

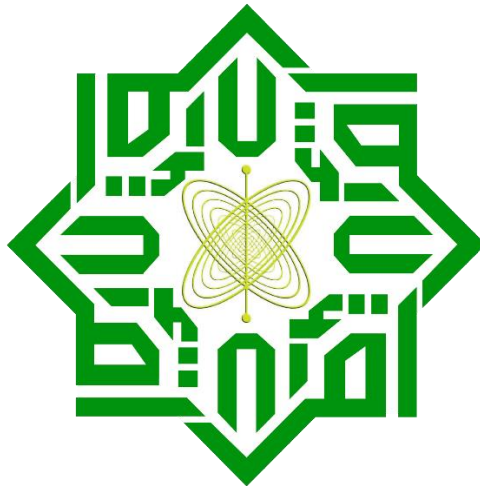
**PENGEMBANGAN *REST-API MUROJA'AH* JUZ 30  
BERBASIS *HONO FRAMEWORK***

**LAPORAN KERJA PRAKTIK**

Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik (S.T.)  
Program Studi Teknik Informatika

Oleh

**M. FARHAN AULIA PRATAMA**  
**NIM. 12250113521**



**UIN SUSKA RIAU**

**TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU  
PEKANBARU  
2025**

# KATA PENGANTAR



*Assalamu'alaikum wa rahmatullahi wa barakatuh.*

Segala puji dan syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala atas segala limpahan rahmat, nikmat, dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Kerja Praktik dengan judul “Pengembangan *REST-API Muroja'ah Juz 30* Berbasis *Framework Hono Typescript*” ini dengan baik dan tepat waktu. Shalawat serta salam senantiasa tercurah kepada junjungan Nabi Muhammad Shalallahu 'Alaihi Wasallam, yang telah membawa umat manusia menuju jalan kebenaran dan ilmu pengetahuan.

Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah Kerja Praktik, serta mendapatkan gelar Sarjana Teknik pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, UIN Suska Riau. Dalam pelaksanaan kerja praktik ini, penulis mengembangkan *REST-API* untuk mencatat kegiatan *muroja'ah Juz 30* sebagai bagian dari proses digitalisasi persyaratan akademik di lingkungan jurusan Teknik Informatika, yang mana sebelumnya proses tersebut masih dilakukan secara manual menggunakan formulir kertas.

Penulis menyadari bahwa tanpa bantuan, dukungan, serta doa dari berbagai pihak, laporan ini tidak akan terselesaikan dengan baik. Oleh karena itu, penulis menyampaikan rasa terima kasih dan penghargaan yang setulus-tulusnya kepada:

1. Bapak Iwan Iskandar, S.T., M.T., selaku Ketua Jurusan Teknik Informatika UIN Sultan Syarif Kasim Riau.
2. Ibu Liza Afriyanti, M.Kom., selaku Koordinator Kerja Praktik Jurusan Teknik Informatika.
3. Bapak Pizaini, S.T., M.Kom., selaku Dosen Pembimbing Lapangan yang telah banyak memberikan arahan dan masukan.
4. Kedua orang tua serta keluarga tercinta yang selalu memberikan doa, motivasi, dan semangat.
5. Seluruh tim *Riau Devops 2022* yang telah memberikan tenaga, waktu, dan pikirannya dalam membersamai pelaksanaan kerja praktik ini.
6. Seluruh rekan mahasiswa Teknik Informatika angkatan 2022 yang telah memberi dukungan selama pelaksanaan kerja praktik.
7. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung, yang tidak dapat penulis sebutkan satu per satu.

Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca maupun pengembangan sistem selanjutnya.

Pekanbaru, 2 Juni 2025

Penulis

# DAFTAR ISI

KATA PENGANTAR .....	2
DAFTAR ISI.....	3
DAFTAR GAMBAR .....	4
DAFTAR TABEL.....	5
BAB I PENDAHULUAN.....	6
1.1 Latar Belakang .....	6
1.2 Rumusan Masalah .....	6
1.3 Manfaat dan Tujuan .....	7
BAB II ANALISIS DAN PERANCANGAN.....	8
2.1 Analisis Kebutuhan .....	8
2.2 Perancangan Arsitektur .....	8
2.3 Perancangan <i>Database</i> .....	10
2.4 Perancangan <i>Dockerfile</i> .....	11
2.5 Perancangan <i>Endpoint API</i> .....	12
BAB III IMPLEMENTASI DAN PENGUJIAN .....	13
3.1 Mekanisme Penggunaan <i>API Muroja'ah</i> .....	13
3.2 Dokumentasi <i>API Muroja'ah</i> .....	13

## DAFTAR GAMBAR

Gambar 1: Alur Arsitektur Umum .....	9
Gambar 2: ERD Database Muroja'ah .....	10
Gambar 3: schema.prisma DB_Muroja'ah .....	11
Gambar 4: entripoint.sh API Muroja'ah .....	11
Gambar 5: Dockerfile API Muroja'ah .....	11

## DAFTAR TABEL

Tabel 1: Rancangan Umum Endpoint API Muroja'ah .....	12
Tabel 2: Authentication Mechanism .....	13
Tabel 3: Endpoint /mahasiswa/kartu-murojaah-digital/:id.....	13
Tabel 4: Basic Flow endpoint /mahasiswa/kartu/murojaah-digital/:id.....	13
Tabel 5: Exception Flow 1 endpoint mahasiswa/kartu-murojaah-digital/:id .....	15
Tabel 6: endpoint /mahasiswa/check-murojaah/:nim?syarat= .....	16
Tabel 7: Basic Flow /mahasiswa/check-murojaah/:nim?syarat= .....	16
Tabel 8: Exception Flow 1 /mahasiswa/check-murojaah/:nim?syarat= .....	17
Tabel 9: Exception Flow 2 /mahasiswa/check-murojaah/:nim?syarat= .....	17
Tabel 10: endpoint /dosen/pa-saya .....	17
Tabel 11: Basic Flow /dosen/pa-saya.....	17
Tabel 12: Exception Flow 1 /dosen/pa-saya .....	18
Tabel 13: endpoint POST /mahasiswa/setoran/:nim .....	19
Tabel 14: Basic Flow endpoint POST /mahasiswa/setoran/:nim.....	19
Tabel 15: Exception Flow 1 endpoint POST /mahasiswa/setoran/:nim .....	19
Tabel 16: Exception Flow 2 endpoint POST /mahasiswa/setoran/:nim .....	20
Tabel 17: Exception Flow 3 endpoint POST /mahasiswa/setoran/:nim .....	20
Tabel 18: endpoint GET /mahasiswa/setoran/:nim .....	21
Tabel 19: Basic Flow endpoint GET /mahasiswa/setoran/:nim.....	21
Tabel 20: Exception Flow 1 endpoint GET /mahasiswa/setoran/:nim .....	23
Tabel 21: Exception Flow 2 endpoint POST /mahasiswa/setoran/:nim .....	23

