

Algoritma *Grouping Genetic* berdasarkan Kepangkatan dan Grup Riset untuk Penjadwalan Ujian Skripsi

Genetic Grouping Algorithm based on Rank and Research Group for Timetabling Thesis Examination

Habibie Ed Dien*, M. Hasyim Ratsanjani, Andhika Satrio Wiratama, Vit Zuraida
Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
Kota Malang, Indonesia
*e-mail: habibie@polinema.ac.id

(received: 2 Maret 2023, revised: 4 Mei 2023, accepted: 13 Mei 2023)

Abstrak

Penjadwalan adalah masalah umum yang dihadapi oleh berbagai institusi akademik, terutama dalam proses penjadwalan ujian skripsi. Pada proses penjadwalan skripsi ada beberapa masalah yang muncul yaitu masalah penentuan penguji dan urutannya, mengatur slot ruang dan waktu, yang membuat proses penjadwalan tidak efisien. Masalah tersebut akan diselesaikan dengan algoritma *grouping genetic* (GGA) yang dikombinasikan dengan parameter Daftar Urut Kepangkatan (DUK) dan grup riset (GR) dosen, yang terbukti efisien digunakan untuk menyelesaikan masalah seperti pada proses penjadwalan ujian skripsi. Hasil penelitian menunjukkan bahwa GGA dengan DUK dan GR berhasil meningkatkan efisiensi sebesar 99,97% apabila diterapkan untuk menyelesaikan masalah tersebut pada proses penjadwalan ujian skripsi.

Kata kunci: penjadwalan skripsi, *grouping genetic*, kepangkatan, grup riset.

Abstract

Timetabling is a common problem faced by various academic institutions, especially in the process of timetabling thesis examination. In the process of timetabling thesis examination there are several problems that arise, namely the problem of determining examiners and their order, arranging space and time slots, which makes the timetabling process inefficient. This problem will be solved by using a genetic grouping algorithm (GGA) combined with the parameters of rank and research groups (RG), which have proven to be efficient to use to solve problems such as in the thesis exam timetabling process. The results showed that GGA with ran and RG succeeded in increasing efficiency by 99,97% when applied to solving this problem in the thesis exam timetabling process..

Keywords: thesis timetabling, *grouping genetic*, rank, research groups.

1 Pendahuluan

Penjadwalan merupakan permasalahan umum yang dihadapi oleh berbagai institusi akademik seperti sekolah, perguruan tinggi atau universitas [1], [2]. Permasalahannya selalu dihadapkan dengan kombinasi yang sangat kompleks dengan permasalahan optimasi yang cukup menarik perhatian para peneliti [2], [3]. Salah satu permasalahan yang dihadapi dalam menyusun jadwal seminar proposal dan sidang skripsi adalah selain jumlah mahasiswa, penguji, waktu, ruang dan judul, juga adanya *hard constraint* (HC) dan *soft constraint* (SC) [4] yang harus diperhatikan. HC harus dapat terpenuhi, misalnya tidak boleh ada jadwal bentrok dengan munculnya nama dosen yang sama di waktu dan ruang yang sama. Sedangkan SC lebih baik dapat dipenuhi atau boleh juga tidak, misalnya dosen maksimal mendapat jadwal dalam sehari dua atau tiga kali karena mempertimbangkan tingkat kejenuhan atau kepadatan jadwal.

Masalah penjadwalan ujian biasanya terjadi karena ujian memiliki durasi yang seragam dan jumlahnya terbatas, ketersediaan ruangan yang terkadang tidak memungkinkan untuk menjadwalkan ujian karena beberapa ujian dilakukan pada waktu yang sama, atau kendala lain seperti formasi penguji harus sesuai Daftar Urut Kepangkatan (DUK) [5] dan Grup Riset (GR) yang ditetapkan oleh institusi akademik.

GGA merupakan salah satu modifikasi dari algoritma genetika yang digunakan untuk permasalahan penjadwalan [2], [6]. Algoritma ini memiliki konsep mengelompokkan beberapa variabel gen menjadi satu gen [2]. Algoritma ini sangat cocok dengan permasalahan yang dihadapi untuk menyelesaikan permasalahan penjadwalan skripsi yang menggunakan parameter DUK dan GR. Judul skripsi dan penguji harus dalam satu GR yang sama dengan memperhatikan formasi penguji berdasarkan DUK.

Penjadwalan untuk seminar proposal skripsi di instansi biasanya dilakukan pada akhir semester ganjil dan ujian/sidang skripsi pada akhir semester genap. Pelaksanaannya terdiri dari 3 (tahap) yang mempunyai jeda dua pekan antar tahap. Biasanya setiap tahap terdapat 50 sampai 100 judul skripsi yang mendaftar untuk ujian. Sedangkan slot waktu yang diberikan seminimal mungkin dapat dilakukan, mengingat tahap berikutnya akan dibuka kembali. Seiring waktu berjalan, peningkatan jumlah mahasiswa terus terjadi. Hal tersebut berdampak terhadap jumlah judul skripsi yang terus bertambah. Jika tidak dibangun sebuah sistem yang dapat menyusun jadwal ujian skripsi, maka akan semakin rumit permasalahan yang dihadapi serta waktu yang dibutuhkan.

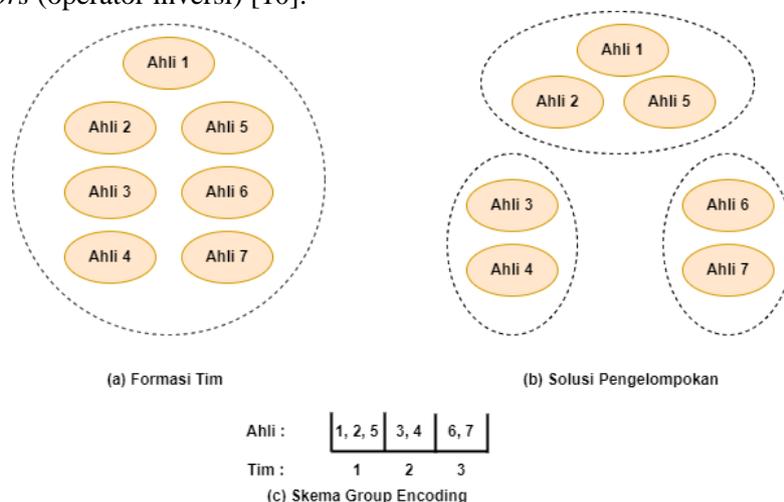
2 Tinjauan Literatur

Riset permasalahan penjadwalan telah lama dilakukan lebih dari 40 tahun, dimulai oleh Gotlieb pada tahun 1962 [2], [7]. Para peneliti telah melakukan banyak pendekatan untuk menyelesaikan permasalahan penjadwalan tersebut [8], [9], diantaranya [6], [10] menggunakan metode *graph coloring*, *constraint-based*, *population-based* (misalnya algoritma genetika (AG), optimasi koloni semut, algoritma memetic), *metaheuristic* (misalnya *tabu search* (TS), *simulated annealing* (SA), dan *great deluge*), *variable neighborhood search* (VNS), *hybrid* dan *hyperheuristic*.

