**SKRIPSI**

**IMPLEMENTASI FRAMEWORK LARAVEL UNTUK PENDAFTARAN SAKRAMEN GEREJA BERBASIS WEB (STUDI KASUS : GEREJA SANTO MIKAEL ADISUTJIPTO YOGYAKARTA)**

Diajukan sebagai salah satu syarat untuk menyelesaikan studi jenjang strata satu (S1)

Program Studi Informatika

Sekolah Tinggi Manajemen Informatika dan Komputer

Akakom

Yogyakarta

Disusun Oleh

|  |
| --- |
| **MARIA ALVIRA SOY DJUMA** |
| **Nomor Mahasiswa : 175410120** |

**PROGRAM STUDI INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER AKAKOM**

**YOGYAKARTA**

**2020**

**HALAMAN PERSETUJUAN**

Judul : Implementasi Framework Laravel Untuk Pendaftaran Sakramen Gereja Berbasis Web (Studi Kasus : Gereja Santo Mikael Adisutjipto Yogyakarta)

Nama : Maria Alvira Soy Djuma

NIM : 175410120

Program Studi : Informatika

Jenjang : Strata Satu (S1)

Tahun : 2021

Telah memenuhi syarat dan disetujui untuk diselenggarakan di  
Hadapan dosen penguji seminar tugas akhir  
Yogyakarta, ………………….

Dosen pembimbing,

Indra Yatini Buryadi, S.Kom,.M.Kom.

# **HALAMAN PENGESAHAN**

**SKRIPSI**

**IMPLEMENTASI FRAMEWORK LARAVEL UNTUK PENDAFTARAN SAKRAMEN GEREJA BERBASIS WEB (STUDI KASUS : GEREJA SANTO MIKAEL ADISUTJIPTO YOGYAKARTA)**

Telah dipertahankan di depan Dewan Penguji Skripsi dan dinyatakan diterima untuk memenuhi sebagai syarat guna memperoleh Gelar Sarjana Komputer Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM

YOGYAKARTA

Yogyakarta, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mengesahkan

|  |  |
| --- | --- |
| **Dewan Penguji** | **Tanda Tangan** |
|  | **…………………..……..** |
|  | **……………….….……..** |
|  | **………………..………..** |

**Mengetahui**

**Ketua Program Studi Teknik Informatika**

**(Dini Faktasari, S.T.,M.T.)**

**HALAMAN PERSEMBAHAN**

Puji dan Syukur saya panjatkan kehadirat Tuhan yang Maha Esa karena atas berkat dan kasih sayang-Nya yang telah dilimpahkan kepada kita semua. Akhirnya terselesaikan juga tugas akhir ini dan untuk itu saya ingin mempersembahkannya untuk orang-orang yang saya cintai dan sayangi, yaitu :

1. Bapak dan ibu saya yang memberikan kasih sayang dan dukungan penuh atas setiap jalan yang saya tempuh. Terimakasih bapak dan ibu untuk semua pengorbanan kalian yang sangat luar biasa.
2. Terimakasih juga kepada kakak dan adik-adik saya, kakak Jefrianus Tozu Djuma, adik Maria Edela Senia Wula Djuma, Theresia Yuyuanita Djuma, dan adik bungsu saya Sisilia Chelita Wea Djuma yang selalu memberikan dukungan penuh serta hiburan untuk apa yang saya lakukan dan karena kalian sumber semangat saya selama ini.
3. Kepada Ibu Indra Yatini Buryadi, S.Kom,.M.Kom. yang telah menjadi pembimbing yang selalu baik dan sabar untuk memberikan bimbingan terhadap saya.
4. Kepada Bapak M. Agung Nugroho, S.Kom.,M.Kom. Sebagai Dosen wali yang telah memberikan bimbingan dan saran terbaiknya.
5. Kepada Aji Karuniadi Irawan yang berjasa dan sabar membimbing saya serta telah memberikan motivasi dan referensi dalam menyelesaikan karya ilmiah ini.
6. Kepada Siti Dahlia dan Muhamad Afrizal yang senantiasa mengerjakan dan menyelesaikan karya ilmiah bersama, melewati suka dan duka bersama.
7. Kepada Yanuarius Rafael Seda Pia yang telah mendampingi dan memberikan semangat dalam penyelesaian karya ilmiah ini.
8. Terimakasih juga kepada Keluarga Besar Lemon yang telah menjadi bagian suka dan duka selama ini. Pelajaran, semangat, motivasi, sahabat dan pengalaman yang belum tentu bisa saya dapatkan di tempat lain.
9. Kepada teman dan sahabat seperjuangan yang selalu memberikan perhatian, semangat dan dukungan.
10. *Last but not least, i wanna thank me, for believing in me, for doing all this hard work, for having no days off, for never quitting, for just being me at all the times.*

**MOTTO**

“Musuh yang Paling Berbahaya di atas Dunia ini adalah Penakut dan Bimbang. Teman yang Paling Setia, Hanyalah Keberanian dan Keyakinan yang Teguh”

*-Andrew Jackson-*

“Jangan pernah menyerah meskipun Anda merasa tidak memiliki satu pun bakat. Karena selama mau bekerja keras, bakat kalah. Dan Anda akan menang”

*-Pandji Pragiwaksono-*

“Sebagai pejuang, sebelum terlambat larilah dengan cepat.”

***-****Najwa Shihab-*

“Do And Done”

***-****Diwita Anggraini-*

**INTISARI**

*Microservice* adalah kumpulan proses independen dan kecil yang berkomunikasi antara satu dengan lainnya untuk membentuk aplikasi kompleks yang agnostik terhadap bahasa API apa pun. Servis-servis ini terdiri dari blok-blok kecil, terpisah, dan fokus pada tugas-tugas ringan untuk memfasilitasi metode modular dalam pembangunan sistem. Arsitektur bergaya microservice mulai menjadi standar dalam pembangunan sistem yang dinamis dan konstan berkembang.

*REST API* merupakan implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakkan protokol HTTP untuk pertukaran data. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan (scale) terutama dalam pertukaran dan komunikasi data.

Pada penelitian ini mencoba menerapkan teknologi *Microservice* pada aplikasi Input Nilai Praktikum Mahasiswa STMIK AKAKOM YOGYAKARTA dengan menggunakan REST API untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan (scale) terutama dalam pertukaran dan komunikasi data.

Penelitian ini menghasilkan sebuah sistem yang lebih flexible baik dalam pengembangan atau maintenance karena menggunakan teknologi *Microservice*yang memisahkan antara bagian *Frontend*dengan bagian*Backend* sehingga performa aplikasi menjadi lebih baik, cepat dan scale.

Kata Kunci : *Arsitektur, Microservice, Backend, Frontend, REST API.*

**KATA PENGANTAR**

Puji syukur kepada Tuhan Yang Maha Esa atas limpahan nikmat sehat-Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis dapat menyelesaikan Skripsi yang berjudul “IMPLEMENTASI ARSITEKTUR MICROSERVICE UNTUK INPUT NILAI PRAKTIKUM MAHASISWA STMIK AKAKOM YOGYAKARTA MENGGUNAKAN RESTFUL API”. Skripsi ini merupakan kewajiban formal untuk melengkapi sebagian persyaratan mencapai derajat Sarjana Komputer (S-1) pada Jurusan Teknik Informatika STMIK AKAKOM Yogyakarta.

Dengan segala keterbatasan dan kekurangan, penulis menyadari tidak akan mampu menyelesaikan Skripsi ini sendiri. Oleh karena itu penulis ingin menyampaikan terima kasih kepada :

1. Allah SWT atas segala nikmat dan kasih karuniaNya,
2. Kedua orang tua yang selalu memberi dukungan dan motivasi selama ini, Bapak Bambang Irawan dan Ibu Husniwati Irawan,
3. Bapak Ir. Totok Suprawoto, M.M., M.T., Ketua STMIK AKAKOM Yogyakarta,
4. Ibu Dini Fakta Sari, S.T., M.T., dan Ibu Femi Dwi Astuti, S.Kom., M.Cs. , Ketua dan Sekretaris Jurusan Teknik Informatika STMIK AKAKOM Yogyakarta,
5. Y. Yohakim Marwanta, S. Kom., M.Cs. selaku Dosen Pembimbing yang telah memberikan bimbingan dan saran terbaik pada karya ilmiah ini.
6. Bapak Wagito S.T., M.T. dan Bapak M. Agung Nugroho, S.Kom., M.Kom selaku Selaku dosen narasumber yang telah memberikan saran terbaik pada karya ilmiah ini.
7. Keluarga besar UKM I & K yang telah memberikan banyak pengalaman selama di STMIK AKAKOM Yogyakarta.
8. Teman-teman Mahasiswa STMIK AKAKOM Yogyakarta yang telah memberikan saran untuk terselesaikannya skripsi ini.

Menyadari akan kekurangan dan kesalahan penulis dalam skripsi ini karena keterbatasan ilmu penulis, maka penulis mengharapkan masukan berupa saran yang membangun dari semua pihak. Semoga skripsi ini dapat memberikan manfaat dan berguna bagi kita semua.

Yogyakarta, 2020

Aji Karuniadi Irawan

# **DAFTAR ISI**

HALAMAN JUDUL i

HALAMAN PERSETUJUAN ii

HALAMAN PENGESAHAN iii

HALAMAN PERSEMBAHAN iv

MOTTO vi

INTISARI vii

KATA PENGANTAR viii

DAFTAR ISI x

DAFTAR GAMBAR xii

DAFTAR TABLE xiv

BAB I\_PENDAHULUAN 1

* 1. Latar Belakang 1
  2. Rumusan Masalah 3
  3. Ruang Lingkup 3
  4. Tujuan Penelitian 4
  5. Manfaat Penelitian 4
  6. Sistem Penulisan 4

BAB II\_TINJAUAN PUSTAKA DAN DASAR TEORI 6

* 1. Tinjauan Pustaka 6
  2. Dasar Teori 9
     1. Microservices 9
     2. Restful Api 10
     3. Laravel Lumen 11
     4. Vuetify 12
     5. Mysql 13
     6. Gitlab CI/CD 14
     7. Heroku 14

BAB III\_METODE PENELITIAN 16

* 1. Bahan/Data 16
  2. Peralatan 16
     1. Perangkat Keras (Hardware) 16
     2. Perangkat Lunak (Software) 17
  3. Prosedur Dan Pengumpulan Data 17
  4. Analisis Dan Perancangan Sistem 18
     1. Arsitektur Sistem 18
     2. DAD (Diadram Alir Sistem) 19
        1. Dad Level 0 19
        2. Dad Level 1 20
     3. Perancangan Basis Data 21
        1. Skema Basis Data 21
        2. Rancangan Tabel 22
     4. Perancangan User Interface 25

BAB IV\_IMPLEMENTASI DAN PEMBAHASAN 28

1. Implementasi dan Uji Coba Sistem 28
2. Rest Api Admin Login 29
3. Rest Api Admin Register 30
4. Rest Api GET Nilai Berdasarkan Matakuliah, Kelas dan

Username 31

1. Rest Api POST Nilai 32
2. Rest Api PUT Nilai 34
3. Rest Api DELETE Nilai 35
4. Pembahasan Sistem 36
5. Tampilan Login 36
6. Tampilan Register 37
7. Tampilan Dashboard Admin 37
8. Tampilan Dashboard Pengguna 38
9. Tampilan Tambah Nilai 39
10. Tampilan Edit Nilai 40
11. Tampilan Laporan Per Hari 41
12. Tampilan Laporan Per Mahasiswa 44

BAB V\_PENUTUP 46

1. Kesimpulan 46
2. Saran 46

DAFTAR PUSTAKA 47

# **DAFTAR GAMBAR**

Gambar 2.1 Microservices Architecture 10

Gambar 2.2 Arsitektur RESTful API 11

Gambar 2.3 Perbedaan Requests Per Second Micro Framework 12

Gambar 2.4 Arsitektur Gitlab CI/CD 14

Gambar 3.1 Arsitektur Sistem 18

Gambar 3.2 DAD Level 0 20

Gambar 3.3 DAD Level 1 21

Gambar 3.4 Skema Basis Data 22

Gambar 3.5 Halaman Login Admin 25

Gambar 3.6 Halaman Register Admin 26

Gambar 3.7 Halaman Tambah Nilai 26

Gambar 3.8 Halaman Utama Pengguna 27

Gambar 4.1 Rest Api Admin Login 28

Gambar 4.2 Rest Api Admin Register 29

Gambar 4.3Rest Api GET Nilai Berdasarkan Matakuliah, Kelas dan

Username 31

Gambar 4.4 Aliran Proses Fungsi  GET Nilai Berdasarkan Mata Kuliah,

Kelas dan Username 32

Gambar 4.5 Rest Api POST Nilai 33

Gambar 4.6Aliran Proses Fungsi POST Nilai 34

Gambar 4.7 Rest Api PUT Nilai 34

Gambar 4.8Aliran Proses Fungsi PUT Nilai 36

Gambar 4.9 Rest Api DELETE Nilai 36

Gambar 4.10 Aliran Proses Fungsi DELETE Nilai 37

Gambar 4.11 Tampilan Login 37

Gambar 4.12 Tampilan return data login 38

Gambar 4.13 Tampilan Register 38

Gambar 4.14 Tampilan Dashboard Admin 39

Gambar 4.15 Tampilan Dashboard Pengguna 40

Gambar 4.16 Tampilan Tambah Nilai 40

Gambar 4.17 Tampilan proses save data 41

Gambar 4.18 Tampilan Edit Nilai 42

Gambar 4.19 Tampilan Laporan Per Hari 43

Gambar 4.20 Hasil Filter Data Nilai 44

Gambar 4.21Download File Nilai Praktikum 45

Gambar 4.22 Tampilan Laporan Per Mahasiswa 46

# **DAFTAR TABEL**

Tabel 2.1 Tinjauan Pustaka 7

Tabel 3.1 Tabel Admin 22

Tabel 3.2 Tabel Dosen 23

Tabel 3.3 Tabel Matkul 23

Tabel 3.4 Tabel Mahasiswa 24

Tabel 3.5 Tabel Nilai 24

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Praktikum menurut Kamus Besar Bahasa Indonesia (KBBI) adalah bagian dari pengajaran yang bertujuan agar mahasiswa mendapat kesempatan untuk menguji dan mengaplikasikan dalam keadaan nyata apa yang diperoleh dalam teori. Praktikum di STMIK AKAKOM Yogyakarta bersifat wajib, maka STMIK AKAKOM Yogyakarta menerapkan sistem penilaian berupa nilai pretest dan nilai laporan sebagai salah satu indikator penentuan nilai akhir mahasiswa.

