**IMPLEMENTASI ALGORITMA GENETIKA**

**PADA PENJADWALAN MATA KULIAH**

**(STUDI KASUS: PROGRAM STUDI INFORMATIKA UNIVERSITAS MULTIMEDIA NUSANTARA)**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom.)**



**Disusun Oleh**

**Ardhi Prago**

**00000015924**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS MULTIMEDIA NUSANTARA**

**TANGERANG**

**2020**

# HALAMAN PERSETUJUAN

**Skripsi dengan Judul**

**IMPLEMENTASI ALGORITMA GENETIKA**

**PADA PENJADWALAN MATA KULIAH (STUDI KASUS: PROGRAM STUDI INFORMATIKA UNIVERSITAS MULTIMEDIA NUSANTARA)**

Oleh

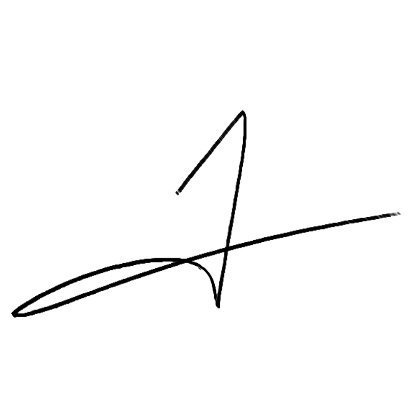
Ardhi Prago

00000015924

**Telah disetujui untuk diajukan pada**

**Sidang Ujian Skripsi Universitas Multimedia Nusantara**

Tangerang, 15 Juni 2020

Menyetujui,

Dosen Pembimbing 1 Dosen Pembimbing 2

Julio Christian Young, S.Kom., M.Kom. Nunik Afriliana, S.Kom., MMSI

Ketua Program Studi Informatika

Nunik Afriliana, S.Kom., MMSI

# PERNYATAAN TIDAK MELAKUKAN PLAGIAT

Dengan ini saya:

Nama : Ardhi Prago

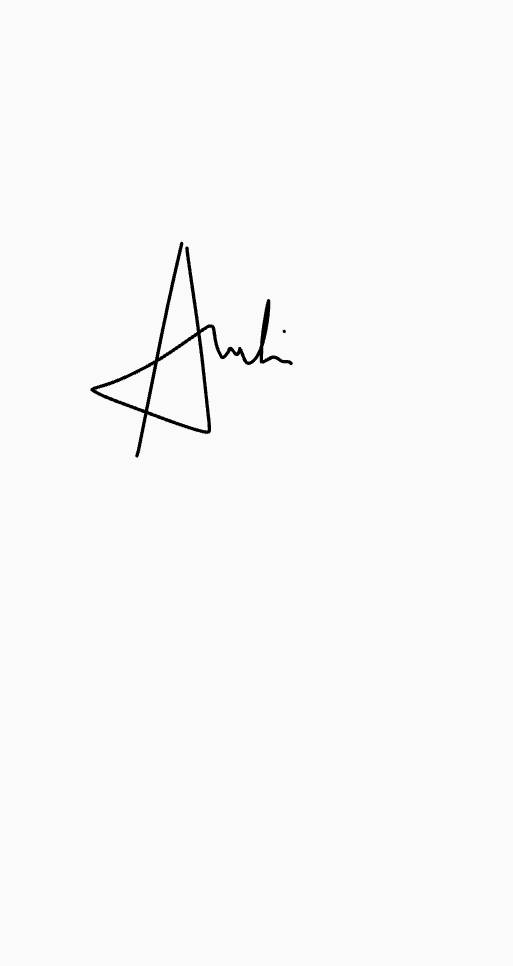
NIM : 00000015924

Program Studi : Informatika

Fakultas : Teknik dan Informatika

Menyatakan bahwa skripsi yang berjudul “**Implementasi Algoritma Genetika Pada Penjadwalan Mata Kuliah (Studi Kasus: Program Studi Informatika Universitas Multimedia Nusantara)**” adalah karya ilmiah saya sendiri, bukan plagiat dari karya ilmiah yang ditulis oleh orang lain atau Lembaga lain, dan semua karya ilmiah yang dirujuk dalam Skripsi ini telah disebutkan sumber kutipannya serta dicantumkan di daftar pustaka.

Jika pada kemudian hari terbukti ditemukan kecurangan/penyimpangan, baik dalam pelaksanaan Skripsi maupun dalam penulisan laporan Skripsi, saya bersedia menerima konsekuensi dinyatakan TIDAK LULUS untuk mata kuliah Skripsi yang telah saya tempuh.

Tangerang, 15 Juni 2020

Ardhi Prago

# PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Multimedia Nusantara, saya yang bertanda tangan di bawah ini:

Nama : Ardhi Prago

NIM : 00000015924

Program Studi : Informatika

Fakultas : Teknik dan Informatika

Jenis Karya : Skripsi

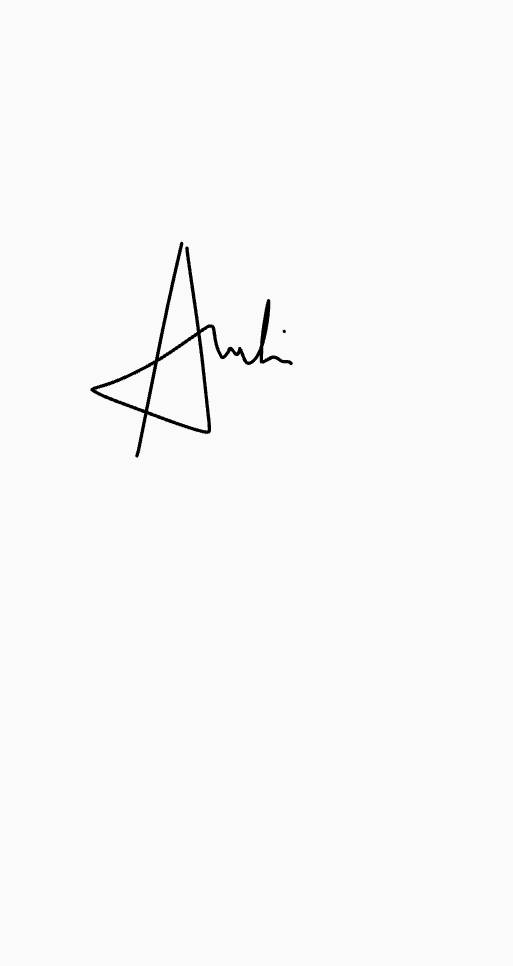
Demi pengembangan ilmu pengetahuan, menyetujui dan memberikan izin kepada Universitas Multimedia Nusantara hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**Implementasi Algoritma Genetika Pada Penjadwalan Mata Kuliah (Studi Kasus: Program Studi Informatika Universitas Multimedia Nusantara)**

beserta perangkat yang diperlukan.

Dengan Hak Bebas Royalti Non-eksklusif ini, pihak **Universitas Multimedia Nusantara** berhak menyimpan, mengalihmedia atau *format*-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mendistribusi dan menampilkan atau mempublikasikan karya ilmiah saya di internet atau media lain untuk kepentingan akademis, tanpa perlu meminta izin dari saya maupun memberikan royalti kepada saya, selama tetap mencantumkan nama saya sebagai penulis karya ilmiah tersebut.

Demikian pernyataan ini saya buat dengan sebenarnya untuk dipergunakan sebagaimana mestinya.

Tangerang, 15 Juni 2020

Ardhi Prago

# HALAMAN PERSEMBAHAN

“*Never give up. Today is hard, Tomorrow will be worse. But the day after tomorrow will be sunshine”*

* *Jack Ma*

# KATA PENGANTAR

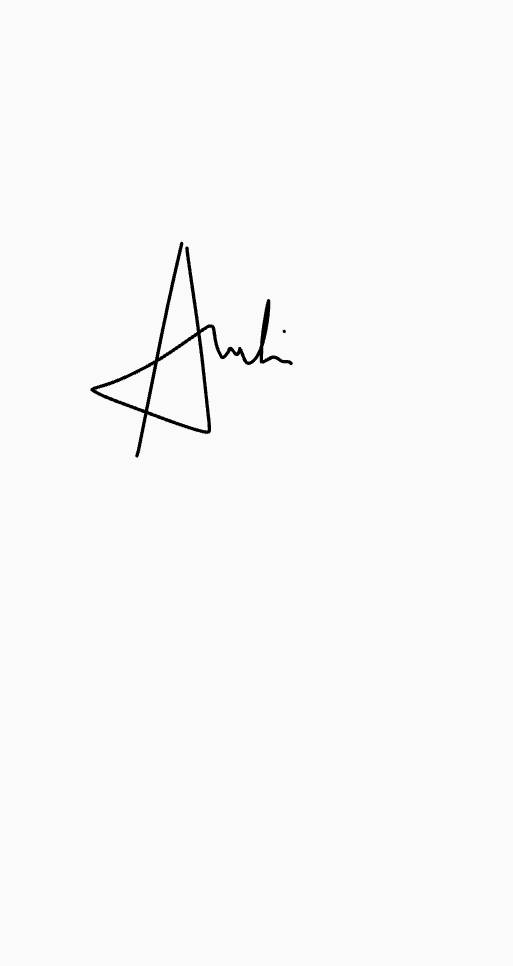
Puji dan syukur atas berkat rahmat Tuhan Yang Maha Esa yang telah melimpahkan rahmat-Nya sehingga penulis senantiasa mendapatkan kekuatan untuk menyelesaikan skripsi ini dengan baik dan tepat waktu. Skripsi ini disusun dan diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana.

Dalam proses penyelesaian skripsi, penulis banyak mendapatkan bantuan dari berbagai pihak. Karena itu, dalam kesempatan kali ini, penulis juga ingin mengucapkan banyak terima kasih kepada:

1. Dr. Ninok Leksono selaku Rektor Universitas Multimedia Nusantara.
2. Nunik Afriliana, S.Kom., M.M.S.I. selaku Kepala Program Studi Informatika Universitas Multimedia Nusantara serta selaku dosen pembimbing kedua yang selalu memberikan bimbingan, arahan, dan motivasi kepada penulis baik selama penulisan laporan Skripsi maupun selama masa perkuliahan.
3. Julio Christian Young, S.Kom., M.Kom. selaku dosen pembimbing pertama yang telah membimbing dengan sabar melalui kesalahan dan kegagalan yang telah dilakukan sehingga laporan Skripsi dapat terselesaikan dengan baik.
4. Kedua orang tua penulis yang selalu memberikan kekuatan, motivasi, dukungan secara moral dan material, dan teguran selama pembuatan laporan Skripsi.
5. Seluruh dosen Universitas Multimedia Nusantara yang telah mengajar penulis selama menempuh masa perkuliahan.
6. Teman-teman penulis yang telah memberikan banyak masukan serta semangat dalam pembuatan program serta laporan Skripsi.
7. Gemma Cahya Hafifah Suhengki dan Ahsanul Qalbi Fajar Islami sebagai teman yang selalu menemani penulis dalam penulisan dan penyusunan laporan Skripsi.
8. Tak lupa penulis ingin mengucapkan banyak terima kasih kepada pihak-pihak terkait lainnya yang telah banyak membantu baik untuk penyelesaian program serta laporan Skripsi.