Pada tahun 1993, Falkenauer [11] memperkenalkan algoritma *grouping genetic* hasil modifikasi algoritma genetika yang menerapkan pengkodean grup dan operator genetik terkait untuk memecahkan masalah pengelompokan. Perkembangan ini dilatarbelakangi oleh kebutuhan untuk meningkatkan kinerja AG pada masalah pengelompokan [10].

Algoritma ini lahir dari fakta bahwa yang diperhitungkan dalam masalah pengelompokan merupakan kelompok serta operator genetik harus fokus pada manipulasi kelompok dari pada item individual. Menurut Falkenauer [11], AG umumnya tidak bekerja dengan baik di permasalahan pengelompokan karena redundansi yang tinggi di antara kromosom dan ketidakpekaan konteks dari operator *crossover*. Berbeda dengan AG yang berfokus pada item individual, GGA berfokus pada kelompok item yang menjadi dasar fungsi dan *fitness* (nilai kebugaran kromosom) [12].

Algoritma *grouping genetic* (GGA) diilustrasikan seperti pada Gambar 1 berikut ini. Proses yang dilakukan antara lain (a) formasi tim, (b) solusi pengelompokan, (c) skema group encoding, yang didalamnya terdapat proses *group encoding* (pengkodean grup), *group crossover*, mutasi dan *inversion operators* (operator inversi) [10].

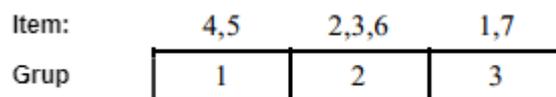


Gambar 1. Formasi Tim, Solusi Pengelompokan dan Skema Group Encoding

Menurut Falkenauer, tiap kelompok mewakili gen dan urutan item dalam kelompok tidak signifikan. Berbeda dengan pendekatan algoritma genetika, skema pengkodean grup tidak mengalami redundansi [10].

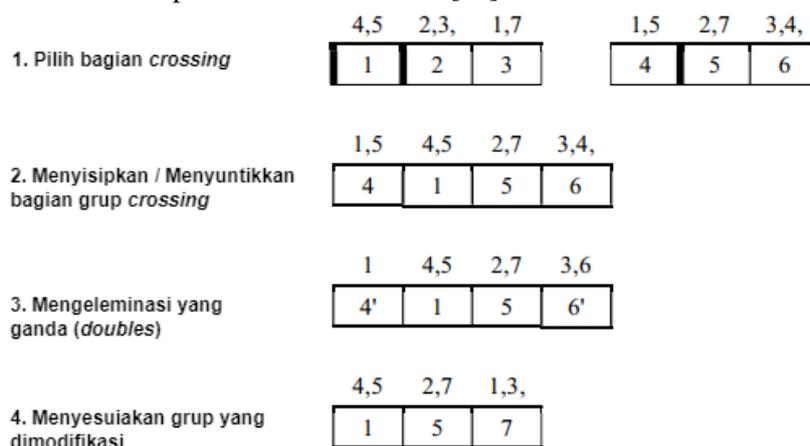
Pengelompokan masalah berfungsi untuk membangun himpunan bagian dari anggota (item) dari anggota himpunan yang diberikan berdasarkan beberapa kriteria optimasi sehingga kriteria keputusan dipusatkan pada komposisi kelompok item [6].

Skema pengkodean untuk masalah pengelompokan diilustrasikan seperti pada Gambar 2 [10] yang terdiri dari tiga kelompok, yaitu kelompok 1, 2, 3. Berisi item masing-masing {4,5}, {2,3,6}, dan {1,7}.



Gambar 2. Pengkodean Grup

Langkah selanjutnya adalah *group crossover* yang merupakan operator utama sebagai pertukaran informasi terpandu antar kromosom untuk menghasilkan kromosom baru tanpa gangguan yang merugikan dari struktur kelompok kromosom saat ini [10].



Gambar 3. Ilustrasi Operasi Group Crossover

Berdasarkan Gambar 3 sebagai contoh *group crossover*, langkah pertama adalah memilih dua titik silang dalam dua kromosom, dan pilih bagian persilangan dari tetua pertama, yang diberi label 1 pada Gambar 3. Langkah kedua adalah menyisipkan atau menyuntikkan bagian persilangan ke induk kedua yang menghasilkan keturunan baru yang kemungkinan mengandung item berulang, yang disebut *doubles* (ganda). Langkah ketiga adalah menyingkirkan item yang berulang, sambil menghindari item yang baru disuntikkan. Beberapa gen pada orang tua kedua dimodifikasi, misalnya, gen 4 dan 6 masing-masing dimodifikasi menjadi 4' dan 6'. Selanjutnya, gen yang dimodifikasi diadaptasi menggunakan heuristik spesifik masalah, yang menghasilkan gen 7. Berikutnya adalah saling menukar peran dua kromosom induk: induk 1 dan 2. Langkah terakhir adalah mengulangi langkah 2 sampai 4 sampai jumlah keturunan yang dibutuhkan dihasilkan [10].

Setelah *group crossover* adalah menjalankan *group mutation* yang memiliki tiga strategi utama yaitu membuat grup baru, menghilangkan grup yang dipilih, dan mengacak sejumlah kecil item yang dipilih di antara grup. Namun, ini juga akan tergantung pada masalah spesifik yang harus dipecahkan [10].

Operator inversi (*inversion*) menjadi operator pelengkap. Tujuan dari inversi adalah untuk memfasilitasi transmisi skemata yang baik dari orang tua (*parent*) ke keturunannya (*child*) untuk memastikan peningkatan laju pengambilan sampel skema tersebut yang berkinerja lebih baik. Sebagai contoh, kromosom [1 2 3 4] dapat dibalik menjadi kromosom [1 4 3 2], Karena item 1 dan 4 sekarang lebih dekat satu sama lain berimbang meningkatkan kemungkinan transmisi kedua gen 1 dan 4 bersama-sama ke generasi berikutnya ketika crossover berikutnya dilakukan [10].

Proses penjadwalan yang dilakukan dalam penelitian ini, dikombinasikan dengan pengelompokan parameter GR dan DUK. Daftar Urut Kepangkatan (DUK) adalah daftar yang memuat nama Pegawai Negeri Sipil dari suatu satuan organisasi negara yang disusun menurut

tingkatan kepangkatan. Acuan yang digunakan untuk menetapkan nomor urut dalam DUK menurut PP No. 15 Tahun 1979 tentang Daftar Urut Kepangkatan Pegawai Negeri Sipil [5] secara berturut-turut adalah a) pangkat; b) jabatan; c) masa kerja; d) latihan jabatan; e) pendidikan; dan f) usia.

3 Metode Penelitian

Kegiatan penelitian ini dilakukan di Jurusan Teknologi Informasi Politeknik Negeri Malang dengan studi kasus menggunakan data judul skripsi dari program studi D4 Teknik Informatika. Penelitian ini dilakukan dalam beberapa tahapan sebagai berikut.