Sulitnya akses yang dimiliki oleh Mahasiswa, Dosen dan Penjaga LAB untuk melihat hasil perkembangan nilai praktikum menjadi masalah utama di STMIK AKAKOM Yogyakarta, karena belum tersedianya media yang menampilkan nilai praktikum mahasiswa di setiap pertemuannya ditambah dengan adanya Asisten Dosen (ASDOS) yang masih melakukan input nilai menggunakan Microsoft Excel, yang membuat perbedaan format penilaian untuk setiap ASDOS. Sehingga laporan yang diserahkan ke penjaga lab menjadi berbeda-beda.

Secara tradisional, aplikasi web yang dibangun dengan menggunakan pendekatan *monolitik* dimana seluruh aplikasi dibangun, dirancang, dikerahkan dan dipelihara didalam satu unit. Ketika bekerja dengan aplikasi *monolitik* berbagai masalah dapat timbul dari waktu ke waktu. Dalam kasus aplikasi web, struktur aplikasi *monolitik* biasanya terdiri dari *clien-side*, *servers-side* dan database. Arsitektur *monolitik* ini tentunya punya kelemahan-kelemahan yang solusinya bisa dialamatkan oleh *Microservice*.

*Microservice* adalah kumpulan proses independen dan kecil yang berkomunikasi antara satu dengan lainnya untuk membentuk aplikasi kompleks yang agnostik terhadap bahasa API apa pun. Servis-servis ini terdiri dari blok-blok kecil, terpisah, dan fokus pada tugas-tugas ringan untuk memfasilitasi metode modular dalam pembangunan sistem. Arsitektur bergaya microservice mulai menjadi standar dalam pembangunan sistem yang dinamis dan konstan berkembang.

Salah satu teknologi yang digunakan yaitu berkaitan dengan framework yang digunakan. Laravel lumen adalah Micro *Framework* yang diciptakan pengembang Laravel untuk mengakomodasi kebutuhan developer yang ingin membuat aplikasi dalam skala lebih kecil dari Laravel. Karena banyak library yang dihilangkan dalam bundle source code, Lumen bisa dijadikan *framework* untuk membuat *REST API.*

*REST API* merupakan implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakkan protokol HTTP untuk pertukaran data. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan (scale) terutama dalam pertukaran dan komunikasi data.

Berdasarkan permasalahan diatas, aplikasi ini dibangun sebagai media input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta yang dapat digunakan oleh ASDOS untuk membuat laporan nilai praktikum dengan format yang sama sehingga Mahasiswa, Dosen dan Penjaga LAB dapat mengakses nilai praktikum di setiap pertemuannya, dengan memanfaatkan Arsitektur *Microservice* menggunakan *RESTful API*, dimana Arsitektur *Microservice* membuat aplikasi menjadi lebih scalable, secure dan reliable, sehingga dalam proses transfer data menjadi lebih cepat dan aman.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang permasalahan diatas, maka rumusan masalahnya adalah bagaimana menghasilkan aplikasi sebagai media input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta berbasis web oleh ASDOS dengan format yang sama sehingga Mahasiswa, Dosen dan Penjaga LAB dapat mengakses nilai praktikum di setiap pertemuannya, dengan mengimplementasikan Arsitektur *Microservice* sebagai layanan servermenggunakan *RESTful API* sebagai metode pertukaran data dari *backend* ke *frontend*.

* 1. **Ruang Lingkup**

Adapun ruang lingkup masalah yang diangkat adalah sebagai berikut :

1. Aplikasi ini berbasis web
2. Membuat *REST API* untuk input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta menggunakan konsep arsitektur microservice.
3. Data yang akan dibuat oleh backend berupa data inputan nilai pretest dan nilai laporan.
4. Bagian frontend menggunakan framework javascript yaitu vuetify.
5. Untuk integrasi API atau untuk menghubungkan antara backend dengan frontend menggunakan Axios.
6. Bagian backend menggunakan micro-framework PHP Hypertext Preprocessor (PHP) yaitu laravel lumen untuk membuat API.
7. Pengujian *REST API* menggunakan Insomnia.
8. Untuk database menggunakan Mysql.
9. Gitlab CI/CD untuk proses continuous integration dan continuous delivery.
   1. **Tujuan Penelitian**

Tujuan yang ingin diperoleh dari pembuatan aplikasi ini adalah menghasilkan sebuah aplikasi sebagai media input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta, dengan memanfaatkan Arsitektur *Microservice* sebagai layanan serverdengan menggunakan *RESTful API* sebagai metode pertukaran data dari *backend* ke *frontend*sehingga performance aplikasi menjadi lebih cepat karena proses logika program terpisah dengan proses loading untuk tampilan*.*

* 1. **Manfaat Penelitian**

Manfaat dari pembuatan aplikasi ini adalah sebagai media input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta untuk ASDOS dengan format yang sama sehingga Mahasiswa, Dosen dan Penjaga LAB dapat mengakses nilai praktikum di setiap pertemuannya dengan mudah.

* 1. **Sistematika Penulisan**

Berikut merupakan sistematika penulisan skripsi yang akan dibuat :

BAB I. PENDAHULUAN

Pada bab ini berisi tentang penjelasan mengenai latar belakang masalah, rumusan masalah, ruang lingkup, tujuan penelitian, manfaat penalitian dan sistematika penulisan.

BAB II. TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini berisi tentang pembahasan sumber pustaka yang digunakan sebagai pedoman perancangan penelitian dan penjelasan yang berhubungan dengan penelitian yang digunakan sebagai landasan dalam penelitian.

BAB III. METODE PENELITIAN

Pada bab ini berisi tentang pembahasan analisis kebutuhan, bahan/data, peralatan dan perancangan sistem yang akan digunakan.

BAB IV. IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini menguraikan tentang pembuatan aplikasi yang merupakan implementasi dari hasil analisa dan perancangan, pengujian sistem dan kesimpulan.

BAB V. PENUTUP

Pada bab ini berisi kesimpulan yang dihasilkan dari pembahasan penerapan sistem dan saran-saran guna pengembangan sistem yang telah dibuat.

**BAB II**

**TINJAUAN PUSTAKA DAN DASAR TEORI**

* 1. **Tinjauan Pustaka**

Heri Purnama (2016), telah melakukan penelitian tentang Aplikasi Pengolahan Skripsi Di STMIK AKAKOM YOGYAKARTA Menggunakan Arsitektur Microservice Dengan Node.js. Aplikasi dibuat bertujuan untuk mengecek duplikasi skripsi dengan berdasarkan judul dari skripsi. Dibuat dengan Bahasa pemrograman JavaScript pada Node.js, menggunakan database MongoDB yang mengimplementasikan arsitektur Microservice.

Ahmad Qomaruddin (2018), telah melakukan penelitian tentang Implementasi Arsitektur Microservice menggunakan RESTful API untuk portal akademik PP AL-MUNAWWIR. Aplikasi ini bertujuan untuk membangun aplikasi portal akademik untuk Pondok Pesantren Al-Munawwir, dimana service-service dari aplikasi dibangun secara terpisah berdasarkan fungsionalitasnya masing-masing dengan menggunakan RESTful API sebagai komunikasi antara Frontend dengan Endpoint. Dibuat dengan Bahasa pemrograman Golang, menggunakan database CockroachDB.

Andre Putra (2020), telah melakukan penelitian tentang RESTful API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices untuk ditampilkan di frontend. Aplikasi bertujuan untuk membangun sebuah sistem yang lebih dinamis dan berkembang dengan memanfaatkan teknologi Arsitektur *Microservice*. Dibuat dengan Bahasa pemrograman Python, menggunakan database PostgreSQL.

Pembahasan yang dibuat kali ini tentang “Implementasi Arsitektur Microservice Untuk Input Nilai Praktikum Mahasiswa STMIK AKAKOM Yogyakarta Menggunakan RESTFUL API”. Penelitian ini bertujuan untuk menghasilkan aplikasi untuk media input nilai praktikum mahasiswa STMIK AKAKOM Yogyakarta dan memberikan informasi nilai hasil praktikum untuk Mahasiswa, Dosen dan Penjaga LAB. Dibuat dengan Bahasa pemrograman PHP, menggunakan database MySQL.

Tabel tinjauan pustaka merupakan tabel yang dibuat untuk mendefinisikan penelitian yang sebelumnya hampir sama dilakukan dengan penelitian yang diajukan saat ini, adapun perbandingan yang menjadi tabel tinjauan pustaka penelitian yakni :

Table 2.1 Tinjauan Pustaka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Penulis | Judul | Bahasa Pemrograman | Teknologi | Database |
| Heri Purnama  (2016) | Pengolahan Skripsi Menggunakan Arsitektur Microservice Dengan Node.js | Javascript | Microservice, Node.js, Private Cloud (Docker) | NoSQL-MongoDB |
| Ahmad Qomaruddin  (2018) | Implementasi Arsitektur Microservice Menggunakan RESTful API Pada Portal Akademik | Go | Microservice, RESTful API | CockroachDB |
| Andre Putra  (2020) | RESTFUL API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices. | Python, Javascript | Microservice, RESTful API | PostgreSQL |
| Aji Karuniadi Irawan  (2020) | Implementasi Arsitektur Microservice Untuk Input Nilai Praktikum Mahasiswa STMIK AKAKOM Yogyakarta Menggunakan RESTFUL API | PHP, JavaScript | Microservice,  Vuetify, Laravel Lumen, RESTful API | MySQL |

* 1. **Dasar Teori**
     1. ***Microservices***

Microservices adalah sebuah pendekatan untuk mengembangkan aplikasi dengan rangkaian service-service yang kecil, yang mana setiap service berjalan pada prosesnya sendiri-sendiri. Setiap service dapat berkomunikasi dengan mekanisme yang ringan.

Tiap-tiap service yang dibuat harus mengenkapsulasi data dengan logika bisnis yang beroperasi pada data itu sendiri, dan hanya dapat diakses melalui published service interface. Tidak ada database yang dapat diakses secara langsung dari luar service dan tidak ada data yang disharing antara setiap service.

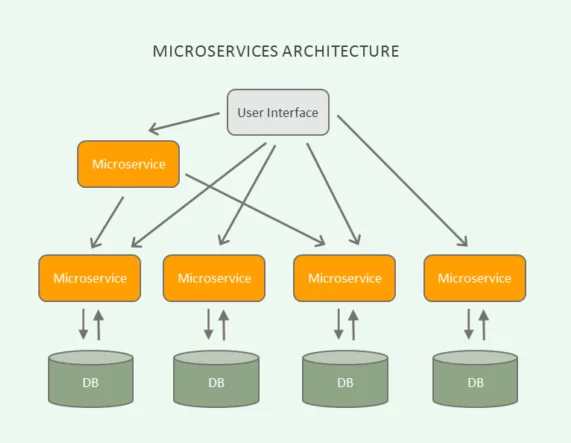
Dengan begitu, setiap service yang dibangun harus memiliki landasan domainnya sendiri-sendiri dan process sharing data domain yang satu dengan yang lainnya hanya dapat dilakukan melalui published service interface.

Microservices memperkuat struktur modular yang sangat penting bagi tim yang sangat besar. Menurut Martin Fowler ini adalah key benefit yang juga aneh jika dikatakan kelebihan, karena tidak ada alasan apapun mengapa microservices memiliki struktur modular yang lebih kuat daripada monolithic.

Dalam arsitektur monolithic pada umumnya, sangatlah mudah bagi developer untuk melewati batas. Umumnya digunakan untuk mencari jalan pintas dalam mengimplementasikan fitur dengan lebih cepat. Akan tetapi berakhir pada merusak struktur modular yang berimplikasi pada penurunan produktivitas Tim.

Service yang sederhana lebih mudah di deploy dan digunakan. Karena mereka berdiri sendiri, kecil kemungkinan kegagalan sistem terjadi saat salah satu service mengalami kesalahan.

Dengan microservices, kita dapat mencampur dan menggunakan beragam bahasa pemrograman, framework, dan teknologi penyimpanan database yang digunakan.(Refractory, 2017)



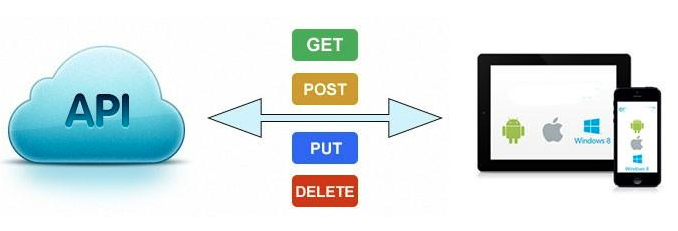
**Gambar 2.1** *Microservices Architecture*

* + 1. **RESTful API**

RESTful API didasarkan pada teknologi state transfer (representational state transfer / REST), gaya arsitektur dan pendekatan komunikasi yang sering digunakan dalam pengembangan layanan web.

Meskipun REST dapat digunakan di hampir semua protokol, tapi biasanya memanfaatkan HTTP ketika digunakan untuk Web API. Hal ini membantu pengembang web tidak perlu menginstal library atau perangkat lunak tambahan untuk memanfaatkan desain REST API. Design REST API pertama kali diperkenalkan oleh Dr. Roy Fielding dalam disertasi doktor tahun 2000-nya. REST API terkenal karena fleksibilitasnya yang luar biasa. Data tidak terikat dengan metode dan sumber daya, REST memiliki kemampuan untuk menangani beberapa jenis panggilan, mengembalikan format data yang berbeda dan bahkan mengubah secara struktural tentunya dengan implementasi yang benar.

REST yang digunakan oleh browser dapat dianggap sebagai bahasa internet. Dengan meningkatnya penggunaan cloud, API muncul untuk mengekspos layanan web. REST adalah pilihan logis untuk membangun API yang memungkinkan pengguna untuk terhubung dan berinteraksi dengan layanan cloud. API telah banyak digunakan oleh situs-situs seperti Amazon, Google, LinkedIn dan Twitter. (Yudana, 2019)



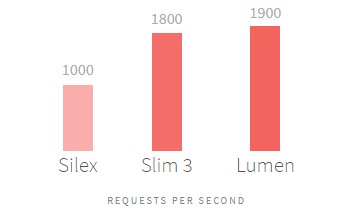
**Gambar 2.2** A*rsitektur RESTful API*

* + 1. **Laravel Lumen**

Lumen adalah Micro Framework yang diciptakan pengembang Laravel untuk mengakomodasi kebutuhan developer yang ingin membuat aplikasi dalam skala lebih kecil dari Laravel. Karena banyak library yang dihilangkan dalam bundle source code, Lumen bisa dijadikan framework untuk membuat REST API.

