

ALGORITMA GENETIKA DALAM PENGOPTIMALAN JADWAL PERKULIAHAN

Hendi Kusnadi^{1*)}, Mico Fahrizal²
¹Informatika
*) micofahrizal2019@gmail.com

Abstrak

Proses penjadwalan mata kuliah adalah suatu proses untuk menerapkan event yang berisi komponen mata kuliah, kelas dan dosen pada time slot yang berisi komponen waktu dan ruang. Jadwal mata kuliah merupakan hal yang sangat penting bagi kelancaran proses belajar mengajar. Oleh karena itu dibutuhkan suatu sistem dan metode optimasi yang dapat diterapkan didalam menyusun jadwal mata kuliah.

Algoritma Genetika merupakan salah satu algoritma yang tepat digunakan dalam menyelesaikan masalah optimasi kompleks. Agar proses penjadwalan yang dihasilkan dapat optimal dan merata. Diharapkan dengan digunakannya Algoritma Genetika ini akan membantu di dalam proses penjadwalan dan hasil penjadwalan yang diperoleh bisa lebih optimal.

Salah satu algoritma yang muncul untuk menyelesaikan persoalan ini adalah algoritma genetika, Pada penelitian ini akan dibuat program algoritma genetika untuk membuat penjadwal mata kuliah. Sistem algoritma genetik didesain menggunakan representasi kromosom, inisialisasi populasi, reproduksi dan seleksi, crossover, mutasi dan evaluasi fitness.

Kata Kunci: Algoritma Genetika, penjadwalan, Optimasi.

PENDAHULUAN

Penjadwalan mata kuliah adalah kegiatan administratif yang paling utama di universitas. Dalam masalah penjadwalan mata kuliah, sejumlah mata kuliah yang dialokasikan ke sejumlah ruang kelas yang tersedia dan sejumlah slot waktu disertai dengan constraints. Constraint terbagi atas dua jenis, yaitu hard constraints dan soft constraints (Hasan et al., 2016).

Penjadwalan kegiatan belajar mengajar dalam suatu kampus merupakan pekerjaan rutin yang dilakukan setiap semester. Salah satu masalah penjadwalan yang umum dijumpai di perguruan tinggi adalah masalah penjadwalan mata kuliah. Penjadwalan mata kuliah merupakan proses penyusunan jadwal pelaksanaan yang menginformasikan sejumlah mata kuliah yang diberikan, dosen yang mengajar, ruang tempat belajar, waktu serta mahasiswa yang mengambil mata kuliah tersebut. Setiap semester, beberapa perguruan tinggi menghadapi permasalahan yang sama, yaitu bagaimana menjadwalkan mata kuliah dengan kendala waktu, dosen yang tersedia, kapasitas ruangan yang terbatas, dan mahasiswa yang mengambil mata kuliah, dimana proses penjadwalan tersebut tetap memperhatikan sejumlah batasan dan syarat-syarat tertentu (Ariany, n.d.).

Masalah optimasi dapat diselesaikan dengan beberapa cara, salah satunya dengan menggunakan algoritma pencarian heuristik. Untuk kasus yang sederhana digunakan algoritma pencarian heuristik yang biasa seperti breadth-first search and depth-first search.

Untuk input dan persyaratan yang lebih rumit seperti pada kasus penjadwalan mata kuliah, algoritma pencarian heuristik sudah tidak dapat digunakan dengan baik untuk mendapatkan solusi yang diinginkan. Dalam kasus penjadwalan mata kuliah, diperlukan algoritma yang lebih baik yaitu algoritma yang dapat menyelesaikan masalah multi-kriteria dan multi-objektif (Optimasi Arsip Penyimpanan Dokumen Foto Menggunakan Algoritma Kompresi Deflate (Studi Kasus: Studio Muezzart) Bahrudin et al., 2020) satu algoritma yang dapat digunakan adalah algoritma genetika.

Ada tiga keunggulan dari aplikasi algoritma genetika dalam proses optimasi yaitu (Gunawan D, 2020):

- a. Algoritma genetika tidak terlalu banyak memerlukan persyaratan matematika dalam penyelesaian proses optimasi. Algoritma genetika dapat diaplikasikan pada beberapa jenis fungsi obyektif dengan beberapa fungsi pembatas baik berbentuk linier maupun non-linier.
- b. Operasi evolusi dari algoritma genetika sangat efektif untuk mengobservasi posisi global secara acak.
- c. Algoritma genetika mempunyai fleksibilitas untuk diimplementasikan secara efisien pada problematika tertentu.

Berdasarkan permasalahan diatas, maka peneliti menggunakan Algoritma Genetika untuk menyelesaikan masalah penjadwalan perkuliahan dalam penelitian ini. Dengan dibuatnya sebuah aplikasi penyusunan jadwal perguruan tinggi ini, diharapkan dapat mempermudah para pengguna dalam penyusunan jadwal kuliah sesuai dengan sumber daya dan syarat atau batasan yang sudah ditentukan.

KAJIAN PUSTAKA

Optimasi

Optimasi adalah proses memaksimalkan atau meminimasi suatu fungsi tujuan dengan tetap memperhatikan pembatas yang ada (Giovani et al., 2020). Optimasi memegang peranan penting dalam mendesain suatu sistem. Dengan optimasi, suatu sistem dapat menghasilkan ongkos yang lebih murah atau profit yang lebih tinggi, menurunkan waktu proses dan sebagainya. Optimasi dalam waktu sekarang memerlukan bantuan software untuk menyelesaikan permasalahan yang ditemukan untuk mendapatkan solusi yang optimal dengan waktu komputasi yang lebih singkat. Teknik-teknik optimasi tidak banyak berubah dalam beberapa tahun belakangan ini (Parinata & Puspaningtyas, 2021). Tetapi aplikasi dari teknik optimasi telah menjamur di berbagai bidang secara cepat. Keberhasilan penerapan teknik optimasi paling tidak memerlukan tiga syarat, yaitu kemampuan membuat model matematika dari permasalahan yang dihadapi, pengetahuan akan program komputer dan pengetahuan akan teknik optimasi itu sendiri.

Pemakaian software dalam menyelesaikan masalah optimasi sangatlah penting. Persoalan sederhana mungkin bisa diselesaikan dengan suatu algoritma yang hanya memerlukan satu atau dua iterasi, namun jika sudah menyangkut permasalahan dengan skala besar dan melibatkan banyak iterasi dalam menemukan solusi optimal dari permasalahan itu maka diperlukan pemakaian software untuk menyelesaikannya.

Masalah optimasi dapat dikategorikan ke dalam dua kelas besar, yaitu optimasi tanpa pembatas (unconstrained optimization) dan optimasi dengan pembatas (constrained optimization). Dari namanya kita bisa mengetahui bahwa optimasi tanpa pembatas hanya melibatkan fungsi tujuan, tidak ada pembatas (constraint), sedangkan optimasi dengan pembatas, selain fungsi tujuan kita juga mempunyai tambahan pembatas yang membuat permasalahan lebih rumit. Dalam constrained optimization, dengan adanya pembatas diperlukan algoritma yang berbeda untuk menyelesaikannya.

Prosedur pemecahan masalah optimasi adalah memodelkan persoalannya ke dalam sebuah program matematis dan kemudian memecahkannya dengan menggunakan teknik-teknik atau metode optimasi seperti program linier, program nonlinier, program tujuan ganda, dan metode-metode lainnya yang sudah berkembang saat ini. Dalam penelitian ini akan menggunakan metode tabu search untuk menyelesaikan persoalan optimasi (Nurkholis & Sitanggang, 2020).

Penjadwalan

Jadwal adalah daftar output (tabel kegiatan)/rencana kegiatan yang harus dihasilkan dalam jangka waktu tertentu, biasanya disusun menurut urutan prioritas dengan pembagian waktu pelaksanaan yang terperinci, sedangkan penjadwalan adalah proses pembuatan/cara menjadwalkan suatu data menjadi jadwal (Hasan et al., 2016).

