

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



Disusun oleh

Nama : Maulana Daffa Ardiansyah

NIM : 123200130

PROGRAM STUDI INFORMATIKA

JURUSAN INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”

YOGYAKARTA

2024

HALAMAN PENGESAHAN

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

Pembimbing Lapangan

Dr. Novrido Charibaldi, S.Kom., M.Kom.

NIK . 2 6811 96 0066 1

Arif Santoso

Mengetahui

Koordinator Kerja Praktik JIF

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

NIP. 19840218 201803 1 001

KATA PENGANTAR

Puji syukur kehadiran 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
HALAMAN PERSETUJUAN	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN.....	viii
ABSTRAK	9
BAB I PENDAHULUAN	10
1.1 Latar Belakang	10
1.2 Rumusan Masalah	11
1.3 Tujuan	11
1.4 Manfaat	11
BAB II PROFIL INSTANSI KERJA PRAKTIK	12
2.1 Profil Instansi	12
2.2 <i>Job Description</i>	13
BAB III METODOLOGI	16
3.1 Analisis Permasalahan	16
3.2 Diagram Alur Perancangan Solusi	17
3.3 Perancangan Solusi	18
3.3.1 Metode Pengembangan	18
3.3.2 Kebutuhan Fungsional dan Non Fungsional	20
3.3.3 Perancangan.....	21
3.3.4 Metode Pengujian	23
BAB IV HASIL DAN PEMBAHASAN	27
4.1 Hasil	27
4.1.1 Requirement Analysis	27
4.1.2 Design	28
4.1.3 Implementation	29

4.1.4	Testing	32
4.1.5	Operation and Maintenance	35
4.2	Pembahasan	36
BAB V	PENUTUP	37
5.1	Kesimpulan	37
5.2	Saran	37
5.3	Keberlanjutan	37
DAFTAR PUSTAKA	39
LAMPIRAN	40

DAFTAR TABEL

Tabel 2. 1 Visi dan Misi Hacktiv8	13
Tabel 2. 2 Daftar Produk Hacktiv8	13
Tabel 2. 3 Timeline pengerjaan luaran kerja praktik.....	14
Tabel 2. 4 Timeline pengerjaan luaran kerja praktik.....	15
Tabel 3. 1 Hasil analisis permasalahan dan daftar solusi yang diusulkan	16
Tabel 3. 2 Kebutuhan hardware	20
Tabel 3. 3 Kebutuhan software.....	20
Tabel 3. 4 Skenario Pengujian User (/users)	24
Tabel 3. 5 Skenario Pengujian Foto (/photos)	24
Tabel 3. 6 Skenario Pengujian Komentar (/comments).....	25
Tabel 3. 7 Skenario Pengujian Media Sosial (/socialmedias)	25
Tabel 4. 1 Response dari GET photos	29
Tabel 4. 2 Response dari GET photos (lanjutan)	30
Tabel 4. 3 Response dari GET comments	30
Tabel 4. 4 Response dari GET comments (lanjutan).....	31
Tabel 4. 5 Response dari GET socialMedias.....	32

DAFTAR GAMBAR

Gambar 2. 1	Logo Hacktiv8	12
Gambar 3. 1	Diagram Alur Perancangan Solusi.....	17
Gambar 3. 2	Ilustrasi Metode Pendekatan Waterfall.....	19
Gambar 3. 3	Data Flow Diagram (DFD) level 0	21
Gambar 3. 4	Data Flow Diagram (DFD) level 1	21
Gambar 3. 5	Entity Relationship Diagram (ERD).....	22
Gambar 4. 1	Relasi Tabel	28
Gambar 4. 2	Hasil unit testing dari user	33
Gambar 4. 3	Hasil unit testing dari photo.....	34
Gambar 4. 4	Hasil unit testing dari comment.....	34
Gambar 4. 5	Hasil unit testing dari socialmedia.....	35
Gambar 4. 6	Hasil dari keseluruhan sistem testing	35

DAFTAR LAMPIRAN

Lampiran 1 Bukti Penerimaan	40
Lampiran 2 Catatan Kegiatan	41
Lampiran 3 Dokumentasi Kegiatan Inti	46
Lampiran 4 Screenshot Sadewa.....	47
Lampiran 5 Daftar Relasi.....	48
Lampiran 6 Daftar Kompetensi.....	50
Lampiran 7 Sertifikat dan Nilai Akhir Kegiatan.....	51

ABSTRAK

Program Studi Independen Bersertifikat dengan nama NodeJs for Back-End Web Development merupakan salah satu program dari Kampus Merdeka yang disediakan oleh PT Hacktivate Teknologi Indonesia (Hacktiv8) dengan tujuan membangun talenta digital kelas dunia yang siap berkontribusi pada kemajuan bangsa. Program ini dilakukan secara daring melalui pembelajaran mandiri, live session, dan pengerjaan proyek akhir. Kurikulum yang diberikan yaitu dimulai dari dasar-dasar back-end, dasar-dasar NodeJs, hingga yang terakhir yaitu deployment. Luaran dari program ini berupa RESTful-API yang dibuat dengan bahasa pemrograman JavaScript dan framework ExpressJs. Dengan memanfaatkan runtime NodeJs, aplikasi dapat berjalan dengan baik dan cepat. Permasalahan yang diangkat pada proyek ini yaitu dimana persebaran informasi terjadi dengan sangat cepat dan masyarakat belum sepenuhnya dapat beradaptasi akan hal tersebut. Metode analisis SWOT berperan dalam menganalisa permasalahan tersebut untuk menentukan solusi. Setelah solusi ditemukan, pengembangan aplikasi dilakukan dengan pendekatan Waterfall dimana tahapannya dimulai dari Requirement Analysis, Design, Implementation, Testing, dan Operation. Pengujian aplikasi ini dilakukan dengan menggunakan framework testing bernama JEST yang tersedia pada bahasa pemrograman JavaScript. Dengan memanfaatkan framework ini, dapat sangat memudahkan dalam pengujian aplikasi yang dibuat. Hasil akhir dari proyek ini adalah sebuah RESTful-API aplikasi bernama MyGram yang dapat membagikan informasi foto dan memberikan komentar untuk setiap foto. Hal ini dapat berguna bagi banyak kalangan dari pelaku usaha yang ingin mempromosikan usahanya hingga seorang seniman yang ingin mengekspos hasil karya mereka.

Kata kunci: Back-End, RESTful-API, SWOT, Waterfall, NodeJs

BAB I

PENDAHULUAN

1.1 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 (Ashari Y, 2022).

Kesempatan inilah yang dapat dikembangkan dengan teknologi digital untuk mempermudah pelaku bisnis dalam beradaptasi dengan teknologi masa ini (Akhmad K, 2021). Salah satunya adalah dengan membangun sebuah sistem yang dapat menjadi sebuah alat bagi mereka yang ingin mempromosikan usaha, *personal branding*, ataupun seniman yang ingin membagikan kreativitas mereka (Ardianto D, 2020).

