

## PENGEMBANGAN LAYANAN RESTFUL API SISTEM AKADEMIK MAHASISWA ITN MALANG

Yonanda Haryono, Ahmad Faisol, Deddy Rudhistiar

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

2118105@scholar.itn.ac.id

### ABSTRAK

Sistem akademik di ITN Malang masih menggunakan teknologi lama yang memiliki keterbatasan dalam mendukung perkembangan kebutuhan akademik. Seiring bertambahnya jumlah pengguna dan data, sistem seperti ini cenderung mengalami penurunan kinerja. Untuk mengatasi hal tersebut, penelitian ini mengembangkan RESTful API sebagai solusi terintegrasi untuk mendukung pertukaran data secara fleksibel melalui standar HTTP. API dirancang untuk mempercepat akses layanan akademik dan meningkatkan keterhubungan antar sistem. Proses pengembangan meliputi perancangan arsitektur sistem, pemodelan basis data relasional, serta implementasi menggunakan Node.js, Express.js, dan PostgreSQL. Pengujian fungsional menggunakan metode *blackbox* dilakukan untuk mengevaluasi keandalan fungsionalitas dan validasi input. Hasil pengujian *blackbox* menunjukkan seluruh skenario *test case* berhasil dijalankan dengan hasil 100% sesuai harapan. Hal ini membuktikan bahwa sistem merespons dengan tepat terhadap berbagai kondisi input yang diberikan.

**Kata kunci :** RESTful API, sistem akademik, Node.js, PostgreSQL, integrasi data, blackbox testing

### 1. PENDAHULUAN

Sistem akademik memiliki peran penting dalam mendukung pengelolaan data dan informasi akademik di institusi pendidikan. Sistem ini digunakan untuk keperluan rencana studi oleh mahasiswa. Namun, menurut peneliti sistem akademik ITN Malang saat ini masih menggunakan teknologi lama yang memiliki keterbatasan dalam mendukung perkembangan kebutuhan akademik. Seiring bertambahnya jumlah pengguna dan data, sistem seperti ini cenderung mengalami penurunan kinerja.

Untuk mengatasi keterbatasan sistem akademik di ITN Malang, diperlukan solusi yang memungkinkan integrasi dengan berbagai aplikasi lain secara lebih efisien dan terstruktur. Salah satu solusi yang dapat diterapkan adalah pengembangan RESTful API (*Representational State Transfer - Application Programming Interface*). RESTful API adalah arsitektur layanan web yang memungkinkan pertukaran data secara fleksibel dan efisien melalui standar HTTP, sehingga sistem akademik dapat diakses oleh berbagai platform tanpa bergantung pada teknologi tertentu. Dengan penerapan RESTful API, sistem akademik ITN Malang dapat meningkatkan keterhubungan antar sistem, mempercepat pertukaran data, serta mendukung skalabilitas dan efisiensi dalam pengelolaan informasi akademik.

Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengembangkan RESTful API untuk meningkatkan fleksibilitas sistem, serta menyesuaikan sistem akademik dengan perkembangan teknologi informasi.

### 2. TINJAUAN PUSTAKA

#### 2.1. Penelitian Terdahulu

Penggunaan RESTful API juga telah diterapkan dalam berbagai studi terdahulu, salah satunya oleh

Mebiyantara (2021) yang mengembangkan aplikasi mobile untuk manajemen *remote laboratory* berbasis RESTful *Web Service*. Aplikasi tersebut memungkinkan praktikum jarak jauh secara efisien melalui perangkat *mobile*, dengan pengujian yang menunjukkan sistem memenuhi kebutuhan fungsional dan non-fungsional. Studi ini memperkuat potensi RESTful API dalam membangun sistem layanan terdistribusi yang responsif dan ringan [1].

#### 2.2. Restful API

Dalam pengembangan aplikasi, database berfungsi sebagai tempat penyimpanan data yang diperlukan oleh aplikasi. Namun, di lingkungan perusahaan dengan berbagai departemen, pertukaran data antar sistem menjadi penting untuk efisiensi kerja. API (*Application Programming Interface*) memungkinkan komunikasi antar aplikasi, dengan REST (*Representational State Transfer*) sebagai arsitektur yang sering digunakan. REST mengatur cara pengiriman data melalui HTTP, di mana server menyediakan *resource* dan klien menerima data dalam format JSON, XML, atau teks biasa [2].

#### 2.3. Node.js

Node.js adalah lingkungan *runtime* JavaScript *open-source* dan *cross-platform* yang memungkinkan eksekusi kode JavaScript di luar *browser*. Dengan desain yang mendukung aplikasi jaringan skalabel, Node.js cocok untuk membangun aplikasi real-time seperti server *web*, *game*, dan layanan chatting [3].

#### 2.4. PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional berbasis objek (ORDBMS) *open-source* yang mendukung SQL dan dilengkapi fitur seperti

transaksi, *subqueries*, dan *triggers* untuk pengelolaan data kompleks[4].

