**LAPORAN KERJA PRAKTIK**

**STUDI INDEPENDEN BERSERTIFIKAT TAHUN 2023**

**RESTFUL-API APLIKASI MYGRAM**

**PT HACKTIVATE TEKNOLOGI INDONESIA**



Disusun oleh

**Nama : Maulana Daffa Ardiansyah**

**NIM : 123200130**

**PROGRAM STUDI INFORMATIKA**

**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” YOGYAKARTA**

**2024**

**STUDI INDEPENDEN BERSERTIFIKAT TAHUN 2023**

**RESTFUL-API APLIKASI MYGRAM**

**PT HACKTIVATE TEKNOLOGI INDONESIA**

**KERJA PRAKTIK**

Diajukan untuk memenuhi sebagian persyaratan kurikulum Sarjana

pada Program Studi Informatika

# HALAMAN JUDUL

A yellow and red logo

Description automatically generated

Disusun oleh

**Nama : Maulana Daffa Ardiansyah**

**NIM : 123200130**

**PROGRAM STUDI INFORMATIKA**

**JURUSAN INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” YOGYAKARTA**

**2024**

**HALAMAN PENGESAHAN**

# HALAMAN PERSETUJUAN

**KERJA PRAKTIK**

**STUDI INDEPENDEN BERSERTIFIKAT TAHUN 2023**

**RESTFUL-API APLIKASI MYGRAM**

**PT HACKTIVATE TEKNOLOGI INDONESIA**



Disusun Oleh :

Maulana Daffa Ardiansyah 123200130

Telah diperiksa dan disetujui oleh Dosen Pembimbing Kerja Praktik

Pada tanggal : ………………….

|  |  |
| --- | --- |
| Pembimbing Kerja Praktik  **Dr. Novrido Charibaldi, S.Kom., M.Kom.**  **NIK . 2 6811 96 0066 1** | Pembimbing Lapangan  **Arif Santoso** |

Mengetahui

Koordinator Kerja Praktik JIF

**Simon Pulung Nugroho, S.Kom., M.Cs.**

**NIP. 19840218 201803 1 001**

# KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas segala limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan akhir kerja praktik ini dengan lancar dan tepat waktu. Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan mata kuliah kerja praktik di Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Penulis mengucapkan terima kasih kepada:

1. Dr. Novrido Charibaldi, S.Kom, M.Cs., selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan inspirasi dalam perjalanan penulis.
2. Arif Santoso selaku Mentor Kelas Back-End.
3. Teman satu kelas pada program studi independent yang selalu membantu dalam segala kegiatan dan tugas.
4. Teman kelompok proyek yang telah berkerjasama dalam menyelesaikan proyek akhir studi independen.
5. Seluruh pengurus dan tim PT Hacktivate Teknologi Indonesia lainnya yang telah membimbing penulis dalam menyelesaikan kegiatan Studi Independen ini.
6. Kedua orang tua penulis yang selalu memberikan dukungan serta doa yang tidak pernah berhenti agar penulis dapat menyelesaikan program dan laporan kerja praktik ini. Seluruh pihak yang telah membantu penulis dalam menyelesaikan kerja praktik ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak untuk perbaikan di masa depan. Demikianlah kata pengantar ini penulis susun. Semoga bermanfaat.

Yogyakarta, Mei 2024

Penulis

# DAFTAR ISI

[HALAMAN JUDUL i](#_Toc159474271)

[HALAMAN PENGESAHAN ii](#_Toc159474272)

[KATA PENGANTAR iii](#_Toc159474273)

[DAFTAR ISI iv](#_Toc159474274)

[DAFTAR TABEL v](#_Toc159474275)

[DAFTAR GAMBAR vi](#_Toc159474276)

[DAFTAR LAMPIRAN vii](#_Toc159474277)

[ABSTRAK viii](#_Toc159474278)

[BAB I PENDAHULUAN 1](#_Toc159474279)

[1.1 Latar Belakang 1](#_Toc159474280)

[1.2 Rumusan Masalah 2](#_Toc159474281)

[1.3 Tujuan 3](#_Toc159474282)

[1.4 Manfaat 3](#_Toc159474283)

[BAB II PROFIL INSTANSI KERJA PRAKTIK 4](#_Toc159474284)

[2.1 Profil Instansi 4](#_Toc159474285)

[2.2 *Job Description* 5](#_Toc159474286)

[BAB III METODOLOGI 6](#_Toc159474287)

[3.1 Analisis Permasalahan 6](#_Toc159474288)

[3.2 Diagram Alur Perancangan Solusi 6](#_Toc159474289)

[3.3 Perancangan Solusi 7](#_Toc159474290)

[BAB IV HASIL DAN PEMBAHASAN 9](#_Toc159474291)

[4.1 Hasil 9](#_Toc159474292)

[4.2 Pembahasan 9](#_Toc159474293)

[BAB V PENUTUP 10](#_Toc159474294)

[5.1 Kesimpulan 10](#_Toc159474295)

[5.2 Saran 10](#_Toc159474296)

[5.3 Keberlanjutan 10](#_Toc159474297)

[DAFTAR PUSTAKA 11](#_Toc159474298)

[LAMPIRAN 12](#_Toc159474299)

# DAFTAR TABEL

# DAFTAR GAMBAR

# DAFTAR LAMPIRAN

# ABSTRAK

Abstrak merupakan *outline* luaran kerja praktik yang berupa deskripsi secara singkat dari seluruh hasil. Abstrak disusun dengan kalimat yang singkat, jelas, runtut, dan sistematis dan dapat menggambarkan isi laporan secara keseluruhan. Abstrak disusun dalam bahasa Indonesia, **disusun menjadi minimal 3 alinea, tidak lebih dari 1 halaman, berkisar antara 150-250 kata, diketik dengan jarak 1 spasi.**

Abstrak laporan kerja praktik memuat ***apa dan mengapa*** luaran kerja praktik dibuat/dikerjakan, ***bagaimana dikerjakan, dan apa hasil penting*** yang diperoleh? **Alinea (paragraph) pertama dalam Abstrak berisi hasil analisis masalah dan tujuan penelitian**. **Alinea kedua berisi metode penelitian (langkah-langkah penyelesaian masalah). Alinea ketiga mengungkapkan hasil dari penelitian (secara singkat).**

Di bagian bawah abstrak dituliskan kata-kata kunci, bisa berupa kata-kata penting dalam abstrak atau kata yang sering muncul, berjumlah minimal 3 (tiga) dan maksimal 5 (lima) kata.

**Kata kunci:** satu, dua, tiga, empat, lima

# BAB I PENDAHULUAN

## Latar Belakang

Kecepatan penyaluran informasi berbanding lurus dengan cepatnya perkembangan teknologi yang ada di masa ini. Semakin canggih teknologi yang ada semakin cepat pula informasi dapat menyebar. Hal ini dapat menjadi pisau bermata dua yang berarti sebagai kesempatan dan juga hambatan bagi pelaku-pelaku bisnis atau bahkan yang bukan pelaku bisnis tergantung dengan sikap mereka dalam menghadapi keadaan tersebut.

Pihak yang menganggap hal ini sebagai kesempatan akan mencoba untuk beradaptasi dan mencoba untuk memanfaatkan teknologi yang ada. Sedangkan pihak yang menganggap sebagai hambatan akan merasa terbebani karena tidak dapat beradaptasi pada kondisi tersebut.

Kesempatan inilah yang dapat dikembangkan dengan teknologi digital untuk mempermudah pelaku bisnis dalam beradaptasi dengan teknologi masa ini. Salah satunya adalah dengan membangun sebuah sistem yang dapat menjadi sebuah alat bagi mereka yang ingin mempromosikan usaha mereka, *personal branding*, ataupun sekedar berbagi momen-momen yang tak terlupakan.