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 (Ehsan A, 2022).

1.2 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?

1.3 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.

1.4 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

2.1 Profil Instansi

PT. Hacktivate Teknologi Indonesia atau yang biasa dikenal dengan Hacktiv8, seperti namanya, merupakan sebuah perusahaan yang bergerak di bidang teknologi. Hacktiv8 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 Indonesia 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
<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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 pihak untuk memajukan industri teknologi di Indonesia.

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

Tabel 2. 2 Daftar Produk Hacktiv8

Bootcamp	Lainnya
<ul style="list-style-type: none"> Pengembangan Web: Front-End, Back-End, Full-Stack. Pemrograman Mobile: Android, iOS. Data Science: Machine Learning, Artificial Intelligence. Desain: UI/UX Design. 	<ul style="list-style-type: none"> 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.

2.2 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 teknis 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. 3 Timeline pengerjaan luaran kerja praktik

Penulis	Rekan Kelompok
<ul style="list-style-type: none"> • Membuat <i>Photos Service</i> • Membuat <i>Comments Service</i> • Membuat <i>script testing</i> untuk <i>Photos Service</i> • Membuat <i>script testing</i> untuk <i>Comments Service</i> 	<ul style="list-style-type: none"> • Membuat script migrasi Database • Membuat User Service • Membuat <i>script testing</i> untuk <i>User Service</i> • Melakukan <i>Deployment</i>

Tabel 2. 4 Timeline pengerjaan luaran kerja praktik

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

BAB III

METODOLOGI

3.1 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 terdapat kurangnya kemampuan masyarakat dalam beradaptasi pada kecepatan tersebut. Kedua hal inilah yang dapat dianalisa untuk mendapatkan solusi yang relevan.

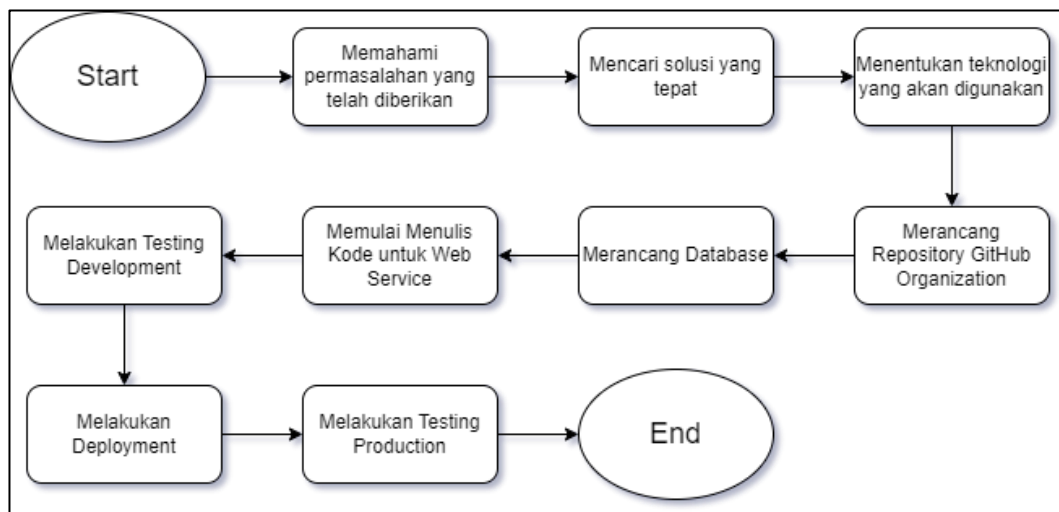
Framework atau kerangka kerja yang digunakan dalam menganalisis permasalahan pada kasus ini adalah *Framework* SWOT. *Framework* ini sudah banyak digunakan organisasi dalam berbagai macam kasus untuk menganalisa kekurangan dan kelebihan dari internal maupun external (Mukhlisin A, 2020). SWOT sendiri 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.
Adanya perbedaan perangkat yang sangat bervariasi di kalangan masyarakat.	Membuat sistem dapat diakses dari berbagai macam platform dan perangkat yang berbeda beda.

3.2 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 pemrograman, *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 proyek 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 *White Box Testing* dengan memanfaatkan *framework* JEST yang terdapat pada bahasa pemrograman Javascript.

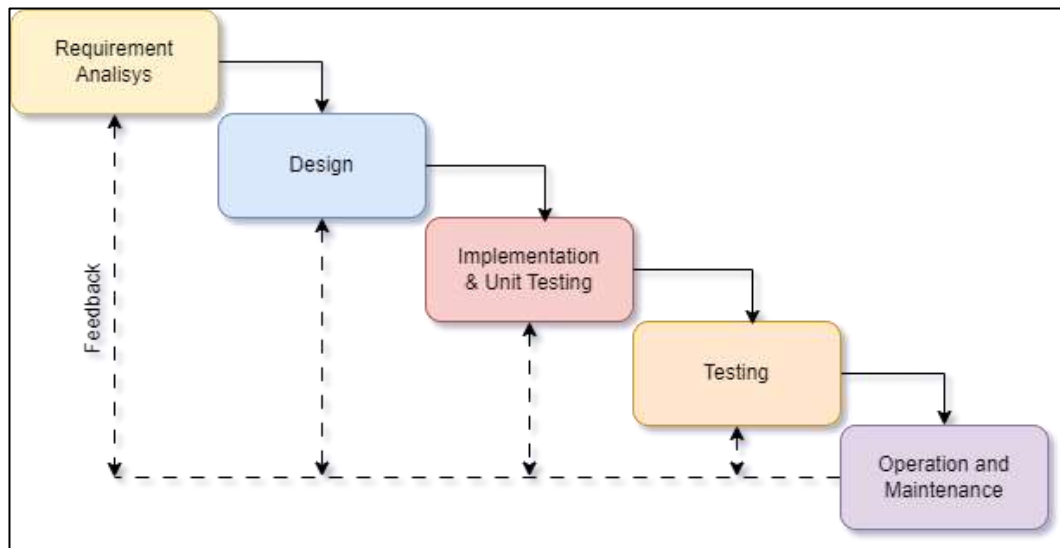
Ketika keseluruhan dari proses *testing* sudah selesai dan tidak terdapat lagi *error*, maka dilanjutkan ke tahap *deployment*. Pada tahap ini, sebuah *platform hosting* gratis bernama Railway dimanfaatkan. Terdapat dua mesin yang di *deploy* pada tahap ini yaitu PostgreSQL sebagai database dan NodeJs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

3.3 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. RESTful-API merupakan arsitektur yang digunakan untuk merancang layanan dikonsumsi di berbagai platform. Keadaan *stateless* dan kesiapan untuk dikonsumsi saja merupakan sebuah standar yang diikuti secara luas (Ehsan A, 2022).

