

Penerapan Analytic Hierarchy Process (AHP) untuk Menentukan Jurusan Siswa SMA Kelas X Berdasarkan Data Akademik

Muhammad Zayyid Mubarok¹, Kevin Yudha Pratama², Baginda Oloan Lubis^{3*}

^{1,2}Teknik dan Informatika, Sitem Informasi, Universitas Bina Sarana Informatika, Jakarta, Indonesia

³Teknik dan Informatika, Rekayasa Perangkat Lunak, Universitas Bina Sarana Informatika, Jakarta, Indonesia
Jl. Kramat Raya No.98 Jakarta Pusat

E-mail: ¹19210844@bsi.ac.id, ²19210645@bsi.ac.id, ^{3*}baginda.bio@bsi.ac.id

(*: corresponding author)

Abstrak— Pemilihan peminatan mata pelajaran bagi siswa kelas X di tingkat Sekolah Menengah Atas merupakan tahapan krusial yang akan menentukan arah pendidikan dan peluang karier mereka di masa depan. Di SMA Negeri 64 Jakarta Timur, mekanisme pemilihan tersebut masih dilakukan secara manual dan cenderung subjektif, sehingga tidak semua siswa memperoleh jurusan yang sesuai dengan kemampuan akademik serta minat pribadi. Penelitian ini bertujuan merancang dan membangun sistem pendukung keputusan berbasis *web* yang mampu memberikan rekomendasi pemilihan mata pelajaran secara objektif dan sistematis. Sistem ini mengadopsi metode *Analytical Hierarchy Process* (AHP) untuk mengolah data akademik, minat, kesiapan siswa, serta hasil tes teori sebagai kriteria utama dalam menentukan jurusan yang tepat, yaitu IPA atau IPS. Bobot kriteria diperoleh melalui wawancara dengan pakar (guru dan wakil kurikulum), kemudian diterapkan pada algoritma sistem untuk menghasilkan rekomendasi yang lebih akurat. Sistem dirancang dengan fitur *multi-level user* (siswa, guru, admin), rekapitulasi data siswa, uji kuota jurusan, serta antarmuka yang sederhana dan mudah digunakan. Berdasarkan pengujian *blackbox*, sistem terbukti berfungsi sesuai rancangan, memberikan rekomendasi secara tepat, serta membantu sekolah dalam proses bimbingan akademik dengan lebih efisien dan berbasis data. Dengan adanya sistem ini, diharapkan proses penentuan peminatan menjadi lebih adil, terarah, dan mampu mendukung pengembangan potensi siswa secara maksimal.

Kata Kunci—*Analytical Hierarchy Process* (AHP), Minat Siswa, Prestasi Akademik

Abstract— The selection of subject specialization for tenth-grade students in Senior High School is a crucial process that shapes their educational pathways and potential career opportunities in the future. At SMA Negeri 64 Jakarta Timur, this process is still carried out manually and tends to be subjective, resulting in some students not being placed in fields that align with their academic abilities and interests. This study aims to design and develop a *web-based* decision support system that can provide objective and structured recommendations for subject specialization. The system implements the *Analytical Hierarchy Process* (AHP) method to analyze academic performance, interests, student readiness, and theoretical test results as the main criteria in determining the appropriate track, namely Science (IPA) or Social Studies (IPS). The weighting of criteria was obtained through interviews with experts (teachers and curriculum coordinators) and applied within the system's algorithm

to generate accurate recommendations. The system is equipped with *multi-level user access* (students, teachers, administrators), *student data management*, *track quota testing*, and a *user-friendly interface*. Based on *black-box testing*, the system was proven to function as expected, provide accurate recommendations, and support schools in the academic guidance process more efficiently and data-driven. With this system, the subject specialization process is expected to become fairer, more structured, and capable of fostering students' potential optimally.

Keywords—*Analytical Hierarchy Process* (AHP), *Academic Performance*, *Student Interests*

I. PENDAHULUAN

Pendidikan merupakan salah satu faktor fundamental dalam mencetak sumber daya manusia yang berkualitas. Pada jenjang Sekolah Menengah Atas (SMA), siswa dihadapkan pada pilihan peminatan yang akan menentukan jalur pembelajaran mereka, antara lain Matematika dan Ilmu Alam (MIPA), Ilmu-Ilmu Sosial (IPS), serta Bahasa dan Budaya. Keputusan mengenai peminatan ini memiliki dampak jangka panjang terhadap perkembangan siswa, baik dari segi akademik, psikologis, maupun perencanaan karier. Oleh karena itu, pemilihan peminatan yang tepat menjadi salah satu tahapan penting dalam perjalanan pendidikan seorang siswa [1].

Dalam praktiknya, banyak siswa masih mengalami kebingungan dalam menentukan peminatan yang sesuai dengan kemampuan akademik dan minat mereka. Kebingungan ini dapat terjadi karena keterbatasan informasi, tekanan dari lingkungan sekitar, maupun kurangnya pemahaman diri siswa terhadap potensi yang dimilikinya. Jika pemilihan peminatan tidak sesuai, hal tersebut berpotensi menimbulkan dampak negatif seperti penurunan motivasi belajar, kesulitan dalam memahami materi pelajaran, hingga rendahnya capaian akademik. Pada akhirnya, hal ini dapat memengaruhi kesiapan siswa dalam melanjutkan pendidikan ke jenjang perguruan tinggi maupun dunia kerja [2].

Di SMA Negeri 64 Jakarta Timur, proses pemilihan peminatan siswa masih dilakukan secara manual dengan mengandalkan penilaian subjektif dari guru dan wali kelas. Meskipun guru memiliki pemahaman yang cukup terhadap siswa, pendekatan ini tidak selalu menghasilkan rekomendasi

yang akurat karena adanya unsur bias dan keterbatasan dalam mempertimbangkan seluruh aspek penilaian. Kondisi ini menunjukkan perlunya sistem yang mampu memberikan rekomendasi secara objektif, terukur, dan berbasis data.

Salah satu solusi yang dapat diimplementasikan adalah penerapan Sistem Pendukung Keputusan (SPK). SPK merupakan sistem berbasis komputer yang digunakan untuk mendukung pengambilan keputusan dalam situasi kompleks dengan mempertimbangkan berbagai kriteria [3]. Sistem ini tidak menggantikan peran pengambil keputusan, melainkan memberikan rekomendasi yang lebih obyektif, akurat, dan dapat dipertanggungjawabkan. Dalam konteks pendidikan, SPK dapat membantu siswa dan pihak sekolah untuk melakukan proses penentuan peminatan secara lebih terstruktur [4].

Metode yang banyak digunakan dalam SPK adalah AHP. AHP, yang diperkenalkan oleh Thomas L. Saaty, adalah metode pengambilan keputusan multikriteria yang mampu menyelesaikan masalah kompleks dengan cara memecahnya menjadi hierarki, kemudian melakukan perbandingan berpasangan (*pairwise comparison*) untuk mendapatkan prioritas dari setiap kriteria [5]. Keunggulan AHP terletak pada kemampuannya menggabungkan aspek kuantitatif dan kualitatif dalam proses pengambilan keputusan. Dengan demikian, faktor-faktor seperti nilai akademik, minat, motivasi, dan bakat siswa dapat dipertimbangkan secara lebih komprehensif.