Oleh karena itu, untuk menyelesaikan permasalahan pada adaptasi dan kebutuhan masyarakat dengan teknologi, direkomendasikan pengembangan Restful API yang dapat diakses berbagai macam platform sebagai mesin pengolahan informasi daring yang dapat dengan mudah menyebarluaskan segala informasi yang diinginkan.

## Rumusan Masalah

Berdasarkan latar belakang masalah yang ada, maka dapat dirumuskan permasalahan dalam kegiatan ini yaitu:

1. Bagaimana teknologi digital dapat digunakan untuk memfasilitasi penyebaran informasi secara efisien?
2. Bagaimana agar aplikasi yang dikembangkan dapat diakses dari berbagai macam platform?

## Tujuan

Berdasarkan rumusan masalah yang telah dibuat, tujuan dari aplikasi ini yaitu:

1. Mengembangkan sebuah aplikasi berupa sosial media yang memanfaatkan teknologi digital untuk menyebarkan informasi secara efisien.
2. Memastikan bahwa aplikasi yang dikembangkan memiliki ketersediaan dan aksesibilitas oleh pengguna dari berbagai macam platform.

## Manfaat

Adapun manfaat yang didapatkan dari pembuatan produk digital berupa RESTful-API sosial media adalah sebagai berikut:

1. Memfasilitasi akses informasi dan interaksi antara pengguna dari berbagai macam platform, termasuk desktop, mobile, dan web.
2. Memperluas jangkuan informasi yang bisa didapatkan dan disebarkan oleh pengguna.

# BAB II PROFIL INSTANSI KERJA PRAKTIK

## Profil Instansi

PT. Hacktivate Teknologi Indonesia atau yang biasa dikenal dengan Hacktiv8, seperti namanya, merupakan sebuah perusahaan yang bergerak di bidang teknologi. Hactiv8 adalah penyelenggara program pelatihan intensif atau disebut bootcamp untuk mentransformasi pemula menjadi talenta digital siap kerja.

Hacktiv8 memiliki beberapa kantor di beberapa kota di Indonesia, yaitu di Jakarta, Tangerang, Surabaya, dan Malang. Pada pelaksanaan program Studi Independen Kampus Merdeka, kantor yang bertanggung jawab yaitu yang berada di Jakarta dengan alamat Gedung Aquarius Lt. 1&2, Jl. Sultan Iskandar Muda No. 7, RT. 005 RW. 009, Kebayoran Lama Selatan, Kebayoran Lama, Jakarta Selatan 12240. (<https://maps.app.goo.gl/cxzCQWq4ftQ6cgTL8>).



**Gambar 2.1** Logo Hacktiv8

Hacktiv8 didirikan oleh Riza Fahmi (Co-Founder & Developer Evangelist) dan Ronald Ishak (Co-Founder & CEO) pada tahun 2016 dengan tujuan yang jelas dan ambisius untuk mengatasi kelangkaan *software* *developer* yang terampil pada sektor teknologi di Indonesia. Mereka mengidentifikasi bahwa pesatnya perluasan bisnis yang mendukung teknologi di Indoensia terhambat oleh kurangnya talenta yang berkualitas dan melihat itu sebagai sebuah kesempatan untuk membuat sebuah perubahan yang berarti untuk melatih generasi *developer* berikutnya.

Terlepas dari sejarah terbentuknya Hacktiv8, perusahaan ini juga memiliki visi dan misi yang tidak kalah penting, visi misi Hacktiv8 adalah sebagai berikut:

**Tabel 2.1** Visi dan Misi Hacktiv8

|  |  |
| --- | --- |
| **Visi** | **Misi** |
| * Menjadi pemimpin dalam pendidikan dan pelatihan teknologi di Indonesia. * Membangun talenta digital kelas dunia yang siap berkontribusi pada kemajuan bangsa. * Menjadi katalisator transformasi digital di Indonesia. | * Menyediakan pendidikan dan pelatihan teknologi yang berkualitas tinggi dan berstandar internasional. * Membangun ekosistem pendidikan dan pelatihan teknologi yang berkelanjutan. * Meningkatkan akses Pendidikan dan pelatihan teknologi bagi seluruh masyarakat Indonesia. * Berkolaborasi dengan berbagai pihal untuk memajukan industri teknologi di Indonesia. |

Hacktiv8 terkenal dengan program bootcamp-nya yang imersif, dirancang untuk mengubah pemula menjadi talenta digita; yang siap kerja dalam waktu singkat. Hacktiv8 menawarkan berbagai program bootcamp maupun selain bootcamp antara lain:

**Tabel 2.2** Daftar Produk Hacktiv8

|  |  |
| --- | --- |
| **Bootcamp** | **Lainnya** |
| * **Pengembangan Web**: Front-End, Back-End, Full-Stack. * **Pemrogramman Mobile**: Android, iOS. * **Data Science**: Machine Learning, Artificial Intelligence. * **Desain**: UI/UX Design. | * **KODE**: Program studi sarjana terapan di bidang informatika yang berfokus pada pengembangan perangkat lunak. * **Hacktiv8 Enterprise**: Solusi pelatihan dan pengembangan keahlian untuk perusahaan. * **Magang Mandiri**: Program magang yang memungkinkan mahasiswa untuk mendapatkan pengalaman kerja di perusahaan teknologi. |

## *Job Description*

Program Studi Independen yang dilaksanakan di PT Hacktivate Teknologi Indonesia memiliki beberapa sesi kegiatan. Kegiatan tersebut antara lain belajar mandiri, *live* *session*, *assignment*, dan yang terakhir adalah pengerjaan proyek akhir.

Pada sesi belajar mandiri, peserta akan diberikan materi melalui platform pembelajaran dari Hacktiv8 yang bernama KODE. Platform ini menyediakan berbagai macam materi dalam bentuk tulisan maupun video yang dapat diakses melalui sebuah proses transaksi. Peserta diberikan 4 kelas yang dapat diakses secara gratis untuk dapat memulai pembelajaran mandiri selama program Studi Independen berlangsung.

Kemudian, ditengah-tengah pembelajaran mandiri, live session akan diadakan oleh mentor sebagai kelas sinkron yang berlangsung secara daring melalui platform Google Meet. Kegiatan live session ini dilaksanakan dua kali seminggu setiap hari senin dan kamis. Selama proses belajar mandiri dan live session berlangsung, peserta diberikan tugas yang harus dikerjakan secara mandiri mengenai materi-materi teknikal yang sudah diberikan.

Menjelang akhir masa program Studi Independen, peserta diberikan proyek akhir untuk dikerjakan dan kemudian akan di presentasikan di depan mentor. Proyek akhir dikerjakan secara berkelompok dan setiap kelompok memiliki 2 anggota. Pemagian tugas antar dua anggota dibebaskan kepada masing-masing kelompok.