Penjadwalan merupakan proses untuk menyusun suatu jadwal atau urutan proses yang diperlukan dalam sebuah permasalahan (Samsugi & Burlian, 2019). Permasalahan penjadwalan biasanya berhubungan dengan penjadwalan kelas dalam sekolah atau perkuliahan dan juga dalam lingkup yang tidak jauh berbeda seperti penjadwalan pelajaran sekolah, penjadwalan ujian, atau bisa juga penjadwalan karyawan, baik dalam suatu perusahaan ataupun dalam rumah sakit. Dalam penjadwalan kuliah, akan dibahas tentang pembagian jadwal untuk tiap mahasiswa pada kuliah tertentu sekaligus dosen pengajarnya, dalam penjadwalan pelajaran sekolah akan dibahas tentang pembagian jadwal pelajaran untuk tiap-tiap kelas yang ada beserta guru pengajar pelajaran tersebut, dalam penjadwalan ujian akan dibahas pengaturan dosen yang menjaga ujian dan mahasiswa atau murid yang menempati ruang ujian yang ada, sedangkan pada penjadwalan karyawan, dilakukan pengaturan karyawan yang akan bekerja pada waktu tertentu di bagian tertentu.

Proses tersebut tentu saja dibuat berdasarkan permasalahan yang ada. Beberapa proses umum yang perlu dilakukan untuk menyelesaikan suatu proses penjadwalan adalah:

1. Mendefinisikan atau membuat model dari permasalahan.

Model yang dibuat mencakup proses apa saja yang akan dikerjakan dalam persoalan penjadwalan yang ada. Atau lebih jelasnya jadwal apa saja yang akan dibuat.

2. Mendesain metode penyelesaian untuk permasalahan penjadwalan tersebut.

Dari model yang telah ada, ditentukan metode apa yang akan digunakan untuk menyelesaikan permasalahan penjadwalan tersebut.

3. Mencari bermacam-macam contoh permasalahan penjadwalan yang telah dibuat.

Dalam proses ini dilakukan pencarian penyelesaian penjadwalan yang pernah digunakan agar dapat dipakai sebagai referensi dalam proses yang sedang dilakukan.

Sedangkan pembahasan penjadwalan menurut Tomas Muller dan Roman Bartak sebagai berikut:

1. Aktivitas yang dilakukan

Maksudnya adalah penentuan dari permasalahan penjadwalan yang dibahas.

2. Sumber-sumber yang dipakai

Sumber-sumber yang dipakai berarti hal-hal yang dapat digunakan untuk menyelesaikan permasalahan penjadwalan (aktivitas) yang telah ditentukan atau bisa juga dikatakan sebagai data yang digunakan. Misalnya pada penjadwalan kuliah diperlukan data mata kuliah, dosen, mahasiswa, kelas dan sumber-sumber lain yang diperlukan.

3. Syarat-syarat yang diperlukan

Syarat disini adalah hal-hal yang perlu diperhatikan ketika menyusun suatu penjadwalan.

4. Hubungan Timbal Balik

Yang dimaksud hubungan timbal balik disini adalah bagaimana jadwal yang telah dibuat tersebut dapat sesuai dengan yang diinginkan oleh user.

Berikut contoh ilustrasi peranan penjadwalan dalam situasi kehidupan sehari-hari:

1. Rencana produksi penggilingan kertas.

Bahan-bahan dalam penggilingan kertas adalah serat kayu dan pulp, hasilnya adalah sejumlah gulungan kertas. Komponen utama dalam penggilingan kertas adalah mesin. Setiap mesin mengerjakan tugas yang berbeda-beda digolongkan berdasarkan berat kertas, kualitas dan warna. Penjadwalan mempertimbangkan perputaran waktu. Waktu penggunaan mesin bergantian satu sama lain berdasarkan kualitas kertas yang dihasilkan. Mesin tidak akan berhenti selama pertukaran tugas, karena selama pertukaran (transfer) produksi kertas tidak mengalami masalah yang spesifik pada sistem produksi. Objektif masalah penjadwalan dalam kasus ini adalah memaksimalkan hasil produksi dengan biaya minimum. Memaksimalkan produksi dalam arti sudah termasuk meminimumkan waktu pertukaran antar mesin.

2. Menyusun jadwal perawat di suatu rumah sakit.

Setiap rumah sakit mempunyai susunan kepegawaian yang memerlukan pertukaran dari hari ke hari. Sebagai contoh sejumlah perawat menghendaki libur pada hari minggu, dan biasanya lebih banyak dari pada hari sabtu. Dan rumah sakit juga menghendaki shift malam lebih sedikit dibanding siang. Keadaan dan pengaturan tersebut menghendaki adanya penjadwalan. Objektif dari permasalahan ini adalah menuntut pengaturan shift yang berbeda-beda, dengan biaya yang berbeda juga.

3. Contoh lain yang memerlukan penjadwalan adalah penjadwalan mata pelajaran di sekolah, penjadwalan mata kuliah di universitas, penjadwalan karyawan di suatu perusahaan, dan sebagainya.

Algoritma Gentika

Algoritma genetika (AG) adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional (Gunawan D, 2020).

Algoritma genetika pertama kali ditemukan oleh John Holland pada tahun 1975 melalui sebuah penelitian dari Universitas Michigan yang kemudian dikembangkan oleh muridnya David Goldberg pada tahun 1989. Awalnya tujuan Holland adalah bukan untuk merancang algoritma yang mampu memecahkan masalah yang spesifik, melainkan hanya untuk mempelajari fenomena adaptasi seperti yang terjadi di alam sekaligus untuk mengembangkan cara-cara bagaimana mekanisme adaptasi alam dapat diimpor ke sebuah sistem komputer. John Holland dalam bukunya yang berjudul *Adaptation In Natural And Artificial Systems*, menyatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Untuk memahami algoritma genetika diperlukan 3 tahapan pendalaman yaitu: aspek filosofis, struktur dan komponen.

Struktur Umum Algoritma Genetika

Secara garis besar algoritma genetika memiliki banyak kesamaan dengan mekanisme genetika alami, baik dalam tahapan prosesnya maupun definisi dari istilah-istilah atau terminologi yang digunakan. Ekuivalensi istilah algoritma genetika dan genetika alami pada biologi dapat ditunjukkan pada Tabel 1.

Tabel 1. Perbandingan Istilah pada Genetika Alami dan Algoritma Genetika .

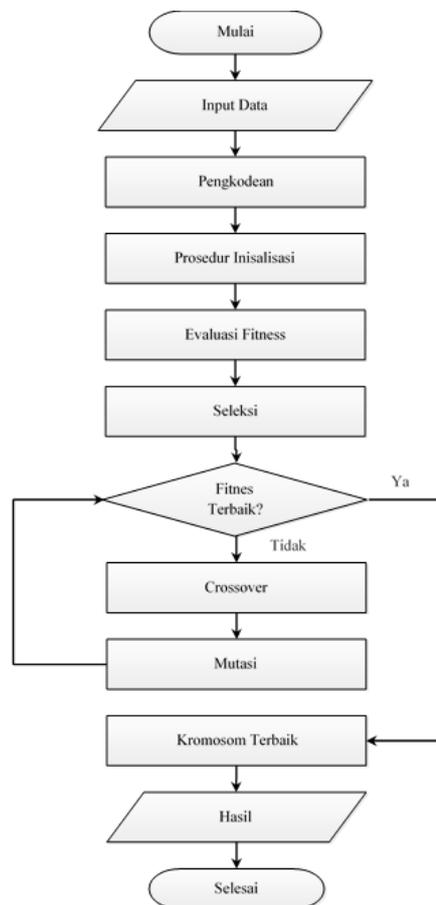
Genetika Alami	Algoritma Genetika
Kromosom	String
Gen	Fitur, Karakter, atau Detector
Allel	Nilai fitur
Locus	Posisi String
Genotip	Struktur
Fenotip	Set parameter, solusi alternatif, struktur yang dikodekan

Secara alamiah semua organisme terdiri dari sel, di dalam setiap sel terdiri dari sekumpulan kromosom. Kromosom terbentuk dari sekumpulan gen, membuat satu kesatuan yang tersusun dalam rangkaian linier. Setiap gen mempunyai letak tersendiri di dalam kromosom, disebut lokus. Gen tersusun dari Deoxyribo Nucleic Acid (DNA) yang membawa sifat-sifat keturunan. Setiap gen menyandikan protein tertentu suatu sifat,

contoh : gen warna mata binatang dengan posisi lokus 10. Bagian tertentu dari gen di dalam genome disebut genotip. Beberapa sifat individu yang menunjukkan perbedaan gen dan berada pada bagian yang berbeda disebut alel.

