr. | Ahmad Ashari, Drs., M.I.Kom., Dr.Techn. |
| 7 | I Nyoman Prayana Trisna | Edi Winarko, Drs., M.Sc., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. |
| 8 | Sarah Lintang Sariwening | Edi Winarko, Drs., M.Sc., Ph.D. | Azhari, Drs., M.T., Dr. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 9 | Astrid Noviana Paradhita | Anny Kartika Sari, S.Si., M.Sc., Ph.D. | Khabib Mustofa, S.Si., M.Kom., Dr.techn. | Afiahayati, S.Kom., M.Cs., Ph.D. |
| 10 | FEBI ELVIRA MESSE | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 11 | Danur Wijayanto | Mardhani Riasetiawan, M.T., Dr. | Tri Kuntoro Priyambodo, Drs., M.Sc., Dr. | Ahmad Ashari, Drs., M.I.Kom., Dr.Techn. |
| 12 | Canggih Gelar Setyo Adhi | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 13 | Faisal Dharma Adhinata | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 14 | Taufan Bagus Dwi Putra Aditama | Edi Winarko, Drs., M.Sc., Ph.D. | Azhari, Drs., M.T., Dr. | Sigit Priyanta, S.Si., M.Kom., Dr. |
| 15 | Shabrina | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 16 | Luthfil Khairi | Edi Winarko, Drs., M.Sc., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. | Azhari, Drs., M.T., Dr. |
| 17 | Riszki Wijayatun Pratiwi | Edi Winarko, Drs., M.Sc., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. | Azhari, Drs., M.T., Dr. |
| 18 | Wendel Herman Selsily | Ahmad Ashari, Drs., M.I.Kom., Dr.Techn. | Ahmad Ashari, Drs., M.I.Kom., Dr.Techn. | Tri Kuntoro Priyambodo, Drs., M.Sc., Dr. |

| | | | | |
|----|---------------------------------|------------------------------------------|------------------------------------------|-----------------------------------------|
| 19 | Nirwana Hendrastuty | Azhari, Drs., M.T., Dr. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. |
| 20 | Muhammad Ikhsan | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 21 | Susilo Romadhon | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 22 | Nicolaus Euclides Wahyu Nugroho | Agus Harjoko, Drs., M.Sc., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 23 | Muhammad Auzan | Danang Lelono, S.Si., M.T., Dr. | Andi Dharmawan, S.Si., M.Kom., Dr. | Danang Lelono, S.Si., M.T., Dr. |
| 24 | Rian Junianto Suryo Prayoga | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. |
| 25 | Dino Dwi Jayanto | Sigit Priyanta, S.Si., M.Kom., Dr. | Edi Winarko, Drs., M.Sc., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 26 | Deska Rizka Amalia | Sigit Priyanta, S.Si., M.Kom., Dr. | Tri Kuntoro Priyambodo, Drs., M.Sc., Dr. | Sigit Priyanta, S.Si., M.Kom., Dr. |
| 27 | Kurnia Febrianti | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Azhari, Drs., M.T., Dr. |
| 28 | WIDHI SULISTYO | Nur Rokhman, S.Si., M.Kom., Dr. | Edi Winarko, Drs., M.Sc., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 29 | SARYANTO | Agus Harjoko, Drs., M.Sc., Ph.D. | Mardhani Riasetiawan, M.T., Dr. | Nur Rokhman, S.Si., M.Kom., Dr. |
| 30 | Stephani Dian Angelina | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 31 | Akhmad Jayadi | Andi Dharmawan, S.Si., M.Kom., Dr. | Danang Lelono, S.Si., M.T., Dr. | Danang Lelono, S.Si., M.T., Dr. |
| 32 | Abraham Koroh | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Azhari, Drs., M.T., Dr. |
| 33 | Rory Florensa | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. |
| 34 | Rahmad Mulya | Mardhani Riasetiawan, M.T., Dr. | Edi Winarko, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 35 | Andy Supriyadi | Mardhani Riasetiawan, M.T., Dr. | Nur Rokhman, S.Si., M.Kom., Dr. | Anny Kartika Sari, S.Si., M.Sc., Ph.D. |
| 36 | muhammad anshar | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 37 | wiranti kusuma hapsari | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 38 | Caraka Wedhatama | Khabib Mustofa, S.Si., M.Kom., Dr.techn. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 39 | Muhammad Fajar B | Azhari, Drs., M.T., Dr. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Azhari, Drs., M.T., Dr. |
| 40 | Fajar Dwi Mawan | Wahyono, S.Kom., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |

| | | | | |
|----|------------------------|----------------------------------------|----------------------------------------|---------------------------------------------|
| 41 | Dimas Elang Setyoko | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. |
| 42 | Jaka Persada Sembiring | Andi Dharmawan, S.Si., M.Kom., Dr. | Danang Lelono, S.Si., M.T., Dr. | Danang Lelono, S.Si., M.T., Dr. |
| 43 | Jarot Achid Alvian | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 44 | Arief Ardiansyah | Wahyono, S.Kom., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing |
| 45 | Zahra Khaerunnisa | Edi Winarko, Drs., M.Sc., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 46 | Fredy Aga Nugroho | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Agus Sihabudin, S.Si., M.Kom., Dr. | Mardhani Riasetiawan, M.T., Dr. |

