

Buku Petunjuk Penggunaan Aplikasi

SISTEM PREDIKSI RELEVANSI PEKERJAAN MAHASISWA DENGAN BIDANG STUDI MENGGUNAKAN ALGORITMA *RANDOM FOREST*

Oleh :

M Galih Fikran Syah

M. Nishom, M.Kom.

Dyah Apriliani, S.T. M.Kom.

1. PENDAHULUAN

1.1. Tujuan Pembuatan dokumen

Dokumen *user manual* “Sistem Prediksi Relevansi Pekerjaan Mahasiswa dengan Bidang Studi “ ini dibuat untuk tujuan seperti berikut :

1. Membantu Pengguna Memahami Fungsi dan Fitur Aplikasi

Tujuan utama dari manual book ini adalah memberikan pemahaman yang komperhensif kepada pengguna tentang fungsi dan fitur yang ada dalam aplikasi. Hal ini akan membanti pengguna dalam mengoperasikan aplikasi dengan efektif dan memaksimalkan manfaat yang dapat mereka peroleh dari penggunaan aplikasi tersebut.

2. Memberikan Panduan Penggunaan yang Jelas dan Terstruktur

Manual book ini bertujuan untuk menyajikan panduan penggunaan yang jelas dan terstruktur bagi pengguna. Dalam manual book ini, pengguna akan mendapatkan instruksi Langkah demi Langkah tentang cara menggunakan berbagai fitur, memasukkan data, dan mengakses informasi penting dalam aplikasi. Tujuannya adalah agar pengguna dapat dengan mudah mengikuti panduan yang disajikan dan merasa percaya diri dalam menjalankan aplikasi tersebut.

3. Menyediakan Tips dan Trik untuk Penggunaan Optimal

Selain memberikan panduan penggunaan dasar, tujuan *manual book* ini juga adalah memberikan tips dan trik kepada pengguna untuk menggunakan aplikasi secara optimal. Ini termasuk saran praktis, strategi, atau fitur-fitur yang mungkin tidak terlalu jelas pada panduan penggunaan dasar, tetapi dapat membantu pengguna dalam memperoleh hasil terbaik dari aplikasi.

4. Meningkatkan Pemecahan Masalah dan Dukungan Pengguna

Salah satu tujuan penting dari *manual book* ini adalah memberikan pemecaha masalah umum yang mungkin dihadapu oleh pengguna. Hal ini dapat mencakup penjelasan mengenai pesan kesalahan yang muncul, Langkah-langkah pemecahan masalah umum, atau cara menghubungi tim dukungan teknis jika dibutuhkan. Dengan menyajikan solusu yang jelas dan mendukung, tujun ini adalah untuk memberikan pengalaman pengguna yang lancar dan meminimalisir hamabatan yang mungkin terjadi.

1.2. Deskripsi Umum Sistem

1.2.1. Deskripsi Umum Aplikasi

Sistem Prediksi Relevansi Pekerjaan Mahasiswa dengan Bidang Studi merupakan sebuah sistem yang memungkinkan analisis relevansi pekerjaan mahasiswa berdasarkan bidang studi mereka. Dengan melibatkan bagian alumni sebagai admin, aplikasi ini bertujuan untuk membantu memperhatikan dan menjembatani mahasiswa dengan karir yang sesuai. Dengan menggunakan metode klasifikasi Random Forest dan berdasarkan variabel-variabel seperti jenis kelamin, IPK, jumlah sertifikat, prestasi, organisasi, penghasilan orang tua, dan pekerjaan orang tua, aplikasi ini dapat memberikan prediksi dengan tingkat akurasi sebesar 84%. Melalui dashboard aplikasi yang *user-friendly*, pengguna dapat melihat statistik analisis relevansi pekerjaan mahasiswa serta grafik visualisasi yang memperlihatkan tren dan pola terkait. Secara keseluruhan, aplikasi ini memberikan wawasan berharga dan bermanfaat bagi pengambilan keputusan karir mahasiswa berdasarkan bidang studi mereka.

1.2.2. Deskripsi Umum Kebutuhan Aplikasi

Deksripsi umum kebutuhan aplikasi yang akan diimplementasikan meliputi semua informasi yang bersifat teknis yang menjadi acuan dalam pengembangan aplikasi.

1.3. Deskripsi Dokumen

Dokumen ini dibuat untuk memberikan panduan penggunaan Aplikasi Analisis Relevansi Pekerjaan Mahasiswa dengan Bidang Studi. Dokumen ini berisikan informasi sebagai berikut :

1. BAB I.

Berisi informasi umum yang merupakan bagian pendahuluan, yang meliputi tujuan pembuatan dokumen, deksripsi umum sistem serta deksripsi dokumen.

2. BAB II.

Berisi perangkat yang dibutuhkan untuk penggunaan aplikasi meliputi perangkat lunak dan perangkat keras.

3. BAB III.

Berisi user manual Sistem Relevansi Pekerjaan Mahasiswa dengan Bidang Studi.

