

**IMPLEMENTASI PENGGUNAAN ALGORITMA *STEEPEST ASCENT HILL*
CLIMBING DALAM PENYELESAIAN PERMASALAHAN
PENJADWALAN UJIAN
(STUDI KASUS : SMKN 13 MEDAN)**

SKRIPSI

SHAFIRA VAN DEYLI

201401067



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN 2023**

**IMPLEMENTASI PENGGUNAAN ALGORITMA *STEEPEST ASCENT HILL*
CLIMBING DALAM PENYELESAIAN PERMASALAHAN
PENJADWALAN UJIAN
(STUDI KASUS : SMKN 13 MEDAN)**

SKRIPSI

SHAFIRA VAN DEYLI

201401067



**PROGRAM STUDI S-1 ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN 2023**

PERSETUJUAN

Judul : IMPLEMENTASI PENGGUNAAN ALGORITMA
STEEPEST ASCENT HILL CLIMBING DALAM
PENYELESAIAN PERMASALAHAN
PENJADWALAN UJIAN (STUDI KASUS : SMKN
13 MEDAN)

Kategori : SKRIPSI

Nama : SHAFIRA VAN DEYLI

Nomor Induk Mahasiswa : 201401067

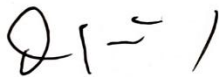
Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 15 Juli 2024

Komisi Pembimbing :

Pembimbing I



Dian Rachmawati, S.Si., M.Kom
NIP. 198307232009122004

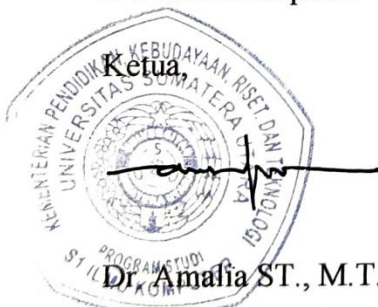
Pembimbing II



Dr. Jos Timanta Tarigan, S.Kom., M.Sc
NIP. 198501262015041001

Diketahui/disetujui oleh Program Studi
S-1 Ilmu Komputer

Ketua

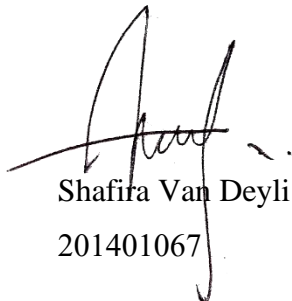


Dr. Amalia ST., M.T. NIP.
197812212014042001

PERNYATAAN**IMPLEMENTASI PENGGUNAAN ALGORITMA *STEEPEST ASCENT HILL CLIMBING* DALAM PENYELESAIAN PERMASALAHAN
PENJADWALAN UJIAN (STUDI KASUS : SMKN 13 MEDAN)****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil penelitian saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah dicantumkan sumbernya.

Medan, 15 Juli 2024



Shafira Van Deyli
201401067

PENGHARGAAN

Memanjatkan pujian kepada Allah SWT atas karunia-Nya, memungkinkan penulis untuk menuntaskan penulisan skripsi dalam proses mendapatkan gelar Sarjana Ilmu Komputer dari Program Studi Ilmu Komputer USU. Salam untuk Nabi Muhammad SAW yang telah membawa umat manusia dari kegelapan ke cahaya, dikatakan bersamaan dengan doa.

Dengan rasa terdalam, penulis menyampaikan terimakasih kepada bunda tercinta, Dra. Yusrifina yang secara konsisten menawarkan cinta, doa, dan dukungan untuk penulis serta tidak lupa juga penulis berterimakasih terhadap papa tercinta, Drs. Zulfan Nazli, M.Ap yang membantu penulis dan memberikan dukungan terhadap penulis dalam setiap proses penulisan skripsi ini.

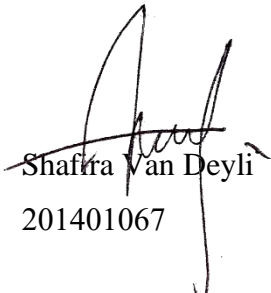
Penulis ingin mengungkapkan beribu terimakasih kepada semua orang yang membantu dan menawarkan dukungan dan arahan disaat penulis melakukan penyusunan skripsi ini.

1. Bapak Prof. Dr. Muryanto Amin S.Sos., M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara serta Dosen Pembimbing Akademik yang memberikan banyak motivasi serta bantuan kepada penulis.
3. Bapak Dr. Mohammad Andri Budiman S.T., M.Comp.Sc., M.E.M. selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara dan Dosen Pembimbing II yang telah memberi masukan yang membangun kepada penulis selama penyusunan skripsi ini.
4. Ibu Dr. Amalia, S.T., M.T. selaku Ketua Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Ibu Dian Rachmawati, S.Si, M.Kom. selaku Dosen Pembimbing I yang memberikan saran yang berharga kepada penulis selama penyusunan skripsi ini.
6. Bapak Dr. Jos Timanta Tarigan, S.Kom., M.Sc. selaku Dosen Pembimbing II yang memberikan masukan kepada penulis selama penyusunan skripsi ini.

7. Ibu Dr. Elviawaty Muisa Zamzami S.T., M.T., MM., IPU. selaku Dosen Pembimbing I yang telah memberi saran dan kritik yang membangun kepada penulis terhadap penyusunan skripsi ini.
8. Terima kasih kepada semua dosen di Program Studi S-1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara atas komitmen yang telah diberikan terhadap pendidikan penulis dan untuk membantu penulis sampai ke titik ini.
9. Seluruh pegawai dan staf Fasilkom-TI USU yang telah memberi bantuan selama masa perkuliahan.
10. Kakak tercinta, Dian Van Deyli S.E dan Fanny Van Deyli, S.KM yang telah berdoa, mendukung, membantu, dan memberikan nasihat yang mendalam kepada penulis.
11. Keluarga besar Bunda, yaitu Om, Tante, sepupu yang telah mendoakan dan mendukung penulis.
12. Sahabat-sahabat yang telah menemani masa perkuliahan “Kom B”, “Ciwi-Ciwi Kom B” dan seluruh teman stambuk 2020 yang memberikan tawa dan semangat kepada penulis
13. Para sahabat yang telah menemani sejak putih abu-abu, diantaranya Dea, Iwa, Geo, Suryani, Fahira, Rizal, Fai, Hazza dan Dio yang telah ada sejak duka maupun suka dan yang selalu memberikan tawa kepada penulis.

Selain itu, kepada semua yang telah membantu dan mendoakan, meskipun tidak mungkin menyebutkan setiap orang, penulis mengungkapkan rasa syukur yang mendalam. Semoga semua bantuan yang telah diberikan diberkahi oleh Tuhan Yang Maha Esa dan diharapkan penelitian ini bisa menambahkan kebaikan serta inspirasi untuk kedepannya.

Medan, 15 Juli 2024


Shafira Van Deyli
201401067

ABSTRAK

Penjadwalan ujian merupakan sebuah permasalahan kompleks yang sering dihadapi oleh lembaga pendidikan, termasuk SMKN 13 Medan. Untuk mengatasi persoalan tersebut, algoritma *Steepest Ascent Hill Climbing* dimanfaatkan sebagai pendekatan untuk menemukan solusi penjadwalan ujian yang optimal. Langkah-langkah implementasi melibatkan penggunaan data guru, ruangan, mata pelajaran, dan jadwal ujian sebagai masukan, dengan tujuan meminimalkan tabrakan antar ujian, memenuhi ketersediaan ruangan dan pengawas, serta mematuhi batasan-batasan lainnya. Algoritma *Steepest Ascent Hill Climbing* bekerja dengan cara mencari solusi yang terbaik dari solusi saat ini, dengan tujuan untuk mencapai solusi optimal. Pencarian solusi dilakukan secara iteratif, dimulai dari solusi awal yang dihasilkan secara acak atau berdasarkan aturan yang ditentukan. Selanjutnya, algoritma melakukan perpindahan ke solusi yang lebih baik dalam setiap iterasi hingga mencapai solusi yang memenuhi kriteria yang ditetapkan. sehingga memberikan hasil jadwal ujian yang meminimalkan tabrakan antar ujian, memastikan ketersediaan ruangan dan pengawas, serta memenuhi batasan-batasan lainnya yang berlaku. Implementasi ini memberikan kontribusi penting dalam meningkatkan efisiensi proses penjadwalan ujian di sekolah, sehingga dapat meningkatkan pengalaman belajar siswa.

Kata Kunci: Heuristik, *Steepest Hill Climbing*, Algoritma *Hill Climbing*, Penjadwalan, Ujian Sekolah, Sekolah Menengah Kejuruan.

ABSTRACT

IMPLEMENTATION OF THE USE OF STEEPEST ASCENT HILL CLIMBING ALGORITHM IN SOLVING OF EXAM SCHEDULING PROBLEM (CASE STUDY: SMKN 13 MEDAN)

Scheduling exams is a complex issue commonly faced by educational institutions, including SMKN 13 Medan. To address this challenge, The Steepest Ascent Hill Climbing algorithm can be used as an approach in finding optimal exam scheduling solutions. Implementation steps involve utilizing data on teachers, rooms, subjects, and exam schedules as inputs, with the aim of minimizing exam conflicts, meeting room and proctor availability, and adhering to other constraints. Steepest Ascent Hill Climbing algorithm operates by seeking best solution from current solution, aiming to achieve an optimal outcome. Solution search is conducted iteratively, starting from an initial solution generated randomly or based on predefined rules. Subsequently, the algorithm moves to a better solution in each iteration until reaching a solution that meets the specified criteria. This results in an exam schedule that minimizes conflicts, ensures room and proctor availability, and complies with relevant constraints. This implementation significantly contributes to enhancing the efficiency of the exam scheduling process in schools, thus improving students' learning experience.

Keywords: Heuristics, Steepest Hill Climbing, Hill Climbing Algorithm, Timetabling, School Exam, Vocational High School.

DAFTAR ISI

PERSETUJUAN	Error! Bookmark not defined.
PERNYATAAN.....	iv
PENGHARGAAN	v
ABSTRAK.....	vii
ABSTRACT.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Penelitian Relevan.....	3
1.7 Metodologi Penelitian	4
1.8 Sistematika Penulisan.....	5
BAB II.....	7
LANDASAN TEORI.....	7
2.1 Penjadwalan	7
2.2 Ujian	7
2.3 <i>Artificial Intelligence</i>	8
2.4 Algoritma	8

2.5	Heuristik	8
2.5.1	Definisi Heuristik	8
2.5.2	Fungsi Heuristik	9
2.5.3	Kriteria Heuristik	9
2.6	Algoritma Hill Climbing	9
2.6.1	Cara Kerja Algoritma <i>Hill Climbing</i>	10
2.7	Algoritma <i>Steepest Ascent Hill Climbing</i>	10
2.7.1	Langkah-Langkah Algoritma <i>Steepest Ascent Hill Climbing</i>	11
2.7.2	Contoh kasus Algoritma <i>Steepest Ascent Hill Climbing</i>	11
BAB III		13
ANALISIS DAN PERANCANGAN		13
3.1	Analisis Sistem	13
3.1.1	Analisis Masalah	13
3.1.2	Analisis Kebutuhan	14
3.1.3	Analisis Proses	14
3.1.4	Arsitektur Sistem	15
3.2	Pemodelan Sistem	16
3.2.1	<i>Use Case Diagram</i>	17
3.2.2	<i>Activity Diagram</i>	17
3.2.3	<i>Sequence Diagram</i>	18
3.3	<i>Flowchart</i> (Diagram Alir)	19
3.3.1	Flowchart Sistem	19
3.3.2	Flowchart Algoritma <i>Steepest Ascent Hill Climbing</i>	20
3.4	Perancangan Antarmuka	21
3.4.1	Rancangan Halaman <i>Login</i>	21
3.4.2	Rancangan Beranda	21
3.4.3	Rancangan Halaman Data & <i>Input Data</i>	22

3.4.4	Rancangan Halaman Penjadwalan Ujian	24
3.5	Perancangan <i>Database</i>	24
BAB IV		27
IMPLEMENTASI DAN PENGUJIAN		27
4.1	Implementasi Sistem	27
4.1.1	Halaman Form Login	27
4.1.2	Halaman Dashboard	28
4.1.3	Halaman List Data	28
4.1.4	Halaman <i>Add Data</i>	31
4.1.5	Halaman Edit Data	33
4.2	Pengujian Sistem	35
4.2.1	Perhitungan Manual	37
4.3	Hasil Pengujian	40
BAB V		44
KESIMPULAN DAN SARAN		44
5.1	Kesimpulan	44
5.2	Saran	44
DAFTAR PUSTAKA		45
LAMPIRAN		47

DAFTAR GAMBAR

Gambar 2.1 Graf Pada Contoh Kasus Pencarian Pencarian Arah Tercepat Tempat Wisata Bandung	11
Gambar 2.2 Hasil Pencarian Pada Contoh Kasus Pencarian Pencarian Arah Tercepat Tempat Wisata Bandung	12
Gambar 3.1 <i>Fishbone Diagram</i>	18
Gambar 3.2 Arsitektur Sistem	16
Gambar 3.3 Detail Pemrosesan Data	16
Gambar 3.4 <i>Use Case Diagram</i>	17
Gambar 3.5 <i>Activity Diagram</i>	18
Gambar 3.6 <i>Sequence Diagram</i>	19
Gambar 3.7 Diagram Alir Sistem	20
Gambar 3.8 Diagram Alir Algoritma <i>Steepest Ascent Hill Climbing</i>	21
Gambar 3.9 <i>Wireframe Form Login</i>	22
Gambar 3.10 <i>Wireframe Dashboard</i>	23
Gambar 3.11 <i>Wireframe Data Guru</i>	24
Gambar 3.12 <i>Wireframe Input Data Guru</i>	24
Gambar 3.13 <i>Wireframe Penjadwalan Ujian</i>	25
Gambar 4.1 Tampilan Form <i>login</i>	29
Gambar 4.2 Tampilan <i>Dashboard</i>	30
Gambar 4.3 Tampilan <i>List Data Jadwal Ujian</i>	30
Gambar 4.4 Tampilan <i>List Data Admin</i>	31
Gambar 4.5 Tampilan <i>List Data Guru</i>	31
Gambar 4.6 Tampilan <i>List Data Keseluruhan Mata Pelajaran</i>	31
Gambar 4.7 Tampilan <i>List Data Mata Pelajaran Sesuai Jurusan</i>	32
Gambar 4.8 Tampilan <i>List Data Jurusan</i>	32
Gambar 4.9 Tampilan <i>List Data Ruang Kelas</i>	32
Gambar 4.10 Tampilan <i>Add Data Admin</i>	33
Gambar 4.11 Tampilan <i>Add Data Guru</i>	33
Gambar 4.12 Tampilan <i>Add Data Keseluruhan Mata Pelajaran</i>	34
Gambar 4.13 Tampilan <i>Add Data Mata Pelajaran Sesuai Jurusan</i>	34

Gambar 4.14 Tampilan <i>Add</i> Data Jurusan	34
Gambar 4.15 Tampilan <i>Add</i> Data Ruang Kelas.....	35
Gambar 4.16 Tampilan <i>Edit</i> Data Admin	35
Gambar 4.17 Tampilan <i>Edit</i> Data Guru.....	36
Gambar 4.18 Tampilan <i>Edit</i> Data Keseluruhan Mata Pelajaran.....	36
Gambar 4.19 Tampilan <i>Edit</i> Data Mata Pelajaran Sesuai Jurusan	36
Gambar 4.20 Tampilan <i>Edit</i> Data Jurusan.....	37
Gambar 4.21 Tampilan <i>Edit</i> Data Ruang Kelas.....	37
Gambar 4.22 Hasil Pengujian Jadwal Ujian Jurusan Akuntansi	41
Gambar 4.23 Hasil Pengujian Jadwal Ujian Jurusan BDP.....	42
Gambar 4.24 Hasil Pengujian Jadwal Ujian Jurusan PHT	42

DAFTAR TABEL

Tabel 2.1 Kondisi Data Yang Digunakan	11
Tabel 3.1 Tabel Admin.....	26
Tabel 3.2 Tabel Guru	26
Tabel 3.3 Tabel Ruang Kelas	26
Tabel 3.4 Tabel Jurusan	27
Tabel 3.5 Tabel Mata Pelajaran.....	27
Tabel 3.6 Tabel Mata Pelajaran Setiap Jurusan	27
Tabel 3.7 Tabel Jadwal Ujian.....	27
Tabel 4.1 Data Mata Pelajaran.....	35
Tabel 4.2 Data Guru.....	36
Tabel 4.3 Data Hari Ujian.....	36
Tabel 4.4 Data Waktu Ujian.....	37
Tabel 4.5 Tabel Jadwal Acak	37
Tabel 4.6 Tabel Jadwal Pembanding	38
Tabel 4.7 Tabel Jadwal Akhir	39

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengelolaan waktu ujian menimbulkan tantangan dalam pengelolaan akademik. Melakukan pengelolaan waktu ujian secara manual menghadapi sejumlah masalah dan kompleksitas yang melibatkan empat parameter penting yaitu batasan waktu, sumber daya yang terbatas, jumlah pertemuan yang terbatas, serta batasan-batasan yang harus dipatuhi (Hermansyah & Muklason, 2020). Kendala-kendala ini meliputi keterbatasan ruang kelas, jumlah siswa, jurusan yang beragam di sekolah menengah kejuruan, dan juga waktu yang terbatas.