Ada beberapa Micro Framework lain yang dikhususkan untuk membuat REST API pada PHP seperti Slim Framework, Silex dll tentu dengan keistimewaannya masing-masing, namun yang menarik adalah Lumen dalam lamannya mampu menghandle hingga 1900 request per second! Kalah tipis dari Slim Framework yang katanya hanya dapat menghandle 1800 request per second.

Perbedaan lain yaitu Lumen menggunakan PHP versi ke 7 sebagai syarat penggunaannya, sedangkan Slim masih bisa digunakan PHP versi 5 ke atas. (Rizky Fauzi, 2017)



**Gambar 2.3** Perbedaan Requests Per Second Micro Framework

* + 1. **Vuetify**

Vuetify.js adalah framework desain komponen material untuk UI/UX yang berasosiasi langsung dengan Vue.js. Vue.js adalah framework Javascript yang progresif dan bersifat open-source dan sangat berguna untuk membangun UI dan aplikasi berhalaman tunggal (*single-page application*).

Desain vuetify sangat mirip dengan yang dimiliki Google dan sangat sulit di-*customize* sehingga banyak para desainer frontend atau UI/UX yang main di plain CSS.(Akidna Hanavia, 2019)

* + 1. **MySQL**

Menurut Arief (2011d:152) **“**MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya”.

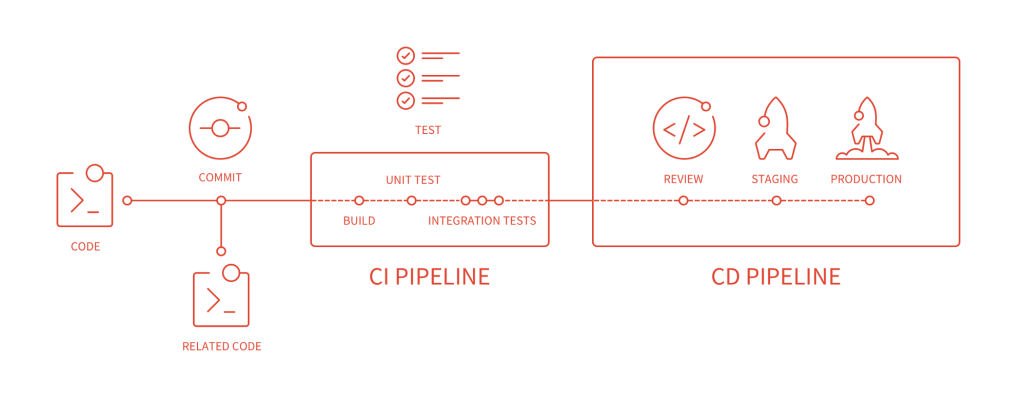
Menurut Adi Nugroho (2011) MySQL (My Structured Query Language) adalah: “ Suatu sistem basis data relation atau Relational Database management System (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi multi user (banyak pengguna). MySQL didistribusikan gratis dibawah lisensi GPL (General Public License). Dimana setiap program bebas menggunakan MySQL namun tidak bisa dijadikan produk turunan yang dijadikan closed source atau komersial”.

MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman script PHP.

* + 1. **Gitlab CI/CD**

Gitlab CI (*Continuous Integration*) dikenal dengan *Test* dan *Build* yang artinya setiap kali tim *push* kode ke dalam aplikasi, kita hanya akan melakukan *Test* dan *Build* dengan cara manual maupun otomatis.

Gitlab CD ([*Continuous Delivery*](https://continuousdelivery.com/)) merupakan pendekatan rekayasa perangkat lunak yaitu **c*ontinuous integration****,* ***automated testing****,* dan **a*utomated deployment*** yang memungkinkan perangkat lunak dikembangkan dan digunakan dengan cepat. (David Hs, 2017)



**Gambar 2.4** *Arsitektur Gitlab CI/CD*

* + 1. **Heroku**

Heroku adalah sebuah cloud platform yang menjalankan bahasa pemrograman tertentu, Heroku mendukung bahasa pemrograman seperti Ruby, Node.js, Python, Java, PHP, dan lain-lain.

Heroku termasuk ke dalam kriteria *Platform As A Service* (PaaS), sehingga bagi anda yang ingin melakukan deploy aplikasi ke heroku cukup hanya dengan melakukan konfigurasi aplikasi yang ingin di deploy dan menyediakan platform yang memungkinkan pelanggan untuk mengembangkan, menjalankan, dan mengelola aplikasi tanpa kompleksitas membangun dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan peluncuran aplikasi. (Efendi Hariyadi, 2017)

**BAB III**

**METODE PENELITIAN**

Metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:

* 1. **Bahan/Data**

Data yang disiapkan dalam penelitian ini yaitu :

1. Data Mahasiswa yang berisi informasi tentang mahasiswa yang mengambil mata kuliah praktikum.
2. Data Dosen yang berisi tentang informasi dosen pengampu mata kuliah praktikum.
3. Data Matakuliah yang berisi informasi tentang mata kuliah apa saja yang termasuk kategori praktikum.
4. Data ASDOS yang berisi informasi tentang Asisten Dosen pembantu mata kuliah praktikum.
   1. **Peralatan**

Peralatan yang digunakan pada penelitian ini adalah perangkat keras dan perangkat lunak

* + 1. **Perangkat Keras (*Hardware*)**

Perangkat keras yang digunakan pada penelitian ini adalah :

1. Laptop dengan spesifikasi :
   1. Processor Intel Core 2 Duo.
   2. Memory 4 GB.
   3. Hard Disk 500 GB.
   4. VGA NVIDIA GeForce 320M 256 MB.
      1. **Perangkat Lunak (*Software*)**

Perangkat Lunak yang digunakan pada penelitian ini adalah :

1. Mac Os X Yosemite digunakan sebagai Sistem Operasi.
2. PHP 7.2.x sebagai bahasa pemrograman yang digunakan untuk pengembangan bagian *Backend.*
3. Laravel Lumen digunakan sebagai framework untuk membangun RESTful API.
4. Vuetify digunakan sebagai framework dalam pengembangan bagian *Frontend*.
5. Axios digunakan untuk integrasi API atau menghubungkan *Backend* dengan *Frontend*.
6. MySQL digunakan sebagai manajemen basis data.
7. Heroku digunakan untuk deploy API agar bisa digunakan secara publik.
8. Insomnia digunakan sebagai tool untuk testing API.
9. Vs Code digunakan sebagai Text Editor.

Google Chrome digunakan sebagai web browser untuk menampilkan hasil program.

* 1. **Prosedur dan Pengumpulan Data**

Metode yang digunakan dalam proses pengumpulan data sebagai berikut:

1. Studi Literatur

Studi Literatur dilakukan dengan mencari bahan materi berhubungan dengan REST API. Guna mempermudah proses implementasi sistem. Pencarian materi dilakukan melalui pencarian di buku panduan, dan internet.

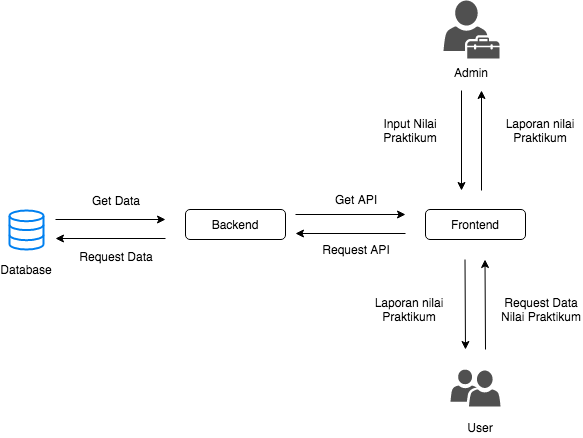
1. Dokumen

Pengumpulan data melalui file dokumen pada lab terpadu STMIK AKAKOM YOGYAKARTA untuk mendapatkan data mahasiswa, dosen dan mata kuliah praktikum.

* 1. **Analisis dan Perancangan Sistem**

Pada sistem yang dibuat ini terdapat beberapa rancangan yang akan digunakan sebagai pedoman implementasi sistem ke dalam suatu program aplikasi yang akan dibuat.

* + 1. **Arsitektur Sistem**



**Gambar 3.1** *Arsitektur Sistem*

Dari Arsitektur di atas artinya user melakukan akses melalui web, maka frontend nantinya akan melakukan proses request API ke backend dan dari backend melakukan request data ke database, dimana admin melakukan inputan data melalui frontend, maka nantinya frontend akan melakukan request API ke backend dan data akan dikirim ke dalam database.

* + 1. **DAD (Diagram Alir Data)**

Perancangan pada aplikasi Input Nilai Mahasiswa STMIK AKAKOM YOGYAKARTA ini menggunakan perancangan sistem DAD.

### DAD Level 0

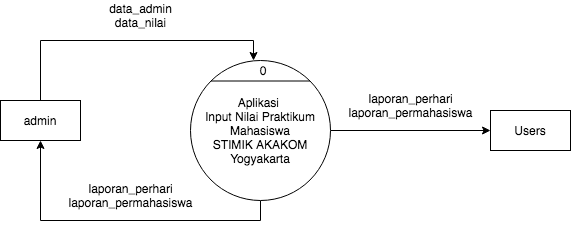
Diagram konteks adalah bagian dari data *flow* diagram yang berfungsi memetakan model lingkungan, yang di presentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Pada gambar 3.2 menjelaskan bahwa terdapat 2 entitas yaitu :

* 1. Admin

Pada entitas Admin akan menginputkan data nilai dan data admin yang nantinya akan di proses oleh sistem, dimana nantinya akan mendapatkan keluaran berupa laporan perhari dan permahasiswa yang sebelumnya di proses oleh sistem.

* 1. Users

Pada entitas Users akan mendapatkan keluaran laporan perhari dan permahasiswa dari sistem yang sebelumnya di proses oleh sistem dari data nilai yang di inputkan oleh entitas Admin.

****

**Gambar 3.2** *DAD Level 0*

### DAD Level 1

Pada gambar 3.3 menjelaskan bahwa ada 3 proses yang ada pada sistem, yaitu sebagai berikut :

* + - 1. Input Data Admin

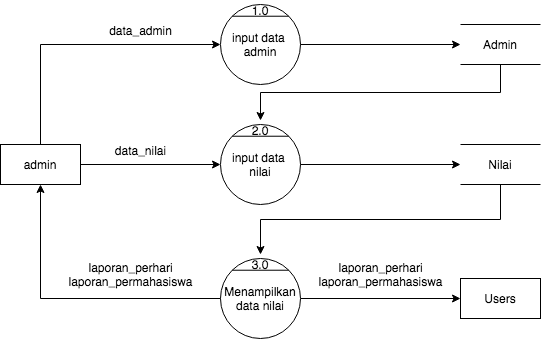
Pada proses input data admin merupakan proses login untuk admin dan melakukan penyimpanan data.

* + - 1. Input Data Nilai

Pada proses input data nilai merupakan proses penginputan nilai yang dimana proses ini akan menerima data admin untuk bisa melakuakan proses penginputan dan penyimpanan data.

* + - 1. Menampilkan Data Nilai

Pada proses ini merupakan proses pengambilan data nilai yang akan di tampilkan ke users berupa data laporan.

****

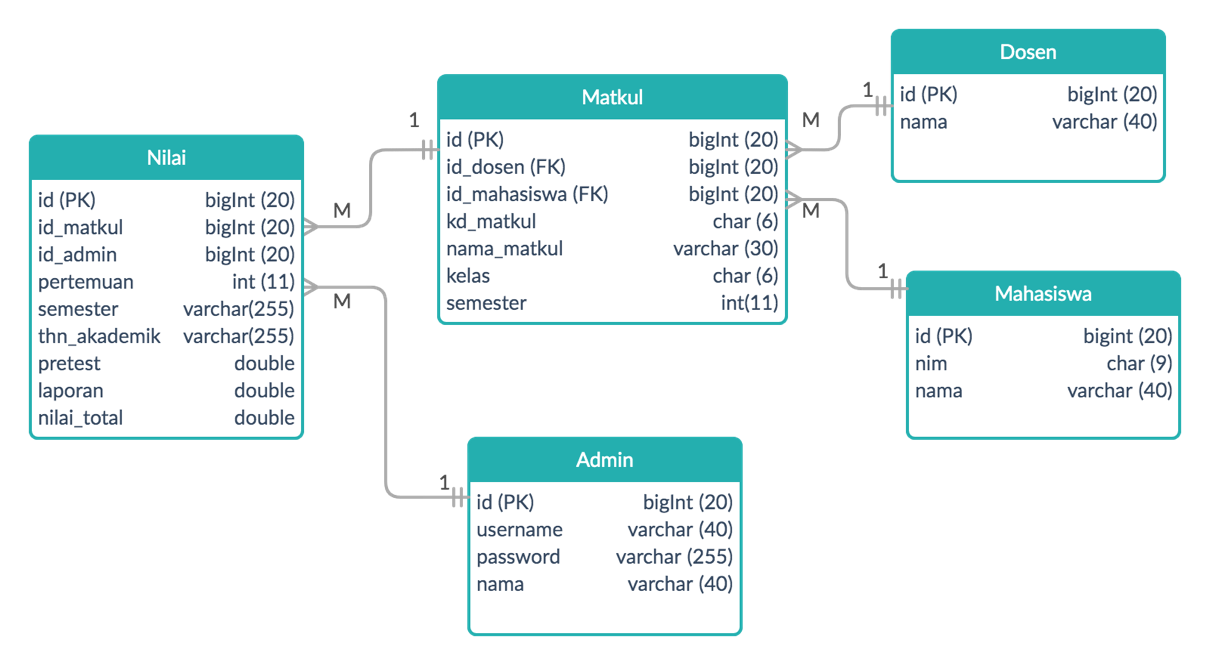
**Gambar 3.3** *DAD Level 1*

* + 1. **Perancangan Basis Data**

Pada tahap ini akan dilakukan proses transformasi dari data manual kedalam bentuk basis data yang akan dibagi menjadi berbagai table, yang di dalamnya terdapat *field-field*. Untuk lebih jelasnya akan di lakukan beberapa rancangan *database*, tabel-tabel *field* serta relasi antar tabel yang dibutuhkan dalam membuat sistem ini.

* + - 1. Skema Basis Data

Pada model data relasional hubungan antar tabel direlasikan dengan kunci utama (*primary key*) dari masing-masing tabel. Perancangan basis data pada sistem ini dapat dilihat pada gambar 3.4

****

**Gambar 3.4** *Skema Basis Data*

* + - 1. Rancangan Tabel

Tabel Admin

Tabel admin digunakan untuk menyimpan data admin. Adapun atributnya dapat dilihat pada tabel 3.1.