3.3.1 Metode Pengembangan

Metode pendekatan Waterfall dimanfaatkan sebagai SDLC (*Software Development Life Cycle*) dalam pembuatan proyek ini. Waterfall digunakan dalam pengembangan perangkat lunak ini karena proyek ini memiliki persyaratan yang tetap dan tidak akan berubah secara signifikan. Selain itu, proyek ini juga memiliki rentang waktu yang ketat serta tim yang sudah terstruktur (Thesing T, 2021). 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 Analysis*

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 perangkat lunak didapatkan dan perangkat lunak dapat segera dikembangkan.

2. *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*.

3. *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.

4. 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.

5. 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.

3.3.2 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*

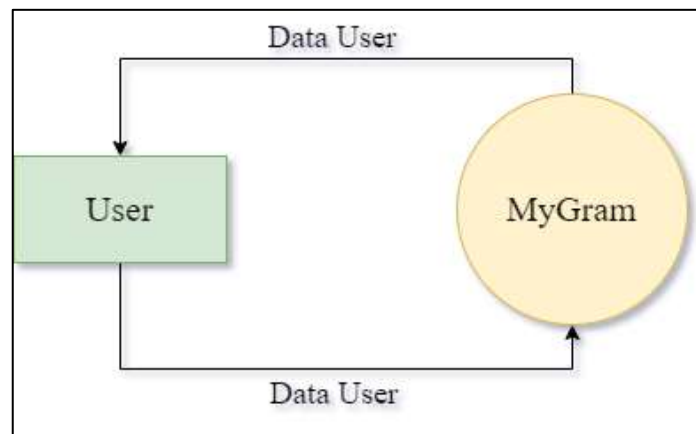
Komputer/Laptop	Spesifikasi
Processor	Setara Intel Core i5 gen4, dengan kecepatan 2.4 GHz atau lebih.
RAM	Direkomendasikan 8 GB, minimal 4 GB dengan kecepatan minimal 1333 MHz.
Storage	HDD/SSD dengan ruang kosong minimal 25 GB, lebih besar lebih baik.

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	-

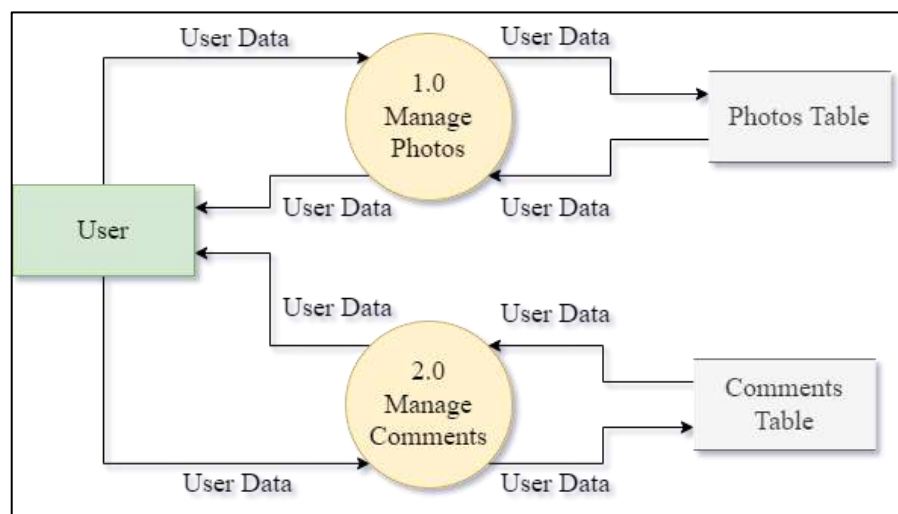
3.3.3 Perancangan

Sistem dirancang menggunakan model terstruktur, dimana ilustrasi dapat dilihat dalam bentuk Data Flow Diagram dan juga hubungan entitas atau Entity Relationship 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

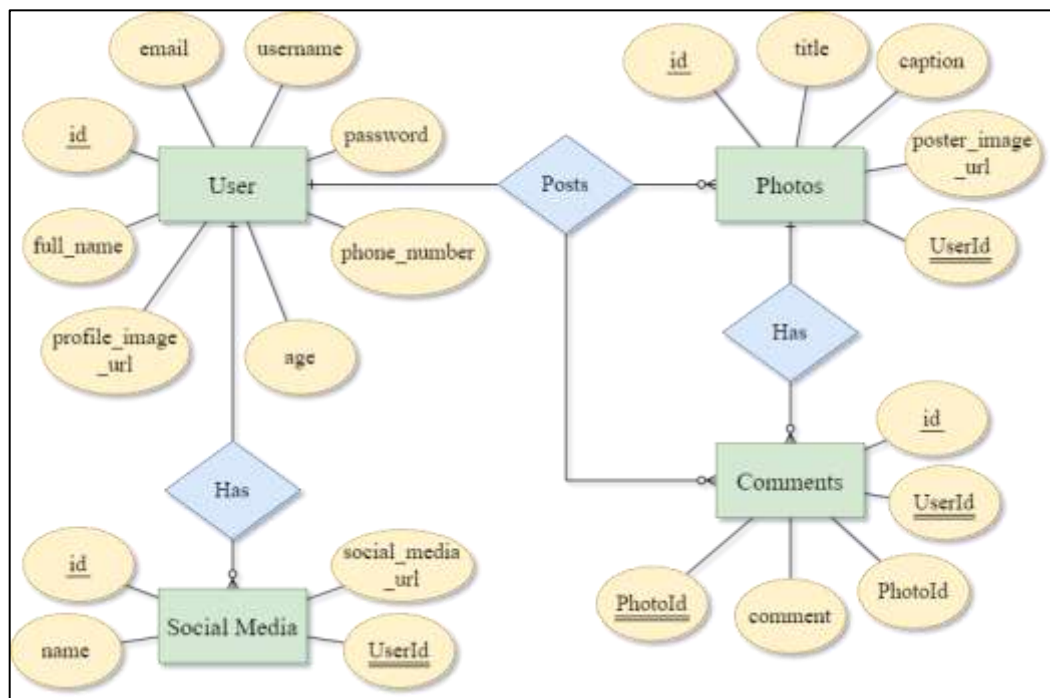
Pada gambar 3.2 yang menunjukkan DFD level dasar atau 0 diatas dapat terlihat bahwa user dapat mengunggah, menghapus, dan melihat informasi mengenai gambar atau komentar. Diagram ini menggambarkan secara umum bagaimana data tersebut akan diolah oleh user.



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 Relationship Diagram atau ERD. ERD dapat dibuat dengan node-node dengan berbagai bentuk yang saling terhubung (Afiifah K, 2022). 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 Relationship 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 memiliki 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.

3.3.4 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 menentukan 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 Errors*) (H, 2020) (Kim M, 2022).

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 (Huong, 2022). 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 terdaftar.	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 socialMedia 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 socialMedia 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

4.1 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.