Algoritma genetika merepresentasikan individu sebagai sebuah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Algoritma ini menyelesaikan permasalahan dalam pencarian kromosom yang terbaik. Kromosom- kromosom terbaik diperoleh dengan cara populasi awal yang dibangun secara acak dan populasi berikutnya yang merupakan hasil evolusi kromosom-kromosom melalui proses iterasi (seleksi, crossover dan mutasi) yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi fitness. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Setelah melalui beberapa generasi, maka akan diperoleh kromosom terbaik.

Algoritma genetika secara umum dapat diilustrasikan pada flowchart berikut:



Gambar 1. Flowchart Algoritma Genetika

Dari Gambar 1. diatas, struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut:

Membangkitkan populasi awal

Populasi awal dibangkitkan secara random, dimana populasi tersebut berisi beberapa kromosom yang telah didefinisikan sehingga didapatkan solusi awal. Populasi itu sendiri terdiri dari sejumlah kromosom yang merepresentasikan solusi yang diinginkan.

Membentuk generasi baru

Dalam membentuk generasi baru digunakan tiga operator yaitu operator reproduksi/seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru dimana generasi baru ini merupakan representasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (offspring).

Evaluasi solusi

Pada tiap generasi, kromosom akan diukur dengan fungsi fitness. Nilai fitness suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Apabila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Beberapa kriteria berhenti yang sering digunakan antara lain:

1. Berhenti pada generasi tertentu.
2. Berhenti setelah beberapa generasi berturut-turut didapatkan nilai fitness tertinggi.
3. Berhenti pada n generasi dimana nilai fitness dari populasi tidak mengalami perubahan.

Kontrol Parameter Algoritma Genetika

ada dua parameter dasar dari algoritma genetika adalah:

Probabilitas Crossover (P_c)

Kemungkinan terjadinya operator crossover dikendalikan oleh nilai P_c . Jika probabilitas crossover terlalu tinggi, kemungkinan besar kromosom yang memiliki kualitas baik akan cepat hilang dalam populasi. Sebaliknya, probabilitas yang rendah akan membuat kromosom-kromosom sulit untuk konvergen.

Probabilitas Mutasi (P_m)

Probabilitas mutasi digunakan untuk meningkatkan variasi populasi, setiap gen pada kromosom akan terjadi perubahan. Nilai P_m yang rendah mengakibatkan kurangnya variasi individu.

Komponen-komponen Utama Algoritma Genetika

Ada 6 komponen utama yang terdapat di dalam algoritma genetika, yaitu:

Teknik Pengkodean

Menurut (Gen dan Cheng, 2000), pengkodean adalah suatu teknik untuk menyatakan populasi awal sebagai calon solusi suatu masalah ke dalam suatu kromosom sebagai suatu kunci pokok persoalan ketika menggunakan algoritma genetika. Teknik pengkodean ini meliputi pengkodean gen dan kromosom. Gen merupakan bagian dari kromosom yang dapat direpresentasikan dalam bentuk string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program, atau representasi lain yang dapat diimplementasikan untuk operator genetika.

Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Inisialisasi kromosom akan dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

Fungsi Evaluasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Inisialisasi kromosom akan dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

$$fitness(Vz) = \frac{1}{f(Vz)}, z = 1, 2, \dots, pop_size$$

Kromosom dengan nilai fitness tertinggi mempunyai kesempatan yang lebih besar untuk terpilih sebagian parent dan diharapkan dapat menghasilkan keturunan-keturunan yang lebih baik dari orang tuanya sehingga didapat suatusolusi yang optimal setelah beberapa generasi.

Fungsi Seleksi

Menurut Desiani dan Arhami (2006), fungsi seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi paling fit. Salah satu metode seleksi adalah metode roulette wheel selection (roda roulette).

Langkah-langkah metode ini adalah sebagai berikut :

1. Hitung Total Fitness (F) :|

$$Total\ Fitness = \sum Fk ; k = 1, 2, 3 \dots, pop_size$$

2. Hitung Fitness relative tiap individu :

$$Pk = \frac{Fk}{Total\ Fitness}$$

3. Hitung fitness kumulatif:

$$q_1 = p_1$$

$$q_k = q_{k-1} + p_k ; k = 2, 3 \dots pop_{size}$$

4. Pilih induk yang akan menjadi kandidat untuk crossover dengan cara bangkitkan bilangan random r , jika $q_k \leq r < q_{k+1}$ maka pilih kromosom ke $(k+1)$ sebagai kandidat induk.

Operator Crossover

Operasi crossover dalam algoritma genetika digunakan untuk menambah keanekaragaman string dalam populasi melalui penyilangan antarstring yang diperoleh dari proses seleksi sebelumnya.

Langkah-langkah crossover tersebut adalah sebagai berikut :

1. Bangkitkan bilangan acak r antara $[0,1]$
2. Pilih bilangan acak tersebut yang kurang dari pc . Jika jumlah bilangan acak yang dihasilkan dari pilihan tersebut ganjil maka buang salah satunya.
3. Pilih bilangan acak antara 1 sampai $(L-1)$ di mana L adalah panjang kromosom. Bilangan ini akan menentukan posisi crossover 1 titik.
4. Penyilangan ini dilakukan untuk semua pasangan kromosom yang akan disilangkan.

Operator Mutasi

Mutasi dilakukan untuk menghindari terjadinya konvergensi prematur. Proses mutasi dilakukan dengan mengubah gen-gen tertentu secara acak sehingga memunculkan beragam calon solusi. Pemilihan gen dilakukan secara acak untuk dimutasi. Mutasi dapat ditentukan dengan langkah-langkah berikut :

1. Hitung jumlah bit yang ada pada populasi, yaitu $Jlh.Bit = Ukuran\ populasi \times L$
2. Bangkitkan bilangan acak antara $[0\ 1]$ sebanyak $Jlh.Bit$
3. Pilih bilangan acak tersebut yang kurang dari pm . kemudian, dilakukan mutasi.

UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Suri & Puspaningrum, 2020).

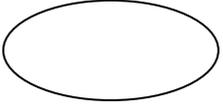
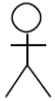
Perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya. Standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman. Diagram UML saling terkait kekonsistenan rancangan diagram yang satu dengan lainnya (Dinasari et al., 2020).

Berikut adalah keterkaitan diagram-diagram pada UML untuk pengembangan penelitian (Dyah Ayu Megawaty & Putra, 2020):

Use Case Diagram

Use Case Diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat dan mendeskripsikan sebuah interaksi antara satu atau lebih aktor (Ayunandita & Riskiono, 2021). Fungsi *use case diagram* digunakan untuk mengetahui apa saja yang. Simbol-simbol yang digunakan dijelaskan pada tabel 2.

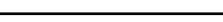
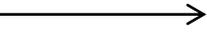
Tabel 2. Simbol-simbol Use Case Diagram (D A Megawaty, 2020)

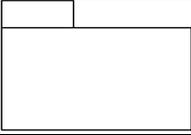
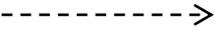
SIMBOL	KETERANGAN
	Use case , Sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal.
<<extend>> 	Ektensi / <<extend>> Untuk melakukan proses dengan komputer.
	Asosiasi komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
	Actor Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	Generalisasi / Generalitation Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> .
<<include >> 	Menggunakan / <<include >> untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini

Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Budiman et al., 2019). Pembanguna *class diagram* antara lain deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Ariyanti et al., 2020). Penjelasan simbol-simbol *class diagram* pada tabel 3.