Pembagian tugas pada proyek kelompok adalah sebagai berikut:

**Tabel 2.2** *Timeline* pengerjaan luaran kerja praktik

|  |  |
| --- | --- |
| **Penulis** | **Rekan Kelompok** |
| * Membuat *Photos Service* * Membuat *Comments Service* * Membuat *script* t*esting* untuk *Photos Service* * Membuat *script testing* untuk *Comments Service* | * Membuat script migrasi Databse * Membuat User Service * Membuat *script testing* untuk *User Service* * Melakukan *Deployment* |

**Tabel 2.2** *Timeline* pengerjaan luaran kerja praktik

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Oktober** | | **November** | | | | **Desember** | | | |
| **III** | **IV** | **I** | **II** | **III** | **IV** | **I** | **II** | **III** | **IV** |
| 1. | *Setup* GitHub *Organization* |  |  |  |  |  |  |  |  |  |  |
| 2. | *Creating* Database |  |  |  |  |  |  |  |  |  |  |
| 3. | *Project* *Initialization* |  |  |  |  |  |  |  |  |  |  |
| 4. | *Creating User Service* |  |  |  |  |  |  |  |  |  |  |
| 5. | *Creating Photos Service* |  |  |  |  |  |  |  |  |  |  |
| 6. | *Creating Comments Service* |  |  |  |  |  |  |  |  |  |  |
| 7. | *Service Testing* |  |  |  |  |  |  |  |  |  |  |
| 8. | *Project Deployment* |  |  |  |  |  |  |  |  |  |  |
| 9. | *Project Presentation* |  |  |  |  |  |  |  |  |  |  |

# BAB III METODOLOGI

## Analisis Permasalahan

Berdasarkan pada latar belakang, kelebihan yang dapat diambil pada permasalahan terdapat pada kecepatan penyebaran informasi. Hal ini menjadi nilai positif dari permasalahan yang dialami. Kekurangannya terpada kurangnya kemampuan masyarakat dalam beradaptasi pada kecepatan tersebut. Kedua hal inilah yang dapat dianalisa untuk mendapatkan solusi yang relevan.

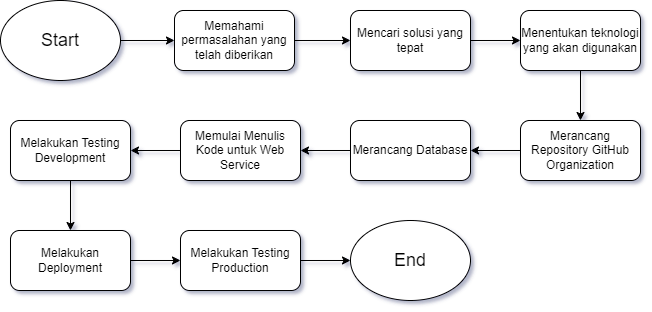
*Framework* atau kerangka kerja yang digunakan dalam menganalisis permsalahan pada kasus ini adalah *Framework* SWOT. *Framework* ini sudah banyak digunakan organisasi dalam berbagai macam kasus untuk menganalisa kekurangan dan kelebihan dari internal maupun external. SWOT sendiri merupakan sebuah kepanjangan dari *Strength*, *Weaknesses*, *Opportunities*, dan *Threats*. Framework ini menganalisa hal-hal yang berfokus pada Kelebihan (*Strength*) dan Kekurangan (*Weaknesses*) untuk mengidentifikasi Kesempatan (*Opportunities*) dan Ancaman (*Threats*).

**Tabel 3.1.** Hasil analisis permasalahan dan daftar solusi yang diusulkan

|  |  |
| --- | --- |
| **Permasalahan** | **Solusi Yang Diusulkan** |
| Kurangnya kemampuan masyarakat dalam beradaptasi dengan kecepatan penyebaran informasi. | Dibuat suatu sistem yang dapat membantu masyarakat dalam memanfaatkan kecepatan penyebaran informasi. |
| Adanaya perbedaan perangkat yang sangat bervariasi di kalangan masyarakat. | Membuat sistem dapat diakses dari berbagai macam platform dan perangkat yang berbeda beda. |

## Diagram Alur Perancangan Solusi

Alur perancangan solusi dapat dilihat pada gambar 3.1 yang telah digambarkan dalam bentuk diagram. Hal pertama yang dilakukan yaitu memahami dengan tepat permasalahan yang telah didapatkan dan menganalisa permasalahan tersebut dengan *framework* analisa SWOT yang telah disebutkan sebelumnya. Setelah menganalisa permasalahan dan menemukan beberapa solusi, kemudian menentukan teknologi yang akan digunakan. Hal ini mencakup mengenai teknologi seperti *database*, bahasa pemrogramman, *framework*, dan juga *tools* kolaborasi seperti GitHub, serta *platform* *deployment* berupa Railway.



**Gambar 3.1** Diagram Alur Perancangan Solusi

Setelah dapat menentukan teknologi yang akan digunakan, barulah kemudian projek dapat diinisialisasi dengan membuat sebuah *repository* pada GitHub untuk mempermudah kolaborasi antar anggota kelompok. Kemudian dilanjutkan dengan perancangan database dengan PostgreSQL untuk menentukan setiap tabel dan kolom diperlukan dalam penyimpanan data.

Penulisan kode dimulai dari membuat *script* untuk membuat tabel dan kolom secara otomatis saat program mulai dijalankan. Kemudian dilanjutkan dengan menulis kode untuk setiap *service* yang ada mulai dari *user* *service*, *photos*, dan *comments*. Setelah semua kode service selesai, *testing* dilakukan dengan tiga metode yaitu *Black Box Testing*, *White Box Testing*, dan dengan memanfaatkan *framework* JEST yang terdapat pada bahasa pemrograman Javascript.

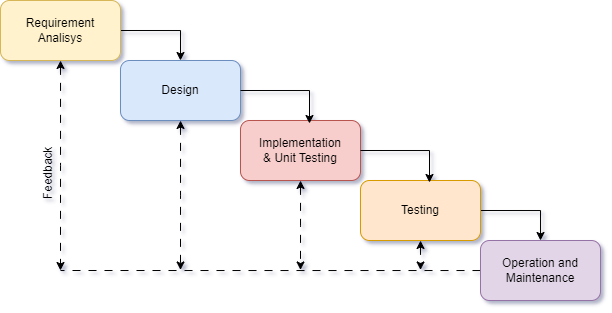
Ketika keseluruhan dari proses *testing* sudah selesai dan tidak terdapat lagi *error*, maka dilanjutkan ke tahap *deployment*. Pada tahap ini, sebuah *platform* *hosting* gratis bernama Railway dimanfaatkan. Terdapat dua mesin yang di *deploy* pada tahap ini yaitu PostgreSQL sebagai database dan NodeJs sebagai *runtime* dari kode program yang telah ditulis. Setelah deployment dilakukan, *testing* terhadap program kembali dilakukan untuk memastikan tidak adanya kesalahan pada tahap *deployment*.

## Perancangan Solusi

Berdasarkan pada permasalahan-permasalahan yang telah disebutkan pada bagian sebelumnya, pengembangan aplikasi yang dapat menjadi platform penyebaran informasi yang cepat dan efisien menjadi solusinya. Demi mewujudkan ide ini, diperlukan suatu teknologi untuk aplikasi agar dapat diakses berbagai macam platform. Penulis menawarkan solusi berupa pembuatan RESTful-API dengan menggunakan NodeJs yang akan di-*deploy* pada layanan *hosting* Railway.

### Metode Pengembangan

Metode pendekatan Waterfall dimanfaatkan sebagai SDLC (Software Development Life Cycle) dalam pembuatan proyek ini. Waterfall digunakan dalam pengembangan perangkat lunak ini karena projek ini memiliki persyaratan yang tetap dan tidak akan berubah ecara signifikan. Selain itu, proyek ini juga memiliki rentang waktu yang ketat serta tim yang sudah terstruktur. Hal tersebut menjadikan pendekatan Waterfall sebagai pendekatan yang paling sesuai dalam pengembangan perangkat lunak MyGram.



**Gambar 3.2** Ilustrasi Metode Pendekatan Waterfall

Terdapat 5 tahapan yang ada pada metode pendekatan Waterfall. Tahapan-tahapan tersebut memiliki tujuan yang berbeda-beda:

1. *Requirement* *Analisis*

Pada tahap ini, keseluruhan informasi mengenai kebutuhan-kebutuhan pengguna terhadap perangkat lunak sudah harus ditentukan. Informasi ini diperoleh dengan cara berdiskusi yang kemudian hasil diskusi tersebut dapat dianalisis dan diolah sehingga kebutuhan pengguna terhadap perangkan lunak didapatkan dan perangkat lunak dapat segera dikembangkan.

