

NOTE: This template is shareware downloaded from www.processimpact.com. All shareware payments are donated to the Norm Kerth Benefit Fund to help a consultant who is disabled with a brain injury. Please visit http://www.processimpact.com/norm_kerth.html to make a shareware payment (\$10 suggested). Thank you!

Software Requirements Specification

for

AstroClassify

Version 1.0 approved

Prepared by:

1. Kevin Jonathan Rotty (103052330047)
2. Faaris Khairrudin (103052300115)
3. Fauzan Ahsanudin Alfikri (103052300003)
4. Muhammad Fikri Hanif (103052300118)
5. Farrel Farruqh Effendi (103052330092)

Kelompok MAHALL DS-47-03
25-03-2025

eLearning versions of several popular Process Impact training seminars are available at www.processimpact.com/elearning.shtml, including "In Search of Excellent Requirements," "Exploring User Requirements with Use Cases," "Writing High-Quality Requirements," "Software Inspections and Peer Reviews," and "Project Management Best Practices". Single-user and corporate-wide site licenses are both available.

DAFTAR ISI

Daftar Isi	i
Revision History	ii
Jobdesc Anggota Kelompok	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.2.1 Format dan Penulisan.....	1
1.2.2 Konvensi Penulisan Kode	1
1.3 Intended Audience and Reading Suggestions	2
1.4 Project Scope.....	2
1.5 References	3
2. Overall Description.....	3
2.1 Product Perspective	3
2.2 Product Features	4
2.2.1 Data Input.....	4
2.2.2 Model Selection	4
2.2.3 Result Visualization	5
2.2.4 Export Result.....	5
2.3 User Classes and Characteristics.....	5
2.4 Operating Environment.....	5
2.5 Design and Implementation Constraints	6
2.6 User Documentation.....	6
2.7 Assumptions and Dependencies.....	6
3. System Features	6
3.1 Data Input & Preprocessing	6
3.1.1 Description and Priority	6
3.1.2 Stimulus/Response Sequences	7
3.1.3 Functional Requirements	7
3.2 Model Selection.....	7
3.2.1 Description and Priority	7
3.2.2 Stimulus/Response Sequences	7
3.2.3 Functional Requirements	7
3.3 Classification & Model Performance Comparison	8
3.3.1 Description and Priority	8
3.3.2 Stimulus/Response Sequences	8
3.3.3 Functional Requirements	8
3.4 Result Visualization	8
3.4.1 Description and Priority	8

3.4.2	Stimulus/Response Sequences	8
3.4.3	Functional Requirements	9
3.5	User Interaction & Export Feature	9
3.5.1	Description and Priority	9
3.5.2	Stimulus/Response Sequences	9
3.5.3	Functional Requirements	9
4.	Skenario Testing	10
4.1	Data Input – 7 Skenario Testing.....	10
4.2	Model Selection – 5 Skenario Testing	11
4.3	Result Visualization – 6 Skenario Testing	12
4.4	Export Result – 7 Skenario Testing.....	13
5.	Hasil Testing (Testing Results).....	14
5.1	Tujuan Pengujian.....	14
5.2	Metode Pengujian.....	14
5.3	Hasil Pengujian	15
5.3.1	Fitur Data Input	15
5.3.2	Fitur Model Selection.....	16
5.3.3	Fitur Result Visualization	17
5.3.4	Fitur Export Result.....	18
6.	Interface Aplikasi.....	20
6.1	Data Input.....	20
6.2	Model Selection.....	21
6.3	Result Visualitation	22
6.4	Export Result.....	23
7.	Praktik Git & Github.....	24

Revision History

Name	Date	Reason For Changes	Version

Jobdesc Anggota Kelompok

No	Anggota Kelompok	Jobdesc
1.	Kevin Jonathan Rotty	Menangani fitur 1 (Data input) + implementasi pada streamlit
2.	Faaris Khairrudin	Menangani fitur 2 (preprocessing & Modeling) + implementasi pada streamlit
3.	Farrel Farruqh Effendi	Menangani fitur 3 (model selection) + implementasi pada streamlit
4.	Fauzan Ahsanudin Alfikri	Menangani fitur 4 (Result Visualization) + implementasi pada streamlit
5.	Muhammad Fikri Hanif	Menangani fitur 5 (Export Results) + implementasi pada streamlit

1. Introduction

1.1 Purpose

AstroClassify adalah aplikasi berbasis data science yang dirancang untuk mengklasifikasikan dan memvisualisasikan objek astronomi seperti bintang, galaksi, dan quasar berdasarkan karakteristik spektralnya menggunakan algoritma machine learning.

Tujuan utama aplikasi ini adalah:

1. Membantu astronom, peneliti, dan para penikmat astronomi dalam mengidentifikasi objek langit berdasarkan data pengamatan dari Sloan Digital Sky Survey (SDSS).
2. Menyediakan alat untuk eksplorasi data astronomi secara interaktif dengan bantuan teknik visualisasi data.
3. Memudahkan klasifikasi otomatis menggunakan algoritma machine learning, seperti Random Forest, XGBoost, dan LightGBM.

Dokumen ini menjelaskan spesifikasi perangkat lunak AstroClassify, termasuk fitur yang didukung, antarmuka pengguna, persyaratan sistem, serta batasan dan asumsi yang digunakan dalam pengembangan sistem.

1.2 Document Conventions

1.2.1 Format dan Penulisan

1. Istilah teknis pertama kali disebutkan dalam dokumen akan dijelaskan dalam Appendix A: Glossary.
2. Persyaratan wajib ditandai dengan kata kunci "harus" (*must*), sedangkan persyaratan opsional menggunakan "dapat" (*may*).
3. Semua diagram mengikuti standar UML (Unified Modeling Language) untuk merepresentasikan arsitektur dan proses sistem.
4. Kutipan atau rujukan ke dokumen lain diberikan dalam format [Nomor Referensi], misalnya [1].

1.2.2 Konvensi Penulisan Kode

1. Nama kelas menggunakan CamelCase, misalnya DataProcessor.
2. Variabel dan fungsi menggunakan snake_case, misalnya preprocess_data().
3. Paradigma pemrograman:

- a. Functional Programming (FP) diterapkan dalam tahap preprocessing data untuk menjaga immutability dan meningkatkan keandalan kode.
- b. Object-Oriented Programming (OOP) digunakan untuk desain arsitektur utama, seperti pengelolaan objek dan modul dalam sistem.
4. Standar kode lebih lanjut dapat ditemukan dalam 2.5 Design and Implementation Constraints.

1.3 Intended Audience and Reading Suggestions

Dokumen ini ditujukan untuk berbagai pemangku kepentingan yang memiliki peran dalam pengembangan dan penggunaan AstroClassify, termasuk:

1. **Astronom & Peneliti:** Untuk menganalisis data pengamatan dan mengklasifikasikan objek langit berdasarkan karakteristik spektralnya.
2. **Data Scientist & Machine Learning Engineer:** Untuk memahami metode analisis data dan algoritma klasifikasi yang digunakan dalam aplikasi.
3. **Pengembang Perangkat Lunak:** Untuk memahami spesifikasi sistem, arsitektur perangkat lunak, dan mengembangkan fitur yang dibutuhkan.
4. **Penggemar Astronomi:** Untuk mengeksplorasi data astronomi dan mendapatkan wawasan tentang objek langit berdasarkan data observasi dari SDSS.

Saran Pembacaan:

1. **Astronom & Peneliti** disarankan untuk membaca **bagian 1 (Introduction)** dan **bagian 4 (External Interface Requirements)** untuk memahami fitur aplikasi dan cara penggunaannya dalam penelitian.
2. **Data Scientist & Machine Learning Engineer** lebih disarankan untuk fokus pada **bagian 3 (System Features)** dan **bagian 5 (Nonfunctional Requirements)** untuk mengetahui algoritma yang digunakan dan aspek teknis lainnya.
3. **Pengembang Perangkat Lunak** dapat langsung membaca **bagian 2 (Overall Description)** dan **bagian 3 (System Features)** untuk memahami bagaimana aplikasi ini dirancang dan dikembangkan.
4. **Penggemar Astronomi** disarankan untuk membaca **bagian 4 (External Interface Requirements)** dan **Appendices** untuk mengeksplorasi fitur aplikasi dan cara penggunaannya.

1.4 Project Scope

AstroClassify dikembangkan untuk memberikan solusi yang lebih **cepat, akurat, dan otomatis** dalam menganalisis data astronomi. Beberapa manfaat utama dari perangkat lunak ini meliputi:

1. Meningkatkan efisiensi klasifikasi objek astronomi dengan otomatisasi menggunakan machine learning.
2. Mempermudah eksplorasi dan analisis data spektral melalui visualisasi interaktif.