Sejumlah penelitian sebelumnya telah menunjukkan efektivitas AHP dalam bidang pendidikan. Penelitian oleh yang menemukan bahwa AHP mampu meningkatkan akurasi dalam pemilihan Pemilihan Guru TK Terbaik [6]. Sementara itu, penelitian lain menunjukkan bahwa penerapan SPK berbasis AHP dapat mengurangi bias subjektif guru dalam memberikan rekomendasi penjurusan [7]. Penelitian lain menegaskan bahwa SPK berbasis web dengan metode AHP dapat mempercepat proses peminatan sekaligus meningkatkan kepuasan siswa terhadap hasil rekomendasi [8]. Temuan-temuan tersebut memperkuat bahwa metode AHP relevan untuk digunakan dalam konteks pemilihan peminatan siswa di SMA.

Selain itu, perkembangan teknologi informasi saat ini semakin mendorong digitalisasi dalam berbagai aspek pendidikan, termasuk dalam hal bimbingan akademik. Sistem berbasis web memberikan keunggulan dalam hal aksesibilitas, kemudahan penggunaan, serta kemampuan menyajikan data secara *real-time*. Dengan adanya SPK berbasis web, sekolah dapat melakukan proses peminatan secara lebih efisien, transparan, dan terukur. Bagi siswa, sistem ini dapat menjadi sarana bimbingan yang membantu mereka lebih memahami potensi dan arah pendidikan yang sesuai dengan kemampuan dan minatnya [9].

Berdasarkan uraian tersebut, dapat disimpulkan bahwa masih terdapat permasalahan mendasar dalam proses pemilihan peminatan di SMA, khususnya di SMA Negeri 64 Jakarta Timur, yang selama ini dilakukan secara manual dan subjektif. Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan pembangunan Sistem Pendukung Keputusan berbasis web dengan metode AHP sebagai solusi alternatif. Sistem ini diharapkan mampu memberikan rekomendasi

peminatan siswa secara objektif, akurat, serta dapat mendukung proses bimbingan akademik yang lebih efektif dan efisien.

II. METODE PENELITIAN

A. Sistem Pendukung Keputusan (SPK)

SPK merupakan suatu sistem yang dapat membantu dalam memberikan pemecahan masalah, baik dalam bentuk komunikasi maupun rekomendasi keputusan yang bersifat terstruktur maupun tidak terstruktur [10]. SPK atau *Decision Support System* (DSS) adalah sistem berbasis komputer yang terdiri dari tiga komponen interaktif. Pertama, sistem bahasa yang berfungsi sebagai mekanisme komunikasi antara pengguna dengan komponen-komponen dalam DSS. Kedua, knowledge system yang berisi pengetahuan terkait domain permasalahan, baik berupa data maupun prosedur. Ketiga, sistem pemrosesan permasalahan yang bertindak sebagai penghubung antara sistem bahasa dan knowledge system, serta menyediakan kemampuan manipulasi masalah yang dibutuhkan dalam pengambilan Keputusan [11]. Adapun tujuan utama dari SPK adalah membantu pengambil keputusan dalam memilih alternatif terbaik pada persoalan semi-terstruktur, serta meningkatkan efisiensi proses pengambilan keputusan [12].

B. Analytical Hierarchy Process (AHP)

AHP merupakan salah satu metode yang digunakan dalam SPK untuk menyelesaikan permasalahan kompleks yang tidak terstruktur dengan cara memecahnya ke dalam beberapa komponen dalam suatu susunan hierarki [13]. Metode ini memungkinkan adanya pemberian nilai subjektif mengenai tingkat kepentingan setiap variabel secara relatif, kemudian menetapkan variabel mana yang memiliki prioritas paling tinggi dalam memengaruhi hasil keputusan [14]. Secara lebih rinci, terdapat tiga prinsip dasar dalam AHP. Pertama, dekomposisi (*decomposition*), yaitu memecah persoalan utama menjadi unsur-unsurnya hingga tidak dapat dipecah lebih lanjut sehingga terbentuk hierarki yang jelas. Kedua, penilaian komparasi (*comparative judgment*), yaitu membuat perbandingan relatif antar elemen pada satu tingkat tertentu terhadap elemen yang berada pada tingkat di atasnya. Prinsip ini merupakan inti dari AHP karena menghasilkan matriks perbandingan berpasangan (*pairwise comparison*) yang akan digunakan untuk menentukan bobot prioritas. Ketiga, penentuan prioritas (*synthesis of priority*), yaitu proses sintesis bobot lokal menjadi bobot global dengan menggunakan nilai eigenvektor dari matriks perbandingan. Hasil akhir sintesis inilah yang menjadi dasar dalam menentukan alternatif terbaik [15].

C. Sistem Pakar

Sistem pakar atau expert system merupakan bagian dari sistem berbasis pengetahuan yang dirancang untuk mendukung pengambilan keputusan atau pemecahan masalah pada bidang tertentu. Sistem ini disebut sistem pakar karena berfungsi layaknya seorang profesional yang memiliki pengetahuan dan pengalaman dalam menyelesaikan permasalahan tertentu [16].

Dalam penelitian ini, sistem pakar digunakan sebagai sumber basis pengetahuan yang diperoleh dari para ahli yang berkompeten dalam menentukan peminatan siswa.

D. Objek Pakar dan Hasil Wawancara

Objek pakar dalam penelitian ini adalah tiga orang yang memiliki peran sentral dalam proses penentuan peminatan siswa di SMA Negeri 64 Jakarta, yaitu guru, wakil kurikulum, dan ketua pemilihan mata pelajaran. Guru dipilih karena berinteraksi langsung dengan siswa dalam pembelajaran dan penilaian akademik. Wakil kurikulum memiliki peran penting dalam merancang serta mengimplementasikan kurikulum, termasuk kebijakan peminatan. Sedangkan ketua pemilihan mata pelajaran bertanggung jawab dalam mengoordinasikan proses seleksi dan rekomendasi jurusan bagi siswa kelas X. Pemilihan ketiga narasumber ini didasarkan pada pengalaman, wawasan, serta tanggung jawab mereka yang langsung berkaitan dengan proses penentuan peminatan, sehingga informasi yang diberikan sangat relevan sebagai dasar pembangunan sistem.

Wawancara terstruktur yang dilakukan terhadap ketiga pakar menghasilkan beberapa temuan penting. Pertama, penentuan peminatan siswa didasarkan pada kombinasi nilai akademik (khususnya pada mata pelajaran yang berkaitan dengan IPA dan IPS), hasil kuesioner minat, serta lima soal tambahan yang menguji pengetahuan dasar siswa pada mata pelajaran IPA dan IPS. Kedua, proses penilaian dilakukan secara objektif dan transparan agar siswa memperoleh rekomendasi yang sesuai dengan kemampuan dan minat mereka. Ketiga, wakil kurikulum dan ketua pemilihan jurusan menekankan perlunya sistem pendukung keputusan berbasis komputer untuk membantu pemetaan data siswa secara otomatis, sehingga dapat mengurangi bias subjektif yang kerap muncul dalam pengambilan keputusan.

Informasi yang diperoleh dari hasil wawancara tersebut selanjutnya digunakan sebagai dasar untuk menyusun kriteria, menentukan pembobotan menggunakan metode AHP, serta merumuskan aturan (rule base) pada sistem pendukung keputusan yang dikembangkan.[17] Dengan demikian, sistem yang dihasilkan diharapkan mampu memberikan rekomendasi peminatan yang lebih akurat, objektif, dan sesuai dengan potensi siswa.

