

SISTEM MANAJEMEN TUGAS “TASKIFY”



Disusun oleh:

Mohammad Januar Rizkyanto	1203210111
Moh. Adib Daffa	1203210119
Muhammad Hanafi Choirullah	1203210141
Moh. Rizky Prama Abdillah	1203210143

**PROGRAM STUDI INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM SURABAYA
2024**

DAFTAR ISI

DAFTAR ISI	2
DAFTAR GAMBAR	3
BAB 1 PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	5
1.3 Tujuan dan Manfaat	5
1.4 Batasan	5
BAB 2 LANDASAN TEORI	6
2.1 Pemrograman Fungsional	6
2.2 RESTful API	6
2.3 Flask	7
BAB 3 METODE	8
BAB 4 HASIL PENGEMBANGAN	10
4.1 Desain Perangkat Lunak	10
4.2 Fitur	11
4.2.1 Sign Up	11
4.2.2 Login	12
4.2.3 Home	13
4.2.4 Detail Task	14
4.2.5 Create Task	15
4.2.6 Delete Task	16
LAMPIRAN	18
1. Link Github	18
2. Link PPT	18
3. Jobdesk Anggota	18
DAFTAR PUSTAKA	19

DAFTAR GAMBAR

Gambar 3.1 Software Development Life Cycle	8
Gambar 4.1 Desain Aplikasi	10
Gambar 4.2 Sign Up Screen	11
Gambar 4.3 Backend SignUp	11
Gambar 4.4 API Register	12
Gambar 4.5 Login Screen	12
Gambar 4.6 Backend Login	13
Gambar 4.7 API Login	13
Gambar 4.8 Home Page	14
Gambar 4.9 Detail Page	15
Gambar 4.10 Create Page	15
Gambar 4.11 Delete Page	16
Gambar 4.12 Profile Screen Page	16

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dalam dunia dan lingkungan kerja yang modern, tata kelola waktu dan manajemen tugas menjadi kunci untuk produktivitas yang optimal. Banyak orang sering dihadapkan pada tugas-tugas yang kompleks dan beragam, serta jadwal yang padat dan cukup singkat. Banyaknya tugas yang harus dikerjakan kurangnya sistematisasi manajemen tugas sehingga tak sedikit mengakibatkan peningkatan stres dan menurunkan produktivitas seseorang [1].

Banyaknya tugas dan sistem deadline yang harus segera diselesaikan membuat para pekerja ataupun mahasiswa harus disiplin dan pandai dalam mengatur segala tugasnya untuk diselesaikan berdasarkan waktu, tingkat kesulitan ataupun prioritas. Dalam memudahkan hal tersebut, perlu adanya sistem manajemen tugas untuk memudahkan dalam mengatur tugas tugas yang harus dikerjakan.

Sistem manajemen tugas digital menjadi solusi yang optimal untuk para pekerja dan mahasiswa saat ini untuk membantu dalam mengatur dan menyelesaikan tugas dengan efisien. Oleh karena itu, pengembangan sistem manajemen tugas yang efektif dapat membantu mengatasi tantangan ini dengan menyediakan platform yang memungkinkan pengguna untuk mengatur tugas-tugas mereka dengan lebih baik.

Dengan adanya sistem manajemen tugas digital ini diharapkan dapat menjadikan karyawan agar dapat bekerja secara remote atau memiliki jadwal yang fleksibel. Hal ini tentunya akan penting untuk kedepannya bagi sebuah organisasi untuk memiliki sistem manajemen tugas yang memungkinkan dalam hal kolaborasi yang mudah dan pemantauan kemajuan proyek tim secara real-time. Selain itu sistem manajemen yang efektif juga dapat meningkatkan kolaborasi tim, mengurangi silo informasi, dan meningkatkan visibilitas terhadap proyek-proyek yang sedang berlangsung [2].

Munculnya teknologi web dan paradigma pengembangan perangkat lunak berbasis layanan, Representational State Transfer (RESTful API) menjadi salah satu solusi yang sangat baik untuk membangun sistem manajemen tugas yang efisien dan terintegrasi. RESTful API memungkinkan komunikasi antara berbagai sistem dan perangkat lunak dengan standar yang terdefinisi dengan baik, memfasilitasi pertukaran data yang lancar dan aman.