## 2.5. Agile

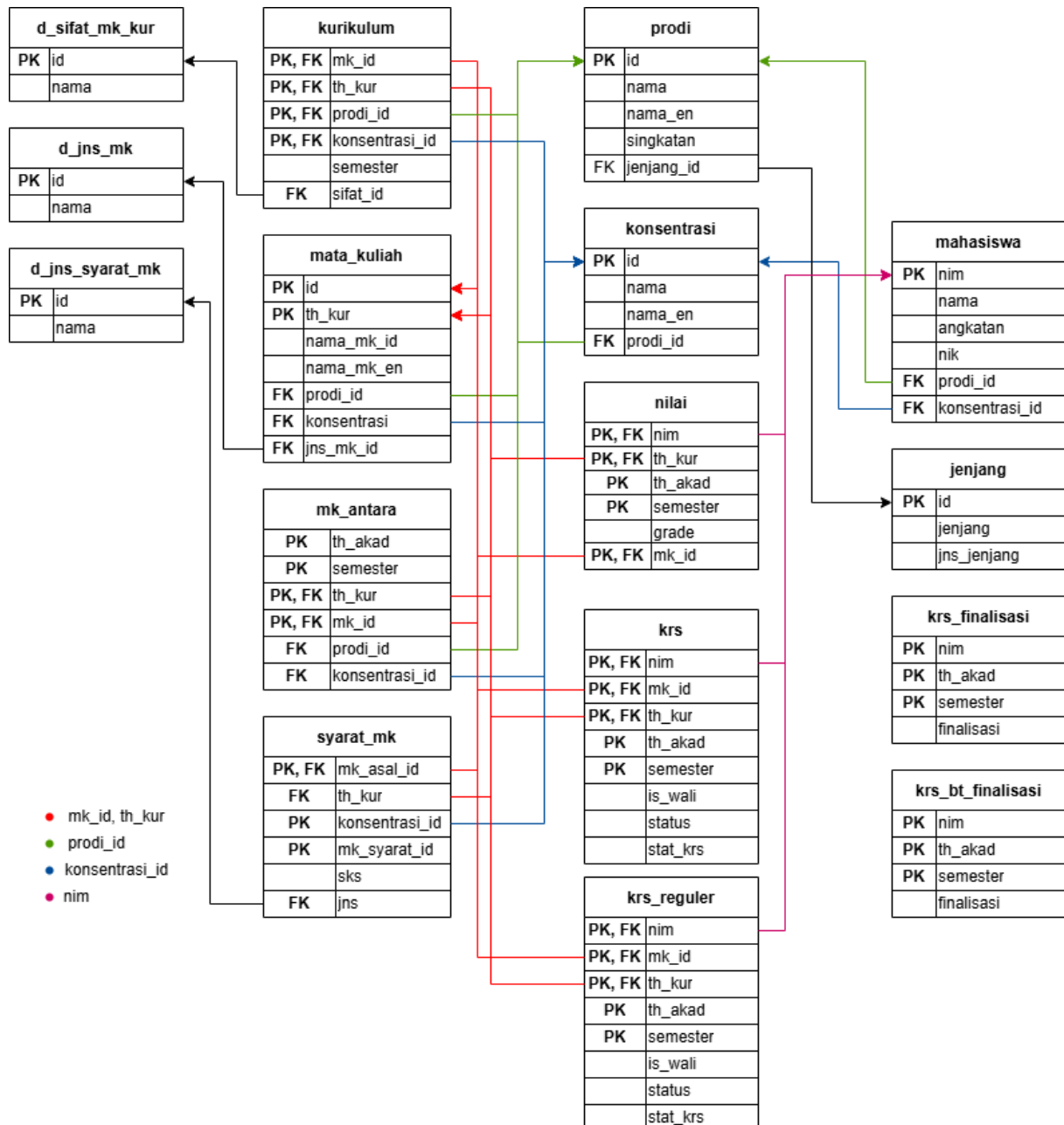
*Agile Software Development* adalah metode pengembangan perangkat lunak yang bersifat fleksibel

dan iteratif. Prosesnya meliputi pengumpulan kebutuhan, perancangan, pengkodean, pengujian, perilsan, dan evaluasi untuk perbaikan berkelanjutan[5].

## 3. METODE PENELITIAN

### 3.1. Relasi Tabel

Berikut merupakan hubungan antar tiap tabel dari pengembangan RESTful API Sistem Akademik.



Gambar 1. Relasi Tabel Layanan RESTful API Sistem Akademik

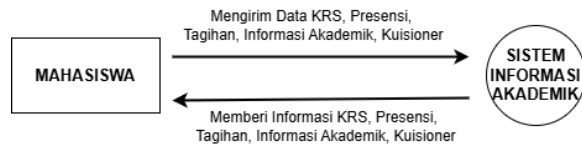
Gambar 1 menampilkan ERD sistem akademik yang mencakup entitas Mahasiswa, Kurikulum, Mata Kuliah, KRS, dan Nilai. Mahasiswa terhubung dengan program studi dan konsentrasi, sementara data

akademiknya tercatat pada KRS, Nilai, dan KRS Finalisasi. Kurikulum memuat struktur perkuliahan dan terhubung ke Mata Kuliah. Tabel Syarat\_MK mencatat prasyarat mata kuliah, dengan referensi jenis

syarat dan jenis mata kuliah untuk menjaga konsistensi data [6].

### 3.2. DFD lvl 0

Berikut merupakan Gambaran sistem secara keseluruhan ditunjukkan pada Gambar 4.

