

SISTEM INFORMASI PENJADWALAN ASISTEN LABORATORIUM MENGUNAKAN ALGORITMA GENETIKA PADA LABORATORIUM KOMPUTER STMIK WIDYA CIPTA DHARMA

*Mohammad Febrian Triadi ¹⁾, Heny Pratiwi ²⁾, Ita Arfyanti ³⁾

^{1,2,3)}Sistem Informasi, STMIK Widya Cipta Dharma

Jl. M. Yamin, Gn. Kelua, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75123

Email : febriant.34@gmail.com

ABSTRAK

Penelitian dilakukan untuk membuat sebuah aplikasi penjadwalan asisten laboratorium STMIK Widya Cipta Dharma yang dimana jika penelitian ini berhasil dapat membantu staf lab dalam melakukan penjadwalan asisten laboratorium STMIK Widya Cipta Dharma. Penelitian ini dilakukan di laboratorium STMIK Widya Cipta Dharma. Metode pengumpulan data yang digunakan yaitu dengan wawancara kepada kepala laboratorium dan staf laboratorium yang berkaitan tentang penjadwalan asisten laboratorium di STMIK Widya Cipta Dharma. Dengan cara observasi, yaitu pengamatan cara kerja penjadwalan asisten sebelumnya, aspek yang menjadi acuan penjadwalan yang nantinya akan digunakan dalam pembuatan aplikasi penjadwalan. Dalam penelitian ini, metode pengembangan sistem yang digunakan yaitu black box dan beta test dengan perangkat lunak pendukung yang digunakan yaitu laragon 6.0 dan php 8.3.9. Adapun hasil akhir dari penelitian ini yaitu berupa aplikasi penjadwalan asisten menggunakan algoritma genetika pada laboratorium STMIK Widya Cipta Dharma yang dapat memudahkan staf lab dalam melakukan penjadwalan asisten lab.

Kata Kunci: Penjadwalan, Penjadwalan Asisten, Algoritma Genetika.

1. PENDAHULUAN

Sistem informasi merupakan suatu sistem yang menggunakan teknologi informasi dan basis data untuk mengumpulkan, mengelola, dan menyediakan informasi yang relevan dan akurat bagi pengguna. Pada konteks laboratorium komputer, sistem informasi penjadwalan menjadi penting untuk mengatur jadwal praktikum dan penggunaan asisten laboratorium (aslab) dengan efisien.

Penjadwalan aslab merupakan tugas yang kompleks dan membutuhkan perencanaan yang baik. Proses penjadwalan melibatkan penempatan aslab ke dalam jadwal praktikum yang sesuai dengan keahlian, preferensi, dan ketersediaan mereka.

Melakukan penjadwalan secara manual adalah rentan terhadap kesalahan manusia dan sulitnya mempertimbangkan berbagai faktor yang mempengaruhi penjadwalan. Proses manual seringkali memakan waktu yang lama, terutama ketika harus menyesuaikan jadwal praktikum dengan preferensi asisten laboratorium dan memastikan tidak adanya tumpang tindih jadwal.

Salah satu alasan utama dalam pembuatan sistem informasi penjadwalan adalah untuk meningkatkan efisiensi pengelolaan jadwal praktikum dan penempatan asisten laboratorium. Dengan menggunakan sistem informasi penjadwalan, pengelola laboratorium dapat menghemat waktu dan upaya dalam menyusun jadwal praktikum yang optimal, mempertimbangkan preferensi dan ketersediaan asisten laboratorium. Sistem ini juga dapat membantu dalam

menghindari tumpang tindih jadwal, mengoptimalkan pemanfaatan sumber daya, dan meningkatkan efisiensi proses penjadwalan.

Penerapan algoritma genetika dalam pembuatan sistem informasi penjadwalan menjadi pilihan yang tepat karena algoritma ini dapat mencari solusi jadwal yang optimal dengan mempertimbangkan berbagai batasan dan

preferensi yang ada. Algoritma genetika meniru proses seleksi alam dan evolusi, sehingga mampu mencapai solusi yang lebih baik dari waktu ke waktu. Dalam konteks penjadwalan asisten laboratorium, penerapan algoritma genetika memungkinkan sistem informasi penjadwalan untuk mencari solusi jadwal yang meminimalkan konflik, memenuhi preferensi asisten laboratorium, dan mengoptimalkan pemanfaatan sumber daya.

Dengan adanya sistem informasi penjadwalan asisten laboratorium dapat meningkatkan efisiensi, akurasi, dan transparansi dalam penjadwalan. Dengan menggunakan sistem ini, pengelola laboratorium dapat meminimalkan kesalahan dalam penempatan asisten laboratorium, mempertimbangkan preferensi dan ketersediaan mereka secara lebih baik, serta meningkatkan kualitas layanan praktikum. Sistem informasi penjadwalan juga memungkinkan laboratorium komputer untuk mengoptimalkan pemanfaatan sumber daya, menghindari tumpang tindih jadwal, dan menjaga keteraturan jadwal praktikum.

2. RUANG LINGKUP

Agar pembahasan dalam penelitian ini tidak meluas lebih dari pembahasan yang dimaksud, dan agar dapat lebih jelas serta terarah maka dibuatlah Batasan masalah ini untuk membatasinya pada ruang lingkup penelitian sebagai berikut:

1. Penelitian ini hanya membahas tentang pembuatan sistem informasi penjadwalan asisten laboratorium pada Laboratorium Komputer (Labkom) STMIK Widya Cipta Dharma.
2. Penelitian ini hanya menggunakan Algoritma Genetika sebagai metode optimisasi dalam menjadwalkan Asisten Laboratorium (Aslab).