Dengan demikian, pengembangan sistem manajemen tugas digital yang efektif dan terintegrasi dengan teknologi web, seperti RESTful API, diharapkan para pekerja dapat mengelola tugas-tugas mereka dengan lebih efisien, meningkatkan kolaborasi tim, dan mencapai visibilitas yang lebih baik terhadap proyek-proyek yang sedang berlangsung.

1.2 **Rumusan Masalah**

Berdasarkan latar belakang diatas maka rumusan masalah yang dapat diangkat yaitu :

1. Bagaimana implementasi dari sistem manajemen tugas digital untuk membantu pekerja dalam mengatur tugas-tugas mereka dengan lebih efisien?
2. Apa saja fitur yang ada dalam pengembangan sistem manajemen tugas digital untuk membantu pengguna dalam manajemen tugas sehari-hari?

1.3 **Tujuan dan Manfaat**

Tujuan dari pembuatan sistem manajemen tugas digital yaitu:

1. Untuk merancang dan mengembangkan sistem manajemen tugas digital yang efektif dalam membantu individu mengatur tugas-tugas sehari-hari.
2. Untuk membuat fitur yang diperlukan untuk membantu mengatur dalam sistem manajemen tugas digital untuk meningkatkan produktivitas dan efisiensi pengguna.

1.4 **Batasan**

Batasan pada pembuatan sistem manajemen tugas digital ini yaitu:

1. Penelitian ini berfokus dalam pembuatan sistem manajemen tugas digital untuk penggunaan individu.
2. Evaluasi sistem manajemen tugas digital akan berfokus fungsionalitas sistem.

BAB 2 LANDASAN TEORI

2.1 Pemrograman Fungsional

Paradigma pemrograman fungsional merupakan sebuah paradigma pemrograman yang menganggap sebagai pemetaan dan perhitungan fungsi matematika. Pemrograman fungsional dalam hal ini membahas mengenai ekspresi dimana bahasa pemrograman fungsional merupakan pemrograman yang berorientasi ekspresi, karena dalam pemrograman fungsional semua dijadikan sebagai ekspresi [3].

Pemrograman Fungsional adalah paradigma pemrograman yang berfokus pada penggunaan fungsi sebagai unit dasar komputasi. Dalam pemrograman fungsional, fungsi dianggap sebagai nilai yang dapat diperlakukan secara fleksibel, seperti variabel lainnya. Pada pemrograman fungsional, program merupakan sebuah rantai transformasi program dimana akan berhubungan dari keadaan awal hingga kondisi terakhir yang akan melalui sebuah proses antara sistem melalui aplikasi fungsi [3].

Pada pemrograman fungsional paradigma didalamnya akan memperhatikan keadaan awal dan keadaan akhir tanpa perlu mengetahui bagaimana sistem mengeksekusi atau bagaimana informasi disimpan dalam memori. Dalam pemrograman fungsional harus diolah lebih dari pemrograman prosedural dalam pemrosesan bahasanya karena akan membebankan dalam kinerja dan efisiensi program. Untuk itu dalam paradigma pemrograman fungsional tidak lagi mempermasalahkan memorisasi dan struktur data, pemilahan antara data dan program dan juga tidak ada pengertian tentang variabel dikarenakan untuk memaksimalkan dalam kinerja dan efisiensi program [3].

2.2 RESTful API

Representational State Transfer (REST) merupakan teknologi yang diciptakan oleh Roy Fielding yang dibangun untuk membantu dalam pengembangan layanan berbasis web dan mobile dengan proses yang lebih ringan dan sederhana. Teknologi ini dibangun oleh Roy Fielding dengan prinsip utama desainnya dari REST API yaitu kinerja, probabilitas dan kesederhanaan [4].

RESTful API adalah sebuah metode untuk mendesain API (Application Programming Interface) dalam sebuah aplikasi web agar memenuhi prinsip-prinsip REST (Representational State Transfer). REST adalah suatu arsitektur yang digunakan dalam pengembangan aplikasi web yang mempromosikan pendekatan yang sederhana, mudah dipahami, dan bersifat stateless.

REST API dapat mendukung beragam sistem untuk berinteraksi dalam menerima, mengirim dan merespon data dengan mudah. Dalam penggunaannya, teknologi REST API didukung oleh URL dan HTTP yang data dalam database nya dipetakan dengan endpoint API pada REST API [5].