1. *Design*

Tahap berikutnya pada metode ini yaitu desain. Secara umum, tahap ini meliputi kepentingan desain teknis seperti lapisan data, bahasa pemrograman, dan layanan pembantu seperti *database* dan *hosting*.

1. *Implementation* & *Unit Testing*

Tahap ini merupakan tahap dimana penulisan kode dimulai. Pembuatan perangkat lunak akan dibagi menjadi modul-modul kecil yang akan dikombinasikan pada tahap berikutnya. Pembagian inilah yang disebut dengan unit yang kemudian akan dilakukan *testing* pada setiap unit yang dibuat.

1. Testing

Penggabungan modul-modul kecil dilakukan pada tahap ini. Berbagai modul yang telah dibuat sebelumnya diintegrasikan dalam suatu system secara keseluruhan dan kemudian dilakukan *testing* kembali terhadap keseluruhan sistem yang telah terintegrasi.

1. Operation and Maintenance

Sistem yang telah dibuat, dites, dan di-*deploy* kemudian akan masuk ke tahap ini. Perangkat lunak yang sudah terbentuk akan dijalankan dan dioperasikan oleh pengguna. Pemeliharaan atau *maintenance* akan dilakukan jika terdapat kesalahan yang perlu diperbaiki atau peningkatan sistem sesuai dengan kebutuhan.

### Kebutuhan Fungsional dan Non Fungsional

Perancangan RESTful-API aplikasi MyGram ini tentunya membutuhkan tools yang sesuai baik itu perangkat keras (hardware) maupun perangkat lunak (software). Spesifikasi yang dibutuhkan diantaranya sebagai berikut:

**Tabel 3.2.** Kebutuhan *hardware*

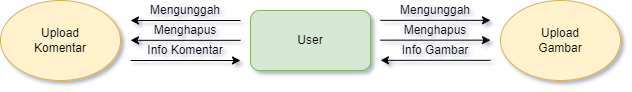
|  |  |
| --- | --- |
| **Komputer/Laptop** | **Spesifikasi** |
| Processor | Setara Intel Core i5 gen4, dengan kecepatan 2.4 GHz atau lebih. |
| RAM | Direkomendasikan 8 GB, minimal 4 GB dengan kecepatan minimal 1333 MHz. |
| Storage | HDD/SSD dengan ruang kosong minimal 25 GB, lebih besar lebih baik. |

**Tabel 3.3.** Kebutuhan *software*

|  |  |  |
| --- | --- | --- |
| **Jenis** | **Software** | **Versi** |
| Sistem Operasi | Windows 10 | 64-bit |
| Text Editor | Visual Studio Code | 1.87.0 |
| Bahasa Pemrograman Back-End | Javascript | - |
| Runtime | NodeJs | 18.17.1 |
| Database | PostgreSQL | 16.0.1 |
| Web Browser | Google Chrome | 124.0.6367.158 |
| Aplikasi pengujian API | Postman | 11.0.7 |
| Layanan Hosting | Railway | - |

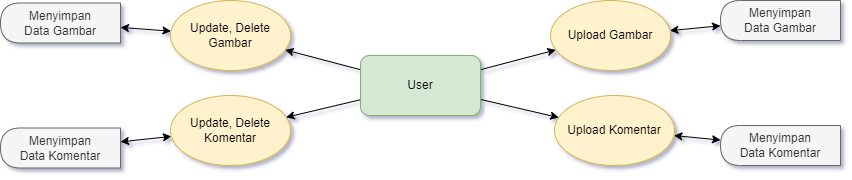
### Perancangan

Sistem dirancang menggunakan model terstruktur, dimana ilustrasi dapat dilihat dalam bentuk Data Flow Diagram dan juga hubungan entitas atau Entity Relational Diagram (ERD). Kedua diagram ini sangat penting pada tahap perancangan sistem untuk dapat mempermudah menentukan alur pengiriman data dan juga dalam melihat hubungan antar entitas atau tabel yang ada pada database.



**Gambar 3.3** Data Flow Diagram (DFD) level 0

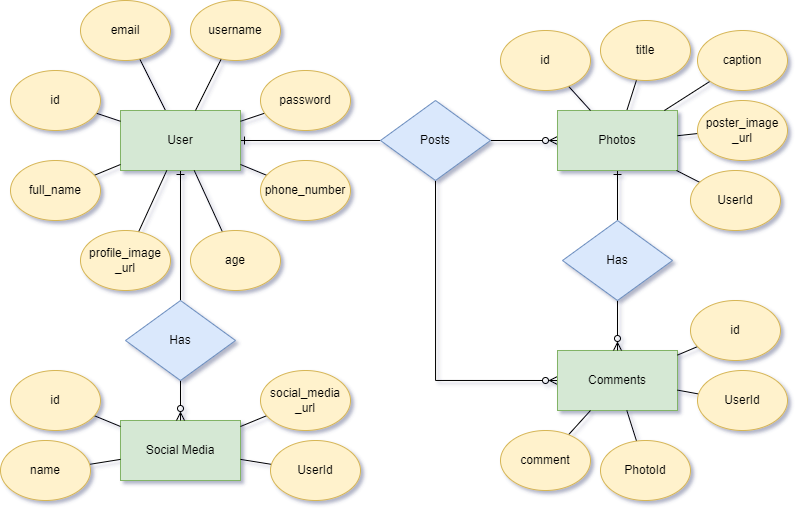
Pada gambar 3.2 yang menunjukkan DFD level dasar atau 0 diatas dapat terlihat bahwa user dapat mengunggah, menghapus, dan melihat informasi mengenai gambar atau komentar. Diagram ini menggambarkan secara umum bagaimana data tersebut akan diolah oleh user.



**Gambar 3.4** Data Flow Diagram (DFD) level 1

Pengembangan dari DFD level 0 dilanjutkan ke DFD level 1, dimana alur data akan terlihat lebih terperinci. Setiap aktivitas yang dilakukan oleh user seperti mengunggah gambar atau komentar akan tersimpan ke dalam database. Begitu pula dengan aktivitas *update* dan *delete* data gambar maupun komentar, yang akan tercatat ke dalam database saat aktivitas tersebut dilakukan.

Data yang tersimpan ke dalam database tentu perlu diberikan ilustrasi berupa diagram juga untuk mempermudah perancangan sistem. Diagram ini disebut dengan Entity Relational Diagram atau ERD. ERD dapat dibuat dengan dua bentuk, yang pertama yaitu dengan node-node dengan berbagai bentuk yang saling terhubung. Bentuk kotak menggambarkan sebuah entitas, atau pada database yang disebut dengan tabel. Bentuk oval menunjukkan atribut-atribut yang dimiliki oleh suatu entitas dan pada database disebut dengan kolom. Kemudian bentuk belah ketupat yang menggambarkan sebuah hubungan atau relasi antar entitas-entitas tersebut.



**Gambar 3.5** Entity Relational Diagram (ERD)

ERD dalam bentuk node dapat dilihat pada gambar 3.5. Terdapat 4 entitas dan 3 relasi pada diagram tersebut. Diagram tersebut menjelaskan entitas user memiliki 8 atribut berupa id, nama, email, username, password, nomor, umur, dan tautan untuk gambar profil. Entitas user terhubung dengan entitas foto melalui relasi posts. Entitas foto memilik 5 atribut berupa id, judul, caption, tautan gambar, dan juga id dari user yang me-posting gambar tersebut. Entitas user juga terhubung dengan entitas komentar dimana entitas ini memiliki atribut berupa id, id pengguna yang memberikan komentar, id foto yang diberikan komentar, dan isi dari komentar tersebut. Secara bersamaan, entitas foto dan komentar juga berhubungan dengan relasi has, dimana setiap foto dapat memiliki komentar masing-masing. Kemudian pada entitas media sosial, dimana setiap entitas user dapat menambahkan media sosial yang mereka miliki melalui tautan media sosial. Entitas ini memiliki 4 atribut berupa id, tautan media sosial, nama, dan juga id dari user yang memiliki media sosial tersebut.