*Catatan : Jika mahasiswa mengusulkan dosen yang sama untuk usulan 1, 2 atau 3 , maka mahasiswa dianggap hanya mengusulkan satu dosen

LAMPIRAN F

TOTAL PENGUSUL DOSEN PEMBIMBING TESIS PERIODE 2018/2019 SEMESTER GENAP

| No | Nama Dosen | Jumlah Usulan 1 | Jumlah Usulan 2 | Jumlah Usulan 3 | Total Pengusul |
|----|-------------------------------------------|-----------------|-----------------|-----------------|----------------|
| 1 | Afiahayati, S.Kom, M.Cs., Ph.D | 0 | 2 | 2 | 4 |
| 2 | Agfianto Eko Putra, M.Si., M.Kom., Dr | 0 | 0 | 0 | 0 |
| 3 | Agus Harjoko, Drs., Dr. | 5 | 5 | 1 | 11 |
| 4 | Agus Sihabuddin, S.Si., M.Kom., Dr. | 0 | 1 | 0 | 1 |
| 5 | Ahmad Ashari, M.Ikom., Dr.-techn. | 1 | 1 | 2 | 4 |
| 6 | Aina Musdholifah, S.Kom., M.Kom., Ph.D | 2 | 5 | 3 | 10 |
| 7 | Andi Dharmawan, S.Si., M.Cs., Dr. | 2 | 1 | 0 | 3 |
| 8 | Anny Kartika Sari, S.Si., M.Sc., Ph.D | 1 | 0 | 1 | 2 |
| 9 | Azhari, Dr., MT. | 2 | 2 | 4 | 8 |
| 10 | Danang Lelono, S.Si, M.T., Dr. | 1 | 2 | 0 | 3 |
| 11 | Edi Winarko, Drs., M.Sc.Ph.D. | 6 | 4 | 2 | 12 |
| 12 | Faizal Makhrus,S.Kom., M.Sc., Ph.D | 0 | 0 | 0 | 0 |
| 13 | Khabib Mustofa, S.Si., M.Kom., Dr.tech. | 1 | 1 | 0 | 2 |
| 14 | Mardhani Riasetiawan, SE., Akt., MT., Dr. | 4 | 1 | 4 | 9 |
| 15 | Moh.Edi Wibowo,S.Kom., M.Kom,Ph.D. | 3 | 4 | 5 | 12 |
| 16 | Nur Rokhman, S.Si., M.Kom., Dr. | 1 | 1 | 1 | 3 |
| 17 | Raden Sumiharto,S.Si.,M.Kom., Dr. | 0 | 0 | 0 | 0 |
| 18 | Reza M.I.Pulungan, S.Si., M.Sc., Dr.-Ing | 0 | 0 | 1 | 1 |
| 19 | Retantyo Wardoyo, Drs., M.Sc.,Ph.D. | 5 | 1 | 0 | 6 |
| 20 | Sigit Priyanta, S.Si., M.Kom., Dr. | 2 | 2 | 3 | 7 |
| 21 | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | 2 | 3 | 1 | 6 |
| 22 | Suprapto, Drs., M.I.Kom. Dr. | 0 | 0 | 0 | 0 |
| 23 | Tri Kuntoro Priyambodo, M.Sc.,Dr. | 0 | 3 | 1 | 4 |
| 24 | Wahyono, S.Kom,Ph.D. | 4 | 2 | 6 | 12 |
| 25 | Yohanes Suyanto, M.Ikom., Dr. | 0 | 0 | 0 | 0 |
| 26 | Yunita Sari, S.Kom., M.Sc.,Ph.D. | 0 | 0 | 0 | 0 |