Tabel 3. Simbol-simbol Relasi Class Diagram (Puspaningrum, 2017)

SIMBOL	KETERANGAN
	Association Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Asosiasi berarah/directed association Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

	Package merupakan sebuah bungkusan dari satu atau lebih kelas			
	Kebergantungan/dependency Depedensi juga menghubungkan dua kelas, tetapi dengan cara yang berbeda dibandingkan asosiasi.			
	Agregasi/aggregation relasi antar kelas dengan makna semua-bagian (<i>Whole-part</i>).			
	Generalisasi/Generalitation Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).			
<table border="1" data-bbox="279 734 587 851"> <tr> <td>Nama Kelas</td> </tr> <tr> <td>-Atribut</td> </tr> <tr> <td>+Operasi()</td> </tr> </table>	Nama Kelas	-Atribut	+Operasi()	Kelas Kelas pada struktur sistem
Nama Kelas				
-Atribut				
+Operasi()				
<i>Name_interface</i> 	Antarmuka/interface Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.			

Multiplicity

Multiplicity menunjukkan jumlah suatu objek yang bisa berhubungan dengan objek satu dengan yang lain (Ayunandita & Riskiono, 2021). Umumnya ditunjukkan dengan berapa banyak objek yang bisa mengisi properti “satu” atau “banyak”, tetapi secara khusus dapat ditunjukkan pula dengan bilangan integer lebih besar atau sama dengan nol. Penjelasan *multiplicity* UML tabel 4.

Tabel 4. Notasi *Multiplicity* UML (Oktaviani & Ayu, 2021)

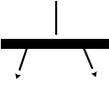
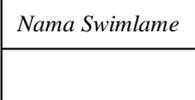
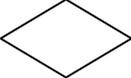
<i>Multiplicity</i>	Arti
*	Banyak
0	Nol
1	Satu
0..*	Antara nol sampai banyak
1..*	Antara satu sampai banyak
0..1	Nol atau satu
1..1	Tepat satu

Activity Diagram

Activity diagram adalah sebuah diagram yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis (Fatimah et al., 2020).

Activity diagram menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Tetapi aktivitas yang dilakukan dapat menjelaskan alur yang sedang berjalan pada BPN bidang PPP, yang kemudian dilanjutkan dengan aktivitas didalam sistem (Monica & Borman, 2017). Berikut symbol - simbol diagram aktivitas yang digunakan dalam tabel 5.

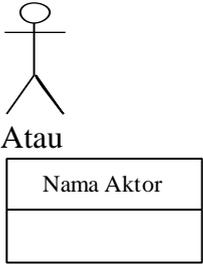
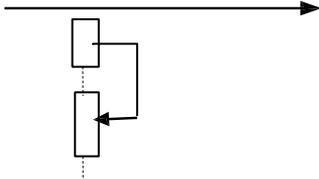
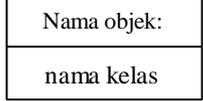
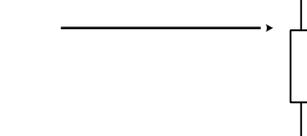
Tabel 5. Simbol – Simbol Activity Diagram (Setiawansyah et al., 2020)

SIMBOL	KETERANGAN
	Status Awal Sebuah diagram aktivitas memiliki sebuah status awal.
	Panah menggambarkan sasaran yang mengawali kegiatan.
	Aktivitas Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
	Fork Digunakan utk menunjukkan kegiatan yg dilakukan secara paralel.
	Join Digunakan untuk menunjukkan kegiatan yg digabungkan.
	Nama Swimlane Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
	Status Akhir Diagram aktivitas memiliki sebuah status akhir.
	Penggabungan / Join Asosiasi penggabungan dimana lebih dari satu aktivitas gabungan menjadi satu.
	Percabangan / Decision Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Sequence Diagram

Sequence diagram menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Maskar & Dewi, 2021). Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan (Ayu et al., 2021). Berikut adalah simbol-simbol yang ada pada *sequence diagram* tabel 6.

Tabel 6. Simbol – Simbol *Sequence Diagram* (Putra et al., 2009)

SIMBOL	KETERANGAN
 <p>Atau</p>	<p>Actor Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>1:nama metode()</p> 	<p>Pesan tipe call Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
	<p>Objek Menyatakan objek yang berinteraksi pesan</p>
	<p>Garis Hidup/lifeline Menyatakan kehidupan suatu objek.</p>
<p><<create>></p> 	<p>Pesan tipe create Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
	<p>Waktu aktif Menyatakan objek dalam keadaan aktif dan berinteraksi pesan</p>
<p>1.masukan</p> 	<p>Pesan tipe send Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>1.keluaran</p> 	<p>Pesan tipe return Metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p><<destroy>></p> 	<p>Pesan tipe destroy Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>

METODE

Metode Pengembangan Sistem

Prototyping paradigma dimulai dengan pengumpulan kebutuhan. Secara ideal prototipe berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak (Ulfa & Saputra, 2019). Bila prototipe sedang bekerja atau dibangun, pengembang harus mempergunakan fragmen-fragmen yang ada atau mengaplikasikan alat-alat bantu.



Gambar 2. Paradigma Prototype (Maskar et al., 2020)

Contohnya : report generator, window manager, dan lain-lain yang memungkinkan program yang bekerja untuk dimunculkan secara cepat.

Prototipe bisa berfungsi sebagai : “sistem pertama”

Prototipe bisa juga menjadi masalah karena alasan-alasan sebagai berikut (Ramadhanu & Priandika, 2021):

- 1) Pelanggan melihat apa yang tampak sebagai versi perangkat lunak yang bekerja tanpa melihat bahwa prototipe itu dijalankan bersama-sama tanpa melihat bahwa didalam permintaan untuk membuatnya bekerja.
- 2) Pengembang sering membuat kompromi-kompromi implementasi untuk membuat untuk membuat prototipe bekerja dengan cepat. Sistem operasi atau bahasa pemrograman yang tidak sesuai bisa dipakai secara sederhana mungkin diperoleh dan dikenal.

Prototyping dipakai bila ditemui kondisi

a. Definisi user bersifat umum, user tidak tahu pasti apa yang diinginkan definisi user bersifat tidak rinci, user tidak tahu pasti apa dan bagaimana bentuk

- 1) Masukan
- 2) Proses
- 3) Keluaran

b. Pengembang merasa tidak tahu pasti tentang

- 1) Pilihan algoritma yang dipakai
- 2) Bagaimana lingkungan sistem yang akan dikembangkan
- 3) Bentuk, sifat dan karakteristik antar muka pegawai

Terdapat ketidak pastian dipihak user yaitu tentang apa diinginkan Terdapat ketidak pastian dipihak pengembang yaitu tentang apa yang harus dilakukan.

Black Box Testing

Pada implementasi sebuah sistem informasi, dilakukan pengujian pada software yang dikembangkan untuk mengetahui kelayakan suatu software. Implementasi pada program software yang dilakukan pengujian berdasarkan kualitas dan menentukan validasi kegunaan suatu program. Salah satu utuk tahap pengujian program software adalah blackbox (Audrilia & Budiman, 2020).

Menurut Romeo Black box testing adalah dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai behavioral testing, specification-based testing, input/output testing atau functional testing (Puspaningrum et al., 2020).

Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software. Dengan adanya black box testing, pereka software dapat menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program (Kadir, 2003). Proses event pengujian black box berdasarkan proses pada gambar 3.