3. Penelitian ini akan melibatkan pengumpulan data mengenai preferensi asisten laboratorium, ketersediaan laboratorium, dan kebutuhan praktikum sebagai masukan dalam sistem informasi penjadwalan
4. Penelitian ini akan mengabaikan faktor-faktor lain yang tidak langsung terkait dengan penjadwalan asisten laboratorium, seperti perencanaan kurikulum, penentuan jumlah peserta praktikum, dan pemilihan perangkat lunak yang akan digunakan dalam praktikum.
5. Evaluasi kinerja sistem informasi penjadwalan akan dilakukan berdasarkan kriteria efisiensi waktu, pengurangan konflik jadwal, dan kepuasan pengguna sistem, terutama asisten laboratorium.
- 6.
7. Penelitian ini hanya mencakup bagian-bagian yang dibangun seperti:
 - 1) Login
 - 2) Jadwal Praktikum
 - 3) Ketersediaan Asisten Lab
 - 4) Hari
 - 5) Ruang
 - 6) Penjadwalan Asisten
 - 7) Logout

3. BAHAN DAN METODE

3.1 Sistem

Sistem adalah seperangkat komponen yang saling berhubungan dan saling bekerja sama untuk mencapai beberapa tujuan (Agustin, 2018). Sedangkan ada pendapat lain juga mengatakan, sistem adalah suatu kesatuan yang terdiri dari unsur-unsur yang saling berhubungan dan berinteraksi dalam mencapai tujuan tertentu (Nursalim, 2016).

3.2 Informasi

Informasi adalah hasil dari pengolahan data yang diperoleh melalui proses pemilihan, pengorganisasian, dan penginterpretasian data, sehingga memiliki nilai tambah dan kegunaan dalam mendukung pengambilan keputusan (Utomo, 2016).

Sedangkan menurut pendapat lain juga, Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang (Agustin, 2018).

3.3 Sistem Informasi

Sistem informasi (SI) adalah suatu sistem yang terdiri dari orang, proses, teknologi, dan data yang saling berinteraksi untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi yang dibutuhkan untuk mendukung pengambilan keputusan dan operasi bisnis. SI dapat digunakan dalam berbagai bidang, seperti bisnis, pemerintahan, pendidikan, kesehatan, dan lain sebagainya.

Menurut Laudon dan Laudon (2018), SI dapat dibagi menjadi empat jenis, yaitu sistem informasi transaksional, sistem informasi manajemen, sistem informasi keputusan, dan sistem informasi eksekutif. Sistem informasi transaksional digunakan untuk mengelola transaksi bisnis sehari-hari. Sistem informasi manajemen digunakan untuk membantu manajemen dalam mengambil keputusan strategis dan operasional.

Sistem informasi keputusan digunakan untuk membantu pengambilan keputusan yang kompleks dan tidak terstruktur. Sistem informasi eksekutif digunakan untuk membantu eksekutif dalam mengambil keputusan strategis.

3.4 Penjadwalan

Penjadwalan merupakan sebuah fungsi pengambilan keputusan yang biasa digunakan banyak perusahaan manufaktur atau jasa yang berhubungan dengan alokasi sumber daya yang mengerjakan tugas selama waktu tertentu dan tujuannya adalah untuk mengoptimalkan satu atau lebih dengan tujuan pengoptimalan (Pinedo, 2016).

3.5 Website

Website merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi, sehingga lebih merupakan media informasi yang menarik untuk dikunjungi. Berdasarkan kategorinya, website digolongkan menjadi 3 (Adelheid, 2015) :

Website yang mempunyai halaman yang tidak berubah. Artinya adalah melakukan perubahan pada suatu halaman dilakukan dengan cara manual dengan mengubah kode yang menjadi struktur dari website tersebut

Website secara struktur diperuntukkan untuk update sesering mungkin. Biasanya selain utama yang dapat diakses oleh user pada umumnya, juga disediakan halaman backend untuk mengubah konten dari website. Contoh umum mengenai website dinamis adalah web berita atau web portal yang didalamnya terdapat fasilitas berita, polling dan sebagainya.

Salah satu contoh website interaktif adalah blog dan forum. Di website ini user dapat berinteraksi dan beradu argument mengenai apa yang menjadi pemikirannya. Biasanya website seperti memiliki moderator untuk mengatur supaya topik yang diperbincangkan tidak keluar jalur.

3.6 HTML

HTML singkatan dari (Hypertext Markup Language) adalah sebuah bahasa formatting yang digunakan untuk membuat sebuah halaman website. Di dalam dunia pemrograman berbasis website (Web Programming), HTML menjadi pondasi dasar pada halaman website. Sebuah file HTML disimpan dengan ekstensi .html (dot html). Dan dapat dieksekusi atau diakses menggunakan web browser (Google Chrome, Mozilla Firefox dan lain-lainnya) (Hadi, 2016).

3.7 Bahasa Pemrograman PHP

Menurut Saputra (2018) Penulis buku yang berjudul “Project PHP : Menyelesaikan Project 30 Juta Secara Profesional” PHP adalah singkatan dari PHP *Hypertext Preprocessor*, dimana huruf pemrograman *server-side* yang paling populer dibanding ASP.NET maupun dengan *java* yang digunakan dalam membuat *website* karena mudah untuk dipelajari. PHP sering dipakai para programmer untuk membuat situs web yang bersifat dinamis karena gratis dan berguna dalam merancang aplikasi web.

3.8 CSS (Cascading Style Sheet)

CSS merupakan singkatan dari “Cascading Style Sheet”, sesuai dengan namanya CSS memiliki sifat “*style sheet language*” yang berarti bahasa pemrograman yang digunakan untuk web *design*. CSS adalah bahasa pemrograman yang digunakan untuk mendesain sebuah halaman *website*. Dalam mendesain halaman *website*, CSS menggunakan penandaan yang dikenal dengan *id* dan *class* (Hadi, 2016)

3.9 BasisData (Database)

Menurut Ladjamudin (2013), Database adalah sekumpulan data store (dapat dalam jumlah besar) yang tersimpan dalam magnetic disk, optical disk, magnetic drum, atau media penyimpanan sekunder lainnya. Selain itu juga basis data juga dapat didefinisikan sebagai kumpulan file/tabel/arsip yang saling berhubungan yang disimpan di media penyimpanan elektronik. Seperti penyimpanan dokumen berisi data dalam file teks, file spread sheet dan lain-lain. MySQL

Menurut Arief (2013), MySQL (My Structure Query Language) adalah “salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber data dan pengelolaan datanya”. MySQL bersifat open source dan menggunakan SQL (Structured Query Language) MySQL dapat dijalankan diberbagai platform misalnya Windows, Linux, dan lain sebagainya.