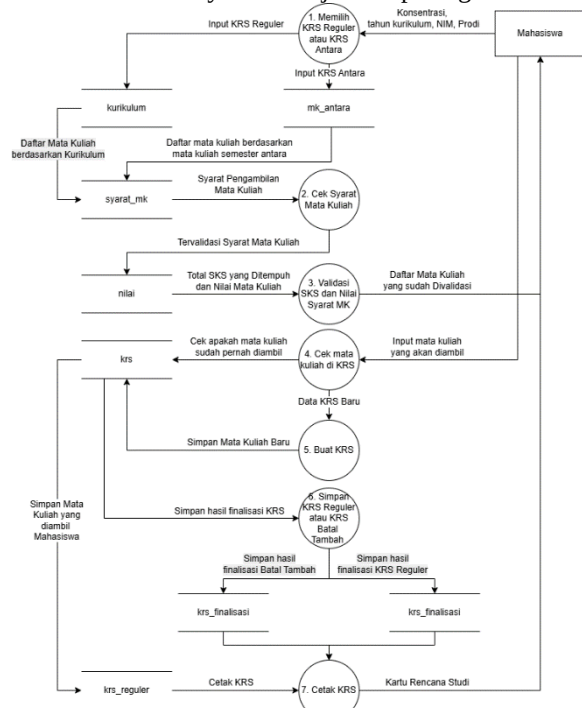


Gambar 2. Flowchart Sistem Layanan RESTful API Sistem Akademik

Gambar 2. DFD Level 0 ini menggambarkan aliran data dalam sistem informasi akademik. Diagram ini menunjukkan hubungan antara mahasiswa dan sistem informasi akademik dalam pengelolaan data akademik. Mahasiswa menginput berbagai data ke dalam sistem, termasuk Kartu Rencana Studi (KRS), kehadiran, pembayaran, informasi akademik, serta kuesioner. Setelah data tersebut diproses, sistem akan mengembalikan informasi yang dibutuhkan mahasiswa, seperti status KRS, rekap kehadiran, rincian pembayaran, hasil akademik, dan kuesioner. Proses ini bersifat dua arah, di mana mahasiswa tidak hanya mengirimkan data tetapi juga memperoleh informasi yang telah diproses oleh sistem, sehingga membantu kelancaran administrasi akademik.

### 3.3. DFD lvl 1

Gambaran proses utama sistem Layanan RESTful API Sistem Akademik yaitu ditunjukkan pada gambar 2.

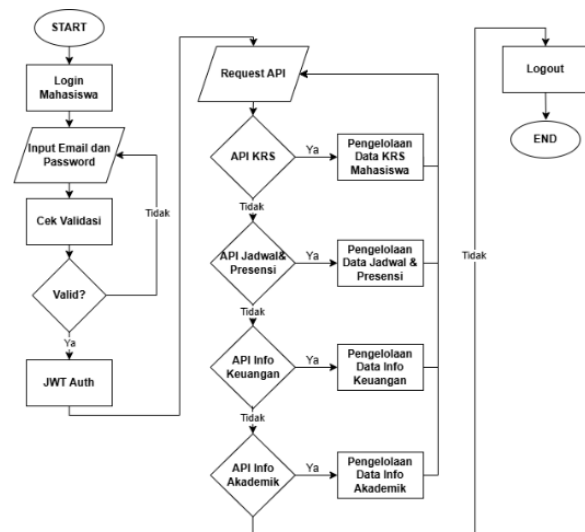


Gambar 3. DFD LVL 1 Layanan RESTful API Sistem Akademik

Gambar 3. menunjukkan tujuh proses utama dalam pengelolaan KRS. Mahasiswa menginput konsentrasi, tahun kurikulum, NIM, dan program studi [7]. Sistem menampilkan mata kuliah sesuai kurikulum, lalu memeriksa syarat pengambilan, SKS, dan nilai yang telah ditempuh. Jika lolos verifikasi dan mata kuliah belum pernah diambil, data disimpan sebagai KRS. Mahasiswa kemudian dapat melakukan finalisasi dan mencetak KRS sebagai dokumen resmi

### 3.4. Flowchart Sistem

Gambaran umum *flowchart* dari sistem yang akan dikembangkan yaitu ditunjukkan pada Gambar 3.



Gambar 4. Flowchart Sistem Layanan RESTful API Sistem Akademik

Gambar 4. menampilkan *flowchart* alur kerja mahasiswa dalam mengelola data KRS melalui autentikasi dan layanan API. Proses dimulai dengan login dilanjutkan validasi. Jika berhasil, sistem menghasilkan token JWT untuk akses. Selanjutnya, sistem memeriksa ketersediaan layanan API (KRS, Jadwal & Presensi, atau Keuangan). Jika tersedia, mahasiswa dapat mengelola data; jika tidak, proses berakhir dengan *logout* [8].

## 4. HASIL DAN PEMBAHASAN

### 4.1. Implementasi Test Design

Pembuatan rencana pengujian mengacu pada *endpoint* API yang dibuat untuk menangani setiap permintaan.

Tabel 1. Tabel Endpoint API Kartu Rencana Studi (KRS)

| HTTP Method | URL                                                   | Keterangan                                                          |
|-------------|-------------------------------------------------------|---------------------------------------------------------------------|
| GET         | /api/krs/init/{konsentrasi}/{th_kur}/{nim}/{semester} | Mendapat data kurikulum berdasarkan konsentrasi dan tahun kurikulum |
| GET         | /api/krs/{th_kur}/{semester}/{th_akad}/{nim}          | Mengambil hasil KRS mahasiswa                                       |

| HTTP Method | URL                                                 | Keterangan                                       |
|-------------|-----------------------------------------------------|--------------------------------------------------|
| POST        | /api/krs                                            | Menambah rencana studi mahasiswa                 |
| DELETE      | /api/krs/{th_kur}/{semester}/{th_akad}/{nim}?mk_id= | Menghapus data rencana studi mahasiswa           |
| POST        | /api/krs/finalisasi                                 | Menambah data finalisasi rencana studi mahasiswa |