Restful merupakan teknologi yang cukup banyak digunakan dalam pengimplementasian sistem. Alasan penggunaan RESTful adalah karena RESTful API sangat baik dalam mengoptimalkan kinerja sistem. Ketika server RESTful diakses dalam skala besar dan secara bersamaan dalam sistem terdistribusi di internet, karena memiliki kinerja yang tinggi dan stabil [6].

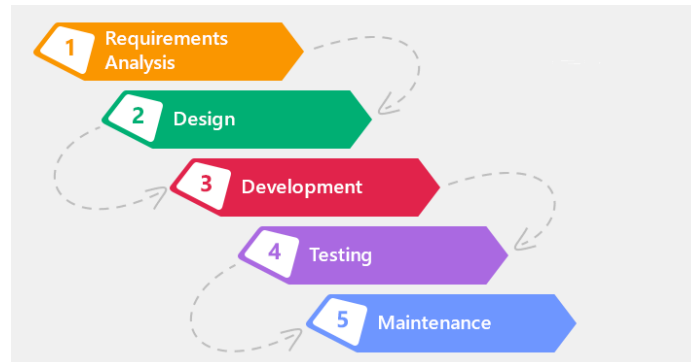
2.3 Flask

Flask adalah sebuah framework web mikro yang ditulis dalam bahasa pemrograman Python tergolong sebagai jenis micro framework [7]. Flask termasuk pada jenis microframework karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Flask dikembangkan oleh Armin Ronacher, Flask dirancang untuk membuat pengembangan aplikasi web dalam Python menjadi lebih sederhana dan mudah dipahami.

Flask dibuat untuk sebagai kerangka kerja aplikasi dan tampilan sebuah web yang nantinya pengembang dapat membuat tampilan dan sistem web menjadi lebih terstruktur dan dapat mengatur behavior suatu web dengan lebih mudah. Web framework flask ditulis menggunakan bahasa python, sehingga tentunya sebelum flask dapat digunakan seutuhnya, maka web developer setidaknya harus mempelajari bahasa pemrograman python terlebih dahulu untuk memahami dan memudahkan dalam penggunaannya [8].

Dalam penggunaan flask fleksibilitas serta skalabilitas dari Flask dapat dikatakan cukup tinggi dibandingkan dengan framework lainnya. Flask juga dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh Flask sendiri sehingga tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya [7].

BAB 3 METODE



Gambar 3.1 *Software Development Life Cycle*

1. Analisis Kebutuhan:

Analisis Kebutuhan meliputi perencanaan dan analisa sistem. Pada tahap ini, peneliti melakukan studi literatur dan observasi untuk mendapatkan kebutuhan dan spesifikasi yang dibutuhkan oleh sistem. Adapun kebutuhan yang diperlukan oleh aplikasi adalah yaitu nantinya aplikasi mampu menyimpan data yang diberikan oleh pengguna, membantu dalam mengatur dan memanajemen tugas tugas yang harus diselesaikan oleh penggunanya. Selain itu pada tahapan ini juga meliputi memahami kebutuhan dari API, termasuk data yang akan diekspos, operasi yang akan didukung didalam sistem, dan alur program yang diharapkan agar dapat memudahkan dalam pembangunan sistem.

2. Desain:

Pada tahapan ini yaitu mendesain tentang bagaimana sistem itu bekerja yaitu dari spesifikasi dan kebutuhan sistem yang telah didapatkan kemudian dilakukan implementasi dalam bentuk desain sistem. Desain yang dibuat antara lain adalah desain URL, dan desain interface. Kami menggunakan software figma untuk desain interface.

3. Development:

Pada tahapan ini yaitu tahap yang berfokus dalam proses pengembangan layanan. Selama tahap kerangka kerja ini akan dilakukan coding dengan mengikuti desain sistem yang telah dibuat. Setiap fungsi yang telah selesai kemudian diuji untuk mencari bug yang ada. Pada tahapan ini juga pengembang akan memilih kerangka kerja atau pustaka yang akan digunakan untuk membangun API. Pada penelitian dan pembuatan sistem ini kerangka kerja yang digunakan yaitu Flask untuk Python.