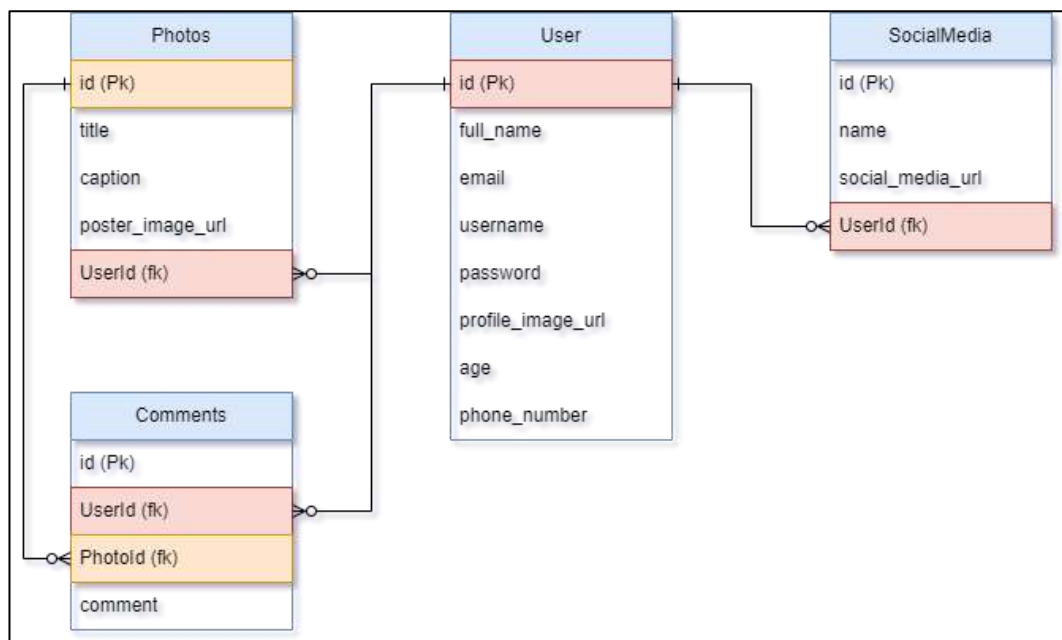
4.1.1 Requirement Analysis

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 komputer 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.

4.1.2 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.

4.1.3 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,
```

Tabel 4. 2 *Response* dari GET *photos* (lanjutan)

```
        "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": [
          {
            "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",
    }
  ]
}
```

Tabel 4. 4 *Response* dari *GET comments* (lanjutan)

```

    "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*

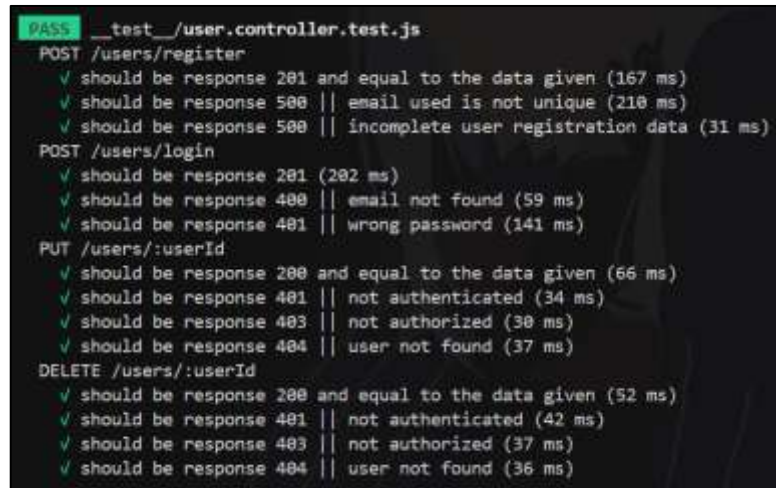
```
{
  "social_medias": [
    {
      "id": 2,
      "name": "ig",
      "social_media_url": "ig.com/user1",
      "UserId": 1,
      "createdAt": "2024-05-10T03:31:40.208Z",
      "updatedAt": "2024-05-10T03:31:40.208Z",
      "User": {
        "id": 1,
        "username": "maulanada",
        "profile_image_url": "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.

4.1.4 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.

A screenshot of a terminal window with a dark background. The text is white and green. At the top, it says 'PASS' in green on a green background, followed by '___test___/user.controller.test.js'. Below this, there are several test cases for the user controller. Each test case starts with a green checkmark and 'should be response' followed by a status code and a description. The test cases are: POST /users/register (3 tests), POST /users/login (3 tests), PUT /users/:userId (4 tests), and DELETE /users/:userId (4 tests). The tests are all successful, as indicated by the green checkmarks and the 'PASS' label at the top.

```
PASS ___test___/user.controller.test.js
POST /users/register
✓ should be response 201 and equal to the data given (167 ms)
✓ should be response 500 || email used is not unique (210 ms)
✓ should be response 500 || incomplete user registration data (31 ms)
POST /users/login
✓ should be response 201 (202 ms)
✓ should be response 400 || email not found (59 ms)
✓ should be response 401 || wrong password (141 ms)
PUT /users/:userId
✓ should be response 200 and equal to the data given (66 ms)
✓ should be response 401 || not authenticated (34 ms)
✓ should be response 403 || not authorized (30 ms)
✓ should be response 404 || user not found (37 ms)
DELETE /users/:userId
✓ should be response 200 and equal to the data given (52 ms)
✓ should be response 401 || not authenticated (42 ms)
✓ should be response 403 || not authorized (37 ms)
✓ should be response 404 || user not found (36 ms)
```

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.

```

PASS __test__/photo.controller.test.js (9.458 s)
  POST /photos/
    ✓ Should be response 401 || not authenticated (824 ms)
    ✓ Should be response 500 || validation error (505 ms)
    ✓ Should be response 201 (473 ms)
  GET /photos/
    ✓ Should be response 401 || not authenticated (642 ms)
    ✓ Should be response 200 (638 ms)
  PUT /photos/:photoId
    ✓ Should be response 404 || photo not found (526 ms)
    ✓ Should be response 403 || not authorized (501 ms)
    ✓ Should be response 401 || not authenticated (586 ms)
    ✓ Should be response 200 (479 ms)
  DELETE /photos/:photoId
    ✓ Should be response 404 || photo not found (478 ms)
    ✓ Should be response 403 || not authorized (454 ms)
    ✓ Should be response 401 || not authenticated (440 ms)
    ✓ Should be response 200 (445 ms)

```

Gambar 4. 3 Hasil unit *testing* dari *photo*

```

PASS __test__/comment.controller.test.js (7.574 s)
  POST /comments/
    ✓ Should be response 401 || not authenticated (764 ms)
    ✓ Should be response 201 (960 ms)
  GET /comments/
    ✓ Should be response 401 || not authenticated (570 ms)
    ✓ Should be response 200 (504 ms)
  PUT /comments/:commentId
    ✓ Should be response 404 || comment not found (488 ms)
    ✓ Should be response 403 || not authorized (508 ms)
    ✓ Should be response 401 || not authenticated (483 ms)
    ✓ Should be response 200 (503 ms)
  DELETE /comments/:commentId
    ✓ Should be response 404 || comment not found (482 ms)
    ✓ Should be response 403 || not authorized (471 ms)
    ✓ Should be response 401 || not authenticated (471 ms)
    ✓ Should be response 200 (471 ms)

```

Gambar 4. 4 Hasil unit *testing* dari *comment*