**Tabel 3.1 Tabel Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type Data** | **Panjang** | **Keterangan** |
| 1 | id | bigint | 11 | sebagai kunci utama table admin |
| 2 | username | varchar | 40 | username untuk login admin |
| 3 | password | varchar | 40 | password untuk login admin |
| 4 | nama | varchar | 40 | nama untuk admin |

Tabel Dosen

Tabel Dosen digunakan untuk menyimpan data dosen. Adapun atributnya dapat dilihat pada tabel 3.2.

**Tabel 3.2 Tabel Dosen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type Data** | **Panjang** | **Keterangan** |
| 1 | id | bigint | 11 | sebagai kunci utama tabel dosen |
| 2 | nama | varchar | 40 | nama untuk dosen |

1. Tabel Matkul

Tabel Matkul digunakan untuk menyimpan data matkul. Adapun atributnya dapat dilihat pada tabel 3.3.

**Tabel 3.3 Tabel Matkul**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type Data** | **Panjang** | **Keterangan** |
| 1 | id | bigint | 11 | sebagai kunci utama tabel matkul |
| 2 | kd\_matkul | char | 6 | kode matkul untuk mata kuliah |
| 3 | nama\_matkul | varchar | 40 | nama untuk mata kuliah |
| 4 | kelas | char | 6 | kelas untuk mata kuliah |
| 5 | id\_mahasiswa | bigint | 11 | sebagai kunci tamu dari tabel mahasiswa |
| 6 | id\_dosen | bigint | 11 | sebagai kunci tamu dari tabel dosen |
| 7 | Semester | Int | 11 | Semester untuk mata kuliah |

1. Tabel Mahasiswa

Tabel Mahasiswa digunakan untuk menyimpan data mahasiswa. Adapun atributnya dapat dilihat pada tabel 3.4.

**Tabel 3.4 Tabel Mahasiswa**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type Data** | **Panjang** | **Keterangan** |
| 1 | id | bigint | 11 | sebagai kunci utama tabel mahasiswa |
| 2 | nim | char | 9 | nim untuk mahasiswa |
| 3 | nama | varchar | 40 | nama untuk mahasiswa |

1. Tabel Nilai

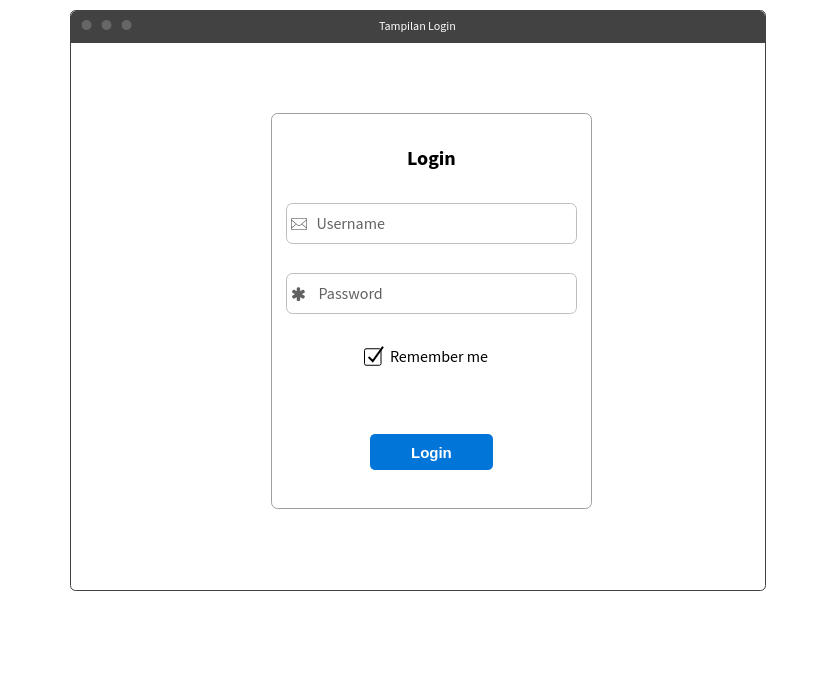
Tabel Nilai digunakan untuk menyimpan data nilai. Adapun atributnya dapat dilihat pada tabel 3.5.

**Tabel 3.5 Tabel Nilai**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Field** | **Type Data** | **Panjang** | **Keterangan** |
| 1 | id | bigint | 11 | sebagai kunci utama tabel nilai |
| 2 | id\_admin | bigint | 11 | sebagai kunci tamu dari tabel admin |
| 3 | id\_matkul | bigint | 11 | sebagai kunci tamu dari tabel matkul |
| 4 | pertemuan | int | 11 | untuk mengetahui praktikum pertemuan ke berapa |
| 5 | pretest | double | 11 | nilai untuk pretest |
| 6 | laporan | double | 11 | nilai untuk laporan |
| 7 | nilai\_total | double | 11 | nilai total mahasiswa |
| 8 | semester | varchar | 255 | Semester genap/ganjil |
| 9 | thn\_akademik | varchar | 255 | Tahun akademik praktikum |

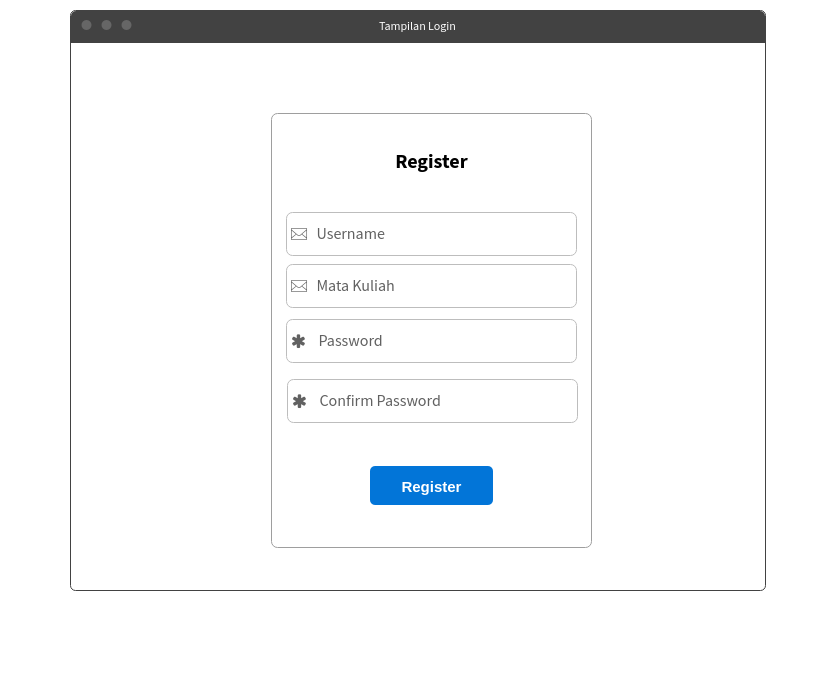
* + 1. **Perancangan User Interface**

1. Halaman Login Admin



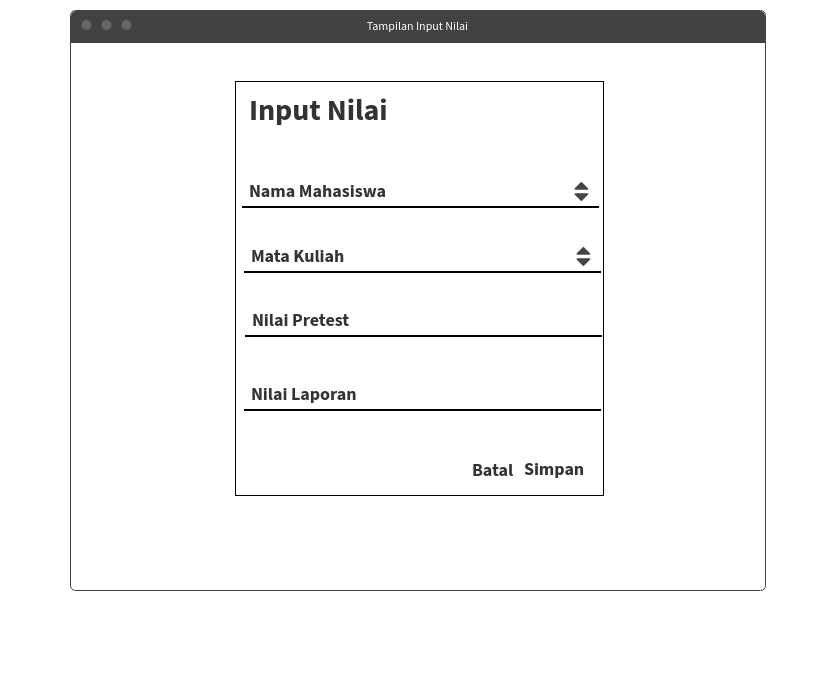
**Gambar 3.5** *Halaman Login Admin*

1. Halaman Register Admin



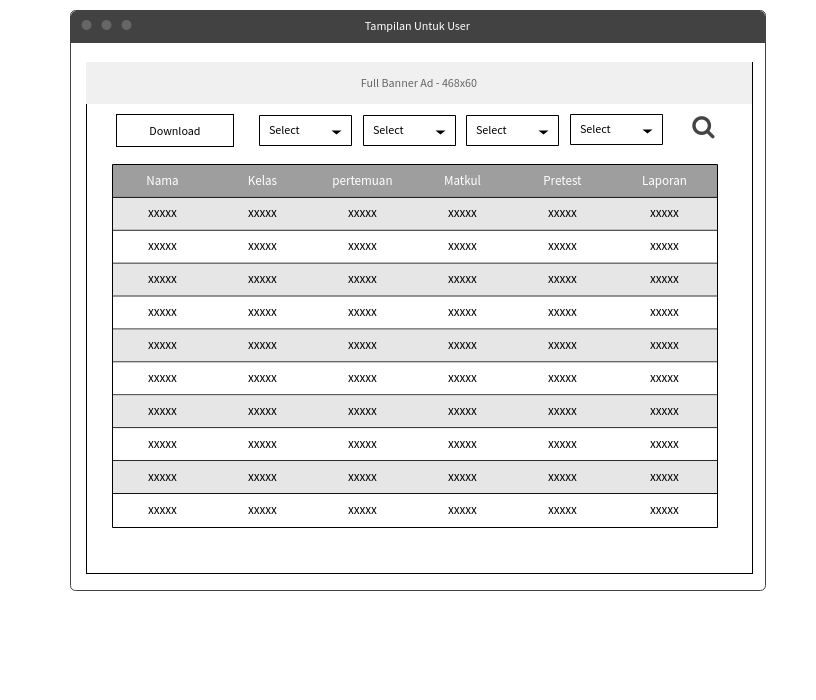
**Gambar 3.6** *Halaman Register Admin*

1. Halaman Tambah Nilai



**Gambar 3.7** *Halaman Tambah Nilai*

1. Halaman Utama Pengguna



**Gambar 3.8** *Halaman Utama Pengguna*

**BAB IV**

**IMPLEMENTASI DAN PEMBAHASAN**

1. **Implementasi dan Uji Coba Sistem**

Berdasarkan analisis dan perancangan sistem yang telah dilakukan, maka penulisan kode program dapat dilakukan, sehingga dapat diharapkan program yang dihasilkan nantinya mampu memenuhi kebutuhan yang diharapkan secara optimal. Selanjutnya pada bab ini akan dibahas bagaimana program tersebut dibuat dan tampilan-tampilan yang dihasilkan dari rancangan yang telah dibuat.

1. **Rest Api Admin Login**

|  |
| --- |
| /\*\* fungsi login\*/  public function login(Request $request)     {           //validate incoming request         $this->validate($request, [             'username' => 'required|string',             'password' => 'required|string',         ]);         $credentials = $request->only(['username', 'password']);         if (! $token = Auth::attempt($credentials)) {             return response()->json(['message' => 'Akun Belum Terdaftar'], 401);         }         return $this->respondWithToken($token);     }    /\*\* route url untuk frontend \*/   $router->post('login', 'AuthController@login');  /\*\* akses rest api using axios \*/    const response = await axios.post(basedurl +"/auth/login", {                     username:this.username,                     password:this.password,                 })                 this.result = response.data;                 Cookies.set('user', this.username)                 Cookies.set('token', this.result.token)                     if(this.result.token){                         this.$router.push('/admin/dashboard')                     } |

**Gambar 4.1***Rest Api Admin Login*

Pada gambar 4.1 merupakan potongan program yang digunakan untuk validasi data admin ketika login, dengan memanfaatkan library JWT (Json Web Token) dimana username dan password akan di convert ke dalam bentuk token, sehingga proses request data user dari *backend* ke *frontend* menjadi lebih aman. Karena menggunakan token sebagai autentifikasi dimana token tersebut disimpan di dalam cookies browser yang expired dalam 1 jam, setelah expired user akan di redirect ke halaman login sehingga data user akan lebih aman dari proses *injection* data sql.

1. **Rest Api Admin Register**

|  |
| --- |
| /\*\* fungsi get nilai \*/  public function register(Request $request)     {         //validate incoming request         $this->validate($request, [             'nama' => 'required|string',             'username' => 'required|unique:admin',             'password' => 'required|confirmed',         ]);         try {             $admin = new Admin;             $admin->nama = $request->input('nama');             $admin->username = $request->input('username');             $plainPassword = $request->input('password');             $admin->password = app('hash')->make($plainPassword);             $admin->save();             //return successful response             return response()->json(['admin' => $admin, 'message' => 'CREATED'], 201);         } catch (\Exception $e) {             //return error message             return response()->json(['message' => 'User Registration Failed!'], 409);         }     }  /\*\* route url untuk frontend \*/     $router->post('register', 'AuthController@register');  /\*\* akses rest api using axios \*/  axios.post(basedurl +"/auth/register", {                     username:this.username,                     nama:this.nama,                     id\_matkul:this.matkul,                     password:this.password,                     password\_confirmation:this.password\_confirmation,                 }) |

**Gambar 4.2***Rest Api Admin Register*

Pada gambar 4.2 merupakan potongan program yang digunakan untuk proses pendaftaran admin, dengan memanfaatkan axios sebagai request data melalui HTTP (*Hypertext Transfer Protocol*) request. Dibandingkan dengan library HTTP request lainnya seperti ajax atau fetch api, axios lebih ringan, promised-based, serta mendukung async dan await untuk kode yang *asynchronous*. Dengan menggunakan HTTP *request*method POST data tersebut akan disimpan ke dalam database yang ada di backend.