3. Mendukung penelitian dan edukasi dalam bidang astronomi dengan menyediakan alat analisis berbasis data.

Mengoptimalkan proses identifikasi objek langit, mengurangi ketergantungan pada klasifikasi manual yang rentan terhadap kesalahan.

1.5 References

Dokumen ini merujuk pada sumber dan standar berikut:

1. Sloan Digital Sky Survey (SDSS) Database - <https://www.sdss.org/>
2. Scikit-learn Documentation - <https://scikit-learn.org/>
3. Pandas Documentation - <https://pandas.pydata.org/>
4. Matplotlib Documentation - <https://matplotlib.org/>
5. Bokeh Documentation - <https://docs.bokeh.org/>
6. Plotly Documentation - <https://plotly.com/python/>

2. Overall Description

2.1 Product Perspective

AstroClassify adalah aplikasi berbasis Python yang dikembangkan sebagai produk baru dan self-contained untuk klasifikasi objek astronomi. Aplikasi ini tidak merupakan bagian dari sistem yang lebih besar atau penerus dari aplikasi sebelumnya, melainkan dikembangkan secara mandiri untuk menyediakan alat klasifikasi berbasis machine learning yang dapat dijalankan secara lokal tanpa ketergantungan pada cloud computing.

AstroClassify dirancang untuk astronom amatir, mahasiswa, serta peneliti yang ingin mengklasifikasikan objek langit berdasarkan fitur tertentu. Input yang digunakan dalam sistem ini terdiri dari alpha, delta, serta magnitudo pada berbagai filter fotometri (u, g, r, i, z). Berdasarkan fitur tersebut, aplikasi akan mengklasifikasikan objek ke dalam beberapa kategori, seperti Bintang, Galaksi, dan Quasar.

Aplikasi ini sepenuhnya dijalankan melalui Streamlit, yang memungkinkan pengguna untuk mengunggah data, melakukan analisis, serta menampilkan hasil klasifikasi dalam bentuk interaktif melalui browser. AstroClassify dapat beroperasi sebagai sistem mandiri, tetapi juga dapat diintegrasikan dengan database astronomi eksternal (misalnya, katalog bintang) untuk memperkaya analisis data serta menghasilkan visualisasi yang mendukung pemahaman hasil klasifikasi.

2.2 Product Features

AstroClassify dirancang untuk memberikan pengalaman yang **mudah digunakan, cepat, dan akurat** dalam mengklasifikasikan objek astronomi berdasarkan data spektralnya. Berikut adalah fitur utama dari aplikasi ini:

2.2.1 Data Input

AstroClassify memungkinkan pengguna untuk menginput data astronomi tanpa label guna diklasifikasikan ke dalam kategori yang sesuai. Terdapat dua metode input:

1. **Upload Dataset:** Pengguna dapat mengunggah file CSV yang berisi data spektral objek astronomi tanpa label kelas.
2. **Input Manual:** Pengguna dapat memasukkan satu atau beberapa data objek langsung melalui antarmuka aplikasi.

Proses preprocessing yang diterapkan sebelum klasifikasi:

1. **Penanganan Nilai Hilang:** Menggunakan metode imputasi atau penghapusan data yang tidak lengkap.
2. **Normalisasi dan Standarisasi:** Menyesuaikan skala data agar sesuai dengan kebutuhan model.
3. **Seleksi Fitur:** Menghilangkan variabel yang kurang relevan agar meningkatkan akurasi model.

2.2.2 Model Selection

AstroClassify menyediakan pilihan model machine learning yang telah dilatih sebelumnya, yang memungkinkan pengguna memilih model untuk proses klasifikasi:

1. **Random Forest**
2. **XGBoost**
3. **LightGBM**

Sebelum pengguna melakukan klasifikasi, mereka dapat melihat performa dari masing-masing model yang telah kami latih sebelumnya dengan dataset referensi. Metrik evaluasi yang tersedia meliputi:

1. **Akurasi Model**
2. **Precision, Recall, dan F1-score**
3. **Confusion Matrix**

Pengguna dapat memilih model yang paling sesuai sebelum menjalankan klasifikasi data yang mereka masukkan.

2.2.3 Result Visualization

AstroClassify menyajikan hasil klasifikasi dalam berbagai bentuk visualisasi yang interaktif dan informatif:

1. **3D Scatter Plot (Plotly):**
 - Menampilkan distribusi objek astronomi dalam ruang **Right Ascension (α)**, **Declination (δ)**, dan **Redshift (z)**.
 - Warna objek ditentukan berdasarkan hasil klasifikasi.
 - Pengguna dapat melakukan filter terhadap visualisasi berdasarkan objek yang dipilih.
2. **Histogram & Distribution Plot:**
 - Menunjukkan distribusi spektral untuk melihat pola karakteristik masing-masing kelas.

2.2.4 Export Result

Export Hasil Analisis dan visualisasi, Hasil klasifikasi dan visualisasi dapat diunduh dalam format CSV atau PNG untuk dokumentasi lebih lanjut.

2.3 User Classes and Characteristics

1. Astronom & Data Scientist & Peneliti: Menggunakan aplikasi untuk analisis ilmiah. Menggunakan aplikasi untuk keperluan edukasi dan penelitian.
2. Pengguna Awam/penggemar astronomi: Dapat memanfaatkan visualisasi interaktif untuk eksplorasi data astronomi.

2.4 Operating Environment

1. **OS:** Windows, Linux, macOS
2. **Python:** Versi 3.8 atau lebih baru
3. **Pustaka utama:**
 - Pandas
 - NumPy
 - Scikit-learn
 - Matplotlib/Bokeh/Plotly
 - Streamlit

2.5 Design and Implementation Constraints

1. Dataset dari Sloan Digital Sky Survey (SDSS) cukup besar (100.000 observasi, 17 fitur spektral).
2. Perlu optimasi memori saat loading dan preprocessing data untuk menjaga performa.
3. Python 3.8 atau lebih baru wajib digunakan untuk kompatibilitas dengan pustaka ML dan data science.

2.6 User Documentation

1. Manual penggunaan dalam bentuk Markdown yang ditampilkan langsung di Streamlit.
2. Contoh kode interaktif menggunakan `st.code()` untuk membantu pemahaman penggunaan aplikasi.
3. Fitur unggah file (CSV) dengan `st.file_uploader()` untuk memasukkan data input.
4. Navigasi antar bagian dokumentasi menggunakan sidebar dengan `st.sidebar.radio()`.
5. Visualisasi hasil klasifikasi menggunakan `st.pyplot()` atau `st.dataframe()`.
6. Bagian troubleshooting dalam format expander (`st.expander()`) untuk solusi cepat.

2.7 Assumptions and Dependencies

1. Dataset dari Sloan Digital Sky Survey (SDSS) dapat diakses dan digunakan tanpa batasan lisensi atau perubahan format. Model menggunakan Scikit-learn.
2. AstroClassify akan berjalan dengan lancar di sistem operasi utama seperti Windows, macOS, dan Linux, dengan Python 3.8+.
3. Pustaka seperti Scikit-learn, Pandas, Matplotlib, dan Bokeh tetap didukung dan tidak mengalami perubahan besar selama pengembangan.

3. System Features

3.1 Data Input & Preprocessing

3.1.1 Description and Priority

Fitur ini bertanggung jawab untuk memungkinkan pengguna mengunggah atau memasukkan data astronomi serta melakukan preprocessing sebelum klasifikasi.

- **Prioritas:** Tinggi

3.1.2 Stimulus/Response Sequences

1. **Input:** Dataset mentah dari SDSS atau input manual
2. **Proses:**
 - Menghapus nilai yang hilang atau outlier.
 - Melakukan normalisasi dan standarisasi data.
 - Melakukan seleksi fitur untuk meningkatkan akurasi model.
3. **Output:** Dataset yang sudah dibersihkan, dinormalisasi, dan siap digunakan untuk klasifikasi.

3.1.3 Functional Requirements

1. **REQ-1:** Sistem harus dapat menangani nilai yang hilang dengan metode imputasi atau penghapusan baris.
2. **REQ-2:** Sistem harus dapat menormalkan dan menstandarisasi data menggunakan metode seperti Min-Max Scaling atau Standardization.
3. **REQ-3:** Sistem harus dapat melakukan seleksi fitur berdasarkan relevansi terhadap model.

3.2 Model Selection

3.2.1 Description and Priority

Fitur ini memungkinkan pengguna untuk memilih model machine learning yang akan digunakan dalam proses klasifikasi.

- **Prioritas:** Tinggi

3.2.2 Stimulus/Response Sequences

1. **Input:** Dataset yang telah diproses
2. **Proses:**
 - Menyediakan pilihan algoritma seperti Random Forest, XGBoost, dan LightGBM.
 - Menampilkan metrik evaluasi model yang telah dilatih.
3. **Output:** Model yang telah dipilih dan evaluasi performanya.