Sekolah Menengah Kejuruan Negeri 13 Medan adalah lembaga pendidikan yang menyelenggarakan beragam program pelatihan dan pendidikan vokasional. Sekolah Menengah Kejuruan Negeri 13 Medan memiliki 9 jurusan dengan jumlah keseluruhan siswa berupa 1480 siswa, 80 guru dan 24 ruang kelas. Pada saat ujian sedang berlangsung 1 meja akan diisi oleh 1 siswa dan setiap jurusan harus dijadwalkan ujian dalam waktu bersamaan sehingga menyebabkan beberapa masalah yaitu ketersediaan ruang kelas yang terbatas, ruang kelas atau waktu yang seharusnya lebih baik digunakan tidak dapat diidentifikasi dengan benar, jadwal yang tumpang tindih, terbatasnya ketersediaan waktu guru serta dapat menyebabkan keterlambatan pendistribusian jadwal ujian. Maka dari itu, diperlukan penjadwalan yang cermat untuk memastikan bahwa semua siswa mendapatkan waktu ujian yang sama dan tidak terjadi tumpang tindih antara jadwal ujian sehingga ujian berlangsung dengan lancar. Penerapan sistem penjadwalan ujian dengan dukungan teknologi dapat membantu menghindari tumpang tindih jadwal, mengurangi beban kerja staf sekolah serta menghasilkan jadwal ujian yang efisien.

Namun, Diperlukan pendekatan yang lebih sistematis dan otomatis untuk mengoptimalkan penjadwalan ujian di SMK Negeri 13 Medan. Salah satu pendekatan yang memiliki potensi untuk mengatasi permasalahan ini adalah dengan memanfaatkan metode heuristik (Icasia et al, 2020). Heuristik merupakan metode yang bertujuan untuk meningkatkan efisiensi proses pencarian (Mustafidah & Nurdiansah, 2020). Algoritma *Hill Climbing* merupakan Algoritma heuristik

yang terkategori cocok untuk diterapkan dalam konteks ini. Ada dua variasi Algoritma tersebut, *Simple Hill Climbing* memilih langkah berikutnya dengan membandingkan solusi saat ini dengan satu solusi tetangga, dan solusi tetangga pertama yang lebih baik akan diambil sebagai solusi berikutnya. Di sisi lain, *Steepest Ascent Hill Climbing* membandingkan solusi saat ini dengan semua solusi tetangga dan memilih solusi yang memberikan peningkatan terbesar atau paling dekat dengan solusi optimal (DJiu et al, 2019). Perbedaan kedua algoritma tersebut terlihat dari bagaimana mereka menentukan langkah berikutnya.

Maka dari itu penulis memanfaatkan Algoritma *Steepest Ascent Hill Climbing* dikarenakan algoritma tersebut berfungsi dalam mengatasi masalah efisiensi. Konsepnya berfokus pada "mencapai nilai terbaik dengan mencari peningkatan tertinggi dari solusi yang mungkin agar mendapatkan solusi terbaik". Algoritma *Steepest Ascent Hill Climbing* telah berhasil menghasilkan penjadwalan dengan nilai terbaik sehingga tidak mengalami tabrakan (Afero, 2021). Hal ini menunjukkan bahwa algoritma ini memiliki potensi untuk memberikan solusi terhadap sistem yang dirancang.

Dengan demikian, diharapkan penelitian ini dapat meningkatkan secara signifikan efisiensi dan efektivitas penjadwalan ujian di Sekolah Menengah Kejuruan 13 Medan. Pada akhirnya, penelitian ini akan membantu guru dan siswa mencapai tujuan pendidikan.

1.2 Rumusan Masalah

Munculnya persoalan, didasarkan dari permasalahan latar belakang sebelumnya berupa, penjadwalan ujian secara manual di sekolah menengah kejuruan mengakibatkan jadwal ujian yang kurang efisien dikarenakan setiap jurusan harus dijadwalkan ujian dalam waktu yang bersamaan, maka dapat terjadi ketidakcukupan ruang ujian, ruang ujian atau waktu yang seharusnya lebih baik digunakan tidak dapat diidentifikasi dengan benar, terbatasnya ketersediaan waktu guru serta adanya keterbatasan waktu sehingga dibutuhkan suatu teknologi berupa sistem penjadwalan ujian sekolah dengan menerapkan algoritma *Steepest Ascent Hill Climbing* agar mencapai nilai optimal untuk mendapatkan solusi terbaik.

1.3 Batasan Masalah

Beberapa ruang lingkup persoalan, yaitu:

1. Menerapkan Algoritma *Steepest Ascent Hill Climbing*
2. Program ditujukan dalam penyusunan jadwal ujian semester Sekolah Menengah Kejuruan Negeri 13 Medan.
3. Referensi data yang diambil berupa jumlah ruang ujian, waktu yang tersedia, data guru, durasi ujian, mata pelajaran dan jumlah siswa dari Sekolah Menengah Kejuruan Negeri 13 Medan yang akan mengikuti ujian.
4. Program dirancang berbasis *website* dengan memanfaatkan bahasa pemrograman Javascript serta MySQL sebagai *Database Management System* yang akan digunakan.

1.4 Tujuan Penelitian

Tujuan tersebut dilakukan agar membantu mengoptimalkan proses penjadwalan ujian semester di lingkungan Sekolah Menengah Kejuruan Negeri 13 Medan berbasis *website* dan memanfaatkan algoritma *Steepest Ascent Hill Climbing*.

1.5 Manfaat Penelitian

Manfaat penelitian ini meliputi:

1. Menginformasikan jadwal ujian yang lebih optimal sehingga menghindari terjadinya konflik jadwal.
2. Mengurangi beban kerja staf di Sekolah Menengah Kejuruan Negeri 13 Medan.
3. Mengurangi terjadinya *human error*.
4. Mengetahui efisiensi dari kegunaan Algoritma *Steepest Ascent Hill Climbing* untuk mengatur penjadwalan ujian sekolah menengah kejuruan.
5. Referensi dalam pengembangan sistem selanjutnya dengan topik penjadwalan ujian dan Algoritma *Steepest Ascent Hill Climbing*.

1.6 Penelitian Relevan

Studi sebelumnya yang relevan, meliputi:

1. Berdasarkan penelitian (Ardiyani, 2022) ditemukan jika algoritma *Steepest Ascent Hill Climbing* lebih efisien dalam hal waktu eksekusi dibandingkan dengan algoritma genetika. Penelitian ini menyoroti pentingnya penjadwalan otomatis dalam menyelaraskan jadwal kuliah, ruangan terbatas, dan ketersediaan profesor.

2. Berdasarkan penelitian (Icasia, Tyasnurita, & Purba, 2020) yang berfokus pada penyelesaian *Examination Timetabling Problem* dengan menggunakan kombinasi heuristik dalam kerangka kerja *hiper-heuristic*. Penelitian ini menggunakan dataset Toronto dan mengimplementasikan Algoritma *Hill-Climbing* dan *Tabu Search*. Hasilnya menunjukkan bahwa algoritma ini berhasil mengotomatiskan proses penjadwalan dan mengurangi penalti dari solusi awal.
3. Berdasarkan penelitian (Muklason D. H., 2020) yang menjelaskan tentang evaluasi penggunaan metode *hyper-heuristic* dengan algoritma *random-hill climbing* untuk menyelesaikan *Examination Timetabling Problem*. Penelitian ini membandingkan hasilnya dengan algoritma lain dan menemukan bahwa meskipun pendekatan *hyper-heuristic* mampu menyelesaikan masalah, namun tidak optimal. Penelitian ini menyarankan eksplorasi lebih lanjut terhadap parameter dan strategi untuk hasil yang lebih optimal. Paper ini menyajikan analisis rinci tentang metode penelitian, hasil, dan kesimpulan.
4. Berdasarkan penelitian (Silvilestari, 2021) dalam pemecahan permainan *puzzle* 8. Penelitian tersebut menganalisis posisi pertama dari *puzzle* dan melakukan pergeseran untuk mendekati susunan yang benar. Tujuannya adalah memberikan sistem pengambilan keputusan yang efisien untuk menyelesaikan kasus permainan *puzzle*. Penelitian ini menyajikan metodologi yang digunakan dan hasil dari iterasi algoritma tersebut. Kesimpulan menyatakan bahwa meskipun algoritma ini dapat berhasil diterapkan pada permainan *puzzle*, namun tidak dapat menyelesaikan semua kasus.

1.7 Metodologi Penelitian

1. Studi Pustaka

Penulis dalam penelitian ini mengumpulkan referensi dari berbagai sumber yang dapat dipercaya. Selain itu, melakukan peninjauan literatur terhadap buku, jurnal, e-book, artikel ilmiah, makalah, dan situs internet terkait dengan subjek penelitian ini berupa sistem penjadwalan ujian.

2. Analisis dan Perancangan Sistem

Penulis akan menggunakan diagram alir (*flowchart*), diagram *ishikawa*, diagram *use case*, diagram *activity* dan diagram *sequence* berdasarkan ruang lingkup penelitian.

3. Implementasi Sistem

Penulis akan melakukan proses pembuatan sebuah sistem yang menerapkan Algoritma *Steepest Ascent Hill Climbing* dengan memanfaatkan *Database Management System* MySQL serta bahasa pemrograman *Javascript*. Sistem akan disajikan berbasis *website*.

4. Pengujian Sistem

Uji coba dilaksanakan pada sistem yang dikembangkan dan memastikan sistem dapat berfungsi dengan baik.

5. Dokumentasi

Dalam tahap ini, seluruh proses penelitian akan dicatat kedalam format skripsi.

1.8 Sistematika Penulisan

Struktur penulisan meliputi:

BAB 1 PENDAHULUAN

Mengeksplorasi alasan di balik terbentuknya judul, pembuatan perumusan persoalan, menetapkan ruang lingkup dan fungsi penelitian, menguraikan kegunaan penelitian, meninjau landasan terkait, mendiskusikan metodologi, dan menguraikan susunan penulisan.

BAB 2 LANDASAN TEORI

Memberikan penjelasan terkait ujian, penjadwalan, *artificial intelligence*, heuristik, dan algoritma yang digunakan.

BAB 3 ANALISIS DAN PERANCANGAN

Menganalisis penerapan algoritma serta rancangan diagram demi menunjang pengembangan sistem

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Mengimplementasikan metode yang dikembangkan serta melakukan analisis hasil pengujian menggunakan data masukan.

BAB 5 KESIMPULAN DAN SARAN

Menyimpulkan isi dari keseluruhan bab yang telah ditelaah, serta memberikan rekomendasi dari peneliti sebagai panduan bagi penelitian mendatang.

BAB II

LANDASAN TEORI

2.1 Penjadwalan

Pengaturan penjadwalan mewakili suatu cara mengambil keputusan dengan mengalokasikan sumber daya untuk melakukan kegiatan dalam jangka waktu yang ditentukan. Ini penting dilakukan dalam berbagai industri diantaranya, manufaktur maupun jasa, terutama untuk mencapai tujuan dan sasaran bisnis secara optimal (Pamungkas, 2020). Penjadwalan adalah proses pengaturan atau perencanaan urutan waktu atau jadwal kegiatan, tugas, atau acara dalam suatu rangkaian atau lingkungan tertentu. Penjadwalan melibatkan pengaturan waktu mulai, durasi, dan urutan kegiatan agar dapat dilaksanakan dengan efisien dan efektif. Penjadwalan berlaku dalam berbagai konteks, seperti penjadwalan pelajaran di sekolah, penjadwalan penerbangan di maskapai, atau bahkan penjadwalan tugas sehari-hari dalam kehidupan pribadi.

Tujuan penjadwalan yaitu untuk mengoptimalkan penggunaan sumber daya yang tersedia, menghindari tumpang tindih, meminimalkan konflik, dan memastikan bahwa kegiatan atau tugas dapat dilakukan dalam waktu yang ditetapkan. Dalam bisnis, penjadwalan juga dapat berdampak pada efisiensi operasional, pelayanan pelanggan, dan produktivitas keseluruhan. Oleh karena itu, pengelolaan penjadwalan yang baik menjadi sangat penting dalam berbagai bidang dan industri.

2.2 Ujian

Pada tahun 2019 menurut Permendikbud Nomor 43, Ujian sekolah adalah kegiatan menilai standar kompetensi lulusan hasil belajar siswa yang dilaksanakan oleh institusi pendidikan. Ujian sekolah bertujuan untuk mengukur pengetahuan, keterampilan, pemahaman, atau kemampuan seorang siswa dalam suatu bidang atau mata pelajaran tertentu.

Ujian melibatkan serangkaian pertanyaan yang dirancang untuk menguji pemahaman dan kemampuan peserta ujian. Ujian dapat berupa tes tertulis, tes praktis, atau evaluasi dalam bentuk lainnya, dan biasanya digunakan di berbagai tingkat pendidikan. Tujuan utama ujian adalah untuk mengukur kemampuan

seseorang dan mengidentifikasi tingkat pengetahuan atau keterampilan yang dimilikinya terkait dengan subjek atau bidang tertentu.

2.3 *Artificial Intelligence*

Salah satu bidang ilmu komputer yang paling tua adalah *Artificial intelligence* atau kecerdasan buatan juga dikenal sebagai (AI), memiliki cakupan yang sangat luas, mencakup semua aspek yang terkait dengan meniru fungsi kognitif untuk memecahkan masalah dunia nyata dan mengembangkan sistem yang mampu belajar dan berpikir seperti manusia (Holzinger et al, 2019). Saat ini, perkembangan *Artificial intelligence* berlangsung pesat dan terlihat dalam berbagai aspek kehidupan sehari-hari. Banyak program dan aplikasi yang telah diperkaya dengan fitur *Artificial intelligence*.