LAMPIRAN G

DATA DOSEN DAN KUOTA TERSEDIA PERIODE 2019/2020 SEMESTER GENAP

| No | Nama Dosen | Kuota Tersedia |
|----|-------------------------------------------|----------------|
| 1 | Afiahayati, S.Kom, M.Cs., Ph.D | 3 |
| 2 | Agfianto Eko Putra, M.Si., M.Kom., Dr | 3 |
| 3 | Agus Harjoko, Drs., Dr. | 4 |
| 4 | Agus Sihabuddin, S.Si., M.Kom., Dr. | 3 |
| 5 | Ahmad Ashari, M.Ikom., Dr.-techn. | 3 |
| 6 | Aina Musdholifah, S.Kom., M.Kom., Ph.D | 5 |
| 7 | Andi Dharmawan, S.Si., M.Cs., Dr. | 7 |
| 8 | Anny Kartika Sari, S.Si., M.Sc., Ph.D | 3 |
| 9 | Azhari, Dr., MT. | 2 |
| 10 | Danang Lelono, S.Si, M.T., Dr. | 5 |
| 11 | Edi Winarko, Drs., M.Sc.Ph.D. | 4 |
| 12 | Faizal Makhrus,S.Kom., M.Sc., Ph.D | 8 |
| 13 | Khabib Mustofa, S.Si., M.Kom., Dr.tech. | 2 |
| 14 | Mardhani Riasetiawan, SE., Akt., MT., Dr. | 6 |
| 15 | Moh.Edi Wibowo,S.Kom., M.Kom,Ph.D. | 3 |
| 16 | Nur Rokhman, S.Si., M.Kom., Dr. | 4 |
| 17 | Raden Sumiharto,S.Si.,M.Kom., Dr. | 1 |
| 18 | Reza M.I.Pulungan, S.Si., M.Sc., Dr.-Ing | 8 |
| 19 | Retantyo Wardoyo, Drs., M.Sc.,Ph.D. | 2 |
| 20 | Sigit Priyanta, S.Si., M.Kom., Dr. | 3 |
| 21 | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | 2 |
| 22 | Suprapto, Drs., M.I.Kom. Dr. | 5 |
| 23 | Tri Kuntoro Priyambodo, M.Sc.,Dr. | 3 |
| 24 | Wahyono, S.Kom,Ph.D. | 1 |
| 25 | Yohanes Suyanto, M.Ikom., Dr. | 4 |
| 26 | Yunita Sari, S.Kom., M.Sc.,Ph.D. | 7 |

LAMPIRAN H

TOPIK RISET PILIHAN MAHASISWA PERIODE 2019/2020 SEMESTER GENAP

| No | Nama Mahasiswa | Lab | Topik Tesis |
|----|------------------------------------|-------------------------|--------------------------------------------------|
| 1 | FACHREZA | Sistem Cerdas | Pembelajaran Mesin |
| 2 | PANJI BINTORO | Sistem Cerdas | Computer Vision |
| 3 | Ardi Yusri Hilmi | RPLD | Web, Mobile, and upiquitous Computing |
| 4 | Wanda Yusuf Alvian | Sistem Cerdas | Komputasi Evolusioner |
| 5 | Yuliana | Algoritma dan Komputasi | Sains Komputational |
| 6 | chrisantonius | RPLD | Enterprise System |
| 7 | Winarto Saputro | RPLD | Data Model, Databases, and Information Retrieval |
| 8 | SEPTA BAGAS KARA | RPLD | Big Data Analysis and Visualization |
| 9 | teguh puji widianto | Sistem Cerdas | NLP |
| 10 | Prasinta Setyo Wati | ELINS | Image Processing |
| 11 | Febry Dwi Putra | ELINS | Signal Processing |
| 12 | Inneke Cynthia Kusumawardhani | Algoritma dan Komputasi | Sains Komputational |
| 13 | HENDRI KURNIAWAN PRAKOSA | RPLD | Data Model, Databases, and Information Retrieval |
| 14 | Sukrisman | Sistem Cerdas | Sistem Pendukung Keputusan |
| 15 | Muhammad Fatih Darmawan | RPLD | Data Model, Databases, and Information Retrieval |
| 16 | Sensa Gudya Sauma Syahra | Sistem Cerdas | NLP |
| 17 | curie habiba | ELINS | Robotics |
| 18 | Mukarramah Thuhury F. | Sistem Cerdas | NLP |
| 19 | Anggun Prasanti | Sistem Cerdas | NLP |
| 20 | Marvina Pamularsih | RPLD | Big Data Analysis and Visualization |
| 21 | Krisnawan Hartanto | RPLD | Data Model, Databases, and Information Retrieval |
| 22 | Muhammad Pajar Kharisma Putra | Sistem Cerdas | Komputasi Evolusioner |
| 23 | Fathurrochman Habibie | Sistem Cerdas | Bioinformatika |
| 24 | Yusril Ihza | ELINS | Image Processing |
| 25 | Bryan Andi Gerrardo | Algoritma dan Komputasi | Sains Komputational |
| 26 | farchan hakim raswa | Sistem Cerdas | Computer Vision |
| 27 | Ahmad Muyassar Ibrahim | Algoritma dan Komputasi | Sains Komputational |
| 28 | Anthonius Adi Nugroho | ELINS | Image Processing |
| 29 | Deffa Rahadiyan | Sistem Cerdas | Intelligent Agent |
| 30 | Hari Surrisyad | Sistem Cerdas | Pembelajaran Mesin |
| 31 | Ahmad Subhan Yazid | Sistem Cerdas | Pembelajaran Mesin |
| 32 | zandy.yudha.perwira@mail.ugm.ac.id | Sistem Cerdas | Intelligent Agent |