2. PERANGKAT YANG DIBUTUHKAN

2.1. Perangkat Lunak

Perangkat lunak yang digunakan adalah :

1. *Windows / Linux Operating System.*
2. *Web Browser*

2.2. Perangkat Keras

Perangkat keras yang digunakan adalah :

1. Laptop/PC (Personal Computer)
2. *Router (Internet)*

3. MENU DAN CARA PENGGUNAAN

3.1. Struktur Menu Utama

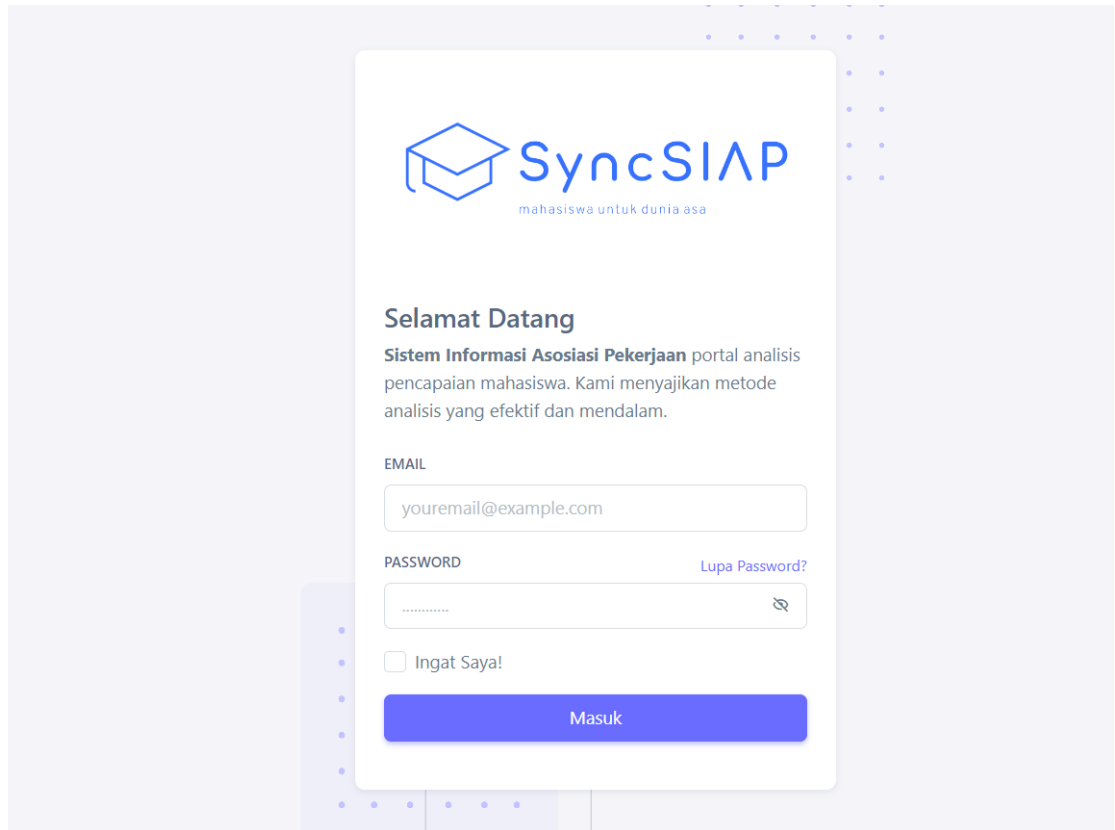
Adapun struktur halaman pada Aplikasi Prediksi Relevansi Pekerjaan Mahasiswa dengan Bidang Studi – “Sinkronisasi Sistem Informasi Asosiasi Pekerjaan (SyncSIAP)” adalah sebagai berikut :

- **Menu User Admin**
 - a. **Data master**
 - b. **Input data**
 - c. **Kelola prodi**
 - d. **Kelola admin prodi**
 - e. **Laporan**
 - f. **Settings**
- **Menu User Admin Prodi**
 - a. **Data master**
 - b. **Input data**
 - c. **Laporan**
 - d. **Settings**

3.2. Penggunaan

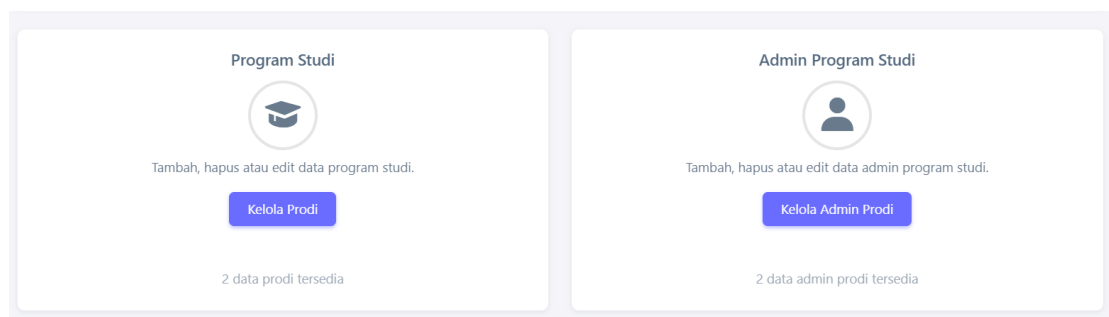
Pada bagian ini akan dijelaskan mengenai menu-menu yang ada pada aplikasi.