A. Studi Literatur dan Definisi Masalah

Pada tahap studi literatur, tim peneliti melakukan kajian lebih lanjut mengenai konsep metode atau algoritma dan teknologi yang akan digunakan. Kajian tersebut antara lain mengenai metode yang tepat untuk melakukan penjadwalan secara tersistem. Selain itu, kajian tentang pengolahan data yang sesuai untuk mendapatkan informasi yang relevan terhadap jadwal yang akan dibangkitkan.

Proses ini dilakukan dengan harapan untuk menemukan inovasi atau ide-ide tambahan yang mampu mendukung prototipe yang akan dibangun. Inovasi atau ide-ide tersebut dapat dilihat dari sisi fitur sistem dan teknologi yang akan digunakan.

Permasalahan yang dihadapi ketika proses penjadwalan dapat didefinisikan sebagai *hard constraint* (HC) dan *soft constraint* (SC) [13]. HC merupakan batasan yang harus dipenuhi, sedangkan SC merupakan batasan yang boleh dipenuhi atau diabaikan. Rincian poin-poin tiap HC dapat dilihat sebagai berikut:

- HC1: Pembimbing utama dan pendamping tidak boleh sama dengan penguji utama dan pendamping.
- HC2: Penguji utama tidak boleh sama dengan penguji pendamping.
- HC3: Nama pembimbing dan penguji tidak boleh muncul lebih dari 1 (satu) kromosom pada hari dan sesi yang sama.
- HC4: Jika pembimbing atau penguji berhalangan hadir pada hari atau sesi tertentu, maka tidak masuk dalam proses penjadwalan pada hari atau sesi tersebut.
- HC5: Hanya judul yang telah disetujui oleh pembimbing dapat masuk proses penjadwalan.
- HC6: Penguji utama memiliki nilai DUK lebih tinggi dibanding penguji pendamping.
- HC7: Penguji harus dalam satu grup riset yang sama dengan judul yang akan diuji.
- HC8: Penguji utama adalah dosen senior dan penguji pendamping adalah dosen junior.

Untuk rincian poin-poin SC dapat dilihat sebagai berikut:

- SC1: Pembimbing atau penguji hanya boleh muncul pada sesi genap atau ganjil saja dalam sehari.
- SC2: Kuota jumlah menguji rata pada semua judul yang diuji.
- SC3: Hanya judul yang telah disetujui dosen pembimbing yang boleh dilakukan proses penjadwalan.

B. Observasi, Wawancara, dan Analisis Kebutuhan

Observasi dan analisis kebutuhan dilakukan dengan melakukan wawancara kepada panitia skripsi yang selama ini telah berpengalaman menyusun jadwal seminar proposal dan sidang skripsi secara konvensional. Hasil dari wawancara tersebut akan dirumuskan dalam bentuk kebutuhan fungsional dan kebutuhan non fungsional. Hasil dari tahapan ini adalah daftar kebutuhan fungsional dan non fungsional tersebut.

Beberapa hal yang perlu diketahui pada saat observasi dan analisis kebutuhan untuk mempermudah pada saat melakukan perancangan prototipe antara lain:

1. Data apa saja yang digunakan untuk menyusun jadwal.
2. Faktor apa saja yang memengaruhi penyusunan jadwal.
3. Faktor apa saja yang memengaruhi perubahan susunan jadwal.
4. Faktor apa saja yang memengaruhi hambatan dalam menyusun jadwal.
5. Aksi apa saja yang diperlukan untuk meningkatkan efektivitas dan efisiensi dalam penyusunan jadwal.

Setelah mendapatkan kebutuhan fungsional dan non-fungsional dari tahapan analisis kebutuhan, maka tim peneliti membuat rancangan prototipe sistem yang merupakan representasi kebutuhan dari pengguna. Rancangan prototipe sistem digambarkan dalam bentuk diagram-diagram yang digunakan sebagai pedoman dalam pembangunan prototipe. Diagram yang digunakan antara lain bisnis proses, diagram fungsional, diagram aktivitas, diagram entitas, serta diagram use case.

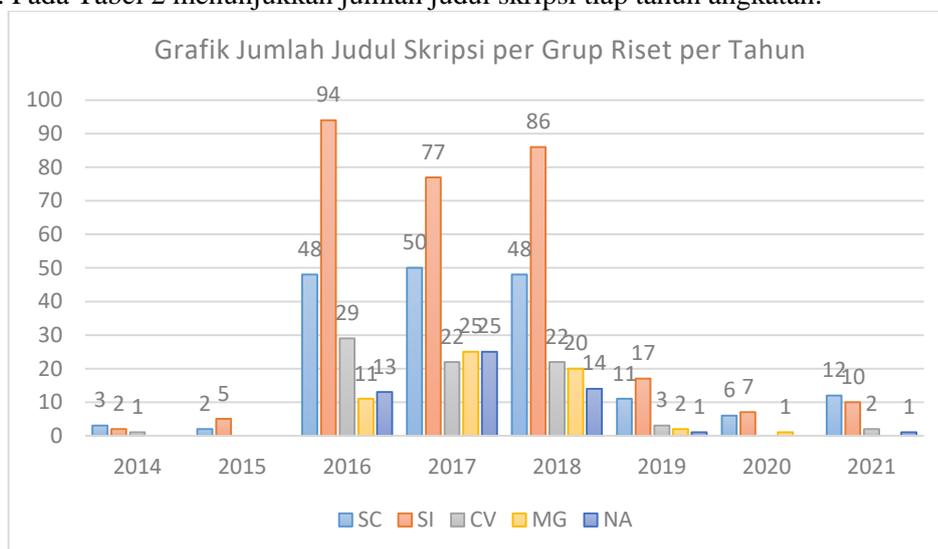
C. Analisis Data

Grup riset merupakan pengelompokan bidang minat untuk riset para dosen. Setiap dosen dapat memiliki grup riset mayor dan minor. Grup riset mayor adalah bidang minat utama, sedangkan grup riset minor adalah bidang minat sekunder. Setiap judul skripsi yang dikerjakan oleh mahasiswa dikelompokkan berdasarkan grup riset ini, karena setiap dosen yang membimbingnya juga merupakan bagian dari grup riset ini. Ada 5 (lima) jenis grup riset yang digunakan seperti terlihat pada Tabel 1 berikut.

Tabel 1. Data Grup Riset

ID	Kode	Nama
1	SC	Sistem Cerdas
2	SI	Sistem Informasi
3	CV	Computer Vision
4	MG	Multimedia dan Game
5	NA	Jaringan Komputer, Arsitektur, dan Keamanan Data

Data jumlah judul skripsi yang digunakan dalam penelitian ini dapat dilihat pada Gambar 4 berikut ini. Data ini diperoleh dari basis data panitia skripsi mulai dari angkatan mahasiswa 2014 sampai 2021. Pada Tabel 2 menunjukkan jumlah judul skripsi tiap tahun angkatan.