Keberadaan *Artificial intelligence* juga memberikan manfaat signifikan di kehidupan sehari-hari, seperti dalam perangkat Asisten Virtual yang dapat membantu dalam menjawab pertanyaan, memberikan informasi dan melaksanakan perintah sesuai dengan masukan yang diberikan oleh pengguna.

2.4 Algoritma

Al-Khwarizmi, seorang ilmuwan dari Persia, adalah orang pertama yang menyarankan kata "algoritma", yang merupakan asal dari kata "*algorithm*". Awalnya, algoritma digunakan untuk menyelesaikan persoalan aritmatika, namun seiring waktu, algoritma telah banyak digunakan untuk menyelesaikan berbagai masalah matematika. Dari penjabaran diatas, Dapat disimpulkan bahwa Algoritma adalah prosedur sistematis yang mengikuti langkah-langkah tertentu untuk menyelesaikan masalah dan membuat Keputusan (Khesya, 2021).

2.5 Heuristik

2.5.1 Definisi Heuristik

Heuristik berasal dari kata "*heuriskein*", memiliki arti "pencarian pengetahuan". Berbagai bidang ilmu menggunakan istilah ini, meskipun definisi heuristik berbeda-beda. Heuristik digunakan dalam biologi dan psikologi sebagai model deskriptif yang menggambarkan bagaimana orang memperoleh informasi dari lingkungan luar dan internal (memori) dan membuat keputusan berdasarkan informasi tersebut. Dalam pendidikan, heuristik terkait dengan cara mengajar siswa untuk belajar atau menemukan sesuatu (Sayono, 2021).

Metode heuristik digunakan untuk melakukan pengujian terhadap kemudahan penggunaan (*usability*) dan membantu menemukan atau mengidentifikasi masalah pada antarmuka pengguna perangkat lunak (Auliaddina et al, 2021). Dengan mempertimbangkan berbagai pendapat, dapat di tarik kesimpulan, heuristik adalah tahap awal yang mencakup mencari, menemukan, dan mengumpulkan sumber sebanyak mungkin untuk digunakan pada langkah berikutnya dengan tujuan membantu menemukan dan mengidentifikasi masalah.

Pencarian Heuristik adalah metode pencarian dalam *Artificial intelligence* yang menggunakan heuristik sebagai panduan langkah-langkahnya dalam melakukan simulasi komputer dari pemikiran (Masadeh, Sharieh, Jamal, Qasem, & Alsaaidah, 2020). Metode pencarian heuristik merupakan suatu metode yang dapat memberikan solusi yang cukup baik dibandingkan dengan metode lain, karena dalam mendapatkan solusi membutuhkan waktu yang cukup lama (Marti & Reinelt, 2022).

2.5.2 Fungsi Heuristik

Heuristik digunakan untuk mencapai solusi yang diinginkan dengan optimal dalam waktu yang singkat. Heuristik sering digunakan dalam berbagai bidang ilmu diantaranya, ilmu komputer, matematika, psikologi dan juga masalah kehidupan sehari-hari.

2.5.3 Kriteria Heuristik

Didalam penelitian yang dilakukan oleh (Marti & Reinelt, 2022) terdapat 3 kriteria dalam algoritma heuristik yang baik, yaitu:

1. Solusi dapat dicapai dengan penggunaan sumber daya komputasi yang masuk akal.
2. Solusi harus mendekati optimal dengan tingkat kepastian yang tinggi.
3. Peluang yang sangat rendah dalam memperoleh solusi yang buruk.

2.6 Algoritma Hill Climbing

Hill Climbing menjelaskan tentang optimasi yang digunakan untuk memecahkan suatu masalah dengan memanfaatkan data *input* melalui algoritma tertentu (Syathirah, Kusuma, & Ningsih, 2021).

Algoritma ini terdiri dari dua bagian. Bagian pertama, *Simple Hill Climbing* memanfaatkan fungsi heuristik (f_x) dalam mengukur jarak dari node x ke tujuan. Pendekatan *Hill Climbing* ini berguna dalam menyelesaikan masalah yang

memiliki sejumlah solusi alternatif agar mendapatkan solusi terbaik di antara mereka. Sementara itu, yang kedua adalah *Steepest Ascent Hill Climbing* yang menyerupai *Simple Hill Climbing*. Adapun perbedaan terletak dalam langkah awal, Langkah awal tidak dimulai dari satu sisi, melainkan dimulai dari nilai terbaik (Syathirah, Kusuma, & Ningsih, 2021).

Menggunakan metode *Simple Hill Climbing* memerlukan waktu yang lama karena setiap langkah harus diproses secara berurutan. Ketika tidak menemukan posisi yang lebih kecil dari state awal, kembali ke posisi tersebut. Sebaliknya, Algoritma *Ascent Hill Climbing* langsung mengevaluasi posisi yang paling tepat, mempercepat pencarian solusi. Proses pencarian *Simple Hill Climbing* menjadi sulit karena sistem yang digunakan harus lebih besar dan sesuai dengan situasi saat ini; jika tidak, maka harus kembali ke posisi awal sehingga memerlukan waktu yang lama karena setiap langkah harus diproses secara berurutan. Sebaliknya, algoritma *Ascent Hill Climbing* lebih mudah menemukan solusi karena prosesnya dapat dilihat langsung, mengurangi waktu proses dan membutuhkan sedikit ruang penyimpanan sehingga mempercepat pencarian Solusi (Putria, 2022).

2.6.1 Cara Kerja Algoritma *Hill Climbing*

Adapun Algoritma *Hill Climbing* pada penelitian (Sari & Sari, 2022) bekerja dengan cara, yaitu:

1. Memahami kondisi awal (*initial state*) dan jika kondisi saat ini sama dengan kondisi tujuan (*goal state*), maka diharuskan untuk mengulang kembali ke langkah sebelumnya agar menemukan solusi dan keluar.
2. Pola dilakukan berulang-ulang sampai mendapatkan keluaran terbaik sampai solusi tidak ditemukan lagi, sebagai langkah dalam menyelesaikan proses.

2.7 Algoritma *Steepest Ascent Hill Climbing*

Algoritma Simple Hill Climbing dimodifikasi menjadi Algoritma *Steepest Ascent*, yang merupakan metode heuristik (Darnila et al., 2019). Puncak dari algoritma ini adalah mencari solusi kenaikan tertinggi, karena *steepest ascent hill climbing* secara harfiah berarti *gain* tertinggi, bertujuan untuk mendapatkan nilai paling optimal dan menghasilkan solusi terbaik (Ardiyani, 2022).

2.7.1 Langkah-Langkah Algoritma *Steepest Ascent Hill Climbing*

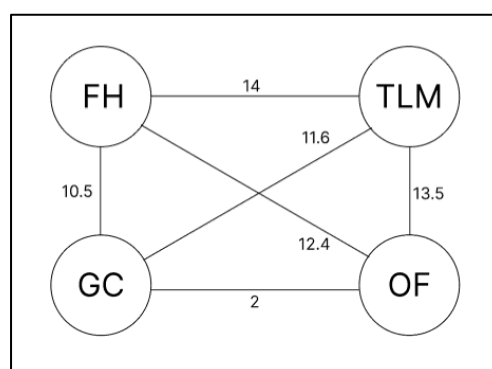
Menurut penelitian oleh (Silvilestari, 2021), algoritma *Steepest Ascent Hill Climbing* melakukan langkah-langkah, meliputi:

1. mengevaluasi keadaan pertama. Jika keadaan pertama dengan keadaan tujuan sama, maka evaluasi berhenti. Namun, jika berbeda, maka keadaan pertama menggantikan keadaan saat ini.
2. Mulailah iterasi keadaan saat ini dengan keadaan awal.
3. Peroleh semua *successor* yang berasal dari keadaan saat ini kemudian digunakan sebagai keadaan selanjutnya.
4. Gunakan fungsi evaluasi untuk menilai semua *successor* dan catat nilainya. Jika salah satu *successor* memiliki nilai yang lebih tinggi daripada keadaan saat ini, maka *successor* dengan nilai terbaik diatur sebagai keadaan saat ini yang baru. Kemudian ulangi proses ini sampai status saat ini sama dengan status tujuan atau tidak berubah.

2.7.2 Contoh kasus Algoritma *Steepest Ascent Hill Climbing*

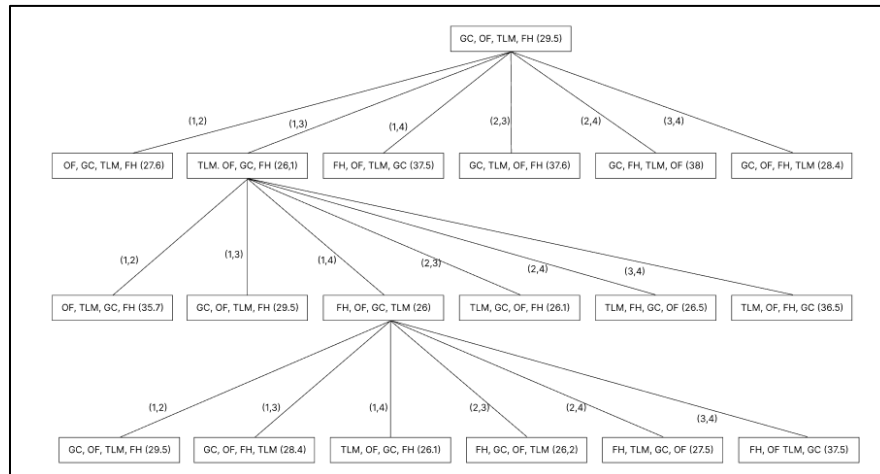
Adapun contoh kasus pencarian arah Tercepat Tempat Wisata Bandung:

Seorang wisatawan yang ingin mengunjungi 4 objek wisata yang ada di bandung diantaranya Grafika Cikole (GC), Orchid Forest (OF), The Lodge Maribaya (TLM), dan Farmhouse (FH). Untuk setiap jarak telah ditentukan dengan menggunakan google maps sehingga dihasilkan jarak yang terlihat di ilustrasi 2.1 berikut:



Gambar 2.1 Graf Pada Contoh Kasus Pencarian Arah Tercepat Tempat Wisata Bandung

Algoritma *Steepest Ascent Hill Climbing* mengolah suatu cara untuk menentukan nilai terbaik dari solusi-solusi yang berdekatan.



Gambar 2.2 Hasil Pencarian Pada Contoh Kasus Pencarian Arah Tercepat Tempat Wisata Bandung

Berdasarkan gambar 2.2, Adapun keadaan pertama dari alur berupa (GC – OF – TLM – FH) sepanjang 29,5 KM.

1. Tingkat pertama, diperoleh (TLM – OF – GC – FH) sepanjang 26,1 KM sebagai solusi terbaik dari solusi tetangga sehingga akan menjadi pilihan untuk level selanjutnya.
2. Pada tingkat kedua, diperoleh (FH – OF – GC – TLM) sepanjang 26 KM sebagai solusi terbaik dari solusi tetangga sehingga akan menjadi pilihan untuk level selanjutnya.
3. Pada tingkat ketiga, tidak terdapat nilai yang lebih baik daripada dengan (FH – OF – GC – TLM) dan dapat diambil menjadi tujuan dari hasil akhir.

BAB III

ANALISIS DAN PERANCANGAN

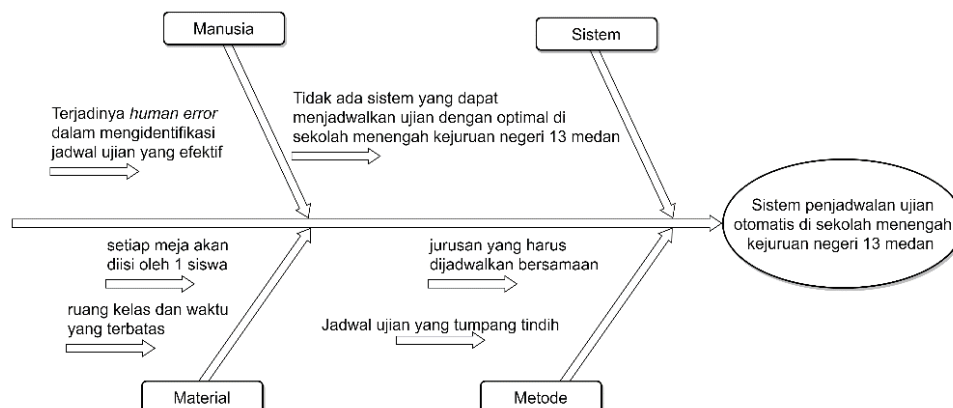
3.1 Analisis Sistem

Memahami penilaian total merupakan bagian dari menganalisis sistem. Proses ini mencakup identifikasi bagian-bagian sistem, bagaimana antar bagian dapat terhubung dan berfungsi serta berinteraksi dengan lingkungannya. Analisis sistem juga melibatkan evaluasi terhadap kekuatan, kelemahan, serta potensi perbaikan atau pengembangan sistem untuk memastikan efisiensi, kinerja optimal, dan pencapaian tujuan yang diinginkan. Adapun tiga tahapan utama untuk menganalisis sistem, diantaranya analisis kebutuhan, analisis masalah, dan analisis proses. Analisis masalah bertujuan mendapatkan pemahaman tentang awal permasalahan muncul dan konsekuensi dari masalah. Sementara itu, analisis kebutuhan berusaha untuk menemukan data yang diperlukan, dan analisis proses menentukan proses yang akan dilakukan selama proses perancangan.

3.1.1 Analisis Masalah

Penelitian ini bermanfaat agar menemukan solusi dalam menetapkan jadwal ujian semester di Sekolah Menengah Kejuruan 13 Medan agar mengurangi beban kerja staf dan menghasilkan jadwal yang optimal.

Untuk mengidentifikasi permasalahan agar lebih jelas, digunakanlah diagram *Ishikawa (fishbone diagram)* yang memperlihatkan penyebab masalah melalui beberapa faktor permasalahan. *Ishikawa* diagram bisa dilihat dalam diagram 3.1.



Gambar 3.1 Diagram *Ishikawa (Fishbone diagram)*

Pada gambar 3.1. terlihat inti persoalan terletak diujung diagram. Ada empat jenis faktor permasalahan yang menjadi faktor pemicu diantaranya metode, manusia, sistem, dan material yang terkait dengan ujung diagram melalui jalur utama.

3.1.2 Analisis Kebutuhan

Analisis kebutuhan terbagi menjadi dua jenis, diantaranya adalah yang tidak berkaitan langsung dengan fungsi tersebut (non-fungsional), sedangkan yang lainnya berkaitan dengan fungsi yang harus dilakukan (fungsional)

3.1.2.1 Kebutuhan Fungsional

Permintaan berdasarkan kegunaan, sebagai berikut:

1. Memberikan solusi dalam penjadwalan ujian semester di sekolah menengah kejuruan negeri 13 medan sesuai kriteria yang dibutuhkan.
2. Menampilkan data siswa
3. Menampilkan data guru
4. Menampilkan data admin
5. Menampilkan data mata Pelajaran
6. Menampilkan data kelas

3.1.2.2 Kebutuhan *Non-Fungsional*

Permintaan tidak berdasarkan kegunaan, sebagai berikut:

1. Antarmuka Pengguna (*User Interface*)
Sistem mampu menampilkan *inteface* yang mudah dipahami oleh *user*.
2. Kinerja
Sistem memiliki kemampuan untuk mengatur jadwal ujian semester di sekolah menengah kejuruan negeri 13 medan.
3. Kemudahan Penggunaan
Sistem didesain dengan sederhana agar *user* dapat memahami sistem dengan baik.