## 4.2. Implementasi API

### a. Memperoleh Data Kurikulum berdasarkan Konsentrasi dan Tahun Kurikulum

Endpoint `GET: /api/krs/init/{konsentrasi}/{th_kur}/{nim}` digunakan untuk mengambil data KRS mahasiswa berdasarkan NIM, tahun kurikulum, dan konsentrasi, serta menampilkan daftar mata kuliah yang akan diambil.

| Code | Description       |
|------|-------------------|
| 200  | Response berhasil |

```

{
  "status": true,
  "message": "berhasil",
  "data": {
    "list_mk": [
      {
        "mata_kuliah": {
          "id": "IF1101",
          "nama": "Matematika",
          "nama_en": "Mathematic",
          "sks_kuliah": 3,
          "sks_seminar": 0,
          "sks_praktik": 0,
          "sks_total": 3,
          "jenis": {
            "id": "1",
            "nama": "Mata Kuliah Kelas"
          }
        },
        "semester": "1",
        "sifat": {
          "id": "1",
          "nama": "Mata Kuliah Wajib"
        },
        "prasyarat": [
          {
            "mata_kuliah": {
              "id": "IF1101",
              "nama": "Algoritma"
            },
            "jenis": {
              "id": "1",
              "nama": "Mata Kuliah Pra Syarat"
            },
            "sks_syarat": 0,
            "status": "lulus/gagal"
          }
        ]
      }
    ],
    "batas_sks": 24,
    "ip_mhs": 3.5
  }
}

```

Gambar 5. Response GET KRS berdasarkan Konsentrasi dan Tahun Kurikulum

Gambar 5. menunjukkan endpoint `GET: /api/krs/init/{konsentrasi}/{th_kur}/{nim}` menampilkan daftar mata kuliah yang akan diambil mahasiswa dalam satu semester, lengkap dengan nama, kode, SKS, jenis, semester, dan sifat mata kuliah.

### b. Memperoleh Data Rencana Studi berdasarkan NIM

Endpoint `GET: /api/krs/{th_kur}/{semester}/{th_akad}/{nim}` digunakan untuk mengambil data KRS mahasiswa pada semester dan tahun akademik tertentu, mencakup daftar mata kuliah, status pengambilan, dan validasi wali dosen.

| Code | Description       |
|------|-------------------|
| 200  | Response berhasil |

```

{
  "status": true,
  "message": "berhasil",
  "data": {
    "krs": [
      {
        "mata_kuliah": {
          "id": "IF1101",
          "nama": "Matematika",
          "nama_en": "Mathematic",
          "sks_total": 3
        },
        "status": "U/B",
        "is_wali": "1"
      }
    ],
    "finalisasi": "0"
  }
}

```

Gambar 6. Response GET KRS berdasarkan NIM

Gambar 6. menampilkan data KRS berisi daftar mata kuliah yang diambil, dilengkapi ID, nama, total SKS, status pengambilan (baru/ulang), status validasi wali dosen (is\_wali), dan status finalisasi KRS.

### c. Membuat Data Rencana Studi

Endpoint `POST: /api/krs` digunakan untuk menambahkan data KRS baru. Request body dalam format JSON harus memuat nim, th\_kur, semester, th\_akad, dan array ID mata kuliah yang diambil.

| Code | Description       |
|------|-------------------|
| 200  | Response berhasil |

```

{
  "nim": "2418001",
  "th_kur": "2024",
  "semester": "1",
  "th_akad": "2024",
  "mata_kuliah": [
    "IF1101",
    "IF1107"
  ]
}

```

| Code | Description       |
|------|-------------------|
| 200  | Response berhasil |

```

{
  "status": true,
  "message": "berhasil",
  "data": {
    "nim": "2418001",
    "mk": "IF1101",
    "th_kur": "2024",
    "semester": "1",
    "th_akad": "2024",
    "is_wali": "0",
    "status": "U/B"
  }
}

```

Gambar 7. Response POST KRS

Response dari endpoint `POST: /api/krs` menampilkan detail KRS yang baru ditambahkan, mencakup nim, ID mata kuliah (mk), tahun kurikulum (th\_kur), semester, tahun akademik (th\_akad), status persetujuan wali dosen (is\_wali), dan status mata kuliah.

## d. Menghapus Data Rencana Studi

Endpoint `DELETE: /api/krs/{th_kur}/{semester}/{th_akad}/{nim}` digunakan untuk menghapus mata kuliah yang belum finalisasi dan validasi wali dosen.

| Code | Description     |
|------|-----------------|
| 200  | Hapus berhasil. |

```
{
  "status": true,
  "message": "berhasil"
}
```

Gambar 8. Response DELETE KRS

Pada endpoint `DELETE: /api/krs/{th_kur}/{semester}/{th_akad}/{nim}`, pengguna mengirimkan permintaan untuk menghapus mata kuliah yang belum finalisasi dan validasi wali dosen.

## e. Membuat Data Finalisasi Rencana Studi

Endpoint `POST: /api/krs/finalisasi` digunakan untuk memfinalisasi KRS mahasiswa, mengonfirmasi data yang telah terinput agar tidak bisa diubah lagi.

```
{
  "nim": "2418001",
  "semester": "1",
  "th_akad": "2024",
  "finalisasi": "1"
}
```

| Code | Description       |
|------|-------------------|
| 200  | Response berhasil |

```
{
  "status": true,
  "message": "berhasil",
  "data": {
    "nim": "2418001",
    "semester": "1",
    "th_akad": "2024",
    "finalisasi": "1"
  }
}
```

Gambar 9. Response POST Finalisasi KRS

Pada endpoint `POST: /api/krs/finalisasi`, pengguna mengirimkan permintaan untuk memfinalisasi KRS berdasarkan nim, semester, dan th\_akad. Jika valid, API merespon dengan informasi nim, semester, th\_akad, dan status finalisasi.