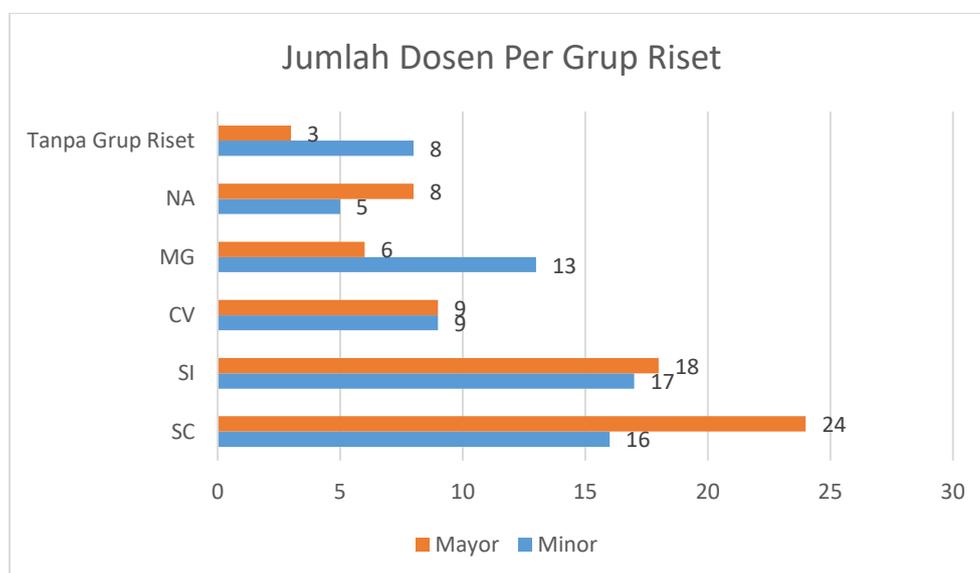


Gambar 4. Grafik Jumlah Judul Skripsi Per Grup Riset Per Tahun

Tabel 2. Data Jumlah Judul Skripsi per Tahun Angkatan

No.	Tahun Angkatan	Jumlah Judul
1.	2014	6
2.	2015	7
3.	2016	196
4.	2017	199
5.	2018	90
6.	2019	34
7.	2020	14
8.	2021	25
Total Keseluruhan		671

Pada Gambar 4 untuk angkatan tahun 2019, 2020 dan 2021 merupakan mahasiswa transfer dari prodi D2 dan D3. Namun, untuk jadwal ujian skripsi mengikuti jadwal dari angkatan tahun 2018, yaitu dijadwalkan pada semester genap tahun ajaran 2021/2022. Akhir semester genap sekitar bulan Juli-Agustus 2022. Atribut data judul proposal terdiri dari id judul, nim mahasiswa, nama mahasiswa, judul skripsi, id grup riset, id pembimbing utama. Sebaran jumlah dosen per grup riset dapat dilihat pada Gambar 5 berikut ini. Tiga dosen yang termasuk dalam kelompok tanpa grup riset mayor adalah dosen baru diterima (CPNS/Kontrak). Sedangkan dosen yang termasuk dalam kelompok tanpa grup riset minor adalah dosen-dosen dari ketua grup riset atau yang hanya fokus di satu bidang. Dosen tanpa grup riset mayor dalam kelompok ini akan dialokasikan sebagai penguji pendamping.



Gambar 5. Grafik Jumlah Dosen Per Grup Riset

Struktur data dosen dapat dilihat seperti pada Tabel 3 berikut ini. Ada 68 (enam puluh delapan) dosen yang digunakan dalam penelitian ini untuk menyusun jadwal skripsi. Kolom bobot nilai DUK akan ditambahkan pada tabel dosen, karena akan digunakan saat penentuan urutan pengujian.

Tabel 3 Sampel Struktur Data Dosen

ID	Nama	NIP	Status	Golongan	Jabfung	Aktif Menguji	Aktif Membimbing	Grup Riset Mayor	Grup Riset Minor	Pendidikan
1	Xxx		PNS SENIOR	III/C	LEKTOR	ya	ya	SI	SC	S2
2	Yyy		PNS SENIOR	IV/A	LEKTOR KEPALA	ya	ya	SI	SC	S3
...				IV/A	LEKTOR KEPALA	ya	ya	CV	SI	S3
...			PNS JUNIOR	III/B	ASISTEN AHLI	tidak	ya	MG	SI	S2
68	Zzz		KONTRAK	NULL	NULL	tidak	ya	NA	SI	S2

Saat penelitian ini dilakukan, ruang sidang skripsi pada saat pandemi Covid-19 menggunakan aplikasi Zoom. Tersedia 10 (sepuluh) ruang daring seperti yang terlihat pada Tabel 4 berikut ini.

Tabel 4. Data Ruang

ID	Kode Ruang	Label Ruang	ID Zoom	Host Key
1	R.7.13	Ruang 01	1234567xxx	170xxx
2	R.7.15	Ruang 02	1234567xxx	170xxx
3	R.7.16	Ruang 03	1234567xxx	170xxx
4	R.7.17	Ruang 04	1234567xxx	170xxx
5	R.7.18	Ruang 05	1234567xxx	170xxx
6	R.7.20	Ruang 06	1234567xxx	170xxx
7	R.7.07	Ruang 07	1234567xxx	170xxx

8	R.7.08	Ruang 08	1234567xxx	170xxx
9	R.7.19	Ruang 09	1234567xxx	170xxx
10	R.7.06	Ruang 10	1234567xxx	170xxx

Durasi waktu pelaksanaan seminar proposal skripsi adalah 60 (enam puluh) menit. Dalam sehari bisa dilakukan 6 (sesi) pelaksanaan seminar seperti pada Tabel 5 berikut ini. Untuk durasi waktu sidang skripsi adalah 90 (sembilan puluh) menit. Lebih sedikit sesi yang bisa dilaksanakan dalam sehari yaitu hanya 4 (empat) sesi seperti terlihat pada Tabel 6 berikut ini.