3.1.3 Analisis Proses

Sistem dirancang berbasis *website* dan memanfaatkan algoritma *Steepest Ascent Hill Climbing* sebagai solusi dalam menentukan jadwal ujian di sekolah menengah kejuruan 13 medan. Kriteria data dan batasan yang terdapat dalam proses yang akan dilakukan oleh sistem bisa terlihat dalam tabel 3.1.

Tabel 3.1 Kondisi Data Yang Digunakan

Kondisi Data	Keterangan
Mata Pelajaran	100 mata pelajaran
Ruangan	24 ruang kelas
Jumlah pengajar	80 guru (tahun ajaran 2023-2024)
Jumlah siswa	1480 siswa aktif belajar (tahun ajaran 2023-2024)

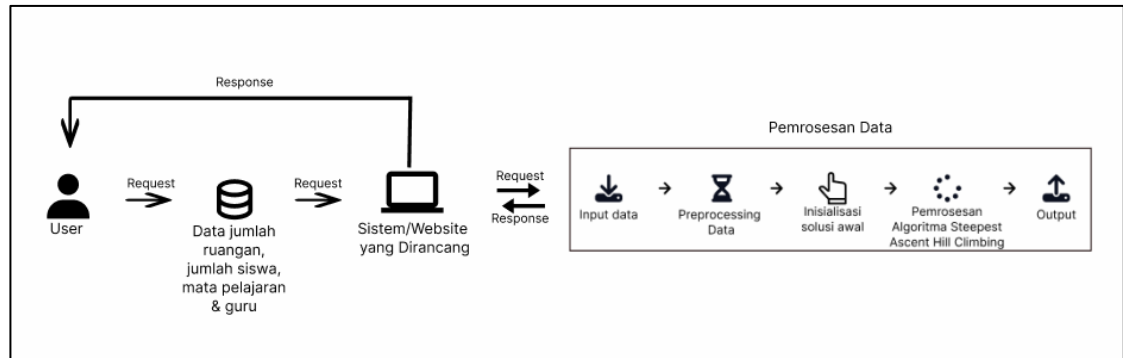
Beberapa faktor yang harus dipertimbangkan saat menyusun jadwal ujian di Sekolah Menengah Negeri 13 Medan, faktor-faktor itu digunakan untuk menentukan kualitas akhir dari jadwal ujian yang dihasilkan. Faktor-faktor tersebut diantaranya :

1. Satu ruangan terdiri 15-17 siswa
2. Satu ruangan akan diisi oleh 2 guru sebagai pengawas ujian
3. Setiap jurusan yang sama harus dijadwalkan ujian secara bersamaan
4. Setiap mata pelajaran yang diujikan terdiri dari 90 menit
5. Ketersediaan ruangan ujian dengan kapasitas yang cukup untuk setiap siswa

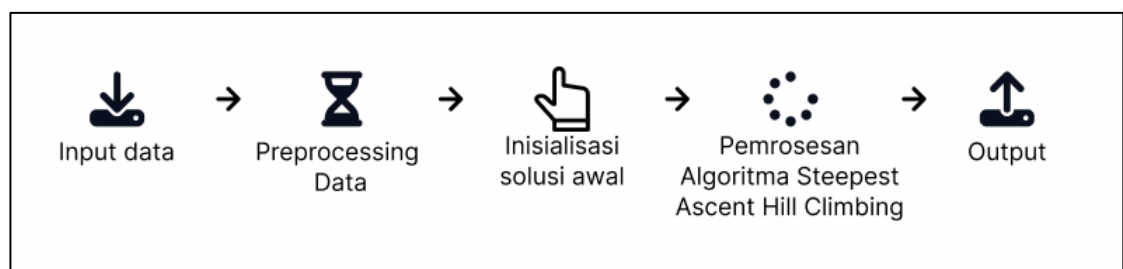
3.1.4 Arsitektur Sistem

Arsitektur sistem adalah struktur atau kerangka kerja yang dirancang dengan berbagai komponen yang berbeda didalam sistem yang saling berinteraksi. Arsitektur sistem merinci komponen utama, hubungan diantara komponen, dan cara bagaimana setiap komponen bekerja sama untuk mencapai tujuan. Arsitektur sistem digunakan dalam berbagai konteks, termasuk dalam pengembangan perangkat lunak, jaringan komputer, sistem informasi, dan banyak lagi.

Adapun arsitektur sistem dari aplikasi yang dibangun bisa dilihat dalam gambar 3.2 serta 3.3



Gambar 3.2 Arsitektur Sistem



Gambar 3.3 Detail Pemrosesan Data

Langkah-langkah dari urutan kerja yang ada dalam gambar 3.1 dapat dijabarkan diantaranya:

1. Data yang dimasukkan oleh *user*/staf sekolah berupa jumlah ruangan, jumlah siswa, mata pelajaran dan data guru .
2. *User* menjalankan sistem untuk menentukan penjadwalan ujian.
3. Sistem akan melakukan proses dari data yang telah ada dan mendapatkan inisialisasi solusi awal
4. Kemudian sistem melakukan proses dari data dan akan menghasilkan *output* lalu akan dikembalikan ke pengguna.

3.2 Pemodelan Sistem

Pemodelan sistem merupakan cara dalam membuat representasi abstrak sistem dengan semua komponennya hubungan antara komponen-komponen tersebut, dan cara sistem tersebut beroperasi. Ini membantu dalam memahami, menganalisis, dan merencanakan sistem sebelum implementasi yang sebenarnya. Pada sistem ini, akan dipresentasikan melalui berbagai diagram seperti diagram *use case*, diagram *activity* dan diagram *sequence*.

3.2.1 Use Case Diagram

Diagram *use case* membantu memahami bagaimana sistem berinteraksi dengan melibatkan beberapa faktor dalam berbagai situasi untuk mencapai tujuan tertentu. Diagram *use case* bisa terlihat dalam gambar 3.4.

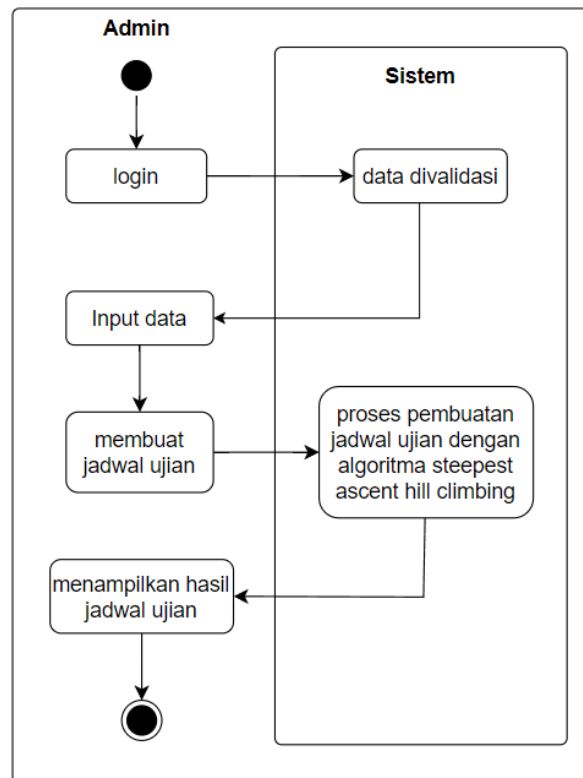


Gambar 3.4 *Use Case Diagram*

Dalam gambar 3.4 dijabarkan sebagai berikut, admin melakukan pengelolaan data diantaranya data mata pelajaran, ruangan, jurusan, siswa maupun guru. Admin juga dapat membuat jadwal ujian secara otomatis dan menampilkan hasil dari jadwal ujian yang telah dibuat oleh sistem.

3.2.2 Activity Diagram

Diagram *Activity* menampilkan serangkaian cara dengan menunjukkan urutan dan hubungan antara aktivitas. Diagram *Activity* bisa terlihat dalam gambar 3.5.

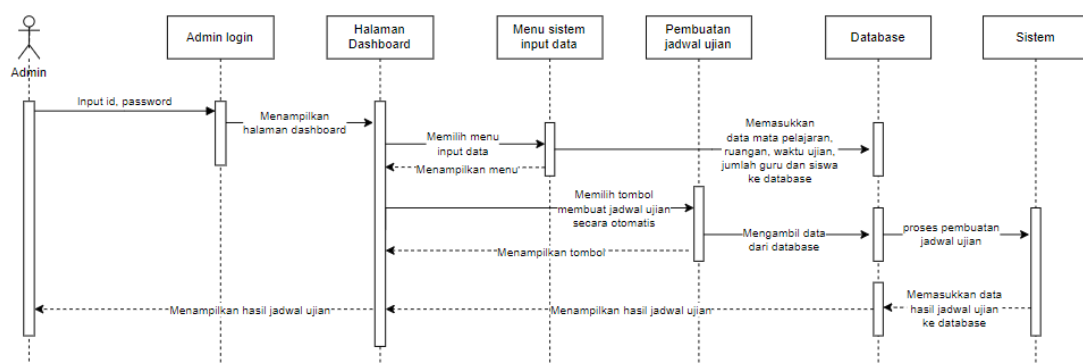


Gambar 3.5 Activity Diagram

Dari ilustrasi gambar 3.5. didapatkan suatu pola interaksi antara admin dan sistem. Aktivitas-aktivitas sistem yang menerima dan merespon kegiatan dari admin tergambar pada kolom sebelah kanan, sementara aktivitas-aktivitas admin dapat terlihat pada kolom sebelah kiri.

3.2.3 Sequence Diagram

Diagram *sequence* bertujuan menampilkan hubungan antara objek disuatu proses dengan menunjukkan urutan waktu dari pesan yang dikirim dan respons yang diterima. Diagram *sequence* dalam sistem tersebut bisa terlihat dalam gambar 3.6.



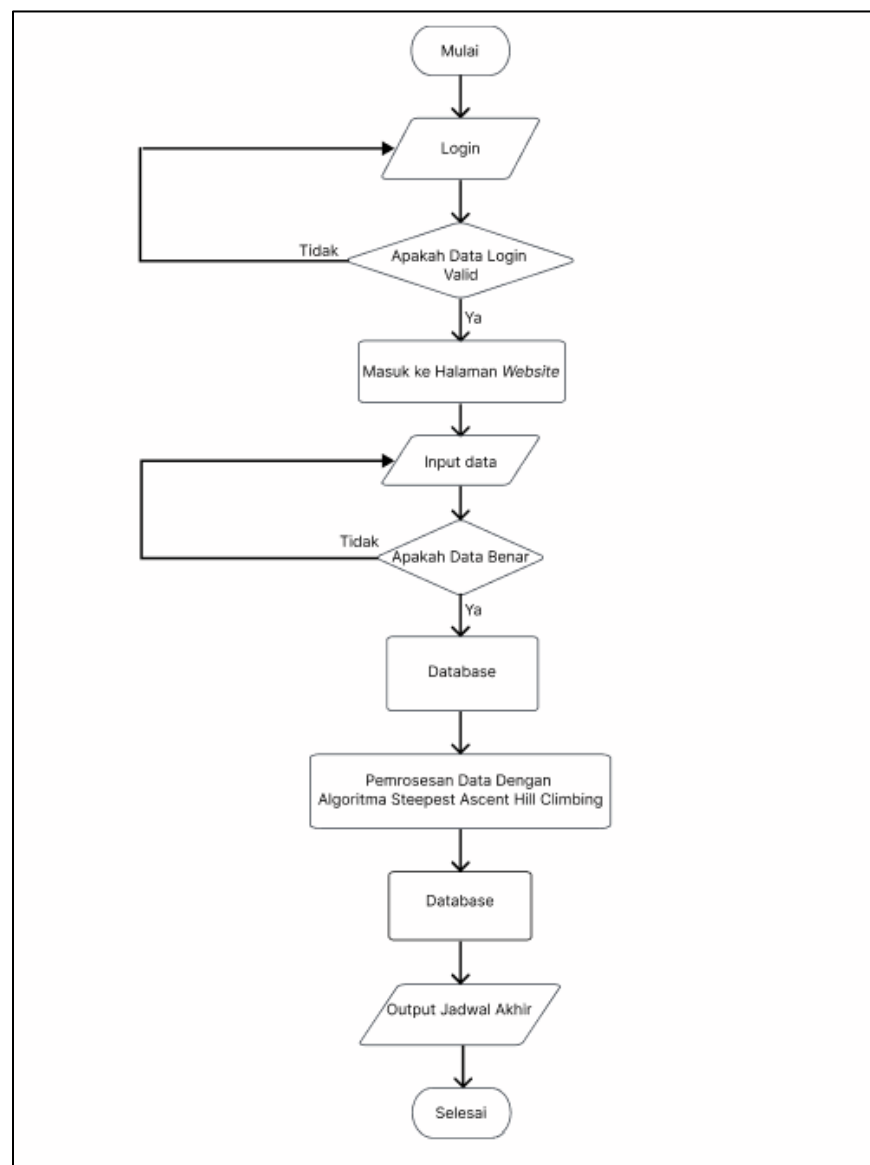
Gambar 3.6 Sequence Diagram

3.3 Flowchart (Diagram Alir)

Diagram Alir merepresentasikan alur atau aliran dari sebuah proses atau sistem. Ini adalah jenis diagram yang dimanfaatkan sebagai rancangan langkah-langkah, keputusan, operasi, maupun tugas yang terlibat disebuah proses secara visual dan hierarkis. *Flowchart* biasanya digunakan dalam berbagai konteks, termasuk rekayasa, pemrograman komputer, bisnis, manufaktur, penelitian ilmiah, dan banyak lagi.

3.3.1 Flowchart Sistem

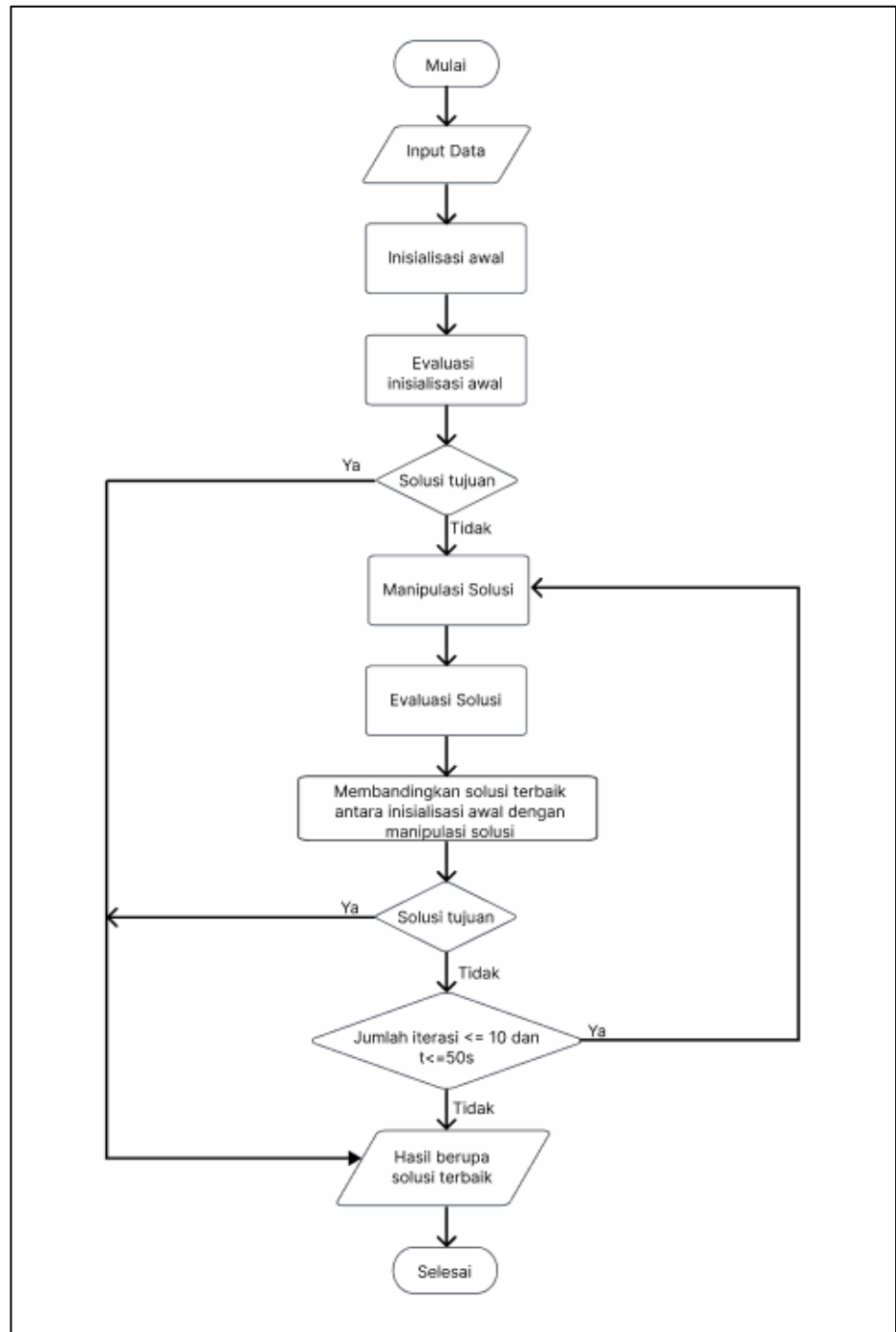
Adapun diagram alir digunakan oleh sistem untuk penelitian yang akan dirancang bisa terlihat di gambar 3.7.