| | | | |
|----|------------------------------|-------------------------|--------------------------------------------------|
| 33 | Ari Wahyono | RPLD | Big Data Analysis and Visualization |
| 34 | Bayu Wibisana | Sistem Cerdas | Pembelajaran Mesin |
| 35 | Reza Pahlevi Sofyan | RPLD | Big Data Analysis and Visualization |
| 36 | Oky Antoro | Sistem Cerdas | Pembelajaran Mesin |
| 37 | Wahyu Cahyo Utomo | RPLD | Big Data Analysis and Visualization |
| 38 | Muhammad Mubarok | RPLD | Data Model, Databases, and Information Retrieval |
| 39 | Shaifudin Zuhdi | Algoritma dan Komputasi | Metode Numerik |
| 40 | Prabowo Wahyu Sudarno | SKJ | Intelligent Environment |
| 41 | WILIA SATRIA | RPLD | Big Data Analysis and Visualization |
| 42 | Hutomo Khairinas | RPLD | Software Model, Architecture and Pattern |
| 43 | Rafika Reza Fitriana | Sistem Cerdas | Komputasi Evolusioner |
| 44 | Budi Ihsan Daulay | Sistem Cerdas | Pembelajaran Mesin |
| 45 | Firnanda Akmal Subarkah | RPLD | Data Model, Databases, and Information Retrieval |
| 46 | Muhammad Zha'farudin Pudya W | Sistem Cerdas | Pembelajaran Mesin |
| 47 | Ahmad Widardi | ELINS | Image Processing |
| 48 | Mulkiah | ELINS | Image Processing |
| 49 | I Putu Budhi Darma Purwanta | ELINS | Image Processing |
| 50 | Abdur Rasyid | RPLD | Big Data Analysis and Visualization |
| 51 | Mukta Hikam | RPLD | Data Model, Databases, and Information Retrieval |
| 52 | Muhammad Irfan Indriyan | Sistem Cerdas | Pembelajaran Mesin |
| 53 | Azis Kurniawan | ELINS | Image Processing |
| 54 | Anang Prasetyo | RPLD | Big Data Analysis and Visualization |

LAMPIRAN I

USULAN DOSEN PEMBIMBING TESIS PERIODE 2019/2020 SEMESTER GENAP

| No | Nama Mahasiswa | Dosen Usulan 1* | Dosen Usulan 2* | Dosen Usulan 3* |
|----|-------------------------------|---------------------------------------------|---------------------------------------------|------------------------------------------|
| 1 | FACHREZA | Afiahayati, S.Kom., M.Cs., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 2 | PANJI BINTORO | Agus Harjoko, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 3 | Ardi Yusri Hilmi | Edi Winarko, Drs., M.Sc., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. |
| 4 | Wanda Yusuf Alvian | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing | Wahyono, S.Kom., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 5 | Yuliana | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. |
| 6 | chrisantonius | Edi Winarko, Drs., M.Sc., Ph.D. | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing | Afiahayati, S.Kom., M.Cs., Ph.D. |
| 7 | Winarto Saputro | Edi Winarko, Drs., M.Sc., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. | Yunita Sari, S.Kom., M.Sc., Ph.D. |
| 8 | SEPTA BAGAS KARA | Edi Winarko, Drs., M.Sc., Ph.D. | Yunita Sari, S.Kom., M.Sc., Ph.D. | Anny Kartika Sari, S.Si., M.Sc., Ph.D. |
| 9 | teguh puji widianto | Agus Sihabudin, S.Si., M.Kom., Dr. | Yunita Sari, S.Kom., M.Sc., Ph.D. | Ahmad Ashari, Drs., M.I.Kom., Dr.Techn. |
| 10 | Prasinta Setyo Wati | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. |
| 11 | Febry Dwi Putra | Anny Kartika Sari, S.Si., M.Sc., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Tri Kuntoro Priyambodo, Drs., M.Sc., Dr. |
| 12 | Inneke Cynthia Kusumawardhani | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. |
| 13 | HENDRI KURNIAWAN PRAKOSA | Nur Rokhman, S.Si., M.Kom., Dr. | Edi Winarko, Drs., M.Sc., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. |
| 14 | Sukrisman | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Retantyo Wardoyo, Drs., M.Sc., Ph.D. |
| 15 | Muhammad Fatih Darmawan | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. |
| 16 | Sensa Gudya Sauma Syahra | Mardhani Riasetiawan, M.T., Dr. | Yunita Sari, S.Kom., M.Sc., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. |