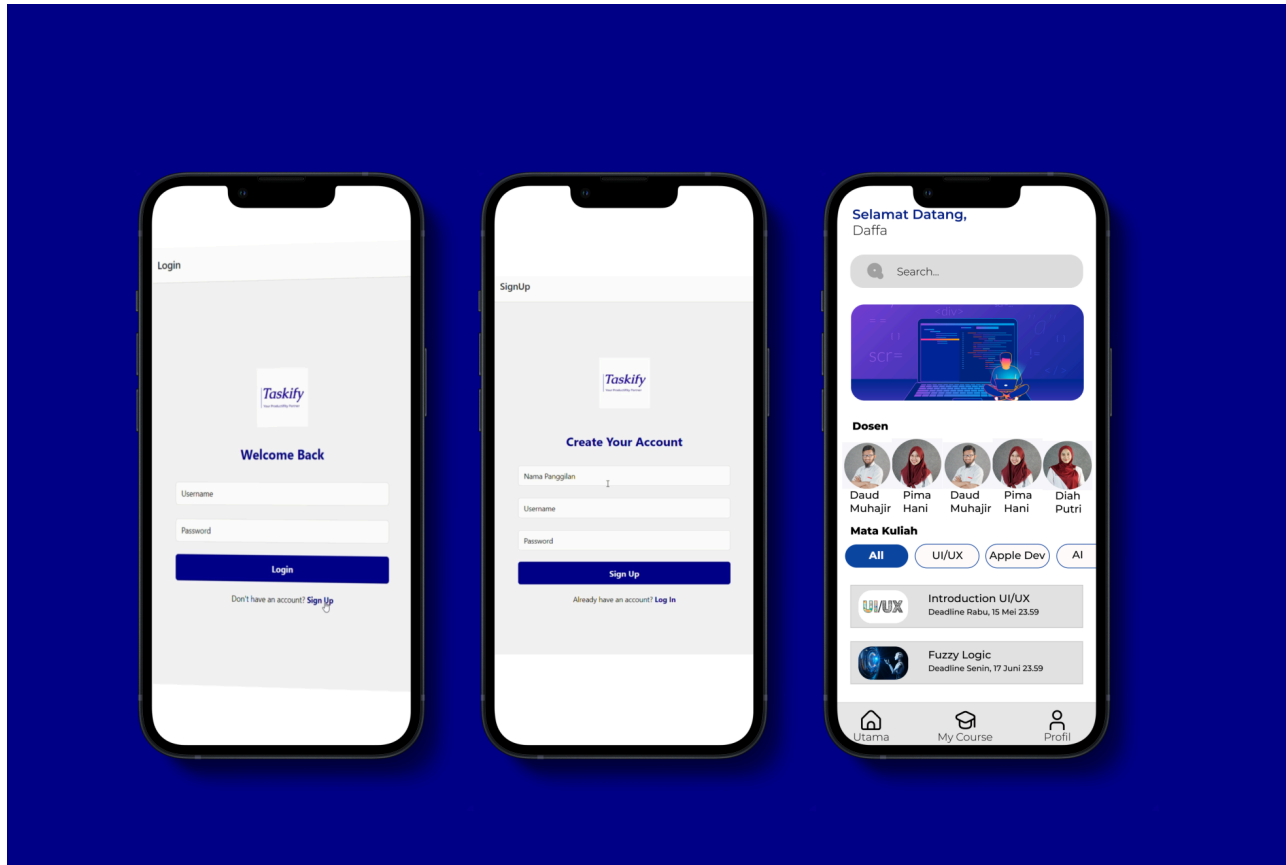
Selain itu di dalam tahap ini juga tahapan dalam mengimplementasikan dalam pembuatan titik-titik akhir API sesuai dengan desain. Ini melibatkan menulis kode untuk menangani permintaan masuk, memproses data, menghasilkan respons yang sesuai. Pastikan mematuhi prinsip REST.

4. Testing:

Apabila aplikasi sudah selesai dikembangkan, maka aplikasi siap untuk di test apakah aplikasi dapat berjalan dengan baik antara frontend maupun backend.

BAB 4 HASIL PENGEMBANGAN

4.1 Desain Perangkat Lunak

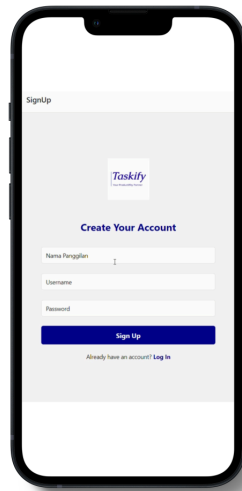


Gambar 4.1 Desain Aplikasi

Dalam pengerjaan aplikasi ini, kami menggunakan Figma untuk mendesain aplikasi bersama tim. Figma adalah *tools* yang digunakan untuk mendesain kapan saja dan dimanapun melalui internet. Pada umumnya, figma digunakan untuk mendesain interface aplikasi yang ingin dibuat.