3.10 Algoritma Genetika

Menurut Assagaf, dkk, (2018), Algoritma genetika (AG) adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Metode Algoritma Genetika ditemukan di Universitas Michigan, Amerika Serikat oleh Holland, (1975) melalui sebuah penelitian dan dipopulerkan oleh salah satu muridnya, Goldberg, (1989).

Terdapat 8 komponen utama dalam algoritma genetika, yaitu :

1. Teknik Penyandian

Teknik penyandian disini meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili satu variabel. Variabel adalah suatu sebutan yang dapat diberi nilai angka (kuantitatif) atau nilai mutu (kualitatif). Variabel merupakan pengelompokan secara logis dari dua atau lebih atribut dari objek yang diteliti. Gen dapat direpresentasikan dalam bentuk *string*, *bit*, pohon, *array* bilangan *real*, daftar aturan, elemen permutasi, elemen program, atau representasi lainnya yang dapat diimplementasikan untuk operator genetika.

String Biner

0 1 0 | 1 0 0 1 | 0 1 1 1

Gen 1 Gen 2 Gen 3

Gambar 2.4 Representasi Variabel

Sumber : Sutabri, T. (2012). Konsep sistem informasi Penerbit Andi.

2. Fungsi Fitness

Fungsi *Fitness*, alat ukur yang digunakan untuk proses evaluasi kromosom. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut.

$$f = \frac{\Sigma \text{Ketersediaan Asisten Lab}}{\text{Hari}}$$

Gambar 2.5 Rumus Fungsi Fitness

Sumber : Sutabri, T. (2012). Konsep sistem informasi Penerbit Andi

3. Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi di tentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan *domain* solusi dan kendala permasalahan yang ada.

4. Pembangkit Populasi Awal

Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui literasi yang disebut generasi.

5. Fungsi Evaluasi

Ada 2 hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Fungsi objektif adalah fungsi yang nilainya akan dioptimalkan. Fungsi objektif bisa bernilai maksimum dan minimum tergantung pada kasusnya. Jika fungsi objektif biaya produksi, maka nilainya dicari yang minimum. Tapi kalau fungsi objektifnya berupa keuntungan, maka nilainya yang maksimum.

6. Kriteria optimasi tercapai

Beberapa kriteria berhenti yang sering digunakan antara lain :

1) Berhenti pada generasi tertentu.

2) Berhenti setelah beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi/terendah.

3) Berhenti bila dalam n generasi berikutnya tidak diperoleh nilai *fitness* yang lebih tinggi/rendah

7. Operator Genetika

Operator standar yang biasa digunakan dalam algoritma genetika adalah *selection*, *crossover* dan *mutation*. Berikut ini akan dijelaskan masing-masing operator yaitu :

1) Seleksi

Seleksi bertujuan memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling sesuai. Langkah pertama dalam seleksi ini adalah pencarian nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung pada nilai objektif dirinya sendiri terhadap semua individu dalam wadah seleksi tersebut. Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Fungsi ini dinamakan fungsi *fitness*. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup. Sedangkan individu yang bernilai *fitness* rendah akan mati. Dalam algoritma genetika, fungsi *fitness* adalah fungsi objektif dari masalah yang akan dioptimasi. Fungsi ini sebagai ukuran keuntungan yang ingin dimaksimalkan atau sebagai ukuran biaya yang ingin diminimumkan.

2) Rekombinasi

Crossover (perkawinan silang) bertujuan menambah keanekaragaman *string* dalam suatu populasi dengan penyilangan antar *string* yang diperoleh dari reproduksi sebelumnya.

3) Mutasi

Mutasi merupakan proses mengubah nilai dari satu atau beberapa generasi dalam kromosom. Operasi *crossover* yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih.

8. Generasi Terakhir

Algoritma genetika adalah algoritma pencarian hasil yang terbaik, yang didasarkan pada perkawinan silang (*crossover*) dan seleksi generasi secara alami. Setelah beberapa generasi maka algoritma genetika akan berada pada generasi terakhir dimana menghasilkan kromosom terbaik, yang diharapkan menghasilkan individu baru.

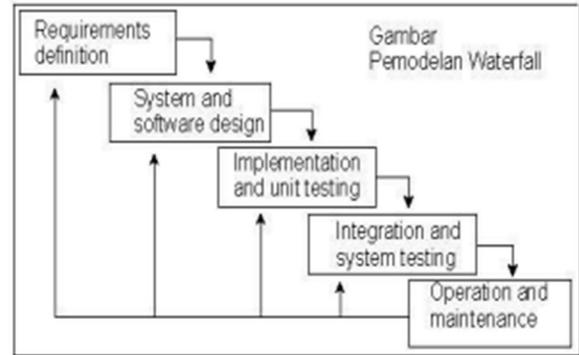
Algoritma genetika membutuhkan beberapa nilai parameter yang menentukan kinerja program. Parameter yang biasa digunakan pada algoritma genetika adalah :

- 1) *Population size*, jumlah individu yang dilibatkan pada setiap generasi.
- 2) *Crossover rate* merupakan rasio perbandingan banyaknya *offspring* yang diproduksi pada tiap generasi dengan banyaknya *population size*.
- 3) *Mutation rate*, merupakan representasi terjadinya kromosom baru sebagai akibat mutasi, dari keseluruhan *population size*. Jika *mutation rate* ini terlalu rendah, banyak kemungkinan solusi yang tidak akan dicoba sementara.

3.11 WaterFall

Waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software dari masing-masing tahapan yaitu *communication*, *planning*, *modeling*, *construction* dan *deployment* serta hasil dari

masing-masing tahapan dalam membangun sistem *point of sale* (Pressman, 2014),..



Gambar 2.6 Model *Waterfall*

Sumber : Nabyla, F. (2021). Sistem Informasi Penjadwalan Laboratorium Berbasis Web.

1. *Requirements Definition* (Analisa Kebutuhan)

Tahapan ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan mencari informasi sebanyak mungkin dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen ini yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrograman.

2. *System and Software Design* (Desain Sistem)

Tahapan yang dimana membutuhkannya ide dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (*data flow diagram*), diagram hubungan entitas (*entity relationship diagram*) serta struktur dan bahasan data.