| | | | | |
|----|------------------------------------|----------------------------------------|-----------------------------------------|------------------------------------------|
| 17 | curie habiba | Andi Dharmawan, S.Si., M.Kom., Dr. | Danang Lelono, S.Si., M.T., Dr. | Agfianto Eko Putro, M.Si., Dr. |
| 18 | Mukarramah Thuhury F. | Edi Winarko, Drs., M.Sc., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. | Yunita Sari, S.Kom., M.Sc., Ph.D. |
| 19 | Anggun Prasanti | Edi Winarko, Drs., M.Sc., Ph.D. | Yunita Sari, S.Kom., M.Sc., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 20 | Marvina Pamularsih | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. |
| 21 | Krisnawan Hartanto | Afiahayati, S.Kom., M.Cs., Ph.D. | Yunita Sari, S.Kom., M.Sc., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 22 | Muhammad Pajar Kharisma Putra | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 23 | Fathurrochman Habibie | Afiahayati, S.Kom., M.Cs., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Yunita Sari, S.Kom., M.Sc., Ph.D. |
| 24 | Yusril Ihza | Wahyono, S.Kom., Ph.D. | R. Sumiharto, S.Si., M.Kom., Dr. | Agus Harjoko, Drs., M.Sc., Ph.D. |
| 25 | Bryan Andi Gerrardo | Afiahayati, S.Kom., M.Cs., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 26 | farchan hakim raswa | Agus Harjoko, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. | Tri Kuntoro Priyambodo, Drs., M.Sc., Dr. |
| 27 | Ahmad Muyassar Ibrahim | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. |
| 28 | Anthonius Adi Nugroho | R. Sumiharto, S.Si., M.Kom., Dr. | R. Sumiharto, S.Si., M.Kom., Dr. | R. Sumiharto, S.Si., M.Kom., Dr. |
| 29 | Deffa Rahadiyan | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. |
| 30 | Hari Surrisyad | Wahyono, S.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Edi Winarko, Drs., M.Sc., Ph.D. |
| 31 | Ahmad Subhan Yazid | Edi Winarko, Drs., M.Sc., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. | Yunita Sari, S.Kom., M.Sc., Ph.D. |
| 32 | zandy.yudha.perwira@mail.ugm.ac.id | Danang Lelono, S.Si., M.T., Dr. | Andi Dharmawan, S.Si., M.Kom., Dr. | Agfianto Eko Putro, M.Si., Dr. |
| 33 | Ari Wahyono | Agus Sihabudin, S.Si., M.Kom., Dr. | Afiahayati, S.Kom., M.Cs., Ph.D. | Nur Rokhman, S.Si., M.Kom., Dr. |
| 34 | Bayu Wibisana | Afiahayati, S.Kom., M.Cs., Ph.D. | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 35 | Reza Pahlevi Sofyan | Agus Sihabudin, S.Si., M.Kom., Dr. | Yunita Sari, S.Kom., M.Sc., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. |
| 36 | Oky Antoro | Edi Winarko, Drs., M.Sc., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Afiahayati, S.Kom., M.Cs., Ph.D. |
| 37 | Wahyu Cahyo Utomo | Agus Sihabudin, S.Si., M.Kom., Dr. | Afiahayati, S.Kom., M.Cs., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. |
| 38 | Muhammad Mubarok | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Afiahayati, S.Kom., M.Cs., Ph.D. | Mardhani Riasetiawan, M.T., Dr. |
| 39 | Shaifudin Zuhdi | Suprapto, Drs., M.I.Kom., Dr. | Sigit Priyanta, S.Si., M.Kom., Dr. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |

| | | | | |
|----|------------------------------|-----------------------------------------|-----------------------------------------|---------------------------------------------|
| 40 | Prabowo Wahyu Sudarno | Ahmad Ashari, Drs., M.I.Kom., Dr.Techn. | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. |
| 41 | WILIA SATRIA | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. | Mardhani Riasetiawan, M.T., Dr. |
| 42 | Hutomo Khairinas | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | Agus Sihabudin, S.Si., M.Kom., Dr. | Nur Rokhman, S.Si., M.Kom., Dr. |
| 43 | Rafika Reza Fitriana | Afiahayati, S.Kom., M.Cs., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Sri Hartati, Dra., M.Sc., Ph.D., Prof. |
| 44 | Budi Ihsan Daulay | Afiahayati, S.Kom., M.Cs., Ph.D. | Retantyo Wardoyo, Drs., M.Sc., Ph.D. | MHD. Reza MI Pulungan, S.Si., M.Sc., Dr.Ing |
| 45 | Firnanda Akmal Subarkah | Sigit Priyanta, S.Si., M.Kom., Dr. | Edi Winarko, Drs., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 46 | Muhammad Zha'farudin Pudya W | Afiahayati, S.Kom., M.Cs., Ph.D. | Wahyono, S.Kom., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 47 | Ahmad Widardi | Wahyono, S.Kom., Ph.D. | Faizal Makhrus, S.Kom., M.Sc., Ph.D. | Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D. |
| 48 | Mulkiah | Anny Kartika Sari, S.Si., M.Sc., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. |
| 49 | I Putu Budhi Darma Purwanta | Edi Winarko, Drs., M.Sc., Ph.D. | Agus Harjoko, Drs., M.Sc., Ph.D. | Wahyono, S.Kom., Ph.D. |
| 50 | Abdur Rasyid | Agus Sihabudin, S.Si., M.Kom., Dr. | Afiahayati, S.Kom., M.Cs., Ph.D. | Nur Rokhman, S.Si., M.Kom., Dr. |
| 51 | Mukta Hikam | Edi Winarko, Drs., M.Sc., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. | Yunita Sari, S.Kom., M.Sc., Ph.D. |
| 52 | Muhammad Irfan Indriyan | Afiahayati, S.Kom., M.Cs., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. | Sigit Priyanta, S.Si., M.Kom., Dr. |
| 53 | Azis Kurniawan | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | Afiahayati, S.Kom., M.Cs., Ph.D. | Aina Musdholifah, S.Kom., M.Kom., Ph.D. |
| 54 | Anang Prasetyo | Edi Winarko, Drs., M.Sc., Ph.D. | Anny Kartika Sari, S.Si., M.Sc., Ph.D. | Suprapto, Drs., M.I.Kom., Dr. |

*Catatan : Jika mahasiswa mengusulkan dosen yang sama untuk usulan 1, 2 atau 3 , maka mahasiswa dianggap hanya mengusulkan satu dosen.

LAMPIRAN J
TOTAL PENGUSUL DOSEN PEMBIMBING TESIS
PERIODE 2019/2020 SEMESTER GENAP

| No | Nama Dosen | Jumlah Usulan 1 | Jumlah Usulan 2 | Jumlah Usulan 3 | Total Pengusul |
|----|-------------------------------------------|-----------------|-----------------|-----------------|----------------|
| 1 | Afiahayati, S.Kom, M.Cs., Ph.D | 9 | 6 | 4 | 19 |
| 2 | Agfianto Eko Putra, M.Si., M.Kom., Dr | 0 | 0 | 2 | 2 |
| 3 | Agus Harjoko, Drs., Dr. | 2 | 3 | 3 | 8 |
| 4 | Agus Sihabuddin, S.Si., M.Kom., Dr. | 5 | 1 | 0 | 6 |
| 5 | Ahmad Ashari, M.Ikom., Dr.-techn. | 1 | 0 | 1 | 2 |
| 6 | Aina Musdholifah, S.Kom., M.Kom., Ph.D | 0 | 5 | 4 | 9 |
| 7 | Andi Dharmawan, S.Si., M.Cs., Dr. | 1 | 1 | 0 | 2 |
| 8 | Anny Kartika Sari, S.Si., M.Sc., Ph.D | 2 | 1 | 1 | 4 |
| 9 | Azhari, Dr., MT. | 0 | 1 | 0 | 1 |
| 10 | Danang Lelono, S.Si, M.T., Dr. | 1 | 0 | 0 | 1 |
| 11 | Edi Winarko, Drs., M.Sc.Ph.D. | 11 | 4 | 2 | 17 |
| 12 | Faizal Makhrus,S.Kom., M.Sc., Ph.D | 0 | 1 | 0 | 1 |
| 13 | Khabib Mustofa, S.Si., M.Kom., Dr.tech. | 0 | 0 | 0 | 0 |
| 14 | Mardhani Riasetiawan, SE., Akt., MT., Dr. | 7 | 0 | 4 | 11 |
| 15 | Moh.Edi Wibowo,S.Kom., M.Kom,Ph.D. | 1 | 0 | 5 | 6 |
| 16 | Nur Rokhman, S.Si., M.Kom., Dr. | 1 | 0 | 3 | 4 |
| 17 | Raden Sumiharto,S.Si.,M.Kom., Dr. | 1 | 1 | 0 | 2 |
| 18 | Reza M.I.Pulungan, S.Si., M.Sc., Dr.-Ing | 1 | 1 | 1 | 3 |
| 19 | Retantyo Wardoyo, Drs., M.Sc.,Ph.D. | 2 | 3 | 0 | 5 |
| 20 | Sigit Priyanta, S.Si., M.Kom., Dr. | 1 | 4 | 1 | 6 |
| 21 | Sri Hartati, Dra., M.Sc., Ph.D., Prof. | 3 | 1 | 3 | 7 |
| 22 | Suprapto, Drs., M.I.Kom. Dr. | 1 | 0 | 1 | 2 |
| 23 | Tri Kuntoro Priyambodo, M.Sc.,Dr. | 0 | 0 | 2 | 2 |
| 24 | Wahyono, S.Kom,Ph.D. | 4 | 5 | 2 | 11 |
| 25 | Yohanes Suyanto, M.Ikom., Dr. | 0 | 0 | 0 | 0 |
| 26 | Yunita Sari, S.Kom., M.Sc.,Ph.D. | 0 | 6 | 5 | 11 |