## 4.3. Pengujian Fungsional (Blackbox Testing)

a. Pengujian Endpoint `GET` Data Kurikulum berdasarkan Konsentrasi dan Tahun Kurikulum

Pengujian ini memastikan endpoint `GET /api/krs/init/{konsentrasi}/{th_kur}/{nim}/{semester}` berfungsi dengan baik dalam menampilkan data kurikulum berdasarkan input. Pengujian mencakup skenario input valid dan tidak valid, dengan hasil yang menunjukkan respons yang benar dan pesan error yang sesuai [9].

Tabel 2. Tabel Pengujian Fungsional Blackbox `GET` Data Kurikulum berdasarkan Konsentrasi dan Tahun Kurikulum

| Skenario Pengujian                                                                                                                       | Hasil yang Diharapkan                                                                         | Hasil    |
|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------|
| Melakukan request <code>GET</code> pada API <code>{{base_url}}/krs/init/010801/2024/2118105/1</code>                                     | Menampilkan daftar kurikulum berdasarkan konsentrasi dan tahun kurikulum                      | Berhasil |
| Melakukan request <code>GET</code> dengan data kosong pada API <code>{{base_url}}/krs/init/010801/2024/2118105/1</code>                  | Menampilkan response error status code 404 dengan pesan "Data Tidak Ditemukan."               | Berhasil |
| Melakukan request <code>GET</code> tanpa konsentrasi pada API <code>{{base_url}}/krs/init/2024/2118105/1</code>                          | Menampilkan response error status code 400 dengan pesan "Konsentrasi tidak boleh kosong."     | Berhasil |
| Melakukan request <code>GET</code> dengan konsentrasi berupa abjad pada API <code>{{base_url}}/krs/init/asd/2024/2118105/1</code>        | Menampilkan response error status code 400 dengan pesan "Konsentrasi harus berupa angka."     | Berhasil |
| Melakukan request <code>GET</code> dengan konsentrasi yang tidak valid pada API <code>{{base_url}}/krs/init/010801/2024/2118105/1</code> | Menampilkan response error status code 400 dengan pesan "Data konsentrasi tidak ditemukan."   | Berhasil |
| Melakukan request <code>GET</code> tanpa tahun kurikulum pada API <code>{{base_url}}/krs/init/010801/2118105/1</code>                    | Menampilkan response error status code 400 dengan pesan "Tahun kurikulum tidak boleh kosong." | Berhasil |
| Melakukan request <code>GET</code> dengan tahun kurikulum berupa abjad pada API <code>{{base_url}}/krs/init/010801/asd/2118105/1</code>  | Menampilkan response error status code 400 dengan pesan "Tahun kurikulum harus berupa angka." | Berhasil |
| Melakukan request <code>GET</code> tanpa NIM pada API <code>{{base_url}}/krs/init/010801/2024/1</code>                                   | Menampilkan response error status code 400 dengan pesan "NIM tidak boleh kosong."             | Berhasil |
| Melakukan request <code>GET</code> dengan API <code>{{base_url}}/krs/init/010801/2024/asd/1</code>                                       | Menampilkan response error status code 400 dengan pesan "NIM harus berupa angka."             | Berhasil |

| Skenario Pengujian                                                                                                                   | Hasil yang Diharapkan                                                                     | Hasil    |
|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|----------|
| Melakukan <i>request GET</i> dengan NIM yang tidak valid pada API <code>{{base_url}}/krs/init/010801/2024/21181051/1</code>          | Menampilkan response error status code 400 dengan pesan "Data mahasiswa tidak ditemukan." | Berhasil |
| Melakukan <i>request GET</i> tanpa semester pada API <code>{{base_url}}/krs/init/010801/2024/2118105/</code>                         | Menampilkan response error status code 400 dengan pesan "Semester tidak boleh kosong."    | Berhasil |
| Melakukan <i>request GET</i> dengan semester berupa abjad pada API <code>{{base_url}}/krs/init/010801/2024/2118105/asd</code>        | Menampilkan response error status code 400 dengan pesan "Semester harus berupa angka."    | Berhasil |
| Melakukan <i>request GET</i> dengan token kadaluarsa pada API <code>{{base_url}}/krs/init/010801/2024/2118105/1</code>               | Menampilkan response error status code 401 dengan pesan "Invalid Token."                  | Berhasil |
| Melakukan <i>request GET</i> dengan tidak ada token pada API Authorization: <code>{{base_url}}/krs/init/010801/2024/2118105/1</code> | Menampilkan response error status code 403 dengan pesan "Token is Required."              | Berhasil |

Berdasarkan Tabel 2, sistem mampu menampilkan data kurikulum berdasarkan parameter konsentrasi, tahun kurikulum, NIM, dan semester. Dari total 14 jumlah *test case* yang dilakukan, seluruhnya berhasil dijalankan dengan baik, menunjukkan hasil 100% berhasil.

b. Pengujian Endpoint GET Data Rencana Studi berdasarkan NIM

Pengujian ini memastikan *endpoint* `GET /api/krs/{th_kur}/{semester}/{th_akad}/{nim}` berfungsi dengan baik dalam menampilkan data kurikulum berdasarkan input [10]. Pengujian fungsional blackbox ditampilkan pada Tabel 3.