Penulis juga ingin berterima kasih kepada para pembaca Skripsi ini dan mohon maaf jika ada kesalahan dalam penelitian serta penulisan Skripsi ini. Apabila, nantinya terdapat kesalahan dalam Skripsi ini, penulis sangat mengharapkan kritik dan sarannya.

Akhir kata, semoga Skripsi ini dapat memberikan banyak manfaat bagi pembaca baik sebagai sumber informasi, inspirasi, maupun referensi.

Tangerang, 15 Juni 2020

Ardhi Prago

**IMPLEMENTASI ALGORITMA GENETIKA**

**PADA PENJADWALAN MATA KULIAH**

**(STUDI KASUS: PROGRAM STUDI INFORMATIKA UNIVERSITAS MULTIMEDIA NUSANTARA)**

# ABSTRAK

Penjadwalan memiliki peran sebagai teknologi inti dalam sistem produksi. Di dalam dunia perkuliahan, terdapat sejumlah permasalahan yang berkaitan dengan penjadwalan, yaitu penjadwalan Kartu Rencana Studi (KRS), dosen, ruangan kelas dan lain-lain. Sebelum penjadwalan KRS, terdapat proses yaitu menyusun mata kuliah berdasarkan kelas yang dibuka dan pembagian dosen sesuai dengan mata kuliah yang di ajar. Dalam penjadwalan mata kuliah dan dosen, terdapat beberapa *constraint* untuk menentukan jadwal tersebut secara otomatis yang lebih optimal untuk digunakan oleh prodi Informatika Universitas Multimedia Nusantara. Berdasarkan hasil wawancara dengan ibu Nunik Afriliana selaku kepala prodi Informatika Universitas Multimedia Nusantara, selama proses penjadwalan yang dilakukan menjadi *time* *consuming* karena tidak adanya program atau *software* yang membuat penjadwalan secara otomatis sehingga dilakukan secara manual. Untuk itu dbuatlah sistem yang dapat melakukan *generate* jadwal secara otomatis. Penelitian ini melakukan pembahasan mengenai penerapan salah satu metode optimisasi, yaitu Algoritma Genetika yang digunakan untuk melakukan penyusunan penjadwalan mata kuliah secara langsung dengan jadwal yang bervariasi. Penjadwalan yang dilakukan terdiri dari penyusunan mata kuliah dan dosen berdasarkan jumlah kelas yang dibuka pada semester yang ingin dilakukan penyusunan. Penelitian ini menggunakan *dataset* berupa seluruh mata kuliah dan dosen yang mengajar berdasarkan jumlah kelas yang dibuka pada prodi Informatika dari BIA UMN. Pada penerapannya, penelitian ini menggunakan *list* dosen dan *list* *course* yang terdiri dari kelas yang dibuka pada seluruh mata kuliah dan dosen yang mengajar pada mata kuliah tersebut. Hasil dari pengujian yang paling ideal diperoleh pada pengujian semester 1 dengan pemetaan mata kuliah berdasarkan jumlah kelas yang dibuka sebanyak 36 dengan dosen ajar sebanyak 24 orang menggunakan jumlah populasi sebanyak 10 dengan *mutation rate* 1.0 menghasilkan rata-rata generasi adalah 4 generasi dengan rata-rata waktu 1,42 detik.

Kata Kunci: Penjadwalan, Algoritma Genetika, Kartu Rencana Studi (KRS)

**IMPLEMENTATION OF GENETIC ALGORITHM**

**IN COURSE SCHEDULING**

**(CASE STUDY: INFORMATICS STUDY PROGRAM MULTIMEDIA NUSANTARA UNIVERSITY)**

# ABSTRACT

Scheduling has a role as a core technology in production systems. In the world of lectures, there are some of problems related to scheduling, for example scheduling of Study Plan Cards (KRS), lecturers, classroom and others. Before scheduling KRS, there is a process arranging courses based on classes opened and lecturer division in accordance with the taught subjects. In scheduling courses and lecturers, there are several constraints to determine the schedule automatically which is more optimal for use by the Informatics Study Program at Multimedia Nusantara University. Based on the results of interviews with Ms. Nunik Afriliana as head of the Informatics Study Program at Multimedia Nusantara University, during the scheduling process it was time consuming because there were no programs or software that made scheduling automatically so it was done manually. For this reason, a system is created that can generate schedules automatically. This research discusses the application of one of the optimization methods, the Genetic Algorithm which is used to arrange the scheduling of courses directly with a varied schedule. Scheduling consists of arranging courses and lecturers based on the number of classes opened in the semester to be arranged. This study uses a dataset in the form of all courses and lecturers who teach in theses courses. The most ideal test results are obtained in semester 1 testing by mapping subjects based on the number of classes opened by 36 with 24 lecturers using a population of 10 with a mutation rate of 1.0 resulting in an average generation of 4 generation with an average time 1,42 seconds.

Keywords: Scheduling, Genetic Algorithm, Study Plan Cards (KRS)

# DAFTAR ISI

[HALAMAN PERSETUJUAN ii](#_Toc43068710)

[PERNYATAAN TIDAK MELAKUKAN PLAGIAT iii](#_Toc43068711)

[PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS iv](#_Toc43068712)

[HALAMAN PERSEMBAHAN v](#_Toc43068713)

[KATA PENGANTAR vi](#_Toc43068714)

[ABSTRAK viii](#_Toc43068715)

[ABSTRACT ix](#_Toc43068716)

[DAFTAR ISI x](#_Toc43068717)

[DAFTAR GAMBAR xii](#_Toc43068718)

[DAFTAR TABEL xiii](#_Toc43068719)

[BAB I 1](#_Toc43068720)

[1.1. Latar Belakang 1](#_Toc43068721)

[1.2. Rumusan Masalah 3](#_Toc43068722)

[1.3. Batasan Masalah 4](#_Toc43068723)

[1.4. Tujuan Penelitian 4](#_Toc43068724)

[1.5. Manfaat Penelitian 5](#_Toc43068725)

[1.6. Sistematika Penulisan 5](#_Toc43068726)

[BAB II 7](#_Toc43068727)

[2.1. Penjadwalan 7](#_Toc43068728)

[2.2 Algoritma Genetika 7](#_Toc43068729)

[BAB III 14](#_Toc43068730)

[3.1 Metodologi Penelitian 14](#_Toc43068731)

[3.2 Perancangan Sistem 16](#_Toc43068734)

[3.2.1 Flowchart 16](#_Toc43068735)

[BAB IV 25](#_Toc43068736)

[4.1. Spesifikasi Sistem 25](#_Toc43068740)

[4.2. Implementasi Algoritma 25](#_Toc43068741)

[4.3. Uji Coba Pengubahan nilai Populasi 35](#_Toc43068742)

[4.3.1. Uji Coba Pergantian Nilai Mutation Rate 35](#_Toc43068743)

[4.4. Evaluasi Hasil Uji Coba 38](#_Toc43068744)

[BAB V 41](#_Toc43068745)

[5.1. Simpulan 41](#_Toc43068747)

[5.2. Saran 42](#_Toc43068748)

[DAFTAR PUSTAKA 43](#_Toc43068749)

[DAFTAR LAMPIRAN 46](#_Toc43068750)

# DAFTAR GAMBAR

[Gambar 3. 1 Flowchart Utama 17](#_Toc43071758)

[Gambar 3. 2 Flowchart Pengolahan Data 18](#_Toc43071759)

[Gambar 3. 3 Flowchart Tahap Proses Pengolahan Data Dosen 19](#_Toc43071760)

[Gambar 3. 4 Flowchart Tahap Proses Pengolahan Data *Course* 20](#_Toc43071761)

[Gambar 3. 5 Flowchart Tahap Proses Pengolahan Tetapan Waktu 21](#_Toc43071762)

[Gambar 3. 6 Flowchart Tahap Proses Implementasi Algoritma Genetika 22](#_Toc43071763)

[Gambar 4. 1 Proses Membaca *Dataset...................................................................*26](#_Toc43071764)

[Gambar 4. 2 Hasil Proses Membaca *Dataset* 26](#_Toc43071765)

[Gambar 4. 3 Proses Menyimpan data ke *List Array* 27](#_Toc43071766)

[Gambar 4. 4 Lanjutan Proses Penyimpanan Data dari *Dataset* 28](#_Toc43071767)

[Gambar 4. 5 Lanjutan Proses Penyimpanan Data dari *Dataset* 28](#_Toc43071768)

[Gambar 4. 6 *List* Dosen pada *Current* Semester 29](#_Toc43071769)

[Gambar 4. 7 *List* *Course* pada *Current* Semester 29](#_Toc43071770)

[Gambar 4. 8 *List* *Meeting Time* 30](#_Toc43071771)

[Gambar 4. 9 Proses Inisialisasi Populasi Awal 31](#_Toc43071772)

[Gambar 4. 10 Hasil Inisialisasi Populasi Awal 31](#_Toc43071773)

[Gambar 4. 11 Proses *Crossover Schedule* 32](#_Toc43071774)

[Gambar 4. 12 Proses *Mutation Schedule* 32](#_Toc43071775)

[Gambar 4. 13 Proses Pemberian *Constraint* dan Nilai *Fitness* 33](#_Toc43071776)

[Gambar 4. 14 Hasil Penghitungan *Conflict* dan Pemberian Nilai *Fitness* 34](#_Toc43071777)

[Gambar 4. 15 Proses Pencari Jadwal Berdasarkan Nilai *Fitness* 34](#_Toc43071778)

[Gambar 4. 16 Hasil Penjadwalan 35](#_Toc43071779)

[Gambar 4. 17 *Set Population Size* 10 35](#_Toc43071780)

[Gambar 4. 18 Hasil Penjadwalan dengan *Mutation Rate* 0.1 dan *Population* 10 36](#_Toc43071781)

[Gambar 4. 19 *Set Population Size* 40 36](#_Toc43071782)

[Gambar 4. 20 Hasil Penjadwalan dengan *Mutation Rate* 0.1 dan *Population* 40 37](#_Toc43071783)

[Gambar 4. 21 *Set Population Size* 80 37](#_Toc43071784)

[Gambar 4. 22 Hasil Penjadwalan dengan *Mutation Rate* 0.1 dan *Population* 80 38](#_Toc43071785)

# DAFTAR TABEL

[Tabel 4. 1 Hasil Uji Coba Pengujian dengan *population size* 10,40,80 39](#_Toc43071790)

# BAB I

**PENDAHULUAN**

## Latar Belakang

Penjadwalan memiliki peran sebagai teknologi inti dalam sistem produksi. Semua industri membutuhkan penjadwalan yang tepat untuk pengaturan pengalokasian sumber daya agar sistem produksi berjalan dengan cepat dan tepat sehingga mendapatkan hasil produksi yang optimal (Pinedo, 2018). Penjadwalan merupakan pengalokasian waktu yang tersedia untuk melaksanakan masing-masing pekerjaan dalam rangka menyelesaikan suatu proyek hingga tercapai hasil optimal dengan mempertimbangkan keterbatasan-keterbatasan yang ada (Abrar Husen, 2009).

Di dalam dunia perkuliahan, terdapat sejumlah permasalahan yang berkaitan dengan penjadwalan, yaitu penjadwalan kartu rencana studi (KRS), dosen, ruang kelas dan lain-lain. Penjadwalan KRS adalah hal yang umum dilakukan saat mahasiswa melakukan pergantian semester. Mahasiswa akan menentukan mata kuliah apa saja yang akan diambil pada semester selanjutnya berdasarkan jadwal yang telah disusun sebelumnya.