Gambar 3.7 Diagram Alir *Sistem*

3.3.2 Flowchart Algoritma *Steepest Ascent Hill Climbing*

Diagram alir dari Algoritma *Steepest Ascent Hill Climbing* yang digunakan dalam sistem yang akan dirancang terlihat dalam gambar 3.8.



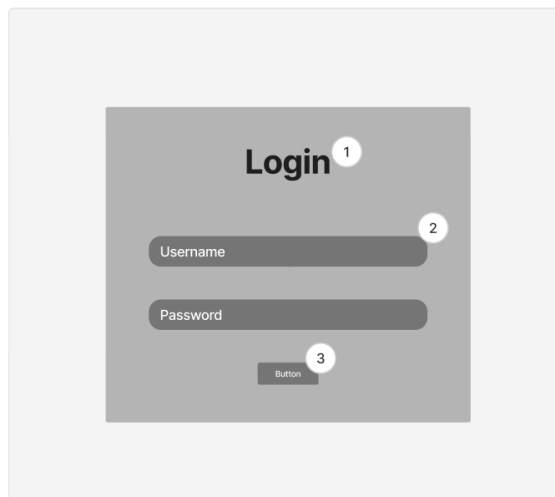
Gambar 3.8 Diagram Alir Algoritma *Steepest Ascent Hill Climbing*

3.4 Perancangan Antarmuka

Rancangan antarmuka adalah suatu cara merancang agar pengguna bisa berinteraksi dengan sistem atau produk, termasuk tata letak elemen-elemen antarmuka, navigasi antarmuka, elemen grafis, dan fitur interaksi lainnya. Ini bertujuan untuk memastikan pengalaman pengguna yang baik saat menggunakan aplikasi, situs web, atau perangkat lunak, serta memfasilitasi interaksi yang efisien dan efektif antara pengguna dan sistem. Teknik *wireframe* akan digunakan dalam perancangan antarmuka sistem ini yang bertujuan untuk penentuan tata letak dan fungsi setiap komponen dalam aplikasi tanpa memerlukan desain yang rumit.

3.4.1 Rancangan Halaman *Login*

Form login dimanfaatkan agar *website* hanya dapat diakses oleh admin sebelum menuju kehalaman utama dari *website*. *Form login* terlihat di gambar 3.9.



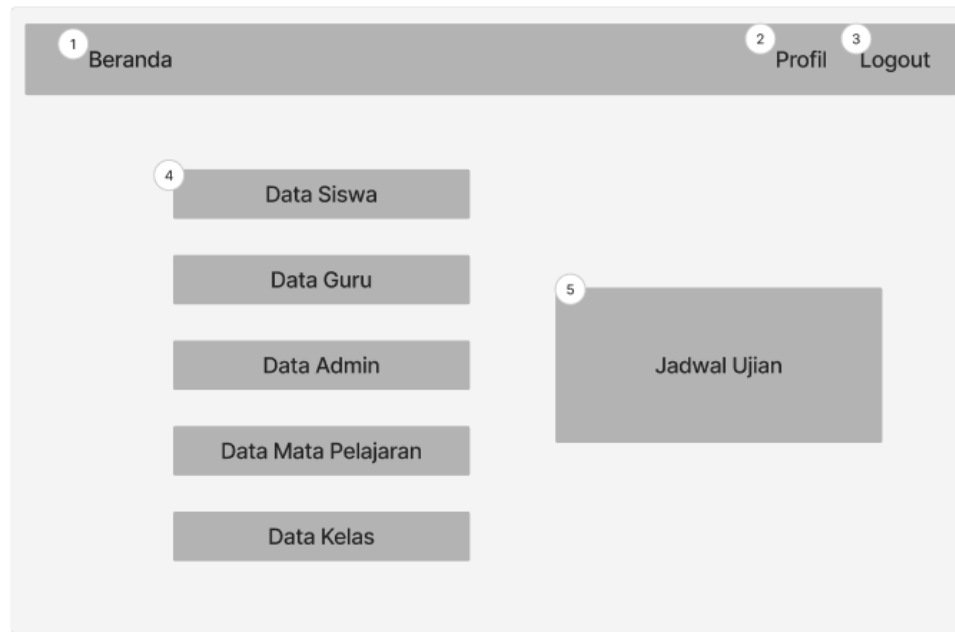
Gambar 3.9 *Wireframe Form login*

Keterangan *wireframe form login*:

1. Label judul *form login*
2. *Input button*
3. *Submit button data login*

3.4.2 Rancangan Beranda

Beranda muncul setelah admin berhasil melakukan *login*. Beranda bisa terlihat dalam gambar 3.10.



Gambar 3.10 *Wireframe Dashboard*

Keterangan *wireframe* dari gambar 3.10:

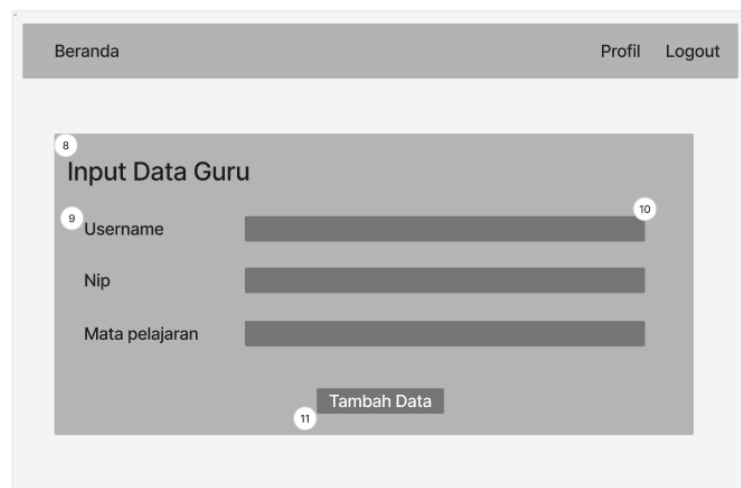
1. Navigasi ke halaman beranda
2. Navigasi ke halaman profil
3. Navigasi untuk melakukan *logout* dari halaman
4. *Toggle button* ke halaman data
5. *Toggle button* ke halaman jadwal ujian

3.4.3 Rancangan Halaman Data & Input Data

Halaman ini dimanfaatkan untuk menampilkan data yang bisa dilihat oleh admin dan juga terdapat halaman untuk memasukkan data. *Wireframe* halaman tersebut dapat terlihat di gambar 3.11 dan 3.12.



Gambar 3.11 Wireframe Data guru



Gambar 3.12 Wireframe Input Data Guru

Keterangan rancangan halaman *input* data sebagai berikut:

1. Navigasi ke halaman beranda
2. Navigasi ke halaman profil
3. Navigasi untuk melakukan *logout* dari halaman
4. Label judul halaman data
5. *Table head* dari data yang ditampilkan
6. *Table data* dari data yang ditampilkan
7. *Toggle button* ke halaman *input* data
8. Label judul halaman input

9. Label keterangan dari *input button* data
10. *Input button*
11. *Submit button*

3.4.4 Rancangan Halaman Penjadwalan Ujian

Halaman penjadwalan ujian digunakan oleh admin agar dapat melihat jadwal ujian yang telah terbentuk secara otomatis oleh sistem.

NO	Jurusan	Hari	Waktu	Kelas	Mata Pelajaran	Ruangan	Pengawas 1	Pengawas 2
1	TKJ	27 June 2024	08:00-09.00	XI TKJ	Mapel 1	3		
2	TKJ	27 June 2024	08:00-09.00	XII TKJ	Mapel 1	6		
3	AKUNTANSI	27 June 2024	08:00-09.00	XI AK	Mapel 2	8		
4	AKUNTANSI	27 June 2024	08:00-09.00	XI AK	Mapel 2	9		
5	AKUNTANSI	27 June 2024	12:30-13.30	XII AK	Mapel 2	2		
6	AKUNTANSI	28 June 2024	12:30-13.30	XII AK	Mapel 2	9		
7	BDP	28 June 2024	12:30-13.30	XI BDP	Mapel 1	5		
8	BDP	28 June 2024	12:30-13.30	X BDP	Mapel 1	2		

Gambar 3.13 Halaman penjadwalan ujian

Adapun keterangan rancangan pada gambar 3.13:

1. Navigasi ke halaman beranda
2. Navigasi ke halaman profil
3. Navigasi untuk melakukan *logout* dari halaman
4. *Table head* jadwal ujian
5. *Table data* jadwal ujian

3.5 Perancangan Database

Perancangan *database* adalah proses merencanakan struktur dan organisasi dari sebuah basis data. Pada perancangan ini, sistem manajemen basis data yang dipilih adalah MySQL, dan dalam konteks penelitian ini, nama basis data yang digunakan adalah *penjadwalanUjian_13*. *Database* tersebut dapat dilihat pada tabel 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8.

Tabel 3.1 Tabel Admin

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Admin	Integer (3)	<i>Primary Key</i>	Id admin
Kode_Admin	Varchar (100)	<i>Unique</i>	Kode nama admin
Nama	Varchar (30)		Nama admin
Email	Varchar (30)		Email admin
Password	Varchar (20)		Kata sandi admin

Tabel 3.2 Tabel Guru

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Guru	Integer (3)	<i>Primary Key</i>	Id guru
Kode_Guru	Varchar (100)	<i>Unique</i>	Kode nama guru
Nama	Varchar (30)		Nama guru
Id_Mapel	Integer (3)	<i>Foreign Key</i>	Relasi tabel mata pelajaran

Tabel 3.3 Tabel Ruang Kelas

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Ruang	Integer (3)	<i>Primary Key</i>	Id ruang kelas
Kode_Ruang	Varchar (100)	<i>Unique</i>	Kode nama ruang kelas
Nama_Ruang	Varchar (10)		Nama ruang kelas
Kapasitas	Integer (4)		Kapasitas ruang kelas
Id_Major	Integer (3)	<i>Foreign Key</i>	Relasi table jurusan
Angkatan	Enum		Angkatan pada kelas tersebut

Tabel 3.4 Tabel Jurusan

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Jurusan	Integer (3)	<i>Primary Key</i>	Id jurusan
Kode_Jurusan	Varchar (100)	<i>Unique</i>	Kode nama jurusan
Nama_Jurusan	Varchar (30)		Nama jurusan

kapasitas	Integer (4)		Jumlah siswa satu jurusan
-----------	-------------	--	---------------------------

Tabel 3.5 Tabel Mata Pelajaran

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Mapel	Integer (3)	<i>Primary Key</i>	Id mata pelajaran
Kode_Mapel	Varchar (100)	<i>Unique</i>	Kode nama pelajaran
Nama_Mapel	Varchar (30)		Nama pelajaran
Jenis	Enum		Jenis pelajaran

Tabel 3.6 Tabel Mata Pelajaran Setiap Jurusan

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Mapel_Jurusan	Integer (3)	<i>Primary Key</i>	Id mata pelajaran
Kode_Mapel_Jurusan	Varchar (100)	<i>Unique</i>	Kode nama mata pelajaran
Id_Mapel	Integer (3)	<i>Foreign Key</i>	Relasi tabel mapel
Id_Jurusan	Integer (3)	<i>Foreign Key</i>	Relasi tabel jurusan
Angkatan	Enum		Tingkatan angkatan

Tabel 3.7 Tabel Jadwal Ujian

Nama Field	Tipe Data	Keterangan	Deskripsi
Id_Jadwal	Integer (5)	<i>Primary Key</i>	Id jadwal
Kode_Jadwal	Varchar (100)	<i>Unique</i>	Kode jadwal
Id_Mapel_Jurusan	Integer (3)	<i>Foreign Key</i>	Relasi tabel mapel
Id_Guru	Integer (3)	<i>Foreign Key</i>	Relasi tabel guru
Id_Kelas	Integer (3)	<i>Foreign Key</i>	Relasi tabel kelas
Id_Jurusan	Integer (3)	<i>Foreign Key</i>	Relasi tabel jurusan
Hari/Tanggal	Date		Hari dan tanggal ujian
Waktu	Time		Jam ujian berlangsung
Ruangan	Integer (3)		Ruangan ujian

BAB IV

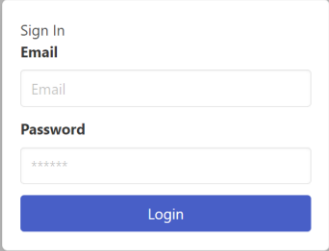
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Proses dalam penelitian ini mengembangkan program penjadwalan ujian secara otomatis. Sistem ini dikembangkan menggunakan *framework ReactJS* sebagai *frontend*, serta *framework Express JS* sebagai *backend* dan terdapat pula *environment node.js* sebagai dasar aplikasi. Sistem ini memiliki antarmuka yang terdiri dari halaman *form login*, *Dashboard*, *List Data*, *Add Data*, *Update Data*, Halaman jadwal ujian yang telah berhasil dibuat.

4.1.1 Halaman Form Login

Saat program dijalankan, halaman awal yang muncul adalah Halaman *Form login*. Seperti yang terlihat dalam ilustrasi 4.1.

The image shows a 'Sign In' form centered on a gray background. The form is a white rectangle with a thin gray border. At the top left of the form, the text 'Sign In' is displayed. Below it, the label 'Email' is followed by a text input field containing the placeholder text 'Email'. Further down, the label 'Password' is followed by a text input field containing six asterisks '*****'. At the bottom of the form is a blue rectangular button with the white text 'Login'.