```
PASS __test__/socialmedia.controller.test.js (8.391 s)
  POST /socialmedias/
    ✓ should be response 201 (696 ms)
    ✓ should be response 500 || validation error (438 ms)
    ✓ should be response 401 || not authenticated (465 ms)
  GET /socialmedias/
    ✓ should be response 200 (595 ms)
    ✓ should be response 401 || not authenticated (905 ms)
  PUT /socialmedias/:socialMediaId
    ✓ should be response 200 (637 ms)
    ✓ should be response 401 || not authenticated (992 ms)
    ✓ should be response 403 || not authorized (578 ms)
    ✓ should be response 404 || socmed data not found (486 ms)
  DELETE /socialmedias/:socialMediaId
    ✓ should be response 200 (456 ms)
    ✓ should be response 401 || not authenticated (447 ms)
    ✓ should be response 403 || not authorized (436 ms)
    ✓ should be response 404 || socmed data not found (512 ms)
```

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.

```
Test Suites: 4 passed, 4 total
Tests:      52 passed, 52 total
Snapshots:  0 total
Time:       28.845 s
Ran all test suites.
```

Gambar 4. 6 Hasil dari keseluruhan *system testing*

4.1.5 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.

4.2 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

5.1 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.

5.2 Saran

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

- a) 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.
- b) 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.

5.3 Keberlanjutan

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

- a) Pembuatan fungsi yang berguna untuk mengurutkan data yang akan ditampilkan. Pengurutan bisa berdasarkan kapan data tersebut dibuat.
- b) 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

- Afiifah K, F. A. (2022). Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literatur Review. *Jurnal INTECH*.
- Akhmad K, P. S. (2021). Pengaruh Penerapan Teknologi Informasi Pada Usaha Mikro Kecil Dan Menengah Di Kota Surakarta. *Sebatik*, 234-240.
- Ardianto D, R. B. (2020). Film Tari; Sebuah Hibridasi Seni Tari, Teknologi Sinema, dan Media Baru. *Jurnal Seni Budaya*, 112-116.
- Ashari Y, S. F. (2022). Peran Mahasiswa Dalam Membantu Adaptasi Teknologi Terhadap Guru Pada Program Kampus Mengajar 1 Di SD Pelita Bangsa Surabaya. *Jurnal Pengembangan Pendidikan Dasar*, Vol. 6, No. 1.
- Ehsan A, A. M. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *MDPI*.
- H, T. (2020). A Practical Approach to JavaScript Testing.
- Huong, N. (2022). Single-Paged Application And Front-End Testing Methods.
- Kim M, X. Q. (2022). Automated test generation for REST APIs: No time to rest yet. *ISSTA*, 289-301.
- Mukhlisin A, P. M. (2020). ANALISIS SWOT DALAM MEMBUAT KEPUTUSAN DAN MENGAMBIL KEBIJAKAN YANG TEPAT. *Journal Research and Education Studies*.
- Thesing T, F. C. (2021). Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project. *Procedia Computer Science*, 746-756.

LAMPIRAN

Lampiran 1 Bukti Penerimaan



SURAT KETERANGAN DITERIMA
NO: 282/H8/OPS-PTP/SK/VIII/2023

Yang bertanda tangan di bawah ini memberi keterangan bahwa:

Nama : MAULANA DAFFA ARDIANSYAH
Nomor Induk Mahasiswa : 123200130
Universitas : Universitas Pembangunan Nasional Veteran Yogyakarta
Program Studi : Informatika
Jenjang Pendidikan : Sarjana/Strata I (S1)

Telah dinyatakan **LOLOS SELEKSI** dan **DITERIMA** di program **Kampus Merdeka**,



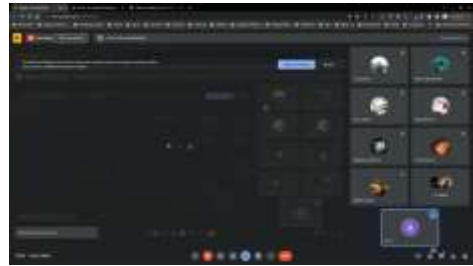
Program MSIB : Studi Independen Bersertifikat
Institusi : Hactiv8 Indonesia
Pilihan Studi / Magang : Node js for Back End Web Developer
Durasi Program : 5 (lima) bulan





Demikian surat keterangan ini kami buat untuk dipergunakan sebagaimana mestinya.

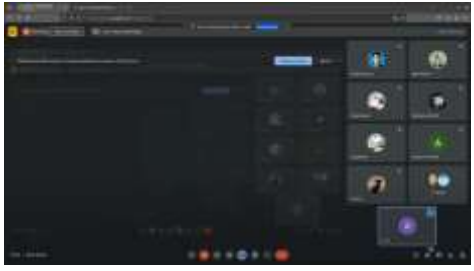


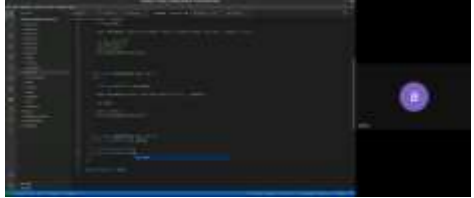
Jakarta Selatan, 7 Agustus 2023






Lutfi Dwimulya
PIC Program

Lampiran 2 Catatan Kegiatan

Minggu	Kegiatan	Dokumentasi
1	Saya belajar banyak pada minggu ini di program msib. Mulai dari soft skill hingga hard skill. Pada studi independen di Hacktiv8 ini, diberikan materi mengenai Engineering Empathy, dimana saya diajarkan bagaimana menjadi orang yang adaptif dan memiliki growth mindset. Sedangkan untuk hard skill, materi yang diberikan mulai dari proses instalasi NodeJS hingga basic-basic dari NodeJs tersebut. Ada pula tugas yang diberikan yaitu untuk membuat summary dari beberapa video yang ada di course.	
2	Minggu ini saya melanjutkan pembelajaran pada pre-requisite course yaitu NodeJs Tingkat Pemula. Pada sisa materi di course tersebut, saya mempelajari mengenai framework ExpressJs dan dikenalkan pada Template Engine berupa Embedded JavaScript atau Ejs. Setelah semua materi selesai pada course tersebut, diberikan contoh pembuatan website mengenai cuaca dimana kita mengambil data dari API dan ditampilkan pada website. Website tersebut dibuat dengan mengimplementasikan ExpressJs dan Ejs kemudian ditambahkan dengan penggunaan CSS agar website memiliki tampilan yang lebih menarik. Pada pembuatan website tersebut diberikan demo pembuatan yang cukup jelas dari awal hingga website tersebut selesai dibuat.	