### Metode Pengujian

Pengujian diperlukan agar kualitas dari perangkat lunak yang dibangun terjamin dan sudah merepresentasikan kajian pokok dari spesifikasi, desain, dan penulisan kode. Metode pengujian diharapkan memiliki mekanisme untuk mentukan data uji yang dapat menguji perangkat lunak secara keseluruhan (Completeness of Test) dan memiliki kemungkinan yang tinggi untuk menemukan suatu kesalahan (High Likelihood for Uncovering Erros).

Pengujian dibagi menjadi 2 tahap, unit testing dan system testing. Metode pengujian yang digunakan dalam pengujian aplikasi MyGram adalah White Box Testing yang memanfaatkan framework JEST. JEST merupakan sebuah testing framework yang populer digunakan untuk pengujian aplikasi berbasis JavaScript dan React. JEST memiliki fitur-fitur yang kuat seperti mocking, assertion built-in, parallel testing, dan snapshot testing yang memudahkan pengembang untuk menulis, menjalankan, dan menganalisis hasil pengujian.

Berikut merupakan skenario rencana pengujian yang akan dilakukan pada setiap endpoint:

**Tabel 3.4** Skenario Pengujian User (/users)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Endpoint** | **Method** | **Skenario** | **Respon** |
| 1. | /register | POST | Field tidak lengkap. | 500, Incomplete Data |
| Email yang sudah terdaftarkan. | 500, Email not unique |
| Data sesuai. | 201, Created, data user |
| 2. | /login | POST | Email tidak terdaftar. | 404, Email not found |
| Password tidak sesuai. | 401, Wrong password |
| Data sesuai. | 200, Token |
| 3. | /:userId | PUT | Tanpa token. | 401, Not authenticated |
| UserId params tidak ditemukan. | 404, UserId not found |
| UserId params tidak sesuai dengan userId token. | 403, Not authorized |
| Data sesuai. | 200, Updated, data user |
| 4. | /:userId | DELETE | Tanpa token. | 401, Not authenticated |
| UserId params tidak ditemukan. | 404, UserId not found |
| UserId params tidak sesuai dengan userId token. | 403, Not authorized |
| Data sesuai. | 200, Deleted |

**Tabel 3.5** Skenario Pengujian Foto (/photos)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Endpoint** | **Method** | **Skenario** | **Respon** |
| 1. | / | POST | Field url tidak dalam format url. | 500, Validation error |
| Tanpa token. | 401, Not authentication |
| Data sesuai. | 201, Created, data photo |
| 2. | / | GET | Tanpa token. | 401, Not authenticated |
| Data sesuai. | 200, List photos |
| 3. | /:photoId | PUT | Tanpa token. | 401, Not authenticated |
| UserId data photo tidak sesuai dengan userId token. | 403, Not authorized |
| UserId tidak ditemukan. | 404, PhotoId not found |
| Data sesuai. | 200, Updated, data photo |
| 4. | /:photoId | DELETE | Tanpa token. | 401, Not authenticated |
| UserId data photo tidak sesuai dengan userId token. | 403, Not authorized |
| PhotoId tidak ditemukan. | 404, PhotoId not found |
| Data sesuai. | 200, Deleted |

**Tabel 3.6** Skenario Pengujian Komentar (/comments)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Endpoint** | **Method** | **Skenario** | **Respon** |
| 1. | / | POST | Tanpa token. | 401, Not authenticated |
| Data sesuai. | 201, Created |
| 2. | / | GET | Tanpa token. | 401, Not authenticated |
| Data sesuai. | 200, List comment |
| 3. | /:commentId | PUT | Tanpa token. | 401, Not authenticated |
| UserId data comment tidak sesuai dengan userId token. | 403, Not authorized |
| CommentId tidak ditemukan. | 404, CommentId |
| Data sesuai | 200, Updated, data comment |
| 4. | /:commentId | DELETE | Tanpa token. | 401, Not authenticated |
| UserId data comment tidak sesuai dengan userId token. | 203, Not authorized |
| CommentId tidak ditemukan. | 404, CommentId not found |
| Data sesuai | 200, Deleted |

**Tabel 3.7** Skenario Pengujian Media Sosial (/socialmedias)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Endpoint** | **Method** | **Skenario** | **Respon** |
| 1. | / | POST | Url media sosial tidak dalam bentuk url. | 500, Validation error |
| Tanpa token. | 401, Not authenticated |
| Data sesuai | 201, Created |
| 2. | / | GET | Tanpa token. | 401, Not authenticated |
| Data sesuai. | 200, List socialMedia |
| 3. | /:socialMediaId | PUT | Tanpa token. | 401, Not authenticated |
| UserId data sosialMedia tidak sesuai dengan userId token. | 403, Not authorized |
| SocialMediaId tidak ditemukan. | 404, SocialMediaId not found |
| Data sesuai. | 200, Updated, data socialMedia |
| 4. | /:socialMediaId | DELETE | Tanpa token. | 401, Not authenticated |
| UserId data sosialMedia tidak sesuai dengan userId token. | 203, Not authorized |
| SocialMediaId tidak ditemukan. | 404, SocialMediaId not found |
| Data sesuai. | 200, Deleted |

Pada setiap *endpoint* yang ada pada masing-masing *service* dengan yang berbeda akan diuji dengan beberapa skenario seperti akses tanpa menggunakan token, mengakses data yang tidak boleh diakses, data tidak sesuai validasi, dan data yang sudah sesuai. Pengujian dilakukan dengan menggunakan *framework* pengujian JEST dan dijalankan melalui IDE yang digunakan (Visual Studio Code).

# BAB IV HASIL DAN PEMBAHASAN

## Hasil

Berdasarkan pada diagram alur perancangan solusi sebelumnya dan penggunaan metode pengembangan Waterfall didapatkan hasil dari RESTful-API aplikasi MyGram. Aplikasi ini dijalankan secara lokal pada perangkat masing-masing dengan memanfaatkan runtime NodeJs. Jika diinginkan, deploy dapat segera dilakukan kapan saja karena fungsional aplikasi sudah teruji. Berikut penjelasan yang lebih rinci mengenai hasil untuk setiap langkah yang terdapat pada metode Waterfall.

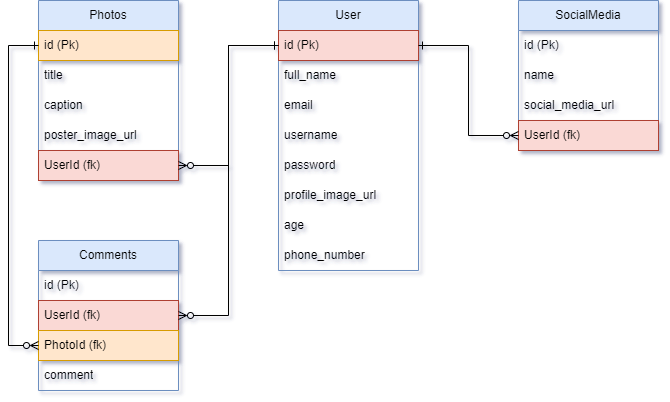
### Requirement Analysis

Keperluan atau kebutuhan dari keseluruhan sistem ditentukan disini. Diskusi merupakan usaha yang dilakukan untuk mendapatkan hasil pada tahap ini. Berdasarkan pada permasalahan yang telah dibahas sebelumnya, hasil yang diperoleh yaitu bahwa diperlukan suatu sistem yang dapat digunakan pada berbagai macam platform. Dengan begitu didapatkan solusi yaitu pembuatan RESTful-API. Secara otomatis, aplikasi yang dibuat hanya akan dalam bentuk API, maka diperlukan sebuah tool untuk melakukan pengujian terhadap sistem yang dibuat.