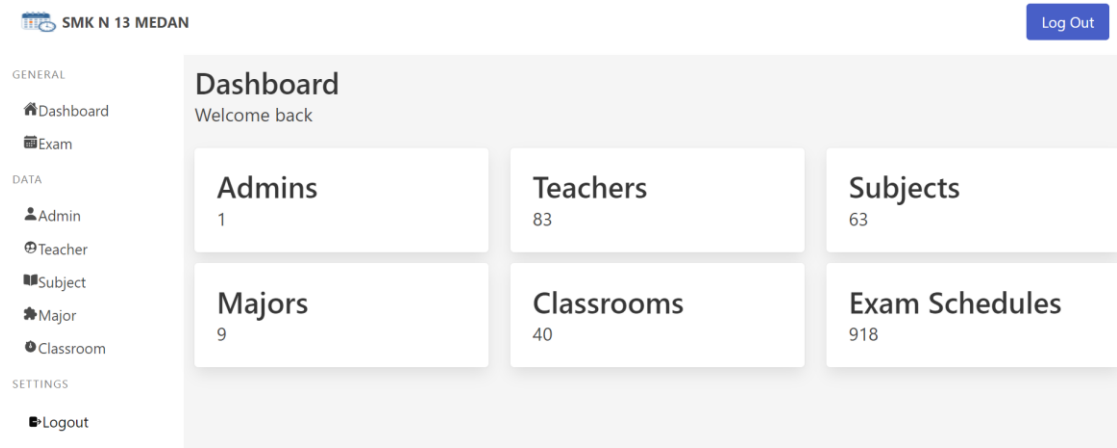
Gambar 4.1 Halaman *Form Login*

Admin diminta untuk menginputkan email dan juga password agar dapat melakukan *login* agar admin dapat mengakses halaman selanjutnya.

4.1.2 Halaman Dashboard

Dashboard akan ditampilkan apabila admin berhasil melakukan *login*.

Halaman tersebut terlihat dalam ilustrasi 4.2



Gambar 4.2 Halaman *Dashboard*

Pada ilustrasi 4.2 admin mampu melihat jumlah keseluruhan data admin, guru, mata pelajaran, jurusan, ruang kelas dan jumlah data keseluruhan jadwal ujian yang telah di *generate*.

4.1.3 Halaman *List Data*

Halaman tersebut berfungsi untuk memperlihatkan keseluruhan data. Adapun data tersebut berupa data jadwal ujian, admin, guru, keseluruhan mata pelajaran, mata pelajaran sesuai jurusan, jurusan, dan ruang kelas. Halaman tersebut bisa diperhatikan dalam ilustrasi 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9.

No	Jurusan	Hari	Waktu	Kelas	Mata Pelajaran	Ruangan	Pengawas 1	Pengawas 2
1	Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13:30	X AKL	SI	1	GRI CAHYA NINGSIH, S.Pd.	MEI MINAR ARITONANG, M.A.
2	Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13:30	X AKL	SI	17	HAMIDAH PASARIBU, S.Pd.	JHON PILIHAN SIMANULLANG, S.Pd.
3	Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13:30	X AKL	SI	21	EDY SYAHPUTRA LANGGENG, S.Pd.I.	JUHRAH, S.Pd.

Gambar 4.3 Tampilan *List Data* Jadwal Ujian

SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Admins

List of Admin

Add new

No	Name	Email	Actions
1	vivi	vivi@gmail.com	Edit Delete

Gambar 4.4 Tampilan *List* Data Admin

SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Teachers

List of Teacher

Add new

No	Name	Subject	Actions
1	JAKFAR, S.H.	PKN	Edit Delete
2	RAMLES AYETMAN BALUT SIHOMBING, S.Kom.	IPAS	Edit Delete
3	HASMAYUNI, S.Pd.	PKN	Edit Delete
4	KHAIRIAH, S.Pd.	B.ING	Edit Delete
5	ISMAYANTI HARAHAH, Dra.	PKK	Edit Delete
6	RIMA HAYATI, S.E.	PKK	Edit Delete
7	RUSMAIDA BANCIN, S.Pd.	PKK	Edit Delete

Gambar 4.5 Tampilan *List* Data Guru

SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Subjects

List of Major

Akuntansi dan Keuangan Lembaga Bisnis Daring dan Pemasaran Desain Grafika Perhotelan dan Pariwisata PTV Teknik Instalasi Tenaga Listrik

Teknik Komputer dan Jaringan Teknik Otomasi Industri Teknik Pengelasan

List of AllSubject

Add new

No	Nama	Jenis	Action
1	SI	wajib	Edit Delete
2	B.JND	wajib	Edit Delete
3	MM	wajib	Edit Delete
4	KONS	wajib	Edit Delete

Gambar 4.6 Tampilan *List* Data Keseluruhan Mata Pelajaran

SMK N 13 MEDAN Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Subjects

Subject list of Akuntansi dan Keuangan Lembaga

[Add new](#)

No	Mata Pelajaran	Jurusan	Angkatan	Actions
1	SI	Akuntansi dan Keuangan Lembaga	X	Edit Delete
2	B.IND	Akuntansi dan Keuangan Lembaga	X	Edit Delete
3	MM	Akuntansi dan Keuangan Lembaga	X	Edit Delete
4	KONS	Akuntansi dan Keuangan Lembaga	X	Edit Delete
5	EP	Akuntansi dan Keuangan Lembaga	X	Edit Delete
6	AD	Akuntansi dan Keuangan Lembaga	X	Edit Delete

Gambar 4.7 Tampilan *List* Data Mata Pelajaran Sesuai Jurusan

SMK N 13 MEDAN Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Majors

List of Major

[Add new](#)

No	Name	Kapasitas	Actions
1	Akuntansi dan Keuangan Lembaga	218	Edit Delete
2	Bisnis Daring dan Pemasaran	108	Edit Delete
3	Desain Grafika	108	Edit Delete
4	Perhotelan dan Pariwisata	181	Edit Delete
5	PTV	108	Edit Delete
6	Teknik Instalasi Tenaga Listrik	217	Edit Delete

Gambar 4.8 Tampilan *List* Data Jurusan

SMK N 13 MEDAN Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Classroom

List of Classroom

[Add new](#)

No	Name	Angkatan	Jurusan	Kapasitas	Actions
1	X AKL 1	X	Akuntansi dan Keuangan Lembaga	37	Edit Delete
2	X AKL 2	X	Akuntansi dan Keuangan Lembaga	37	Edit Delete
3	X BDP	X	Bisnis Daring dan Pemasaran	36	Edit Delete
4	X PHT 1	X	Perhotelan dan Pariwisata	36	Edit Delete
5	X PHT 2	X	Perhotelan dan Pariwisata	35	Edit Delete
6	X TIL 1	X	Teknik Instalasi Tenaga Listrik	35	Edit Delete

Gambar 4.9 Tampilan *List* Data Ruang Kelas

4.1.4 Halaman *Add Data*

Halaman *Add Data* berfungsi untuk menambahkan data yang diperlukan oleh admin. Adapun data yang dapat ditambahkan berupa data admin, guru, keseluruhan mata pelajaran, mata pelajaran sesuai jurusan, jurusan, dan ruang kelas. Dapat diperhatikan dalam gambar 4.10, 4.11, 4.12, 4.13, 4.14, 4.15.

The screenshot shows the 'Admin' form with the following fields and layout:

- GENERAL** sidebar: Dashboard, Exam.
- DATA** sidebar: Admin, Teacher, Subject, Major, Classroom.
- SETTINGS** sidebar: Logout.
- Form Title:** Admin, Add new admin.
- Fields:**
 - Name:** Text input field.
 - Email:** Text input field.
 - Password:** Password input field (masked with dots).
 - Confirm Password:** Password input field (masked with dots).
- Action:** Save button.

Gambar 4.10 Tampilan *Add Data Admin*

The screenshot shows the 'Teacher' form with the following fields and layout:

- GENERAL** sidebar: Dashboard, Exam.
- DATA** sidebar: Admin, Teacher, Subject, Major, Classroom.
- SETTINGS** sidebar: Logout.
- Form Title:** Teacher, Add new teacher.
- Fields:**
 - Name:** Text input field.
 - Subject:** Dropdown menu with 'mata pelajaran' selected.
- Action:** Save button.

Gambar 5.11 Tampilan *Add Data Guru*

The screenshot shows the 'Subject' form with the following fields and layout:

- Header:** SMK N 13 MEDAN, Log Out button.
- GENERAL** sidebar: Dashboard, Exam.
- DATA** sidebar: Admin, Teacher, Subject, Major, Classroom.
- SETTINGS** sidebar: Logout.
- Form Title:** Subject, Add new Subject.
- Fields:**
 - Mata Pelajaran:** Text input field.
 - Jenis:** Dropdown menu with 'Select dropdown' selected.
- Action:** Save button.

Gambar 4.12 Tampilan *Add Data Keseluruhan Mata Pelajaran*

SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Subject

Add new Akuntansi dan Keuangan Lembaga

Mata Pelajaran

Mata Pelajaran ▼

Angkatan

Select dropdown ▼

Save

Gambar 4.13 Tampilan *Add* Data Mata Pelajaran Sesuai Jurusan

SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Major

Add new Major

Name

Name

Kapasitas

Kapasitas

Save

Gambar 4.14 Tampilan *Add* Data Jurusan

SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Classroom

Add new classroom

Name

Name

Angkatan

Select dropdown ▼

Jurusan

mata pelajaran ▼

Kapasitas

Kapasitas

Save

Gambar 4.15 Tampilan *Add* Data Ruang Kelas

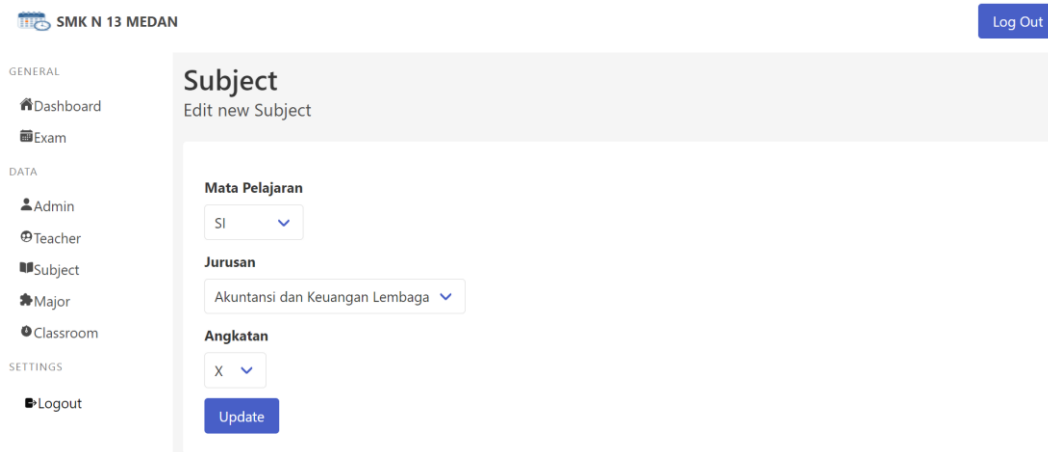
4.1.5 Halaman *Edit Data*

Halaman untuk memperbaharui data berfungsi untuk melakukan perbaikan data oleh admin. Adapun data yang dapat di perbaiki berupa data admin, keseluruhan mata pelajaran, guru, mata pelajaran yang sesuai jurusan, jurusan, dan ruang kelas. Bisa terlihat dalam ilustrasi 4.16, 4.17, 4.18, 4.19, 4.20, 4.21.

Gambar 4.16 Tampilan *Edit Data Admin*

Gambar 4.17 Tampilan *Edit Data Guru*

Gambar 4.18 Tampilan *Edit Data Keseluruhan Mata Pelajaran*



SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Subject

Edit new Subject

Mata Pelajaran

SI

Jurusan

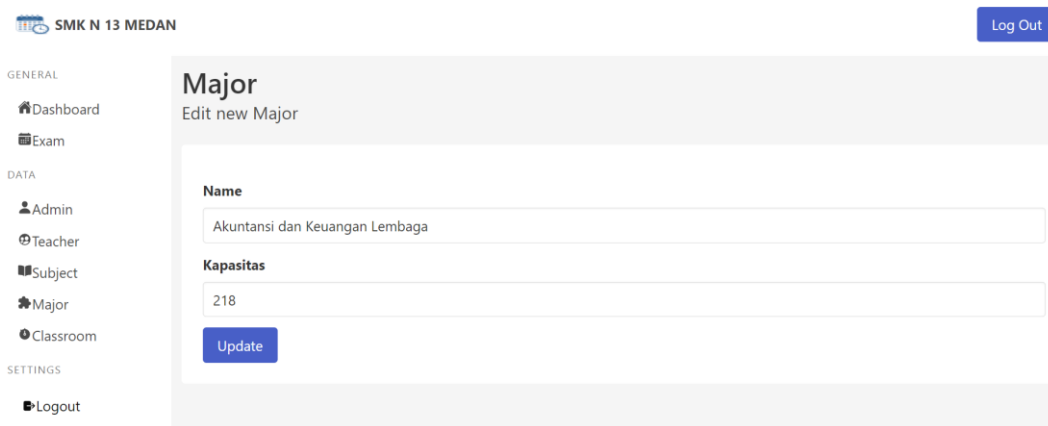
Akuntansi dan Keuangan Lembaga

Angkatan

X

Update

Gambar 4.19 Tampilan *Edit* Data Mata Pelajaran Sesuai Jurusan



SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Major

Edit new Major

Name

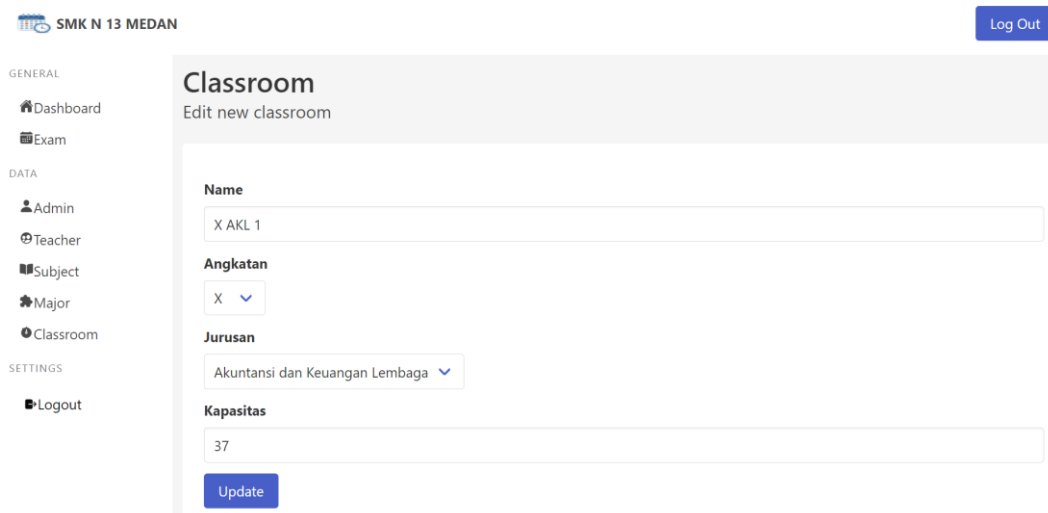
Akuntansi dan Keuangan Lembaga

Kapasitas

218

Update

Gambar 4.20 Tampilan *Edit* Data Jurusan



SMK N 13 MEDAN

Log Out

GENERAL

- Dashboard
- Exam

DATA

- Admin
- Teacher
- Subject
- Major
- Classroom

SETTINGS

- Logout

Classroom

Edit new classroom

Name

X AKL 1

Angkatan

X

Jurusan

Akuntansi dan Keuangan Lembaga

Kapasitas

37

Update

Gambar 4.21 Tampilan *Edit* Data Ruang Kelas

4.2 Pengujian Sistem

Pengujian berfungsi untuk mengetahui keberhasilan serta mencapai tujuan penelitian. Pengujian dengan memanfaatkan algoritma *Steepest Ascent Hill Climbing* bertujuan menemukan jadwal teroptimal dari solusi sebelumnya. Berikut adalah inputan yang diperlukan admin untuk melakukan pengujian:

a. Mata Pelajaran

Berikut adalah contoh data mata pelajaran, angkatan serta jurusan. Dapat dilihat pada tabel 4.1.

