# Ringkasan Dokumentasi Backend - Sistem Prediksi Prestasi Siswa

## 📋 Daftar Isi

1. Arsitektur Sistem
2. File Utama
3. Models
4. Routers/Endpoints
5. Fitur Utama
6. Teknologi yang Digunakan

## 🏗️ Arsitektur Sistem

### Struktur Direktori Backend

backend/  
├── main.py # Aplikasi FastAPI utama  
├── database.py # Konfigurasi database & models  
├── schemas.py # Pydantic schemas untuk validasi  
├── requirements.txt # Dependencies Python  
├── models/  
│ ├── user.py # Model User untuk authentication  
│ └── c45\_model.py # Implementasi algoritma C4.5  
└── routes/  
 ├── auth\_router.py # Authentication endpoints  
 ├── siswa\_router.py # CRUD siswa + Excel import/export  
 ├── nilai\_router.py # CRUD nilai raport  
 ├── presensi\_router.py # CRUD presensi siswa  
 ├── penghasilan\_router.py # CRUD penghasilan orang tua  
 └── prediksi\_router.py # Machine learning predictions

## 📁 File Utama

### 1. main.py - Aplikasi FastAPI Utama

**Fungsi Utama:**

* Inisialisasi aplikasi FastAPI dengan metadata
* Konfigurasi CORS untuk cross-origin requests
* Registrasi semua router dengan prefix dan tags
* Endpoint root dan health check
* Database initialization saat startup

**Endpoints:**

* GET / - Welcome message
* GET /health - Health check
* GET /api/decision\_tree - Visualisasi pohon keputusan

### 2. database.py - Konfigurasi Database

**Komponen Utama:**

* **Engine Setup**: PostgreSQL connection dengan SQLAlchemy
* **Session Management**: SessionLocal factory untuk database operations
* **Models Definition**: 5 model utama (Siswa, NilaiRaport, PenghasilanOrtu, Presensi, Prestasi)
* **Relationships**: One-to-many relationships antar tabel
* **Utility Functions**: get\_db() dan init\_db()

**Models:**

* **Siswa**: Data siswa (nama, NIS, kelas, dll)
* **NilaiRaport**: Nilai 11 mata pelajaran + rata-rata
* **PenghasilanOrtu**: Data ekonomi keluarga
* **Presensi**: Data kehadiran siswa
* **Prestasi**: Hasil prediksi prestasi

### 3. schemas.py - Pydantic Schemas

**Fungsi:**

* Validasi input/output API
* Type hints untuk semua endpoints
* Base, Create, Update, Response schemas untuk setiap model
* Field validation dengan Pydantic validators

## 🤖 Models

### 1. models/user.py - User Authentication

Python

class User(Base):  
 id: int (Primary Key)  
 username: str (Unique)  
 hashed\_password: str

### 2. models/c45\_model.py - Machine Learning Model

**Fitur Utama:**

* **Algoritma C4.5**: Decision Tree dengan entropy criterion
* **Data Preparation**: Mengambil data dari database untuk training
* **Training**: Split data, encoding categorical variables
* **Prediction**: Prediksi dengan confidence score
* **Visualization**: Generate pohon keputusan dalam format PNG
* **Rule Extraction**: Extract decision rules dalam format readable

**Fitur Machine Learning:**

* 3 fitur utama: rata\_rata, kategori\_penghasilan, kategori\_kehadiran
* Label otomatis berdasarkan scoring system
* Feature importance untuk explainability
* Model evaluation dengan accuracy dan classification report

## 🛣️ Routers/Endpoints

### 1. auth\_router.py - Authentication

**Endpoints:**

* POST /api/auth/token - Login dan dapatkan JWT token
* POST /api/auth/register - Registrasi user baru
* GET /api/auth/users - List semua users
* DELETE /api/auth/users/{user\_id} - Hapus user
* POST /api/auth/create-dummy-users - Buat dummy users

**Fitur Keamanan:**

* JWT token dengan expiration (2 jam)
* Password hashing dengan bcrypt
* OAuth2 authentication scheme
* Username validation (alphanumeric, min 3 karakter)

### 2. siswa\_router.py - Manajemen Data Siswa

**Endpoints:**

* POST /api/siswa/ - Tambah siswa baru
* GET /api/siswa/ - List siswa dengan pagination & search
* GET /api/siswa/{siswa\_id} - Detail siswa
* PUT /api/siswa/{siswa\_id} - Update siswa
* DELETE /api/siswa/{siswa\_id} - Hapus siswa
* POST /api/siswa/upload/excel - Import dari Excel
* GET /api/siswa/export/excel - Export ke Excel

**Fitur Khusus:**

* Excel import/export dengan pandas
* Search multi-field (nama, NIS, kelas)
* Validasi NIS unique
* File validation untuk Excel format