3. *Implementation and Unit Testing* (Penulisan Kode Program)

Penulisan kode program atau coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh programmer yang akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4. *Integration and System Testing* (Pengujian Program)

Tahap pengujian merupakan tahapan akhir yang dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan

kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna. Pengujian dilakukan menggunakan *Black*

Box. Dengan melakukan pengujian *Black Box*, programmer dapat memastikan bahwa tidak akan ada *error* pada program aplikasi yang telah dibuat. *Black Box* testing berfokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi *input* dan melakukan pengujian pada fungsional program.

5. *Operation and Maintenance* (Penerapan Program dan Pemeliharaan)

Perangkat lunak yang sudah diberikan kepada pengguna pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

3.12 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan dalam rekayasa perangkat lunak untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak sistem perangkat lunak. Ini menyediakan serangkaian teknik notasi grafis untuk membuat model visual sistem intensif perangkat lunak berorientasi objek. UML umumnya digunakan untuk merancang sistem perangkat lunak dan didasarkan pada paradigma berorientasi objek. Ini mencakup serangkaian teknik notasi grafis untuk membuat model visual sistem intensif perangkat lunak berorientasi objek. UML banyak digunakan dalam industri pengembangan perangkat lunak dan didukung oleh *Object Management Group (OMG)* (Putra, dkk., 2019).

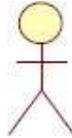
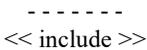
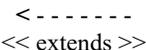
1. *Usecase Diagram*

Usecase digunakan untuk menggambarkan fungsi dasar dari sebuah sistem informasi. *Use case* mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya (Dennis, dkk., 2015). Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut :

Tabel 1 *Usecase Diagram*

Gambar	Keterangan
	<i>Use case</i> adalah bagian utama dari fungsionalitas sistem. Bisa <i>extend</i> (memperluas) <i>use case</i> lainnya. Ditempatkan di dalam system <i>boundary</i> (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.

Tabel 1 *Usecase Diagram*(Lanjut)

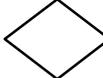
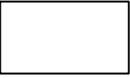
Gambar	Keterangan
	<i>Actor/role</i> adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini.
	Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas <i>boundary</i> . Mewakili ruang lingkup sistem. <i>Actor</i> berada di luar ruang lingkup sistem.
	Menghubungkan actor dengan <i>use case</i> . Menunjukkan komunikasi dua arah (Menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda * untuk keragaman dari asosiasi (<i>multiplicity of the association</i>).
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat.

Sumber : Indriyani dkk., (2019), Analisa Perancangan Sistem Informasi.

2. Activity Diagram

Activity diagram mengilustrasikan kegiatan utama dan hubungan di antara kegiatan dalam suatu proses (Dennis, dkk., 2015). Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2 Activity Diagram

Gambar	Keterangan
	<i>Initial Node</i> , Menggambarkan awal dari serangkaian tindakan atau kegiatan.
	<i>Final-activity Node</i> , Digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).
	<i>Final-flow Node</i> , Digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.
	<i>Activity</i> , Digunakan untuk mewakili sekumpulan tindakan (action). Dilabeli dengan namanya.
	<i>Action</i> , Perilaku yang sederhana dan tidak dapat diuraikan. Dilabeli dengan namanya.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Object Node</i> , Digunakan untuk mewakili sebuah objek yang terhubung ke sekumpulan arus objek. Dilabeli dengan nama classnya.
	<i>Object Flow</i> , Menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke aktivitas (atau tindakan) lain.
	<i>Swimlane</i> , Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan). Dilabeli dengan nama individu atau objek yang bertanggung jawab.

Sumber : Indriyani dkk., (2019), Analisa Perancangan Sistem Informasi.

3. Sequence Diagram

Sequence diagram menunjukkan pesan yang lewat di antara objek untuk *use case* tertentu dari waktu ke waktu (Dennis, dkk., 2015). *Sequence diagram* mengilustrasikan objek-objek yang berpartisipasi di dalam suatu *use case*. Adapun Simbol-simbol yang digunakan dalam *sequence diagram* adalah:

Tabel 3 Sequence Diagram

Gambar	Keterangan
	<i>Life Line</i> , objek entity antarmuka yang saling berinteraksi.
	<i>Object Message</i> , spesifikasi dari komunikasi antar objek yang membuat informasi-informasi tentang aktivitas yang terjadi.
	<i>Message (Return)</i> , Menyatakan kembali dalam 1 <i>object Life Line</i> .
	<i>Message</i> , simbol mengirim pesan antar class Menyatakan arah kembali antara <i>Life Line</i> .
	<i>Activation</i> , menyatakan objek dalam keadaan aktif.
	<i>Message (Destroy)</i> , menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika <i>create</i> maka ada <i>destroy</i> .
	<i>Actor</i> , menspesifikasikan himpunan peran yang pengguna ketika berinteraksi dengan sistem.

Sumber : Indriyani dkk., (2019), Analisa Perancangan Sistem Informasi.

3.13 Blackbox Testing

Pengujian black box atau dapat disebut juga behavioral testing adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik. Untuk melakukan pengujian, penguji tidak harus memiliki kemampuan menulis kode program. Pengujian ini dapat dilakukan oleh siapa saja. Dalam pengujian Black Box, ada beberapa teknik yang dapat digunakan seperti :

1. All pair testing

Teknik *all pair testing* ini dikenal juga dengan *pairwise testing*. Pengujian ini digunakan untuk menguji semua kemungkinan kombinasi dari seluruh pasangan berdasarkan *input* parameternya.

2. *Boundary value analysis*
Teknik ini berfokus pada pencarian *error* dari luar atau sisi dalam perangkat lunak.
3. *Cause-effect graph*
Berikutnya adalah teknik *cause-effect graph*. Teknik pengujian ini menggunakan grafik sebagai patokannya. Grafik ini menggambarkan relasi antara efek dan penyebab dari *error*.
4. *Equivalence partitioning*
Teknik ini bekerja dengan cara membagi data *input* dari beberapa perangkat lunak menjadi beberapa partisi data.
5. *Fuzzing*
Fuzzing merupakan teknik pencarian *bug* dalam perangkat lunak dengan memasukan data yang tidak sempurna.
6. *Orthogonal array testing*
Selanjutnya adalah *orthogonal array testing*. Teknik ini digunakan jika *input* berukuran kecil, akan tetapi cukup berat jika digunakan dalam skala yang besar.
7. *State transition*
Terakhir adalah *state transition*. Teknik ini berguna untuk melakukan pengujian terhadap mesin dan navigasi dari *UI* dalam bentuk grafik.