Gambar 3. Proses Pengujian Black Box (Raharjo, 2016)

Pengujian black box diatas dapat dijelaskan bahwa, suatu program software yang akan diuji dengan alur ketentuan sebagai berikut:

1. Form, pada tahap ini dilakukan pengujian form yang ada pada program software.
2. Input, berupa entri yang ada di form.
3. Proses, pada tahap ini melakukan proses mengenai skrip yang dibuat untuk pemanggilan form.
4. Output, hasil tampilan yang dilakukan oleh proses dari input.
5. Hasil pengujian, tahap terakhir mengenai kesimpulan dari output sudah sesuai atau belum dengan tampilan dan pemanggilan form pada skrip yang telah dibuat

HASIL DAN PEMBAHASAN

Implementasi Sistem

Tahap implementasi pada sebuah sistem informasi merupakan tahap dimana sistem yang telah dirancang, menjelaskan mengenai pembuatan sistem sesuai dengan analisis dan perancangan sebelumnya. Setelah tahap implementasi dilakukan maka dibutuhkan sebuah pengujian sistem untuk membuktikan bahwa aplikasi dapat berjalan sesuai yang diharapkan. Aplikasi penyusunan jadwal ini dibangun dengan menggunakan bahasa pemrograman java, Algoritma Genetika sebagai penyelesaian masalah dan *MySQL* sebagai *database* nya.

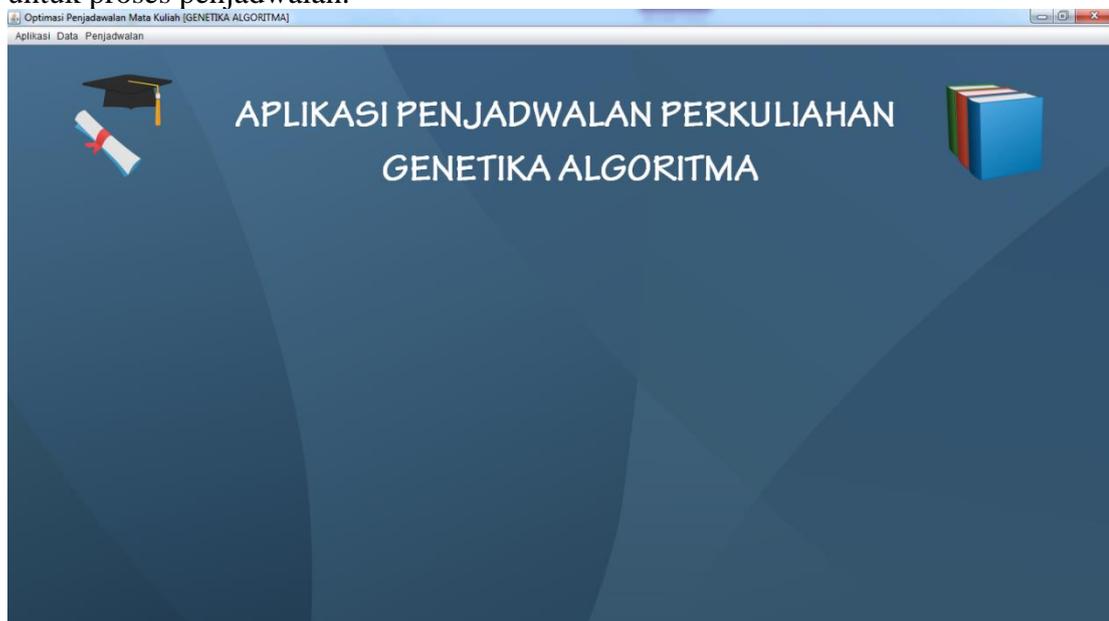
Implementasi Antar Muka

Tahap implementasi sistem merupakan tahap penerapan sistem agar dapat dioperasikan secara optimal sesuai kebutuhan. Implementasi antarmuka yang dibuat pada tahap perancangan, diimplementasikan menjadi bentuk halaman *java* yang dibangun dengan menggunakan perangkat lunak yang dijelaskan pada implementasi program.

Interface Menu Utama

Didalam form utama ini terdapat tiga menu yaitu :

1. Menu Aplikasi yang didalamnya terdapat satu submenu keluar, yang digunakan sebagai salah satu perintah untuk keluar dari aplikasi.
2. Menu Data yang didalamnya terdapat lima submenu terdiri dari, data dosen, data matakuliah, data ruang, hari & jam, dan waktu tidak bersedia.
3. Menu pengampunan semua mata kuliah dan dosen pengampunya.
4. Menu Penjadwalan yang didalamnya terdapat submenu proses, yang digunakan untuk proses penjadwalan.



Gambar 4. *Interface* Menu Utama

Interface Master Dosen

Tampilan ini digunakan untuk *inputkan* data, menyimpan data, mengubah data dan menghapus data master dosen. Berikut *interface* master dosen dapat dilihat pada gambar 5. sebagai berikut:

kode	NIDN	Nama	Alamat	Telp
0		Dwi Budi Setiawa...		
1		Agung Rizky Wirab...		
2		Harits, S.S.		
3		Ari Sulistiyawati, S...		
4		Zaenal Abidin, S.Si.		
5		M. Firdaus, S.Kom.		
6		Jaka Persada Se...		
7		Yohana Tri Utami, ...		
8		Suhaili, S.Ag.		
9		Permata, S.Si.		
10		Parjito, S.Kom.		
11		Tri Karyono, S.ST.		
12		M. Fitratullah, S.S.		
13		M. Firdaus, S.Kom.		
14		Suryono, S.Si.		
15		Rizal Taufiq, M.Ag.		
16		Susidih, S.Kom.		

Gambar 5. Interface Master Dosen

Interface Master Mata Kuliah

Tampilan ini digunakan untuk menginputkan data, menyimpan data, mengubah data dan menghapus data master mata kuliah. Didalam form Mata kuliah inilah data Mata kuliah diinputkan, inputan tersebut antara lain adalah, Kode MK, Nama, SKS, Semester, kategori dan status aktif tidaknya suatu Mata kuliah. Data inputan tersebut lalu disimpan kedalam database Mata kuliah dan kemudian ditampilkan pada *Gridview*.

Berikut interface master mata kuliah dapat dilihat pada gambar 6. sebagai berikut:

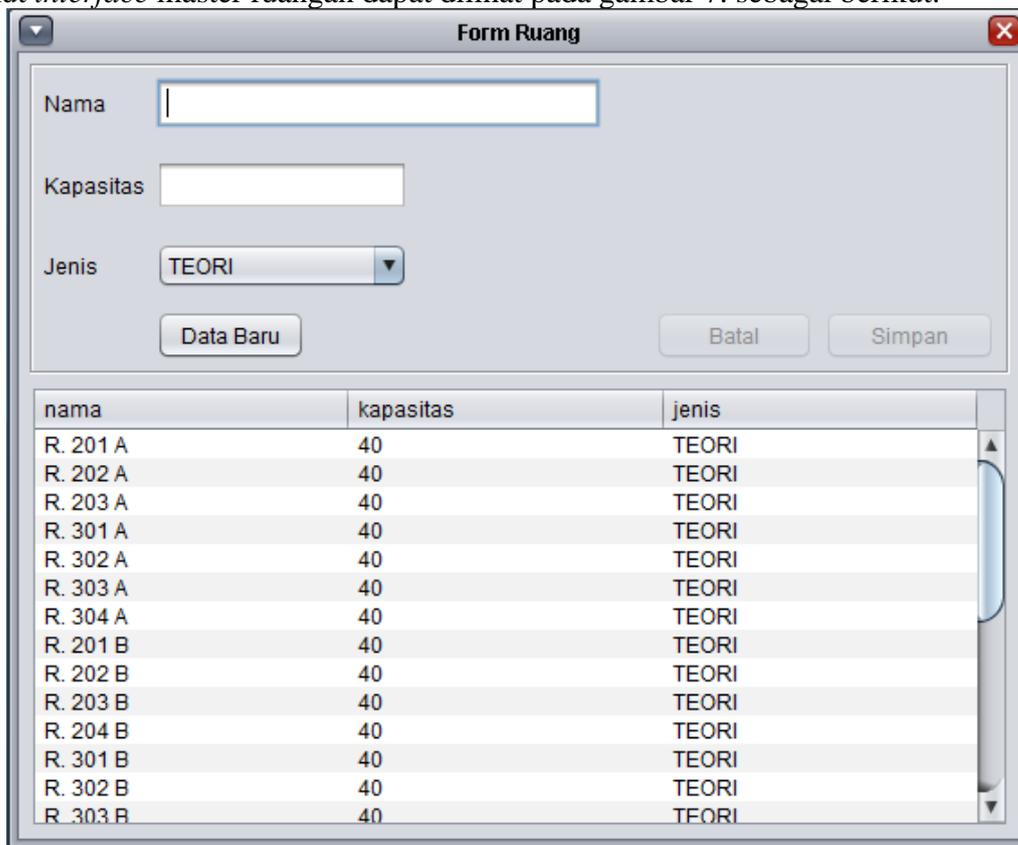
Kode	Nama	SKS	Semester	Jenis
TIS 353	Aljabar Linier	2	5	TEORI
TIS 354	Analisis dan Pemo...	4	5	TEORI
SIS 112	Aplikasi Komputer 1	2	1	TEORI
PTT 102	Bahasa Inggris 1	2	1	TEORI
TIS 111	Dasar-dasar Pemr...	2	1	TEORI
TIS 115	Elektronika Dasar	2	1	TEORI
TIS 478	Embeded System	2	7	TEORI
TIS 351	Jaringan dan Kom...	2	5	TEORI
TIS 233	Kalkulus II	2	3	TEORI
TIS 479	Keamanan Jaringan	4	7	TEORI
PTT 107	Kewirausahaan	2	5	TEORI
TIS 113	Logika Informatika	2	1	TEORI
PTT 105	Manajemen Bisnis	2	3	TEORI
TIS 114	Matematika Diskrit	2	1	TEORI
TIS 356	Mekanika Elektroni...	4	5	TEORI
TIS 352	Metode Numerik	2	5	TEORI
TIS 475	Pemrograman Web	2	7	PRAKTIKUM