1. Login



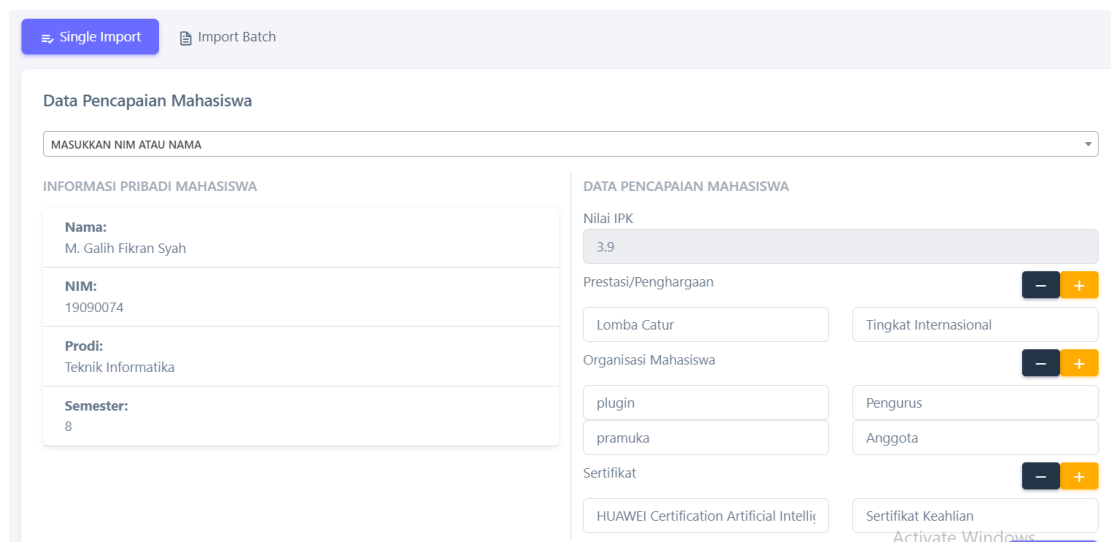
Login digunakan untuk autentikasi user sehingga tidak semua orang berhak mengakses aplikasi ini. User atau pengguna dalam aplikasi ini terdiri dari admin dan admin prodi, dalam hal ini yang bertindak sebagai admin adalah bagian alumni, lalu admin prodi adalah admin dari masing-masing prodi. Setelah mengisi form login user akan dibawa ke halaman dashboard aplikasi

2. Prodi management



Aplikasi memiliki fitur untuk mengelola admin dan admin prodi antara lain menambah, menghapus, atau mengedit prodi dan admin prodi.

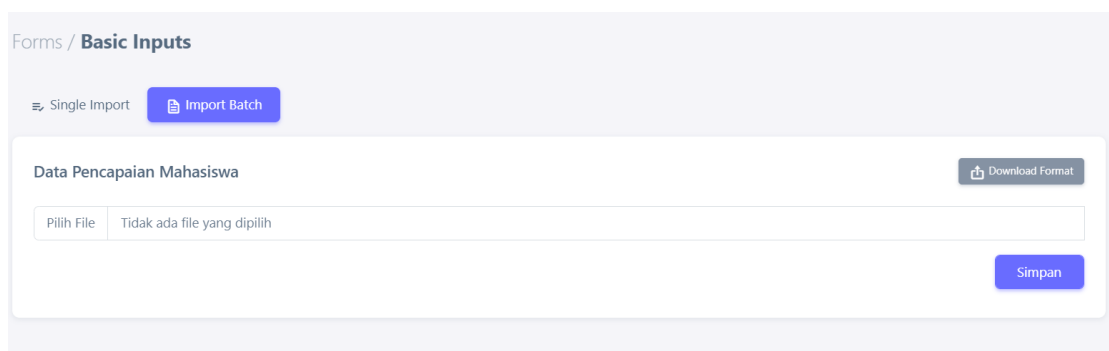
3. Input data



The screenshot shows the 'Single Import' form for 'Data Pencapaian Mahasiswa'. At the top, there are two tabs: 'Single Import' (active) and 'Import Batch'. Below the tabs is a search bar labeled 'MASUKKAN NIM ATAU NAMA'. The form is divided into two main sections: 'INFORMASI PRIBADI MAHASISWA' on the left and 'DATA PENCAPAIAN MAHASISWA' on the right. The 'INFORMASI PRIBADI MAHASISWA' section contains fields for 'Nama' (M. Galih Fikran Syah), 'NIM' (19090074), 'Prodi' (Teknik Informatika), and 'Semester' (8). The 'DATA PENCAPAIAN MAHASISWA' section contains fields for 'Nilai IPK' (3.9), 'Prestasi/Penghargaan' (Lomba Catur, Tingkat Internasional), 'Organisasi Mahasiswa' (plugin, pramuka), and 'Sertifikat' (HUAWEI Certification Artificial Intellig, Sertifikat Keahlian). There are minus and plus buttons next to the 'Prestasi/Penghargaan', 'Organisasi Mahasiswa', and 'Sertifikat' sections. An 'Activate Windows' watermark is visible at the bottom right.