LAMPIRAN K

HASIL RAPAT PENUGASAN DOSEN PEMBIMBING
PERIODE 2018/2019 SEMESTER GENAP

| No | Nama Mahasiswa | Dosen Pembimbing |
|----|------------------------------------|------------------------|
| 1 | Abdul Aziz | Tri Kuntoro Priyambodo |
| 2 | Abraham Koroh | Moh. Edi Wibowo |
| 3 | Akhmad Jayadi | Andi Dharmawan |
| 4 | Andy Supriyadi | Anny Kartika Sari |
| 5 | Arief Ardiansyah | Wahyono |
| 6 | Astrid Noviana Paradhita | Anny Kartika Sari |
| 7 | Canggih Gelar Setyo Adhi | Wahyono |
| 8 | Caraka Wedhatama | Khabib Mustofa |
| 9 | Danur Wijayanto | Tri Kuntoro Priyambodo |
| 10 | Deska Rizka Amalia | Sigit Priyanta |
| 11 | Dimas Elang Setyoko | Moh. Edi Wibowo |
| 12 | Dino Dwi Jayanto | Sigit Priyanta |
| 13 | Ekki Rinaldi | Aina Musdholifah |
| 14 | Faisal Dharma Adhinata | Wahyono |
| 15 | Febi Elvira Messe | Sri Hartati |
| 16 | Fredy Aga Nugroho | Agus Sihabuddin |
| 17 | I Nyoman Prayana Trisna | Aina Musdholifah |
| 18 | Jaka Persada Sembiring | Andi Dharmawan |
| 19 | Jarot Achid Alvian | Moh. Edi Wibowo |
| 20 | Kurnia Febrianti | Sri Hartati |
| 21 | Luthfil Khairi | Edi Winarko |
| 22 | M. Tahir | R. Sumiharto |
| 23 | Ma'ruf Aminudin | Moh. Edi Wibowo |
| 24 | muhammad anshar | Moh. Edi Wibowo |
| 25 | Muhammad Auzan | Danang Lelono |
| 26 | Muhammad Fajar B | Azhari |
| 27 | Muhammad Ikhsan | Wahyono |
| 28 | Nicolaus Euclides Wahyu Nugroho | Agus Harjoko |
| 29 | Nirwana Hendrastuty | Azhari |
| 30 | Rahmad Mulya | Agus Sihabuddin |
| 31 | Rian Junianto Suryo Prayoga | Aina Musdholifah |

| | | |
|----|--------------------------------|------------------|
| 32 | Riszki Wijayatun Pratiwi | Yunita Sari |
| 33 | Rory Florensa | Y. Suyanto |
| 34 | Sarah Lintang Sariwening | Azhari |
| 35 | Saryanto | Agus Harjoko |
| 36 | Shabrina | Sri Hartati |
| 37 | Susilo Romadhon | Retantyo Wardoyo |
| 38 | Taufan Bagus Dwi Putra Aditama | Azhari |
| 39 | Wendel Herman Selsily | Ahmad Ashari |
| 40 | WIDHI SULISTYO | Nur Rokhman |
| 41 | Wiranti kusuma hapsari | Aina Musdholifah |
| 42 | Zahra Khaerunnisa | Yunita Sari |