Sebelum penjadwalan KRS, terdapat proses yaitu menyusun mata kuliah yang tersedia di Prodi Informatika Universitas Multimedia Nusantara, pembagian dosen yang sesuai dengan mata kuliah yang diajar. Penjadwalan mata kuliah yang diajar terdapat *constraint* yang menentukan jadwal tersebut agar mencapai optimal untuk

digunakan, misalkan dalam pembagian mata kuliah kepada dosen tidak boleh*overload,* mata kuliah dengan kode “IF” tidak dapat memiliki waktu yang bersamaan dengan mata kuliah kode “IF” lainnya, setiap mata kuliah yang diajar oleh dosen tidak bisa memiliki waktu yang sama, dan lainnya.

Berdasarkan hasil wawancara yang dilakukan dengan ibu Nunik Afriliana selaku kepala prodi Informatika Universitas Multimedia Nusantara dapat disimpulkan bahwa proses penyusunan jadwal perkuliahan untuk KRS secara garis besar dilakukan secara manual, belum adanya sistem yang dimiliki untuk membuat jadwal secara otomatis. Namun, terdapat *software* yang digunakan yaitu “*ASC time table”* yang bisa digunakan untuk membantu memudahkan dalam penyusunan jadwal. Selama proses penyusunan jadwal menjadi *time cosuming* karena tidak ada *software* atau program yang membuat penjadwalan secara otomatis dalam *generate* jadwal. Oleh karena itu, diperlukan program untuk proses penjadwalan tersebut agar dapat dilakukan secara efektif dan efisien.

Pada penelitian sebelumnya, implementasi algoritma Genetika untuk penjadwalan sidang magang telah berhasil dilakukan oleh Muhammad Arief Widyanto (2019). Penelitian tersebut dievaluasi dengan metode EUCS yaitu *content*, *accuracy,* format, *ease of use*, dan *timelines*. Hasil pengujian dari penelitian tersebut menghasilkan rata-rata iterasi 1,26 kali dengan waktu yang dibutuhkan pengujian ini 3,6 detik dengan rata-rata 0,24 detik. Hasil akurasi 100% karena tidak ada batasan yang dilanggar.

Komang Setemen (2010) melakukan implementasi algoritma Genetika pada pengembangan sistem aplikasi penjadwalan kuliah. Pada penelitian tersebut dilakukan pengujian sebanyak tiga kali. Pada pengujian pertama dilakukan pada 53 jenis pemetaan mata kuliah dengan rincian jumlah ruangan maksimal yang bisa digunakan adalah sebanyak 14 buah termasuk ruang kuliah biasa dan laboratorium, jumlah dosen pengampu mata kuliah 26 orang. Pada pengujian kedua dilakukan pada 88 jenis pemetaan mata kuliah dengan rincian jumlah mata kuliah ada 58 dan ruangan maksimal sebanyak 14 buah dan jumlah dosen pengampu sebanyak 28 orang. Pada pengujian ketiga dilakukan pada 141 jenis pemetaan mata kuliah untuk semua jurusan FTK, dengan rincian mata kuliah ada 101 buah, ruangan maksimal sebanyak 14 buah dan dosen pengampu mata kuliah 49 orang. Ketiga pengujian tersebut memperoleh hasil bahwa algoritma Genetika mampu meyelesaikan pembuatan jadwal dengan tidak ada bentrokan sama sekali dengan nilai *fitness* sama dengan 0.

Berdasarkan penelitian sebelumnya, penelitian ini akan mengimplementasikan algoritma Genetika untuk aplikasi penjadwalan dosen dan pembagian kelas pada sistem KRS Program Studi Informatika UMN. Penelitian ini akan dibangun dengan bahasa pemrograman Python. Penelitian ini diukur dengan melihat hasil jadwal yang berhasil di *generate*.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, berikut rumusan masalah dalam penelitian yang diadakan:

1. Bagaimana mengimplementasikan algoritma Genetika pada penjadwalan mata kuliah khususnya prodi Informatika?
2. Apakah algoritma Genetika mampu menyelesaikan permasalahan penjadwalan mata kuliah pada prodi Informatika?
3. Berapa banyak iterasi yang dilakukan dan seberapa besar tingkat akurasi yang menghasilkan jadwal mata kuliah tanpa ada batasan yang dilanggar?

## Batasan Masalah

Batasan masalah yang diperhatikan dalam penelitian ini dapat dijabarkan dalam beberapa poin sebagai berikut:

1. Penelitian penjadwalan KRS mata kuliah dan dosen khususnya pada prodi Informatika Universitas Multimedia Nusantara
2. Data yang digunakan untuk penelitian ini berupa *list* dosen, *list* kelas yang sudah tersedia beserta waktu mengajar
3. Penjadwalan tidak untuk sistem pemaketan mata kuliah sehingga tidak melihat distribusi semester pembukaan mata kuliah

## Tujuan Penelitian

Penelitian ini memiliki tujuan sebagai berikut.

1. Menimplementasikan algoritma Genetika terhadap penjadwalan mata kuliah khususnya pada prodi Informatika
2. Membuktikan bahwa dengan algoritma Genetika dapat digunakan untuk melakukan penjadwalan mata kuliah pada prodi Informatika di sistem KRS
3. Mendapatkan jumlah iterasi dan tingkat akurasi dari penjadwalan mata kuliah yang dihasilkan tanpa adanya batasan pelanggaran

## Manfaat Penelitian

Melalui penelitian ini diharapkan dapat memberikan manfaat sebagai berikut.

1. Menambah wawasan pengetahuan dan pengalaman dalam menimplementasikan algoritma Genetika untuk penjadwalan mata kuliah
2. Menyediakan solusi penjadwalan KRS prodi Informatika UMN secara terkomputerisasi
3. Menunjukan kinerja algoritma Genetika untuk permasalahan penjadwalan

## Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyajian laporan skripsi ini adalah sebagai berikut.

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, serta sistematika penulisan.

1. BAB II LANDASAN TEORI

Bab ini berisi tentang teori-teori dan konsep dasar yang mendukung penelitian terkait permasalahan yang dibahas, seperti Sistem Penjadwalan dan Algoritma Genetika.

1. BAB III METODE DAN PERANCANGAN APLIKASI

Bab ini berisi tentang metode penelitian dan perancangan sistem yang terdiri dari flowchart dan struktur tabel sistem yang dibangun.

1. BAB IV IMPLEMENTASI HASIL PENELITIAN

Bab ini berisi tentang hasil implementasi Algoritma Genetika pada sistem penjadwalan yang menghasilkan jadwal mata kuliah dan dosen untuk seluruh kelas yang dibuka per semester berdasarkan *constraint* yang berlaku.

1. BAB V SIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan hasil analisa sistem dalam rangka menjawab tujuan penelitian yang diajukan, serta saran-saran yang penulis berikan untuk pengembangan penelitian lebih lanjut.

# BAB II

**LANDASAN TEORI**

## 2.1. Penjadwalan

Penjadwalan merupakan sebuah proses pengambilan keputusan yang sering dilakukan di industri manufaktur maupun di industri yang bergerak di bidang pelayanan atau jasa (Pinedo, 2012). Dalam penjadwalan KRS terdapat dua kategori batasan yaitu batasan yang bersifat mutlak dan batasan yang bersifat preferensi atau disebut batasan lunak. Batasan yang bersifat mutlak yang wajib dipenuhi, yaitu:

1. Seorang dosen tidak dapat mengajar lebih dari satu mata kuliah di dua kelas yang berbeda dalam waktu yang sama
2. Untuk seluruh kelas dalam satu semester, jika memiliki kode kelas yang sama tidak boleh bertabrakan

Sedangkan batasan lunak yang bersifat preferensi merupakan batasan yang boleh dilanggar jika perlu, yaitu:

1. Waktu jeda antar mata kuliah
2. Dosen sebaiknya memiliki hari libur mengajar

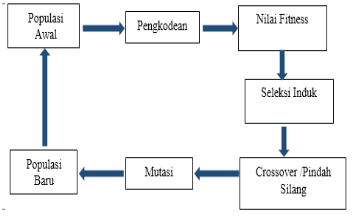
## 2.2 Algoritma Genetika

Algoritma Genetika merupakan salah satu algoritma optimasi yang cukup handal dan sering dipakai dalam permasalahan penjadwalan. Algoritma Genetika adalah algoritma yang mengambil sekumpulan individu dari sebuah populasi dan akan terus menerus mencari individu terbaik dengan menghasilkan generasi baru secara iterative (Haldurai, 2016)

Menurut Stuart (2010) terdapat beberapa hal penting yang digunakan untuk membangun penyelesaian permasalahan dengan algoritma Genetika, yaitu:

1. *Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma Genetika, gen ini bisa berupa nilai biner, *float*, *integer* maupun karakter atau kombinatorial
2. *Allele*, nilai dari gen
3. Kromosom, gabungan gen-gen yang membentuk nilai tertentu
4. Individu, menyatakan satu nilai keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi
6. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam algoritma Genetika

Adapun tahap algoritma Genetika (Jollyta, Johan & Hajjah, 2017) adalah sebagai berikut.

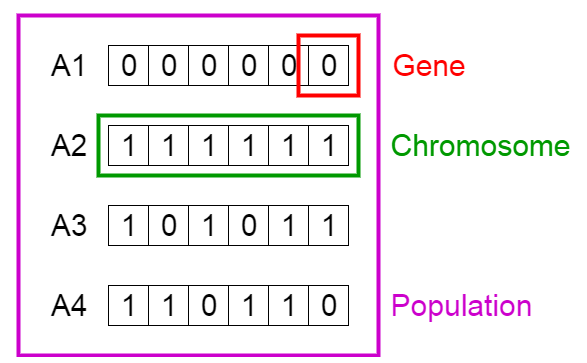


Gambar 6.1 Tahapan Algoritma Genetika

Tujuan dari algoritma Genetika adalah mencari nilai *fitness value* dari individu di suatu populasi. Terdapat 5 fase dalam algoritma Genetika, yaitu sebagai berikut

1. Inisialiasi Populasi

Proses dimulai dengan menginisialisasi beberapa individu atau disebut dengan populasi. Setiap individu adalah suatu solusi atau *fitness value* yang ini dicari. Setiap individu merupakan sekumpulan dari *Genes* atau gen atau disebut dengan *chromosome* (kromosom). Dalam algoritma Genetika, kumpulan gen individu diwakili menggunakan *string* dalam alfabet. Nilai gen tersebut direpresentasikan ke dalam bentuk kode biner.



Gambar 6.2 Gambar Gen, Chromosome dan Populasi

1. *Fitness Function*

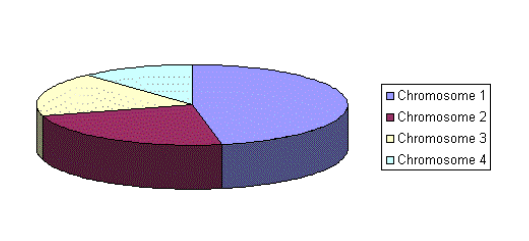
Nilai *fitness* menentukan kemampuan individu yang diambil untuk dapat bersaing dengan individu yang lain. Probabilitas individu yang dipilih adalah berdasarkan dari nilai *fitness*. Algoritma Genetika dapat ditunjukkan pada persamaan Pesamaan 1 dan Persamaan 2 dibawah ini (Mahmudy, 2013)

(1)

Dengan Fx adalah (1.0\*jumlah *conflict*) + 1

1. *Selection*

Pada tahap *selection* adalah memilih individu mana yang paling cocok dan invidu tersebut yang dipilih untuk melanjutkan generasi berikutnya. Pada penelitian ini metode *selection* yang digunakan adalah *Roulette Wheel Selection*.