Tabel 3. Tabel Pengujian Fungsional Blackbox GET Data Rencana Studi berdasarkan NIM

| Skenario Pengujian                                                                                                         | Hasil yang Diharapkan                                                                         | Hasil    |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------|
| Melakukan <i>request GET</i> pada API <code>{{base_url}}/krs/2024/1/2024/2118105</code>                                    | Menampilkan daftar rencana studi berdasarkan NIM                                              | Berhasil |
| Melakukan <i>request GET</i> dengan data kosong pada API <code>{{base_url}}/krs/2024/1/2024/2118105</code>                 | Menampilkan response error status code 404 dengan pesan "Data Tidak Ditemukan."               | Berhasil |
| Melakukan <i>request GET</i> tanpa tahun kurikulum pada API <code>{{base_url}}/krs//1/2024/2118105</code>                  | Menampilkan response error status code 400 dengan pesan "Tahun kurikulum tidak boleh kosong." | Berhasil |
| Melakukan <i>request GET</i> dengan tahun kurikulum berupa abjad pada API <code>{{base_url}}/krs/asd/1/2024/2118105</code> | Menampilkan response error status code 400 dengan pesan "Tahun kurikulum harus berupa angka." | Berhasil |
| Melakukan <i>request GET</i> tanpa semester pada API <code>{{base_url}}/krs/2024/2024/2118105</code>                       | Menampilkan response error status code 400 dengan pesan "Semester tidak boleh kosong."        | Berhasil |
| Melakukan <i>request GET</i> dengan semester berupa abjad pada API <code>{{base_url}}/krs/2024/asd/2024/2118105</code>     | Menampilkan response error status code 400 dengan pesan "Semester harus berupa angka."        | Berhasil |
| Melakukan <i>request GET</i> tanpa tahun akademik pada API <code>{{base_url}}/krs/2024/1//2118105</code>                   | Menampilkan response error status code 400 dengan pesan "Tahun akademik tidak boleh kosong."  | Berhasil |
| Melakukan <i>request GET</i> dengan tahun akademik berupa abjad pada API <code>{{base_url}}/krs/2024/1/asd/2118105</code>  | Menampilkan response error status code 400 dengan pesan "Tahun akademik harus berupa angka."  | Berhasil |
| Melakukan <i>request GET</i> NIM dengan NIM yang tidak valid pada API <code>{{base_url}}/krs/2024/1/2024/21181051</code>   | Menampilkan response error status code 400 dengan pesan "Data mahasiswa tidak ditemukan."     | Berhasil |
| Melakukan <i>request GET</i> tanpa NIM pada API <code>{{base_url}}/krs/2024/1/2024/</code>                                 | Menampilkan response error status code 400 dengan pesan "NIM tidak boleh kosong."             | Berhasil |
| Melakukan <i>request GET</i> dengan NIM berupa abjad pada API <code>{{base_url}}/krs/2024/1/2024/asd</code>                | Menampilkan response error status code 400 dengan pesan "NIM harus berupa angka."             | Berhasil |
| Melakukan <i>request GET</i> dengan token kadaluarsa pada API <code>{{base_url}}/krs/2024/1/2024/2118105</code>            | Menampilkan response error status code 401 dengan pesan "Invalid Token."                      | Berhasil |
| Melakukan <i>request GET</i> tanpa token pada API Authorization: <code>{{base_url}}/krs/2024/1/2024/2118105</code>         | Menampilkan response error status code 403 dengan pesan "Token is Required."                  | Berhasil |

Berdasarkan Tabel 3. Permintaan data rencana studi berdasarkan parameter tahun kurikulum, semester, tahun akademik, dan NIM berhasil dijalankan. Dari total 13 jumlah *test case* yang dilakukan, seluruhnya berhasil dijalankan dengan baik, menunjukkan hasil 100% berhasil.

c. Pengujian Endpoint POST Data Rencana Studi

Pengujian ini memastikan *endpoint* `POST /api/krs` dapat memproses data rencana studi dengan benar, mencakup skenario data valid, NIM tidak terdaftar, dan kode mata kuliah yang tidak sesuai. Pengujian fungsional blackbox disajikan pada Tabel 4.

Tabel 4. Tabel Pengujian Fungsional Blackbox POST Data Rencana Studi

| Skenario Pengujian                                                         | Hasil yang Diharapkan                                                                  | Hasil    |
|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------|----------|
| Melakukan <i>request</i> pada /api/krs                                     | Menampilkan data mata kuliah, kode status, dan pesan berupa json.                      | Berhasil |
| Melakukan <i>request POST</i> dengan NIM tidak valid pada /api/krs         | Menampilkan response error status code 422 dengan pesan “Mahasiswa tidak ditemukan.”   | Berhasil |
| Melakukan <i>request POST</i> dengan mata kuliah tidak valid pada /api/krs | Menampilkan response error status code 422 dengan pesan “Mata kuliah tidak ditemukan.” | Berhasil |
| Melakukan <i>request POST</i> dengan token kadaluarsa pada /api/krs        | Menampilkan <i>response</i> error status code 401 dengan pesan “Invalid Token.”        | Berhasil |
| Melakukan <i>request POST</i> tanpa token pada /api/krs                    | Menampilkan <i>response</i> error status code 403 dengan pesan “Token is Required.”    | Berhasil |

Berdasarkan Tabel 4. Penambahan rencana studi melalui *body request* berhasil diproses dengan benar, mencakup NIM, tahun kurikulum, semester, tahun akademik, dan daftar ID mata kuliah. Dari total 5 jumlah test case yang dilakukan, seluruhnya berhasil dijalankan dengan baik, menunjukkan hasil 100% berhasil.

d. Pengujian Endpoint DELETE Data Rencana Studi  
Pengujian ini memastikan *endpoint* DELETE /api/krs/{th\_kur}/{semester}/{th\_akad}/{nim}?mk\_id= menghapus data rencana studi dengan benar, mencakup skenario data valid dan kondisi kesalahan seperti parameter kosong atau data tidak ditemukan. Pengujian fungsional blackbox disajikan pada Tabel 5.