3	Pada minggu ini, live session baru dimulai karena terdapat kendala pada minggu sebelumnya. Live session dimulai mengikuti materi yang sudah ada di course pada website kode.id. Pada sesi pertama ini mentor memberikan pengenalan mengenai apa itu NodeJs, bagaimana cara menginstallnya, dan bagaimana cara menjalankan file JavaScript pada NodeJs. Kemudian pada sesi kedua, mentor mengajarkan mengenai dasar-dasar dari NodeJs berupa Event Handler, Bufferm Streams, dan diberikan materi tambahan berupa CLI Exploration dimana hal ini akan sangat berguna sebagai Back-End Developer yang akan berurusan dengan server dan database, terutama pada database PostgreSQL yang akan dipelajari mulai pada pertemuan ke-7.	

4	<p>Minggu ini, saya mempelajari mengenai file system pada NodeJs dan bagaimana cara membuat server tanpa menggunakan framework dengan package bawaan pada NodeJs yaitu HTTP server. Setelah mempelajari HTTP server, kemudian mentor mengajarkan mengenai framework yang biasa digunakan pada back-end JavaScript berupa ExpressJs di pertemuan berikutnya. Mentor mengajarkan mulai dari bagaimana cara menginstall ExpressJs hingga pembuatan server dan routing pada NodeJs dengan memanfaatkan framework tersebut.</p>	
5	<p>Pada pertemuan hari senin mentor mengajarkan mengenai file statis dan middleware yang kemudian pada pertemuan hari kamis dilanjutkan dengan Template Engine dan Querystring. Diberikan contoh oleh mentor dalam membuat website dengan memanfaatkan materi-materi pada pertemuan sebelumnya yaitu routing dan middleware yang kemudian dikombinasikan dengan Template Engine Ejs. Diajarkan juga cara menggunakan GET dan POST parameter pada Ejs dalam pembuatan website.</p>	
6	<p>Materi pada minggu ini sudah tidak lagi membahas mengenai framework ataupun template engine. Mentor mulai melanjutkan materi mengenai Database menggunakan PostgreSQL. Dimulai dari membahas hal-hal dasar seperti Relationship DataBase Management System atau RDBMS hingga menginstall sendiri PostgreSQL pada perangkat masing-masing.</p>	
7	<p>Pada minggu ini, dipelajari tools yang dapat mempermudah dalam pengelolaan database sql pada PostgreSQL yaitu PgAdmin. Tools ini memiliki interface yang mudah dipahami sehingga tidak hanya berisi tulisan saja, dimana dalam interface tersebut juga dapat memperlihatkan isi dari database dan table yang ada. Setelah belajar mengenai PgAdmin, kemudian pembelajaran dilanjutkan dengan memahami Rest API, dimana hal tersebut tidak berkaitan langsung dengan database PostgreSQL.</p>	

8	<p>Minggu ini mempelajari mengenai REST API dan bagaimana cara melakukan pembuatan, pembacaan, update, dan penghapusan pada endpoint sebuah API. API yang dipelajari yaitu API dengan bentuk JSON dimana data yang ada pada API didapatkan dari sebuah file .JSON yang ada pada folder. Kemudian bagaimana melakukan CRUD pada data di endpoint tersebut.</p>	
9	<p>Pada minggu kesembilan ini, materi yang dipelajari yaitu mengenai Validasi dan Testing. Validasi yang dimaksud adalah pemberian sebuah token menggunakan package JWT pada saat login berhasil, serta validasi data menggunakan Sequelize. Pada materi testing, digunakan package JEST pada JavaScript yang berguna untuk mengetes sebuah fungsi/unit. Kedua materi ini menurut saya sangat menarik untuk dipejari. Namun, pada minggu ini juga terdapat tugas untuk masing-masing materi. Pada weekend saya baru dapat menyelesaikan tugas pertama yaitu pembuatan API dan authorization menggunakan JWT dan memanfaatkan tools berupa Postman. Untuk tugas kedua baru saya kerjakan pada minggu berikutnya.</p>	
10	<p>Pada minggu ini waktu kebanyakan dihabiskan untuk mengerjakan tugas berupa unit testing, dimana pada tugas tersebut diperintahkan untuk menulis kode yang berguna untuk melakukan testing API dari sebuah aplikasi website yang telah dibuat pada sesi sebelumnya. Selain itu, diperkenalkan juga dua tools baru yaitu Travis-Ci yang digunakan untuk melakukan testing dari aplikasi yang sudah terintegrasi dan Railway yang digunakan untuk melakukan deployment secara gratis.</p>	
11	<p>Pada minggu ini, pengerjaan Final Project 1 berupa Reflection API dimulai. Endpoint-endpoint dibagi menjadi dua bagian untuk dikerjakan bersama dengan rekan kelompok. Setelah mencoba untuk mengerjakan proyek tersebut, terdapat beberapa kendala. Untuk itu, dilanjutkan dengan melakukan mentoring dengan mentor pada hari Jum'at untuk mendapatkan solusi dari kendala yang dialami.</p>	

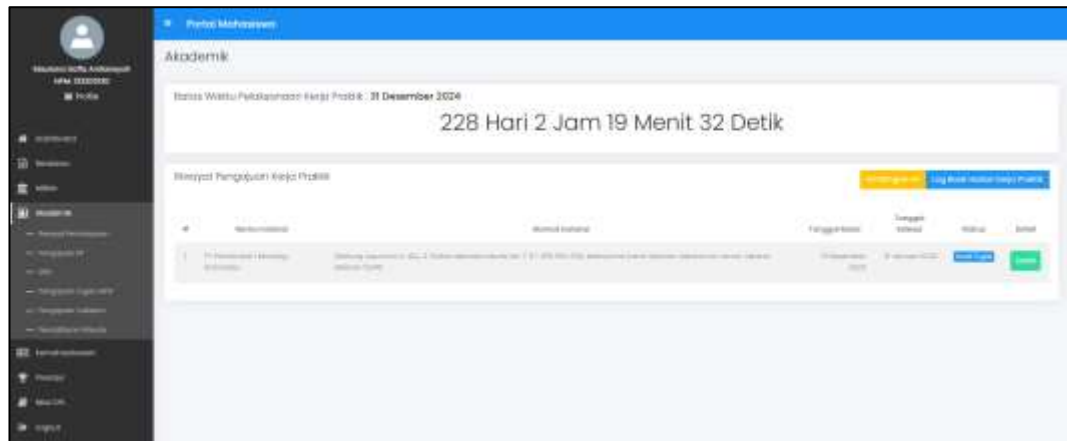
12	Menyelesaikan dan menyempurnakan aplikasi pada Final Project 1. Kemudian dilanjutkan untuk mengikuti meet yang membahas progress masing-masing kelompok dan briefing mengenai Final Project selanjutnya. Setelah itu menginisiasi pengerjaan Final Project 2 dengan membuat repository di GitHub dan menambahkan register dan login API.	
13	Minggu ini saya manfaatkan untuk menyelesaikan Final Project 2 sesuai dengan tugas masing-masing anggota kelompok. Tugas saya yaitu untuk membuat fungsi dari fitur/endpoint sesuai dengan ketentuan Final Project 2. Pada project ini mulai dimanfaatkan ORM yang mana pada Final Project 1 belum digunakan. Penggunaan ORM sangat membantu agar dalam pengiriman data ke dalam database dapat meminimalisir terjadinya kesalahan syntax karena tidak perlu menuliskan query.	
14	Melakukan finishing terhadap Final Project 2 dan melakukan deployment ke platform Railway. Setelah di deploy, dilakukan testing terlebih dahulu untuk mengecek apakah semua endpoint dan fungsi sudah bekerja dengan baik. Kemudian mengsubmit pekerjaan ke dalam Google Classroom. Dilanjutkan dengan mengikuti meet online bersama dengan mentor untuk melakukan briefing terhadap Final Project 3.	