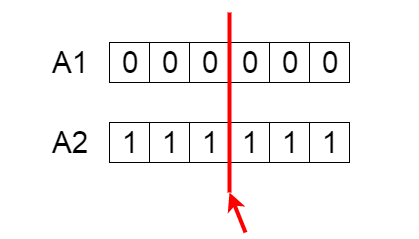


Gambar 6.3 Gambar *Roulette Wheel Selection*

Pada metode *Roulette Wheel Selection* ini sebanding dengan nilai *fitness* dari *chromosome*. Semakin tinggi nilai *fitness* dari *chromosome* semakin tinggi peluang untuk dipilih.

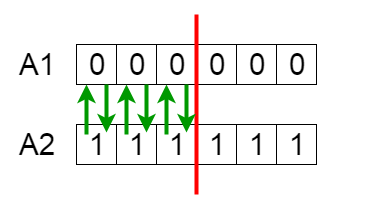
1. *Crossover*

Pada tahap ini untuk setiap pasangan invidu kemudian disilangkan untuk membentuk individu baru. Persilangan antar individu dilakukan dengan menentukan nilai *crossover point* secara acak didalam kromosom.



Gambar 6.4 *Crossover point*

Individu baru tercipta dengan menukar gen diantara invidividu sampai titik *crossover point*.



Gambar 6.5 Persilangan antar individu

1. *Mutation*

Setelah keturunan invidu baru, beberapa gen pada setiap individu dilakukan mutasi dengan *probability* yang rendah. Mutasi berfungsi untuk mengganti gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal (Zukhri, 2004).

Algoritma akan berhenti ketika populasi telah konvergen, tidak lagi memproduksi keturunan yang signifikan dari generasi sebelumnya. Pada tahap ini jadwal mata kuliah yang diampu oleh dosen telah berhasil diselesaikan dengan algoritma Genetika.

Keunggulan algoritma Genetika dalam proses penjadwalan adalah (Ahmat Josi, 2017)

* Algoritma Genetika dapat membantu mempercepat waktu proses penjadwalan
* Algoritma Genetika dapat diterapkan dalam berbagai bahasa pemrograman
* Algoritma Genetika dapat mengatasi bentrok jadwal sehingga penjadwalan menjadi lebih efektif

# BAB III

**METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM**

## Metodologi Penelitian

Penelitian “Implementasi Algoritma Genetika pada Penjadwalan Mata Kuliah (Studi Kasus: Program Studi Informatika Universitas Multimedia Nusantara)” menggunakan beberapa taha, tahap-tahap yang dilaksanakan antara lain adalah sebagai berikut

* + 1. Wawancara

Pada tahap ini, peneliti melakukan wawancara dengan ibu Nunik Afriliana selaku kepala program studi Informatika Universitas Multimedia Nusantara. Berdasarkan.hasil wawancara dapat disimpulkan bahwa proses penyusunan jadwal perkuliahan untuk Kartu Rencana Studi (KRS) secara garis besar dilakukan secara manual. Penyusunan secara manual tersebut menjadi *time consuming* karena tidak adanya *software* atau program yang membuat penjadwalan secara otomatis. Oleh karena itu, diperlukan program untuk proses penjadwalan tersebut agar dapat dilakukan secara efektif dan efisien.

* + 1. Studi Literatur

Tahap Studi Literatur dilakukan dengan mencari, membaca dan mempelajari sumber dari jurnal ilmiah dan karya tulis terkait dengan topik yang diteliti. Tahap studi literatur bertujuan untuk memahami teori mengenai Algoritma Gentika dan *constraint* yang terhubung dalam penjadwalan.

* + 1. Pengumpulan Data

Data yang digunakan dalam penelitian ini didapatkan dari Biro Informasi Akademik Universitas Multimedia Nusantara (BIA UMN). Data yang didapatkan berupa jadwal referensi/jadwal acuan yang berisikan *list* matakuliah, *list* dosen, *list* ruangan, dan waktu mengajar dosen. Pada penelitian ini data yang digunakan dari data tersebut adalah *list* dosen dan *list* matakuliah berdasakarn jumlah kelas yang dibuka.

* + 1. Perancangan dan Pengembangan Program

Pada tahap ini perancangan program terdiri atas pembuatan alur kerja untuk mengambil data dari dataset lalu melakukan implementasi pada Algoritma Genetika dengan tahapan melakukan inisialisasi populasi awal, seleksi, *crossover*, *mutation*, penghitungan nilai *fitness* dan memilih jadwal yang sesuai dengan *constraint* yang berlaku.

* + 1. Pengujian Aplikasi

Pada tahap ini bertujuan untuk mengukur pengaruh jumlah populasi terhadap kecepatan proses pencarian jadwal.

* + 1. Penulisan Laporan

Tahap terakhir dalam penelitian ini adalah menulis laporan yang berisikan hasil keseluruhan penelitian. Penulisan dilakukan secara terstruktur sesuai dengan kaidah penulisan laporan ilmiah. Penulisan laporan dilakukan secara paralel sembari melakukan penelitian, sehingga proses dapat berjalan sesuai rencana dan tidak ada yang terlewatkan dalam penulisan laporan. Penulisan laporan juga dilakukan secara bertahap, dimulai dari pendahuluan, latar belakang, kesimpulan, dan saran.

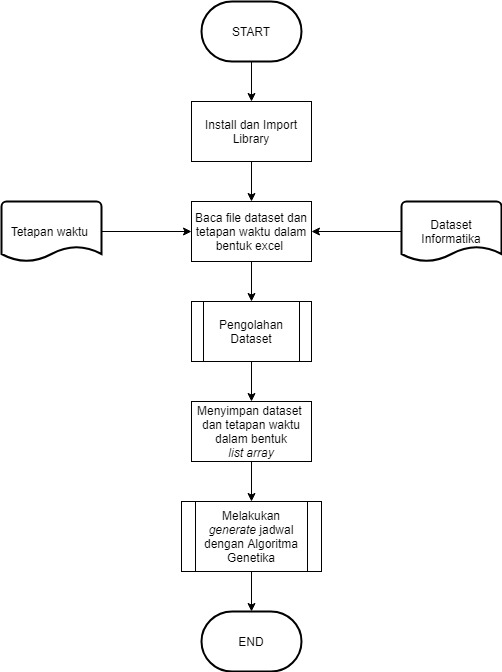


## Perancangan Sistem

Perancangan sistem dilakukan dengan merancang *flowchart* yang menjabarkan tahapan dalam penerapan Algoritma Genetika yang dimulai dari pengambilan dan pengolahan data dari dataset, serta melakukan proses penjadwalan dengan menggunakan Algoritma Genetika.

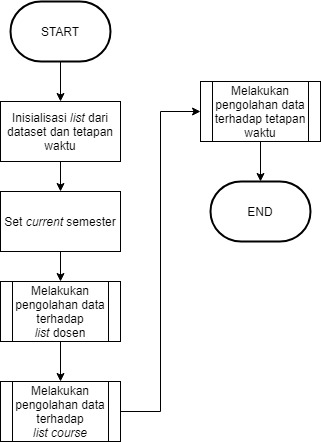
### 3.2.1 Flowchart

Proses yang ada dalam sistem dirancang berdasarkan *flowchart.* Dimulai dari *flowchart* Utama, *flowhcart* Pengolahan data (dosen dan *course*) hingga *flowchart* implementasi Algoritma Genetika. Gambar 3.1 adalah *flowchart* atau diagram alir dari proses pengolahan dataset dan melakukan *generate* jadwal menggunakan Algoritma Genetika yang digambarkan secara umum. Proses pertama yang dilakukan adalah *install* dan *import* *library* yang dibutuhkan seperti *time* yang digunakan untuk mengukur waktu proses program menghasilkan jadwal, *prettytable* untuk menampilkan visualisasi data menjadi lebih rapih sehingga mudah untuk dibaca, *random* digunakan untuk melakukan inisialisasi populasi awal secara random, *pandas* digunakan untuk membaca file *dataset* dan tetapan waktu. Kemudian, *dataset* dalam bentuk *file* *excel* yang telah diolah sebelumnya akan disimpan ke dalam *list* untuk melanjutkan proses berikutnya. *Dataset* yang digunakan adalah berupa *list* seluruh dosen dan mata kuliah pada prodi Informatika Universitas Multimedia Nusantara.

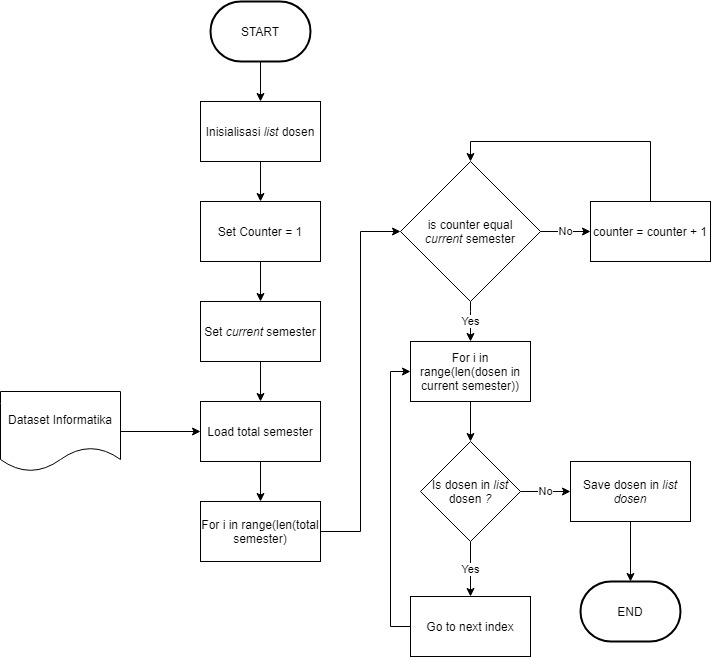


Gambar 3. 1 Flowchart Utama

*Dataset* yang telah dibaca terdiri dari *list* mata kuliah, *list* dosen, *list* kelas yang dibuka dan *list* tetapan waktu pada prodi Informatika. Lalu tahap selanjutnya adalah melakukan pengolahan pada *dataset* Informatika. Dalam proses pengolahan data, ada beberapa tahapan yang dilakukan. Pada Gambar 3.2 Tahap-tahap yang dilakukan adalah dimulai dengan inisialisasi *list* yang digunakan untuk menampung dan menyimpan data dari *list* dataset Informatika dan tetapan waktu, selanjutnya adalah menentukan semester berapa yang ini di *generate* yang disimpan dalam *current* semester. Berikutnya melakukan pengolahan pada data tersebut. Dalam mengolah data *course* diperlukan untuk melihat berapa banyak kelas yang dibuka dari setiap mata kuliah dan menggabungkan dengan dosen yang mengajar pada setiap mata kuliah. Dan pada tahap selanjutnya adalah melakukan pengolahan data terhadap tetapan waktu.