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Penyelesaian *muroja'ah Juz 30 Al-Qur'an* merupakan salah satu prasyarat akademik yang berlaku di Universitas Islam Negeri Sultan Syarif Kasim Riau. Secara khusus di Program Studi Teknik Informatika, pemenuhan kewajiban *muroja'ah Juz 30 Al-Qur'an* ini diimplementasikan secara bertahap dan menjadi syarat administratif untuk berbagai tahapan akademik esensial lainnya, diantaranya sebagai salah satu syarat kelengkapan administrasi pengajuan kerja praktik, pelaksanaan seminar kerja praktik, proses pengajuan judul tugas akhir, seminar proposal, dan terakhir digunakan untuk salah satu prasyarat sidang tugas akhir.

Saat ini, proses pencatatan dan validasi *muroja'ah Juz 30* di lingkungan Program Studi Teknik Informatika UIN Suska Riau masih dilakukan secara konvensional. Mahasiswa yang telah menyelesaikan *muroja'ah* per surat kepada dosen Pembimbing Akademik (PA), akan mendapatkan paraf pada lembar kertas sesuai dengan *template* yang telah disediakan oleh prodi, kertas ini disebut dengan “Kartu Setoran Hafalan Juz 30”. Kertas ini kemudian disimpan oleh mahasiswa dan seringkali perlu difoto sebagai bukti untuk dilampirkan dalam pengurusan berbagai persyaratan akademik.

Sistem manual berbasis kertas fisik ini memiliki beberapa kendala. Pertama, proses ini rentan terhadap risiko kerusakan atau kehilangan, karena wujud kertas ini berupa dokumen fisik, ini tentunya dapat menjadi hambatan tak terduga yang akan dialami oleh mahasiswa saat sedang memenuhi persyaratan akademiknya. Kedua, rekapitulasi dan pelacakan progres *muroja'ah* mahasiswa oleh dosen maupun pihak program studi menjadi kurang efisien dan bahkan hampir tidak bisa dilakukan. Ketiga, proses ini kurang mendukung transparansi. Oleh karena itu, digitalisasi proses pencatatan *muroja'ah Juz 30* menjadi sebuah kebutuhan untuk meningkatkan efisiensi, akurasi, dan kemudahan akses data, sejalan dengan upaya modernisasi layanan akademik di TIF UIN Suska Riau. Proyek ini bertujuan untuk mengembangkan sebuah *REST-API* yang akan menangani logika bisnis dan pengelolaan data untuk sistem pencatatan *muroja'ah Juz 30* secara digital.

### **1.2 Rumusan Masalah**

Permasalahan utama terletak pada proses pencatatan *muroja'ah Juz 30* yang masih konvensional menggunakan kertas fisik. Sistem manual ini menimbulkan kendala signifikan, yaitu proses menjadi tidak efisien, data rentan hilang atau rusak, serta menyulitkan dosen dalam melakukan rekapitulasi dan pemantauan progres mahasiswa secara akurat. Proses konvensional ini juga menyulitkan pihak program studi khususnya koordinator kerja praktik dan koordinator tugas akhir dalam memvalidasi keabsahan progres *muroja'ah* mahasiswa sebagai salah satu syarat administratif yang wajib dipenuhi untuk melaksanakan rangkaian proses kerja praktik atau tugas akhir. Oleh karena itu, kerja praktik ini berfokus pada bagaimana merancang sebuah *REST-API* yang menciptakan sistem pencatatan digital *muroja'ah Juz 30* untuk mengoptimalkan salah satu layanan akademik pada prodi TIF UIN Suska Riau yang terpusat, aman, dan efisien.

### 1.3 Manfaat dan Tujuan

Tujuan utama dari pelaksanaan kerja praktik ini adalah untuk memenuhi syarat pada mata kuliah wajib Program Studi Teknik Informatika dan sebagai sarana untuk mengimplementasikan ilmu yang telah dipelajari di perkuliahan maupun secara mandiri. Untuk itu, sesuai masalah yang telah dirumuskan, penerapan ilmu tersebut akan ditujukan dalam merancang dan membangun sebuah *REST-API* sebagai fondasi sistem pencatatan muroja'ah digital. Dengan menciptakan sistem pengelolaan data yang terpusat dan aman, proyek ini secara langsung memberikan manfaat berupa kemudahan dan keamanan data bagi mahasiswa, sekaligus menyediakan mekanisme validasi syarat administratif yang cepat dan akurat bagi Koordinator KP dan TA. Pada akhirnya, pengembangan ini bermanfaat untuk meningkatkan kualitas serta efisiensi layanan akademik di lingkungan Program Studi Teknik Informatika secara menyeluruh.

## BAB II

### ANALISIS DAN PERANCANGAN

#### 2.1 Analisis Kebutuhan

Berdasarkan wawancara dengan stakeholder (mahasiswa dan dosen), kebutuhan sistem diidentifikasi untuk mengatasi keterbatasan proses konvensional berbasis dokumen fisik saat ini. Kebutuhan yang teridentifikasi mencerminkan kebutuhan masing-masing stakeholder dan potensi pengembangan sistem di masa depan.

- a) Bagi Mahasiswa:
  - a. Sistem harus menyediakan halaman untuk melihat statistik, riwayat aktivitas, serta detail progres *muroja'ah Juz 30* miliknya secara transparan.
  - b. Sistem harus memungkinkan untuk mencetak kartu PDF *muroja'ah Juz 30* digital miliknya.
- b) Bagi Dosen Pembimbing Akademik (PA):
  - a. Sistem harus menampilkan daftar mahasiswa bimbingannya.
  - b. Sistem harus menyediakan halaman untuk melihat statistik, riwayat aktivitas, serta detail progres *muroja'ah Juz 30* milik mahasiswa bimbingannya secara transparan..
  - c. Sistem harus memiliki fitur untuk mencatat atau memberikan validasi (menyetujui) *muroja'ah* per surat dari setiap mahasiswa bimbingannya.
  - d. Sistem harus memiliki fitur untuk menghapus catatan atau membatalkan validasi *muroja'ah* per surat dari setiap mahasiswa bimbingannya.
- c) Bagi Koordinator KP & TA:
  - a. Sistem harus menyediakan cara yang cepat untuk memverifikasi keabsahan status penyelesaian *muroja'ah* seorang mahasiswa sebagai syarat pengajuan.
- d) Untuk Integrasi Antar-Sistem:
  - a. Sistem harus menyediakan *endpoint* khusus yang aman untuk dapat digunakan oleh layanan *backend* lain di masa depan (misalnya, sistem pendaftaran seminar KP atau Tugas Akhir). Endpoint ini berfungsi untuk memberikan respons validasi (contoh: *true/false*) secara otomatis mengenai status kelengkapan *muroja'ah* seorang mahasiswa per-syarat akademik, sehingga menghilangkan kebutuhan verifikasi manual pada sistem-sistem tersebut kedepannya.