Tabel 5. Data Sesi Proposal

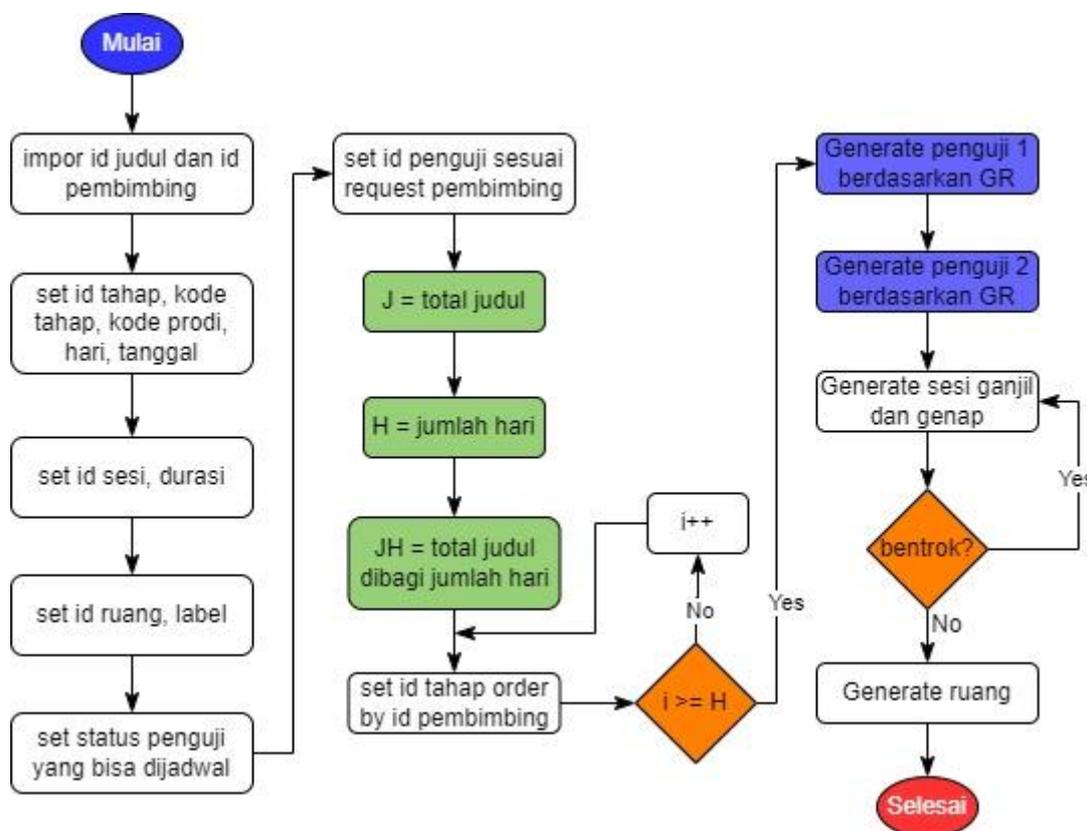
ID	Sesi
1	08:00 – 09:00
2	09:15 – 10:15
3	10:30 – 11:30
4	13:00 – 14:00
5	14:15 – 15:15
6	15:30 – 16:30

Tabel 6. Data Sesi Sidang

ID	Sesi
1	08:00 – 09:30
2	10:00 – 11:30
3	13:00 – 14:30
4	15:00 – 16:30

D. Perancangan Algoritma dan Pembangunan Sistem Informasi

Algoritma untuk membangkitkan jadwal seminar dan ujian skripsi dapat dilihat seperti pada Gambar 6 berikut ini.



Gambar 6. Flowchart Algoritma Penjadwalan Skripsi

Berdasarkan flowchart Gambar 6 di atas, langkah pertama adalah mengimpor data id judul dan id pembimbing sebagai moderator ke tabel kromosom (jadwal). Lalu tentukan id tahap, kode tahap, kode prodi, hari dan tanggal ujian yang akan dilaksanakan. Kemudian tentukan jumlah sesi dan jam pelaksanaan dalam sehari, maka diperoleh id sesi. Selanjutnya tentukan jumlah ruang yang akan digunakan, maka diperoleh id ruang. Lalu tentukan jumlah dosen penguji yang bisa hadir, jika berhalangan hadir maka tidak dijadwalkan. Masukkan id dosen yang tidak bisa hadir pada hari

tertentu ke dalam daftar *exclude* sesuai hari. Berikutnya tentukan beberapa id judul untuk diset pengujinya jika ada permintaan khusus atau hari/sesi tertentu. Lalu hitung jumlah judul skripsi per grup riset yang masih kosong pada id tahap, sesi, atau pengujinya. Ambil total judul skripsi lalu dibagi dengan jumlah hari pelaksanaan ujian, lalu set id tahap secara merata per hari pada masing-masing judul. Isi id sesi ganjil dan genap bergantian per hari dengan cek bentrok paralel proses bangkitkan penguji utama dengan status dosen adalah PNS senior dengan grup riset yang sesuai judul skripsi. Bangkitkan penguji pendamping dengan status dosen adalah bukan PNS senior dengan grup riset yang sesuai judul skripsi. Tinjau kembali hasil pembangkitan jadwal, jika sudah selesai dan tidak ada isu, maka jadwal siap dirilis.

Parameter DUK digunakan untuk menilai bobot dari penguji yang berhak menjadi penguji utama atau pendamping. Pada Tabel 7 berikut menunjukkan bobot nilai DUK untuk pangkat dosen. Sedangkan pada Tabel 8 untuk jabatan dosen dan Tabel 9 untuk jenjang pendidikan dosen.

Tabel 7. Nilai DUK untuk Pangkat Dosen

No.	Pangkat Golongan Ruang	Nilai DUK
1	CPNS/Kontrak	0
2	III/b	1
3	III/c	2
4	III/d	3
5	IV/a	4
6	IV/b	5
7	IV/c	6
8	IV/d	7
9	IV/e	8

Tabel 8. Nilai DUK untuk Jabatan Dosen

No.	Jabatan Fungsional	Nilai DUK
1	NULL	0
2	Asisten Ahli	1
3	Lektor	2
4	Lektor Kepala	3
5	Profesor	4

Tabel 9. Nilai DUK untuk Jenjang Pendidikan Dosen

No.	Jenjang Pendidikan	Nilai DUK
1	S2	1
2	S3	2

Setelah dihitung bobot nilai DUK setiap dosen, kemudian disimpan ke kolom yang telah disediakan di dalam basis data. Susunan masing-masing bobot nilai DUK kemudian dijadikan seperti pada Tabel 10 berikut ini.

Tabel 10. Susunan bobot nilai DUK

Susunan parameter nilai DUK	
ABCCDEE	
A	= nilai bobot pangkat
B	= nilai bobot jabatan
CC	= nilai bobot masa kerja
D	= nilai bobot pendidikan
EE	= nilai bobot usia

Untuk pembobotan nilai latihan jabatan tidak digunakan, karena data tersebut sulit diperoleh. Namun dari 5 (lima) kriteria DUK, sudah cukup untuk mewakili nilai DUK yang akan digunakan sebagai pembobotan untuk menentukan urutan penguji utama dan pendamping. Bobot nilai DUK dapat dimanfaatkan ketika terjadi perubahan penguji secara mendadak di hari pelaksanaan ujian. Karena kejadian yang selalu berulang adalah mencari kandidat penguji yang sesuai dengan grup riset dan urutan nilai DUK. Sehingga pemilihan penguji akan dapat dipertanggungjawabkan dan bukan keputusan personal subjektif dari *stakeholder* atau pihak lain yang berkepentingan.

Contoh pemenggalan NIP untuk mengambil tahun kelahiran dan tahun angkatan CPNS dosen sebagai perhitungan nilai DUK adalah sebagai berikut.