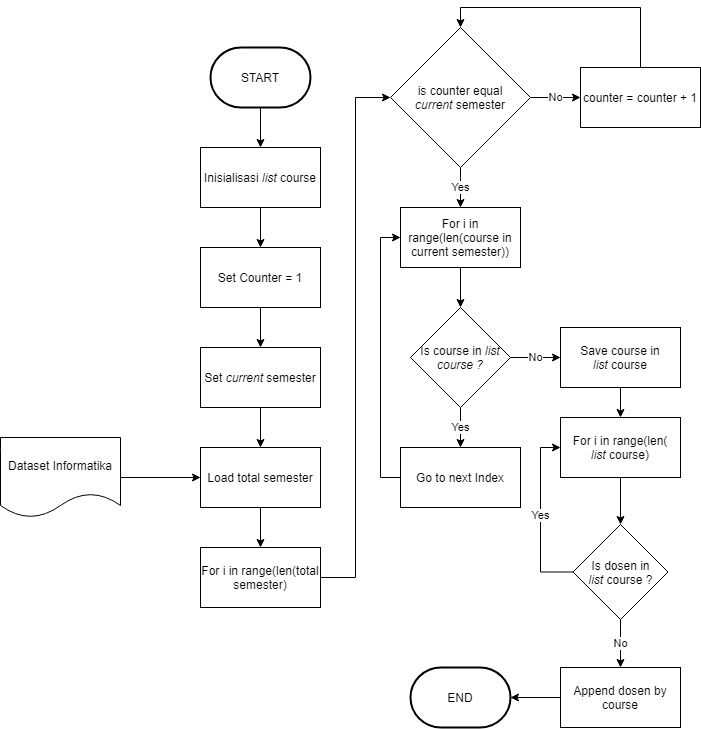


Gambar 3. 2 Flowchart Pengolahan Data



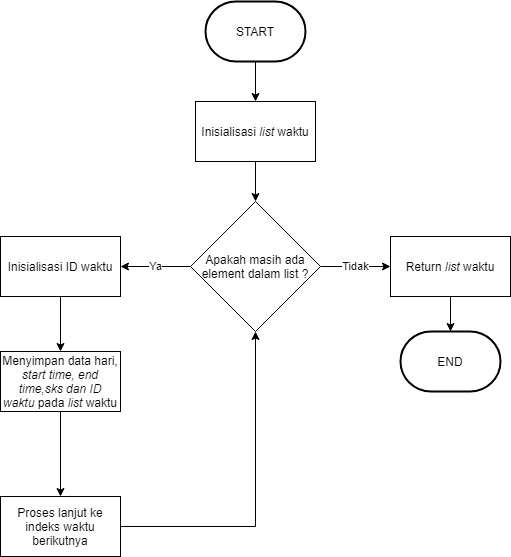
Gambar 3. 3 Flowchart Tahap Proses Pengolahan Data Dosen

Pada Gambar 3.3 yang merupakan tahapan dalam proses pengolahan data dosen dari *dataset* Informatika. Pada tahap pertama adalah inisialisasi *list* dosen. Selanjutnya, *set* *counter* = 1 dan *current* semester adalah semester berapa yang ingin dilakukan penjadwalan. Selanjutnya adalah melakukan *load* semua semester dari *dataset* Informatika. *Counter* digunakan untuk perulangan untuk mendapatkan semester yang sesuai dengan *current* semester. Selanjutnya untuk setiap dosen yang mengajar pada semester tersebut akan disimpan di dalam *list* dosen.



Gambar 3. 4 Flowchart Tahap Proses Pengolahan Data *Course*

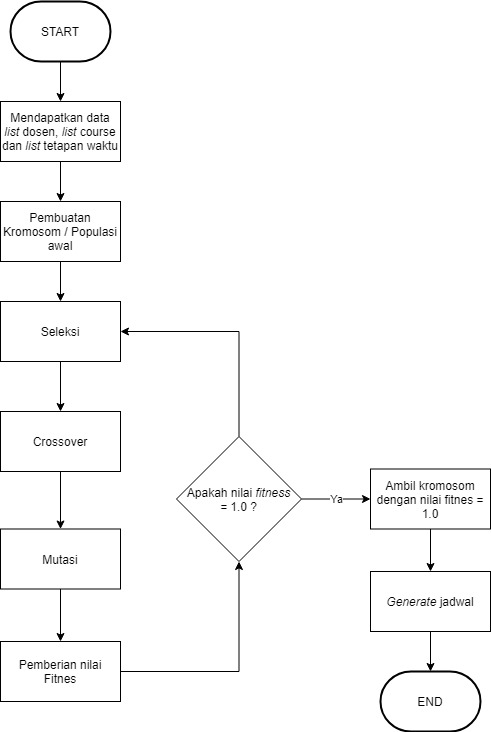
Pada tahap selanjutnya adalah membentuk *list* *course* yang dimana isinya adalah data seluruh kode mata kuliah, data seluruh mata kuliah, data dosen yang mengajar pada mata kuliah tersebut dan data kelas yang dibuka pada *current* semester berdasarkan *dataset* Informatika. Pada Gambar 3.4 langkah pertama adalah menginisialisasi *list* *course* yang digunakan untuk menyimpan data mata kuliah dan data dosen yang sesuai dengan mata kuliah yang diajar. Selanjutnya adalah melakukan *load* data semua semester. *Counter* yang ada disini sama fungsinya dengan yang ada pada pengolahan data dosen yaitu untuk melakukan perulangan sehingga mendapatkan semester yang sesuai dengan *current* semester. Selanjutnya untuk setiap *course* yang ada pada semester tersebut disimpan terlebih dahulu ke dalam *list* course. Selanjutnya untuk setiap *course* akan digabungkan dengan dosen yang ada pada semester tersebut. Penggabungan dosen sesuai dengan mata kuliah yang diajarkan.



Gambar 3. 5 Flowchart Tahap Proses Pengolahan Tetapan Waktu

Selain mendapatkan data dosen dan mata kuliah, penjadwalan membutuhkan waktu untuk menyusun jadwal tersebut agar tidak saling bertabrakan. Pada penelitian ini waktu dalam penjadwalan menggunakan tetapan waktu. Pada Gambar 3.5 merupakan tahapan proses pengambilan data pada tetapan waktu. Langkah pertama adalah menginisialisasi *list* *array* waktu yang digunakan untuk menampung data waktu dari *dataset*. Jika terdapat data dari tetapan waktu maka selanjutnya akan menginisialisasi ID waktu. Semua data dari *dataset* akan disimpan pada *list* waktu. Data yang disimpan dalam *list* waktu yaitu berupa, data Hari, *start time* yaitu waktu mulai, *end time* yaitu waktu selesai, dan sks.

Selanjutnya setelah mendapatkan data *list* dosen, *list* *course* dan tetapan waktu maka dilanjutkan dengan mengimplementasikan Algoritma Genetika untuk penjadwalan.



Gambar 3. 6 Flowchart Tahap Proses Implementasi Algoritma Genetika

Pada Gambar 3.6 setelah mendapatkan data *list* dosen, *list* *course* dan *list* tetapan waktu selanjutnya melakukan implementasi Algoritma Genetika untuk menghasilkan penjadwalan berdasarkan semester yang di inginkan. Langkah pertama adalah membuat kromosom atau populasi awal. Dalam menentukan populasi awal, data dosen, mata kuliah dan *meeting time* diambil secara acak. Pada inisialisasi populasi awal untuk data *meeting time* diambil waktu berdasarkan jumlah sks yang sesuai dengan sks pada mata kuliah. Setelah mendapatkan populasi awal, selanjutnya adalah melakukan seleksi. Pada proses ini akan memilih jadwal mana yang cocok berdasarkan inisialisasi pertama kali untuk melanjutkan generasi berikutnya. Seleksi dilakukan sebesar 33% dari jumlah *population size*. Setelah mendapatkan individu baru maka selanjutnya dilakukan penyilangan antar individu atau *crossover.* Pada tahap *crossover* setiap pasang individu kemudian disilangkan untuk membentuk individu baru. Persilangan antar individu dilakukan dengan menentukan nilai *crossover point* secara acak di dalam kromosom. Setelah persilangan menghasilkan keturunan individu baru. Selanjutnya adalah tahap Mutasi. Pada tahap ini berfungsi untuk mengganti gen yang hilang dari populasi selama proses selesi serta menyediakan gen yang tidak ada dalam populasi awal. Pembuatan jadwal mata kuliah dan dosen memiliki aturan berupa *constraint* yang berlaku. Selanjutnya adalah pemberian nilai *fitness* berdasarkan persamaan berikut:

(1)

Dengan Fx adalah (1.0\*jumlah *conflict*) + 1.

Selanjutnya adalah mengecek berapa nilai *fitness* dan ada berapa banyak *conflict* yang ada pada setiap jadwal dalam setiap populasi tersebut. Jika nilai *fitness* tidak sama dengan 1.0 maka akan melakukan kembali proses seleksi antar individu baru tersebut. Apabila nilai *fitness* adalah 1.0 yang dimana artinya tidak ada batasan yang dilanggar untuk setiap jadwal yang terbentuk sehingga tidak ada nilai *conflict* maka kromosom tersebut akan diambil. Lalu data kromosom tersebut akan digunakan untuk *generate* jadwal yang menghasilkan jadwal per semester berupa *list* mata kuliah serta kelas yang dibuka pada semester tersebut, dosen yang mengajar untuk setiap mata kuliah beserta waktu mengajar dosen pada masing-masing kelas.

# BAB IV

**IMPLEMENTASI DAN ANALISIS**



## Spesifikasi Sistem

Pada pelaksanaan penelitian ini digunakan beberapa tool atau alat untuk mendukung pelaksanaan penelitian baik itu perangkat keras (*hardware*) maupun perangkat lunak (*software*). Berikut merupakan komponen perangkat keras yang digunakan selama pengerjaan penelitian adalah

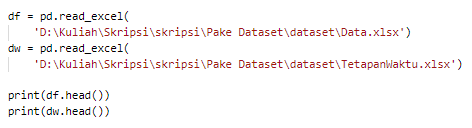
1. Laptop: ASUS X450JB
2. Processor: Intel Core i7-4720HQ
3. Graphic Processing Unit: Nvidia GeForce 940M
4. Memory: 12 GB RAM

Adapun *software* yang digunakan selama penelitian antara lain adalah

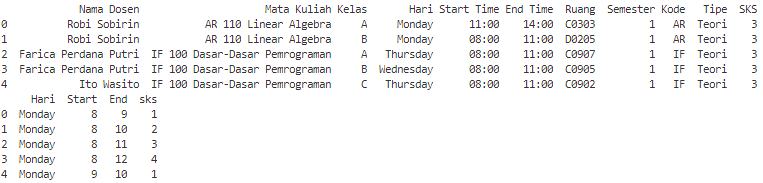
1. Python 3.7.1
2. Visual Studio Code
3. Microsoft Excel

## Implementasi Algoritma

Pada Gambar 4.1 menunjukkan proses pengambilan data *list* dosen, *list* *course* dan tetapan waktu dari file *Excel* untuk *dataset* Informatika dan Tetapan waktu. Gambar 4.2 menunjukkan dua hasil pembacaan dataset Informatika dan tetapan waktu. Pada *dataset* Informatika data yang digunakan untuk penelitian ini adalah kolom nama dosen, mata kuliah, kelas, semester, dan sks. Sedangkan untuk tetapan waktu terdapat kolom hari, *start time, end time* dan sks.