### 3. nilai\_router.py - Manajemen Nilai Raport

**Endpoints:**

* POST /api/nilai/ - Tambah nilai raport
* GET /api/nilai/ - List nilai dengan filter siswa
* GET /api/nilai/{nilai\_id} - Detail nilai
* PUT /api/nilai/{nilai\_id} - Update nilai
* DELETE /api/nilai/{nilai\_id} - Hapus nilai

**Fitur Khusus:**

* Auto-calculate rata-rata dari 11 mata pelajaran
* Validasi duplikasi per siswa/semester/tahun
* Recalculate rata-rata saat update

### 4. prediksi\_router.py - Machine Learning Predictions

**Endpoints:**

* POST /api/prediksi/train - Latih model C4.5
* POST /api/prediksi/ - Prediksi prestasi siswa
* GET /api/prediksi/rules - Dapatkan decision rules
* GET /api/prediksi/visualization - Visualisasi pohon keputusan
* POST /api/prediksi/generate-dummy-data - Generate data dummy
* GET /api/prediksi/model-info - Informasi model
* POST /api/prediksi/batch - Prediksi batch multiple siswa

**Fitur Machine Learning:**

* Auto-training jika model belum dilatih
* Save hasil prediksi ke database
* Feature importance untuk explainability
* Confidence score untuk setiap prediksi
* Dummy data generation untuk testing

## ✨ Fitur Utama

### 1. Machine Learning dengan C4.5

* Implementasi algoritma C4.5 (Decision Tree dengan entropy)
* 3 fitur prediksi: nilai rata-rata, kategori penghasilan, kategori kehadiran
* Auto-labeling berdasarkan scoring system
* Visualisasi pohon keputusan
* Rule extraction untuk interpretability

### 2. Excel Import/Export

* Import data siswa dari file Excel (.xlsx/.xls)
* Export data siswa ke Excel dengan formatting
* Validasi kolom dan data integrity
* Batch processing dengan error handling

### 3. Authentication & Authorization

* JWT-based authentication
* Password hashing dengan bcrypt
* Protected endpoints dengan dependency injection
* User management (CRUD operations)

### 4. Data Management

* CRUD operations untuk semua entitas
* Pagination dan search functionality
* Data validation dengan Pydantic
* Foreign key relationships dan data integrity

### 5. API Documentation

* Auto-generated OpenAPI/Swagger documentation
* Comprehensive error handling
* Structured response format
* Type hints dan validation

## 🛠️ Teknologi yang Digunakan

### Backend Framework

* **FastAPI**: Modern Python web framework
* **Uvicorn**: ASGI server untuk production
* **SQLAlchemy**: ORM untuk database operations
* **PostgreSQL**: Database utama

### Machine Learning

* **scikit-learn**: Decision Tree implementation
* **pandas**: Data manipulation dan analysis
* **numpy**: Numerical computing
* **graphviz**: Decision tree visualization

### Authentication & Security

* **python-jose**: JWT token handling
* **passlib**: Password hashing dengan bcrypt
* **python-multipart**: File upload support

### Data Processing

* **openpyxl**: Excel file processing
* **python-multipart**: File upload handling

### Development Tools

* **pydantic**: Data validation dan serialization
* **python-dotenv**: Environment variables management

## 📊 Statistik Sistem

### Total Functions: 50+ functions

* **Main App**: 4 functions
* **Database**: 8 functions
* **Authentication**: 7 functions
* **Siswa Management**: 8 functions
* **Nilai Management**: 5 functions
* **Prediksi ML**: 10+ functions
* **Utility Functions**: 8+ functions

### API Endpoints: 25+ endpoints

* Authentication: 5 endpoints
* Siswa: 7 endpoints
* Nilai: 5 endpoints
* Prediksi: 8+ endpoints

### Database Tables: 5 tables

* users, siswa, nilai\_raport, penghasilan\_ortu, presensi, prestasi

## 🎯 Kesimpulan

Sistem Prediksi Prestasi Siswa ini merupakan aplikasi web modern yang mengintegrasikan:

1. **RESTful API** dengan FastAPI untuk backend yang robust
2. **Machine Learning** dengan algoritma C4.5 untuk prediksi prestasi
3. **Database Management** dengan PostgreSQL dan SQLAlchemy ORM
4. **Authentication** dengan JWT untuk keamanan
5. **Excel Integration** untuk import/export data
6. **Comprehensive Documentation** untuk maintenance dan development

Sistem ini dirancang untuk membantu institusi pendidikan dalam memprediksi prestasi siswa berdasarkan data akademik, ekonomi keluarga, dan kehadiran siswa menggunakan pendekatan machine learning yang dapat diinterpretasi.