1992 0412 2019 031013

Setelah proses pemenggalan NIP, diperoleh angka tahun kelahiran di penggalan pertama dan angka tahun angkatan CPNS di penggalan kedua yang ditunjukkan pada teks bercetak tebal tersebut. Selanjutnya dihitung bobot masa kerja dengan rumus seperti pada persamaan (1) berikut ini dengan cara menghitung selisih tahun saat ini dan tahun diangkatnya CPNS.

$$f(MK) = Tskr - Tcpns \quad (1)$$

Kemudian dihitung bobot nilai DUK berdasarkan parameter usia menggunakan rumus seperti pada persamaan (2) berikut ini dengan menghitung selisih tahun saat ini dan tahun kelahiran dosen.

$$f(U) = Tskr - Tlahir \quad (2)$$

Algoritma dalam proses pembobotan nilai DUK tiap dosen dapat dilihat pada Algoritma 1 berikut ini. Lalu dilakukan secara iteratif kepada setiap individu dosen, sehingga masing-masing dosen akan mempunyai bobot nilai DUK yang berbeda-beda berupa bilangan bulat (integer). Hasil bobot nilai DUK ini mudah dilakukan perbandingan oleh sistem atau program dengan logika matematika sederhana seperti lebih dari atau kurang dari.

Algoritma 1 Pembobotan nilai DUK

- 1: **while** ambil daftar dosen dari Tabel 3 **do**
 - 2: pangkat = ambil bobot nilai dari Tabel 7
 - 3: jabatan = ambil bobot nilai dari Tabel 8
 - 4: masa kerja = rumus nomor (1)
 - 5: pendidikan = ambil bobot nilai dari Tabel 9
 - 6: usia = rumus nomor (2)
 - 7: simpan bobot nilai DUK tiap dosen
 - 8: **end while**
-

Algoritma yang digunakan untuk penentuan posisi penguji berdasarkan bobot nilai DUK dapat dilihat pada Algoritma 2 berikut ini. Selain atribut PNS Senior dan non-senior membantu dalam proses penentuan posisi penguji, namun tidak semua penguji memiliki formasi PNS senior dan non-senior. Terkadang formasi dapat terjadi berupa sama-sama PNS/CPNS non-senior, hal ini sangat membantu ketika ada bobot nilai DUK.

Algoritma 2 Penentuan posisi penguji berdasarkan bobot nilai DUK

- 1: **if** bobot nilai DUK dosen A lebih besar dari dosen B
 - 2: **then** set dosen A sebagai penguji utama, B sebagai pendamping
 - 3: **else** set dosen A sebagai penguji pendamping, B sebagai utama
 - 4: **endif**
-

Algoritma utama untuk membangkitkan jadwal dapat dilihat pada Algoritma 3 berikut ini. Algoritma ini merupakan bagian inti dalam proses pembangkitan jadwal seminar proposal dan sidang skripsi.

Algoritma 3 Pembangkitan Jadwal

- 1: impor data id judul dan id moderator (pembimbing)
 - 2: tentukan id tahap (berapa hari pelaksanaan)
 - 3: tentukan id sesi dan durasi/jam pelaksanaan
 - 4: tentukan id ruang dan ruang/link zoom yang akan digunakan
 - 5: cek dosen yang bisa menguji, jika tidak bisa hadir maka dimasukkan ke daftar *exclude*
 - 6: set manual beberapa *request/constraint* dari dosen misalnya hari, sesi, atau penguji untuk judul tertentu
 - 7: TJ := hitung jumlah total judul per grup riset
-

-
- 8: for i to jumlah id grup riset
 - 9: jumlah judul dalam sehari = TJ / jumlah hari pelaksanaan
 - 10: *BangkitkanPenguji(utama/pendamping)*
-

Fungsi untuk menentukan penguji menggunakan Algoritma 4 berikut ini. Dalam menentukan formasi penguji juga mempertimbangkan kehadiran dan jumlah penguji.

Algoritma 4 *BangkitkanPenguji()*

- 1: inisial id judul, id tahap, id sesi, id pembimbing, id grup riset
 - 2: ambil kandidat dosen dengan grup riset yang sama
 - 3: ambil kandidat dosen yang bukan pembimbing
 - 4: ambil kandidat dosen dengan status senior jika untuk penguji utama, ambil status non senior jika untuk penguji pendamping
 - 5: ambil kandidat dosen dengan hak menguji dan tidak termasuk dalam daftar exclude
 - 6: ambil kandidat dosen dari penguji seminar proposal dan bukan pembimbing
 - 7: hitung jumlah menguji kandidat dosen
 - 8: ambil kandidat dosen pada tahap dan sesi yang tersedia
 - 9: ambil kandidat dosen dengan jumlah menguji yang paling sedikit
 - 10: set id dosen sebagai penguji utama atau pendamping
-

Setelah jadwal terbentuk, proses selanjutnya adalah mengecek setiap jadwal, jika ada yang bentrok, maka akan ditampilkan sebagai jadwal bentrok. Algoritmanya dapat dilihat pada Algoritma 5 berikut ini.

Algoritma 5 Fungsi cek jadwal bentrok

- 1: **init** id tahap, id sesi
 - 2: **obtain** semua dosen dengan tahap dan sesi yang sama
 - 3: **count** kemunculan dosen pada tahap dan sesi yang sama
 - 4: **if** kemunculan > 1
 - 5: **then** terdapat bentrok
 - 6: **endif**
-

Algoritma 6 berikut untuk memastikan pembimbing tidak menguji judulnya sendiri, artinya formasi pembimbing utama dan pendamping serta penguji utama dan pendamping harus berbeda semua.

Algoritma 6 Fungsi cek pembimbing tidak sama dengan penguji

- 1: **for each** jadwal **in** kromosom jadwal
 - 2: get pembimbing utama
 - 3: get pembimbing pendamping
 - 4: get penguji utama
 - 5: get penguji pendamping
 - 6: **if** pembimbing utama == penguji utama **or** pembimbing pendamping == penguji utama **or** pembimbing utama == penguji pendamping **or** pembimbing pendamping == penguji pendamping
 - 7: **then** tandai pembimbing == penguji, get id judul
 - 8: **endif**
 - 9: **endfor**
-

Algoritma 7 berikut untuk menghitung jumlah judul yang sudah terjadwal, sehingga tidak timpang jumlah menguji antar dosen. Sehingga beban setiap dosen memiliki jumlah yang rata. Walaupun secara pasti tidak mungkin sama rata, karena itu termasuk dalam SC.

Algoritma 7 Fungsi counter penguji sidang

-
- 1: Ambil semua id penguji utama pada semua tahap
 - 2: Ambil semua id penguji pendamping pada semua tahap
 - 3: Hitung jumlah menguji tiap dosen penguji
-

E. Pengujian Hasil

Pengujian dilakukan dengan beberapa metode, yaitu menghitung nilai fitness seperti pada persamaan (3) dari algoritma yang telah dijalankan, pengujian fungsional dan non-fungsional pada sistem yang telah dibangun. Rumus untuk menghitung nilai fitness [2], [13] dari GGA adalah sebagai berikut.

$$f(f) = \frac{1}{1+(\sum_n HC)} \quad (3)$$

Nilai F adalah nilai Fitness. Fungsi angka “1” pada denominator (besaran nilai pembagi) adalah agar nilai maksimal yang didapatkan = “1” untuk tidak adanya nilai konflik (nilai konflik = “0”), serta mencegah denominator = “0”.