Terdapat dua metode dalam penginputan data yang pertama yaitu dengan single input. User akan memilih mahasiswa berdasarkan nim dan atau nama kemudian mengisi data secara manual.

4. Import batch



The screenshot shows the 'Import Batch' form for 'Data Pencapaian Mahasiswa'. At the top, there are two tabs: 'Single Import' and 'Import Batch' (active). Below the tabs is a search bar labeled 'MASUKKAN NIM ATAU NAMA'. The form is divided into two main sections: 'INFORMASI PRIBADI MAHASISWA' on the left and 'DATA PENCAPAIAN MAHASISWA' on the right. The 'INFORMASI PRIBADI MAHASISWA' section contains fields for 'Nama', 'NIM', 'Prodi', and 'Semester'. The 'DATA PENCAPAIAN MAHASISWA' section contains fields for 'Nilai IPK', 'Prestasi/Penghargaan', 'Organisasi Mahasiswa', and 'Sertifikat'. There are minus and plus buttons next to the 'Prestasi/Penghargaan', 'Organisasi Mahasiswa', and 'Sertifikat' sections. A 'Download Format' button is located at the top right of the form. A 'Simpan' button is located at the bottom right of the form. An 'Activate Windows' watermark is visible at the bottom right.

Cara yang kedua dalam menginputkan data yaitu input massal. User dapat dengan mudah menginputkan banyak data mahasiswa dengan mengupload file berformat excel dan menyesuaikan format penulisan sesuai panduan.

5. Report & Rincian

Laporan hasil prediksi

-- PILIH PROGRAM STUDI --

-- PILIH TAHUN AKADEMIK --

Download Excel

NO	NAMA	SEMESTER	PRODI	RELEVAN	
1	Galih	8	Teknik Informatika	Relevan	⋮
2	Reaz	8	Teknik Informatika	Relevan	⋮
3	Andre	8	Teknik Informatika	Relevan	⋮
4	Rojali	8	Teknik Informatika	Tidak Relevan	⋮
5	Kibuli	8	Teknik Informatika	Tidak Relevan	⋮
6	Sasa	8	Teknik Informatika	Tidak Relevan	⋮
7	Dicky	8	Teknik Informatika	Relevan	⋮

Data Identitas Mahasiswa

NIM

18090125

Nama


Dwi Febi Fauzi

Jenis Kelamin

Laki-laki

Alamat

Jalan Pusponegoro 1, Desa Pekauman Rt 06 Rw 02



Data Akademik

IPK

3.67

Pencapaian Prestasi

1. Juara 3 Lomba Hology 3.0 Katagori App Inovation Tingkat Nasional
2. Juara 1 Showcase Competition UINIC 4.0 Tingkat Nasional

Data Sosial & Ekonomi

Pekerjaan Orang Tua

Tidak disebutkan

Penghasilan Orang Tua

< Rp. 1.000.000,00

Untuk melihat hasil prediksi dari system relevansi pekerjaan dengan bidang studi terdapat di menu laporan. User dapat melihat hasil prediksi mahasiswa yang relevan dan tidak relevan kemudian user juga dapat melihat rincian data yang terbagi menjadi informasi pribadi mahasiswa, data pencapaian akademik, data sosial & ekonomi, data pengalaman dan prestasi, dan data keaktifan mahasiswa dalam bersosial atau dalam berorganisasi.

6. Settings

Settings / **Notifications**

[Akun](#) [Ganti Password](#)

Pengaturan Akun

NAMA LENGKAP	JENIS KELAMIN
<input type="text" value="Galih Fikran Syah"/>	<input type="text" value="Laki-laki"/>
EMAIL	NO HP
<input type="text" value="galih11120@gmail.com"/>	<input type="text" value="08986676180"/>

[Simpan](#)

Settings / **Notifications**

[Akun](#) [Ganti Password](#)

Ganti Password

Masukkan Password Saat Ini

Masukkan Password Baru

Konfirmasi Password

Minimal 8 karakter

[Simpan](#)

Pada menu settings terdapat fitur untuk melengkapi informasi pengguna dan fitur untuk melakukan penggantian kata sandi.

Dokumen Teknikal

SISTEM PREDIKSI RELEVANSI PEKERJAAN MAHASISWA DENGAN BIDANG STUDI MENGGUNAKAN ALGORITMA *RANDOM FOREST*

Oleh :

M Galih Fikran Syah

M. Nishom, M.Kom.

Dyah Apriliani, S.T. M.Kom.

1. Profil

Aplikasi SyncSIAP merupakan sebuah sistem yang dirancang untuk menganalisis relevansi pekerjaan mahasiswa dengan bidang studi mereka. Dalam aplikasi ini, bagian alumni dari sebuah instansi pendidikan bertindak sebagai admin yang bertanggung jawab untuk memperhatikan dan menjembatani mahasiswa dengan karir. Sistem ini terintegrasi dengan sistem akademik yang ada di kampus melalui penggunaan API, yang memungkinkan pengambilan data mahasiswa secara langsung. Admin prodi pada setiap program studi memiliki peran dalam menginputkan variabel-variabel penting seperti jenis kelamin, IPK, jumlah sertifikat, jumlah prestasi, jumlah organisasi, penghasilan orang tua, dan pekerjaan orang tua.