Gambar 4. 1 Proses Membaca *Dataset*

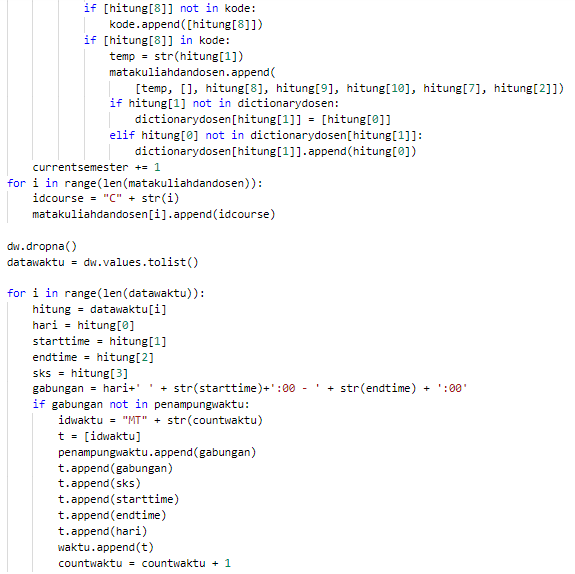


Gambar 4. 2 Hasil Proses Membaca *Dataset*

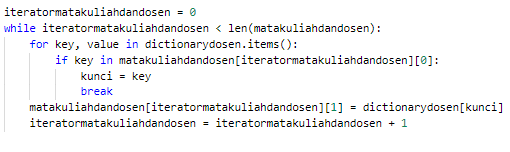
Pada Gambar 4.3, 4.5 dan 4.6 merupakan proses penyimpanan data *list* dosen, *list* *course* dan *list* tetapan waktu. Pada proses data *list* dosen data yang diambil dari *dataset* Informatika adalah berupa seluruh nama dosen yang ada pada *current* semester. Pada *list* course dapat dilihat pada *variable* “matakuliahdandosen” yang berisi list seluruh mata kuliah pada *current* semester dan dosen yang mengajar pada mata kuliah tersebut. Tetapan waktu disimpan dalam *list* waktu yang terdiri dari hari, *start* *time*, *end time* dan sks.



Gambar 4. 3 Proses Menyimpan data ke *List Array*

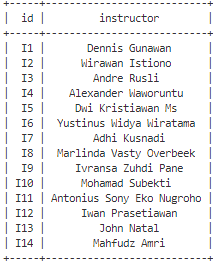


Gambar 4. 4 Lanjutan Proses Penyimpanan Data dari *Dataset*

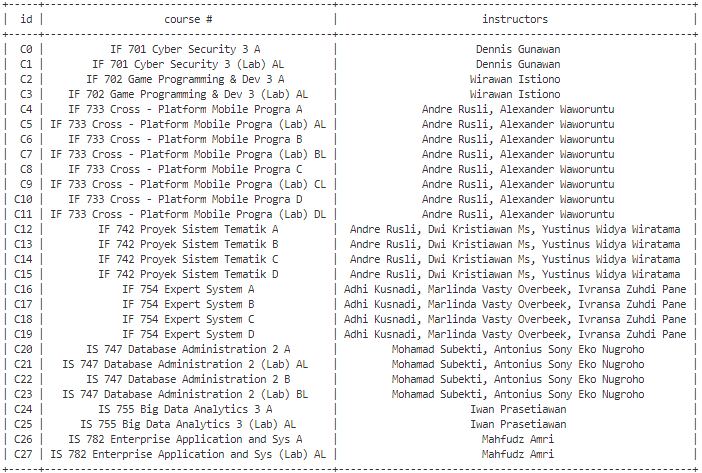


Gambar 4. 5 Lanjutan Proses Penyimpanan Data dari *Dataset*

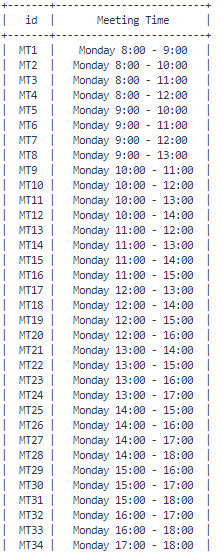
Pada Gambar 4.6 merupakan hasil dari *list* dosen yang diambil dari *dataset* Informatika. *List* dosen terdiri dari id dan nama dosen. Untuk hasil *list* *course* dapat dilihat pada Gambar 4.7 yang terdiri dari kolom id, nama mata kuliah beserta kelas yang dibuka dan dosen yang mengajar. Tetapan waktu pada Gambar 4.8 merupakan salah satu contoh hari Senin dengan kemungkinan *meeting time* yang dilaksanakan dalam satu hari. Penetepan waktu dilakukan dengan ketentuan seperti tidak ada penjadwalan pada hari minggu dan tidak ada jadwal *meeting* *time* pada pertengahan jam 12 di hari Jumat.



Gambar 4. 6 *List* Dosen pada *Current* Semester

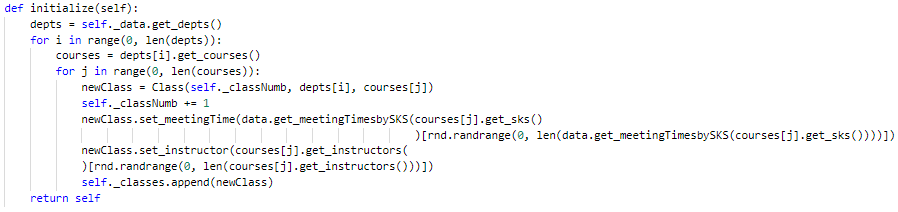


Gambar 4. 7 *List* *Course* pada *Current* Semester

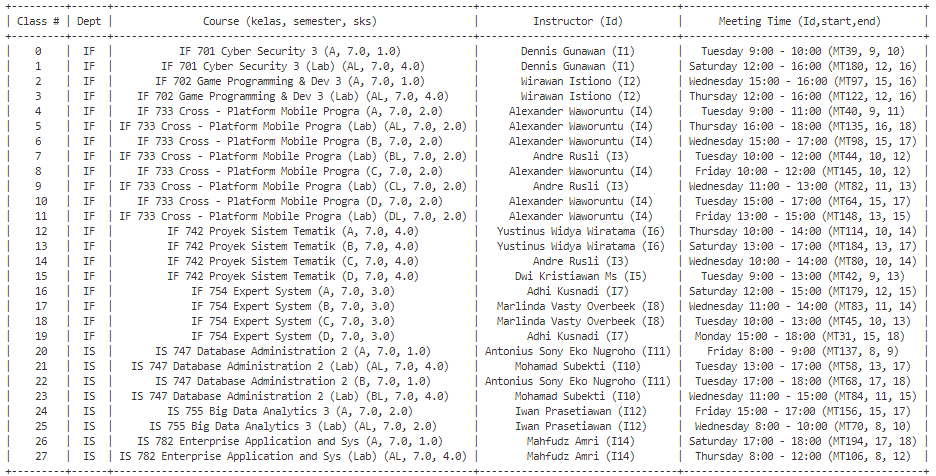


Gambar 4. 8 *List* *Meeting Time*

Gambar 4.9 merupakan langkah pertama pada Algoritma Genetika yaitu inisialisasi kromosom atau populasi awal. Inisialisasi Populasi awal dilakukan dengan dengan membentuk jadwal random sebanyak jumlah kelas yang dibuka pada *current* semester. Pada kondisi *meeting time* random tidak dilakukan ke semua data, melainkan merandom *meeting* *time* dengan ketentuan *random* sesuai dengan jumlah sks yang ada pada mata kuliah tersebut. Misalkan, mata kuliah *machine* *learning* sebanyak 3 sks maka akan merandom semua kemungkinan *list* *meeting time* dengan sks sebanyak 3. Gambar 4.10 merupakan hasil jadwal dari inisialisasi populasi awal.

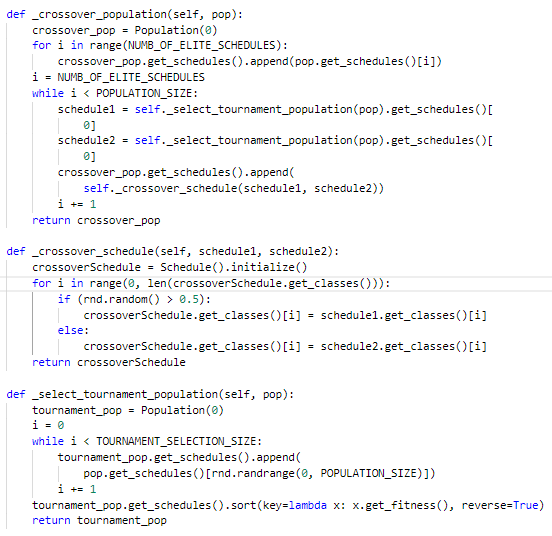


Gambar 4. 9 Proses Inisialisasi Populasi Awal

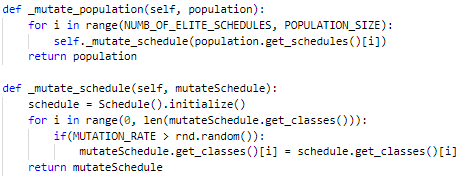


Gambar 4. 10 Hasil Inisialisasi Populasi Awal

Setelah mendapatkan Populasi Awal, dilaksanakan proses seleksi. Pada Gambar 4.11 merupakan proses seleksi yang kemudian dilanjutkan dengan proses *crossover*. Seleksi dilakukan dengan *tournament selection* = 33 %\* *population size*. Setelah melakukan seleksi, selanjutnya adalah proses *crossover* antar kelas yang terpilih dari proses seleksi. *Crossover* jadwal terjadi pada \_*crossover\_schedule* yaitu penyilangan antara kelas. Proses *crossover schedule* dilakukan dengan menentukan nilai *crossover point* secara acak yang dilakukan sebanyak jumlah berapa banyak populasi yang terbentuk. Pada penelitian ini jumlah populasi di set sebanyak 10.



Gambar 4. 11 Proses *Crossover Schedule*



Gambar 4. 12 Proses *Mutation Schedule*

Gambar 4.12 merupakan kelanjutan proses yaitu mutasi. Proses mutasi antar jadwal dilakukan pada *\_mutate\_schedule*. Mutasi antar individu dilakukan dengan *probability* yang rendah yaitu dengan *MUTATION\_RATE* di tetapkan 0.1 yang terbentuk dengan *NUM\_OF\_ELITE\_SCHEDULES* adalah 1.