REST API terletak pada bagian *frontend* dimana dengan memanfaatkan axios sebagai library untuk menampung url API seperti yang di tunjukan pada gambar 4.2*Rest Api Admin Register* bagian akses rest api using axios, sehingga proses request data menjadi lebih cepat dibandingkan tanpa menggunakan REST API karena tidak perlu lagi melakukan perulangan component pada frontend sehingga tidak membebani memory yang dapat mengakibatkan lambatnya request data. Dengan memanfaatkan asycronous await proses load data menjadi lebih flexible dibandingkan dengan sycronous, dimana proses load datanya harus menunggu proses load data lainnya selasai baru proses tersebut dijalankan sehingga ketika terjadi kegagalan proses diawal maka proses selanjutnya tidak akan bisa di eksekusi.

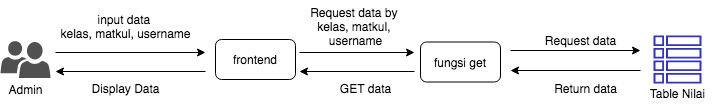
1. **Rest Api GET Nilai Berdasarkan Matakuliah, Kelas dan Username**

|  |
| --- |
| /\*\* fungsi get nilai by username admin \*/  $public function filterNilaiByMatkul($input, $input2, $input3){          $nilai = Nilai::with('matkul.mahasiswa')->with('admin')->get();          $collections = collect($nilai);          $filter = $collections->filter(function ($value, $key) use ($input, $input2, $input3){              return $value->matkul->kd\_matkul == $input && $value->matkul->kelas == $input2 && $value->admin->username == $input3;          })->paginate(25);          return $filter;      }    /\*\* route url untuk frontend \*/  $router->get('/get-nilai/kode/{input}/kelas/{input2}/admin-username/{input3}', 'NilaiController@filterNilaiByMatkul');  /\*\* akses rest api using axios \*/  const nilai = await axios.get(basedurl + '/nilai/get-nilai/kode/' +this.kodeFil+ '/kelas/' +this.kelasFil+ '/admin-username/' + Cookies.get('user'))  this.nilais = nilai.data; |

**Gambar 4.3***Rest Api GET Nilai Berdasarkan Mata Kuliah, Kelas dan Username*

Pada gambar 4.3 merupakan potongan program yang digunakan untuk get nilai berdasarkan matkul, kelas, dan username admin, dimana dengan menggunakan fitur collections yang ada di dalam laravel lumen sehingga data bisa di filter berdasarkan data turunan dari nested json yang akan digunakan. Yang nantinya akan di hit melalui sebuah url yang diakses menggunakan axios untuk mendapatkan data dari *Backend* ke *Frontend*.

REST API method GET digunakan untuk melakukan pengambilan data sehingga proses pengambilan data lebih cepat dan proses maintance koding menjadi lebih rapi, sehingga ketika terdapat bug atau kesalahan dalam proses load data mudah untuk di deteksi. Pada gambar 4.3 *Rest Api GET Nilai Berdasarkan Mata Kuliah, Kelas dan Username* bagian fungsi get nilai by username admin menggunakan eloquent sehingga proses query data menjadi lebih cepat dan pada bagian route url untuk frontend merupakan url api yang digunakan untuk mengakses fungsi yang ada sehingga lebih memudahkan dalam proses maintance koding ketika menggunakan method yang berbeda pada url api yang ada.



**Gambar 4.4***Aliran proses fungsi  GET Nilai Berdasarkan Mata Kuliah, Kelas dan Username*

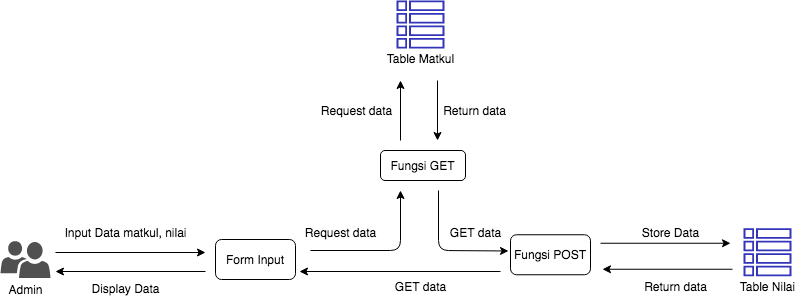
1. **Rest Api POST Nilai**

|  |
| --- |
| /\*\* fungsi post nilai \*/  public function postNilai(Request $request){         $nilai = new Nilai;         $nilai->id\_matkul = $request->id\_matkul;         $nilai->id\_admin = $request->id\_admin;         $nilai->pertemuan = $request->pertemuan;         $nilai->pretest = $request->pretest;         $nilai->laporan = $request->laporan;         $nilai->nilai\_total = ($request->laporan \* 60/100) + ($request->pretest \* 40/100);         $nilai->save();         return $nilai;     }  /\*\* route url untuk frontend \*/     $router->post('/post-nilai', 'NilaiController@postNilai');  /\*\* akses rest api using axios \*/  console.log('create data')             console.log(this.defaultItem)             for(const index of this.defaultItem){               index.id\_matkul = index.id               index.id\_admin = this.user.id               index.semester = this.semester               index.thn\_akademik = this.thn\_akademik               index.pertemuan = this.pertemuan               if(index.pretest == null || index.pretest == ""){                 index.pretest = 0               }if(index.laporan == null || index.laporan == ""){                 index.laporan = 0               }               const response = await axios.post(basedurl +'/nilai/post-nilai', index)               this.nilais = response.data             } |

**Gambar 4.5***Rest Api POST Nilai*

Gambar 4.5 merupakan potongan program yang digunakan untuk menambahkan nilai mahasiswa kedalam database, dimana fungsi postNilai(Request $request) sebuah query builder untuk menambahkan data mahasiswa kedalam database yang kemudian diakses oleh frontend melalui sebuah url dengan menggunakan axios. Dimana pada bagian frontend data yang akan disimpan berupa array yang kemudian berdasarkan index dari inputan data nilai mahasiswa tersebut. Ketika data berhasil ditambahkan maka data nilai mahasiswa akan di reload ulang untuk menampilkan data terbaru dari inputan tersebut.

REST API method POST ini digunakan dalam proses store data kedalam database sehingga proses pengiriman data menjadi lebih aman, karena data yang dikirim di sembunyikan terutama kode unique dari data tersebut yang harus di jaga kerahasiaannya. Pada gambar 4.5 *Rest Api POST Nilai* bagian akses rest api using axios “axios.post(basedurl +'/nilai/post-nilai', index)” tidak menggunakan parameter sehingga data yang dikirimkan tidak bisa di lihat melalui url api yang digunakan.



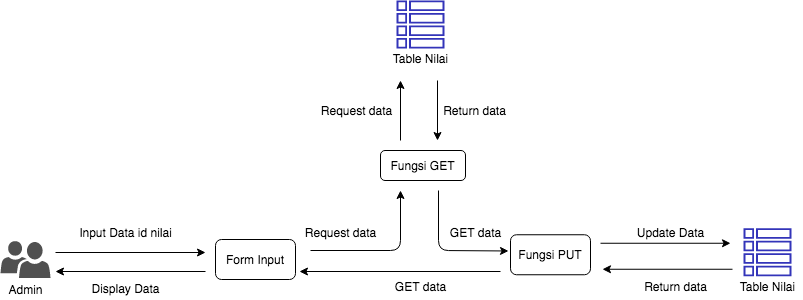
**Gambar 4.6***Aliran Proses POST Nilai*

1. **Rest Api PUT Nilai**

|  |
| --- |
| /\*\* fungsi put nilai \*/  public function putNilai(Request  $request, $id){         $nilai = Nilai::find($id);         $nilai->id\_matkul = $request->id\_matkul;         $nilai->pertemuan = $request->pertemuan;         $nilai->pretest = $request->pretest;         $nilai->laporan = $request->laporan;         $nilai->nilai\_total = ($request->laporan \* 60/100) + ($request->pretest \* 40/100);         $nilai->save();           return $nilai;     }  /\*\* route url untuk frontend \*/     $router->put('/put-nilai/{id}', 'NilaiController@putNilai');  /\*\* akses rest api using axios \*/  console.log('edit data')           console.log(this.editedItem)           const response = await axios.put(basedurl +'/nilai/put-nilai/'+this.editedItem.id, this.editedItem)           this.nilais = response.data           this.getNilai()           this.close() |

**Gambar 4.7***Rest Api PUT Nilai*

Gambar 4.1.6 merupakan potongan program yang digunakan untuk edit data berdasarkan id dari nilai mahasiswa, fungsi putNilai($id) merupakan sebuah *query builder* yang digunakan untuk edit data berdasarkan id nilai mahasiswa. Kemudian diakses melalui sebuah url dengan menggunakan axios kemudian data nilai mahasiswa akan di reload untuk menampilkan hasil dari perubahan data tersebut.



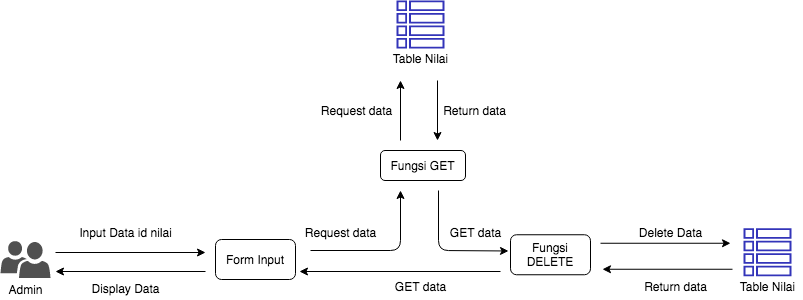
**Gambar 4.8***Aliran Proses PUT Nilai*

1. **Rest Api DELETE Nilai**

|  |
| --- |
| /\*\* fungsi delete nilai \*/  public function deleteNilai($id){         $nilai = Nilai::find($id);         $nilai->delete();         return $nilai;     }  /\*\* route url untuk frontend \*/     $router->delete('/delete-nilai/{id}', 'NilaiController@deleteNilai');  /\*\* akses rest api using axios \*/  const response = this.nilais.data.indexOf(item)         confirm('Apakah anda yakin untuk menghapus data ini?') &&         await axios.delete(basedurl +'/nilai/delete-nilai/'+item.id)         this.nilais.data = response.data         this.getNilai() |

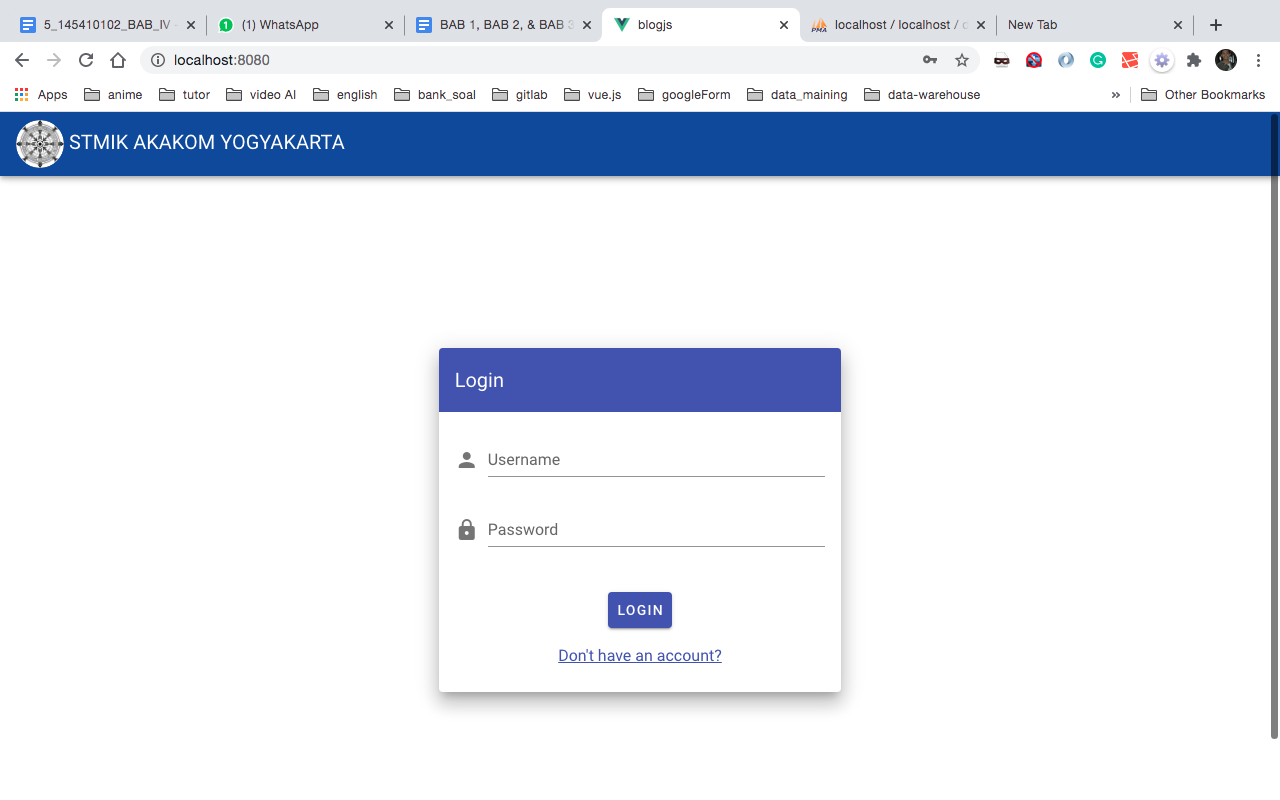
**Gambar 4.9***Rest Api DELETE Nilai*

Gambar 4.1.5 merupakan potongan program yang digunakan untuk menghapus data berdasarkan id data nilai mahasiswa, dimana fungsi deleteNilai($id), dimana parameter $id digunakan untuk get id dari data nilai mahasiswa tersebut, kemudian frontend akan meload data yang terbaru setelah dihapus berdasarkan idnya.