3.14 Beta Testing

Beta Testing merupakan metode pembantu dalam memvalidasi perangkat lunak dan memperoleh umpan balik pengguna nyata. Menurutnya, pengujian *beta* dapat mengungkapkan masalah yang belum terdeteksi sebelumnya dan memberikan kesempatan kepada pengguna untuk berpartisipasi dalam proses pengembangan perangkat lunak dengan memberikan masukan dan saran (Pressman, 2014).

4. HASIL DAN PEMBAHASAN

4.1 Variabel Penelitian

Dalam sub bab ini akan dipaparkan hasil dari metode penelitian yang digunakan yaitu metode waterfall seperti, Analisis, Desain, Pengkodean, Pengujian, Penerapan & pemeliharaan.

4.1.1 Analisis

Pada sistem informasi penjadwalan asisten laboratorium komputer, jika penjadwalan dilakukan dengan cara manual maka akan membuat sulit bagi staf asisten lab dalam menyusun jadwal. Sehingga digunakan algoritma genetika untuk memecahkan masalah tersebut. Fungsi Algoritma Genetika sendiri bukan untuk menyelesaikan masalah melainkan untuk mengoptimalkan permasalahan penjadwalan asisten di laboratorium komputeri STMIK Widya Cipta Dharma

1. Analisis Permasalahan

Berdasarkan kegiatan penjadwalan sebelumnya, penjadwalan asisten dilakukan oleh staf asisten dengan

memanfaatkan Google Spreadsheet, dengan jadwal yang disusun secara manual pada saat rapat evaluasi. Meskipun metode ini cukup efektif, namun tidak efisien karena sering kali terjadi masalah seperti memakan banyak waktu dan sering terjadi overload pada saat melakukan generate jadwal. Kendala terkait yang menyebabkan sulitnya dalam penyusunan jadwal asisten yang sesuai.

2. Analisis Data

Pada sistem penjadwalan ini adapun data yang digunakan yaitu :

- 1) Data asisten : Nama asisten, kode asisten, keahlian asisten, ketersediaan asisten
- 2) Data mata kuliah : Jadwal praktikum semester genap tahun akademik 2023/2024, *id*, *period_id*, *room_id*, *skill_id*
- 3) Data ruangan : Id ruang, nama ruangan dan kapasitas
- 4) Data waktu : Terdiri dari data jam dan data hari, data jam yaitu kode jam, nama jam dan rentang jam. sedangkan data hari yaitu kode hari dan nama hari

3. Analisis Fungsional

Berdasarkan hasil dari wawancara, membutuhkan dua tipe akun pengguna. Yaitu staf lab sebagai administrator dan asisten lab sebagai pengguna yang mempunyai fitur yang berbeda sebagai berikut:

1) Fitur Staf Lab

Login : Berisi fitur yang digunakan sebagai *login*/masuk aplikasi sebagai pemilik hak akses *User*.

Sesi Asisten : Berisi pemroses dari algoritma genetika menjadi jadwal asisten lab hasil dari proses penjadwalan asisten lab berdasarkan ketersediaan asisten lab, skill asisten lab dan jadwal praktikum.

Input Jadwal Praktikum : Berisi *input* pengaturan waktu dan ruangan untuk pelaksanaan praktikum selama satu semester.

Ketersediaan Asisten Lab : Berisi ketersediaan waktu dari asisten yang berlaku selama satu semester kedepan.

Hari : Berisi format Hari dan waktu praktikum.

- Ruangan : Berisi daftar ruangan dan kapasitas yang dapat ditampung selama praktikum
- Logout : Untuk keluar dari sesi akun
- 2) Fitur Asisten Lab
 - Login : Berisi fitur yang digunakan sebagai login/masuk aplikasi sebagai pemilik hak akses *user*.
 - Sesi Asisten : Berisi hasil dari proses penjadwalan asisten lab berdasarkan ketersediaan asisten lab, skill asisten lab dan jadwal praktikum
 - Ketersedian Asisten Lab : Berisi ketersediaan waktu dari asisten yang berlaku selama satu semester kedepan
 - Logout : Untuk keluar dari sesi akun

4. Analisis Teknologi

Pada sistem penjadwalan ini adapun perangkat keras dan perangkat lunak yang digunakan yaitu :

- 1) Software
 - a. Bahasa pemrograman ; php 8.3.9
 - b. Aplikasi Editor Program : Visual Studio Code
 - c. Aplikasi Pendukung : *Web Browser*
 - d. *Local Web Server* : Laragon
- 2) Hardware
 - a. Intel(R) Core(TM) i5-9300U CPU @ 2.7 GHz
 - b. RAM 8 GB DDR 4
 - c. 1 TB Storage
 - d. Windows 11 Home Singel Language 64-bit
 - e. *Smartphone* generasi terbaru

4.1.2 Desain

1. Unified Modeling Language(UML)

1) Usecase Diagram User

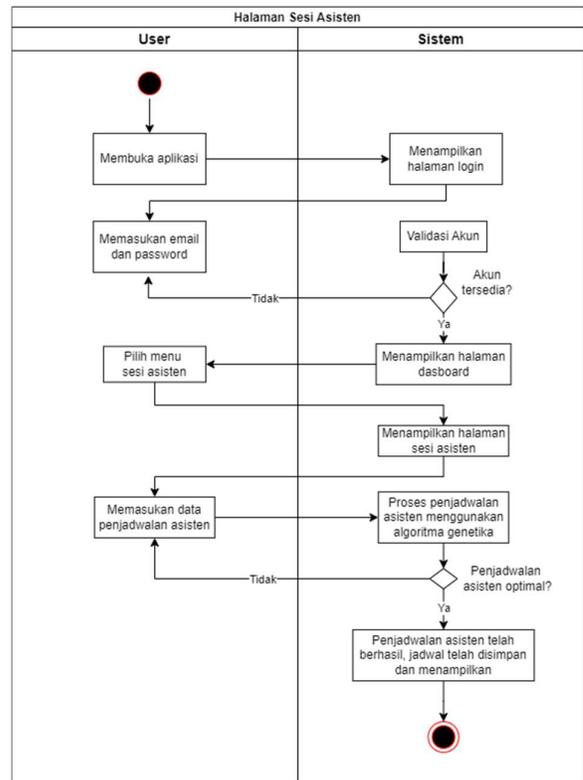


Gambar 1 Usecase Diagram User

Pada Gambar 1, menjelaskan bahwa staf lab dapat menjalankan proses sesi asisten, input jadwal praktikum, data ketersediaan asisten, data hari dan data ruangan. Sementara untuk asisten lab hanya bisa melihat hasil dari proses sesi asisten dan menginputkan ketersediaan asisten per-akun.