4.2 Fitur

4.2.1 Sign Up



Gambar 4.2 Sign Up Screen

Pada halaman sign up, user bisa mendaftarkan akunnya, meliputi nama panggilan, username dan password.

```
@api_blueprint.route('/register', methods=['POST'])
def register():
    username = request.json.get('username')
    password = request.json.get('password')
    if not username or not password:
        return jsonify({'error': 'Username and password are required'}), 400

    # Check if username already exists
    if Account.query.filter_by(username=username).first():
        return jsonify({'error': 'Username already exists'}), 400

    # Create new account and hash the password
    new_account = Account(username=username)
    new_account.set_password(password)
    db.session.add(new_account)
    db.session.commit()

    return jsonify({'message': 'User registered successfully'}), 201
```

Gambar 4.3 Backend SignUp

Menggunakan function decorator untuk GET API, kemudian datanya masuk ke database MySQL.

POST register

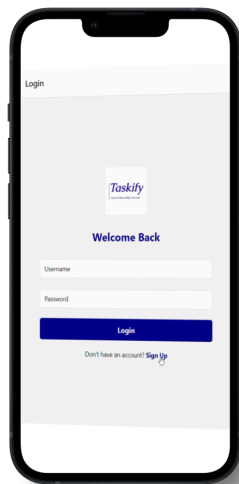
http://127.0.0.1:5000/api/register

Body raw (json)

```
json
{
  "username": "jeje",
  "password": "jeje11",
  "name": "jejechoi"
}
```

Gambar 4.4 API Register

4.2.2 Login



Gambar 4.5 Login Screen

Pada halaman login, user bisa memasukkan username dan password yang sudah didaftarkan sebelumnya.

```

@api_blueprint.route('/login', methods=['POST'])
def login():
    username = request.json.get('username')
    password = request.json.get('password')
    if not username or not password:
        return jsonify({'error': 'Missing username or password'}), 400
    account = Account.query.filter_by(username=username).first()
    if account and account.check_password(password):
        session['user_id'] = account.id
        return jsonify({'message': 'Login successful'}), 200
    return jsonify({'error': 'Invalid username or password'}), 401

```

Gambar 4.6 Backend Login

Jika username atau password salah, maka ada pop up jika username dan password salah dan tidak bisa login.

POST login

http://127.0.0.1:5000/api/login

Body raw (json)

json

```

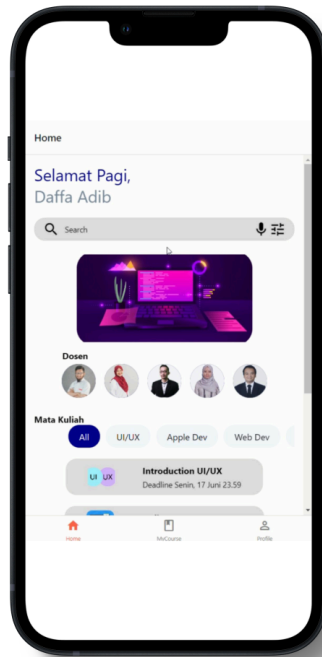
{
  "username": "jeje",
  "password": "jeje11"
}

```

Gambar 4.7 API Login

4.2.3 Home

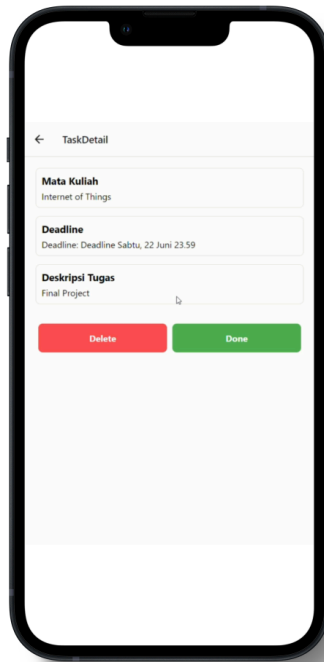
home screen atau dashboard adalah titik utama yang memberikan akses cepat dan ringkas ke informasi penting, alat, dan fungsi yang sering digunakan oleh pengguna. Ini bertujuan untuk meningkatkan efisiensi pengguna dalam berinteraksi dengan platform dan memberikan pengalaman yang disesuaikan dan intuitif.