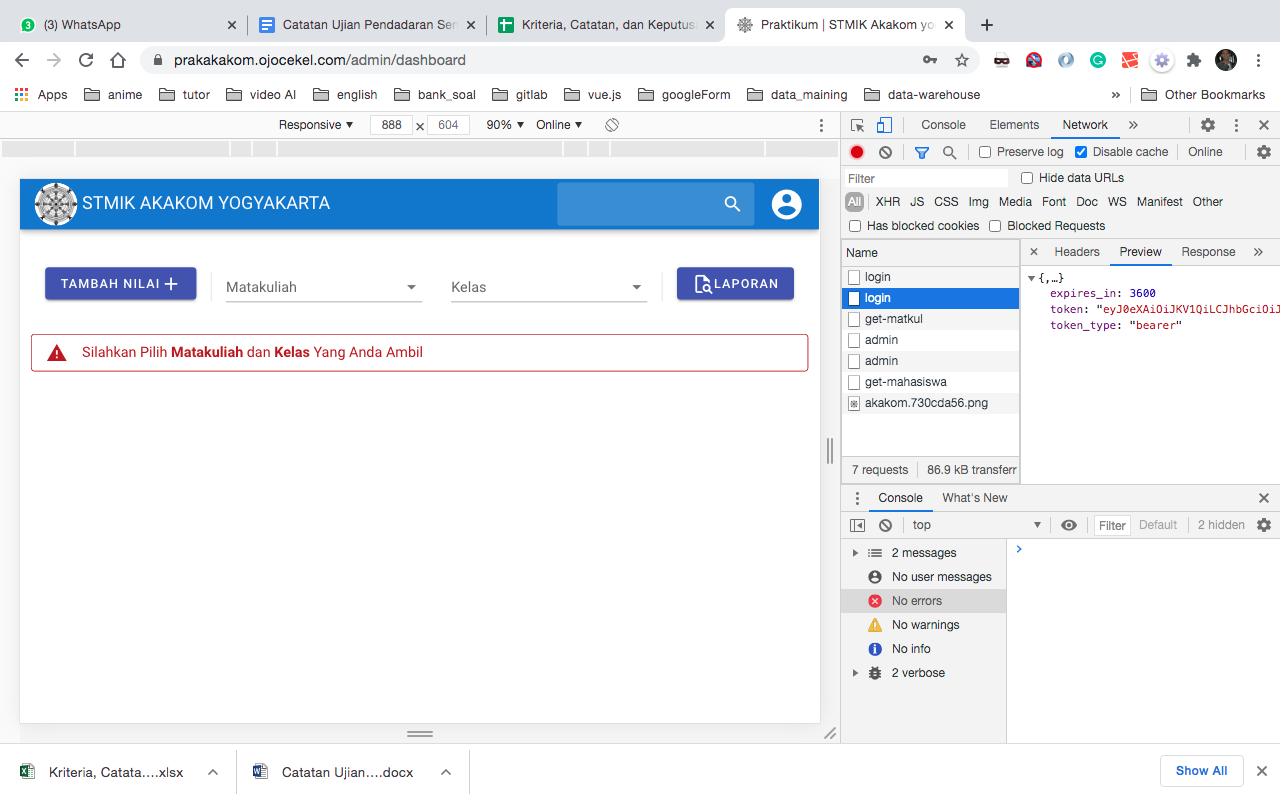
**Gambar 4.10***Aliran Proses DELETE Nilai*

1. **Pembahasan Sistem**
2. **Tampilan Login**



**Gambar 4.11***Tampilan Login*

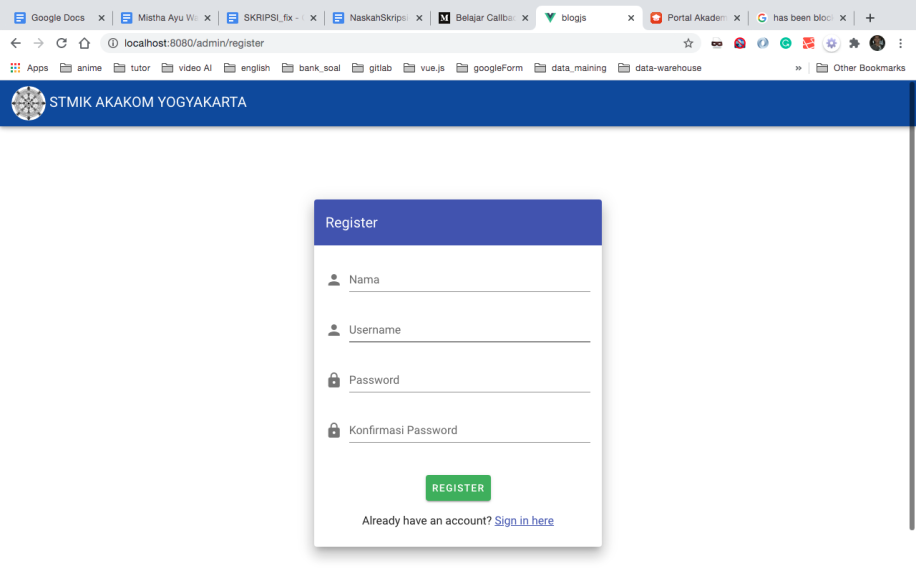
Pada gambar 4.11 adalah tampilan yang digunakan untuk login admin menuju dashboard, yang dimana memberikan izin hak akses kepada asdos yang sudah terdaftar di dalam database untuk masuk ke dalam aplikasi, dimana admin bisa melakukan pengolahan data nilai praktikum mahasiswa.



**Gambar 4.12** *Tampilan return data login*

Gambar diatas merupakan data token yang di return ketika ASDOS melakukan proses login, dimana terdapat data token expirednya, data token dan type tokennya. Data token ini berfungsi untuk melindungi data ASDOS sehingga menjadi lebih secure karena terdapat waktu expirednya sehingga hanya bisa di gunakan dalam jangka waktu 1 jam.

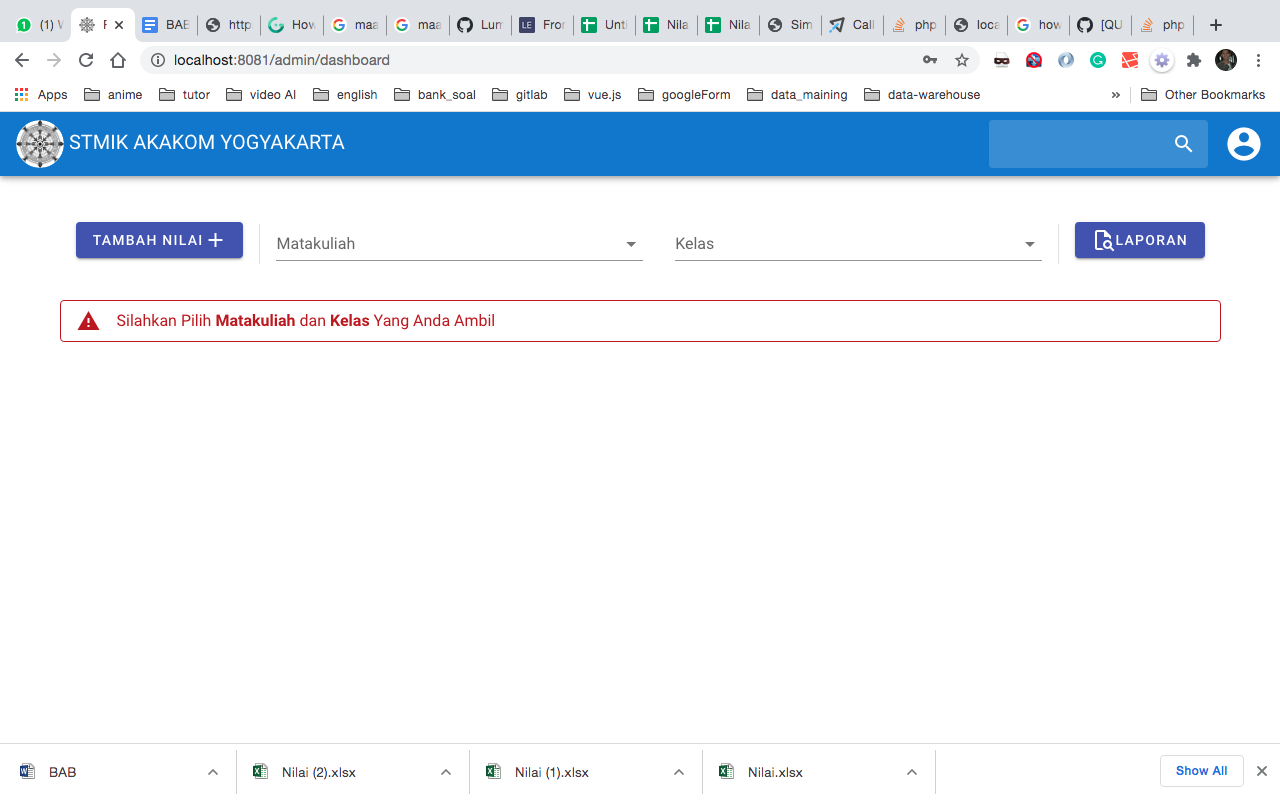
1. **Tampilan Register**



**Gambar 4.13***Tampilan Register*

Pada gambar 4.12 adalah tampilan yang digunakan untuk melakukan input data user untuk disimpan kedalam database yang nantinya akan digunakan sebagai data login admin.

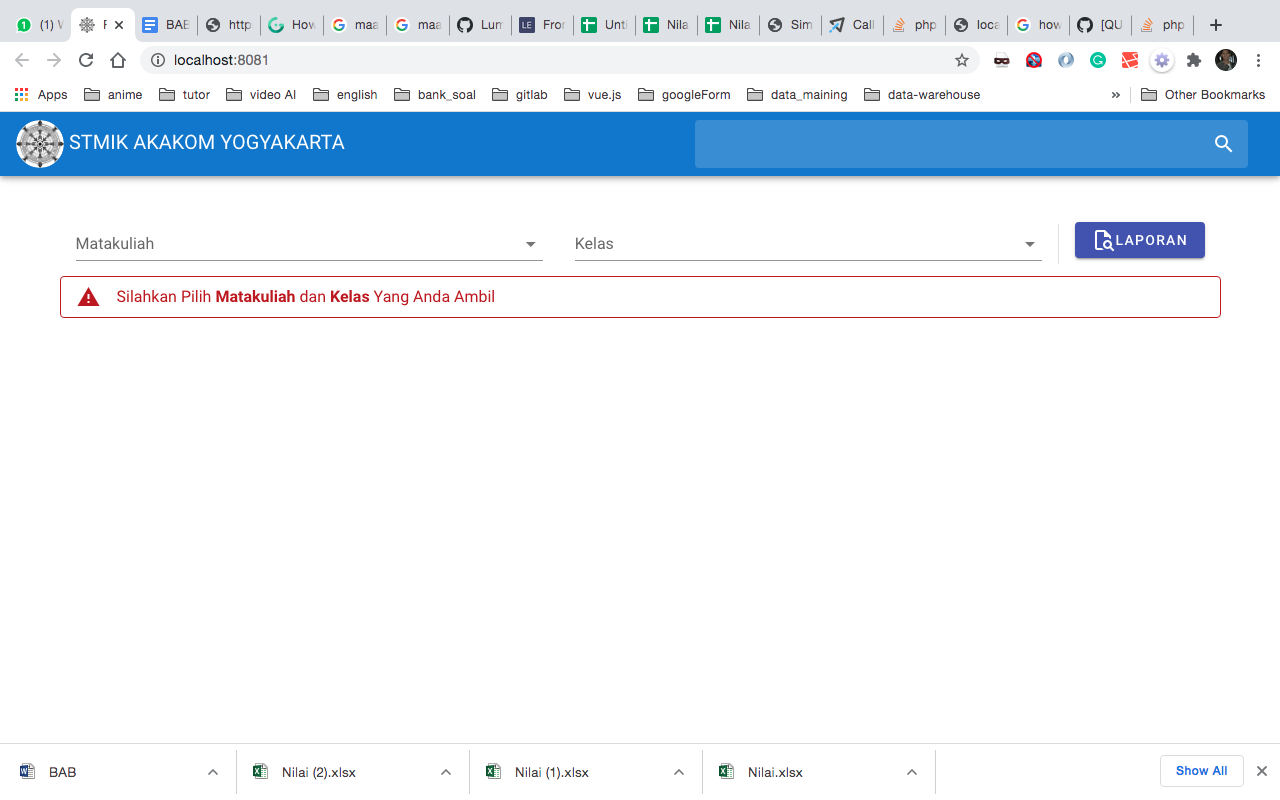
1. **Tampilan Dashboard Admin**



**Gambar 4.14***Tampilan Dashboard Admin*

Pada gambar 4.13 adalah tampilan awal setelah asdos login dimana aplikasi ini menggunakan SPA (*Single Page Applications*), yang dimana user bisa melakukan input, edit dan delete data nilai praktikum di dalam satu halaman. Dimana data yang akan ditampilkan berdasarkan hasil filter per hari atau per mahasiswa.