Gambar 6. Interface Master Mata Kuliah

Interface Master Ruangan

Tampilan ini digunakan untuk menginputkan data, menyimpan data, mengubah data dan menghapus data master ruangan. Didalam form Ruang inilah data Ruangan diinputkan, inputan tersebut antara lain adalah, Nama, Kapasitas, SKS, dan Jenis ruangan dengan pilihan regular atau laboratorium sedangkan untuk kode ruang yang diinputkan ke database dibuat Auto Increment. Data inputan tersebut lalu disimpan kedalam database Ruang dan kemudian ditampilkan pada *Gridview*. Apabila kita mau merubah data yang sudah ada maka pilih salah satu data yang ada pada *Gridview* tadi, kemudian *field* yang ada pada *textbox* diganti setelah itu di klik button simpan yang beralih fungsi menjadi button *Update*.

Berikut *interface* master ruangan dapat dilihat pada gambar 7. sebagai berikut:



The screenshot shows a window titled "Form Ruang" with a close button in the top right corner. The form contains three input fields: "Nama" (a text box), "Kapasitas" (a text box), and "Jenis" (a dropdown menu currently showing "TEORI"). Below these fields are three buttons: "Data Baru", "Batal", and "Simpan". At the bottom of the form is a table with three columns: "nama", "kapasitas", and "jenis". The table contains 15 rows of data, all with a capacity of 40 and type "TEORI".

nama	kapasitas	jenis
R. 201 A	40	TEORI
R. 202 A	40	TEORI
R. 203 A	40	TEORI
R. 301 A	40	TEORI
R. 302 A	40	TEORI
R. 303 A	40	TEORI
R. 201 B	40	TEORI
R. 202 B	40	TEORI
R. 203 B	40	TEORI
R. 204 B	40	TEORI
R. 301 B	40	TEORI
R. 302 B	40	TEORI
R. 303 B	40	TEORI

Gambar 7. Interface Master Ruangan

Interface Dosen Pengampu

Tampilan ini digunakan untuk menginputkan data, menyimpan data, mengubah data dan menghapus data dosen pengampu. Pengampuan bertujuan agar dosen mengampu matakuliah yang sesuai dengan kompetensi dosen. inputan pengampu antara lain adalah, Semester, Tahun Akademik, Matakuliah, Dosen dan Kelas. Inputan matakuliah dan dosen mengambil data dari table matakuliah dan dosen. Sedangkan kode pengampu dibuat *Auto Increment*. Data inputan tersebut lalu disimpan kedalam table Pengampu dan kemudian ditampilkan pada *Gridview*.

Berikut *interface* dosen pengampu dapat dilihat pada gambar 8. sebagai berikut:

Nama MK	Nama Dosen	Kelas	Tahun Akademik
Aljabar Linier	Permata, S.Si.	TI 12 A	2014/2015
Aljabar Linier	Permata, S.Si.	TI 12 B	2014/2015
Aljabar Linier	Permata, S.Si.	TI 12 C	2014/2015
Aljabar Linier	Agus Mulyanto, S.Kom., M.T., M...	TI 12 C	2014/2015
Aljabar Linier	Agus Mulyanto, S.Kom., M.T., M...	TI 12 D	2014/2015
Aljabar Linier	Zaenal Abidin, S.Si.	TI 12 D	2014/2015
Aljabar Linier	Zaenal Abidin, S.Si.	TI 12 E	2014/2015
Aljabar Linier	Bentar Priyopradono, S.Kom., ...	TI 12 E	2014/2015
Analisis dan Pemodelan Pera...	Zulkifli, M.T.	TI 12 A	2014/2015
Analisis dan Pemodelan Pera...	Zulkifli, M.T.	TI 12 B	2014/2015

Gambar 8. *Interface* Dosen Pengampu

Interface hari dan waktu

Didalam form inilah data hari dan sesi diinputkan, inputan tersebut antara lain adalah, kode hari, nama hari dan kode sesi, range jam. Data inputan tersebut lalu disimpan kedalam database dan kemudian ditampilkan pada *Gridview*. Apabila kita mau merubah data yang sudah ada maka pilih salah satu data yang ada pada *Gridview* tadi, kemudian *field* yang ada diganti setelah itu di klik button simpan yang beralih fungsi menjadi button *Update*.

Berikut *interface* hari dan waktu dapat dilihat pada gambar 9. sebagai berikut:

kode	nama
1	Senin
2	Selasa
3	Rabu
4	Kamis
5	Jumat
6	Sabtu

kode	range_jam
1	07.00-09.00
2	09.00-11.00
3	11.00-13.00
4	13.00-15.00
5	15.00-17.00

Gambar 9. *Interface* Hari dan Waktu

Interface Waktu Tidak Bersedia

Submenu ini memberikan pilihan waktu untuk tidak mengajar pada saat- saat dosen tersebut berhalangan untuk hadir. inputan berupa data dosen yang diambil dari table dosen, kemudian pada *Gridview* menentukan ketidak sediaan hari dan sesi dengan

inputan Ceckbox. Sedangkan untuk kode dibuat *Auto Increment*. Data inputan tersebut lalu disimpan kedalam table waktu_tidak_bersedia. Apabila kita mau merubah data yang sudah ada maka pilih salah satu data yang ada pada *Gridview* tadi, kemudian *field* yang ada diganti setelah itu di klik button simpan yang beralih fungsi menjadi button *Update*. Untuk form interfacenya lihat gambar dibawah ini :

Hari	Jam	Tidak Bersedia
Senin	07.00-09.00	<input type="checkbox"/>
Senin	09.00-11.00	<input type="checkbox"/>
Senin	11.00-13.00	<input type="checkbox"/>
Senin	13.00-15.00	<input type="checkbox"/>
Senin	15.00-17.00	<input type="checkbox"/>
Selasa	07.00-09.00	<input type="checkbox"/>
Selasa	09.00-11.00	<input type="checkbox"/>
Selasa	11.00-13.00	<input type="checkbox"/>
Selasa	13.00-15.00	<input type="checkbox"/>
Selasa	15.00-17.00	<input type="checkbox"/>
Rabu	07.00-09.00	<input type="checkbox"/>
Rabu	09.00-11.00	<input type="checkbox"/>
Rabu	11.00-13.00	<input type="checkbox"/>
Rabu	13.00-15.00	<input type="checkbox"/>
Rabu	15.00-17.00	<input type="checkbox"/>
Kamis	07.00-09.00	<input type="checkbox"/>
Kamis	09.00-11.00	<input type="checkbox"/>
Kamis	11.00-13.00	<input type="checkbox"/>
Kamis	13.00-15.00	<input type="checkbox"/>
Kamis	15.00-17.00	<input type="checkbox"/>
Jumat	07.00-09.00	<input type="checkbox"/>
Jumat	09.00-11.00	<input type="checkbox"/>
Jumat	11.00-13.00	<input type="checkbox"/>