Tabel 4.1 Data Mata Pelajaran

Jurusan	Angkatan	Mata Pelajaran
Akuntansi Dan Keuangan Lembaga	X	SI, B.IND, MM, KONS, EP, AD, SS, PD, INF, IPAS, AGM, SB, PKN, B.ING, PJOK
Akuntansi Dan Keuangan Lembaga	XI	MM, PAPJDM, PKK, PJOK, KA, PALIP, AK, B.IND, AGM, B.ING, AP
Akuntansi Dan Keuangan Lembaga	XII	PKK, PAPJDM, AK, KA, PALIP, AP, B.IND, PKN, MM, AGM, B.ING
Bisnis Daring Dan Pemasaran	X	INF, SI, PKN, B.ING, SB, PJOK, IPAS, AGM, C2, B.IND, MM, KONS
Bisnis Daring Dan Pemasaran	XI	MM, BO, PKK, PKN, PJOK, BR, B.IND, AT, B.ING, PP, AGM
Bisnis Daring Dan Pemasaran	XII	PKK, BR, PKN, B.IND, BO, MM, PP, AT, B.ING, AGM

b. Guru

Berikut adalah contoh data guru, mata pelajaran yang diajarkan, beserta waktu ketersediaan di SMK Negeri 13 Medan. Dapat dilihat pada tabel 4.2.

Tabel 4.2 Data Guru

Nama	Hari Ketersediaan
JAKFAR, S.H.	Senin, Selasa, Kamis, Jumat, Sabtu
RAMLES AYETMAN BALUT SIHOMBING, S.Kom.	Senin, Selasa, Rabu, Jumat, Sabtu
HASMAYUNI, S.Pd	Senin, Selasa, Rabu, Jumat, Sabtu
KHAIRIAH, S.Pd.	Senin, Selasa, Kamis, Jumat, Sabtu
ISMAYANTI HARAHAP, Dra.	Senin, Rabu, Kamis, Jumat, Sabtu
RIMA HAYATI, S.E.	Senin, Rabu, Kamis, Jumat
RUSMAIDA BANCIN, S.Pd.	Senin, Selasa, Kamis, Jumat, Sabtu
HERIVANA SEMBIRING, S.S.	Senin, Selasa, Rabu, Jumat, Sabtu
APRILNA ROSMAULI SIANIPAR, S.S.	Senin, Selasa, Rabu, Kamis, Sabtu
MEI MINAR ARITONANG, M.A	Senin, Selasa, Jumat, Sabtu

c. Hari dan Waktu Ujian

Berikut adalah contoh data hari dan waktu ujian di SMK Negeri 13 Medan.
Dapat dilihat pada tabel 4.3 dan 4.4.

Tabel 4.3 Data Hari Ujian

Hari Ujian	
Senin, 17 Juni 2024	Senin, 24 June 2024
Selasa, 18 Juni 2024	Selasa, 25 June 2024
Rabu, 19 Juni 2024	Rabu, 26 June 2024
Kamis, 20 Juni 2024	Kamis, 27 June 2024
Jumat, 21 June 2024	Jumat, 28 June 2024
Sabtu, 22 June 2024	Sabtu, 29 June 2024

Tabel 4.4 Data Waktu Ujian

Waktu Ujian
08:00-09.00
09:30-10.30
11:00-12.00
12.30 – 13.30

4.2.1 Perhitungan Manual

Pada pengujian ini, akan dilakukan perhitungan manual algoritma *Steepest Ascent Hill Climbing*. Untuk melakukan perhitungan manual maka dibutuhkan jadwal acak sebagai permulaan dalam pengujian, adapun jadwal yang disediakan akan diambil sebanyak 8, sebagai berikut:

Tabel 4.5 Jadwal Acak

Nama Jadwal	Jurusan	Hari	Waktu	Kelas	Mapel	Ruangan	Pengawas 1	Pengawas 2
A	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	AD	10	Laksamana	M.Faris
B	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	AD	15	Jhon	Ionessy
C	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	SB	1	Eliza	Wakhida
D	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	SB	12	Pertiwi	Rusmaida
E	Bisnis Daring	Jumat, 21 Juni	08.00 – 09.00	X BDP	B.Ind	19	Sugianto	Jhon

F	Bisnis Daring	Jumat, 21 Juni	08.00 – 09.00	X BDP	B.Ind	15	Ranggi	Muhibba
G	Bisnis Daring	Senin, 17 Juni	12.30 – 13.30	X BDP	C2	12	Roma	Wagito
H	Bisnis Daring	Senin, 17 Juni	12.30 – 13.30	X BDP	C2	2	Laksamana	Sugianto

Pada jadwal acak yang telah dirancang terjadi beberapa tabrakan diantaranya:

1. Waktu ujian pada jadwal A dan B dengan jadwal C dan D.
2. Tabrakan ruangan pada waktu ujian yang sama antar jadwal D dan G.
3. Ditemukan jadwal A dan H terjadi tabrakan pengawas dalam waktu ujian yang sama.

Sehingga nilai heuristik menjadi $= 1+1+1 = 3$

Kemudian jadwal terbaru akan terbentuk dan mengulang kembali proses iterasi.

Tabel 4.6 Jadwal Pembanding

Nama Jadwal	Jurusan	Hari	Waktu	Kelas	Mapel	Ruangan	Pengawas 1	Pengawas 2
A	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	AD	10	Laksamana	M.Faris
B	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	AD	15	Jhon	Ionessy
C	Akuntansi	Selasa, 18 Juni	08.00 – 09.00	X AKL	SB	1	Eliza	Wakhida

D	Akuntansi	Selasa, 18 Juni	08.00 – 09.00	X AKL	SB	12	Pertiwi	Rusmaida
E	Bisnis Daring	Jumat, 21 Juni	08.00 – 09.00	X BDP	B.Ind	19	Sugianto	Jhon
F	Bisnis Daring	Jumat, 21 Juni	08.00 – 09.00	X BDP	B.Ind	15	Ranggi	Muhibba
G	Bisnis Daring	Senin, 17 Juni	12.30 – 13.30	X BDP	C2	12	Roma	Wagito
H	Bisnis Daring	Senin, 17 Juni	12.30 – 13.30	X BDP	C2	2	Laksamana	Sugianto

Pada tabel 4.6 didapatkan nilai heuristik 2. Proses tersebut akan terus terulang hingga mendapatkan nilai heuristik terkecil, sebagai berikut :

Tabel 4.7 Jadwal Akhir

Nama Jadwal	Jurusan	Hari	Waktu	Kelas	Mapel	Ruangan	Pengawas 1	Pengawas 2
A	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	AD	10	Laksamana	M.Faris
B	Akuntansi	Senin, 17 Juni	12.30 – 13.30	X AKL	AD	15	Jhon	Ionessy
C	Akuntansi	Selasa, 18 Juni	08.00 – 09.00	X AKL	SB	1	Eliza	Wakhida

D	Akuntansi	Selasa, 18 Juni	08.00 – 09.00	X AKL	SB	14	Pertiwi	Rusmaida
E	Bisnis Daring	Jumat, 21 Juni	08.00 – 09.00	X BDP	B.Ind	19	Sugianto	Jhon
F	Bisnis Daring	Jumat, 21 Juni	08.00 – 09.00	X BDP	B.Ind	15	Ranggi	Muhibba
G	Bisnis Daring	Senin, 17 Juni	12.30 – 13.30	X BDP	C2	12	Roma	Wagito
H	Bisnis Daring	Senin, 17 Juni	12.30 – 13.30	X BDP	C2	2	Eliza	Sugianto

Pada tabel 4.7 yang merupakan nilai akhir dikarenakan nilai heuristik berupa 0 dan menjadi solusi terbaik yang didapatkan.

4.3 Hasil Pengujian

Pengujian sistem penjadwalan ujian di SMKN 13 Medan menghasilkan temuan signifikan. Adapun contoh dari hasil pengujian oleh sistem dapat dilihat pada gambar 4.22, 4.23, 4.24 sebagai berikut:

Jurusan	Hari	Waktu	Kelas	Mata Pelajaran	Ruangan	Pengawas 1	Pengawas 2
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13.30	XI AKL	AK	5	MAWADDAH ARIYANTI, S.Kom.	HERIVANA SEMBIRING, S.S.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	09:30-10.30	X AKL	B.IND	8	NURUL HUDA, S.Pd.	CITRA EVI MARITO N, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13.30	XI AKL	AK	9	ADY ALEXANDER BARUS, S.Pd.	DEVI TRI GUSTANTI, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13.30	XI AKL	AK	14	JUHRAH, S.Pd.	M. FARIS MUHADRIYAN
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	12:30-13.30	XI AKL	AK	20	TIARMA TANJUNG, Dra.	HARPA ARIHTA TARIGAN, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	09:30-10.30	X AKL	B.IND	11	HASMAYUNI, S.Pd.	ALI PERNANTIAN PANE, S.Pd., M.Psi.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	XI AKL	B.ING	4	AGAHARI AHMAD, S.Pd.	ELIZA, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	XI AKL	B.ING	13	HERIVANA SEMBIRING, S.S.	JHON PILIHAN SIMANULLANG, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	XI AKL	B.ING	18	JAILANI NILA KUSUMA, S.Pd.	MARTHA EVI BEATRIKS SIHOMBING, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	XI AKL	B.ING	21	HARPA ARIHTA TARIGAN, S.Pd.	GRI CAHYA NINGSIH, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	09:30-10.30	X AKL	B.IND	16	REMINI Y. SIMANJUNTAK, S.Pd.	MERISA HANDIRA, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	X AKL	INF	1	ROMA PUTRI ANNA MARIA SIMANGUNSONG, S.Pd.	SUGIANTO, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	X AKL	INF	2	JAILANI NILA KUSUMA, S.Pd.	MIAN AMRAN RITONGA, S.Pd.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	X AKL	INF	9	MEI RIANA RITONGA, S.S.	SUGENG SUTARYO, S.T.
Akuntansi dan Keuangan Lembaga	Senin, 17 Juni 2024	11:00-12.00	X AKL	INF	14	MAWADDAH ARIYANTI, S.Kom.	AL RAMADHONA, S.Pd.I.

Gambar 4.22 Hasil Pengujian Jadwal Ujian Jurusan Akuntansi

Bisnis Daring dan Pemasaran	Senin, 17 Juni 2024	08:00-09.00	XII BDP	PKK	2	OKY JUAN SYAHADA ROHADI, S.Pd.	RAMLES AYETMAN BALUT SIHOMBING, S.Kom.
Bisnis Daring dan Pemasaran	Senin, 17 Juni 2024	08:00-09.00	XII BDP	PKK	24	SYAFRIADI, S.Pd.	DEBORA LUMBANTORUAN, S.Pd.K.
Bisnis Daring dan Pemasaran	Senin, 17 Juni 2024	09:30-10.30	XII BDP	BO	19	TIARMA TANJUNG, Dra.	SITI FATIMAH, S.E.
Bisnis Daring dan Pemasaran	Senin, 17 Juni 2024	09:30-10.30	XII BDP	BO	20	NOPA ESTERLINA BR KARO, S.Pd.	OKY JUAN SYAHADA ROHADI, S.Pd.
Bisnis Daring dan Pemasaran	Senin, 17 Juni 2024	12:30-13.30	XII BDP	PP	2	SRI WAHYUNI SIREGAR, S.Pd.	PERTIWI PUJI LESTARI, S.Pd.
Bisnis Daring dan Pemasaran	Senin, 17 Juni 2024	12:30-13.30	XII BDP	PP	12	AGAHARI AHMAD, S.Pd.	MEI RIANA RITONGA, S.S.

Gambar 4.23 Hasil Pengujian Jadwal Ujian Jurusan BDP

Perhotelan dan Pariwisata	Senin, 17 Juni 2024	08:00-09.00	XI PHT	PJOK	1	JUDYCARLAN SIANIPAR, S.Pd.	MIAN AMRAN RITONGA, S.Pd.
Perhotelan dan Pariwisata	Senin, 17 Juni 2024	08:00-09.00	XI PHT	PJOK	23	RINI WININGSIH, S.Pd.	JAKFAR, S.H.
Perhotelan dan Pariwisata	Senin, 17 Juni 2024	11:00-12.00	XI PHT	IP	2	RINI WININGSIH, S.Pd.	AGAHARI AHMAD, S.Pd.
Perhotelan dan Pariwisata	Senin, 17 Juni 2024	11:00-12.00	XI PHT	IP	22	APRILNA ROSMAULI SIANIPAR, S.S.	FERA LUSIANA NASUTION, S.Pd.

Gambar 4.24 Hasil Pengujian Jadwal Ujian Jurusan PHT

Dalam proses pengujian pada gambar 4.22, 4.23 dan 4.24, ditemukan beberapa kasus tabrakan yang melibatkan antar pengawas di dalam ruangan. Tabrakan antar pengawas diantaranya:

1. Pada jam 11.00 – 12.00, jurusan akuntansi ruangan 2 dengan jurusan akuntansi ruangan 18
2. Pada jam 11.00 – 12.00, jurusan akuntansi ruangan 4 dengan jurusan perhotelan dan pariwisata ruangan 2

Hal ini disebabkan oleh kompleksitas data yang terlibat dalam penjadwalan ujian. Meskipun algoritma mampu mengoptimalkan jadwal dengan baik, tantangan tetap muncul akibat berbagai faktor seperti jumlah guru yang terbatas dan Pengacakan solusi tetangga yang tidak cukup bervariasi untuk memperbaiki tabrakan yang ada sehingga tidak ditemukan solusi jadwal yang lebih baik. Maka dari itu untuk mengatasi hal ini, diperlukan koordinasi yang cermat dalam pengaturan algoritma dan sumber daya yang lebih efisien dan program berjalan lebih optimal.

Secara keseluruhan, menunjukkan bahwa algoritma *Steepest Ascent Hill Climbing* dapat memperoleh pengolahan jadwal ujian dengan meminimalkan konflik, meskipun beberapa tantangan tetap ditemui. Langkah-langkah perbaikan dan penyesuaian terus dilakukan untuk meningkatkan efektivitas algoritma dalam mengatasi berbagai tantangan yang dapat muncul dalam penjadwalan ujian di SMKN 13 Medan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan tahap implementasi dan pengujian penelitian tentang penjadwalan ujian menggunakan Algoritma *Steepest Ascent Hill Climbing*, dapat disimpulkan bahwa:

1. Algoritma *Steepest Ascent Hill Climbing* dapat memberikan solusi yang optimal dengan meminimalkan konflik
2. Terdapat beberapa kasus tabrakan antar pengawas, yang terjadi akibat kompleksitas data.
3. Algoritma telah berusaha memperbaiki solusi, kondisi yang kompleks membutuhkan pembaharuan dan penyesuaian lebih lanjut dalam pengaturan parameter atau pengoptimalan algoritma.

5.2 Saran

Dalam studi mendatang terkait penggunaan algoritma *Steepest Ascent Hill Climbing* untuk penjadwalan ujian, adapun beberapa saran yang dapat dipertimbangkan.