3.2.3 Functional Requirements

1. **REQ-4:** Sistem harus mendukung algoritma Random Forest, XGBoost, dan lightGBM untuk klasifikasi.
2. **REQ-5:** Sistem harus menampilkan metrik evaluasi seperti akurasi, precision, recall, dan F1-score.

3. **REQ-6:** Sistem harus memungkinkan pengguna memilih model sebelum melakukan klasifikasi.

3.3 Classification & Model Performance Comparison

3.3.1 Description and Priority

Fitur ini memungkinkan pengguna untuk menjalankan klasifikasi serta membandingkan performa model yang berbeda.

- **Prioritas:** Tinggi

3.3.2 Stimulus/Response Sequences

1. **Input:** Data yang telah diproses dan model yang dipilih
2. **Proses:**
 - Menjalankan klasifikasi terhadap data yang dimasukkan.
 - Menampilkan hasil prediksi untuk setiap objek.
 - Membandingkan performa model berdasarkan metrik evaluasi.
3. **Output:** Hasil klasifikasi dan perbandingan model dalam bentuk visualisasi.

3.3.3 Functional Requirements

1. **REQ-7:** Sistem harus dapat menjalankan klasifikasi menggunakan model yang dipilih.
2. **REQ-8:** Sistem harus menampilkan hasil prediksi untuk setiap objek astronomi.
3. **REQ-9:** Sistem harus menyajikan perbandingan performa model dalam bentuk grafik seperti confusion matrix dan ROC curve.

3.4 Result Visualization

3.4.1 Description and Priority

Fitur ini menyajikan hasil klasifikasi dan analisis model dalam bentuk visualisasi yang interaktif.

- **Prioritas:** Medium

3.4.2 Stimulus/Response Sequences

1. **Input:** Hasil klasifikasi dari model machine learning
2. **Proses:**
 - Menampilkan scatter plot 3D berdasarkan koordinat Right Ascension (α), Declination (δ), dan Redshift (z).
 - Menyediakan histogram dan distribusi spektral.
 - Menampilkan heatmap confusion matrix.
3. **Output:** Grafik interaktif yang membantu pengguna memahami hasil klasifikasi.

3.4.3 Functional Requirements

1. **REQ-10:** Sistem harus menyediakan scatter plot 3D yang menunjukkan distribusi objek astronomi.
2. **REQ-11:** Sistem harus menampilkan histogram dan distribusi spektral.
3. **REQ-12:** Sistem harus menampilkan heatmap confusion matrix untuk evaluasi model.

3.5 User Interaction & Export Feature

3.5.1 Description and Priority

Fitur ini memungkinkan pengguna untuk berinteraksi dengan hasil klasifikasi serta mengunduh data dan visualisasi.

- **Prioritas:** Medium

3.5.2 Stimulus/Response Sequences

1. **Input:** Hasil klasifikasi dan visualisasi
2. **Proses:**
 - Menyediakan filter interaktif untuk eksplorasi data.
 - Mengaktifkan fitur linked brushing untuk menyorot data terkait dalam visualisasi.
 - Memungkinkan pengguna mengunduh hasil analisis dalam berbagai format.
3. **Output:** Hasil klasifikasi yang dapat dieksplorasi dan diunduh.

3.5.3 Functional Requirements

1. **REQ-13:** Sistem harus menyediakan filter interaktif berdasarkan parameter tertentu (misalnya redshift).
2. **REQ-14:** Sistem harus memungkinkan linked brushing antar visualisasi.
3. **REQ-15:** Sistem harus memungkinkan pengguna mengunduh hasil analisis dalam format CSV, JSON, atau gambar (PNG/JPG).

4. Skenario Testing

Setiap test case pada aplikasi AstroClassify diimplementasikan sebagai fungsi terpisah yang menguji satu komponen atau fitur secara spesifik. Aplikasi ini dikembangkan dengan pendekatan gabungan antara Object-Oriented Programming (OOP) untuk struktur kelas dan Functional Programming (FP) untuk fungsi-fungsi murni yang dapat diuji secara modular. Fitur-fitur utama seperti upload data, preprocessing, prediksi model, dan visualisasi disusun secara sekuensial dalam alur penggunaan, namun pengujian dilakukan secara terpisah untuk memastikan keandalan masing-masing fungsi secara independen.

Pengujian mencakup berbagai skenario, termasuk validasi range input, penanganan missing value, evaluasi hasil prediksi model, dan respons sistem saat data tidak tersedia. Dengan pendekatan ini, proses testing tidak hanya memastikan fungsionalitas, tetapi juga stabilitas dan ketahanan sistem terhadap variasi input dan kondisi tak terduga.

4.1 Data Input – 7 Skenario Testing

No	Test Case	Input	Langkah Uji	Ekspetasi
1	Upload File CSV dari SDSS	File CSV dari SDSS	Klik Upload → Pilih file valid → Submit	Data tampil di pratinjau
2	Upload file kosong	File kosong	Klik Upload → Submit	Error: "File tidak mengandung data"
3	Input data manual valid	Atribut satuan	Klik Input Manual → Masukkan alpha, delta, magnitudo → Submit	Data muncul di tabel input
4	Upload file dengan jumlah kolom kurang	Upload file dengan kolom tidak lengkap	Kolom u/g/r/i/z hilang	Sistem menolak dan tampilkan error
5	Input data satuan dengan format yang tidak sesuai	Input berisi teks, bukan angka	Nilai non-numerik dimasukkan	Error: format tidak valid

6	Input data numerik di luar range yang ditentukan (range: 0-360.000)	Atribut numerik	Klik Input Manual → Masukkan alpha, delta, magnitudo dengan value 1M → Submit	Muncul pesan error, “input berada di luar batas/tidak valid”
7	Memasukkan file dengan missing value di beberapa kolom	File csv	Klik Upload → Pilih file valid → Submit	Data akan di preprocess secara otomatis (nilai null terisi)

4.2 Model Selection – 5 Skenario Testing

No	Test Case	Input	Langkah Uji	Ekspetasi
1	Pilih model XGBoost	Dataset preprocessed	Pilih XGBoost → Klik Train	Muncul hasil evaluasi
2	Tanpa memilih model	Dataset preprocessed	Klik Train tanpa pilih model	Model terbaik digunakan secara default
3	Pilih model, tampilkan confusion matrix	Model pilihan	Pilih model “Random Forest” → Klik “Lihat Confusion Matrix”	Confusion Matrix ditampilkan sesuai model
4	Switch model	2 model pilihan	Pilih “Random Forest” → Klik “Lihat Confusion Matrix” → Ganti ke “XGboost”	Tampilan metrik berubah ke evaluasi milik XGboost, model aktif terganti

5	Validasi hasil prediksi	Dataset preprocessed	Pilih model bebas (“random forest”) → klik “prediksi” → bandingkan dengan data uji	Hasil prediksi sama seperti data uji
6	Dataset belum sepenuhnya di preprocessing	Dataset	Pilih model bebas (“random forest”) → klik “prediksi”	Tampil Error: “gagal prediksi: Data gagal di preprocessed”

4.3 Result Visualization – 6 Skenario Testing

No	Test Case	Input	Langkah Uji	Ekspetasi
1	Tampilkan scatter plot 3D	Hasil klasifikasi	Klik “3D Scatter”	Plot interaktif muncul
2	Zoom pada grafik	Grafik hasil	Scroll mouse / gesture pinch pada grafik	Grafik melakukan zoom in/out
3	Interaksi menggunakan linked brushing	Grafik terkait	Sorot elemen pada satu grafik	Elemen terkait di grafik lain tersorot
4	Interaksi dengan plot (hover)	Pointer Mouse	Arahkan kursor ke titik pada grafik	Tooltip muncul dengan detail data
5	Interaksi dengan grafik (Menggeser)	Grafik aktif	Hover/klik elemen	Tooltip atau highlight muncul
6	Ketika Data hasil klasifikasi Kosong	Data Hasil Klasifikasi Kosong	Klik “3D Scatter”	Error: “Tidak ada data untuk visualisasi”

7	Ketika Kolom untuk visualisasi (alpha, delta,dll) tidak ada	Hasil Klasifikasi tanpa kolom alpha, delta	Klik “3D Scatter”	Error: “Tidak ada Kolom ‘alpha’, ‘delta’. Silahkan cek kembali data anda”
---	---	--	-------------------	---