#### 2.2 Perancangan Arsitektur

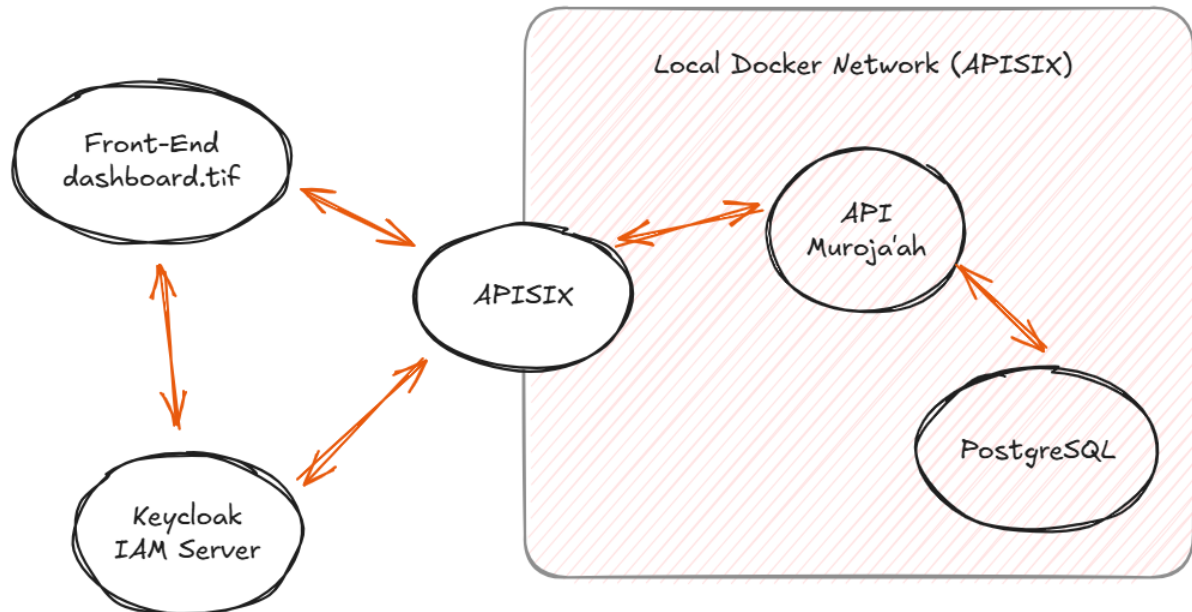
Sistem ini terdiri dari dua lingkungan utama: lingkungan eksternal yang dapat diakses langsung oleh pengguna, dan lingkungan internal yang terisolasi di dalam *Docker Network*.

- a) Lingkungan Eksternal:
  - a. *Frontend*: Aplikasi antarmuka pengguna (*UI*) yang diakses oleh pengguna melalui browser.
  - b. *Keycloak IAM Server*: Server tunggal yang didedikasikan untuk menangani seluruh proses otentikasi (login) dan otorisasi (hak akses).
- b) Lingkungan Internal (Berjalan di dalam satu Docker Network):
  - a. *Apache APISIX API Gateway*: Pintu gerbang utama yang menerima semua permintaan *API* dari *Frontend*. Komponen ini bertanggung jawab dalam



routing integrasi ke-API internal dan untuk integrasi ke-*Keycloak IAM Server* menggunakan plugin *authz-keycloak* dengan alasan keamanan.

- b. *API Muroja'ah*: Layanan backend utama yang berisi semua logika bisnis aplikasi *Muroja'ah Juz 30*. Layanan ini dikembangkan dengan *Framework Hono* dan menggunakan bahasa pemrograman *Typescript*.
- c. *PostgreSQL RDBMS*: Database yang digunakan oleh *API Muroja'ah* untuk menyimpan dan mengambil data.



Gambar 1: Alur Arsitektur Umum

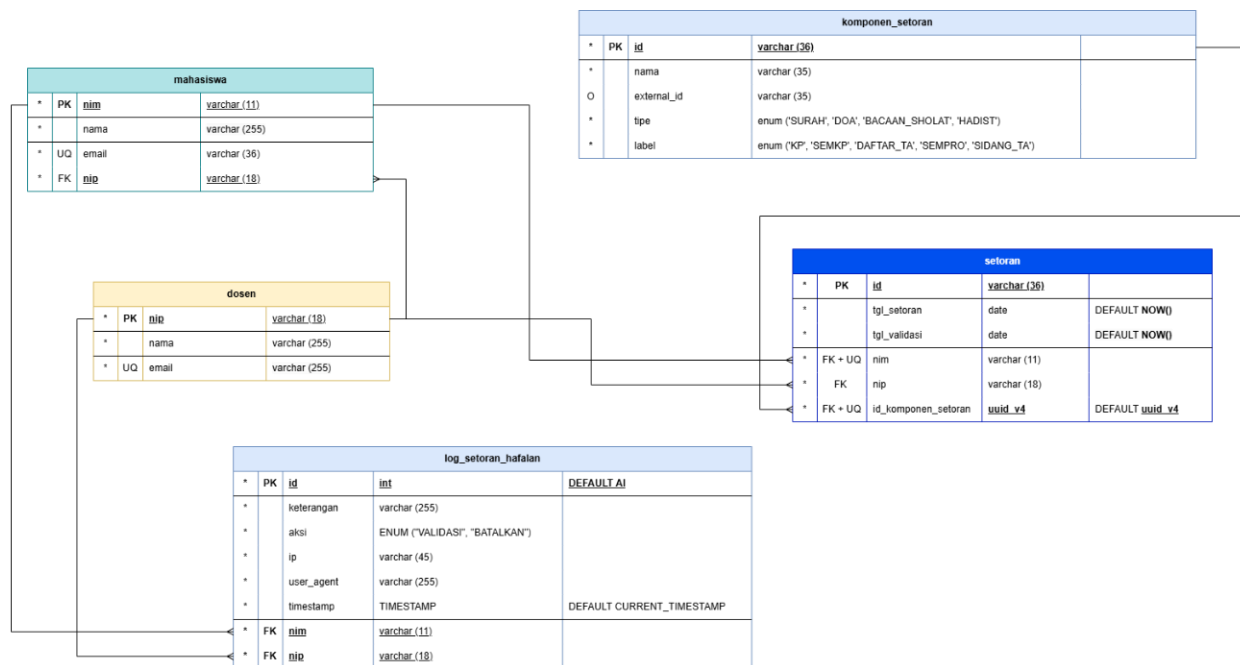
Alur permintaan data dari pengguna berjalan sebagai berikut:

- **Langkah 1: Otentikasi Pengguna**
  1. Pengguna membuka aplikasi Frontend dan memulai proses login.
  2. *Frontend* mengarahkan pengguna ke halaman login yang disediakan oleh *Keycloak IAM Server*.
  3. Setelah pengguna berhasil memasukkan kredensial, *Keycloak IAM Server* akan membuat sebuah kode otorisasi, lalu dikirimkan kembali ke *Frontend*, lalu *Frontend* kembali mengirimkan kode tersebut ke server *Keycloak IAM*, dan *Keycloak IAM Server* akan membuat sebuah *Access Token* dan *Refresh Token*, lalu mengirimkannya kembali ke *Frontend*.
- **Langkah 2: Permintaan API dari Frontend**
  1. Frontend, yang kini sudah memegang *Access Token*, membuat permintaan untuk mengakses route yang telah di definisikan dalam *APACHE Apisix API Gateway*, permintaan ini dikirimkan wajib menyertakan *Access Token* tersebut di dalam *Authorization header* sebagai *Bearer Token*.
- **Langkah 3: Validasi oleh API Gateway**
  1. Apache APISIX API Gateway menerima permintaan dari Frontend.
  2. Gateway kemudian berkomunikasi dengan *Keycloak IAM Server* untuk memvalidasi keaslian dan masa berlaku *Access Token*. Selain itu juga akan dilakukan otorisasi memvalidasi apakah peran dari token tersebut *eligible* mengakses *route* yang diminta.

- Langkah 4: Penerusan Permintaan ke Backend
  1. Jika Keycloak Server mengonfirmasi bahwa token valid, Apache APISIX API Gateway akan meneruskan permintaan tersebut ke *API Muroja'ah*.
  2. Proses ini terjadi di dalam jaringan Docker yang aman dan terisolasi.
- Langkah 5: Pemrosesan dan Akses Database
  1. *API Muroja'ah* menerima permintaan, memproses logika bisnisnya, dan berinteraksi dengan *PostgreSQL RDBMS* untuk mengambil ataupun memanipulasi data.
  2. Setelah selesai, *API Muroja'ah* mengirimkan data respons kembali ke *Apache APISIX API Gateway*.
- Langkah 6: Pengiriman Respons ke Pengguna
  1. *Apache APISIX API Gateway* menerima respons dari *API Muroja'ah*.
  2. *Apache APISIX API Gateway* meneruskan respons akhir tersebut kembali ke *Frontend*, yang kemudian *Frontend* akan mengolah dan menampilkannya kepada pengguna.

## 2.3 Perancangan Database

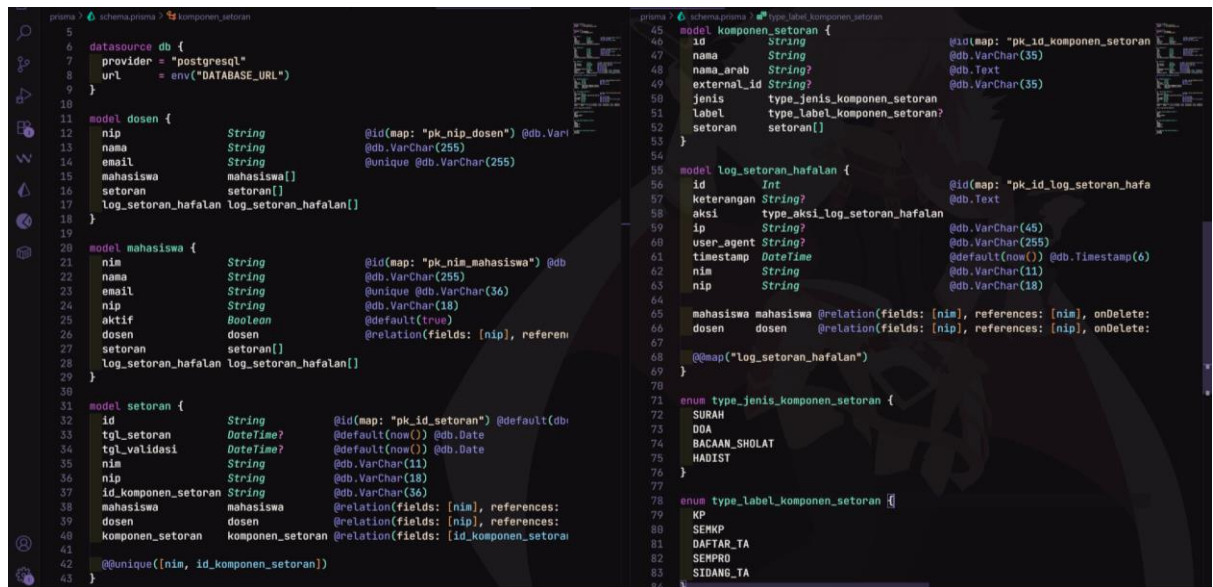
Berikut adalah skema database yang disusun untuk mendukung kebutuhan sistem dan menggantikan proses konvensional dari dokumen fisik menjadi mekanisme pencatatan digital yang terstruktur dan efisien.



Gambar 2: ERD Database Muroja'ah

Gambar diatas merupakan desain rancangan basis data, atau *ERD* dari sistem *muroja'ah* yang akan dikembangkan, setelah rancangan dibentak, akan dibuat implementasi teknisnya dalam bentuk *DDL (Data Definition Language) SQL*, untuk mempermudah proses dan bisa di skalabilitaskan untuk berpindah ke-jenis *RDBMS* lainnya, disini akan menggunakan *ORM-PRISMA* sebagai wadah Implementasi teknis dari rancangan *ERD* diatas.