2) Activity Diagram Penjadwalan

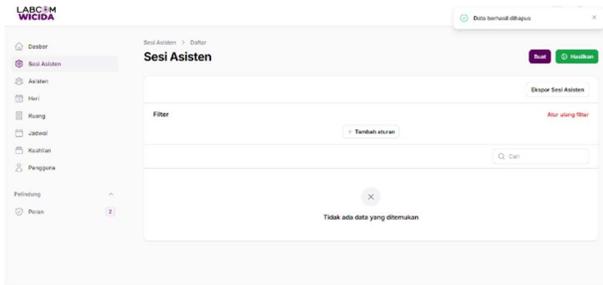
Pada Gambar 4.3, proses dimulai dengan membuka aplikasi, sistem menampilkan halaman login dan meminta Email dan password pengguna. Kemudian, pengguna memasukan email dan password pengguna. Jika data tidak valid, sistem akan meminta memasukan kembali email dan password. Dan jika benar, sistem akan menampilkan halaman dashboard. Setelah pengguna mengisi data penjadwalan asisten lab, sistem melakukan proses algoritma genetika dan melakukan penjadwalan asisten lab, dan kemudian data disimpan. Sistem akan melakukan pemeriksaan terhadap data penjadwalan. Jika data valid, maka data disimpan dan data penjadwalan akan ditampilkan di halaman sesi asisten.



Gambar 2 Activity Diagram Penjadwalan

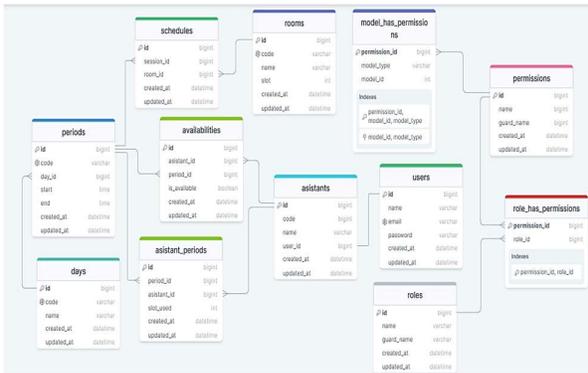
2. Desain Interface

Pada Gambar 3, merupakan tampilan interface aplikasi penjadwalan asisten berfungsi sebagai pemeroses jadwal dan menampilkan hasil dari penjadwalan.



Gambar 3 Rancangan Halaman Penjadwalan

3. Desain Database



Gambar 4 Desain Database

4.1.3 Pengkodean

Pada tahap kali ini dilakukan pengkodean menggunakan bahasa pemrograman php 8.3.9, MySQL sebagai *database* dan menggunakan program editor Visual Studio Code.

1. Implementasi Algoritma Genetika Pada *Syntax*

Pada Gambar 4.6, dimulai dengan merepresentasi data di setiap tabel yang bisa kita sebut kromosom. Setelah itu melakukan inialisasi terhadap kromosom-kromosom. Inialisasi kromosom dilakukan secara acak, tetapi tetap memperhatikan dominan solusi dan kendala yang ada agar terhindar dari kesalahan. Lalu membangun sebuah populasi awal secara acak, setelah itu melakukan evaluasi kromosom. Setelah itu, sistem melakukan evaluasi nilai *fitness* dengan melakukan penyesuaian dengan *course skill* pad

ajadwal praktikum. Lalu dilakukan seleksi berdasarkan populasi menggunakan nilai *fitness*, akan menghasilkan populasi akhir berdasarkan *interval* 0 – 50. Setelah itu, sistem akan membuat populasi baru untuk melakukan proses *crossover* dengan menentukan Gen orang tua 1 dan orang tua 2 berdasarkan populasi akhir, lalu menggunakan *room slot* sebagai *cross_point* menghasilkan keturunan 1 dan keturunan 2. Setelah itu, sistem melakukan penghapusan keturunan yang kembar agar tidak terjadi *error* pada saat proses selanjutnya. Setelah itu melakukan pencocokan keturunan dengan data ketersediaan asisten (*availabilities*) dan data kapasitas ruangan (*room_slot*).

```

if (count($available_assistants) <= $room_slot) {
    return array_slice($available_assistants, 0, $room_slot);
}

// Initial population (random selection)
$population = [];
for ($i = 0; $i < 25; $i++) {
    shuffle($available_assistants);
    $population[] = array_slice($available_assistants, 0, $room_slot);
}

// Fitness function: compatibility with course skills
$fitness = function($individual) use ($course_skill_id) {
    return $this->calculateFitness($individual, $course_skill_id);
};

// Evolution process
for ($generation = 0; $generation < 25; $generation++) {
    // Selection (e.g., tournament selection)
    $used = $population;
    usort($population, function($a, $b) use ($fitness) {
        return $fitness($b) <> $fitness($a);
    });
    $population = array_slice($population, 0, 50); // top 50% selected

    // Crossover
    $new_population = [];
    for ($i = 0; $i < 50; $i += 2) {
        if ($i + 1 >= count($population)) break;
        $parent1 = $population[$i];
        $parent2 = $population[$i + 1];
        $cross_point = rand(1, $room_slot - 1);
        $child1 = array_merge(array_slice($parent1, 0, $cross_point), array_slice($parent2, $cross_point));
        $child2 = array_merge(array_slice($parent2, 0, $cross_point), array_slice($parent1, $cross_point));

        // Remove duplicates in children
        $child1 = array_unique($child1);
        $child2 = array_unique($child2);

        // Ensure the length of children matches the room slot
        while (count($child1) < $room_slot) {
            $child1[] = $available_assistants[array_rand($available_assistants)];
            $child1 = array_unique($child1);
        }
        while (count($child2) < $room_slot) {
            $child2[] = $available_assistants[array_rand($available_assistants)];
            $child2 = array_unique($child2);
        }

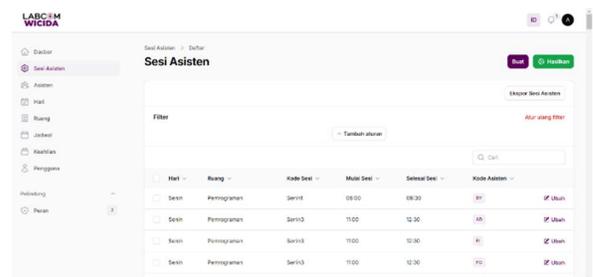
        $new_population[] = $child1;
        $new_population[] = $child2;
    }
}