4 Hasil dan Pembahasan

Pada bagian ini, akan dibahas hasil dan pembahasan penelitian yang telah dilakukan. Penelitian ini memanfaatkan parameter nilai DUK untuk menentukan siapa yang berhak menjadi dosen penguji utama dan pendamping. Meskipun dengan atribut dosen senior dan junior dapat ditentukan, namun adanya DUK sangat bermanfaat ketika mendadak formasi penguji harus digantikan. Sedangkan parameter grup riset digunakan untuk memilih kandidat dosen yang masih linear dengan judul skripsi yang akan diuji. Sehingga diharapkan setiap dosen penguji telah sesuai dengan bidang topik skripsinya.

Sistem informasi yang telah dibangun menggunakan konsep *software wizard*. *Software wizard* atau *setup assistant* adalah antarmuka pengguna yang menampilkan kotak dialog untuk memandu pengguna melalui urutan langkah-langkah kecil [14], [15]. Tabel 10 berikut ini menunjukkan urutan menu dan submenu yang telah dikembangkan ke dalam sebuah sistem informasi penjadwalan skripsi.

Tabel 10. Konsep Menu Dengan Software Wizard

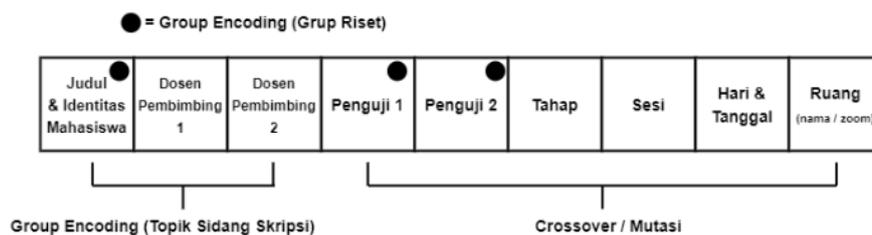
Main Menu	Sub Menu	Deskripsi
Beranda		
Jadwal Sempuro	1. Data Awal	Impor data id judul dan id moderator
	2. Tahap	Tentukan jumlah hari pelaksanaan dengan id tahap
	3. Ruang	Tentukan ruang yang akan digunakan
	4. Sesi	Tentukan jam dan durasi/sesi pelaksanaan
	5. Dosen	Tentukan jumlah dosen atau cek status
	6. Permintaan Khusus	Set manual hari, sesi, atau penguji pada judul tertentu
	7. Bangkitkan Jadwal	Bangkitkan jadwal secara otomatis
	8. Peninjauan	Peninjauan ulang jadwal proposal yang telah berhasil dibangkitkan
Jadwal Sidang	1. Data Awal	Impor data id judul dan id moderator
	2. Tahap	Tentukan jumlah hari pelaksanaan dengan id tahap
	3. Ruang	Tentukan ruang yang akan digunakan
	4. Sesi	Tentukan jam dan durasi/sesi pelaksanaan
	5. Dosen	Tentukan jumlah dosen atau cek status
	6. Permintaan Khusus	Set manual hari, sesi, atau penguji pada judul tertentu
	7. Bangkitkan Jadwal	Bangkitkan jadwal secara otomatis
	8. Peninjauan	Peninjauan ulang jadwal sidang yang telah berhasil dibangkitkan

Penelitian ini menghasilkan prototipe berupa aplikasi berbasis web. Aplikasi ini dapat diakses melalui tautan: <http://tugasakhir.jti.polinema.ac.id/penjadwalan>.

Setelah melakukan proses login, pengguna superadmin dan panitia akan diarahkan langsung ke halaman beranda seperti yang terlihat pada Gambar 7 berikut ini. Pada halaman beranda ini terdapat petunjuk penggunaan aplikasi, terutama untuk proses menjadwalkan seminar proposal dan sidang skripsi.



Gambar 7. Tampilan Dashboard Sistem Informasi Penjadwalan Skripsi Berbasis Web



Gambar 8. Struktur Kromosom Jadwal Skripsi

Pada Gambar 8 mengilustrasikan struktur kromosom pada satu jadwal skripsi. Gen yang harus dalam satu grup riset yang sama adalah topik/judul skripsi dengan kedua penguji. Pembimbing, mahasiswa dan judul tidak dapat dipisahkan, mereka menjadi satu grup genetik. Sedangkan gen lainnya memungkinkan untuk dilakukan mutasi atau *crossover*. ID tahap dan tanggal menjadi satu grup genetik, karena itu tidak dapat dipisahkan dan setiap ID tahap memiliki atribut hari, tanggal, dan kode tahap pendaftaran skripsi. Misalnya kode tahap P1 untuk proposal tahap 1 dan S1 untuk sidang tahap 1 dan seterusnya.

SELECT * FROM `v2_jadwal` LIMIT 50 OFFSET 1300 (0.002 s) Edit

<input type="checkbox"/> Modify	id	id_tahap	id_ruang	sesi	id_judul	id_moderator	id_penguji1	id_penguji2
<input type="checkbox"/> edit	1952	84	1	4	1100	325	14	335
<input type="checkbox"/> edit	1953	84	6	2	1101	345	26	330
<input type="checkbox"/> edit	1954	84	7	2	1103	10	25	337
<input type="checkbox"/> edit	1955	84	1	5	1104	345	26	311
<input type="checkbox"/> edit	1956	84	8	2	1106	301	18	351
<input type="checkbox"/> edit	1957	84	2	4	1107	7	26	311
<input type="checkbox"/> edit	1958	84	4	3	1112	329	3	337

Gambar 9. Sampel Data Kromosom Jadwal Yang Telah Dibangkitkan

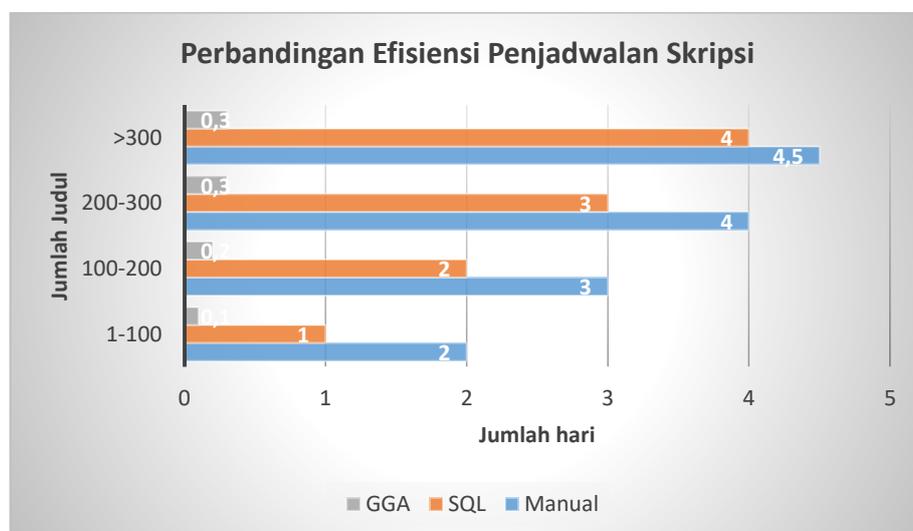
Pada Gambar 9 di atas menunjukkan sampel data kromosom jadwal yang telah berhasil dibangkitkan. Data ini mempunyai total data jadwal sebanyak 1397. Data ini terdiri dari angkatan 2014 hingga 2021, termasuk pembagian tahap ujian per tahun angkatan masing-masing. Hasil validasi dan verifikasi hasil kromosom jadwal berdasarkan HC dan SC yang dihasilkan dapat dilihat pada Tabel 11 berikut ini. M berarti “memenuhi” dan TM adalah “tidak memenuhi”. Perlu dilakukan tiga kali iterasi untuk mencapai hasil jadwal yang maksimum sesuai kriteria HC dan SC yang telah didefinisikan. Pada iterasi-1 memiliki nilai *fitness* sebesar $\frac{9}{11}$ atau 81,8%, pada iterasi-2 sebesar 90,9% dan iterasi-3 baru memperoleh nilai maksimal *fitness*-nya menjadi 100%.