Dibawah ini akan diberikan tangkapan layar hasil Implementasi teknis dari *ERD* diatas dalam bentuk *schema.prisma* yang mana merupakan format khusus *syntax* dai *ORM-PRISMA* yang digunakan pada pengembangan kali ini.



Gambar 3: schema.prisma DB\_Muroja'ah

## 2.4 Perancangan Dockerfile

Berikut rancangan *entrypoint.sh* dan *dockerfile* yang digunakan untuk proses deployment pada server prodi TIF UIN Suska Riau pada platform Portainer.

```

entrypoint.sh
1  #!/bin/sh
2
3  echo "[INFO] Running Prisma migrations! 🔥"
4  bunx prisma migrate deploy
5
6  echo "[INFO] Seeding the database! 🌱"
7  bunx prisma db seed
8
9  echo "[INFO] Starting the server! 🚀"
10 bun run src/index.ts

```

Gambar 4: entrypoint.sh API Muroja'ah

```

Dockerfile > ...
1  FROM oven/bun:1
2
3  WORKDIR /app
4
5  ENV NODE_ENV=production
6
7  RUN apt-get update && apt-get install -y openssl && rm -rf /var/lib/apt/lists/*
8
9  # Install dependencies
10 COPY package.json bun.lockb ./
11 RUN bun install --frozen-lockfile
12
13 # Copy source code
14 COPY . .
15
16 # Generate Prisma Client
17 RUN bunx prisma generate
18
19 # Copy and prepare entrypoint
20 COPY entrypoint.sh /entrypoint.sh
21 RUN chmod +x /entrypoint.sh
22
23 # Use entrypoint script as startup command
24 CMD ["/entrypoint.sh"]

```

Gambar 5: Dockerfile API Muroja'ah

## 2.5 Perancangan *Endpoint API*

API dirancang sebagai 8 endpoint yang telah disesuaikan dengan konsep pendekatan *RESTful*, daftar *endpoint* ini mencerminkan kebutuhan fungsional, menggantikan proses berbasis dokumen fisik dengan interaksi digital.

Tabel 1: Rancangan Umum *Endpoint API Muroja'ah*

Peran	Method	<i>Endpoint</i>	Deskripsi
Publik	GET	/mahasiswa/kartu-murojaah-digital/:id	Mengambil statistik, riwayat aktivitas, serta detail progres <i>muroja'ah Juz 30</i> mahasiswa untuk validasi keabsahan.
Internal Antar Sistem	GET	/mahasiswa/check-murojaah/:nim?syarat=	Mengambil status progres <i>muroja'ah Juz 30</i> mahasiswa persyarat berdasarkan NIM.
Dosen	GET	/dosen/pa-saya	Mengambil daftar semua mahasiswa PA-nya beserta ringkasan singkat progres <i>Muroja'ah Juz 30</i> -nya.
	POST	/mahasiswa/setoran/:nim	Memvalidasi <i>Muroja'ah Juz 30</i> mahasiswa.
	DELETE	/mahasiswa/setoran/:nim	Membatalkan validasi <i>Muroja'ah Juz 30</i> mahasiswa.
	GET	/mahasiswa/setoran/:nim	Mengambil statistik progres, riwayat aktivitas, serta detail progres <i>muroja'ah Juz 30</i> dari mahasiswa PA-nya.
Mahasiswa	GET	/mahasiswa/setoran-saya	Mengambil statistik, riwayat aktivitas, serta detail progres <i>muroja'ah Juz 30</i> miliknya sendiri.
	GET	/mahasiswa/kartu-murojaah-saya	Mengambil <i>PDF</i> Kartu <i>Muroja'ah Juz 30</i> miliknya sendiri.

## BAB III

### IMPLEMENTASI DAN PENGUJIAN

#### 3.1 Mekanisme Penggunaan API Muroja'ah

Setelah sebelumnya dipaparkan rancangan endpoint *API Muroja'ah*, kini akan dilakukan pengujian sekaligus sebagai dokumentasi penggunaan *API Muroja'ah* agar mempermudah *Front-End* dalam melakukan Implementasi teknis-nya. Seperti yang telah dijelaskan pada sub-bab perancangan arsitektur, kita bisa lihat bahwasannya *API Muroja'ah* terhubung ke *Apache APISIX API Gateway*, oleh karna itu setiap *request* yang nantinya akan dilakukan oleh *Frontend*, wajib menyertakan *AccessToken* pada *Header* dengan *key Authorization* sebagai *Bearer Token*, kecuali *request internal* antar sistem.

Tabel 2: Authentication Mechanism

Authentication	
Headers	Authorization: Bearer <AccessToken>

#### 3.2 Dokumentasi API Muroja'ah

Tabel 2 merupakan hal yang wajib disertakan pada tiap *request* apapun, karna semua *route endpoint* sudah terproteksi. Sekarang kita akan menguji semua *Endpoint API Muroja'ah* yang telah dipaparkan pada bagian perancangan sebelumnya.

Tabel 3: Endpoint /mahasiswa/kartu-murojaah-digital/:id

Peran	Method	Endpoint	Deskripsi
Publik	GET	/mahasiswa/kartu-murojaah-digital/:id	Mengambil statistik, riwayat aktivitas, serta detail progres <i>muroja'ah Juz 30</i> mahasiswa untuk validasi keabsahan.

Dibawah ini diberikan tabel pengujian sekaligus dokumentasi masing-masing *flow* lengkap yang terdapat pada *endpoint* diatas:

Tabel 4: Basic Flow endpoint /mahasiswa/kartu/murojaah-digital/:id

ID	Kasus	Request Payload	
BF	ID yang dikirimkan dalam request valid resmi dikeluarkan langsung dari sistem	Params	id = “8safsanf0cjxxsa20xs9f”
		Query	-
		Body	-
Response Result (200 OK)			
{ "response": true, "message": "Berikut ini info detail mahasiswa dengan riwayat muroja'ah-nya 📖", "data": {			

```
"info": {
  "nama": "M. Farhan Aulia Pratama",
  "nim": "12250113521",
  "email": "12250113521@students.uin-suska.ac.id",
  "angkatan": "2022",
  "semester": 6,
  "dosen_pa": {
    "nip": "198804262019032009",
    "nama": "Iis Afrianty, S.T., M.Sc.",
    "email": "iis.afrianty@uin-suska.ac.id"
  }
},
"setoran": {
  "log": [
    {
      "id": 1001,
      "keterangan": "An-Naba', serta memilih tanggal muroja'ah 2025-04-27",
      "aksi": "VALIDASI",
      "ip": "125.165.156.254",
      "user_agent": "PostmanRuntime/7.44.0",
      "timestamp": "2025-06-02T18:16:38.800Z",
      "nim": "12250113521",
      "dosen_yang_mengesahkan": {
        "nip": "198804262019032009",
        "nama": "Iis Afrianty, S.T., M.Sc.",
        "email": "iis.afrianty@uin-suska.ac.id"
      }
    }
  ],
  "info_dasar": {
    "total_wajib_setor": 37,
    "total_sudah_setor": 7,
    "total_belum_setor": 30,
    "persentase_progres_setor": 18.92,
    "tgl_terakhir_setor": "2025-05-29T00:00:00.000Z",
    "terakhir_setor": "4 Hari yang lalu"
  },
  "ringkasan": [
    {
      "label": "KP",
      "total_wajib_setor": 8,
      "total_sudah_setor": 3,
      "total_belum_setor": 5,
      "persentase_progres_setor": 37.5
    },
    {
      "label": "SEMKP",
      "total_wajib_setor": 8,
      "total_sudah_setor": 0,
      "total_belum_setor": 8,
      "persentase_progres_setor": 0
    },
    {
      "label": "DAFTAR_TA",
      "total_wajib_setor": 6,
      "total_sudah_setor": 0,
      "total_belum_setor": 6,
      "persentase_progres_setor": 0
    },
    {
      "label": "SEMPRO",
```

```

        "total_wajib_setor": 12,
        "total_sudah_setor": 4,
        "total_belum_setor": 8,
        "persentase_progres_setor": 33.33
    },
    {
        "label": "SIDANG_TA",
        "total_wajib_setor": 3,
        "total_sudah_setor": 0,
        "total_belum_setor": 3,
        "persentase_progres_setor": 0
    }
],
"detail": [
    {
        "id": "SURAH_78",
        "nama": "An-Naba'",
        "external_id": "78",
        "nama_arab": "النَّبَأْ",
        "label": "KP",
        "sudah_setor": true,
        "info_setoran": {
            "id": "48a9bd48-bec1-4c16-9dca-30e9c07fd148",
            "tgl_setoran": "2025-04-27",
            "tgl_validasi": "2025-06-02",
            "dosen_yang_mengesahkan": {
                "nip": "198804262019032009",
                "nama": "Iis Afrianty, S.T., M.Sc.",
                "email": "iis.afrianty@uin-suska.ac.id"
            }
        }
    },
    {
        "id": "SURAH_79",
        "nama": "An-Naazi'at",
        "external_id": "79",
        "nama_arab": "النَّازِعَات",
        "label": "KP",
        "sudah_setor": false,
        "info_setoran": null
    },
    .....
]
}
}
}

```

Tabel 5: Exception Flow 1 endpoint mahasiswa/kartu-murojaah-digital/:id

ID	Kasus	Request Payload	
EF_1	ID yang dikirimkan dalam request <b>tidak valid</b> , atau <b>tidak resmi</b> dikeluarkan langsung dari sistem	Params	id = “64cxz63dcbxsa20xs9f”
		Query	-
		Body	-
Response Result (404 Not Found)			
{			

```

"response": false,
"message": "Waduh, datanya gak ditemukan, mau nyari apa sih mas? 🙄"
}

```

Sekarang kita akan masuk pada *endpoint* selanjutnya, tabel dibawah ini akan dilakukan pemaparan secara detail terkait *endpoint* tersebut.

Tabel 6: *endpoint* /mahasiswa/check-murojaah/:nim?syarat=

Peran	Method	Endpoint	Deskripsi
Internal Antar Sistem	GET	/mahasiswa/check-murojaah/:nim?syarat=<KP   SEMKP   DAFTAR_TA   SEMPRO   SIDANG_TA>	Mengambil status progres <i>muroja'ah</i> Juz 30 mahasiswa persyarat berdasarkan NIM.

Dibawah ini diberikan tabel pengujian sekaligus dokumentasi masing-masing *flow* lengkap yang terdapat pada *endpoint* diatas:

Tabel 7: Basic Flow /mahasiswa/check-murojaah/:nim?syarat=

ID	Kasus	Request Payload	
BF	Semua payload yang dikirimkan valid sesuai standar	Params	nim = “12250113521”
		Query	syarat = “KP”
		Body	-
Response Result (200 OK)			
<pre>{   "response": true,   "message": "Berikut ini info muroja'ah-nya! 📋",   "data": {     "nim": "12250113521",     "nama": "M. Farhan Aulia Pratama",     "total_sudah_setor": 3,     "total_wajib_setor": 16,     "total_belum_setor": 13,     "persentase_progres_setor": 18.75,     "is_done": false,     "detail": [       {         "label": "KP",         "total_sudah_setor": 3,         "total_wajib_setor": 8,         "total_belum_setor": 5,         "persentase_progres_setor": 37.5,         "is_done": false       }     ]   } }</pre>			



Tabel 8: Exception Flow 1 /mahasiswa/check-murojaah/:nim?syarat=

ID	Kasus	Request Payload	
EF_1	Tidak mengisi query syarat sama sekali	Params	nim = “12250113521”
		Query	???
		Body	-
Response Result (404 Not Found)			
<pre>{   "response": false,   "message": "Waduh, pastikan dulu dong syarat yang mau di-cek mas! 😊" }</pre>			

Tabel 9: Exception Flow 2 /mahasiswa/check-murojaah/:nim?syarat=

ID	Kasus	Request Payload	
EF_2	Mengisikan query syarat yang tidak terdaftar di-sistem	Params	nim = “12250113521”
		Query	syarat = “KFE”
		Body	-
Response Result (404 Not Found)			
<pre>{   "response": false,   "message": "Waduh, syarat yang mau di-cek kagak ada mas! 🙄" }</pre>			

Sekarang kita akan masuk pada *endpoint* selanjutnya, tabel dibawah ini akan dilakukan pemaparan secara detail terkait *endpoint* tersebut.

Tabel 10: *endpoint* /dosen/pa-saya

<i>Peran</i>	<i>Method</i>	<i>Endpoint</i>	<i>Deskripsi</i>
Dosen	GET	/dosen/pa-saya	Mengambil secara lengkap daftar semua mahasiswa PA-nya beserta ringkasan singkat progres dari masing-masing <i>Muroja'ah</i> Juz 30 mahasiswa PA-nya.