E. Algoritma Sistem Pakar

Sistem pendukung keputusan pemilihan peminatan siswa kelas X di SMA Negeri 64 Jakarta Timur menggunakan pendekatan modifikasi dari metode AHP. Algoritma ini mengintegrasikan beberapa variabel penting yang menjadi dasar penilaian, yaitu nilai akademik, minat siswa (keinginan), kesempatan, serta skor soal teori pada masing-masing mata pelajaran utama.

Tahapan perhitungan dalam sistem dapat dijelaskan sebagai berikut:

1) Normalisasi Nilai Akademik

Nilai mata pelajaran dinormalisasi ke rentang 0–1 dengan cara membagi nilai siswa terhadap nilai maksimal (100). Apabila nilai lebih dari 100, maka dianggap bernilai 1.0.

2) Konversi Skor Minat (Keinginan)

Jawaban siswa mengenai minat terhadap suatu mata pelajaran dikonversi menjadi skor dengan kategori sebagaimana ditunjukkan pada Tabel 1.

TABEL 1.
KONVERSI SKOR MINAT

Indeks	Nilai
Sangat Tinggi	1.0
Tinggi	0.75
Cukup	0.50
Kurang	0.30

3) Konversi Skor Kesempatan

Aspek kesempatan juga diukur berdasarkan tanggapan siswa, yang kemudian dikonversi sebagaimana ditunjukkan pada Tabel 2.

TABEL 2.
KONVERSI SKOR KESEMPATAN

Indeks	Nilai
Sangat Tinggi	1.0
Tinggi	0.75

4) Skor Soal Teori

Siswa diberikan soal teori dengan jumlah maksimal 5 soal untuk tiap mata pelajaran. Setiap jawaban benar bernilai 0.03, sehingga skor teori maksimal per mata pelajaran adalah 0.15.

5) Perhitungan Skor Akhir per Mata Pelajaran

Skor akhir setiap mata pelajaran dihitung dengan formula:

$$\begin{aligned} \text{Skor Akhir} = & (0.6 \times \text{Nilai Normalisasi}) + \\ & (0.25 \times \text{Skor Keinginan}) + \\ & (0.15 \times \text{Skor Kesempatan}) + \\ & (0.03 \times \text{Skor Teori}) \end{aligned}$$

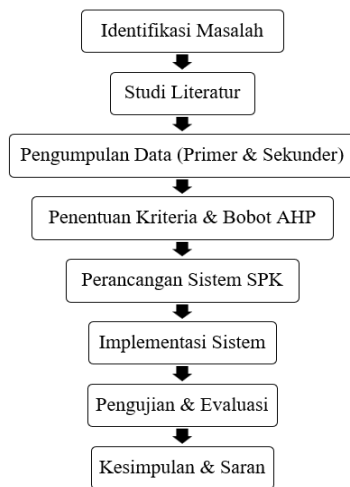
6) Rekapitulasi Skor IPA dan IPS

- Total IPA dihitung dari jumlah skor akhir mata pelajaran Matematika, Fisika, Kimia, dan Biologi.
- Total IPS dihitung dari jumlah skor akhir mata pelajaran Geografi, Ekonomi, Sosiologi, dan Sejarah.

7) Pengambilan Keputusan Rekomendasi Jurusan

- Jika Total IPA > Total IPS, maka rekomendasi jurusan adalah IPA.
- Jika Total IPS > Total IPA, maka rekomendasi jurusan adalah IPS.
- Jika Total IPA = Total IPS, sistem akan membandingkan skor per mata pelajaran pada kedua rumpun secara berurutan (descending). Jurusan dengan mata pelajaran yang memiliki skor dominan lebih tinggi akan ditetapkan sebagai rekomendasi akhir.

F. Tahapan Penelitian



Gambar 1. Tahapan Penelitian

G. Tabel Pakar

Tabel pakar adalah representasi formal pengetahuan pakar dalam format tabel yang berisi semua kriteria, indikator, serta alternatif keputusan yang dipertimbangkan sistem.

TABEL 3.
PAKAR

No	Kriteria	Indikator	Bobot	Penjelasan
1	Nilai Akademik	Nilai mapel (Matematika, Fisika, Kimia, Biologi, Geografi, Ekonomi, Sosiologi, Sejarah)	0.60	Nilai siswa permapel dinormalisasi ke 0-1 (nilai/100)
2	Minat	Jawaban kuesioner minat per mapel	0.25	Skor: Sangat Tinggi=1, Tinggi=0.75, Cukup=0.5, Kurang=0.3
3	Kesempatan	Jawaban kesiapan memilih mapel (Ya/Tidak)	0.15	Skor: Ya=1.0, Tidak=0.0
4	Soal Teori	Jumlah benar dari 5 soal per mapel	0.03/soal	Setiap jawaban benar 0.03, total maksimal 0.15 per mapel (5 soal x 0.03)

H. Rule-rule Pada Pakar

Rule sistem dirumuskan dalam bentuk if-then statements (jika-maka) yang dapat diuraikan secara teknis sebagai berikut:

1) Rule Penentuan Skor Akhir Mapel

If nilai akademik tinggi, minat tinggi, kesempatan “ya”, dan soal teori banyak benar, then skor akhir mapel tinggi.

Rumus:

$$\text{Skor Akhir} = (0.6 \times \text{Nilai Normalisasi}) + (0.25 \times \text{Skor Minat}) + (0.15 \times \text{Skor Kesempatan}) + (0.03 \times \text{Jumlah Soal Benar})$$

2) Rule Penentuan Total Skor Rumpun

a. If total skor 4 mapel IPA lebih besar dari total skor 4 mapel IPS, then rekomendasi IPA.

b. If total skor IPS lebih besar, then rekomendasi IPS.

3) Rule Kondisi Skor Sama

a. If total skor IPA = IPS, then urutkan skor akhir tiap mapel dalam masing-masing rumpun dari tertinggi ke terendah.

b. If skor tertinggi pada mapel IPA lebih besar dari skor tertinggi IPS, then rekomendasi IPA, dan sebaliknya.

4) Rule Tambahan

If skor semua mapel IPA dan IPS benar-benar seimbang, then prioritas akhir pada preferensi minat siswa, lalu kesiapan (kesempatan).

Contoh implementasi rule dalam kode Python:

```
if total_ipa > total_ips: rekomendasi = "IPA"
elif total_ips > total_ipa: rekomendasi = "IPS"
else:
```

```
# Jika sama, bandingkan skor per mapel
if sorted_ipa > sorted_ips: rekomendasi = "IPA"
else:
```

```
rekomendasi = "IPS"
```

I. Pohon Keputusan Pakar

Pohon keputusan menggambarkan urutan pengambilan keputusan berdasarkan kombinasi skor dan kriteria secara visual/teknis.