```

Gambar 5 Implementasi Algoritma Genetika pada *Syntax* Program

2. Tampilan Halaman Penjadwalan Asisten

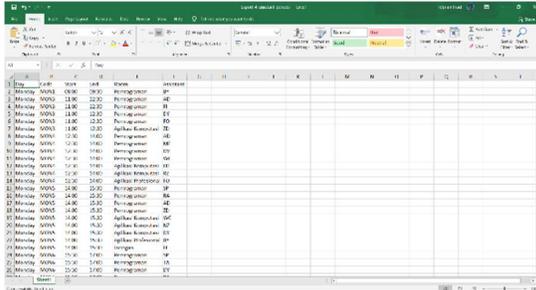
Pada gambar 4.7 merupakan tampilan dari halaman sesi asisten sebagai penjadwalan asisten yang sebagai halaman dari *generate* penjadwalan dan menampilkan hasil dari generate jadwal. Ada beberapa fitur tambahan juga seperti *filter* yang bisa digunakan sebagai menyaring data berdasarkan data yang di dibutuhkan saja. Dan juga ada fitur ekspor hasil dari hasil generate jadwal ke dalam format Excel.



Gambar 4.7 Tampilan Halaman Sesi Asisten

3. Hasil Ekspor Penjadwalan Asisten

Setelah *generate* penjadwalan asisten, sistem akan menampilkan hasil dari generate penjadwalan. Pada gambar 4.8, Hasil dari generate penjadwalan tersebut dapat ekspor ke dalam format Microsoft Excel bila perlu.



Gambar 4.8 Tampilan Ekspor Hasil *Generate* Penjadwalan Asisten

4.1.4 Pengujian

1. Black Box Testing

Tabel 4 Black Box Testing

No	Menu	Fitur dan Fungsi	Hasil yang Diharapkan	Hasil pengujian	
				Berhasil	Gagal
1	Login	Memvalidasi akun pengguna masuk	Pengguna bisa masuk menggunakan akun yang telah dibuat	√	
2	Sesi Asisten	Menampilkan hasil <i>generate</i> penjadwalan asisten dan ekspor hasil penjadwalan	Dapat <i>generate</i> penjadwalan dan menampilkan lainnya. Lalu dapat diekspor ke Excel	√	
3	Jadwal Praktikum	Melakukan proses CRUD data jadwal praktikum dan menampilkan jadwal praktikum	Proses CRUD data jadwal praktikum berhasil dijalankan	√	
4	Ketersediaan asisten	Melakukan CRUD data data asisten, ketersediaan asisten, skill asisten dan menampilkan data asisten	Proses CRUD data asisten, ketersediaan asisten, skill asisten dan menampilkan data jadwal praktikum berhasil dijalankan	√	

Tabel 4 Black Box Testing (Lanjutan)

No	Menu	Fitur dan Fungsi	Hasil yang Diharapkan	Hasil pengujian	
				Berhasil	Gagal
5	Hari	Melakukan CRUD data hari dan waktu sesi dan menampilkan data hari	Proses CRUD data Hari dan menampilkan data hari berhasil dijalankan	√	
6	Ruang	Melakukan CRUD data ruangan dan kapasitas ruangan dan menampilkan data ruangan	Proses CRUD data ruangan dan menampilkan data ruangan berhasil dijalankan	√	

2. Beta Testing

Tabel 5 Beta Testing

No	Pertanyaan	Penilaian				
		SS	S	C	TS	ST
1	Apakah pengguna dapat melakukan <i>Login</i>	6	4	2		
2	Tampilan <i>interface</i> pada aplikasi penjadwalan asisten terlihat rapi	1	5	5	1	
3	Apakah <i>interface</i> pada aplikasi penjadwalan asisten mudah dipahami	1	6	5		
4	Apakah fitur dari aplikasi penjadwalan asisten berfungsi dengan baik	2	5	4	1	
5	Apakah aplikasi penjadwalan asisten berjalan dengan baik secara keseluruhan	1	6	5		

Jumlah responden sebanyak 12 orang yang terdiri dari kepala laboratorium, staf laboratorium dan 10 orang asisten laboratorium, jumlah pertanyaan sebanyak 5 pertanyaan, jumlah nilai tertinggi yaitu 5 dan nilai terendah 1. Rumus untuk menghitung kuesioner menggunakan perhitungan skala likert (Rumus Index % = Total Nilai / Nilai Tertinggi x 100). Dari hasil kuesioner tersebut didapatkan hasil sebagai berikut:

$$\begin{aligned}
 \text{Total Nilai} &= (\text{Total Pemilih} \times \text{Nilai}): \\
 &= (11 \times 5) + (26 \times 4) + (21 \times 3) + (2 \times 2) + (0 \times 1) \\
 &= 55 + 104 + 63 + 4 + 0 = 226
 \end{aligned}$$

$$\begin{aligned} \text{Skor tinggi} &= (\text{Nilai Tertinggi} \times \text{Jumlah Pertanyaan} \\ &\quad \times \text{Jumlah Responden}) \\ &= 5 \times 5 \times 12 = 300 \\ \text{Hasil akhir} &= (\text{Total Nilai} / \text{Skor Tertinggi} \times 100) = \\ 226 / 300 \times 100 &= 75,3 \end{aligned}$$

Dari hasil kuesioner yang dilakukan, maka presentasi nilai yang diperoleh sebesar 75,3% yang termasuk dalam kriteria baik. Sehingga dapat disimpulkan aplikasi penjadwalan asisten dapat digunakan.