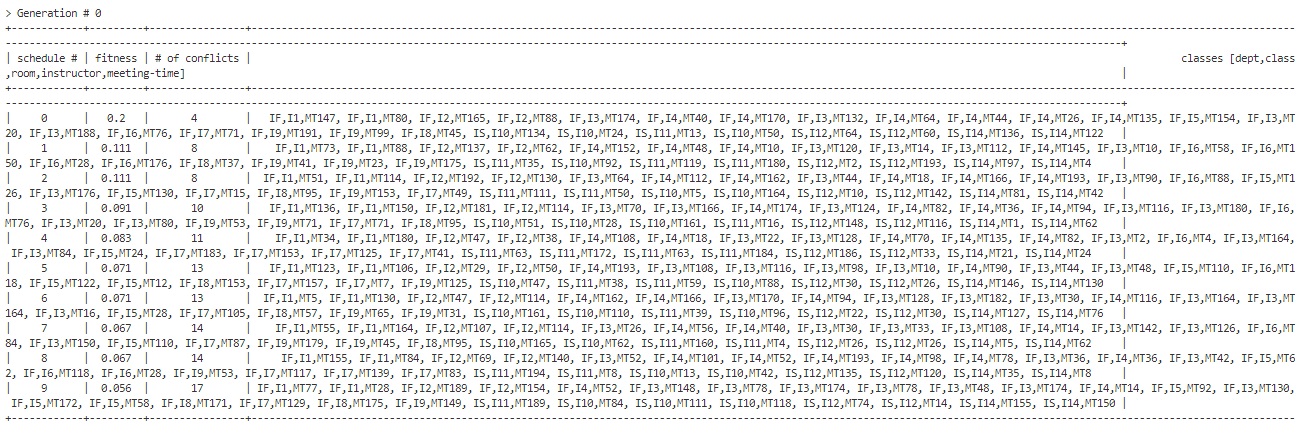
Gambar 4.13 merupakan proses pemberian *constraint* terhadap jadwal yang terbentuk. *Constraint* adalah satu dosen tidak dapat mengajar dua kelas yang berbeda di waktu yang bersamaan atau beririsan dan seluruh mata kuliah dalam *current* semester jika memiliki kode kelas yang sama tidak boleh saling bentrok. Apabila *constraint-constraint* tersebut dilanggar maka akan mendapatkan *penalty* berupa konflik untuk setiap batasan yang dilanggar. Pemberian nilai fitnes untuk setiap populasi jadwal dilakukan dengan rumus:

(1)

Dengan Fx adalah (1.0\*jumlah *conflict*) + 1.

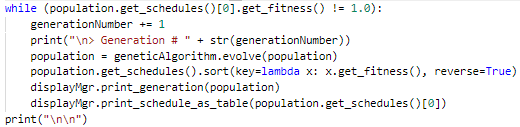


Gambar 4. 13 Proses Pemberian *Constraint* dan Nilai *Fitness*

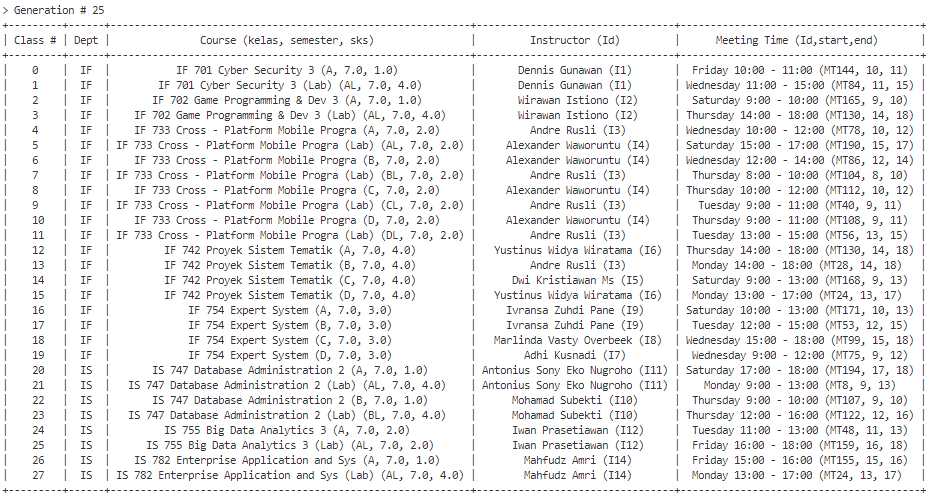


Gambar 4. 14 Hasil Penghitungan *Conflict* dan Pemberian Nilai *Fitness*

Gambar 4.14 merupakan hasil penghitungan dari jadwal yang terbentuk pada generasi pertama. Selanjutnya adalah melakukan seleksi terhadap jadwal yang memiliki nilai *fitness* adalah 1.0 dapat dilihat pada gambar 4.15. Apabila dalam generasi yang terbentuk tidak terdapat nilai *fitness* adalah 1.0, maka akan dilakukan pengulangan proses kembali ke *crossover* dari jadwal terakhir yang tergenerasi. Jika terdapat jadwal yang memiliki nilai *fitness* adalah 1.0, maka Algoritma Genetika akan berhenti dan memilih jadwal tersebut sebagai jadwal yang paling optimal. Hasil Penjadwalan dapat dilihat pada Gambar 4.16



Gambar 4. 15 Proses Pencari Jadwal Berdasarkan Nilai *Fitness*



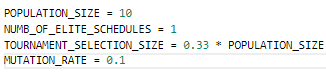
Gambar 4. 16 Hasil Penjadwalan

## Uji Coba Pengubahan nilai Populasi

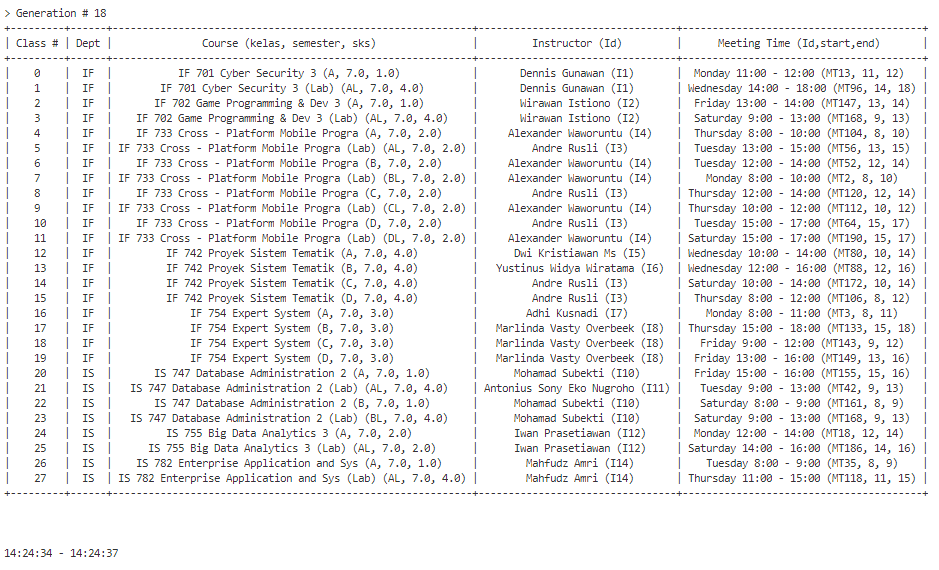
Pada penelitian ini dilakukan dengan mengubah nilai *population size.* Populasi adalah proses dimulai dengan menginisialisasi beberapa individu. Pada penelitian ini akan melakukan uji coba sebanyak 10 kali untuk tiap *current* semester yang dihasilkan.

### 4.3.1. Uji Coba Pergantian Nilai Mutation Rate

Skenari pertama adalah menggunakan nilai *population* *size* = 10. Gambar 4.17 adalah inisialisasi awal untuk nilai *population size* dan nilai *mutation* rate.



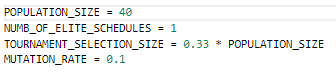
Gambar 4. 17 *Set Population Size* 10



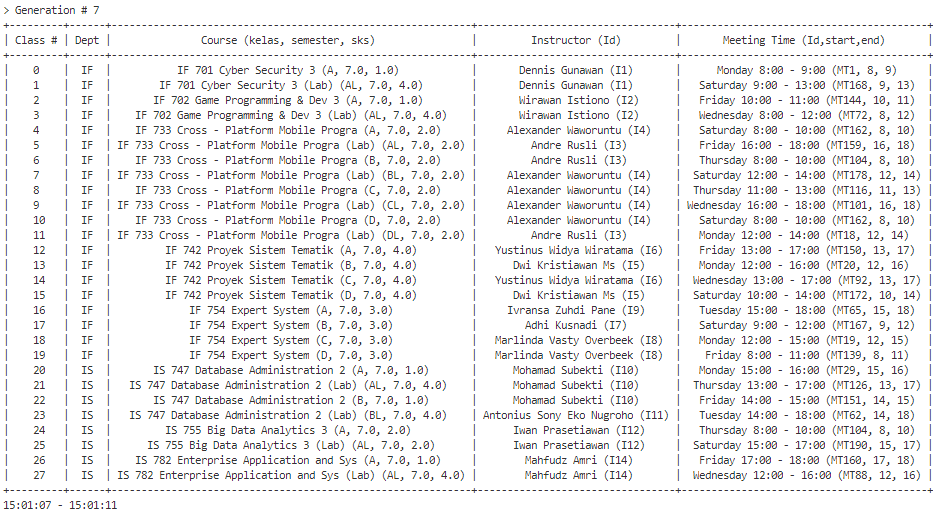
Gambar 4. 18 Hasil Penjadwalan dengan *Mutation Rate* 0.1 dan *Population* 10

Gambar 4.18 merupakan hasil penjadwalan yang sukses dengan nilai *mutation* *rate* 0.1 dan *population size* 10. Pengujian tersebut dengan waktu selama 3 detik dengan jumlah generasi adalah 18.

Skenario kedua seperti Gambar 4.19 adalah mengubah nilai *population size* menjadi 40 dengan nilai *mutation* *rate* sama dengan uji coba pertama. Gambar 4.20 merupakan hasil penjadwalan dengan menggunakan *population size* = 40. Pada skenario kedua didapatkan penjadwalan yang sukses dengan hasil waktu 4 detik dengan generasi yang lebih sedikit yaitu sebanyak 7 generasi.

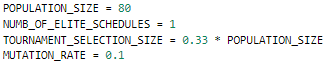


Gambar 4. 19 *Set Population Size* 40

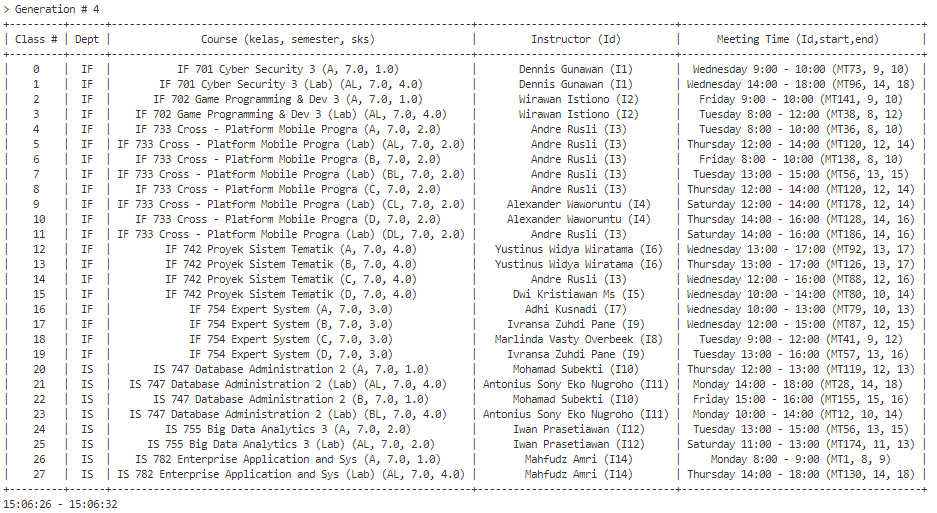


Gambar 4. 20 Hasil Penjadwalan dengan *Mutation Rate* 0.1 dan *Population* 40

Skenario ketiga adalah dengan mengubah nilai *population size* sebanyak 80 dengan nilai *mutation* rate masih sama dengan uji coba sebelumnya yaitu 0.1. Pada Gambar 4.21 merupakan inisialisasi *population size* = 80.