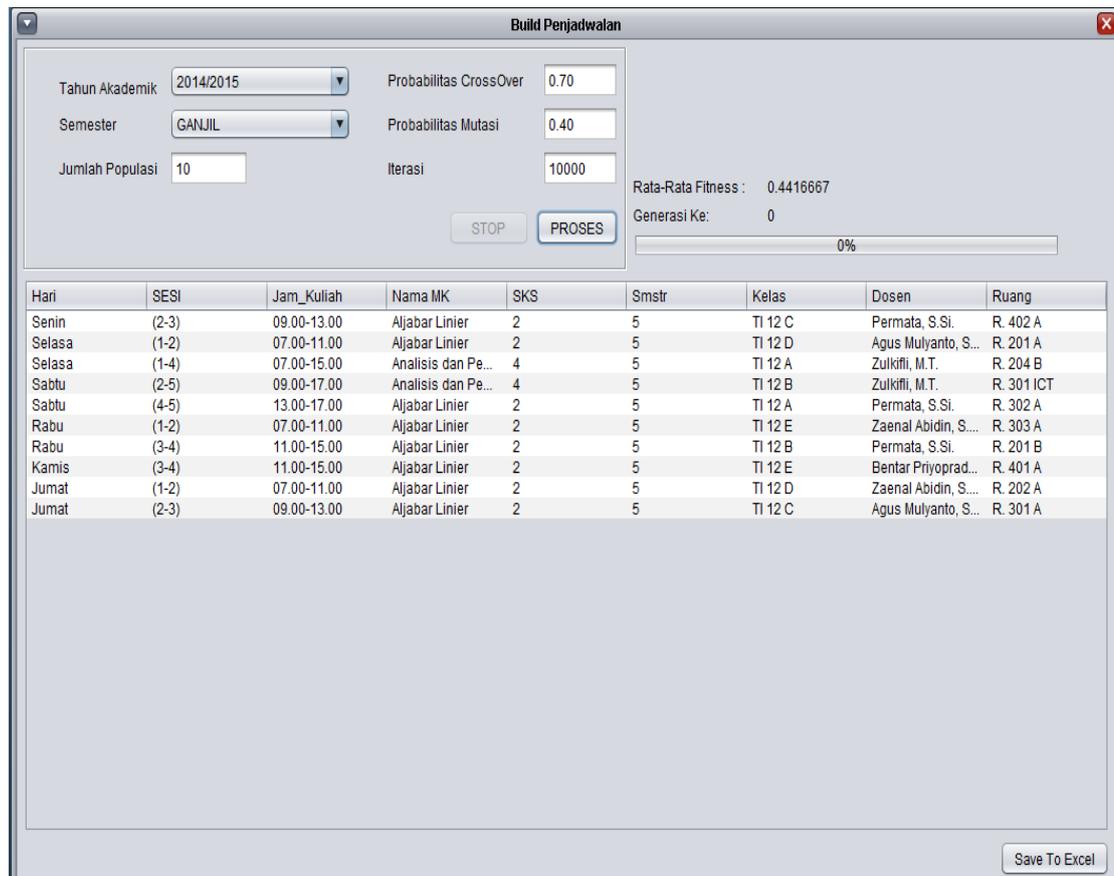
Gambar 10. *Interface* proses waktu tidak bersedia

***Interface* Proses Penjadwalan**

Pada form inilah proses penjadwalan dilakukan, data-data dari semua table yang ada untuk diproses agar menghasilkan jadwal matakuliah yang optimal. Sebelum dilakukan proses penjadwalan harus dilakukan pengaturan tahun akademik ,semester dan kumpulan parameter yang secara default dimuat dari file *config.ini*. Parameter tersebut antara lain :

1. Populasi : Jumlah jadwal dalam setiap generasi
2. Probabilitas *Crossover* : Menentukan nilai random yang dilakukan Proses *Crossover*
3. Probabilitas Mutasi : Menentukan nilai random yang dilakukan Proses Mutasi
4. Iterasi : Menentukan beberapa kali perulangan Proses Algoritma Genetika dilakukan.

Untuk form interfacenya lihat gambar dibawah ini :



Gambar 11 Interface Proses Penjadwalan

Untuk Pseudocode program pada gambar dibawah ini :

```

@SuppressWarnings("empty-statement")
private void btnProsesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //jTable1.setModel(null);
    final int GANJIL = 1;
    final int GENAP = 0;
    found = false;
    int jenis_semester = cmbSemester.getSelectedItem().toString().equals("GANJIL") ? GANJIL : GENAP;
    String tahun_akademik = cmbTahunAkademik.getSelectedItem().toString();
    populasi = Integer.parseInt(txtPopulasi.getText());
    if (populasi % 2 != 0) {
        JOptionPane.showMessageDialog(null, "Populasi harus kelipatan 2");
        return;
    }
    float crossOver = Float.parseFloat(txtCrossOver.getText());
    float mutasi = Float.parseFloat(txtMutasi.getText());
    maxIterasi = Integer.parseInt(txtIterasi.getText());
    genetik = new ClassGenetik(jenis_semester, tahun_akademik, populasi,
        crossOver, mutasi, kode_jumat, range_jumat, kode_dhuhur);

    try {
        final long startTime = System.currentTimeMillis();
        if (!Objects.equals(genetik.AmbilData(), TRUE)) {
            JOptionPane.showMessageDialog(null, "Kemungkinan data tidak ada untuk Tahun Akademik dan Sem
            return;
        }
    };
    genetik.Inisialisasi();
}

```

Gambar 12. Pseudocode Form Proses Penjadwalan

Dari pseudocode diatas bahwa form ini akan memproses seluruh komponen penjadwalan seperti dosen, matakuliah, hari, sesi, dan ruangan dengan memperhitungkan konfigurasi parameter yang sudah ditentukan untuk diperoleh sebuah solusi jadwal yang optimal. Jika solusi jadwal sudah berhasil ditemukan maka akan muncul pemberitahuan berupa windows box, yang berisi tentang lama pemrosesan penjadwalan. Dan kemudian hasil dari jadwal tersebut disimpan dalam format excel.

SIMPULAN DAN SARAN

Simpulan

Dari penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa Algoritma Genetika dapat digunakan sebagai salah satu alternatif solusi untuk menyelesaikan masalah penjadwalan perkuliahan yang selama ini masih dilakukan secara manual, dengan menggunakan parameter hasil uji coba yaitu kumpulan jadwal 10, probabilitas pertukaran 0.70 dan probabilitas perandoman 0.40, dan maksimal iterasi 10000. Algoritma Genetika dapat mengoptimalkan batasan–batasan yang ditentukan tanpa adanya constrain yang terlanggar dengan waktu rata–rata kurang dari 1 menit dalam setiap proses penjadwalan.

Saran

Karena ruang lingkup aplikasi ini masih sempit yaitu hanya penelitian pada Jurusan Teknik Informatika saja, dan untuk constrain-constrainnya pun masih perlu ditambah lagi seperti adanya satu sesi kosong selesai mata kuliah sebelumnya. maka untuk selanjutnya aplikasi penjadwalan ini bisa dikembangkan lagi untuk ruang lingkup yang lebih besar, misal untuk lingkup S-1 keseluruhan dan D3 di Perguruan Tinggi tersebut, karena constrain yang ada pada tingkat tersebut tentunya lebih kompleks lagi.

Kemudian pada hasil proses penjadwalan masih perlu adanya perbaikan didalam penyajiannya. Karena hasil dari proses penjadwalan pada penelitian ini masih dalam tampilan yang sederhana dan belum dibuat plotting. Tampilan jadwal hanya dibuat table dalam format Excel dengan susunan kolom header yang terdiri dari kolom hari, Sesi, jam kuliah, nama matakuliah, sks, semester, kelas, dosen dan ruangan. Sehingga masih perlu dibuat plotting untuk mempermudah didalam pembuktiannya.