15	Minggu ini dimulai dengan mengerjakan Final Project 3. Mulai dari membuat database, ORM Sequelize, dan juga Sequelize Validasinya. Kemudian melanjutkan mengerjakan fungsi setiap endpoint yang ada.	
16	Melakukan finishing terhadap Final Project 3 dan melakukan deployment ke platform Railway. Setelah di deploy, dilakukan testing terlebih dahulu untuk mengecek apakah semua endpoint dan fungsi sudah bekerja dengan baik. Kemudian mengsubmit pekerjaan ke dalam Google Classroom. Dilanjutkan dengan mengikuti meet online bersama dengan mentor untuk melakukan briefing terhadap Final Project 4.	
17	Memulai pengerjaan pada Final Project 4 dengan menambahkan testing kepada Final Project 2 yang telah dibuat sebelumnya. Karena memiliki beberapa kesibukan dalam mempersiapkan kegiatan KKN,	

	pengerjaan Final Project 4 ini sering saya tunda.	
18	Minggu ini saya lalui dengan melanjutkan pengerjaan Final project 4 yang sekaligus menjadi Final Project terakhir.	
19	Minggu ini melanjutkan dan melakukan finishing terhadap Final Project 4 yang merupakan final project terakhir pada program studi independent. Pada penyelesaian dilakukan mentoring bersama dengan mentor untuk menyelesaikan kendala yang dialami. Kemudian dilanjutkan dengan mengikuti meet terakhir bersama mentor sebagai bentuk perpisahan dan mengucapkan terimakasih karena telah membimbing selama program studi independent.	
20	Minggu ini mulai mengerjakan laporan akhir Kampus Merdeka dan harapannya sebelum minggu depan laporan telah selesai.	

Lampiran 3 Dokumentasi Kegiatan Inti



Lampiran 4 Screenshot Sadewa



Lampiran 5 Daftar Relasi

PT Hacktivate Teknologi Indonesia


1. Erin Gunawan

	Rekan Kelompok Proyek No. HP : +62 858-3443-5321 Linkedin : https://www.linkedin.com/in/eringunawan/ Discord : eringnwn
---	--

2. Arif Santoso

	Mentor Back-End No. HP : - Linkedin : https://www.linkedin.com/in/ariftoteles/ Discord : arif.s
--	--

3. Lutfi Dwimulya

	Operation Manager No. HP : - Linkedin : https://www.linkedin.com/in/ldwimulya/ Discord : ldwimulya
---	---

4. Andika Wahyu Permadi

	Rekan Satu Kelas No. HP : +62 823-8053-9018 Linkedin : https://www.linkedin.com/in/andika-wahyu-permadi-903698222/ Discord : akihira.2002
---	---

5. Moch. Bima Pangestu

	<p style="text-align: center;">Rekan Satu Kelas</p> <p>No. HP : +62 823-4096-8471</p> <p>Linkedin : https://www.linkedin.com/in/bpangestu/</p> <p>Discord : punkestu3000</p>
---	--

6. Restu Oktafiandi

	<p style="text-align: center;">Rekan Satu Kelas</p> <p>No. HP : +62 895-6228-50040</p> <p>Linkedin : https://www.linkedin.com/in/restuoktafiandi/</p> <p>Discord : restuoktafiand</p>
---	---

Lampiran 6 Daftar Kompetensi

No.	Kompetensi
Softskill:	
1.	Skill berkomunikasi yang formal.
2.	Kolaborasi dengan rekan kelompok selama pengembangan aplikasi.
3.	Time management selama masa pengerjaan tugas dan proyek.
4.	Mempresentasikan hasil pekerjaan dengan mentor.
Hardskill:	
1.	Memahami fundamental dasar seorang back-end web developer.
2.	Memahami serta mengimplementasikan framework NodeJs terutama untuk keperluan back-end web application.
3.	Mendesain dan membuat database untuk keperluan web application.
4.	Mengamankan dan mengatur user authentication dan access control untuk application back-end.
5.	Men-debug dan mengoptimalkan kinerja web application yang dibuat.

Lampiran 7 Sertifikat dan Nilai Akhir Kegiatan



SERTIFIKAT MSIB

Diberikan Kepada:

MAULANA DAFFA ARDIANSYAH

Sebagai:

Peserta MSIB Angkatan 5

Telah berhasil menyelesaikan tugasnya di PT. Hacktivate Teknologi Indonesia dalam program Studi Independen dengan project/posisi/kegiatan **Introduction to Node Js** yang diselenggarakan pada tanggal **18 Agustus - 30 Desember 2023**.

PIC Program Kampus Merdeka
PT Hacktivate Teknologi Indonesia,



Lutfi Dwimulya

Jakarta 27 Desember 2023
Chief Evangelist
PT Hacktivate Teknologi Indonesia,




Riza Fahmi

CAPAIAN PEMBELAJARAN PROGRAM


No.	Kompetensi	Definisi Kompetensi	Jam	Nilai Capaian	Deskripsi Capaian
1.	Memahami serta mengetahui fundamental dasar seorang back-end web developer	Back end developer bertanggungjawab atas database, server, dan aplikasi	60 jam	80	Student memiliki pengetahuan atas tugas utama dari seorang back end developer dan peran seorang back end developer dalam proses pengembangan website ataupun aplikasi
2.	Memahami serta mengimplementasikan framework Nodejs terutama backend untuk keperluan web application	Framework Nodejs memiliki beberapa fungsi dalam kompetensi ini framework digunakan untuk Web Development	180 jam	80	Back end developer bertugas untuk mempersiapkan server dan pengetahuan database dari suatu website yang di develop. Database ini dikembangkan menggunakan Framework Nodejs
3.	Mendesain dan membuat database untuk keperluan web application	Database yang optimal diupayakan dengan kinerja website yang optimal	60 jam	85	Student mampu membuat struktur database yang optimal sehingga proses running website dapat efisien
4.	Mengamankan dan mengatur user authentication dan access control untuk application backend	User authentication membuat akses satu user dengan yang lain	120 jam	85	Student dapat mengamankan dan mengatur setiap user dan memberikan akses yang sesuai dari setiap user agar data maupun fitur dari satu user dengan yang lain dapat dijaga dan diamankan
5.	Men-debug dan mengoptimalkan kinerja web application yang dibuat	Kinerja website bergantung atas kinerja akses database dan server dari website tersebut	100 jam	80	Student mampu melakukan debug atas issue back end yang ditemukan dan mampu membuat kinerja optimal dari sisi back end dalam suatu pengembangan website