4.4 Export Result – 7 Skenario Testing

No	Test Case	Input	Langkah Uji	Ekpetasi
1	Unduh hasil klasifikasi sebagai CSV	Data hasil klasifikasi	Klik tombol Export CSV	File CSV berhasil diunduh dan berisi hasil klasifikasi dengan struktur benar
2	Unduh hasil klasifikasi sebagai XLSX	Data hasil klasifikasi	Klik tombol Export XLSX	File XLSX berhasil diunduh dan dapat dibuka di Excel
3	Unduh hasil klasifikasi sebagai JSON	Data xls klasifikasi	Klik tombol Export JSON	File JSON berhasil diunduh dengan format key-value sesuai
4	Unduh Hasil klasifikasi sebagai CSV, XLSX atau JSON ketika Data hasil klasifikasi kosong/tidak valid	Data hasil klasifikasi yang kosong/tidak valid	Klik tombol Export CSV, Export XLSX, atau Export JSON	Error: “Data hasil klasifikasi kosong/tidak valid untuk diunduh”

5	Unduh plot 2D hasil klasifikasi sebagai gambar (PNG)	Grafik hasil visualisasi	Klik tombol Unduh plot 2D	Gambar PNG berhasil diunduh dan sesuai tampilan grafik
6	Unduh plot 3D hasil klasifikasi sebagai gambar (PNG)	Grafik hasil visualisasi	Klik tombol Unduh plot 3D	Gambar PNG berhasil diunduh dan sesuai tampilan grafik
7	Unduh plot 2D atau 3D hasil klasifikasi sebagai gambar (PNG) ketika grafik hasil visualisasi kosong atau tidak tersedia	Grafik hasil visualisasi kosong/tidak tersedia	Klik tombol Unduh plot 2D atau Unduh plot 3D	Error: “Grafik visualisasi tidak tersedia, unduhan tidak dapat dilakukan”

5. Hasil Testing (Testing Results)

5.1 Tujuan Pengujian

Pengujian dilakukan untuk memastikan bahwa aplikasi AstroClassify berjalan sesuai dengan yang diharapkan, baik dari segi fungsionalitas, akurasi model klasifikasi, kestabilan sistem, maupun antarmuka pengguna. Secara umum, pengujian bertujuan untuk mengidentifikasi dan memperbaiki potensi kesalahan atau bug, serta memastikan bahwa setiap komponen aplikasi dapat berfungsi secara optimal dan terintegrasi dengan baik sebelum digunakan oleh pengguna akhir.

5.2 Metode Pengujian

Pengujian dilakukan dengan pendekatan pengujian berlapis yang terdiri dari:

1. Unit Testing

- Menguji fungsi-fungsi terkecil secara individual, seperti fungsi input data, praproses data, pemodelan, dan visualisasi.

2. Component Testing

- Menguji integrasi antar beberapa fungsi dalam satu komponen, misalnya antara modul klasifikasi dengan hasil visualisasi.

3. System Testing

- Melakukan pengujian terhadap sistem secara menyeluruh dengan metode pengujian tradisional (mencocokkan output dengan scenario test case), dari input data hingga output akhir yang ditampilkan ke pengguna.
- Dilakukan berdasarkan test scenario yang telah dirancang sebelumnya.
- Output aktual dibandingkan dengan expected output pada setiap skenario untuk menentukan apakah sistem lulus pengujian.

4. Evaluasi Akurasi Model

Selain pengujian sistem, dilakukan juga evaluasi performa model klasifikasi menggunakan metrik seperti:

- Accuracy
- Precision
- Recall
- F1-score

Evaluasi dilakukan pada dataset uji (test set) untuk memastikan performa generalisasi model.

5.3 Hasil Pengujian

5.3.1 Fitur Data Input

No	Test Case	Input	Langkah Uji	Ekspetasi	Status Testing
1	Upload File CSV dari SDSS	File CSV dari SDSS	Klik Upload → Pilih file valid → Submit	Data tampil di pratinjau	LULUS
2	Upload file kosong	File kosong	Klik Upload → Submit	Error: "File tidak mengandung data"	LULUS
3	Input data manual valid	Atribut satuan	Klik Input Manual → Masukkan alpha, delta, magnitudo → Submit	Data muncul di tabel input	LULUS
4	Upload file dengan jumlah kolom kurang	Upload file dengan	Kolom u/g/r/i/z hilang	Sistem menolak dan tampilkan error	LULUS

		kolom tidak lengkap			
5	Input data satuan dengan format yang tidak sesuai	Input berisi teks, bukan angka	Nilai non-numerik dimasukkan	Error: format tidak valid	LULUS
6	Input data numerik di luar range yang ditentukan (range: 0-360.000)	Atribut numerik	Klik Input Manual → Masukkan alpha, delta, magnitudo dengan value 1M → Submit	Muncul pesan error, “input berada di luar batas/tidak valid”	LULUS
7	Memasukkan file dengan missing value di beberapa kolom	File csv	Klik Upload → Pilih file valid → Submit	Data akan di preprocess secara otomatis (nilai null terisi)	LULUS

5.3.2 Fitur Model Selection

No	Test Case	Input	Langkah Uji	Ekspetasi	Status Pengujian
1	Pilih model XGBoost	Dataset preprocessed	Pilih XGBoost → Klik Predik	Muncul hasil evaluasi	LULUS
2	Tanpa memilih model	Dataset preprocessed	Klik Train tanpa pilih model	Model terbaik digunakan secara default	LULUS

3	Pilih model, tampilkan confusion matrix	Model pilihan	Pilih model “Random Forest” → Klik “Lihat Confusion Matrix”	Confusion Matrix ditampilkan sesuai model	LULUS
4	Switch model	2 model pilihan	Pilih “Random Forest” → Klik “Lihat Confusion Matrix” → Ganti ke “XGboost”	Tampilan metrik berubah ke evaluasi milik XGboost, model aktif terganti	LULUS
5	Validasi hasil prediksi	Dataset preprocessed	Pilih model bebas (“random forest”) → klik “prediksi” → bandingkan dengan data uji	Hasil prediksi sama seperti data uji	LULUS
6	Dataset belum sepenuhnya di preprocessing	Dataset	Pilih model bebas (“random forest”) → klik “prediksi”	Tampil Error: “gagal prediksi: Data gagal di preprocessed”	LULUS

5.3.3 Fitur Result Visualization

No	Test Case	Input	Langkah Uji	Ekspetasi	Status Testing
1	Tampilkan scatter plot 3D	Hasil klasifikasi	Klik “3D Scatter”	Plot interaktif muncul	LULUS
2	Zoom pada grafik	Grafik hasil	Scroll mouse / gesture pinch pada grafik	Grafik melakukan zoom in/out	LULUS

3	Interaksi menggunakan linked brushing	Grafik terkait	Sorot elemen pada satu grafik	Elemen terkait di grafik lain tersorot	GAGAL (LIBRARY TIDAK SUPPORT)
4	Interaksi dengan plot (hover)	Pointer Mouse	Arahkan kursor ke titik pada grafik	Tooltip muncul dengan detail data	LULUS
5	Interaksi dengan grafik (Menggeser)	Grafik aktif	Hover/klik elemen	Tooltip atau highlight muncul	LULUS
6	Ketika Data hasil klasifikasi Kosong	Data Hasil Klasifikasi Kosong	Klik “3D Scatter”	Error: “Tidak ada data untuk visualisasi”	LULUS
7	Ketika Kolom untuk visualisasi (alpha, delta,dll) tidak ada	Hasil Klasifikasi tanpa kolom alpha, delta	Klik “3D Scatter”	Error: “Tidak ada Kolom ‘alpha’, ‘delta’. Silahkan cek kembali data anda”	LULUS

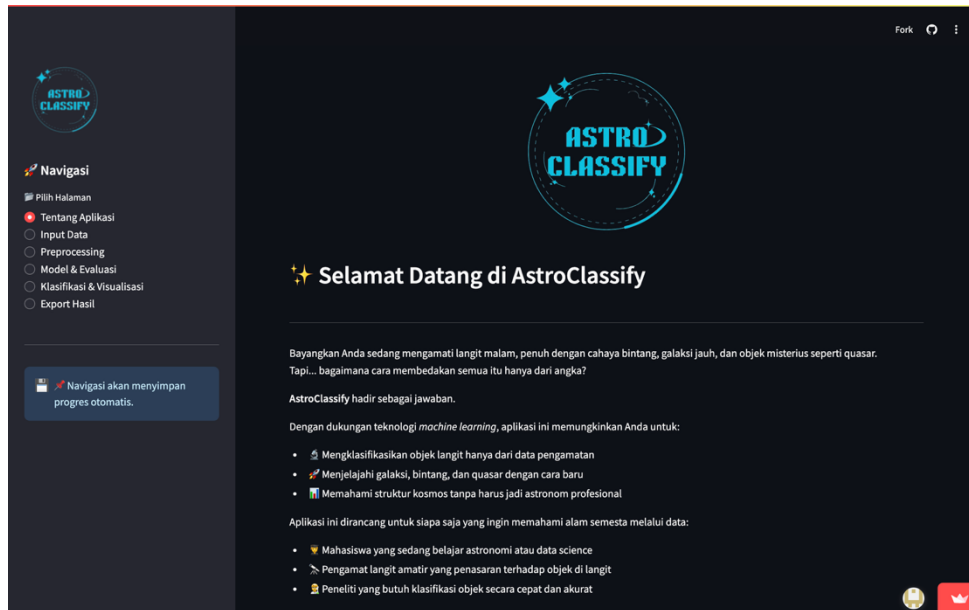
5.3.4 Fitur Export Result

No	Test Case	Input	Langkah Uji	Ekpetasi	Status Pengujian
1	Unduh hasil klasifikasi sebagai CSV	Data hasil klasifikasi	Klik tombol Export CSV	File CSV berhasil diunduh dan berisi hasil klasifikasi dengan struktur benar	LULUS