Metode klasifikasi yang digunakan dalam sistem ini adalah Random Forest, yang telah memberikan tingkat akurasi sebesar 84%. Ini berarti aplikasi mampu memberikan prediksi yang akurat mengenai relevansi pekerjaan mahasiswa dengan bidang studi mereka. Dalam halaman dashboard aplikasi, pengguna dapat melihat informasi terperinci mengenai statistik analisis relevansi pekerjaan mahasiswa, serta grafik visualisasi yang memperlihatkan tren dan pola terkait hasil analisis tersebut. Aplikasi ini memberikan wawasan berharga bagi pengambilan keputusan yang lebih baik dalam pengembangan karir mahasiswa. Aplikasi tersebut dikembangkan dalam *platform* website menggunakan bahasa pemrograman Python, yang memungkinkan pemrosesan data yang efisien dan interaksi melalui antarmuka website yang *user-friendly*.

Secara keseluruhan, aplikasi SyncSIAP adalah sebuah solusi inovatif untuk memahami dan mengevaluasi relevansi pekerjaan mahasiswa berdasarkan berbagai variabel penting. Dengan integrasi yang kuat dengan sistem akademik dan penggunaan metode klasifikasi yang efektif, aplikasi ini memberikan kontribusi yang berarti dalam membantu mahasiswa dalam mengarahkan karir mereka sesuai dengan bidang studi yang dipilih.

2. Latar Belakang

Relevansi antara jurusan yang diampu mahasiswa selama perkuliahan dengan bidang pekerjaan yang didapat menjadi tolok ukur sebagai acuan tingkat keberhasilan perguruan tinggi dalam mencetak sumber daya yang baik dan sesuai dengan keahlian yang dimiliki. Berdasarkan data dari Badan Pusat Statistik tahun 2020 jumlah penduduk dengan tingkat pendidikan sarjana S1 sebanyak 80.747 orang yang terdiri dari 39.344 orang laki-laki dan 40.404 orang perempuan. Selama tahun 2020 jumlah penduduk yang bekerja 507.617 orang yang terdiri dari 312.005 orang laki-laki dan 195.612 orang perempuan. Jumlah

pengangguran terbuka 47.521 orang yang terdiri dari 17.514 orang perempuan dan 30.007 orang laki-laki, dengan tingkat pengangguran 8,56% dengan rincian tingkat pengangguran laki-laki 8,77% dan perempuan 8,22%, artinya masih banyak jumlah pengangguran yang ada saat ini sehingga ini menjadi hal yang sangat serius untuk diselesaikan[4]. Hal ini menjadi salah satu masalah yang sampai saat ini belum terselesaikan, masih banyak lulusan perguruan tinggi yang bekerja tidak sesuai dengan bidang keahliannya, lebih buruk lagi bahkan ada yang kesulitan dalam mendapatkan pekerjaan. Ini akan berdampak pada performa perusahaan jika memiliki karyawan yang tidak ahli pada *domain knowledge* perusahaan.

Berdasarkan temuan di atas, jika kondisi tersebut dibiarkan saja dan tidak mendapat perhatian yang serius maka akan menimbulkan efek yang sistematis bagi pendidikan dan pembangunan nasional. Oleh sebab itu diperlukan perhatian yang serius dari pemerintahan dan institusi penyelenggara pendidikan dan dukungan dari semua pihak terkait dalam menyelesaikan masalah tersebut. Salah satu langkah yang bisa dilakukan dalam menyelesaikan permasalahan tersebut adalah dengan memanfaatkan teknologi informasi. Banyak data berupa informasi dari berbagai bidang yang dihasilkan dari teknologi informasi yang canggih. Dalam dunia pendidikan teknologi informasi dapat menghasilkan data yang berlimpah berupa data history aktivitas perkuliahan mahasiswa misalnya nilai mata kuliah tertentu, organisasi yang pernah diikuti, publikasi yang pernah dikerjakan, hingga prestasi atau lomba yang pernah didapat. Dengan data-data tersebut selanjutnya diolah dengan teknik data mining sehingga menghasilkan *knowledge* atau informasi yang bermanfaat untuk mendukung pengambilan keputusan.

3. Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1) Manfaat bagi perguruan tinggi

- a. Untuk mengatasi persoalan ketidaksesuaian pekerjaan dengan bidang yang ditekuni mahasiswa
- b. Memberikan sumbangan pemikiran dari hasil penelitian bagi pihak perguruan tinggi dalam rangka memperbaiki kualitas lulusan mahasiswa.

2) Manfaat bagi mahasiswa

- a. Sebagai bahan refleksi bagi mahasiswa terhadap apa yang telah didapatkan pada perkuliahan sehingga mahasiswa dapat berbenah.
- b. Menjadi tolok ukur keberhasilan pembelajaran mahasiswa sehingga dapat meningkatkan

semangat belajar.