1. Menerapkan metode heuristik tambahan atau teknik optimasi lainnya yang dapat membantu algoritma menghindari tabrakan dan meningkatkan kualitas jadwal ujian yang dihasilkan.
2. Beberapa data yang digunakan dalam penelitian ini menggunakan data statis, diharapkan dalam penelitian yang akan datang menggunakan data dinamis.
3. Penelitian ini menggunakan data yang kompleks. Diharapkan untuk penelitian selanjutnya agar menggunakan data yang tidak terlalu kompleks.
4. Pengembangan sistem pendukung keputusan yang lebih canggih dan *user-friendly* akan membantu para pengguna dalam mengevaluasi, memodifikasi, dan menyetujui jadwal ujian yang dihasilkan oleh algoritma.

DAFTAR PUSTAKA

- Afero, Y. (2021). PENERAPAN KASUS ALGORTIMA ASCENT *HILL CLIMBING* DALAM PERMAINAN PUZZLE 8. *Jurnal Ilmiah Elektronika dan Komputer*, 325-331.
- Ardiyani, L. P. (2022). PERBANDINGAN ALGORITMA GENETIKA DENGAN ALGORITMA *STEEPEST ASCENT HILL CLIMBING* UNTUK OPTIMASI PENJADWALAN KULIAH. *Jurnal Nasional Pendidikan Teknik Informatika*, 63-74.
- auliaddina, s., puteri, a. a., & anshori, l. f. (2021). PERBANDINGAN ANALISA USABILITY DESAIN USER INTERFACE PADA. *Jurnal Ilmiah "Technologia"*, 188-192.
- Darnila, E., Risawandi, & Nursanti. (2019, July). APLIKASI PENCARIAN RUTE TERDEKAT LOKASI KLINIK KESEHATAN MENGGUNAKAN ALGORITMA *STEEPEST ASCENT HILL CLIMBING*. *Jurusan teknik Informatika*, 11, 268-279.
- Djiu, N., Rahman, S., & Musdar, I. A. (2019, Maret). PENERAPAN METODE *HILL CLIMBING* UNTUK MENDIAGNOSA PENYAKIT GANGGUAN SARAF BERBASIS ANDROID. *JTRISTE*, 6, 59-71.
- Hendriyanto. (2022, Maret 08). *Pentingnya Ujian Sekolah dan Asesmen Nasional*. Retrieved from Direktorat Sekolah Dasar:
<https://ditpsd.kemdikbud.go.id/artikel/detail/pentingnya-ujian-sekolah-dan-asesmen-nasional>
- Icasia, G., Tyasnurita, R., & Purba, E. S. (2020). Aplikasi Kombinasi Heuristik dalam Kerangka Hyper-Heuristic untuk Permasalahan Penjadwalan Ujian. *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4(4), 664-671.
- Khesya, N. (2021). MENGENAL FLOWCHART DAN PSEUDOCODE DALAM ALGORITMA DAN PEMROGRAMAN.
- Marti, R., & Reinelt, G. (2022). Exact and Heuristic Methods in Combinatorial Optimization. *Mathematical Sciences*, 27-28.
- Muklason, A., Irianti, R. G., & Marom, A. (2019). Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic

- algorithm. *Procedia Computer Science*, 191, 656-664.
doi:10.1016/j.procs.2019.11.169
- Muklason, D. H. (2020). Evaluation of Hyper-Heuristic Method Using Random-Hill Climbing Algorithm in the Examination Timetabling Problem. *Journal of Physics: Conference Series*, 1569(2), 1-6. doi:10.1088/1742-6596/1569/2/022101
- Muller, H., Holzinger, A., Plass, M., Brcic, L., Stumptner, C., & Zatloukal, K. (2022, May 6). Causability and explainability of *artificial intelligence* in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 67-72.
- Mustafidah, H., & Nurdiansyah, B. (2019, April). Rancang Bangun Aplikasi Penyelesaian Puzzle 8 Angka Menggunakan Metode *Hill Climbing*. *Sainteks*, 16(1), 55-69.
- Nurhasanah, Y. I., Umaroh, S., & Ghoniyyah, N. H. (2021, Juli). Pencarian Rute Optimal Dengan Metode *Steepest Ascent Hill Climbing* Untuk Tempat Wisata Di Bandung Menggunakan Android. *Infotek : Jurnal Informatika dan Teknologi*, 4(2), 113-124.
- Pamungkas, D. (2020). RANCANG BANGUN SISTEM INFORMASI PENJADWALAN UJIAN TUGAS AKHIR (SIJUKIR) BERBASIS WEB (Studi Kasus: Prodi Informatika Universitas Teknologi Yogyakarta) . *Doctoral dissertation, University of Technology Yogyakarta*.
- Rudianto, A., & Muhandhis, I. (2022). Rancang Bangun Sistem Penyusunan Jadwal Pelajaran Menggunakan Algoritma Genetika Berbasis Web (study Kasus MI Mahalul Ulum). *Journal of System Engineering and Technological Innovation (JISTI)*, 1(1), 33-37.
- Sari, D. P. (2022, Juni). Pemanfaatan Metode *Hill Climbing* Mencari Jalur Terpendek Objek Wisata Kabupaten Lima Puluh Kota. *JOISIE Journal Of Information System And Informatics Engineering*, 6(1), 32-38.
- Sayono, J. (2021). LANGKAH-LANGKAH HEURISTIK DALAM METODE SEJARAH DI ERA. *Jurnal Sejarah, Budaya, dan Pengajarannya*, 369-376.
- Silvilestari. (2021). *Steepest Ascent Hill Climbing* Algorithm To Solve Cases In Puzzle. *International Journal of Information System & Technology*, 366-370.

LAMPIRAN

LISTING PROGRAM

Generate Random Schedule

```
const generateRandomSchedule = () => {
  const schedule = {};
  classroom.forEach((className) => {
    schedule[className.major.name] = {};
    classes.forEach((classExam) => {
      schedule[className.major.name][classExam] = {};
    });
  });
  classroom.forEach((className) => {
    classes.forEach((classExam) => {
      subject.forEach((subject) => {
        if (
          className.name.includes(classExam) &&
          subject.major.name === className.major.name &&
          subject.angkatan === className.angkatan
        ) {
          //Random Room
          var randomRoom = [];
          for (var key in resultCountRoom) {
            if (className.name.includes(key)) {
              roomCount = resultCountRoom[key];
            }
          }
          while (randomRoom.length < roomCount * 2) {
            const room = Math.floor(Math.random() * rooms) + 1;
            if (!randomRoom.includes(room)) {
              randomRoom.push(room);
            }
          }
          newRoom.push(randomRoom);
          //Random Teacher
          var randomTeacher = [];
          while (randomTeacher.length < roomCount * 4) {
```

```

        const teachers =
            teacher[Math.floor(Math.random() * teacher.length)];
        if (!randomTeacher.includes(teachers)) {
            randomTeacher.push(teachers);
        }
        const randomExamTime =
            examTimes[Math.floor(Math.random() *
examTimes.length)];
        const randomExamDays =
            examDays[Math.floor(Math.random() * examDays.length)];

        schedule[className.major.name][classExam][subject.subject.name] = {
            date: randomExamDays,
            time: randomExamTime,
        };
        for (var i = 0; i < roomCount * 2; i++) {

schedule[className.major.name][classExam][subject.subject.name][
            randomRoom[i]
        ] = {
            teacher_1: randomTeacher.pop(),
            teacher_2: randomTeacher.pop(),
        };
        }
    });
});
});

Object.keys(schedule).forEach((className) => {
    Object.keys(schedule[className]).forEach((classNa) => {
        Object.keys(schedule[className][classNa]);
        if (
            Object.values(schedule[className][classNa]).every((value)
=> !value)
        ) {
            delete schedule[className][classNa];
        }
    }
});

```

```

    });
  });
  return schedule;
};

Calculate Heuristic

function calculateHeuristic(schedule) {
  let collisionCount = 0; // Menghitung jumlah tabrakan antara
ujian
  let spreadPenalty = 0; // Menghitung penyebaran waktu antara
ujian
  let roomCollisionCount = 0; // Menghitung jumlah tabrakan antar
ruangan
  let teacherCollisionCount = 0; // Menghitung jumlah tabrakan
antar pengawas
  let availabilityCollision = 0; //Menghitung jumlah tabrakan antar
ketersediaan waktu dan hari guru
  const collisions = {};
  // Menghitung jumlah tabrakan antara ujian
  for (const major in schedule) {
    const className = schedule[major];
    for (const classes in className) {
      const subjects = className[classes];
      for (const subject in subjects) {
        const detailExam = subjects[subject];
        const examTime = detailExam.time;
        const examDate = detailExam.date;
        for (const room in detailExam) {
          const teachers = detailExam[room];
          const teacher_1 = teachers.teacher_1;
          const teacher_2 = teachers.teacher_2;
          if (teacher_1 !== undefined || teacher_2 !== undefined) {
            for (const otherSubject in subjects) {
              const otherExam = subjects[otherSubject];
              const otherExamTime = otherExam.time;
              const otherExamDate = otherExam.date;
              if (subject !== otherSubject) {
                if (

```

```

        examDate === otherExamDate &&
        examTime === otherExamTime
    ) {
        collisionCount++;
        if (!collisions[major]) {
            collisions[major] = [];
        }
        if (!collisions[major][classes]) {
            collisions[major][classes] = [];
        }
        if (!collisions[major][classes][subject]) {
            collisions[major][classes][subject] = [];
        }
        collisions[major][classes][subject].push(otherSub
ject);
    }
}
}
}
}
}
}
}
}
}
}
}

// Menghitung penyebaran waktu antara ujian
for (const major in schedule) {
    const className = schedule[major];
    const examTimes = {};
    for (const classes in className) {
        const subjects = className[classes];
        for (const subject in subjects) {
            const detailExam = subjects[subject];
            const examTime = detailExam.time;
            const examDate = detailExam.date;
            if (!examTimes[examTime]) {
                examTimes[examTime] = true;
            }
        }
        spreadPenalty += Object.keys(examTimes).length - 1;
    }
}

const roomTeacherUsage = {};
const availableTeacher = {};

// Menghitung jumlah tabrakan antar ruangan dan pengawas
for (const major in schedule) {
    const className = schedule[major];

```



```

const majorExam = {};
for (const classes in className) {
  const subjects = className[classes];
  for (const subject in subjects) {
    const detailExam = subjects[subject];
    const examTime = detailExam.time;
    const examDate = detailExam.date;
    for (const room in detailExam) {
      const teachers = detailExam[room];
      const teacher_1 = teachers.teacher_1;
      const teacher_2 = teachers.teacher_2;
      if (teacher_1 !== undefined || teacher_2 !== undefined) {
        const teacher = [teacher_1.id, teacher_2.id];
        const key = examDate + "-" + examTime + "-" + room;
        // console.log(majorExam[major])
        for (var i = 0; i < 2; i++) {
          //memeriksa apakah terjadi tabrakan antara pengawas
pada waktu, tanggal, dan major yang sama
          if (!majorExam[major]) {
            majorExam[major] = {};
          }
          if (!majorExam[major][key]) {
            majorExam[major][key] = teacher[i];
          } else if (majorExam[major][key] === teacher[i]) {
            teacherCollisionCount++;
          }
          //memeriksa apakah terjadi tabrakan antara pengawas
pada ruangan, waktu dan tanggal yang sama
          if (!roomTeacherUsage[key]) {
            roomTeacherUsage[key] = [teacher[i]];
          } else {
            if (roomTeacherUsage[key].includes(teacher[i])) {
              // console.log(roomTeacherUsage[key])
              teacherCollisionCount++;
            }
            roomTeacherUsage[key].push(teacher[i]);
          }
        }
      }
    }
  }
}

```

```

    }
    //memeriksa ketersediaan waktu guru dan ujian
    const teacherDay_1 = teacher_1.available.days;
    const teacherTime_1 = teacher_1.available.times;
    const teacherDay_2 = teacher_2.available.days;
    const teacherTime_2 = teacher_2.available.times;

    const availabilityDate = examDate.split(",")[0];
    const availableDate1 =
teacherDay_1.includes(availabilityDate);
    const availableTime1 =
teacherTime_1.includes(examTime);
    const availableDate2 =
teacherDay_2.includes(availabilityDate);
    const availableTime2 =
teacherTime_2.includes(examTime);

    const available1 = availableDate1 && availableTime1;
    const available2 = availableDate2 && availableTime2;
    const available = available1 && available2;

    if (!available) {
        availabilityColission++;
    }
}
}
}
}
}
}
// Menghitung jumlah tabrakan antar ruangan
for (const key in roomTeacherUsage) {
    const teachers = roomTeacherUsage[key];
    if (teachers.length > 1) {
        roomCollisionCount += teachers.length - 1;
    }
}
}

```

```

// Menghitung total nilai heuristik
const heuristicValue =
    collisionCount +
    spreadPenalty +
    roomCollisionCount +
    teacherCollisionCount +
    availabilityCollision;
return heuristicValue;
}

```

Algoritma *Steepest Ascent Hill Climbing*

```

function steepestAscentHillClimbing() {
    let currentSchedule = schedule;

    let currentHeuristic = calculateHeuristic(currentSchedule);
    let bestSchedule = currentSchedule;
    let bestHeuristic = currentHeuristic;

    for (let i = 0; i < maxIterations; i++) {
        var tempSchedule = {};

        let foundBetterNeighbor = false;
        for (var key in resultCountRoom) {
            for (const major in currentSchedule) {
                const className = currentSchedule[major];
                if (!tempSchedule[major]) {
                    tempSchedule[major] = {};
                }
                for (const classes in className) {
                    const subjects = className[classes];
                    if (!tempSchedule[major][classes]) {
                        tempSchedule[major][classes] = {};
                    }
                    for (const subject in subjects) {
                        if (!tempSchedule[major][classes][subject]) {
                            tempSchedule[major][classes][subject] = {};
                        }
                    }
                }
            }
        }
    }
}

```

```

JSON.parse(JSON.stringify(currentSchedule));

tempSchedule =

if (classes.includes(key)) {
    roomCount = resultCountRoom[key];
    //random room
    var newRoom = [];

    while (newRoom.length < roomCount * 2) {
        const room = Math.floor(Math.random() * rooms) + 1;
        if (!newRoom.includes(room)) {
            newRoom.push(room);
        }
    }
    //random Teacher
    var randomTeacher = [];
    while (randomTeacher.length < roomCount * 4) {
        const teachers =
            teacher[Math.floor(Math.random() * teacher.length)];
        if (!randomTeacher.includes(teachers)) {
            randomTeacher.push(teachers);
        }
    }
    const randomExamTime =
        examTimes[Math.floor(Math.random() *
examTimes.length)];
    const randomExamDate =
        examDays[Math.floor(Math.random() * examDays.length)];

    tempSchedule[major][classes][subject] = {
        date: randomExamDate,
        time: randomExamTime,
    };

    for (var j = 0; j < roomCount * 2; j++) {
        tempSchedule[major][classes][subject][newRoom[j]] =
{

```

```

        teacher_1: randomTeacher.pop(),
        teacher_2: randomTeacher.pop(),
    };
}

// Hitung heuristik dari solusi tetangga
const neighborHeuristic =
calculateHeuristic(tempSchedule);

// Bandingkan dengan heuristik saat ini
if (neighborHeuristic < currentHeuristic) {
    currentSchedule = tempSchedule;
    currentHeuristic = neighborHeuristic;
    foundBetterNeighbor = true;
    // Update best solution jika diperlukan
    if (currentHeuristic < bestHeuristic) {
        bestSchedule = currentSchedule;
        bestHeuristic = currentHeuristic;
    }
}
}
}
}
}
}
}
// Berhenti jika tidak ada tetangga yang lebih baik
if (!foundBetterNeighbor) {
    break;
}
}
return bestSchedule;
}

```