2	Unduh hasil klasifikasi sebagai XLSX	Data hasil klasifikasi	Klik tombol Export XLSX	File XLSX berhasil diunduh dan dapat dibuka di Excel	LULUS
3	Unduh hasil klasifikasi sebagai JSON	Data xls klasifikasi	Klik tombol Export JSON	File JSON berhasil diunduh dengan format key-value sesuai	LULUS
4	Unduh Hasil klasifikasi sebagai CSV, XLSX atau JSON ketika Data hasil klasifikasi kosong/tidak valid	Data hasil klasifikasi yang kosong/tidak valid	Klik tombol Export CSV, Export XLSX, atau Export JSON	Error: "Data hasil klasifikasi kosong/tidak valid untuk diunduh"	LULUS
5	Unduh plot 2D hasil klasifikasi sebagai gambar (PNG)	Grafik hasil visualisasi	Klik tombol Unduh plot 2D	Gambar PNG berhasil diunduh dan sesuai tampilan grafik	LULUS
6	Unduh plot 3D hasil klasifikasi sebagai gambar (PNG)	Grafik hasil visualisasi	Klik tombol Unduh plot 3D	Gambar PNG berhasil diunduh dan sesuai tampilan grafik	LULUS
7	Unduh plot 2D atau 3D hasil klasifikasi sebagai gambar (PNG) ketika grafik hasil	Grafik hasil visualisasi kosong/tidak tersedia	Klik tombol Unduh plot 2D atau Unduh plot 3D	Error: "Grafik visualisasi tidak tersedia, unduhan tidak dapat dilakukan"	LULUS

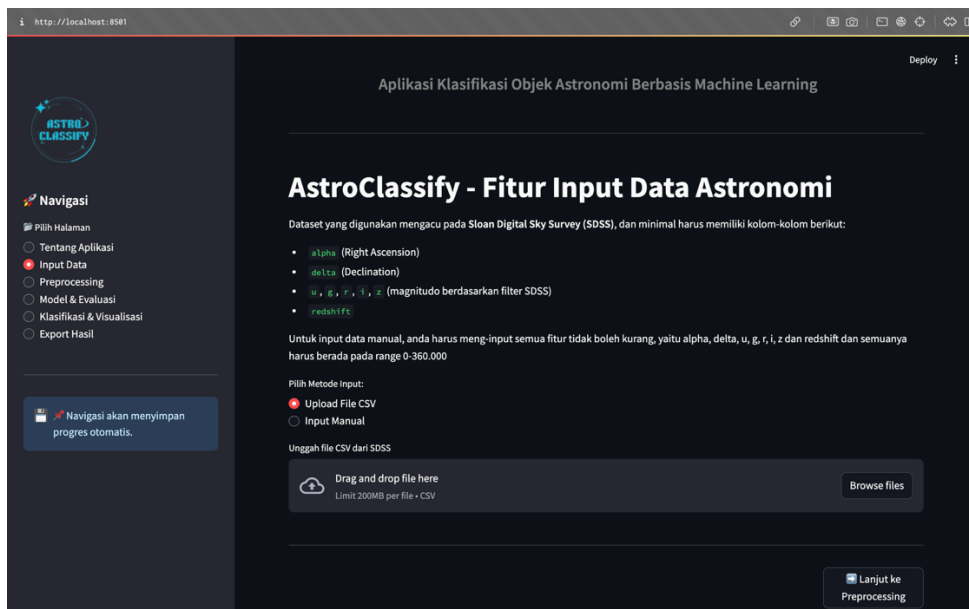
	visualisasi kosong atau tidak tersedia				
--	---	--	--	--	--

6. Interface Aplikasi

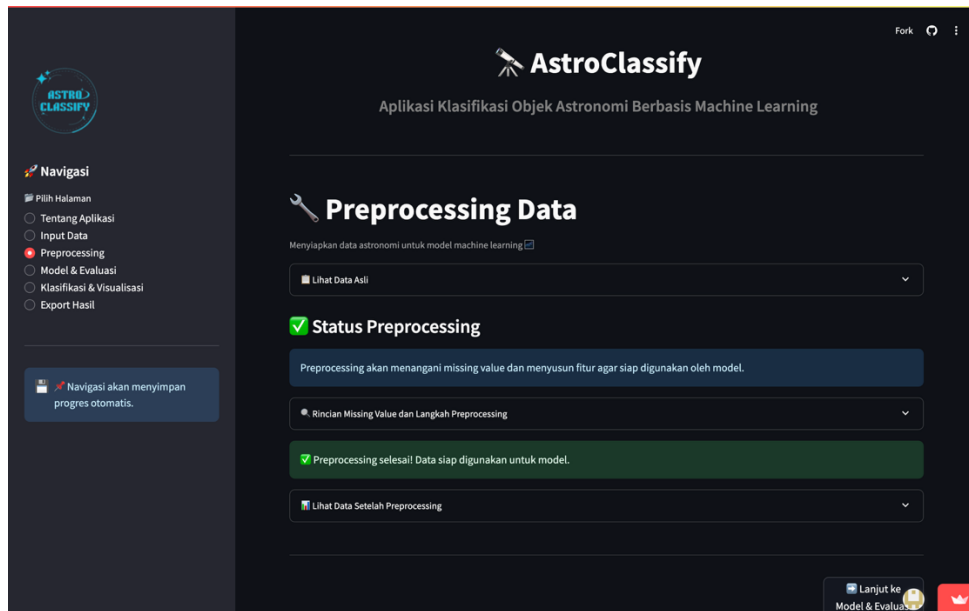


Tampilan awal aplikasi AstroClassify menyambut pengguna dengan antarmuka yang informatif dan visual yang menarik, memperkenalkan tujuan utama aplikasi dalam membantu proses klasifikasi objek langit menggunakan teknologi machine learning. Halaman ini menjelaskan manfaat utama aplikasi, seperti mengidentifikasi objek langit dari data observasi, mengenali jenis objek baru, serta memahami struktur kosmos secara efisien. Ditujukan untuk mahasiswa, pengamat langit, dan peneliti, halaman ini memberikan pengantar yang jelas tentang peran dan potensi AstroClassify dalam eksplorasi data astronomi.

6.1 Data Input

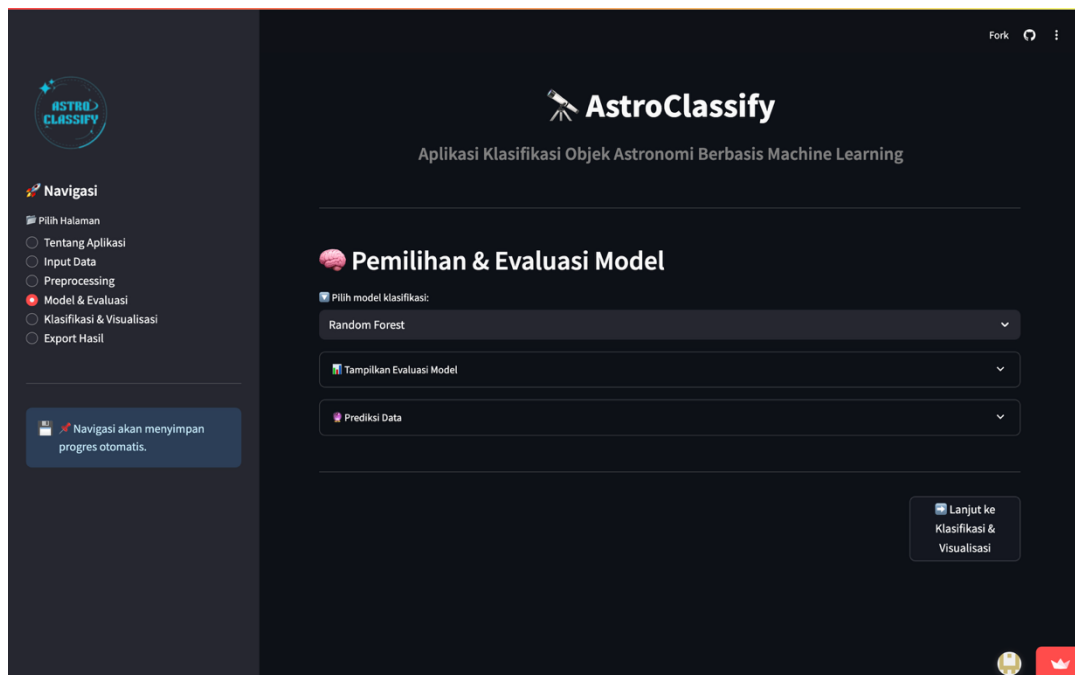


Halaman Input Data Astronomi pada aplikasi AstroClassify dirancang untuk memudahkan pengguna dalam memasukkan dataset berbasis standar Sloan Digital Sky Survey (SDSS). Pengguna dapat memilih metode input melalui unggahan file CSV atau input manual, dengan ketentuan kolom wajib seperti alpha, delta, u, g, r, i, z, dan redshift.