LAMPIRAN L

HASIL RAPAT PENUGASAN DOSEN PEMBIMBING
PERIODE 2019/2020 PERIODE GENAP

| No | Nama Mahasiswa | Dosen Pembimbing |
|----|-------------------------------|---------------------------|
| 1 | Abdur Rasyid | Agus Sihabuddin |
| 2 | Ahmad Muyassar Ibrahim | Mardhani Ria Setiawan |
| 3 | Ahmad Subhan Yazid | Edi Winarko |
| 4 | Ahmad Widardi | Wahyono |
| 5 | Anang Prasetyo | Edi Winarko |
| 6 | Anggun Prasanti | Edi Winarko |
| 7 | Anthonius Adi Nugroho | R. Sumiharto |
| 8 | Ardi Yusri Hilmi | Edi Winarko |
| 9 | Ari Wahyono | Agus Sihabuddin |
| 10 | Azis Kurniawan | Moh. Edi Wibowo |
| 11 | Bayu Wibisana | Aina Musdholifah |
| 12 | Bryan Andi Gerrardo | Azhari |
| 13 | Budi Ihsan Daulay | MHD. Reza MI. Pulungan |
| 14 | Chrisantonius | Tri Kuntoro Priyambodo |
| 15 | Curie habiba | Andi Dharmawan |
| 16 | Deffa Rahadiyan | Sri Hartati |
| 17 | Fachreza | Afiahayati |
| 18 | Farchan hakim raswa | Agus Harjoko |
| 19 | Fathurrochman Habibie | Afiahayati |
| 20 | Febry Dwi Putra | Tri Kuntoro Priyambodo |
| 21 | Firnanda Akmal Subarkah | Sigit Priyanta |
| 22 | Hari Surrisyad | Wahyono |
| 23 | HENDRI KURNIAWAN PRAKOSA | Nur Rokhman |
| 24 | Hutomo Khairinas | Tri Kuntoro Priyambodo |
| 25 | I Putu Budhi Darma Purwanta | Moh. Edi Wibowo |
| 26 | Inneke Cynthia Kusumawardhani | Mardhani Ria Setiawan |
| 27 | Kadek Gemilang Santiyuda | Retantyo Wardoyo |
| 28 | Krisnawan Hartanto | Suprapto |
| 29 | Marvina Pamularsih | Mardhani Ria Setiawan |
| 30 | Muhammad Fatih Darmawan | Mardhani Ria Setiawan |
| 31 | Muhammad Irfan Indriyan | Aina Musdholifah |

| | | |
|----|-------------------------------|---------------------------|
| 32 | Muhammad Mubarok | Nur Rokhman |
| 33 | Muhammad Pajar Kharisma Putra | Wahyono |
| 34 | Muhammad Zha'farudin Pudya W | Moh. Edi Wibowo |
| 35 | Mukarramah Thuhury F. | Edi Winarko |
| 36 | Mukta Hikam | Y. Suyanto |
| 37 | Mulkiah | Suprapto |
| 38 | Nadifa Sophia | Edi Winarko |
| 39 | Oky Antoro | Aina Musdholifah |
| 40 | Panji Bintoro | Agus Harjoko |
| 41 | Prabowo Wahyu Sudarno | Ahmad Ashari |
| 42 | Prasinta Setyo Wati | Moh. Edi Wibowo |
| 43 | Rafika Reza Fitriana | Afiahayati |
| 44 | Rahmat Budiarsa | Retantyo Wardoyo |
| 45 | Reza Pahlevi Sofyan | Agus Sihabuddin |
| 46 | Sensa Gudya Sauma Syahra | Yunita Sari |
| 47 | Septa Bagas Kara | Edi Winarko |
| 48 | Shaifudin Zuhdi | Suprapto |
| 49 | Sukrisman | Retantyo Wardoyo |
| 50 | Teguh puji widianto | Agus Sihabuddin |
| 51 | Wahyu Cahyo Utomo | MHD. Reza MI. Pulungan |
| 52 | Wanda Yusuf Alvian | MHD. Reza MI. Pulungan |
| 53 | WILIA SATRIA | Mardhani Ria Setiawan |
| 54 | Winarto Saputro | Ahmad Ashari |
| 55 | Yuliana | Mardhani Ria Setiawan |
| 56 | Yusril Ihza | Danang Lelono |
| 57 | Zandy Yudha Perwira | Danang Lelono |

LAMPIRAN M

KROMOSOM SOLUSI PERIODE 2018/2019 SEMESTER GENAP

Result:

LAMPIRAN N

Result:

LAMPIRAN O

PENGODEAN KROMOSOM HASIL KEPUTUSAN RAPAT PERIODE 2018/2019 SEMESTER GENAP

LAMPIRAN P

PENGODEAN KROMOSOM HASIL KEPUTUSAN RAPAT PERIODE 2019/2020 SEMESTER GENAP