Gambar 4. 21 *Set Population Size* 80



Gambar 4. 22 Hasil Penjadwalan dengan *Mutation Rate* 0.1 dan *Population* 80

Gambar 4.22 merupakan hasil penjadwalan dengan mengubah nilai *population size* menjadi 80. Pada skenario ini mendapatkan hasil penjadwalan yang sukses dengan waktu 6 detik lebih lama dari skenario pertama dan kedua, dengan generasi sebanyak 4 yang lebih sedikit dibandingkan dengan skenario pertama dan skenario kedua.

## Evaluasi Hasil Uji Coba

Pengujian dengan *population size* 10,40 dan 80 diterapkan pada masing-masing semester dengan jumlah percobaan per masing-masing semester adalah 10 kali untuk mencari nilai rata-rata, nilai minimum dan nilai maximum yang dihasilkan pada penjadwalan sebagia berikut.

Tabel 4. 1 Hasil Uji Coba Pengujian dengan *population size* 10,40,80

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Semester | Rata-rata waktu dan Generasi | Populasi (10) | Populasi (40) | Populasi (80) |
| 1 | Waktu (detik) | 1,42 detik | 1,82 detik | 2,39 detik |
|  | Generasi | 4 | 2 | 1 |
| 2 | Waktu (detik) | 13,72 detik | 10,81 detik | 11,77 |
|  | Generasi | 44 | 9 | 6 |
| 3 | Waktu (detik) | 22,02 detik | 23,19 detik | 26,60 detik |
|  | Generasi | 52 | 12 | 6 |
| 4 | Waktu (detik) | 120 detik (CUT OFF) | 180 detik (CUT OFF) | 390 detik |
|  | Generasi | 500 | 300 | 134 |
| 5 | Waktu (detik) | 120 detik (CUT OFF) | 87,14 detik | 96,23 detik |
|  | Generasi | 500 | 149 | 42 |
| 6 | Waktu (detik) | 120 detik (CUT OFF) | 420 detik (CUT OFF) | 300 detik (CUT OFF) |
|  | Generasi | 500 | 300 | 100 |
| 7 | Waktu (detik) | 3,95 detik | 4,86 detik | 5,7 detik |
|  | Generasi | 12 | 5 | 4 |

Berdasarkan hasil pengujian pada Tabel 4.1, pada semester 1,2,3,7 berhasil melakukan generasi jadwal sebelum terkena *cut off*. *Cut off* terjadi apabila program melebihi waktu yang telah di tetapkan. Untuk pengujian dengan populasi 10 rata-rata generasi dan waktu yang dibutuhkan pada masing-masing semester 1,2,3,7 adalah (4 generasi, 1.42 detik), (44 generasi, 13.72 detik), (52 generasi, 22.02 detik) dan (12 generasi, 3.95 detik). Pada pengujian dengan populasi 40 menghasilkan rata-rata generasi dan waktu yang dibutuhkan adalah (2 generasi, 1.82 detik), (9 generasi, 10.81 detik), (12 generasi, 23.19 detik) dan (5 generasi, 4.86 detik). Pengujian dengan populasi 80 menghasilkan rata-rata generasi dan waktu yang dibutuhkan adalah (1 generasi, 2.39 detik), (6 generasi, 11.77 detik), (6 generasi, 26.60 detik) dan (4 generasi, 5.7 detik). Pada Semester 4 untuk pengujian dengan populasi 10 dan populasi 40 mengalami *cut off* sebelum terbentuknya jadwal. Tetapi, jadwal berhasil dibentuk dengan populasi 80 dengan rata-rata generasi sebanyak 134 generasi dengan rata-rata waktu 390 detik Pada semester 5 juga terjadi *cutt off* saat pengujian menggunakan populasi 10. Pada populasi 40, jadwal berhasil terbentuk dengan rata-rata generasi sebanyak 149 generasi dengan rata-rata waktu yang dibutuhkan sebanyak 87,14 detik dan pada populasi 80, jadwal juga berhasil terbentuk dengan rata-rata generasi sebanyak 42 generasi dengan waktu yang dibutuhkan 96,23 detik. Pada semester 6 uji coba menggunakan populasi 10,40,80 terkena *cut off* sehingga gagal menghasilkan jadwal dengan rata-rata generasi setiap populasi adalah 500,300,100 dan rata-rata waktu 120,420 dan 100 detik. Penjadwalan gagal dikarenakan perbandingan 2:1 antara jumlah kelas yang terbuka lebih banyak dibandingkan dengan jumlah dosen yang mengajar setiap kelas sehingga terjadi banyak *conflict* dalam menyusun jadwal dalam 1 minggu.

# BAB V

**SIMPULAN DAN SARAN**



## Simpulan

Berdasarkan dari hasil implementasi dan hasil uji coba yang telah dilakukan, simpulan dari penelitian adalah sebagai berikut.

1. Implementasi Algoritma Genetika untuk penjadwalan mata kuliah pada sistem KRS telah selesai dilakukan dengan menggunakan *dataset* Informatika BIA UMN.
2. Dari hasil uji coba pengubahan nilai populasi dapat disimpulkan dengan rata-rata populasi 10 menghasilkan penjadwalan dengan generasi paling banyak namun memiliki waktu yang lebih cepat dibandingkan dengan populasi 40 dan 80 karena dengan populasi yang sedikit membuat program untuk cepat melakukan proses *selection, crossover, mutation* dan pemberian nilai *fitness* untuk setiap generasi yang baru dalam mencari penjadwalan yang optimal.

## Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa saran untuk pengembangan lanjutan, yaitu sebagai berikut.

1. Penelitian selanjutnya diharapkan dapat menghasilkan penjadwalan yang tidak hanya berdasarkan semester yang diinginkan, tetapi juga dapat membuat penjadwalan secara menyeluruh untuk semester ganjil dan semester genap.
2. Menambahkan *constraint* atau menggunakan metode yang berbeda untuk mendapatkan penjadwalan seluruh semester ganjil dan seluruh semester genap.
3. Menggunakan dataset yang memiliki ketersediaan ruangan sehingga dapat menghasilkan penjadwalan yang memiliki ruangan.

# DAFTAR PUSTAKA

Ana, Dwi, et al. *Model Penjadwalan Mata kuliah Secara Otomatis Berbasis Algoritma Particle Swarm Optimization (PSO)*.

Arief Widyanto, Muhammad, and Arya Wicaksana. “Implementasi Algoritma Genetika Untuk Penjadwalan Sidang Magang (Studi Kasus: Program Studi Informatika Universitas Multimedia Nusantara).” *Kc.Umn.Ac.Id*, 2019, kc.umn.ac.id/11659/. Accessed 13 Feb. 2020.

Arifudin, Riza. “OPTIMASI PENJADWALAN PROYEK DENGAN PENYEIMBANGAN BIAYA MENGGUNAKAN KOMBINASI CPM DAN ALGORITMA GENETIKA.” *JURNAL MASYARAKAT INFORMATIKA*, vol. 2, no. 4, 22 Mar. 2012, pp. 1–14, ejournal.undip.ac.id/index.php/jmasif/article/view/2649, 10.14710/jmasif.2.4.1-14. Accessed 13 Feb. 2020.

Dwiatmoko, Thomas, and Adhi Kusnadi. “Implementasi Algoritma Genetika Pada Sistem Rekomendasi Spesifikasi Komputer.” *Kc.Umn.Ac.Id*, 2019, kc.umn.ac.id/10872/. Accessed 13 Feb. 2020.

Janata, Ari, and Elin Haerani. “Sistem Penjadwalan Outsourcing Menggunakan Algoritma Genetika (Studi Kasus : PT. Syarikatama).” *Jurnal CoreIT*, vol. 1, no. 2, 2015, pp. 2460–738. Accessed 13 Feb. 2020.

Josi, Ahmat. “Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis Web Dengan Mengadopsi Model Waterfall (Studi Kasus: STMIK Prabumulih).” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 2, no. 2, 21 July 2017, pp. 77–83, www.ejournal.poltektegal.ac.id/index.php/informatika/article/view/517/554, 10.30591/jpit.v2i2.517. Accessed 13 Feb. 2020.

Kusnadi, Adhi, and David Setyadi Santoso. “Implementasi Algoritma Genetika Pada Penempatan Tugas Asisten Laboratorium Berbasis Web.” *Ultimatics : Jurnal Teknik Informatika*, vol. 7, no. 2, 2015, pp. 139–147, ejournals.umn.ac.id/index.php/TI/article/view/353, 10.31937/ti.v7i2.353. Accessed 13 Feb. 2020.

Mutrofin, Siti, et al. “OPTIMASI TEKNIK KLASIFIKASI MODIFIED K NEAREST NEIGHBOR MENGGUNAKAN ALGORITMA GENETIKA.” *Jurnal Gamma*, vol. 10, no. 1, 16 Sept. 2015, ejournal.umm.ac.id/index.php/gamma/article/view/2493.

Oktarina, Dwi, and Alyauma Hajjah. “Perancangan Sistem Penjadwalan Seminar Proposal Dan Sidang Skripsi Dengan Metode Algoritma Genetika.” *JOISIE (Journal Of Information Systems And Informatics Engineering)*, vol. 3, no. 1, 11 July 2019, pp. 32–40, www.ejournal.pelitaindonesia.ac.id/ojs32/index.php/JOISIE/article/view/421/378, 10.35145/joisie.v3i1.421. Accessed 13 Feb. 2020.

Permadi, Ipung. “Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan Tebangan Hutan (Applying of Genetic Algorithm for Scheduling Optimation Cuts Away Forest).” *JUITA*, vol. 1, 2010, media.neliti.com/media/publications/93030-ID-penerapan-algoritma-genetika-untuk-optim.pdf.

Setemen, Komang. “IMPLEMENTASI ALGORITMA GENETIKA DALAM PENGEMBANGAN SISTEM APLIKASI PENJADWALAN KULIAH.” *Jurnal IKA*, vol. 8, no. 1, 2010, ejournal.undiksha.ac.id/index.php/IKA/article/view/156, 10.23887/ika.v8i1.156. Accessed 13 Feb. 2020.

Sistem, Rancang, et al. *JURNAL SISTEM DAN INFORMATIKA*.

Suhartono, Entot. “OPTIMASI PENJADWALAN MATA KULIAH DENGAN ALGORITMA GENETIKA (Studi Kasus Di AMIK JTC Semarang).” *INFOKAM*, vol. 11, no. 5, 15 Dec. 2015, amikjtc.com/jurnal/index.php/jurnal/article/view/86. Accessed 13 Feb. 2020.

Suprayogi, Dwi, and Wayan Mahmudy. *Suprayogi, Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry 121 Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry*. 21 July 2014.

Widodo, Agus, and Wayan Firdaus Mahmudy. *Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner Image Processing and Computer Vision View Project Beef Cattle Feed Optimization View Project*

# DAFTAR LAMPIRAN

1. Daftar Riwayat Hidup
2. Form Bimbingan Skripsi