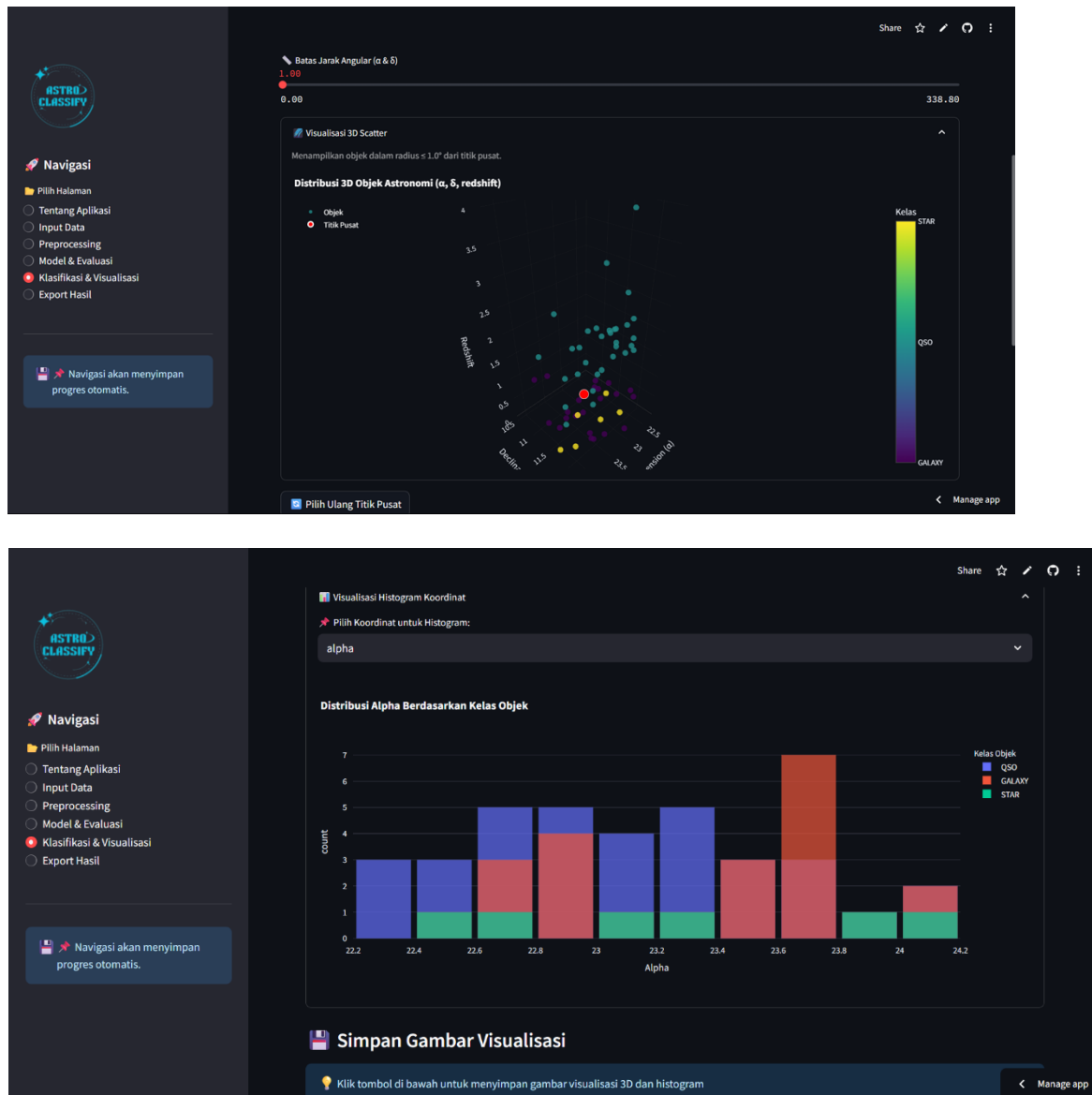
Fitur Preprocessing Data pada aplikasi AstroClassify berfungsi untuk menyiapkan data astronomi agar layak digunakan oleh model machine learning, khususnya dengan menangani nilai hilang (missing values) dan menyusun ulang fitur yang diperlukan. Pengguna dapat melihat data mentah sebelum dan sesudah praproses, serta mendapatkan status progres secara interaktif, yang menunjukkan bahwa data telah berhasil diproses dan siap untuk tahap pemodelan.

6.2 Model Selection



Fitur Pemilihan & Evaluasi Model pada aplikasi AstroClassify memungkinkan pengguna untuk memilih algoritma klasifikasi, seperti Random Forest, dan melakukan evaluasi performa model secara langsung. Pengguna dapat melihat hasil evaluasi seperti akurasi model serta menjalankan prediksi terhadap data yang telah diproses, sebelum melanjutkan ke tahap visualisasi hasil klasifikasi.

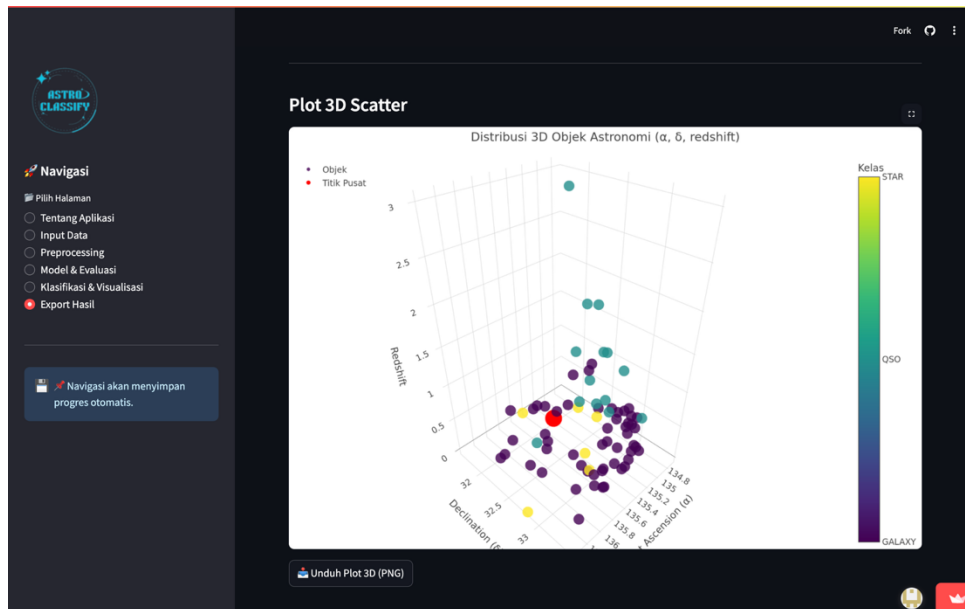
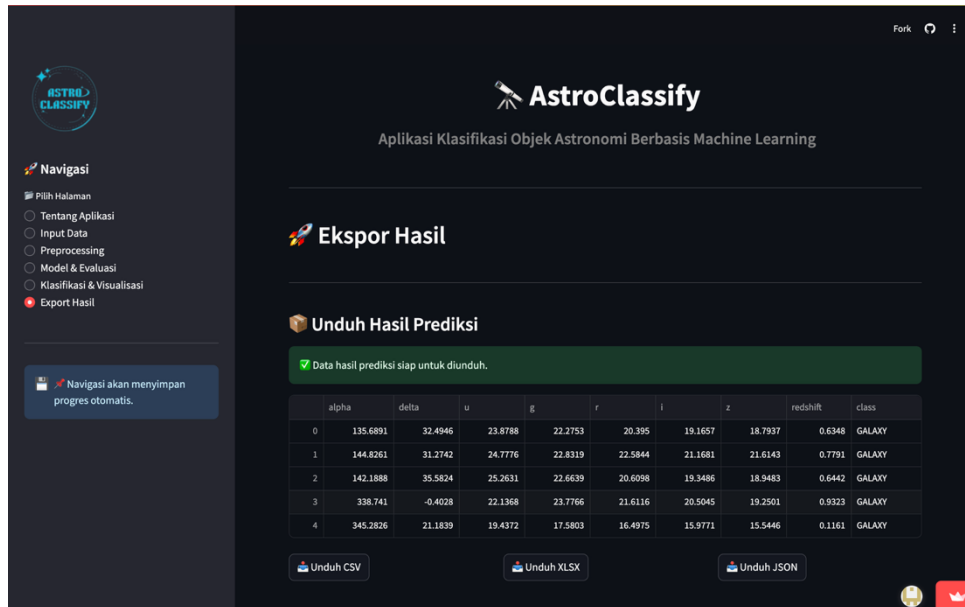
6.3 Result Visualitation