REFERENSI

- Ariany, F. (n.d.). *Hibridisasi Algoritme Genetika dan Tabu Search pada Sistem Penjadwalan Perkuliahan (Studi Kasus di Universitas Teknokrat Indonesia)*. Bogor Agricultural University (IPB).
- Ariyanti, L., Satria, M. N. D., & Alita, D. (2020). Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 90–96.
- Audrilia, M., & Budiman, A. (2020). Perancangan Sistem Informasi Manajemen Bengkel Berbasis Web (Studi Kasus: Bengkel Anugrah). *Jurnal Madani: Ilmu Pengetahuan, Teknologi, Dan Humaniora*, 3(1), 1–12.
- Ayu, M., Sari, F. M., & Muhaqiqin, M. (2021). Pelatihan Guru dalam Penggunaan Website

Grammar Sebagai Media Pembelajaran selama Pandemi. *Al-Mu'awanah: Jurnal Pengabdian Kepada Masyarakat*, 2(1), 49–55.

- Ayunandita, N., & Riskiono, S. D. (2021). PERMODELAN SISTEM INFORMASI AKADEMIK MENGGUNAKAN EXTREME PROGRAMMING PADA MADRASAH ALIYAH (MA) MAMBAUL ULUM TANGGAMUS. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 2(2).
- Budiman, A., Wahyuni, L. S., & Bantun, S. (2019). Perancangan Sistem Informasi Pencarian Dan Pemesanan Rumah Kos Berbasis Web (Studi Kasus: Kota Bandar Lampung). *Jurnal Tekno Kompak*, 13(2), 24–30.
- Dinasari, W., Budiman, A., & Megawaty, D. A. (2020). SISTEM INFORMASI MANAJEMEN ABSENSI GURU BERBASIS MOBILE (STUDI KASUS: SD NEGERI 3 TANGKIT SERDANG). *Jurnal Teknologi Dan Sistem Informasi*, 1(2), 50–57.
- Fatimah, C., Wirnawa, K., & Dewi, P. S. (2020). Analisis Kesulitan Belajar Operasi Perkalian Pada Siswa Sekolah Menengah Pertama (Smp). *Jurnal Ilmiah Matematika Realistik*, 1(1), 1–6.
- Giovani, A. P., Ardiansyah, A., Haryanti, T., Kurniawati, L., & Gata, W. (2020). Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi. *Jurnal Teknoinfo*, 14(2), 115. <https://doi.org/10.33365/jti.v14i2.679>
- Gunawan D. (2020). *Komparasi Algoritma Support Vector Machine Dan Naïve Bayes Dengan Algoritma Genetika Pada Analisis Sentimen Calon Gubernur Jabar 2018-2023*. V(1), 135–138. <https://doi.org/10.31294/jtk.v4i2>
- Hasan, L. O. M., Sholeh, M., & Iswahyudi, C. (2016). Pemanfaatan Twitter Api Untuk Mengakses Jadwal Bimbingan Dosen Pada Ist Akprind Yogyakarta. *Jurnal SCRIPT*, 3(2), 139–147.
- Kadir, A. (2003). *Dasar Pemrograman web dinamis menggunakan PHP*.
- Maskar, S., & Dewi, P. S. (2021). PENINGKATAN KOMPETENSI GURU MA DARUR RIDHO AL-IRSYAD AL ISLAMIYYAH PADA PEMBELAJARAN DARING MELALUI MOODLE. *Journal of Social Sciences and Technology for Community Service (JSSTCS)*, 2(1), 1–10.
- Maskar, S., Dewi, P. S., & Puspaningtyas, N. D. (2020). Online Learning & Blended Learning: Perbandingan Hasil Belajar Metode Daring Penuh dan Terpadu. *PRISMA*, 9(2), 154–166.
- Megawaty, D A. (2020). *SISTEM MONITORING KEGIATAN AKADEMIK SISWA*.
- Megawaty, Dyah Ayu, & Putra, M. E. (2020). Aplikasi Monitoring Aktivitas Akademik Mahasiswa Program Studi Informatika Universitas Xyz Berbasis Android. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 1(1), 65–74.
- Monica, T., & Borman, R. I. (2017). Implementasi Konsep Media Sosial Dalam Sistem

- Informasi Kegiatan Kesiswaan (Studi Kasus: SMK XYZ). *Jurnal Tekno Kompak*, 11(2), 33–37.
- Nurkholis, A., & Sitanggang, I. S. (2020). Optimalisasi model prediksi kesesuaian lahan kelapa sawit menggunakan algoritme pohon keputusan spasial. *Jurnal Teknologi Dan Sistem Komputer*, 8(3), 192–200.
- Oktaviani, L., & Ayu, M. (2021). Pengembangan Sistem Informasi Sekolah Berbasis Web Dua Bahasa SMA Muhammadiyah Gading Rejo. *Jurnal Pengabdian Pada Masyarakat*, 6(2), 437–444.
- Optimasi Arsip Penyimpanan Dokumen Foto Menggunakan Algoritma Kompresi Deflate (Studi Kasus: Studio Muezzart) Bahrudin, A., Permata, P., & Jupriyadi, J. (2020). Optimasi Arsip Penyimpanan Dokumen Foto Menggunakan Algoritma Kompresi Deflate (Studi Kasus: Studio Muezzart). *Jurnal Ilmiah Infrastruktur Teknologi Informasi*, 1(2), 14–18.
- Parinata, D., & Puspaningtyas, N. D. (2021). Optimalisasi Penggunaan Google Form terhadap Pembelajaran Matematika. *MATHEMA: JURNAL PENDIDIKAN MATEMATIKA*, 3(1), 56–65.
- Puspaningrum, A. S. (2017). *Pengukuran Kesesuaian Fungsional Dengan Pendekatan Berorientasi Tujuan Pada Sistem Informasi Akademik (SIA) Berdasarkan Model Kualitas ISO/IEC 25010*. Institut Teknologi Sepuluh Nopember.
- Puspaningrum, A. S., Neneng, N., Saputri, I., & Ariany, F. (2020). PENGEMBANGAN E-RAPORT KURIKULUM 2013 BERBASIS WEB PADA SMA TUNAS MEKAR INDONESIA. *Jurnal Komputasi*, 8(2), 94–101.
- Putra, A. D., Suryono, R. R., & Darmi, D. (2009). Rancang bangun media pembelajaran TOEFL berbasis web. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Raharjo, B. (2016). *Modul Pemrograman WEB (HTML, PHP, MySQL)* (3rd ed.).
- Ramadhanu, P. B., & Priandika, A. T. (2021). RANCANG BANGUN WEB SERVICE API APLIKASI SENTRALISASI PRODUK UMKM PADA UPTD PLUT KUMKM PROVINSI LAMPUNG. *Jurnal Teknologi Dan Sistem Informasi*, 2(1), 59–64.
- Samsugi, S., & Burlian, A. (2019). Sistem penjadwalan pompa air otomatis pada aquaponik menggunakan mikrokontrol Arduino UNO R3. *PROSIDING SEMNASTEK 2019*, 1(1).
- Setiawansyah, S., Sulistiani, H., & Saputra, V. H. (2020). Penerapan Codeigniter Dalam Pengembangan Sistem Pembelajaran Dalam Jaringan Di SMK 7 Bandar Lampung. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, 6(2), 89–95.
- Suri, M. I., & Puspaningrum, A. S. (2020). Sistem Informasi Manajemen Berita Berbasis Web. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 8–14.
- Ulfa, M., & Saputra, V. H. (2019). PENGARUH MEDIA PEMBELAJARAN

MAKROMEDIA FLASH DENGAN PENDEKATAN MATEMATIKA REALISTIK
PADA HASIL BELAJAR SISWA. *Triple S (Journals of Mathematics Education)*,
2(1), 12–21.