Selain kebutuhan perangkat lunak, tentu saja diperlukan juga perangkat keras yang akan digunakan dalam pembuatan aplikasi. Karena aplikasi ini tidak terlalu memiliki proses-proses yang kompleks dan scope yang kecil, maka spesifikasi perangkat keras juga tidak terlalu tinggi, cukup menggunakan laptop atau computer dengan spesifikasi yang biasa-biasa saja sudah cukup. Hal yang perlu diingat yaitu penggunaan database yang mengharuskan perangkat memiliki penyimpanan kosong yang cukup besar, tidak hanya untuk menyimpan data, tetapi juga untuk menjalankan database tersebut.

### Design

Hasil dari tahap ini yaitu penentuan alat-alat yang akan digunakan dalam pembuatan aplikasi. Bahasa pemrograman yang digunakan yaitu JavaScript dengan menggunakan runtime berupa NodeJs. Pengkodean akan dilakukan dengan Visual Studio Code yang memiliki banyak fitur ekstensi yang dapat membantu. Sebagai alat pengujian API diperlukan sebuah aplikasi bernama Postman. Database yang digunakan yaitu PostgreSQL. Kemudian untuk hosting yang digunakan yaitu Railway versi gratis.



**Gambar 4.1** Relasi Tabel

Tidak sampai disitu, tahap ini juga melanjutkan dengan perancangan database, mengenai tabel-tabel dan kolom-kolom apa saja yang akan terdapat pada database. Tentu pengguna akan memiliki tabelnya sendiri dengan atribut berupa profil dari pengguna tersebut. Tabel foto akan memiliki sebuah foreign-key yang mengacu pada identitas dari user yang mengunggah foto tersebut. Sama halnya dengan tabel sosial media yang memiliki foreign-key dari tabel pengguna. Kemudian, untuk tabel komentar memiliki 2 foreign-key yang mengacu pada identitas user yang mengunggah komentar dan juga identitas foto yang akan diberikan komentar. Untuk itu, diperlukan tambahan foreign-key juga pada tabel foto yang mengacu kepada tabel komentar.

### Implementation

Setelah semua kebutuhan dan desain ditentukan, penulisan kode aplikasi dapat mulai dilakukan pada IDE Visual Studio Code. Penulisan kode aplikasi dibagi menjadi beberapa bagian untuk masing-masing fitur. Pembagian tersebut yaitu users, social medias, photos, dan comments. Berikut merupakan contoh reponse dari kode yang telah dibuat untuk method GET.

**Tabel 4.1** Response dari GET photos

|  |
| --- |
| {  "photos": [  {  "id": 3,  "title": "Punya User 2",  "caption": "Punya User 2",  "poster\_image\_url": "google.com",  "UserId": 4,  "createdAt": "2024-05-10T03:07:30.619Z",  "updatedAt": "2024-05-10T03:13:40.528Z",  "Comments": [  {  "comment": "so cooll!",  "User": {  "username": "maulanada"  }  },  {  "comment": "tysm<3",  "User": {  "username": "maulanad77"  }  },  {  "comment": "no probss",  "User": {  "username": "maulanada"  }  }  ],  "User": {  "id": 4,  "username": "maulanad77",  "profile\_image\_url": "google.com"  }  },  {  "id": 1,  "title": "Gambar User 1",  "caption": "Gambar User 1",  "poster\_image\_url": "google.com",  "UserId": 1,  "createdAt": "2024-05-10T03:06:05.086Z",  "updatedAt": "2024-05-10T03:06:05.086Z",  "Comments": [ |

**Tabel 4.2** Response dari GET photos (lanjutan)

|  |
| --- |
| {  "comment": "what a nice pic",  "User": {  "username": "maulanad77"  }  }  ],  "User": {  "id": 1,  "username": "maulanada",  "profile\_image\_url": "google.com"  }  }  ]  } |

Tabel diatas memperlihatkan response dari GET photos dimana setiap foto yang ada akan ditampilkan dan memuat informasi-informasi mendetail seperti kapan foto itu diunggah, judul dari foto tersebut, dan caption yang ada. Selain itu, informasi mengenai user yang mengunggah foto tersebut dan juga komentar-komentar yang ada pada foto tersebut juga ditampilkan.

**Tabel 4.3** Response dari GET comments

|  |
| --- |
| {  "comments": [  {  "id": 4,  "UserId": 1,  "PhotoId": 3,  "comment": "no probss",  "createdAt": "2024-05-10T03:18:32.998Z",  "updatedAt": "2024-05-10T03:18:32.998Z",  "Photo": {  "id": 3,  "title": "Punya User 2",  "caption": "Punya User 2",  "poster\_image\_url": "google.com"  },  "User": {  "id": 1,  "username": "maulanada",  "profile\_image\_url": "google.com",  "phone\_number": "085200001111"  }  },  {  "id": 2,  "UserId": 1,  "PhotoId": 3,  "comment": "so cooll!",  "createdAt": "2024-05-10T03:17:02.128Z",  "updatedAt": "2024-05-10T03:17:02.128Z",  "Photo": {  "id": 3,  "title": "Punya User 2",  "caption": "Punya User 2",  "poster\_image\_url": "google.com"  },  "User": {  "id": 1,  "username": "maulanada",  "profile\_image\_url": "google.com",  "phone\_number": "085200001111"  }  }  ]  } |

Pada GET comments, juga hampir sama dengan GET photos. Daftar komentar yang ditampilkan hanya komentar dari user yang sedang mengakses API tersebut. Contohnya pada table diatas dimana user yang mengakses API tersebut adalah user dengan id 1, maka komentar yang ditampilkan hanyalah komentar yang diunggah oleh user dengan id 1.

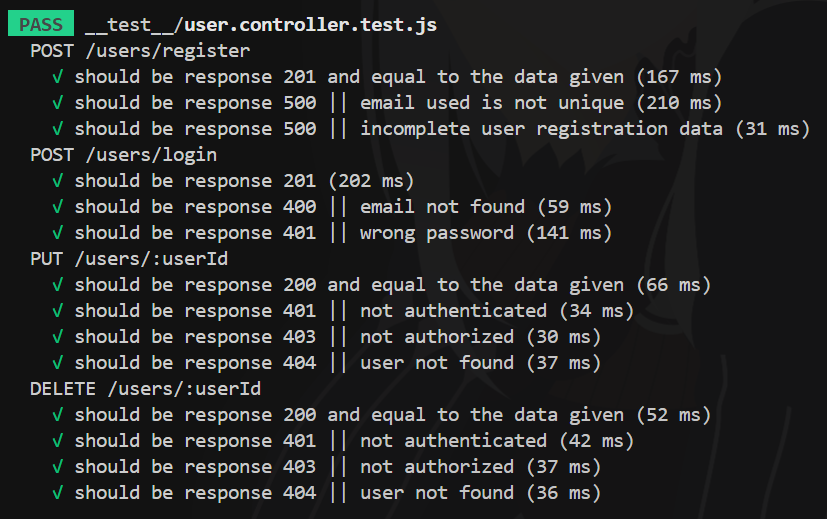
**Tabel 4.5** Response dari GET socialMedias

|  |
| --- |
| {  "social\_medias": [  {  "id": 2,  "name": "ig",  "social\_media\_url": "ig.com/user1",  "UserId": 1,  "createdAt": "2024-05-10T03:31:40.208Z",  "updatedAt": "2024-05-10T03:31:40.208Z",  "User": {  "id": 1,  "username": "maulanada",  "profile\_image\_url": "google.com"  }  },  {  "id": 1,  "name": "youtube",  "social\_media\_url": "youtube.com/user1",  "UserId": 1,  "createdAt": "2024-05-10T03:31:19.205Z",  "updatedAt": "2024-05-10T03:34:22.718Z",  "User": {  "id": 1,  "username": "maulanada",  "profile\_image\_url": "google.com"  }  }  ]  } |

Sama halnya dengan komentar, response dari GET socialMedias juga hanya menampilkan media sosial yang diunggah oleh user yang mengakses API tersebut. Dimana user dengan id 1 yang mengakses API, dan response yang diberikan merupakan daftar dari media sosial yang memiliki id user 1.