4. Spesifikasi Teknis

Spesifikasi Teknik Meliputi:

- Source Code

Berikut uraian spesifikasi yang digunakan untuk membangun aplikasi

- Python
- Web Browser
- Jupyter Notebook
- Visual Studio Code

Source Code

- Login

Route untuk menuju ke halaman login yaitu dengan mengakses /login. Fungsi login dapat menerima request dengan metode GET maupun POST.

```
@auth_bp.route("/login", methods=["GET", "POST"])
def login():
    """
    Log-in page for registered users.

    GET requests serve Log-in page.
    POST requests validate and redirect user to dashboard.
    """
    # Bypass if user is logged in
    if current_user.is_authenticated:
        return redirect(url_for("main_bp.dashboard"))

    form = LoginForm()
    # Validate login attempt
    if form.validate_on_submit():
        user = User.query.filter_by(email=form.email.data).first()
        if user and user.check_password(password=form.password.data):
            login_user(user)
            next_page = request.args.get("next")
            flash("Berhasil Login!")
            return redirect(next_page or url_for("main_bp.dashboard"))
        flash("Username & Password Salah")
        return redirect(url_for("auth_bp.login"))
```

```

return render_template(
    "login.jinja2",
    form=form,
    title="Log in.",
    template="login-page",
    body="Log in with your User account.",
)

```

- Single Input

Single Input digunakan untuk menginputkan data per mahasiswa. Setiap data prestasi, organisasi dan sertifikat akan melalui pembobotan terlebih dahulu sebelum dimasukkan kedalam database.

```

@main_bp.route("/input_data", methods=["POST"])
def input_data():
    """Single Import"""

    # pembobotan terhadap attribut utama
    prestasi =
    galih_helper.Pembobotan.pembobotan_prestasi(request.form.getlist('prestasi'))
    organisasi =
    galih_helper.Pembobotan.pembobotan_organisasi(request.form.getlist('organisasi'))
    sertifikat =
    galih_helper.Pembobotan.pembobotan_sertifikat(request.form.getlist('sertifikat'))

    nama_prestasi = request.form.getlist('nama_prestasi')
    nama_organisasi = request.form.getlist('nama_organisasi')
    nama_sertifikat = request.form.getlist('nama_sertifikat')

    # attribut tambahan
    nim = request.form['nim2']
    name = request.form['nama']
    gender = 0 if request.form['jk'].lower() == 'p' else 1
    # state = 0 if request.form['state'].lower() == 'Aktif' else 1
    # batch_year = request.form['batch_year']
    gpa_score = request.form['ipk']
    parents_income = request.form['parents_income']
    pekerjaan_ortu = request.form['pekerjaan_ortu']
    data_mhs = Mahasiswa.query.filter_by(nim=nim).first()

    # reset achievement dulu supaya tidak redundant

```

```

    all_sertifikat = Sertifikat.query.filter(Sertifikat.mahasiswa_id ==
data_mhs.id).all()
    all_prestasi = Prestasi.query.filter(Prestasi.mahasiswa_id ==
data_mhs.id).all()
    all_organisasi = Organisasi.query.filter(Organisasi.mahasiswa_id ==
data_mhs.id).all()

    # loop for delete
    if all_sertifikat:
        for sert in all_sertifikat:
            db.session.delete(sert)
        db.session.commit()
    if all_prestasi:
        for pres in all_prestasi:
            db.session.delete(pres)
        db.session.commit()
    if all_organisasi:
        for org in all_organisasi:
            db.session.delete(org)
        db.session.commit()

    # Masukkin achievement ke database
    # 1. prestasi
    for index, value in enumerate(request.form.getlist('prestasi')):
        prestasi_store = Prestasi(mahasiswa_id=data_mhs.id,
nama_prestasi=nama_prestasi[index], jenis_prestasi=value)
        db.session.add(prestasi_store)
        db.session.commit()

    # 2. organisasi
    for index, value in enumerate(request.form.getlist('organisasi')):
        organisasi_store = Organisasi(mahasiswa_id=data_mhs.id,
nama_organisasi=nama_organisasi[index], peran_organisasi=value)
        db.session.add(organisasi_store)
        db.session.commit()

    # 3. Sertifikat
    for index, value in enumerate(request.form.getlist('sertifikat')):
        sertifikat_store = Sertifikat(mahasiswa_id=data_mhs.id,
nama_sertifikat=nama_sertifikat[index], jenis_sertifikat=value)
        db.session.add(sertifikat_store)
        db.session.commit()

    if not data_mhs:
        # create data
        mhs = Mahasiswa(nim=nim, gender=gender, gpa_score=gpa_score,
name=name, sertifikat=sertifikat, organisasi=organisasi,
prestasi=prestasi, parents_income=parents_income,
pekerjaan_ortu=pekerjaan_ortu)