Dibawah ini diberikan tabel pengujian sekaligus dokumentasi masing-masing *flow* lengkap yang terdapat pada *endpoint* diatas:

Tabel 11: Basic Flow /dosen/pa-saya

<i>ID</i>	<i>Kasus</i>	<i>Request Payload</i>
-----------	--------------	------------------------

BF	Semua payload yang dikirimkan valid sesuai standar	Params	-
		Query	-
		Body	-

Response Result (200 OK)

```
{
  "response": true,
  "message": "Berikut info dosen lengkap serta detail mahasiswa per angkatan (max 8 akt) 😊",
  "data": {
    "nip": "198804262019032009",
    "nama": "Iis Afrianty, S.T., M.Sc.",
    "email": "iis.afrianty@uin-suska.ac.id",
    "info_mahasiswa_pa": {
      "ringkasan": [
        {
          "tahun": "2024",
          "total": 2
        },
        {
          "tahun": "2022",
          "total": 12
        },
        .....
      ],
      "daftar_mahasiswa": [
        {
          "email": "12450110490@students.uin-suska.ac.id",
          "nim": "12450110490",
          "nama": "Ahmad Hadid",
          "angkatan": "2024",
          "semester": 2,
          "info_setoran": {
            "total_wajib_setor": 37,
            "total_sudah_setor": 0,
            "total_belum_setor": 37,
            "persentase_progres_setor": 0,
            "tgl_terakhir_setor": null,
            "terakhir_setor": "Belum ada"
          }
        },
        .....
      ]
    }
  }
}
```

Tabel 12: Exception Flow 1 /dosen/pa-saya

<i>ID</i>	<b>Kasus</b>	<i>Request Payload</i>	
<i>EF_1</i>	Data dosen tidak terdaftar di-database	Params	-
		Query	-
		Body	-
<i>Response Result (404 Not Found)</i>			

```
{
  "response": false,
  "message": "Waduh, datanya gak ditemukan, kamu siapa sih mas? 🤖"
}
```

Sekarang kita akan masuk pada *endpoint* selanjutnya, tabel dibawah ini akan dilakukan pemaparan secara detail terkait *endpoint* tersebut.

Tabel 13: *endpoint* POST /mahasiswa/setoran/:nim

Peran	Method	Endpoint	Deskripsi
Dosen	POST	/mahasiswa/setoran/:nim	Memvalidasi <i>Muroja'ah</i> Juz 30 mahasiswa.

Dibawah ini diberikan tabel pengujian sekaligus dokumentasi masing-masing *flow* lengkap yang terdapat pada *endpoint* diatas:

Tabel 14: Basic Flow *endpoint* POST /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
BF	Semua payload yang dikirimkan valid sesuai standar	Params	nim = “12250113521”
		Query	-
		Body	<pre>{   "data_setoran": [     {       "nama_komponen_setoran": "An-Naba'",       "id_komponen_setoran": "SURAH_78"     }   ],   "tgl_setoran": "2025-04-27" // Optional: Bisa Null, }</pre>
Response Result (201 Created)			
<pre>{   "response": true,   "message": "Yeay, proses validasi muroja'ah berhasil! ✨" }</pre>			

Tabel 15: Exception Flow 1 *endpoint* POST /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_1	Data dosen tidak terdaftar di-database	Params	nim = "12250113521"
		Query	-

		Body	<pre> {   "data_setoran": [     {       "nama_komponen_setoran": "An-Naba'",       "id_komponen_setoran": "SURAH_78"     }   ],   "tgl_setoran": "2025-04-27" // Optional: Bisa Null, } </pre>
<b>Response Result (404 Not Found)</b>			
<pre> {   "response": false,   "message": "Waduh, datanya gak ditemukan, kamu siapa sih mas? 🤔" } </pre>			

Tabel 16: Exception Flow 2 endpoint POST /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_2	Mahasiswa yang di- validasi bukan mahasiswa PA-nya	Params	nim = “12450113520”
		Query	-
		Body	<pre>{   "data_setoran": [     {       "nama_komponen_setoran": "An-Naba'",       "id_komponen_setoran": "SURAH_78"     }   ],   "tgl_setoran": "2025-04-27" // Optional: Bisa Null, }</pre>
Response Result (403 Forbidden)			
<pre>{   "response": false,   "message": "Heh, kamu bukan PA dari mahasiswa ini, mau ngapain mas! 😏" }</pre>			

Tabel 17: Exception Flow 3 endpoint POST /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_3	komponen_setoran yang di-validasi, telah di-validasi sebelumnya	Params	nim = “12250113521”
		Query	-
		Body	<pre>{  "data_setoran": [    {      "nama_komponen_setoran": "An-Naba'",      "id_komponen_setoran": "SURAH_78"    }  ],  "tgl_setoran": "2025-04-27" // Optional: Bisa Null,}</pre>
Response Result (400 Bad Request)			
<pre>{  "response": false,  "message": "Waduh, data muroja'ah-nya udah tercatat, mau ngapain lagi mas! 😏"}</pre>			

Sekarang kita akan masuk pada *endpoint* selanjutnya, tabel dibawah ini akan dilakukan pemaparan secara detail terkait *endpoint* tersebut.

Tabel 18: *endpoint GET /mahasiswa/setoran/:nim*

Peran	Method	Endpoint	Deskripsi
Dosen	GET	/mahasiswa/setoran/:nim	Mengambil statistik progres, riwayat aktivitas, serta detail progres <i>muroja'ah Juz 30</i> dari mahasiswa PA-nya.

Dibawah ini diberikan tabel pengujian sekaligus dokumentasi masing-masing *flow* lengkap yang terdapat pada *endpoint* diatas:

Tabel 19: Basic Flow *endpoint GET /mahasiswa/setoran/:nim*

ID	Kasus	Request Payload	
BF	Semua payload yang dikirimkan valid sesuai standar	Params	nim = “12250113521”
		Query	-
		Body	-
Response Result (200 OK)			
{ "response": true, "message": "Berikut ini info detail mahasiswa dengan riwayat muroja'ah-nya ☞", "data": { "info": { "nama": "M. Farhan Aulia Pratama", "nim": "12250113521", "email": "12250113521@students.uin-suska.ac.id", "angkatan": "2022", "semester": 6, "dosen_pa": { "nip": "198804262019032009", "nama": "Iis Afrianty, S.T., M.Sc.", "email": "iis.afrianty@uin-suska.ac.id" } }, "setoran": { "log": [ { "id": 1001, "keterangan": "An-Naba', serta memilih tanggal muroja'ah 2025-04-27", "aksi": "VALIDASI", "ip": "125.165.156.254", "user_agent": "PostmanRuntime/7.44.0", "timestamp": "2025-06-02T18:16:38.800Z", "nim": "12250113521", "dosen_yang_mengesahkan": { "nip": "198804262019032009", "nama": "Iis Afrianty, S.T., M.Sc.",			

```
        "email": "iis.afrianty@uin-suska.ac.id"
    }
}
],
"info_dasar": {
    "total_wajib_setor": 37,
    "total_sudah_setor": 7,
    "total_belum_setor": 30,
    "persentase_progres_setor": 18.92,
    "tgl_terakhir_setor": "2025-05-29T00:00:00.000Z",
    "terakhir_setor": "4 Hari yang lalu"
},
"ringkasan": [
    {
        "label": "KP",
        "total_wajib_setor": 8,
        "total_sudah_setor": 3,
        "total_belum_setor": 5,
        "persentase_progres_setor": 37.5
    },
    {
        "label": "SEMKP",
        "total_wajib_setor": 8,
        "total_sudah_setor": 0,
        "total_belum_setor": 8,
        "persentase_progres_setor": 0
    },
    {
        "label": "DAFTAR_TA",
        "total_wajib_setor": 6,
        "total_sudah_setor": 0,
        "total_belum_setor": 6,
        "persentase_progres_setor": 0
    },
    {
        "label": "SEMPRO",
        "total_wajib_setor": 12,
        "total_sudah_setor": 4,
        "total_belum_setor": 8,
        "persentase_progres_setor": 33.33
    },
    {
        "label": "SIDANG_TA",
        "total_wajib_setor": 3,
        "total_sudah_setor": 0,
        "total_belum_setor": 3,
        "persentase_progres_setor": 0
    }
],
"detail": [
    {
        "id": "SURAH_78",
        "nama": "An-Naba'",
        "external_id": "78",
        "nama_arab": "النبا'",
        "label": "KP",
        "sudah_setor": true,
        "info_setoran": {
            "id": "48a9bd48-bec1-4c16-9dca-30e9c07fd148",
            "tgl_setoran": "2025-04-27",
            "tgl_validasi": "2025-06-02",
```

```

        "dosen_yang_mengesahkan": {
            "nip": "198804262019032009",
            "nama": "Iis Afrianty, S.T., M.Sc.",
            "email": "iis.afrianty@uin-suska.ac.id"
        }
    },
    {
        "id": "SURAH_79",
        "nama": "An-Naazi'at",
        "external_id": "79",
        "nama_arab": "النَّازِعَات",
        "label": "KE",
        "sudah_setor": false,
        "info_setoran": null
    },
    .....
]
}
}
}

```

Tabel 20: Exception Flow 1 endpoint GET /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_1	Data dosen tidak terdaftar di-database	Params	nim = “12250113521”
		Query	-
		Body	
Response Result (404 Not Found)			
<pre>{   "response": false,   "message": "Waduh, datanya gak ditemukan, kamu siapa sih mas? 🤔" }</pre>			

Tabel 21: Exception Flow 2 endpoint POST /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_2	Mahasiswa yang di- validasi bukan mahasiswa PA-nya	Params	nim = “12450113520”
		Query	-
		Body	-
Response Result (403 Forbidden)			
<pre>{   "response": false,   "message": "Heh, kamu bukan PA dari mahasiswa ini, mau ngapain mas! 😏" }</pre>			

Sekarang kita akan masuk pada *endpoint* selanjutnya, tabel dibawah ini akan dilakukan pemaparan secara detail terkait *endpoint* tersebut.

Tabel 22: endpoint DELETE /mahasiswa/setoran/:nim

Peran	Method	Endpoint	Deskripsi
Dosen	DELETE	/mahasiswa/setoran/:nim	Membatalkan validasi Muroja'ah Juz 30 mahasiswa.

Dibawah ini diberikan tabel pengujian sekaligus dokumentasi masing-masing *flow* lengkap yang terdapat pada *endpoint* diatas:

Tabel 23: Basic Flow endpoint DELETE /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
BF	Semua payload yang dikirimkan valid sesuai standar	Params	nim = "12250113521"
		Query	-
		Body	<pre>{  "data_setoran": [    {      "id": "a82cae54-4e02-49a2-a651-6f91dab81c2d",      "id_komponen_setoran": "SURAH_78",      "nama_komponen_setoran": "An-Naba'"    }  ]}</pre>
Response Result (200 OK)			
<pre>{  "response": true,  "message": "Yeay, proses pembatalan validasi muroja'ah telah berhasil! 🎉"}</pre>			

Tabel 24: Exception Flow 1 endpoint DELETE /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_1	Data dosen tidak terdaftar di-database	Params	nim = “12250113521”
		Query	-
		Body	<pre>{  "data_setoran": [    {      "id": "a82cae54-4e02-49a2-a651-6f91dab81c2d",      "id_komponen_setoran": "SURAH_78",      "nama_komponen_setoran": "An-Naba'"    }  ] }</pre>
Response Result (404 Not Found)			
<pre>{  "response": false,  "message": "Waduh, datanya gak ditemukan, kamu siapa sih mas? 🤖" }</pre>			



Tabel 25: Exception Flow 2 endpoint DELETE /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_2	Mahasiswa yang di- validasi bukan mahasiswa PA-nya	Params	nim = “12450113520”
		Query	-
		Body	<pre>{   "data_setoran": [     {       "id": "a82cae54-4e02-49a2-a651-6f91dab81c2d",       "id_komponen_setoran": "SURAH_78",       "nama_komponen_setoran": "An-Naba'"     }   ] }</pre>
Response Result (403 Forbidden)			
<pre>{   "response": false,   "message": "Heh, kamu bukan PA dari mahasiswa ini, mau ngapain mas! 😏" }</pre>			

Tabel 26: Exception Flow 3 endpoint DELETE /mahasiswa/setoran/:nim

ID	Kasus	Request Payload	
EF_3	komponen_setoran yang di-validasi, telah di-validasi sebelumnya	Params	nim = “12250113521”
		Query	-
		Body	<pre>{   "data_setoran": [     {       "id": "a82cae54-4e02-49a2-a651-6f91dab81c2d",       "id_komponen_setoran": "SURAH_78",       "nama_komponen_setoran": "An-Naba"     }   ] }</pre>
Response Result (400 Bad Request)			
<pre>{   "response": false,   "message": "Waduh, data muroja'ah yang mau di-batalin kagak ada, mau ngapain mas! 🙄" }</pre>			