Tabel 5. Tabel Pengujian Fungsional Blackbox DELETE Data Rencana Studi

| Skenario Pengujian                                                                                                         | Hasil yang Diharapkan                                                                         | Hasil    |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------|
| Melakukan <i>request DELETE</i> pada API {{base_url}}/krs/2024/1/2024/2118105?mk_id=IF1102                                 | Menampilkan pesan “Berhasil.”                                                                 | Berhasil |
| Melakukan <i>request GET</i> tanpa tahun kurikulum pada API {{base_url}}/krs/1/2024/2118105?mk_id=IF1102                   | Menampilkan response error status code 400 dengan pesan “Tahun kurikulum tidak boleh kosong.” | Berhasil |
| Melakukan <i>request GET</i> dengan tahun kurikulum berupa abjad pada API {{base_url}}/krs/asd/1/2024/2118105?mk_id=IF1102 | Menampilkan response error status code 400 dengan pesan “Tahun kurikulum harus berupa angka.” | Berhasil |
| Melakukan <i>request GET</i> tanpa semester pada API {{base_url}}/krs/2024//2024/2118105?mk_id=IF1102                      | Menampilkan response error status code 400 dengan pesan “Semester tidak boleh kosong.”        | Berhasil |
| Melakukan <i>request GET</i> dengan semester berupa abjad pada API {{base_url}}/krs/2024/asd/2024/2118105?mk_id=IF1102     | Menampilkan response error status code 400 dengan pesan “Semester harus berupa angka.”        | Berhasil |
| Melakukan <i>request GET</i> tanpa tahun akademik pada API {{base_url}}/krs/2024/1//2118105?mk_id=IF1102                   | Menampilkan response error status code 400 dengan pesan “Tahun akademik tidak boleh kosong.”  | Berhasil |
| Melakukan <i>request GET</i> dengan tahun akademik berupa abjad pada API {{base_url}}/krs/2024/1/asd/2118105?mk_id=IF1102  | Menampilkan response error status code 400 dengan pesan “Tahun akademik harus berupa angka.”  | Berhasil |
| Melakukan <i>request GET</i> NIM dengan NIM yang tidak valid pada API {{base_url}}/krs/2024/1/2024/21181051?mk_id=IF1102   | Menampilkan response error status code 400 dengan pesan “Data mahasiswa tidak ditemukan.”     | Berhasil |
| Melakukan <i>request GET</i> tanpa NIM pada API {{base_url}}/krs/2024/1/2024/?mk_id=IF1102                                 | Menampilkan response error status code 400 dengan pesan “NIM tidak boleh kosong.”             | Berhasil |
| Melakukan <i>request GET</i> dengan NIM berupa abjad pada API {{base_url}}/krs/2024/1/2024/asd?mk_id=IF1102                | Menampilkan response error status code 400 dengan pesan “NIM harus berupa angka.”             | Berhasil |
| Melakukan <i>request GET</i> tanpa MK pada API {{base_url}}/krs/2024/1/2024/2118105?mk_id=                                 | Menampilkan response error status code 400 dengan pesan “Mata kuliah tidak boleh kosong.”     | Berhasil |
| Melakukan <i>request GET</i> dengan MK yang tidak valid pada API {{base_url}}/krs/2024/1/2024/2118105?mk_id=IF11021        | Menampilkan response error status code 400 dengan pesan “Mata kuliah tidak ditemukan.”        | Berhasil |
| Melakukan <i>request DELETE</i> dengan token kadaluarsa pada API {{base_url}}/krs/2024/1/2024/2118105?mk_id=IF1102         | Menampilkan <i>response</i> error status code 401 dengan pesan “Invalid Token.”               | Berhasil |
| Melakukan <i>request DELETE</i> tanpa token pada API Authorization:{{base_url}}/krs/2024/1/2024/2118105?mk_id=IF1102       | Menampilkan <i>response</i> error status code 403 dengan pesan “Token is Required.”           | Berhasil |

Berdasarkan Tabel 5. Permintaan penghapusan mata kuliah dilakukan berdasarkan parameter URL dan query string. Hasil dari total 14 pengujian

menunjukkan sistem berhasil dijalankan dengan baik, menunjukkan hasil 100% berhasil. Sistem hanya menghapus data yang memenuhi syarat (belum



divalidasi), serta mampu menangani berbagai kesalahan input seperti parameter kosong atau tidak valid.

e. Pengujian Endpoint POST Data Finalisasi Rencana Studi

Pengujian ini bertujuan memastikan *endpoint* *POST /api/krs/finalisasi* memproses data rencana studi dengan benar, dengan skenario seperti data lengkap dan valid, NIM tidak terdaftar, dan kode mata kuliah tidak sesuai [11].