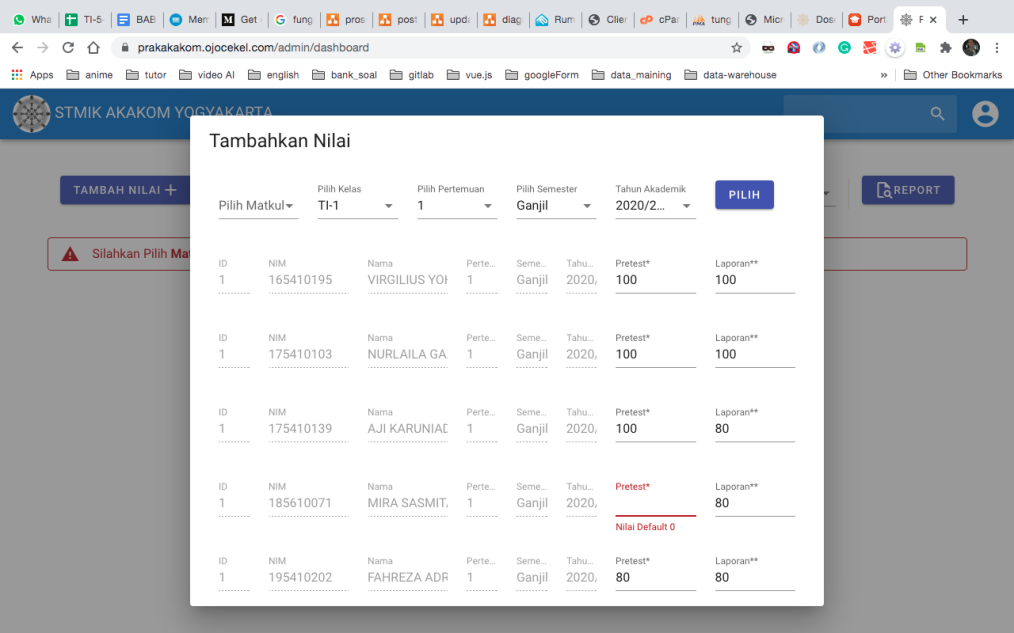
1. **Tampilan Dashboard Pengguna**



**Gambar 4.15***Tampilan Dashboard Pengguna*

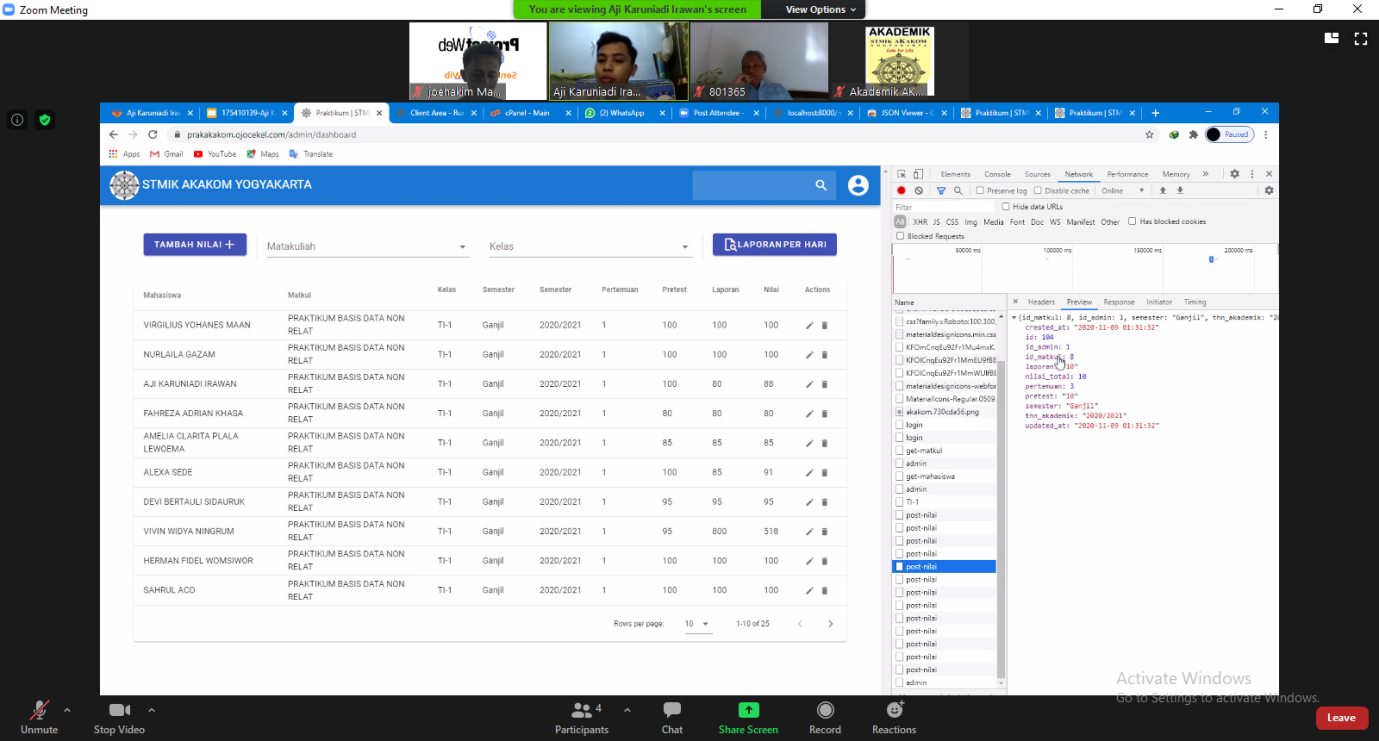
Pada gambar 4.14 merupakan tampilan user untuk melihat nilai praktikum yang didapatkan dan user bisa melakukan download untuk di convert menjadi excel. Dan user juga bisa melakukan filter data berdasarkan mata kuliah dan kelas yang diinginkan.

1. **Tampilan Tambah Nilai**



**Gambar 4.16***Tampilan Tambah Nilai*

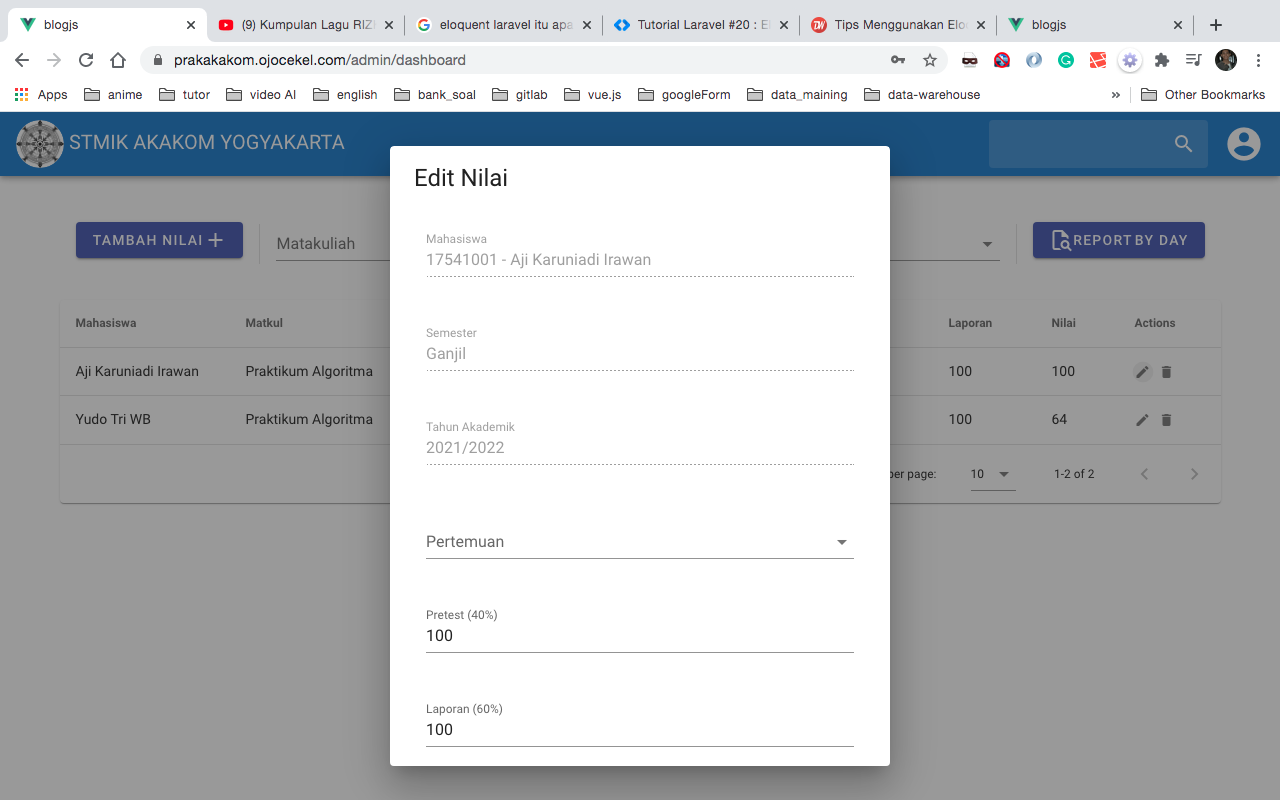
Pada gambar 4.15 adalah tampilan yang digunakan untuk input data nilai praktikum mahasiswa dimana, data yang diinputkan berdasarkan mata kuliah dan kelas yang dipilih. Dimana data matkul di GET berdasarkan mata kuliah dan kelas, dimana datanya di push ke dalam sebuah array, kemudian data array tersebut akan ditampilkan secara otomatis ke dalam form inputan. Admin melakukan inputan data pretest dan laporan berdasarkan index dari array tersebut, dan data tersebut akan disimpan sebagai object dengan menggunakan fungsi "*for of statement*” pada javascript.



**Gambar 4.17***Tampilan proses save data*

Gambar diatas merupakan proses penyimpanan data nilai, dimana data tersebut di simpan dalam bentuk object berdasarkan index dari data tersebut. Sehingga data yang di simpan kedalam database adalah data object bukan data array.

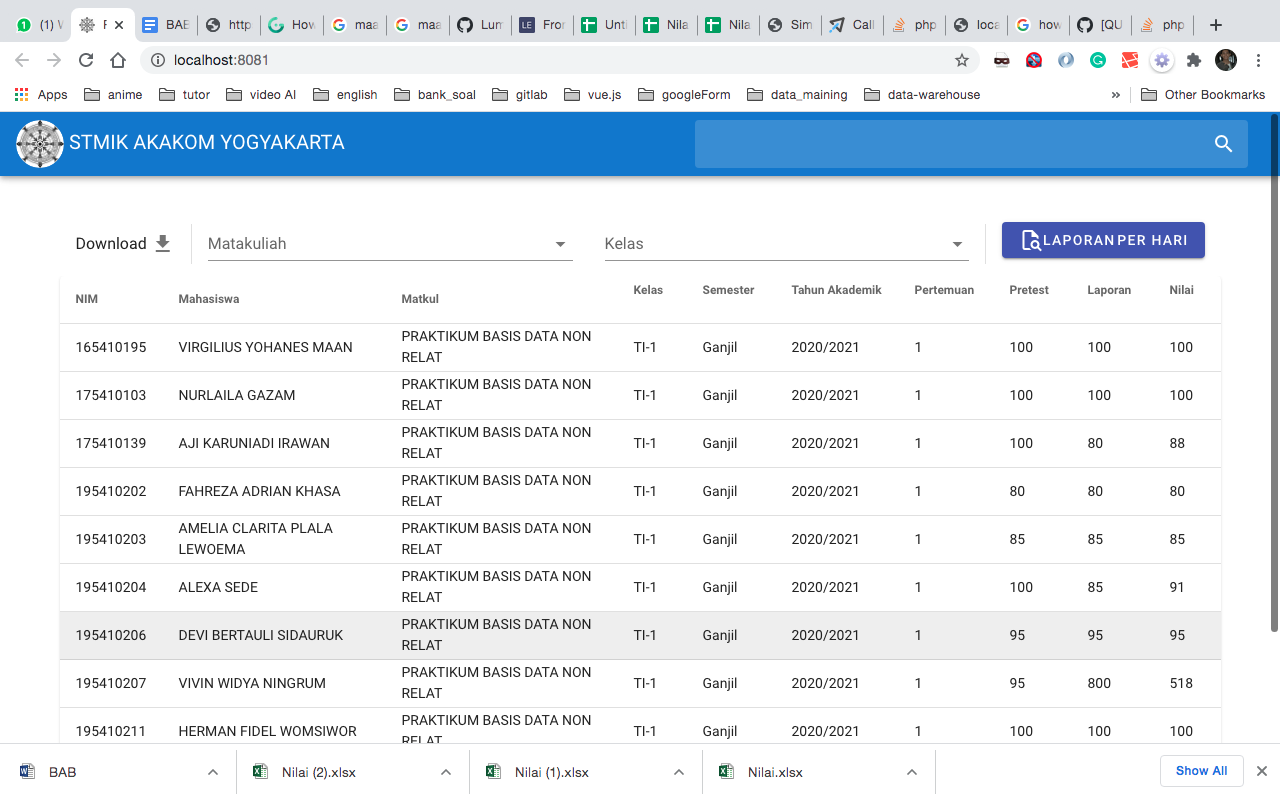
1. **Tampilan Edit Nilai**



**Gambar 4.18***Tampilan Edit Nilai*

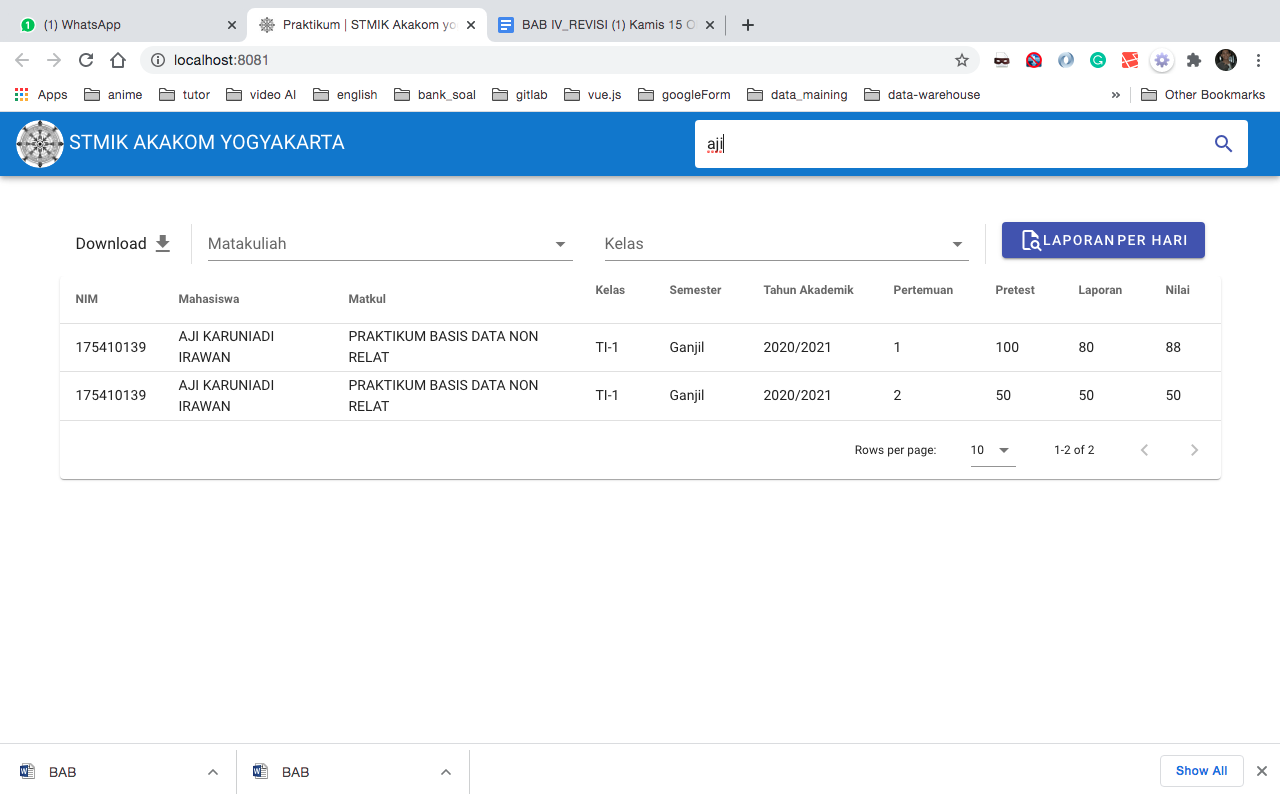
Tampilan ini digunakan untuk melakukan edit data nilai praktikum mahasiswa berdasarkan id nilai praktikum mahasiswa. Dimana data nilai akan di GET berdasarkan id nya kemudian data tersebut akan di return sebagai value untuk form inputan. Dengan menggunakan method PUT data nilai akan di update berdasarkan id kemudian hasil update data nilai akan di return untuk ditampilkan berdasarkan data terupdate.