```

```

        db.session.add(mhs)
        db.session.commit()

        # predict relevance
        res = galih_helper.PredictModel.predict(nim)
        data_mhs.relevan = res[0]
        data_mhs.predict_proba = res[1]

        db.session.commit()
    else:
        # update data
        res = galih_helper.PredictModel.predict(nim)

        existing_mhs = Mahasiswa.query.filter_by(nim=nim).first()
        existing_mhs.gpa_score = gpa_score
        existing_mhs.prestasi = prestasi
        existing_mhs.organisasi = organisasi
        existing_mhs.sertifikat = sertifikat
        existing_mhs.gender = gender
        existing_mhs.relevan = res[0]
        existing_mhs.predict_proba = res[1]
        # existing_mhs.state = state
        # existing_mhs.batch_year = batch_year

        db.session.commit()

    return redirect(url_for("main_bp.basic_input"))

```

- Import Batch

Import batch digunakan ketika ingin menginputkan banyak data sekaligus. Sama seperti single input namun import batch memanfaatkan file excel yang di import dan dibaca datanya kemudian diolah dan dimasukkan kedalam database.

```

@main_bp.route("/input_batch", methods=["POST"])
def input_batch():
    """Import Batch"""
    # Read the File using Flask request
    file = request.files['file']
    # generate filename

    # save file in local directory
    # file.save("upload/"+file.filename)
    # book = xlrd.open_workbook("upload/"+file.filename)
    # sheet = book.sheet_by_name()

    kumpulin_ayok = []

```

```

nim_compare = None
data = pandas.read_excel(file)
for index, row in data[0:len(data.index)].iterrows():
    # each row is returned as a pandas series
    nim = row['nim']
    nama = row['nama']
    jk = True if row['jk'] == 'laki-laki' else False
    ipk = row['ipk']
    penghasilan_ortu = row['penghasilan_ortu']
    pekerjaan_ortu = row['pekerjaan_ortu']
    tahun_akademik = row['tahun_akademik']

    nama_prestasi = row['nama_prestasi'] if row['nama_prestasi'] ==
row['nama_prestasi'] else False
    tingkat_prestasi = row['tingkat_prestasi'] if
row['tingkat_prestasi'] == row['tingkat_prestasi'] else False
    nama_sertifikat = row['nama_sertifikat'] if
row['nama_sertifikat'] == row['nama_sertifikat'] else False
    jenis_sertifikat = row['jenis_sertifikat'] if
row['jenis_sertifikat'] == row['jenis_sertifikat'] else False
    nama_organisasi = row['nama_organisasi'] if
row['nama_organisasi'] == row['nama_organisasi'] else False
    peran_organisasi = row['peran_organisasi'] if
row['peran_organisasi'] == row['peran_organisasi'] else False

    if nim_compare != nim or nim_compare is None:
        nim_compare = nim
        current_mahasiswa =
Mahasiswa.query.filter_by(nim=nim).first()

        if current_mahasiswa:
            if nama: current_mahasiswa.name = nama
            if ipk: current_mahasiswa.gpa_score = ipk
            if jk: current_mahasiswa.gender = jk
            if penghasilan_ortu: current_mahasiswa.parents_income =
penghasilan_ortu
            if pekerjaan_ortu: current_mahasiswa.pekerjaan_ortu =
pekerjaan_ortu
            if tahun_akademik: current_mahasiswa.batch_year =
tahun_akademik
            db.session.commit()
        else :
            mhs = Mahasiswa(nim=nim, gender=jk, gpa_score=ipk,
name=nama, parents_income=penghasilan_ortu,
pekerjaan_ortu=pekerjaan_ortu, batch_year=tahun_akademik)
            db.session.add(mhs)
            db.session.commit()

```



```

        kumpulin_ayok.append(
            {
                'nim': nim,
                'prestasi': [nama_prestasi, tingkat_prestasi],
                'sertifikat': [nama_sertifikat, jenis_sertifikat],
                'organisasi': [nama_organisasi, peran_organisasi],
            }
        )

    else:
        nim_compare = nim
        kumpulin_ayok.append(
            {
                'nim': nim,
                'prestasi': [nama_prestasi, tingkat_prestasi],
                'sertifikat': [nama_sertifikat, jenis_sertifikat],
                'organisasi': [nama_organisasi, peran_organisasi],
            }
        )

start_nim = None
count_index = 0
removed_list = []

while count_index < len(kumpulin_ayok):
    prestasi_per_mahasiswa = []
    sertifikat_per_mahasiswa = []
    organisasi_per_mahasiswa = []
    for index, ka in enumerate(kumpulin_ayok):
        if start_nim == ka['nim'] or start_nim is None:
            if index not in removed_list:
                start_nim = ka['nim']
                removed_list.append(index)

                if ka['prestasi'][0] is not False:
                    prestasi_per_mahasiswa.append([ka['prestasi'][0]
, galih_helper.Pembobotan.bobot['prestasi'][ka['prestasi'][1]]])
                if ka['sertifikat'][0] is not False:
                    sertifikat_per_mahasiswa.append([ka['sertifikat'
][0], galih_helper.Pembobotan.bobot['sertifikat'][ka['sertifikat'][1]]])
                if ka['organisasi'][0] is not False:
                    organisasi_per_mahasiswa.append([ka['organisasi'
][0], galih_helper.Pembobotan.bobot['organisasi'][ka['organisasi'][1]]])

    if (index+1) not in removed_list:
        if start_nim is not None:
            res = galih_helper.PredictModel.predict(start_nim)
            mhs = Mahasiswa.query.filter_by(nim=start_nim).first()