PIC Program Kampus Merdeka
PT Hacktivate Teknologi Indonesia,



Lutfi Dwimulya

Jakarta 27 Desember 2023
Chief Evangelist
PT Hacktivate Teknologi Indonesia,



Riza Fahmi



HACKTIV8 KAMPUS MERDEKA PROGRAM - INTRO TO NODEJS

TRANSCRIPT OF RECORDS TOR No. INJS-2023-Aug-012

Name : MAULANA DAFFA ARDIANSYAH
ID : INJS-KS06-012
Email : maulanada25@gmail.com

Batch : 006
Class Start : 18 Agustus 2023
Attendance : 100%

A. Attendance Report

Day	Course	Learning Outcome	Attendance	Status
1	Introduction NodeJS (Hello World)	Dapat membuat hello word dengan menggunakan node js	☑	Completed
1	Modules	1. Menguasai penggunaan modul 2. Dapat membuat modul sendiri	☑	Completed
2	Event Handler NodeJS	1. memahami cara kerja event 2. menguasai handling event 3. dapat menangani error pada event	☑	Completed
2	Buffer & Streams	1. dapat berinteraksi dengan octet streams dalam TCP streams, dan operasi file sistem 2. dapat mengimplementasikan Uint8Array API dengan cara yang optimal 3. dapat mengimplementasikan antarmuka stream	☑	Completed
2	File Systems	1. mampu berinteraksi dalam sistem file dengan cara yang sama dengan fungsi standar POSIX 2. memahami perbedaan fungsi file sistem synchronous dan asynchronous	☑	Completed
3	HTTP Server	1. dapat membuat server 2. dapat mengkonfigurasi keamanan pada http header	☑	Completed
4	Pengenalan Expressjs	Dapat menginstall dan mengkonfigurasi express	☑	Completed
4	Routes	1. Memahami route methods, paths, parameters, handlers 2. Menguasai response methods 3. mengerti bagaimana titik akhir aplikasi (URIs) merespons permintaan klien.	☑	Completed
5	Static Files	1. Memahami cara menggunakan file static pada express	☑	Completed
5	MiddleWare	1. dapat melakukan akses req, res 2. dapat melakukan perubahan pada permintaan dan objek respons 3. dapat mengakhiri siklus req-res	☑	Completed
6	Templates & Template Engines	Dapat memanipulasi template file statis menjadi variabel-variabel yang kemudian dapat dikirimkan ke klien untuk memudahkan design halaman HTML.	☑	Completed
6	Querystring & Post Parameters	1. Dapat memparsing data dengan menggunakan querystring 2. Dapat mengirimkan informasi ke server melalui url	☑	Completed

7	RDBMS	1. Memahami konsep database 2. Memahami jenis-jenis database	✓	Completed
7	Pengenalan PostgreSQL	1. Dapat melakukan instalasi dan konfigurasi 2. Mengetahui fitur yang terdapat di PostgreSQL/MongoDB	✓	Completed
8	Persiapan Database	Mengetahui cara install postgre	✓	Completed
8	Query Melalui Terminal	Dapat membuat query database dengan menggunakan terminal / CLI (Create, Delete, View, Update)	✓	Completed
9	PgAdmin	Dapat melakukan interaksi postgres menggunakan GUI	✓	Completed
9	Tipe Data Integer	Dapat mengetahui perbedaan masing-masing tipe data dan penggunaannya	✓	Completed
9	Tipe Data Character		✓	Completed
10	Create & Delete Database	Mengetahui macam-macam DDL dan DML Mampu membuat dan menghapus database Menampilkan data menggunakan order dan limit	✓	Completed
10	Create & Delete Table	Mengetahui macam-macam DDL dan DML Mampu membuat dan menghapus database	✓	Completed
10	Insert & Update Table	Dapat mengisi dan memodifikasi isi tabel Menampilkan data menggunakan order dan limit	✓	Completed
11	Perkenalan REST API	Dapat menjelaskan kembali apa itu REST API dan cara kerjanya	✓	Completed
	Perkenalan GraphQL	Mengetahui fungsi penggunaan GraphQL	✓	Completed
12	Persiapan	Dapat mempersiapkan environment dan yang dibutuhkan untuk membuat API	✓	Completed
12	Endpoint Create	Dapat membuat endpoint untuk membuat data baru pada database	✓	Completed
12	Endpoint Read	Dapat membuat endpoint untuk membaca data pada database	✓	Completed
12	Endpoint Update	Dapat membuat endpoint untuk memperbarui data pada database	✓	Completed
12	Endpoint Delete	Dapat membuat endpoint untuk menghapus data pada database	✓	Completed
12	JWT	Dapat membuat token dan autentikasi pada REST API yang telah dibuat untuk meningkatkan keamanan	✓	Completed
13	Integrasi	Dapat melakukan integrasi REST API dengan Aplikasi menggunakan JWT, dan melempar datanya ke front-end	✓	Completed
13	Perkenalan Testing	Mengetahui pentingnya melakukan testing dalam sebuah aplikasi	✓	Completed
14	Tipe Testing	Mampu menentukan jenis testing apa yang diperlukan / cocok untuk aplikasi yang sedang dibangun	✓	Completed
14	Unit Testing Tools	1. Menguasai tools yang ada pada unit testing 2. Mampu menentukan tools yang digunakan sesuai dengan bahasa pemrograman pada sebuah aplikasi yang akan dilakukan pengujian 3. Dapat menggunakan Unit Testing	✓	Completed
14	Continuous Integration	Mengetahui konsep Continuous Integration	✓	Completed
15	Heroku Deployment	Memahami konsep heroku dan dapat deploy aplikasi yang telah dibuat pada Heroku	✓	Completed

16	Final Project - 1		✓	Completed
16	Final Project - 2		✓	Completed
	Final Project - 3		✓	Completed
	Final Project - 4		✓	Completed

No.	Task	Details	Score	Grade
1	Assignment 1		100	A
	Assignment 2		95	A
	Assignment 3		80	B
2	Final Project - 1		90	A
	Final Project - 2		95	A
	Final Project - 3		95	A
	Final Project - 4		90	A
Final Grade			92.08	A

Feedback Form Status : **Complete**
Attendance Status : **Complete**
Assignment Status : **Complete**
Final Project Status : **Complete**

Graduate Status : **Complete the course and eligible for certificate**

Jakarta, 29 Dec 2023



Hacktiv8 Indonesia