Tabel 11. Hasil Validasi HC dan SC Terhadap Kromosom Jadwal

No.	Kriteria	Iterasi-1	Iterasi-2	Iterasi-3
-----	----------	-----------	-----------	-----------

1	HC1	M	M	M
2	HC2	M	M	M
3	HC3	M	M	M
4	HC4	M	M	M
5	HC5	M	M	M
6	HC6	M	M	M
7	HC7	M	M	M
8	HC8	M	M	M
9	SC1	TM	TM	M
10	SC2	TM	M	M
11	SC3	M	M	M

Hasil perbandingan kecepatan penjadwalan dapat dilihat pada grafik Gambar 10 berikut ini. Berdasarkan Gambar 10 menunjukkan bahwa penjadwalan GGA memiliki efisiensi sebesar 99,97% dibanding menggunakan SQL dan cara manual. Proses SQL yang dilakukan sebenarnya memanfaatkan *query*, *view*, dan *store procedure* di basisdata MySQL untuk melakukan proses penjadwalan. Ini merupakan salah satu strategi untuk membuat prototipe sebelum memanfaatkan GGA.



Gambar 10. Perbandingan Efisiensi Penjadwalan Skripsi

5 Kesimpulan

GGA adalah pengembangan dari algoritma genetik klasik untuk menyelesaikan masalah komputasi pengelompokan berdasarkan permasalahan objektif kelompok-kelompok kecil. Penelitian yang telah dilakukan adalah menggabungkan GGA dengan parameter DUK dan GR untuk menentukan formasi jadwal skripsi. Metode ini telah dilakukan menggunakan data riil judul skripsi sejak tahun angkatan 2014 sampai 2021 di Prodi D4 Teknik Informatika, Politeknik Negeri Malang.

Hasil penelitian menunjukkan bahwa GGA dengan DUK dan GR berhasil meningkatkan efisiensi sebesar 99,97% apabila diterapkan untuk menyelesaikan masalah pada proses penjadwalan ujian skripsi dibanding menggunakan metode SQL dan manual. Metode yang diusulkan menunjukkan stabil dan kualitas hasilnya dapat diterima sesuai kriteria HC dan SC yang telah didefinisikan.

Ucapan Terima Kasih

Ucapan terimakasih dan apresiasi kepada seluruh pihak-pihak yang terlibat dalam mendukung penelitian ini dapat terlaksana dengan baik. Ucapan terimakasih juga kami sampaikan kepada Jurusan Teknologi Informasi yang telah memberi kesempatan dan P2M Politeknik Negeri Malang yang telah mendanai kegiatan penelitian ini. Tidak lupa terimakasih kepada segenap pengurus Jurnal SISTEMASI atas kesempatan yang diberikan.

Referensi

- [1] M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall, "A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities," *IEEE Access*, vol. 9, pp. 106515–106529, 2021, doi: 10.1109/ACCESS.2021.3100613.
- [2] O. Ramos-Figueroa, M. Quiroz-Castellanos, E. Mezura-Montes, and R. Kharel, "Variation Operators for Grouping Genetic Algorithms: A Review," *Swarm Evol Comput*, vol. 60, p. 100796, Feb. 2021, doi: 10.1016/J.SWEVO.2020.100796.
- [3] D. Van Bulck and D. Goossens, "The international timetabling competition on sports timetabling (ITC2021)," *Eur J Oper Res*, Nov. 2022, doi: 10.1016/J.EJOR.2022.11.046.
- [4] S. Yang and S. N. Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling," *IEEE Trans Syst Man Cybern*, vol. 41, no. 1, pp. 93–106, 2011, doi: 10.1109/TSMCC.2010.2049200.
- [5] Pemerintah Pusat, *Peraturan Pemerintah (PP) tentang Daftar Urut Kepangkatan Pegawai Negeri Sipil*. Indonesia, 1979. [Online]. Available: <https://peraturan.bpk.go.id/Home/Details/66790/pp-no-15-tahun-1979>
- [6] M. Mutingi and C. Mbohwa, *Grouping Genetic Algorithm Advances and Applications*. 2015. doi: 10.1201/b18889-7.
- [7] C. C. Gotlieb, "The construction of class-teacher timetables," in *Proc. IFIP Congr.*, 1962, pp. 73–77.
- [8] S. B. Bhaskoro, B. Bayu Aji, and S. Aminah, "Sistem Penjadwalan Sidang Tugas Akhir menggunakan Algoritma Genetika," *JTT (Jurnal Teknologi Terapan)*, vol. 7, no. 1, p. 27, May 2021, doi: 10.31884/jtt.v7i1.310.
- [9] Y. Bastian, D. J. Surjawan, and A. Adelia, "Aplikasi Penjadwalan Sidang Tugas Akhir di Fakultas Teknologi Informasi Universitas Kristen Maranatha Dengan Metode Depth First Search," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 4, no. 1, pp. 173 – 184–173 – 184, Apr. 2018, doi: 10.28932/jutisi.v4i1.760.
- [10] M. Mutingi, C. Mbohwa, and H. Musiyarira, "Grouping Genetic Algorithms: An Exploratory Study," in *Proceedings of the World Congress on Engineering and Computer Science*, 2017.
- [11] E. Falkenauer, "The grouping genetic algorithms - widening the scope of the GAs," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 33, pp. 79–102, 1993.
- [12] M. Mutingi and C. Mbohwa, *Grouping Genetic Algorithm Advances and Applications*. 2015. doi: 10.1201/b18889-7.
- [13] S. Yang and S. N. Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling," *IEEE Trans Syst Man Cybern*, vol. 41, no. 1, pp. 93–106, 2011, doi: 10.1109/TSMCC.2010.2049200.
- [14] N. Babich, "Wizard Design Pattern." <https://uxplanet.org/wizard-design-pattern-8c86e14f2a38> (accessed Nov. 02, 2022).
- [15] J. Tidwell, M. van Welie, J. Bollaert, S. Carliner, and B. Baxley, "Wizard Design Pattern." <https://ui-patterns.com/patterns/Wizard> (accessed Nov. 02, 2022).