```

```

        mhs.prestasi =
galih_helper.Pembobotan.pembobotan_prestasi(prestasi_per_mahasiswa)
        mhs.sertifikat =
galih_helper.Pembobotan.pembobotan_sertifikat(sertifikat_per_mahasiswa)
        mhs.organisasi =
galih_helper.Pembobotan.pembobotan_organisasi(organisasi_per_mahasiswa)

        mhs.relevan = res[0]
        mhs.predict_proba = res[1]
        db.session.commit()

        # insert organisasi
        if organisasi_per_mahasiswa:
            for org in organisasi_per_mahasiswa:
                org_new = Organisasi(mahasiswa_id=mhs.id,
nama_organisasi=org[0], peran_organisasi=org[1])
                db.session.add(org_new)
                db.session.commit()

        # insert sertifikat
        if sertifikat_per_mahasiswa:
            for sert in sertifikat_per_mahasiswa:
                sert_new = Sertifikat(mahasiswa_id=mhs.id,
nama_sertifikat=sert[0], jenis_sertifikat=sert[1])
                db.session.add(sert_new)
                db.session.commit()

        # insert prestasi
        if prestasi_per_mahasiswa:
            for pres in prestasi_per_mahasiswa:
                pres_new = Prestasi(mahasiswa_id=mhs.id,
nama_prestasi=pres[0], jenis_prestasi=pres[1])
                db.session.add(pres_new)
                db.session.commit()

        count_index = count_index + 1
        start_nim = None

    form = MahasiswaForm()
    nim = request.args.get('nim')
    mahasiswa = [] if not nim else
Mahasiswa.query.filter_by(nim=nim).first()
    detail_mahasiswa = {'data': [[]]} if not nim else
galih_helper.Api.get_mhs(nim)

    # cuman buat pengingat
    list_selection = [(t.id, t.nim) for t in Mahasiswa.query.all()]
    print(list_selection)
    # hahaha

```

```

return render_template(
    "data/basic_input.jinja2",
    title="Basic Input",
    template="dashboard-template",
    current_user=current_user,
    mahasiswa=mahasiswa,
    detail_mahasiswa=detail_mahasiswa['data'][0],
    form=form,
)

```

- Prediksi

Tentunya prediksi adalah metode yang paling utama dalam aplikasi ini. Fungsi prediksi akan menerima parameter yang dibutuhkan untuk melakukan klasifikasi kemudian mengolah data tersebut sesuai dengan kebutuhan model. Selanjutnya fungsi ini akan mengambil model dan segera memprediksi relevansi sesuai data yang diberikan.

```

class PredictModel():
    def predict(nim):
        model = load_model("/opt/project_ta/siputa/app/model/best.pkl")
        mahasiswa = Mahasiswa.query.filter_by(nim=nim).first()
        x_pred = pd.DataFrame([[mahasiswa.gender,
mahasiswa.pekerjaan_ortu, mahasiswa.parents_income, mahasiswa.gpa_score,
mahasiswa.sertifikat, mahasiswa.prestasi, mahasiswa.organisasi]],
columns=["jk", "pekerjaan_ortu", "penghasilan_ortu", "ipk",
"sertifikasi", "prestasi", "organisasi"])
        x_pred['pekerjaan_ortu'].replace(['buruh', 'wiraswasta',
'pegawai swasta', 'swasta', 'pns'],
[1, 2, 2, 2, 3], inplace=True)
        res = model.predict(x_pred)[0]
        predictions = model.predict_proba(x_pred)
        tmp = predictions.tolist()

        return [0 if res == 'Tidak Relevan' else 1,
round(predictions.tolist()[0][0], 2) * 100]

    def predict_multiple(nims):
        # load model
        model = load_model("/opt/project_ta/siputa/app/model/best.pkl")
        results = []
        # search student by nim
        for nim in nims:
            mahasiswa = Mahasiswa.query.filter_by(nim=nim[0]).first()
            if mahasiswa:
                # prepare the data

```

```

        # gender = "L" if mahasiswa.gender else "P"
        x_pred = pd.DataFrame([[mahasiswa.gender,
mahasiswa.pekerjaan_ortu, mahasiswa.parents_income, mahasiswa.gpa_score,
mahasiswa.sertifikat, mahasiswa.prestasi, mahasiswa.organisasi]],
columns=["jk", "pekerjaan_ortu", "penghasilan_ortu", "ipk",
"sertifikasi", "prestasi", "organisasi"])
        x_pred['pekerjaan_ortu'].replace(['buruh', 'wiraswasta',
'pegawai swasta', 'swasta', 'pns'],
[1, 2, 2, 2, 3], inplace=True)

        # predict
        res = model.predict(x_pred)[0]
        data_result = {
            "nama": mahasiswa.name,
            "nim": mahasiswa.nim,
            "semester": 8,
            "prodi": "Teknik Informatika",
            "relevan": res
        }

        results.append(data_result)
    return results

```