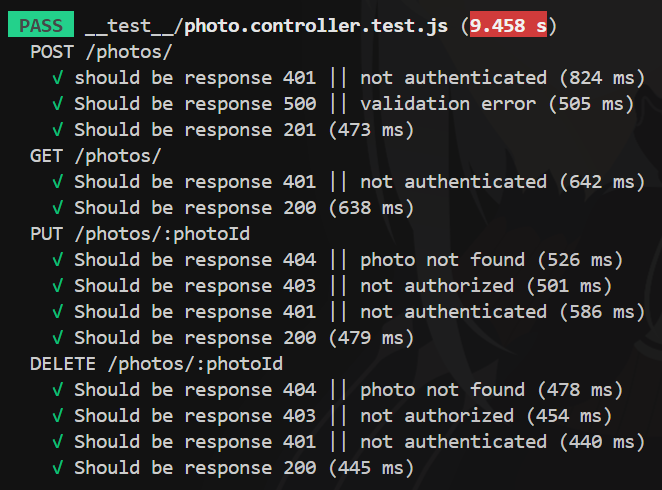
### Testing

Berdasarkan dari metode pengujian pada Bab III, didapatkan hasil pada setiap unit testing dan juga system testing. Unit testing dilakukan 4 kali untuk masing-masing unit berupa user, photos, comments, dan juga socialMedias. Kemudian untuk testing pada sistem yang sudah terintegrasi antar unit dilakukan setelah semua unit testing selesai dan sudah tidak ada lagi kesalahan.

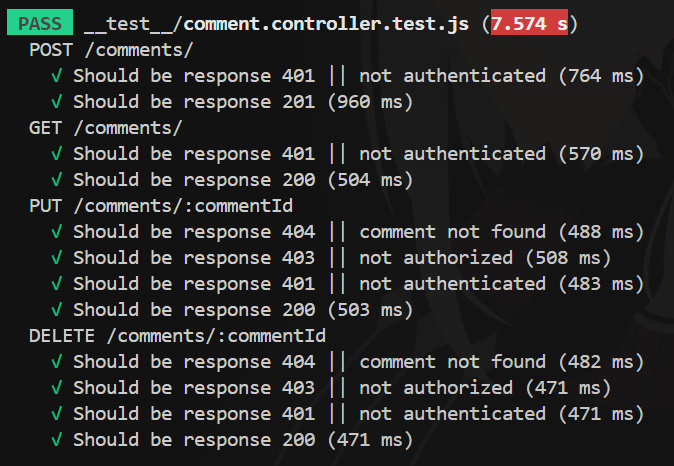


**Gambar 4.2** Hasil unit testing dari user

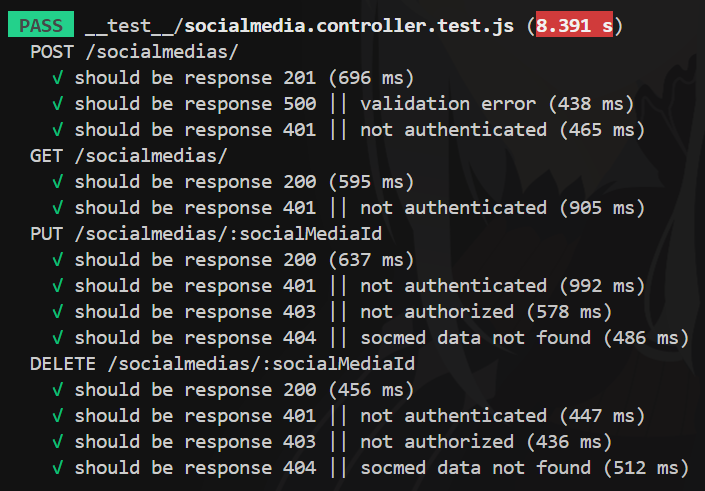
Hasil dari testing pada unit user dapat terlihat pada gambar 4.2. Gambar tersebut merupakan keluaran dari terminal IDE pada saat menajalan test dengan JEST. Informasi yang dapat dilihat yaitu mengenai keberhasilan setiap unit yang dites, ditandai dengan centang berwarna hijau pada bagian kiri. Kecepatan penyelesaian dari testing untuk setiap unit juga dapat dilihat pada bagian kanan dimana terdapat waktu yang ditempuh dalam menyelesaikan tes tersebut dalam satuan ms. Secara keseluruhan, setiap unit pada service diuji coba, dan jika semua unit memenuhi ekspektasi, maka terdapat label hijau bertuliskan “Pass” pada kiri atas yang menunjukkan bahwa testing telah berhasil. Untuk testing pada service lain dapat dilihat pada gambar-gambar dibawah.



**Gambar 4.3** Hasil unit testing dari photo

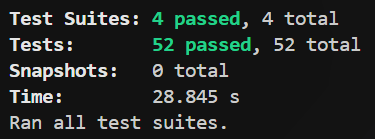


**Gambar 4.4** Hasil dari unit testing comment



**Gambar 4.5** Hasil unit testing dari socialmedia

Setelah semua unit tes selesai dan memenuhi ekspektasi, hasil dari tes keseluruhan sistem dapat terlihat pada gambar dibawah. Terlihat terdapat 52 unit testing dan 4 service testing yang semuanya telah berlabel “passed”. Selain itu waktu yang ditempuh dalam melakukan testing keseluruhan sistem juga terlihat.



**Gambar 4.6** Hasil dari keseluruhan sistem testing

### Operation and Maintenance

Setelah semua kode dapat berjalan dengan baik, maka dilanjutkan dengan deployment. Proses ini dilakukan dengan memanfaatkan layanan hosting gratis bernama Railway. Terdapat dua mesin yang dijalankan pada layanan Railway yaitu database, dan aplikasi. Tautan dari produk yang telah ter-deploy adalah sebagai berikut:

<https://injs06-final-project2.up.railway.app/>

Namun, dikarenakan layanan gratis ini memiliki akses yang terbatas, maka terpaksa untuk di-takedown. Meski begitu, kode yang telah dibuat tetap dapat berjalan dengan baik saat dijalankan pada localhost.

## Pembahasan

Setelah perancangan dan pengembangan sistem selesai dilakukan, didapatkan hasil berupa RESTful-API aplikasi MyGram yang dapat membantu masyarakat dalam menyebarkan informasi secara cepat dan efisien. Pembuatan aplikasi ini dapat membantu para pelaku usaha untuk mempromosikan usaha, seniman dalam mengekspos kreativitas, dan sebagainya. Sebuah RESTful-API yang menjadikan aplikasi ini dapat diakses berbagai macam platform cukup menggunakan aplikasi browser seperti Google Chrome, Firefox, Microsoft Edge membuat fleksibilitas dari aplikasi ini cukup tinggi sehingga seluruh lapisan dan kalangan masyarakat dapat memanfaatkannya.

Pemanfaatan seluruh metode yang digunakan mulai dari analisis SWOT untuk mencari solusi hingga metode pendekatan Waterfall dalam pengembangan sistem terbukti dapat menyelesaikan masalah yang dihadapi. Analisis SWOT bermanfaat dalam menentukan kesempatan dan ancaman yang bisa terjadi dengan menganalisis kelebihan dan kekurangan dari permasalahan. Hal ini membuktikan meskipun analisis SWOT merupakan metode analisis yang sederhana tetapi mampu memberikan solusi. Begitu juga dengan pendekatan Waterfall dimana tahap awal dari aplikasi masih sederhana dan belum memiliki fitur-fitur yang rumit, sehingga metode pengembangan yang sederhana dapat membantu meningkatkan produktivitas pengembangan aplikasi.