Tabel 6. Tabel Pengujian Fungsional Blackbox POST Data Finalisasi Rencana Studi

| Skenario Pengujian                                                                    | Hasil yang Diharapkan                                                                | Hasil    |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|----------|
| Melakukan <i>request</i> pada <i>/api/krs/finalisasi</i>                              | Menampilkan finalisasi dengan status "1" atau "True"                                 | Berhasil |
| Melakukan <i>request POST</i> dengan NIM tidak valid pada <i>/api/krs /finalisasi</i> | Menampilkan response error status code 422 dengan pesan "Mahasiswa tidak ditemukan." | Berhasil |
| Melakukan <i>request</i> dengan token kadaluarsa pada <i>/api/krs/finalisasi</i>      | Menampilkan <i>response</i> error status code 401 dengan pesan "Invalid Token."      | Berhasil |
| Melakukan <i>request</i> tanpa token pada <i>/api/krs/finalisasi</i>                  | Menampilkan <i>response</i> error status code 403 dengan pesan "Token is Required."  | Berhasil |

Berdasarkan Tabel 6. Proses finalisasi KRS berhasil dijalankan dengan menggunakan *body request* yang berisi NIM, semester, tahun akademik, dan status finalisasi. Dari total 4 *test case*, sistem berhasil dijalankan dengan baik, menunjukkan hasil 100% berhasil.

## 5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini adalah berhasil mengembangkan aplikasi layanan akademik berbasis RESTful API menggunakan teknologi terkini seperti Node.js, Express.js, dan PostgreSQL. Hasil penelitian ini menghasilkan sistem akademik dengan rancangan yang mendukung pengembangan layanan dan dapat menjaga kinerja dalam pengelolaan data. Implementasi API yang dikembangkan terbukti efektif dalam meningkatkan efisiensi dan fungsionalitas sistem. Keandalan fungsional sistem dibuktikan melalui serangkaian pengujian menggunakan metode *black box*, yang mencakup berbagai skenario berhasil dan tidak berhasil, seperti parameter kosong, format tidak sesuai, dan validasi *authorization*. Seluruh *test case* yang diuji menunjukkan hasil 100% berhasil, membuktikan bahwa sistem merespons dengan tepat terhadap berbagai kondisi input yang diberikan.

Ke depannya, pengembangan lanjutan dapat mencakup penyimpanan dokumen akademik di layanan *cloud* (seperti Amazon S3 atau Google Cloud Storage), serta dukungan multi-bahasa untuk meningkatkan kenyamanan dan jangkauan pengguna.

## DAFTAR PUSTAKA

- [1] F. Mebiyantara, "Pengembangan Aplikasi Manajemen Remote Laboratory Menggunakan Metode Restful Web Service Berbasis Mobile," *Jurnal Mahasiswa Teknik Informatika(JATI)*, Vol. 5, No.1, Maret 2021.
- [2] B. Ramadhan, "Rancang bangun sistem ticketing parkir menggunakan KTM berbasis Android menggunakan RESTful API," *Jurnal Mahasiswa Teknik Informatika(JATI)*, vol. 7, no. 4, Aug. 2023.
- [3] P. M. Putra, "Rancang bangun virtual assistant chatbot menggunakan Node.js pada layanan sistem informasi akademik," *Proc. seminar nasional pemanfaatan sains dan teknologi informasi*, vol. 1, no. 1, pp. 345–352, 2023.
- [4] M. Firdaus, "Sistem pendukung keputusan pemilihan murid teladan dengan metode SAW berbasis website (studi kasus: SMPN 7 Wuluhan)," *Jurnal Mahasiswa Teknik Informatika(JATI)*, vol. 9, no. 1, Feb. 2025.
- [5] N. Y. Azizah, "Systematic Literature Review Analisis Metode Agile dalam Pengembangan Aplikasi Mobile", *SISTEMASI: Jurnal Sistem Informasi*, Volume 10, Nomor 2, Mei 2021: 369-380, 2021.
- [6] Krisna, W, "Rancang Bangun Sistem Informasi Akademik Menggunakan Framework Codeigniter Pada Universitas Muhammadiyah Purworejo", *Jurnal Sistem Cerdas*, Vol 05 - No 02 eISSN : 2622-8254 pp :107 – 116, 2022.
- [7] S. A. Inggur, "Pengembangan Sistem Layanan Akademik Pengurusan Kartu Rencana Studi Online Pada Sekolah Tinggi Ilmu Kesehatan Stella Maris Makassar", *Jurnal Penelitian Teknik Informatika dan Sistem Informasi (TEMATIKA)*, Vol.10 – No (2), pp :75–86, September 2022.
- [8] D. Mirwansyah, "Perancangan Sistem Informasi Monitoring Akademik Dengan Menggunakan Data Flow Diagram", *Jurnal Locus: Penelitian & Pengabdian*, Volume 2 No. 12 Desember 2023.
- [9] Al Biruni, M., "Penerapan Rest Api Dan Integrasi Midtrans Sebagai Payment Gateway Pada Platform Pelatihan Online (Studi Kasus: Bisabacakitab)," *Jurnal Mahasiswa Teknik Informatika(JATI)*, Vol.7, No.4, Agustus 2023.
- [10] W. P. Priyadi, "Penerapan Data Mining Untuk Clustering Wilayah Produksi Padi Menggunakan Metode K-Means (Studi Kasus : Wilayah Jawa Timur)", *Jurnal Mahasiswa Teknik Informatika (JATI)*, Vol. 8 No. 5, Oktober 2024.
- [11] M.A. Novianto, "Analisis Dan Implementasi Restful Api Guna Pengembangan Sistem Informasi Akademik Pada Perguruan Tinggi", *Jurnal Informatika Terpadu*, Vol. 8 No. 1, pp : 47-61, 2022.