5. KESIMPULAN

Berdasarkan uraian-uraian yang telah dijelaskan pada pembahasan sebelumnya maka dapat disimpulkan sebagai berikut:

1. Telah berhasil dibuat sistem informasi penjadwalan asisten laboratorium STMik Widya Cipta Dharma.
2. Algoritma genetika telah berhasil dijalankan dalam melakukan penjadwalan asisten laboratorium menggunakan sistem informasi penjadwalan asisten laboratorium dan diharapkan aplikasi ini mampu mempermudah dalam proses penjadwalan asisten menjadi lebih efektif dan efisien.
3. Hasil pengujian Black Box telah didapat dengan hasil semua fitur dan fungsi berhasil berjalan.
4. Hasil pengujian Beta Testing telah didapat, dengan memperoleh nilai 75,3%

6. SARAN

Berdasarkan kesimpulan yang telah diuraikan, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut dan penerapan hasil penelitian ini:

1. Disarankan untuk penelitian selanjutnya dapat mengembangkan aplikasi ini lebih baik agar menghasilkan penjadwalan asisten yang lebih optimal.
2. Disarankan untuk kedepannya penelitian ini agar nama asisten lab pada hasil generate penjadwalan asisten dapat tersusun rapi.
3. Disarankan untuk kedepannya penelitian ini agar jadwal yang terdapat pada hasil ekspor dengan format Excel yang disertai dengan format jadwal saat ini.

7. DAFTAR PUSTAKA

- Adelheid, Andrea. 2015. Website No. 1 Cara Mudah Bikin Website Dan Promosi Ke CEO. Yogyakarta : Mediakom
- Agustin, H. (2018). Sistem informasi manajemen menurut perspektif islam. *Jurnal Tabarru': Islamic Banking and Finance*, 1(1), 63-70.
- Antonio, H., & Safriadi, N. (2012). Rancang Bangun Sistem Informasi Administrasi Informatika (SI-ADIF). *Jurnal ELKHA*, 4(2).
- Assagaf, A., Ibrahim, A., & Suranto, C. (2018). Membangun Sistem Informasi Penjadwalan Dengan Metode Algoritma Genetika Pada Laboratorium Teknik Informatika Universitas Muhammadiyah Maluku Utara.
- Fatkhurrohman, Z., & Ardian, Y. (2018). Sistem Informasi Penjadwalan Shift Kerja Karyawan Menggunakan Metode Algoritma Genetika. *Semin. Nas. FST*, 1, 475-483.
- Gautama, E. (2016). Algoritma genetika untuk menyelesaikan coin problem: aplikasi pada mesin atm. *JSI: Jurnal Sistem Informasi (E-Journal)*, 8(2).
- Hadi, Diki Alfarabi. 2016. Belajar HTML & CSS Dasar. *Malasngoding.com* Hidayat, R., & Sayfullloh, A. (2021). Aplikasi Penjadwalan Belajar Mengajar Pada Bimbingan Belajar Dengan Menggunakan Metode Waterfall. *BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer*, 2(2), 60-72.
- Hikmah, A. B., Faqih, H., Hudin, J. M., Ramdhani, L. S., & Mulyani, Y. S. (2022). Sistem Informasi Penjadwalan Maintenance Peralatan Menggunakan Model Waterfall. *Swabumi (Suara Wawasan Sukabumi): Ilmu Komputer, Manajemen, dan Sosial*, 10(2), 141-145.
- Indriyani dkk., (2019), Analisa Perancangan Sistem Informasi.
- Ivan, I., Raphael, S., & Agung, H. (2018). Aplikasi penjadwalan mata pelajaran di SMAN 31 menggunakan algoritma genetika berbasis web. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 9(1), 641-656.
- Laksono, A. T., Utami, M. C., & Sugiarti, Y. (2016). Sistem Penjadwalan Kuliah Menggunakan Metode Algoritma Genetika (Studi Kasus: Fakultas Kedokteran Dan Kesehatan Universitas Muhammadiyah Jakarta). *Studia Informatika: Jurnal Sistem Informasi*, 9(2).
- Laudon, K. C., & Laudon, J. P. (2018). *Sistem Informasi Manajemen*. Jakarta: Penerbit Salemba Empat.
- Nabyla, F. (2021). Sistem Informasi Penjadwalan Laboratorium Berbasis Web (Studi Kasus: Politeknik Harapan Bersama Tegal). *Jurnal Sistem Informasi dan Teknologi Peradaban Nursalim*, H. M. (2016). *Sistem Informasi Manajemen*. Jakarta: Erlangga.
- Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach (8th Edition)*. New York, NY: McGraw-Hill.
- Paranduk, L., Indriani, A., Hafid, M., & Suprianto, S. (2018, August). Sistem informasi penjadwalan mata kuliah menggunakan algoritma genetika berbasis web : Seminar Nasional Aplikasi Teknologi Informasi (SNATI).
- Purwanto, F., Djamal, E. C., & Komarudin, A. (2016, August). Optimalisasi Penempatan Halte Trans Metro Bandung Menggunakan Algoritma Genetika. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Pratama, I. P. A. E. (2014). Sistem Informasi dan Implementasi. *Sistem Informasi Dan Implementasi* (p. 508). Bandung: Informatika
- Pinedo, M., 2016, *Scheduling: Theory, Algorithm, and System*, 5th Edition, New York University, New York
- Saputra, Agus. 2018. *Project PHP : Menyelesaikan Website 30 Juta Secara Profesional*. Cirebon : CV. ASFA Solution

- Sutabri, T. (2012). Konsep sistem informasi. Penerbit Andi.
- Utomo, Teguh Wahyu. (2016). Sistem Informasi Manajemen. Yogyakarta: Penerbit Andi.
- Zebua, A. W. (2018). Pengaruh Panjang Bentang Terhadap Ukuran Penampang Optimum Beton Prategang Pada Balok Sederhana Dengan Menggunakan Algoritma Genetika. Jurnal Teknik Sipil, 7(1), 16-25.