Deskripsi *Decision Tree*:

1) Input nilai, soal teori, minat, kesempatan untuk 8 mapel

2) Hitung skor akhir tiap mapel dengan rumus di atas

3) Jumlahkan total skor rumpun IPA dan IPS

4) Bandingkan total skor:

a. Jika total IPA > IPS → Rekomendasi IPA

b. Jika total IPS > IPA → Rekomendasi IPS

5) Jika total sama:

a. Urutkan skor tiap mapel dalam rumpun secara descending

b. Rumpun dengan skor per mapel tertinggi menjadi rekomendasi

c. Jika tetap sama, gunakan minat dan kesiapan sebagai pembeda terakhir

III. HASIL DAN PEMBAHASAN

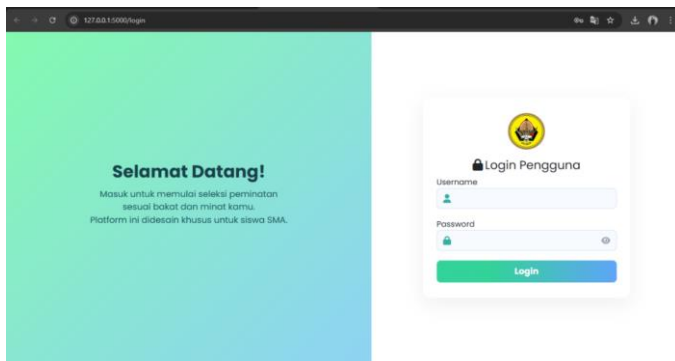
A. Arsitektur Sistem dan Hak Akses User

Sistem pendukung keputusan pemilihan peminatan siswa ini dirancang dengan tiga jenis pengguna utama, yaitu siswa, guru, dan admin. Siswa memiliki akses untuk melakukan login, mengisi data diri, memasukkan nilai, mengerjakan soal teori, serta mengisi kuesioner minat. Setelah seluruh data terinput, siswa dapat melihat hasil rekomendasi penjurusan yang bersifat personal. Guru berperan sebagai pihak yang dapat mengakses hasil rekapitulasi penjurusan siswa untuk keperluan monitoring dan bimbingan. Sementara itu, admin memiliki hak akses penuh terhadap sistem, meliputi login, pengelolaan data pengguna (tambah, edit, dan hapus), reset password, monitoring hasil seluruh siswa, serta pengaturan administrasi sistem.

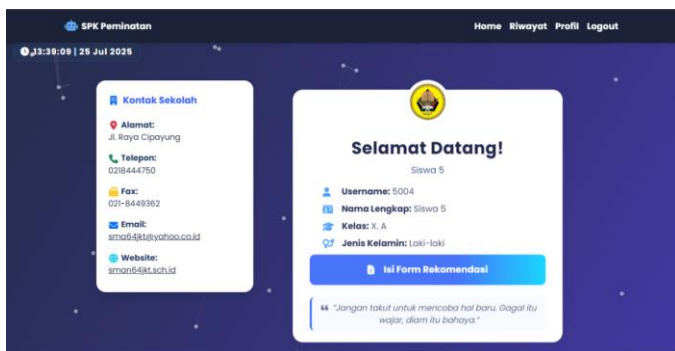
Alur utama sistem diawali dengan proses login yang menyesuaikan tampilan dashboard sesuai dengan role pengguna. Siswa kemudian dapat menginput nilai, mengerjakan soal teori, serta menjawab kuesioner minat. Selanjutnya, sistem akan melakukan proses perhitungan untuk menentukan rekomendasi jurusan berdasarkan algoritma yang telah dirancang. Guru dan admin dapat melihat hasil rekap penjurusan siswa, sementara admin juga bertanggung jawab terhadap pengelolaan data pengguna serta manajemen sistem secara keseluruhan.

B. User Interface

Halaman Siswa: Form input data, form soal teori, form minat & kesiapan, serta halaman hasil rekomendasi pribadi.

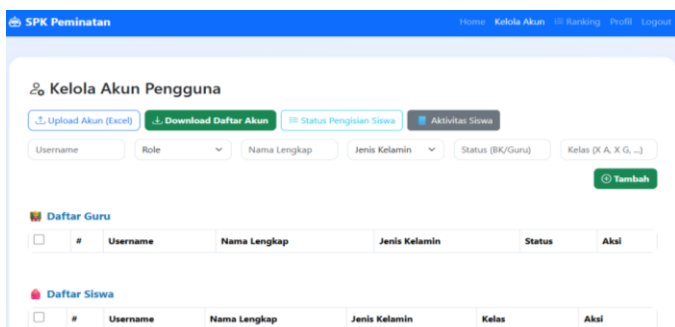


Gambar 2. Halaman Login



Gambar 3. Halaman Dashboard Siswa

Halaman Admin: Menu kelola user (tambah/edit/hapus), monitoring seluruh data siswa,



Gambar 4. Halaman Kelola Akun Admin

C. Implementasi Kode Program

1) Autentikasi dan Manajemen User Multi-Role

Setiap user (siswa/guru/admin) melakukan login untuk mengakses fitur sesuai peran/role. Autentikasi dilakukan dengan mencocokkan username dan password yang tersimpan di database users. Role digunakan untuk membedakan hak akses.

Berikut kodenya:

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        conn = sqlite3.connect(DATABASE)
        c = conn.cursor()
        c.execute('SELECT password, role FROM users WHERE
        username=?;', (username,))
        data = c.fetchone()
        conn.close()
        if data and data[0] == password:
            session['user'] = username
            session['role'] = data[1]
            return redirect(url_for('home'))
        else:
            flash('Username atau password salah', 'danger')
            return render_template('login.html')
```

Kode program ini merupakan bagian dari aplikasi web yang berperan dalam proses autentikasi pengguna. Pada tahap awal, pengguna diarahkan ke halaman login yang menyediakan form untuk memasukkan username dan password. Data yang diinput kemudian dikirim ke server melalui metode POST, lalu sistem mengambil informasi username dan password tersebut untuk diverifikasi. Proses verifikasi dilakukan dengan membuka koneksi ke basis data SQLite, di mana informasi akun pengguna tersimpan. Sistem mencari username yang dimaksud, dan jika ditemukan, akan mengambil data password beserta peran (role) pengguna.

Selanjutnya dilakukan proses validasi dengan membandingkan password yang dimasukkan dengan password yang tersimpan di basis data. Jika validasi berhasil, sistem memberikan otorisasi login dan menyimpan identitas serta hak akses pengguna ke dalam session, sehingga status login tetap dikenali selama sesi berlangsung. Pengguna yang berhasil masuk kemudian diarahkan ke halaman utama aplikasi.

Sebaliknya, jika username tidak terdaftar atau password tidak sesuai, sistem menampilkan pesan kesalahan "Username atau password salah" dan tetap menahan pengguna di halaman login untuk mencoba kembali. Dengan demikian, fungsi utama dari kode ini adalah memvalidasi username dan password yang dimasukkan, sekaligus menyimpan informasi role ke dalam session sebagai penentu hak akses pengguna, baik sebagai siswa, guru, maupun administrator.

2) Manajemen User (CRUD User, Admin Only)

Admin bisa menambah, melihat, dan menghapus user dari sistem melalui halaman admin. Ini penting agar admin bisa

mengelola seluruh pengguna, baik siswa, guru, maupun admin lain.

Berikut kodenya:

```
@app.route('/admin/user/add', methods=['GET', 'POST'])
def admin_user_add():
    if 'role' not in session or session['role'] != 'admin':
        return redirect(url_for('login'))
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        nama = request.form['nama']
        role = request.form['role']
        conn = sqlite3.connect(DATABASE)
        c = conn.cursor()
        c.execute('INSERT INTO users (username, password,
nama, role) VALUES (?, ?, ?, ?)',
            (username, password, nama, role))
        conn.commit()
        conn.close()
        flash('User berhasil ditambahkan', 'success')
        return redirect(url_for('admin_users'))
    return render_template('admin_user_add.html')
```

Kode ini merupakan bagian dari fitur khusus admin yang berfungsi untuk menambahkan pengguna baru secara manual ke dalam sistem aplikasi. Prosesnya dimulai dengan memastikan bahwa hanya pengguna dengan role admin yang dapat mengakses halaman ini, sedangkan pengguna lain akan otomatis diarahkan kembali ke halaman login. Saat halaman diakses, admin disediakan sebuah formulir HTML (admin_user_add.html) untuk menginput data pengguna baru, seperti username, password, nama, dan role.

Setelah formulir diisi dan dikirim menggunakan metode POST, aplikasi akan mengambil data tersebut kemudian melakukan koneksi ke database SQLite. Selanjutnya, sistem mengeksekusi perintah SQL INSERT untuk menyimpan data pengguna baru ke dalam tabel users. Jika penyimpanan berhasil, aplikasi akan menampilkan pesan konfirmasi "User berhasil ditambahkan" sebagai notifikasi kepada admin, lalu secara otomatis mengarahkan admin kembali ke halaman daftar pengguna (admin_users) untuk melihat data terbaru yang tersimpan. Namun, jika metode yang digunakan bukan POST, maka sistem hanya menampilkan formulir input untuk menambahkan data pengguna baru. Fungsi utama dari kode ini adalah menambahkan user baru (Create), sedangkan fungsi lainnya seperti Read, Update, dan Delete tersedia pada route berbeda di aplikasi, tetapi tetap terbatas hanya untuk admin.

3) Input Data Siswa

Siswa wajib mengisi data nilai, kemudian mengerjakan soal teori, lalu mengisi minat dan kesiapan pada setiap mata pelajaran. Data disimpan per siswa, dan diproses bertahap sebelum masuk proses penjurusan.

Berikut kodenya:

```
@app.route('/input_nilai', methods=['GET', 'POST'])
def input_nilai():
    if 'role' not in session or session['role'] != 'siswa':
        return redirect(url_for('login'))
    if request.method == 'POST':
```

```
        nilai = {mapel: request.form[mapel] for mapel in
['matematika', 'fisika', 'kimia', 'biologi', 'geografi', 'ekonomi', 'sosi
ologi', 'sejarah']}
        session['nilai_mapel'] = nilai
        return redirect(url_for('soal_teor'))
    return render_template('input_nilai.html')
```

Kode ini merupakan bagian dari proses input nilai siswa dalam aplikasi pemilihan minat atau penjurusan berbasis web. Pada tahap awal, hanya pengguna dengan peran siswa yang dapat mengakses halaman ini. Jika pengguna belum login atau tidak memiliki role siswa, maka sistem akan otomatis mengarahkan kembali ke halaman login. Setelah berhasil masuk, siswa disajikan sebuah form untuk mengisi nilai dari delapan mata pelajaran, yaitu Matematika, Fisika, Kimia, Biologi (kelompok IPA), serta Geografi, Ekonomi, Sosiologi, dan Sejarah (kelompok IPS). Ketika siswa mengisi form tersebut dan menekan tombol submit (POST), sistem akan mengolah data nilai dari masing-masing mata pelajaran dan menyimpannya dalam bentuk dictionary. Data nilai tersebut kemudian disimpan sementara dalam session Flask dengan nama nilai_mapel, sehingga dapat diproses lebih lanjut tanpa hilang.

Setelah penyimpanan berhasil, siswa langsung diarahkan ke halaman /soal_teor yang berfungsi untuk menampilkan soal-soal teori sebagai tahapan berikutnya dalam proses penjurusan. Jika halaman ini diakses untuk pertama kali (tanpa submit), maka sistem hanya menampilkan form pengisian nilai (input_nilai.html) agar siswa dapat mengisi data nilai. Dengan demikian, fungsi utama kode ini adalah membatasi akses input nilai hanya untuk siswa dan menyimpan data nilai delapan mata pelajaran ke dalam session agar dapat digunakan pada tahap selanjutnya.

4) Kode Route Soal Teori

```
@app.route('/soal_teor', methods=['GET', 'POST'])
def soal_teor():
    if 'role' not in session or session['role'] != 'siswa':
        return redirect(url_for('login'))
    # (kode soal teori dan penilaian di sini)
    if request.method == 'POST':
        # Skor tiap mapel dihitung
        skor_soal_teor = {}
        # Simpan ke session
        session['skor_soal_teor'] = skor_soal_teor
        return redirect(url_for('kuesioner_minat'))
    return render_template('soal_teor.html')
```

Kode di atas merepresentasikan alur siswa dalam mengerjakan soal teori pada sistem penjurusan atau pemilihan minat berbasis web. Saat siswa mengakses halaman /soal_teor, sistem terlebih dahulu melakukan pengecekan untuk memastikan bahwa pengguna sudah login dengan peran sebagai siswa. Jika tidak memenuhi syarat tersebut, pengguna akan langsung diarahkan kembali ke halaman login.

Setelah berhasil masuk, siswa dapat melihat daftar soal teori yang sesuai dengan mata pelajaran terkait. Jawaban yang diisi oleh siswa akan dikirim melalui form menggunakan metode

POST. Data jawaban ini kemudian diproses oleh sistem untuk menghitung skor soal teori masing-masing mata pelajaran. Hasil perhitungan tersebut disimpan dalam sebuah dictionary bernama skor_soal_teoris, lalu dimasukkan ke dalam session Flask agar tetap tersimpan dan dapat digunakan pada tahapan selanjutnya.

Setelah skor berhasil tersimpan, siswa akan diarahkan ke halaman kuesioner_minat, yaitu tahap berikutnya di mana siswa mengisi kuesioner minat sebagai bagian dari proses penjurusan. Jika siswa baru pertama kali membuka halaman ini tanpa mengirim jawaban, maka sistem akan langsung menampilkan halaman soal_teoris.html berisi soal-soal teori yang harus dikerjakan. Dengan demikian, fungsi utama kode ini adalah menghitung serta menyimpan skor soal teori per mata pelajaran untuk mendukung proses penentuan peminatan siswa.

5) Kode Route Kuesioner Minat