Fitur Klasifikasi & Visualisasi pada AstroClassify menyajikan hasil prediksi dalam bentuk grafik 3D scatter yang memvisualisasikan distribusi objek astronomi berdasarkan atribut α (right ascension), δ (declination), redshift, u, g, r, i, z. Pengguna dapat mengatur batas jarak angular untuk menampilkan objek dalam radius tertentu dari titik pusat, serta membedakan kelas objek (seperti GALAXY, STAR, dan QSO) melalui skala warna, sehingga memudahkan analisis spasial dan klasifikasi objek langit secara intuitif.

Selain itu, tersedia pula visualisasi histogram yang menampilkan distribusi nilai Alpha dan Beta. Pengguna dapat memilih salah satu dari kedua parameter ini melalui menu dropdown. Setiap histogram disusun berdasarkan kelas objek, sehingga memudahkan dalam mengamati pola distribusi yang khas pada masing-masing kategori.

6.4 Export Result



Fitur Ekspor Hasil pada AstroClassify memungkinkan pengguna untuk mengunduh hasil prediksi klasifikasi objek astronomi dalam berbagai format file seperti CSV, XLSX, dan JSON, sehingga memudahkan dokumentasi dan analisis lanjutan. Selain data tabular, pengguna juga dapat mengunduh visualisasi grafik 3D scatter dalam format PNG, yang menampilkan distribusi objek berdasarkan atribut α , δ , dan redshift serta klasifikasinya. Fitur ini memberikan fleksibilitas tinggi dalam penyimpanan dan pelaporan hasil analisis.

7. Praktik Git & Github

Selama proses pengembangan proyek AstroClassify, kami menerapkan Git dan GitHub sebagai alat utama dalam mengelola versi dan kolaborasi kode. Praktik ini menjadi pengalaman langsung dalam bekerja secara tim, membagi tugas melalui branch masing-masing, serta menyatukan kontribusi melalui pull request.

Link repository Github proyek ini bisa diakses pada link berikut:

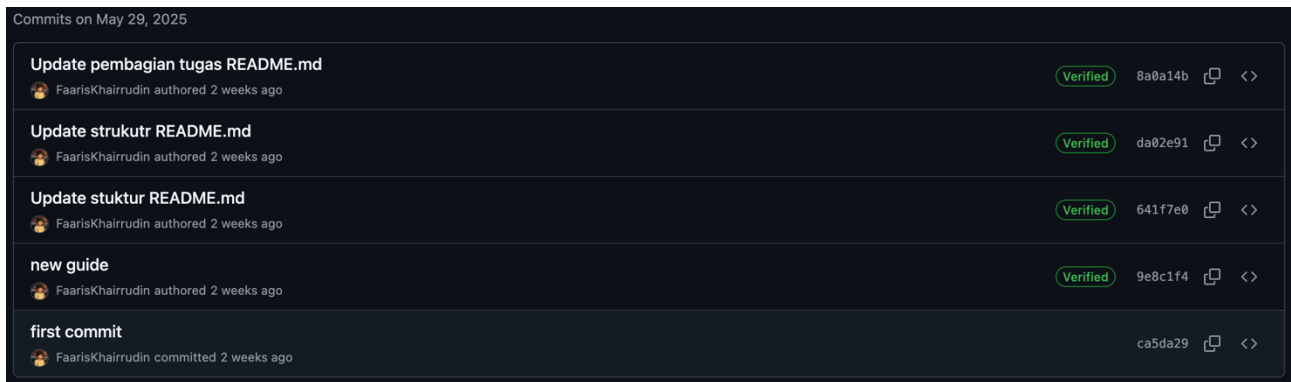
<https://github.com/FaarisKhairrudin/AstroClassify>

Terdapat total 69 commits pada branch utama selama masa pengerjaan proyek (dapat dilihat lengkap pada repository bagian history commits)

Berikut adalah bukti pengerjaan praktik Git & Github:

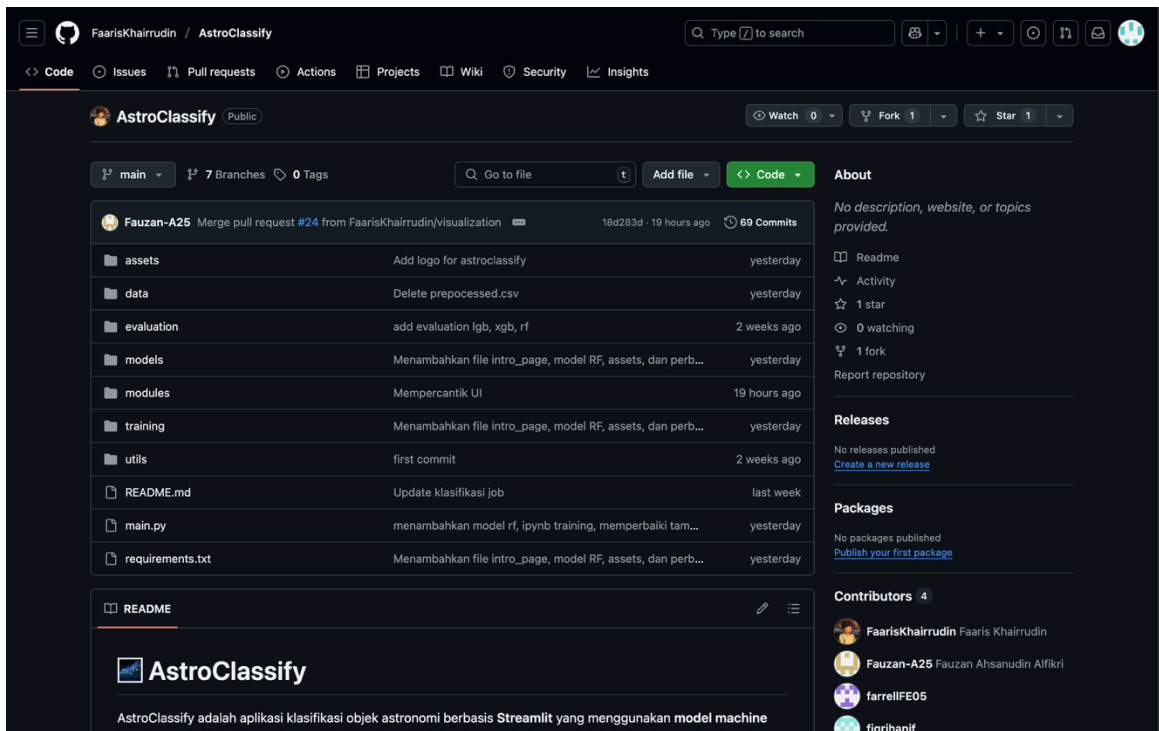
1. Inisialisasi Proyek & First Commit (menambahkan README.md)

Proyek *AstroClassify* diawali dengan proses inisialisasi Git lokal menggunakan perintah git init pada folder proyek. Setelah itu, kami menambahkan file awal README.md untuk memberikan deskripsi singkat proyek, kemudian dilakukan commit awal menggunakan git add dan git commit.