Tidak hanya itu, penggunaan framework testing yang tepat juga sangat membantu dalam meningkatkan efisiensi pengembangan aplikasi. Hal ini menunjukkan bahwa teknologi sudah sangat berkembang dan dengan pengetahuan yang mencukupi hampir semua hal dapat dilakukan dengan lebih efisien.

# BAB V PENUTUP

## Kesimpulan

Pengembangan perangkat lunak yang memanfaatkan analisis SWOT dan metode pendekatan Waterfall yang dilaksanakan dalam program Studi Independen Kampus Merdeka pada PT Hacktivate Teknologi Indonesia yang bekerja sama dengan Kementrian Pendidikan, Kebudayaan, Riset, dan Teknologi berjalan dengan lancar dan menghasilkan produk luaran berupa API dengan arsitektur RESTful yang dapat dikembangkan lebih lanjut.

## Saran

API yang dibangun sudah berjalan dengan baik, meskipun begitu masih terdapat beberapa kekurangan yang terdapat pada api tersebut, diantaranya:

1. Belum adanya pengurutan data, sehingga data yang ditampilkan akan berdasarkan data yang terakhir terupdate. Hal ini dapat diperbaiki dengan mengurutkan data terlebih dahulu sebelum menampilkannya sebagai respon.
2. Penyimpanan gambar masih berupa url yang menjadikan gambar tidak benar-benar dimiliki oleh aplikasi yang sudah dibentuk. Hal ini akan berakibat fatal jika suatu saat layanan yang menyimpan gambar tersebut mengalami kendala.

## Keberlanjutan

Keberlanjutan dari aplikasi dapat berupa enhancement dan juga integration. Pada sisi enhancement atau peningkatan, didapatkan dua poin berdasarkan dari saran yang ada, yaitu:

1. Pembuatan fungsi yang berguna untuk mengurutkan data yang akan ditampilkan. Pengurutan bisa berdasarkan kapan data tersebut dibuat.
2. Penambahan sebuah layanan penyimpanan baru yang dapat menyimpan gambar agar tautan gambar yang tersimpan dapat dilacak dengan mudah.

Selain dari poin yang didasarkan pada saran, dapat juga dilakukan pengintegrasian dimana API yang telah terbuat dihubungkan dengan aplikasi Front-End yang dapat memperindah tampilan yang ada. Tidak hanya itu, gambar yang tersimpan juga akan jauh lebih baik jika dapat benar-benar ditampilkan, tidak hanya dalam bentuk tautan.

# DAFTAR PUSTAKA

Daftar pustaka merupakan daftar referensi dari semua bahan atau literatur, seperti artikel ilmiah jurnal atau prosiding, buku teks, skripsi, tesis, disertasi, dan sebagainya, yang dirujuk pada laporan kerja praktik. Cara penulisan daftar Pustaka dan perujukan ke sumber pustaka yang digunakan dalam laporan kerja praktik adalah adaptasi sistem referensi [**APA *Style***,](https://apastyle.apa.org/style-grammar-guidelines/references/examples) yang merupakan sistem referensi nama dan tahun. Dalam sistem ini, semua referensi dalam daftar Pustaka diurutkan berdasarkan nama belakang penulis dan tahun publikasi pustakanya. Ketika sebuah perujukan dilakukan terhadap pustaka tertentu pada laporan kerja praktik, maka nama belakang penulis dan tahun publikasi pustaka tersebut harus dicantumkan setelah kalimat yang dirujuk pada laporan kerja praktik. Hal ini disebut juga sebagai sitasi.

# LAMPIRAN

Lampiran dapat digunakan untuk menyajikan informasi penting yang jika diletakkan di bagian utama dapat mengganggu pembaca untuk menangkap alur argumentasi tulisan dengan mudah. **Lampiran wajib** laporan kerja praktik dapat dibagi menjadi Lampiran untuk kerja praktik Reguler dan Lampiran untuk kerja praktik MBKM. Lampiran wajib lebih detail dapat dilihat *list* dibawah ini:

Kerja Praktik REGULER

1. Surat Balasan Perusahaan
2. Surat Tugas kerja praktik
3. Catatan Harian aktifitas harian selama kerja praktik beserta Foto/Dokumentasi kegiatan di tempat kerja praktik dari hasil ekspor *logbook* pada sistem
4. Lembar Konsultasi Pembimbing Lapangan, format sesuai [**LAMPIRAN 3**](https://docs.google.com/document/d/1_T_4hQtqnS6po6bBqU_8gzsdrO6GTz1X/edit?usp=sharing&ouid=112671539057094063560&rtpof=true&sd=true).
5. Bukti konsultasi dosen pembimbing (dapat dibuktikan dengan *screenshoot* bimbingan pada sistem). **Bukan foto bersama dengan dosen pembimbing**
6. Dokumentasi Seminar (Link Vlog)
7. Daftar Relasi, format sesuai [**LAMPIRAN 6**](https://docs.google.com/document/d/1Cmow34idCNpFlDNmerOgH8gNVejjQMf4/edit?usp=sharing&ouid=112671539057094063560&rtpof=true&sd=true).
8. Daftar pengetahuan dan keterampilan yang dipelajari selama proses kerja praktik, format sesuai [**LAMPIRAN 7**](https://docs.google.com/document/d/1FLQ-t6DSk8i1AFjtf5X4mnKDqi3WKAfB/edit?usp=sharing&ouid=112671539057094063560&rtpof=true&sd=true).
9. Surat Keterangan selesai melaksanakan kerja praktik.
10. Lembar Penilaian Pembimbing Lapangan sesuai [**LAMPIRAN 8**](https://docs.google.com/document/d/1u1oZtV3HbaUedz7j7JXzLm3Z9TuWMDnn/edit?usp=sharing&ouid=112671539057094063560&rtpof=true&sd=true).
11. Dokumentasi kegiatan kerja praktik (foto bukti kegiatan selama di lokasi kerja praktik)

Kerja Praktik MBKM

1. Bukti Penerimaan / Pengumuman Penerimaan
2. Surat Tugas kerja praktik
3. Catatan Harian aktifitas harian selama kerja praktik beserta Foto/Dokumentasi kegiatan di tempat kerja praktik dari hasil ekspor *logbook* pada sistem
4. Dokumentasi kegiatan inti seperti proses pembuatan produk, rapat dengan anggota ataupun mentor/pembimbing lapangan, kegiatan lapangan, survey ataupun observasi, dan pertemuan dengan *stakeholder*. Untuk bukti pelaksanaan kegiatan daring, **wajib *on camera***.
5. Bukti konsultasi dosen pembimbing (dapat dibuktikan dengan *screenshoot* bimbingan pada sistem). **Bukan foto bersama dengan dosen pembimbing**
6. Dokumentasi Seminar (Link Vlog)
7. Daftar Relasi, format sesuai [**LAMPIRAN 6**](https://docs.google.com/document/d/1Cmow34idCNpFlDNmerOgH8gNVejjQMf4/edit?usp=sharing&ouid=112671539057094063560&rtpof=true&sd=true).
8. Daftar pengetahuan dan keterampilan yang dipelajari selama proses kerja praktik, format sesuai [**LAMPIRAN 7**](https://docs.google.com/document/d/1FLQ-t6DSk8i1AFjtf5X4mnKDqi3WKAfB/edit?usp=sharing&ouid=112671539057094063560&rtpof=true&sd=true).
9. Sertifikat dan nilai akhir dari kegiatan yang di ikuti.