```
@app.route('/kuesioner_minat', methods=['GET', 'POST'])
def kuesioner_minat():
    if 'role' not in session or session['role'] != 'siswa':
        return redirect(url_for('login'))
    if request.method == 'POST':
        minat = {mapel: request.form['minat_'+mapel] for mapel
in
['matematika', 'fisika', 'kimia', 'biologi', 'geografi', 'ekonomi', 'sosi
ologi', 'sejarah']}
        session['minat'] = minat
        return redirect(url_for('kuesioner_lanjutan'))
        return render_template('kuesioner_minat.html')
@app.route('/kuesioner_lanjutan', methods=['GET', 'POST'])
def kuesioner_lanjutan():
    if 'role' not in session or session['role'] != 'siswa':
        return redirect(url_for('login'))
    if request.method == 'POST':
        lanjutan = {}
```

Tahap selanjutnya setelah siswa menyelesaikan soal teori adalah mengisi Kuesioner Minat. Sebelum dapat mengakses halaman ini, sistem memastikan bahwa pengguna sudah login dengan peran sebagai siswa, dan jika tidak, akan diarahkan kembali ke halaman login. Pada halaman ini, siswa diminta untuk mengisi form kuesioner yang dirancang untuk mengukur tingkat ketertarikan mereka terhadap delapan mata pelajaran, yaitu Matematika, Fisika, Kimia, Biologi, Geografi, Ekonomi, Sosiologi, dan Sejarah.

Jawaban yang diberikan siswa kemudian diproses oleh sistem dan disimpan dalam sebuah dictionary bernama minat, yang selanjutnya disimpan sementara dalam session Flask agar dapat digunakan pada tahap berikutnya. Setelah form dikirim, siswa otomatis diarahkan menuju halaman Kuesioner Lanjutan untuk melengkapi data tambahan yang dibutuhkan. Jika siswa baru pertama kali membuka halaman ini tanpa mengirim jawaban, maka sistem akan menampilkan halaman kuesioner_minat.html yang berisi pertanyaan untuk diisi.

Data yang terkumpul dari tahap ini nantinya akan dipadukan dengan informasi sebelumnya, seperti nilai mata pelajaran dan skor soal teori, guna menghasilkan rekomendasi

akhir mengenai jurusan yang paling sesuai bagi siswa. Dengan demikian, fungsi utama dari kode ini adalah untuk mengumpulkan data minat dan kesiapan siswa terhadap setiap mata pelajaran sebagai bagian dari proses penentuan peminatan.

6) Proses Penjurusan (AHP)

```
@app.route('/proses_penjurusan')
def proses_penjurusan():
    if 'role' not in session or session['role'] != 'siswa':
        return redirect(url_for('login'))
    nilai_mapel = session['nilai_mapel']
    minat = session['minat']
    lanjutan = session['lanjutan']
    skor_soal_teoris = session['skor_soal_teoris']
    hasil = calculate_ahp_with_kuesioner(nilai_mapel, minat,
lanjutan, skor_soal_teoris)
    # Simpan hasil ke database
    # ...
    return render_template('hasil.html', hasil=hasil)
```

Kode ini merupakan bagian inti dari sistem pemilihan minat atau proses penjurusan siswa setelah melalui beberapa tahapan pengumpulan data. Setelah siswa menyelesaikan pengisian nilai mata pelajaran (nilai_mapel), mengerjakan soal teori (skor_soal_teoris), mengisi kuesioner minat (minat), serta melengkapi pertanyaan lanjutan (lanjutan), seluruh data tersebut diproses pada route /proses_penjurusan. Pada tahap ini, aplikasi terlebih dahulu memastikan bahwa pengguna sudah login dengan peran sebagai siswa. Jika syarat tidak terpenuhi, pengguna otomatis diarahkan ke halaman login.

Data yang telah terkumpul kemudian diproses menggunakan fungsi calculate_ahp_with_kuesioner, yang merupakan implementasi dari metode Analytical Hierarchy Process (AHP). Metode ini digunakan untuk mengolah berbagai parameter, mulai dari nilai akademik, minat, hingga hasil soal teori, guna menentukan jurusan yang paling sesuai bagi siswa. Hasil perhitungan ini menghasilkan rekomendasi jurusan, seperti IPA atau IPS, yang selanjutnya dapat disimpan ke dalam database agar tersimpan permanen dan dapat diakses kembali oleh siswa maupun pihak sekolah.

Sebagai langkah akhir, sistem menampilkan hasil perhitungan dalam halaman hasil.html, sehingga siswa dapat langsung melihat rekomendasi jurusan yang sesuai dengan profil mereka. Dengan demikian, fungsi utama kode ini adalah memanggil proses perhitungan AHP untuk mengolah data siswa, menghasilkan rekomendasi jurusan, menyimpannya ke dalam database, serta menyajikannya dalam bentuk laporan yang dapat dilihat baik oleh siswa maupun guru atau admin sekolah.

7) Route Rekap Siswa

```
@app.route('/rekap_siswa')
def rekap_siswa():
    if 'role' not in session or session['role'] not in ['guru',
'admin']:
        return redirect(url_for('login'))
    conn = sqlite3.connect(DATABASE)
    c = conn.cursor()
```

```
c.execute('SELECT * FROM hasil')
hasil = c.fetchall()
conn.close()
return render_template('rekap_siswa.html', hasil=hasil)
```

Kode ini merupakan bagian dari aplikasi web yang ditujukan khusus bagi guru maupun admin sekolah untuk melakukan monitoring dan rekapitulasi hasil penjurusan siswa. Saat pengguna mengakses halaman /rekap_siswa, sistem terlebih dahulu memastikan bahwa pengguna telah login dengan peran guru atau admin. Jika syarat ini tidak terpenuhi, pengguna otomatis dialihkan ke halaman login.

Apabila akses valid, aplikasi kemudian terhubung ke database SQLite dan mengambil seluruh data dari tabel hasil, yang berisi rekomendasi akhir penjurusan siswa yang sebelumnya dihitung menggunakan metode *Analytical Hierarchy Process* (AHP). Data yang diperoleh disimpan dalam variabel hasil, lalu koneksi ke database segera ditutup untuk menjaga keamanan dan kinerja aplikasi. Selanjutnya, data tersebut dikirim ke halaman rekap_siswa.html agar dapat ditampilkan secara terstruktur. Dengan demikian, guru maupun admin dapat melihat keseluruhan hasil penjurusan siswa beserta skor yang dihasilkan, sehingga memudahkan mereka dalam melakukan evaluasi, pemantauan, dan analisis secara menyeluruh.

8) Pembatasan Kuota (penjelasan logika)