1. **Tampilan Laporan Per Hari**

****

**Gambar 4.19***Tampilan Laporan Per Hari*

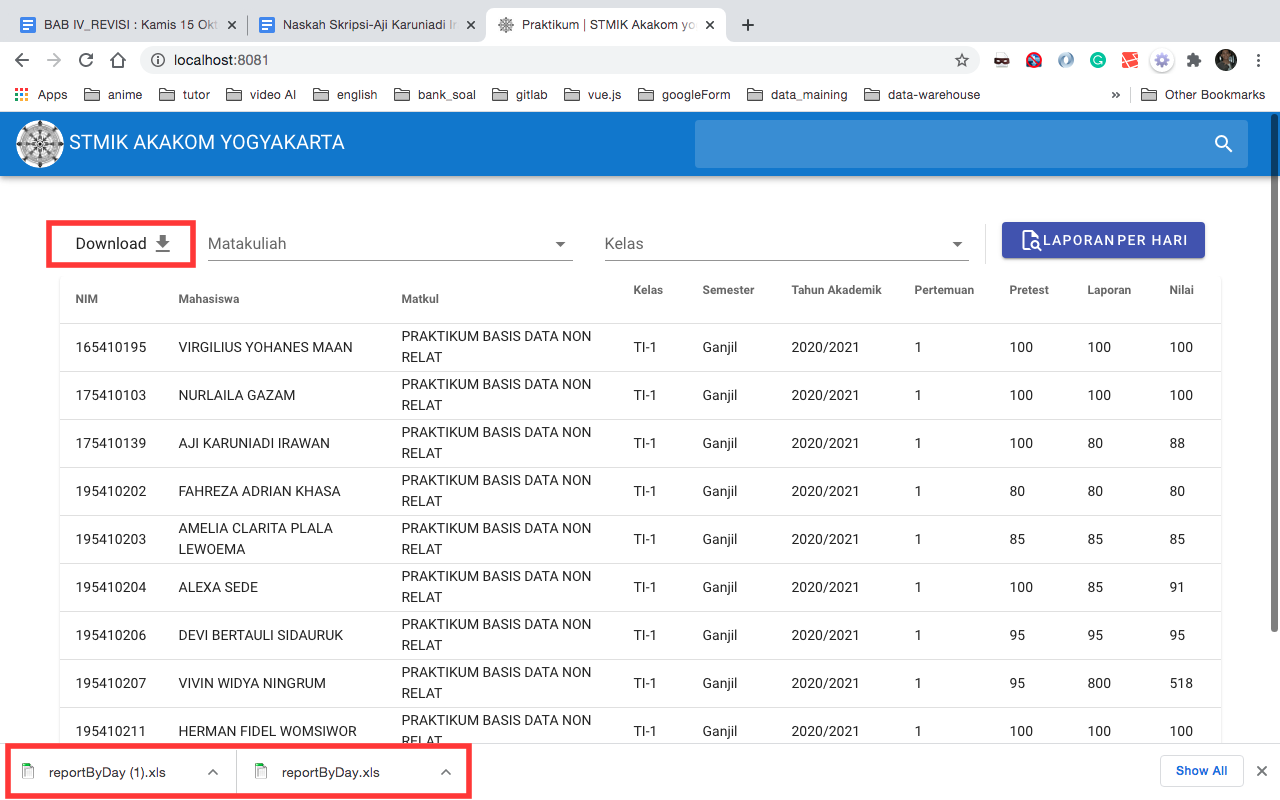
Tampilan ini digunakan untuk melihat nilai user berdasarkan data terupdate dimana data nilai akan ditampilkan berdasarkan hari atau berdasarkan inputan terupdate yang dilakukan oleh Admin (Asdos), user bisa melakukan filter data nilai mahasiswa dengan menginputkan kata kunci pada kolom search di pojok kanan atas.



**Gambar 4.20***Hasil Filter Data Nilai*

Filter data ini bertujuan untuk memudahkan user dalam proses pencarian data nilai mahasiswa sesuai dengan kata kunci yang diinginkan. Sehingga ketika data nilai sudah mencapai ribuan user tidak perlu melihat semua data yang ada cukup melihat data yang ingin ditampilkan berdasarkan kata kunci yang digunakan, fitur ini juga berfungsi untuk mempercepat proses editing data nilai mahasiswa untuk admin, ketika melakukan kesalahan input atau perubahan nilai pada mahasiswa tertentu.

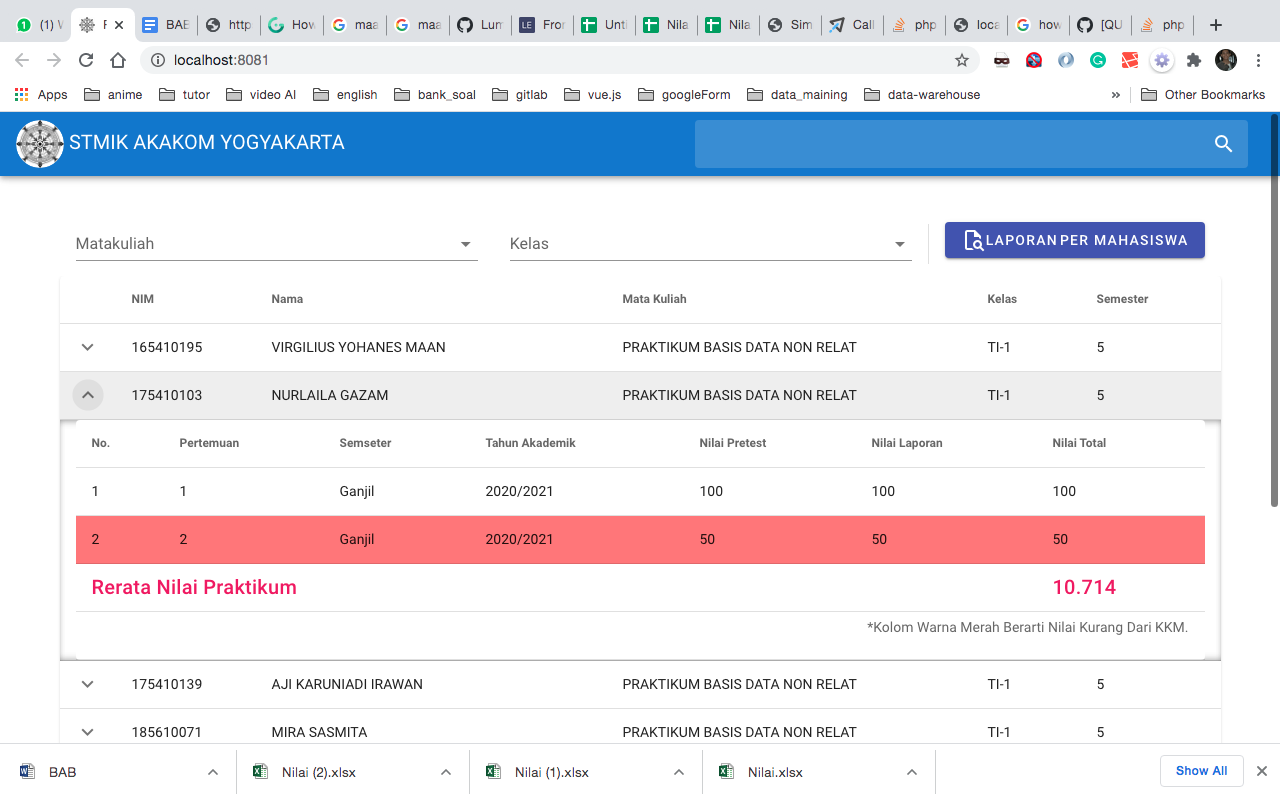
User juga bisa melakukan *export* data nilai laporan per hari dari json ke excel, dengan menggunakan library “vue-json-excel” dimana library tersebut dapat melakukan *export* data nested json.



**Gambar 4.21***Download File Nilai Praktikum*

Untuk melakukan *export* data cukup dengan klik tombol download Download pada kotak merah sebelah kiri gambar, maka data akan langsung di convert dari json ke excel dengan nama file reportByDay.xls, Dimana fitur ini bertujuan untuk memudahkan user melihat data nilai praktikum tanpa harus mengakses website yang ada, sehingga ketika user dalam keadaan offline user tetap bisa melihat nilai praktikum pada file yang sudah didownload.

1. **Tampilan Laporan Per Mahasiswa**

****

**Gambar 4.22***Tampilan Laporan Per Mahasiswa*

Tampilan digunakan untuk menampilkan laporan nilai mahasiswa untuk semua pertemuan berdasarkan matakuliah dan kelas. Nilai yang berada dibawah KKM (Kriteria Ketuntasan Minimal) maka data tersebut akan di blok berwarna merah, sehingga lebih memudahkan user ataupun admin dalam membaca data laporan nilai untuk semua pertemuan. User dan Admin juga bisa melihat rerata nilai dari praktikum selama 14 kali pertemuan.

Berdasarkan hasil Implementasi dan Uji Coba Sistem diatas yaitu menghasilkan sebuah aplikasi input nilai mahasiswa STMIK Akakom Yogyakarta berbasis website dengan memanfaatkan teknologi microservices menggunakan RESTful Api, sehingga mahasiswa (user) dapat melihat nilai praktikum secara langsung melalui aplikasi tersebut. Di sisi lain asdos dapat menginputkan nilai mahasiswa dengan bobot penilaian yang sama sehingga tidak ada lagi perbedaan format untuk laporan penilaian praktikum mahasiswa STMIK Akakom Yogyakarta.

**BAB V**

**PENUTUP**

1. **Kesimpulan**

Berdasarkan perancangan dan implementasi dari BAB sebelumnya, maka penelitian ini dapat diambil kesimpulan sebagai berikut :

1. Berhasil membuat sebuah REST API untuk input nilai mahasiswa.
2. Menghasilkan aplikasi untuk input nilai mahasiswa STIMIK AKAKOM YOGYAKARTA berbasis web app, dimana memudahkan Asdos dalam penginputan nilai dan memudahkan mahasiswa melihat nilai hasil praktikum melalui sebuah website. Dengan memanfaatkan arsitektur microservice sebagai layanan server dan REST API sebagai media pertukaran data.
3. Performance aplikasi menjadi lebih cepat karena proses logika program terpisah dengan proses loading untuk tampilan. Dimana dengan menggunakan REST API dengan method GET, POST, PUT dan DELETE dapat meringankan beban frontend dalam melakukan load data, karena secara tradisional ketika tidak menggunakan REST API bagian frontend akan melakukan load data ditambah dengan perulangan pada element tampilan user interface sehingga beban pada memory menjadi sangat berat dan proses load data menjadi sangat lambat.
4. **Saran**

Dari uraian pembahasan pada bab sebelumnya, sistem aplikasi ini masih mempunyai kekurangan. Adapun saran yang berguna untuk pengembangan aplikasi ini sebagai berikut :

1. Memperbaiki bagian *user interface* aplikasi sehingga menjadi lebih responsif, dimana tampilannya dapat disesuaikan di semua *device* seperti android, ios, tablet ataupun desktop.
2. menambahkan fitur *Authentication* untuk *user* yang berfungsi untuk menampilkan data berdasarkan data *user*, sehingga *user* hanya bisa melihat data nilai praktikumnya sendiri.
3. Pada saat production semua API lebih baik menggunakan method POST dimana untuk menjaga keamanan data karena semua parameter bisa disembunyikan.
4. Hanya menampilkan data matakuliah berdasarkan ASDOS sehingga mengurangi resiko kesalahan dalam inputan data nilai berdasarkan matakuliah.
5. Membuat history penginputan dan update data nilai yang dilakukan oleh ASDOS sehingga dapat memudahkan dalam pengecekan kelasahan input nilai.

**Daftar Pustaka**

Arief, M.Rudianto. 2011. Pemrograman Web Dinamis Menggunakan Php dan Mysql. Yogyakarta: ANDI.

Admin. 2019. *VueJs-#11 Vue Vuetify Login & Crud Auth JWT.* <https://ombagoes.com/vuejs-11-vue-vuetify-login-crud-auth-jwt/>. 10 Agustus 2020, 20:00 PM.

Fauzi, Rizky. 2017.*Pengenalan Lumen Framework, Micro Framework Berbasis PHP.*<https://www.codepolitan.com/pengenalan-lumen-framework-micro-framework-berbasis-php-59f19fe6ea010>. 02 Maret 2020, 20:00 PM.

Fauzi, Rizky. 2017. Pengenalan Lumen Framework, Micro Framework Berbasis PHP. <https://www.codepolitan.com/pengenalan-lumen-framework-micro-framework-berbasis-php-59f19fe6ea010>. 05 April  2020, 21:20 PM.

Hanavian, Akidna. 2019. Hipster Murni dan Vue.js ( Vuetify.js). <https://medium.com/au-ppl/hipster-murni-dan-vue-js-vuetify-js-dacf4dbb513f>. 08 April 2020, 20:00 PM.

Hs, David. 2017. Implementasi Gitlab CI/CD untuk Test dan Deploy Aplikasi Laravel.  <https://medium.com/@davidhsianturi/implementasi-gitlab-ci-cd-untuk-test-dan-deploy-aplikasi-laravel-9c5dab7ce138>. 11 April 2020, 20:00 PM.

Kiddy. 2018. *Kenapa API Harus Microservices?.* <https://medium.com/@kiddy.xyz/pengertian-microservices-dan-kenapa-restful-api-harus-dibuat-seperti-itu-7326f217042d>. 02 Maret 2020, 20:00 PM.

[Ndifreke Friday](https://www.twitter.com/ndiecodes). 2019.*Build a JWT Authenticated API with Lumen(v5.8).* <https://ndiecodes.com/posts/build-a-jwt-authenticated-api-with-lumen>. 10 Agustus 2020, 19:00 PM.

Putra, Andre. 2020. *Restfull Api Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices ( Studi Kasus : Portal Berita ).* Yogyakarta : STMIK AKAKOM Yogyakarta.

Purnama, Heri. 2016. *Aplikasi Pengelolaan Skripsi di STMIK AKAKOM Menggunakan Arsitektur Microservice*.  Yogyakarta : STMIK AKAKOM Yogyakarta.

Qamarudin, Ahmad. 2018. *Implementasi Arsitektur Microservice Menggunakan Restful API Untuk Portal Akademik PP AL-MUNAWWIR.* Yogyakarta : STMIK AKAKOM Yogyakarta.

Redmond, Paul. 2016. *Writing APIs with Lumen: A Hands-on Guide to Writing API Services With PHP.* Paul Redmond.

Yudana. 2019. *Pengertian dan Konsep Restful Api Programming.* <https://www.yudana.id/pengertian-dan-konsep-restful-api-programming/>. 05 April 2020, 20:00 PM.