2. Membuat Repository (Menghubungkan dengan github)

Repository GitHub dibuat dengan nama AstroClassify, lalu Git lokal dikaitkan ke remote repository menggunakan perintah git remote add origin. Setelah itu, commit awal di-push menggunakan git push -u origin main. Struktur awal direktori yang disiapkan terdiri dari beberapa folder kosong seperti modules/, training/, models/, dan utils/, yang nantinya akan diisi oleh setiap anggota sesuai dengan tugasnya masing-masing. Berikut struktur awal repositori kami:

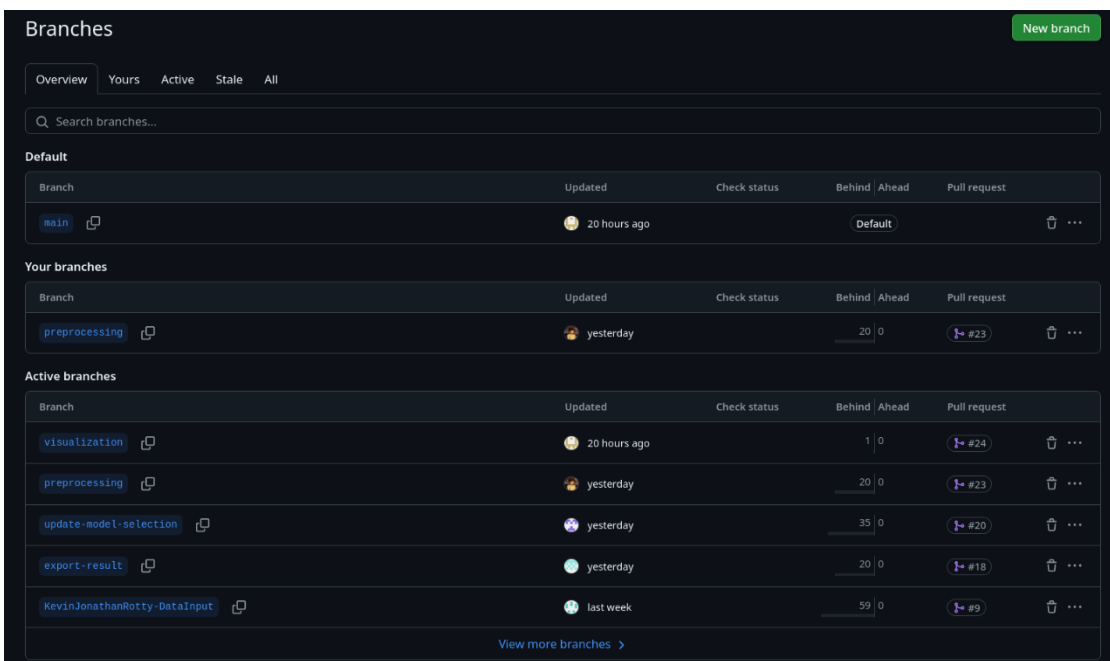


3. Kolaborasi Tim

Setiap anggota membuat branch pribadi dari main untuk mengembangkan fitur masing-masing. Berikut daftar branch dan anggotanya:

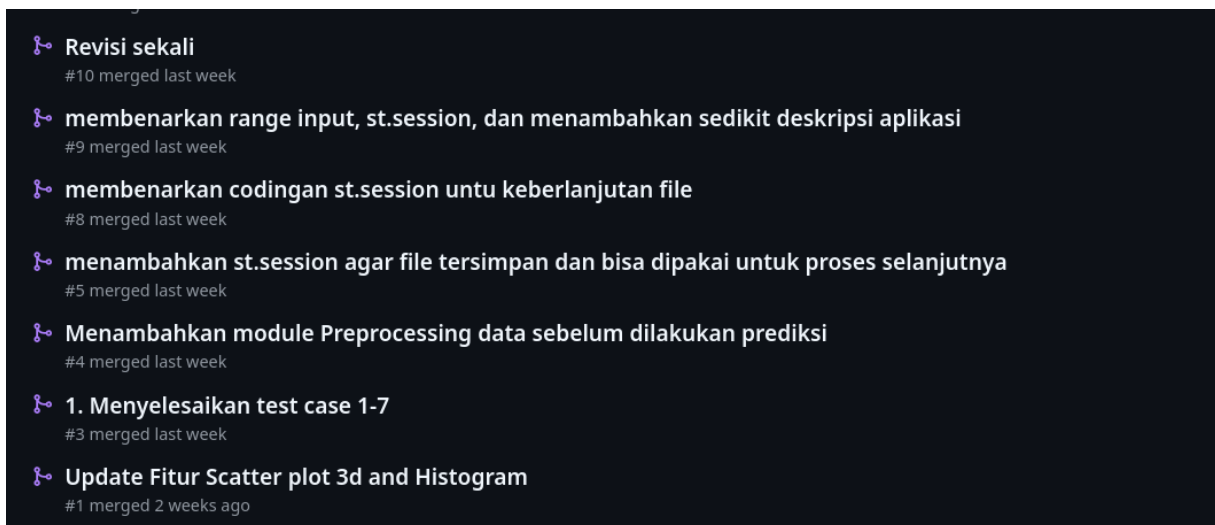
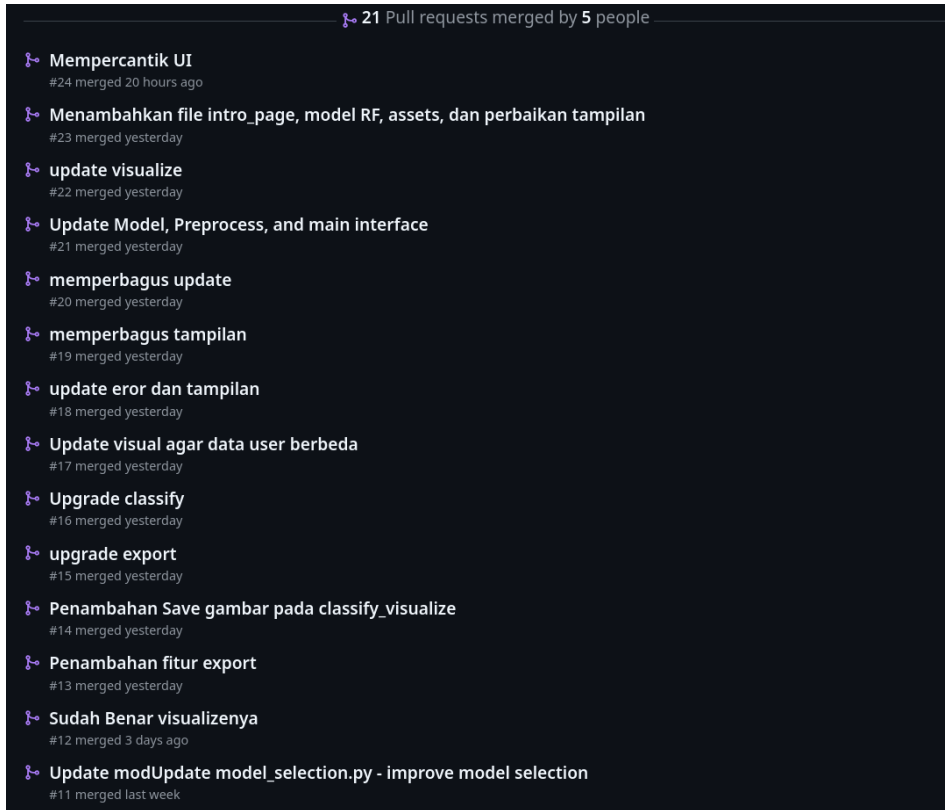
- KevinJonathanRotty-DataInput – Input data
- FaarisKhairudin-Preprocessing – Preprocessing data
- FarrelFaruqh-Model_Selection – Pemilihan model klasifikasi
- FauzanAhsanudin-Visualization – Visualisasi hasil klasifikasi
- MuhammadFikri-ExportResult – Ekspor hasil

Pada branch masing-masing, setiap anggota menambahkan kode, melakukan commit, dan push ke GitHub menggunakan perintah `git push origin nama_branch`.



4. Review dan Merge

Setelah menyelesaikan pekerjaan di masing-masing branch, anggota tim membuat Pull Request (PR) untuk digabungkan ke branch main. PR ini direview oleh anggota lain, dan setelah disetujui dilakukan proses merge. Setiap merge berhasil akan diikuti dengan penghapusan branch jika sudah tidak digunakan lagi.



May 13, 2025 – June 13, 2025

Period: 1 month ▾

Overview

21 Active pull requests

0 Active issues

21

Merged pull requests

0

Open pull requests

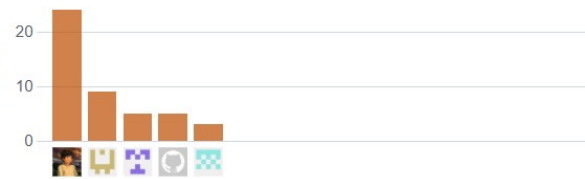
0

Closed issues

0

New issues

Excluding merges, **5 authors** have pushed **46 commits** to main and **46 commits** to all branches. On main, **0 files** have changed and there have been **0 additions** and **0 deletions**.



21 Pull requests merged by 5 people