```
def cek_kuota_jurusan(jurusan):
    conn = sqlite3.connect(DATABASE)
    c = conn.cursor()
    c.execute('SELECT COUNT(*) FROM hasil WHERE
    rekomendasi=?', (jurusan,))
    jumlah = c.fetchone()[0]
    conn.close()
    if jurusan == 'IPA':
        return jumlah < 108
    else:
        return jumlah < 144
```

Jika kuota penuh:

- Siswa dengan skor terendah di jurusan tersebut digeser ke jurusan lain.

- Siswa baru dengan skor lebih tinggi menggantikan posisinya.

Fungsi ini dirancang khusus untuk memeriksa apakah jumlah siswa dalam suatu jurusan, baik IPA maupun IPS, telah mencapai batas kuota yang ditentukan sekolah. Setiap kali seorang siswa memperoleh rekomendasi jurusan, sistem akan memanggil fungsi ini dengan parameter jurusan yang bersangkutan. Prosesnya dimulai dengan membuka koneksi ke database untuk menghitung jumlah siswa yang sudah terdaftar pada jurusan tersebut berdasarkan data rekomendasi yang tersimpan di tabel hasil.

Setelah jumlah siswa diperoleh, koneksi ke database ditutup, lalu sistem membandingkan angka tersebut dengan kuota yang berlaku, yakni maksimal 108 siswa untuk jurusan IPA dan 144 siswa untuk jurusan IPS. Jika jumlah siswa masih berada di bawah batas kuota, fungsi akan mengembalikan nilai

True, menandakan bahwa jurusan tersebut masih dapat menampung siswa baru. Sebaliknya, jika kuota sudah penuh, fungsi mengembalikan nilai False sebagai indikator bahwa jurusan tidak lagi menerima tambahan siswa.

Pada bagian komentar kode juga dicantumkan rencana pengembangan lebih lanjut, yaitu mekanisme otomatis yang akan memindahkan siswa dengan skor terendah ke jurusan lain ketika kuota penuh, sehingga siswa baru dengan skor lebih tinggi dapat menggantikan posisinya. Dengan demikian, fungsi ini berperan penting untuk menjaga keseimbangan jumlah siswa per jurusan sesuai kebijakan sekolah sekaligus meminimalisasi kesalahan saat terjadi kelebihan kuota.

D. Pengujian Sistem

1) Pengujian *Blackbox*

Pengujian *blackbox* (*blackbox testing*) adalah metode pengujian perangkat lunak yang berfokus pada fungsi sistem dari sisi input dan output, tanpa memperhatikan struktur internal kode. Tujuan utamanya adalah memastikan bahwa setiap fitur berjalan sesuai kebutuhan, dan semua proses serta hak akses *user* berjalan benar.

Langkah-langkah pengujian:

1. Menentukan skenario pengujian sesuai fitur dan role user (siswa/guru/admin).
 2. Memberikan data input ke sistem.
 3. Mencatat output dan membandingkannya dengan output yang diharapkan.
 4. Menentukan status pengujian (OK/gagal) dan menganalisis hasil.
- 2) Skenario Pengujian dan Tabel Hasil
- a) Pengujian *Login* dan Hak Akses

TABEL 4.
PENGUJIAN LOGIN DAN HAK AKSES

No	Skenario	Input/User	Output Diharapkan	Status
1	Siswa login, input data & lihat hasil	Siswa	Bisa isi nilai, soal, minat, dan lihat hasil sendiri	OK
2	Admin login, kelola user	Admin	Bisa tambah/edit/hapus user	OK
3	Siswa coba akses data siswa lain	Siswa	Akses ditolak (error/redirect)	OK
4	Admin akses seluruh data	Admin	Bisa akses semua data siswa & user	OK

b) Pengujian Proses Penjurusan

TABEL 5.
PENGUJIAN PROSES PENJURUSAN

No	Skenario	Input	Output Diharapkan	Status
1	Nilai & minat IPA tinggi	Nilai IPA > IPS, minat IPA tinggi	Rekomendasi: IPA	OK
2	Nilai & minat IPS tinggi	Nilai IPS > IPA, minat IPS tinggi	Rekomendasi: IPS	OK
3	Semua nilai & minat sama	Nilai & minat IPA = IPS	Sistem pilih berdasar pohon keputusan	OK
4	Nilai sama, minat IPA lebih tinggi	Nilai sama, minat IPA > IPS	Rekomendasi: IPA	OK

c) Pengujian Kuota Jurusan

TABEL 6.
PENGUJIAN KUOTA JURUSAN

No	Skenario	Input	Output Diharapkan	Status
1	Kuota IPA penuh	109 siswa rekomendasi IPA	Siswa skor IPA terendah dialihkan ke IPS	OK
2	Kuota IPS penuh	145 siswa rekomendasi IPS	Siswa skor IPS terendah dialihkan ke IPA	OK

d) Cara Pelaksanaan Pengujian

Pengujian sistem dilakukan melalui serangkaian skenario yang disesuaikan dengan peran pengguna untuk memastikan seluruh fitur berjalan sesuai fungsi. Pada skenario siswa, pengguna login menggunakan akun siswa, kemudian mengisi data nilai, mengerjakan soal teori, melengkapi kuesioner minat dan kesiapan, hingga akhirnya memperoleh hasil rekomendasi jurusan. Pada skenario guru, pengguna login dengan akun guru untuk mengakses rekap hasil penjurusan siswa, memeriksa data yang tersimpan, serta memastikan bahwa guru hanya memiliki hak akses melihat data tanpa dapat melakukan perubahan.

Selanjutnya, pada skenario admin, pengguna login sebagai admin dengan kewenangan penuh untuk menambah, mengedit, maupun menghapus data pengguna, serta memantau seluruh data siswa yang tersedia di sistem. Untuk menguji fitur pembatasan kuota, dilakukan input data dummy secara massal hingga jumlah siswa melebihi batas kuota jurusan, kemudian diamati apakah sistem secara otomatis memindahkan siswa dengan skor terendah dan memberi prioritas pada siswa dengan skor lebih tinggi. Seluruh hasil uji dibandingkan dengan output yang diharapkan agar dapat dipastikan bahwa sistem bekerja sesuai rancangan dan bebas dari kesalahan logika maupun hak akses.

e) Analisis Hasil Pengujian

Hasil pengujian menunjukkan bahwa fitur login dan mekanisme hak akses telah berfungsi dengan baik, di mana setiap role hanya dapat mengakses fitur sesuai dengan kewenangannya masing-masing. Proses input data siswa hingga penjurusan berjalan secara otomatis, dan rekomendasi yang dihasilkan sesuai dengan logika perhitungan yang dirancang dalam sistem. Penerapan kuota jurusan juga terbukti efektif dalam membatasi jumlah siswa sesuai kapasitas yang ditentukan, sementara mekanisme pemindahan siswa berlangsung secara dinamis berdasarkan perbandingan skor.

Selain itu, sistem mampu menolak berbagai bentuk akses ilegal, misalnya upaya siswa untuk melihat data siswa lain atau percobaan guru mengubah data, dengan memberikan umpan balik berupa pesan error atau melakukan redirect ke halaman yang sesuai, sehingga keamanan dan kejelasan alur penggunaan tetap terjaga.

f) Catatan dan Saran Pengembangan

Pengujian yang dilakukan telah mencakup seluruh alur utama dalam sistem, sehingga dapat dipastikan bahwa setiap fungsi berjalan sesuai dengan rancangan. Hasil pengujian menunjukkan bahwa sistem ini dapat diandalkan dalam mendukung proses peminatan, baik dari aspek keamanan,

validasi data, maupun pemerataan distribusi siswa sesuai kuota yang berlaku.

Meski demikian, masih terdapat ruang untuk pengembangan lebih lanjut, antara lain dengan melakukan pengujian tambahan pada integrasi dengan database sekolah, penerapan sistem pelaporan otomatis, serta penambahan fitur notifikasi hasil yang dapat dikirim langsung ke email siswa maupun guru agar proses informasi menjadi lebih cepat dan efisien.

g) Cara Menjalankan Sistem

Sistem pendukung keputusan yang dikembangkan dapat dijalankan pada komputer atau laptop dengan Python serta library pendukung yang telah terpasang. Tahap awal yang perlu dilakukan adalah instalasi dependensi menggunakan perintah `pip install -r requirements.txt` (jika tersedia) untuk memastikan seluruh library yang dibutuhkan dapat digunakan.

Setelah itu, database disiapkan dengan menjalankan skrip atau fungsi `init_db()` guna membentuk tabel-tabel yang diperlukan, seperti `users`, `profil`, dan `hasil`. Selanjutnya, aplikasi dapat dijalankan melalui perintah `python app.py`, baik dalam mode CLI maupun menggunakan framework Flask untuk tampilan berbasis web. Apabila menggunakan antarmuka web, aplikasi dapat diakses melalui browser dengan membuka alamat `http://localhost:5000`, yang akan menampilkan halaman login. Pengguna kemudian dapat masuk sesuai dengan peran yang dimiliki (siswa, guru, atau admin) dan mengoperasikan fitur sesuai dengan hak akses masing-masing.

h) Dokumentasi dan *Support*

Pengguna yang mengalami kendala teknis, seperti lupa password, aktivasi akun, atau permasalahan lain, dapat langsung menghubungi admin sekolah yang memiliki peran sebagai pengelola sistem. Selain itu, admin juga bertanggung jawab melakukan backup database dan data pengguna secara rutin guna memastikan keamanan serta menjaga keutuhan informasi yang tersimpan di dalam sistem.

IV. PENUTUP

Berdasarkan hasil penelitian dan implementasi sistem pendukung keputusan untuk rekomendasi pemilihan mata pelajaran siswa kelas X di SMA Negeri 64 Jakarta Timur, dapat disimpulkan bahwa sistem berbasis web yang dikembangkan mampu memberikan rekomendasi peminatan secara objektif dan terstruktur dengan menggabungkan data akademik, minat, kesiapan, dan hasil teori siswa. Metode Analytical Hierarchy Process (AHP) yang diterapkan terbukti efektif dalam mengolah berbagai kriteria, menangani skor akhir yang sama, serta membagi kuota siswa sesuai kebijakan sekolah. Hasil pengujian menunjukkan seluruh fitur, termasuk login multi-peran, proses penjurusan, dan pengelolaan kuota, berfungsi dengan baik dan sesuai kebutuhan pengguna. Dengan sistem ini, proses peminatan siswa menjadi lebih efisien, transparan, serta mampu meminimalkan subjektivitas dalam pengambilan keputusan akademik.

REFERENSI

- [1] S. S. Febriani, A. E. Aulia, and S. ayudya Lestari, "Tes Minat dan Bakat: Peran Dalam Bimbingan dan Konseling Karier," *J. Ilmiah*

- Multidisiplin*, vol. 2, no. 5, pp. 457–471, 2025, doi: 10.62017/merdeka.
- [2] E. S. Jaya, “Manajemen Peminatan Peserta Didik Di Sekolah Menengah Atas Negeri (Penelitian Di Sma Negeri 2 Dan Sma Negeri 5 Kota Cimahi),” *J. Heal. Sains*, vol. 1, no. 7, pp. 843–858, 2020, doi: 10.46799/jsa.v1i7.126.
- [3] R. Aditia, S. Supriyono, M. Hamka, and H. Mustafidah, “Sistem Pendukung Keputusan Kelayakan Kenaikan Pangkat Tenaga Kependidikan Menggunakan Metode AHP-TOPSIS,” *Remik*, vol. 9, no. 1, pp. 242–253, 2025, doi: 10.33395/remik.v9i1.14285.
- [4] Y. M. Anjani, F. Muttakin, Z. Zamelly, and I. Permana, “Sistem Pendukung Keputusan Pemilihan Jurusan pada SMA menggunakan Metode Profile Matching,” *J. Comput. Syst. Informatics*, vol. 5, no. 3, pp. 680–689, 2024, doi: 10.47065/josyc.v5i3.5166.
- [5] Resky Kestar, “Sistem Pendukung Keputusan Pemilihan Sekolah Terbaik Di Kota Medan Dengan Menggunakan Metode AHP Berbasis Web,” *J. Multimed. dan Teknol. Inf.*, vol. 3, no. 02, pp. 53–60, 2022, doi: 10.54209/jatilima.v3i02.150.
- [6] A. Kurniawan, S. Setio Nugroho, R. Tullah, and A. Khoirunisa, “Implementasi Sistem Pendukung Keputusan AHP-TOPSIS untuk Pemilihan Guru TK Terbaik,” *Innov. Creat. Inf. Technol. J.*, vol. 11, no. 1, pp. 2654–8704, 2025, doi: 10.33050/icit.v11i1.3451.
- [7] F. Kurniawan, D. L. Rahma, and D. D. Apriyani, “Sistem Pendukung Keputusan Pemilihan Guru Terbaik SDN Curug 04 Menggunakan Metode AHP,” in *Seminar Nasional Riset dan Inovasi Teknologi (SEMNAS RISTEK)*, 2025, vol. 9, no. 1, p. 279, doi: 10.30998/faktorexacta.v15i4.11928.
- [8] N. F. Husnaini, “Sistem Pendukung Keputusan Berbasis Web dengan Metode AHP-TOPSIS untuk Pengukuran Tingkat Kesejahteraan Masyarakat Pesisir di Kabupaten Pidie,” *Comput. J.*, vol. 3, no. 2, pp. 51–60, 2025, doi: 10.58477/cj.v3i1.204.
- [9] S. Rinusantoro, I. Nuryaman, M. M. Yafin, and R. Aryansyah, “Perancangan Aplikasi Web Sistem Pendukung Keputusan Untuk Pemilihan Siswa Terbaik Menggunakan Metode SAW di MIS Nurul Ibad,” *Kohesi J. Multidisiplin Sainstek*, vol. 8, no. 11, pp. 1–8, 2025, doi: 10.2238/7e3ktw18.
- [10] Dahriansah, A. Nata, and I. R. Harahap, “Sistem Pendukung Keputusan Pemilihan Siswa Berprestasi Pada Aliyah Aras Kabu Agung Tanjungbalai Menggunakan Metode AHP,” *J. Teknol. Sist. Inf. dan Sist. Komput.*, vol. 86, no. 1, pp. 86–95, 2020, doi: 10.53513/jsk.v3i1.199.
- [11] G. S. Mahendra, *Sistem Pendukung Keputusan : Metode Fundamental & Perkembangannya*. Jambi: PT. Sonpedia Publishing Indonesia, 2024.
- [12] J. Khoirunnisa Anggraini and M. Orisa, “Sistem Pendukung Keputusan Pemilihan Guru Terbaik Dengan Metode Topsis Berbasis Web (Studi Kasus Sman 1 Kuaro),” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, pp. 1009–1015, 2023, doi: 10.36040/jati.v6i2.5422.
- [13] A. Salim and B. O. Lubis, “Sistem Pendukung Keputusan Pemilihan Jenis Tablet Gaming dengan Menggunakan Analytical Hierarchy Process,” *Pros. Semin. Nas. Energi Telekomun. dan Otomasi*, no. ISBN: 978-602-74127-4-3, pp. 1–9, 2017.
- [14] J. Parhusip, “Penerapan Metode Analytical Hierarchy Process (AHP) Pada Desain Sistem Pendukung Keputusan Pemilihan Calon Penerima Bantuan Pangan Non Tunai (BPNT) Di Kota Palangka Raya,” *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 13, no. 2, pp. 18–29, 2019, doi: 10.47111/jti.v13i2.251.
- [15] S. Kali and A. S. Purnomo, “Sistem Pendukung Keputusan Untuk Memprediksi Kualitas Hp Dalam Transaksi Jual Beli Online Dengan Metode Analytical Hierarchy Process (AHP),” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 2, pp. 1688–1695, 2024, doi: 10.36040/jati.v8i2.9150.
- [16] A. Z. Nashiruddin and R. Hidayat, “Perancangan Sistem Pakar Dalam Identifikasi Kerusakan Mesin Cuci Berbasis Naïve Bayes,” *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 8, no. 2, p. 241, 2022, doi: 10.24036/jtev.v8i2.115245.
- [17] A. Salim, B. O. Lubis, and A. Haidir, “Penentuan Karyawan Terbaik Dengan Metode TOPSIS pada PT Regency Motor,” *Saintekom*, vol. 12, no. 1, pp. 92–102, 2022, doi: 10.33020/saintekom.v12i1.203.