Gambar 4.8 Home Page

4.2.4 Detail Task

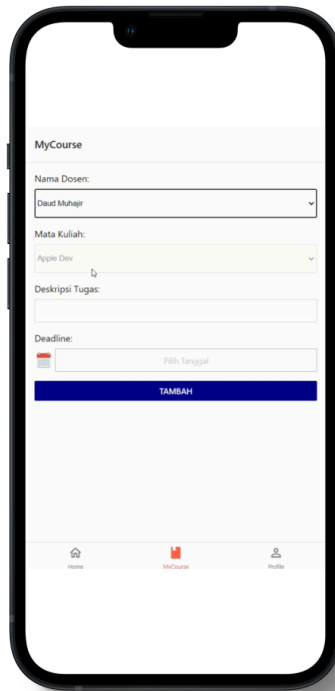
fitur detail untuk melihat detail tugas dan memudahkan user menghapus tugas yang sudah dikerjakan.



Gambar 4.9 Detail Page

4.2.5 Create Task

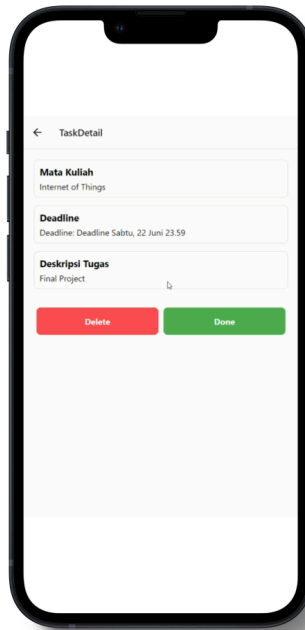
Fitur create pada sistem Task Manager memungkinkan pengguna untuk membuat tugas baru dengan mudah. Pengguna dapat mengisi formulir yang mencakup beberapa informasi penting seperti judul tugas, deskripsi (jika diperlukan), dan tanggal deadline atau batas waktu untuk menyelesaikan tugas tersebut



Gambar 4.10 Create Page

4.2.6 Delete Task

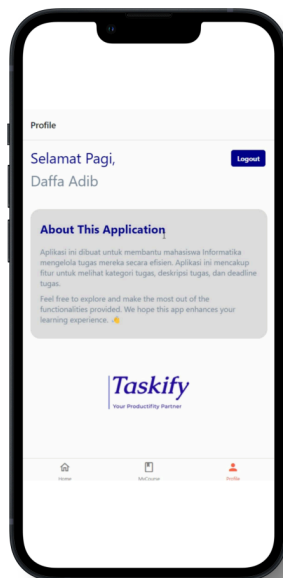
Fitur delete pada sistem Task Manager memungkinkan pengguna untuk menghapus tugas-tugas yang sudah selesai atau tidak relevan lagi. Pengguna dapat memilih tugas yang ingin dihapus dari daftar tugas mereka.



Gambar 4.11 Delete Page

4.2.7 Profile screen

Profil screen dalam konteks aplikasi atau sistem biasanya merujuk pada halaman atau layar yang menampilkan informasi pengguna secara detail.



Gambar 4.12 Profile Screen Page

LAMPIRAN

1. Link Github :

<https://github.com/adibdaffa20/finalproject-pemfung.git>

2. Link PPT :

https://www.canva.com/design/DAGDe36rIQI/9f5nvdTBRX_j7RXvneYJuQ/edit?utm_content=DAGDe36rIQI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

3. Jobdesk Anggota:

Mohammad Januar	: Membuat detail page dan crud
Moh. Adib	: Membuat home page dan profil
Muhammad Hanafi	: Membuat login dan sign up page
Moh. Rizky Prama	: Membuat Dokumentasi (PPT dan Laporan)

DAFTAR PUSTAKA

- [1] Robbins, S. P., & Coulter, M. (2019). Management (14th ed.). Pearson.
- [2] Sami-Ul-Haq, Khan, M. N. A., Mirza, A. M., Saleem, I., Saif Ur Rehman, & Raja Asif Wagan. (2021). Addressing Communication, Coordnation and Cultural Issues in Global Software Development Projects. *EMITTER International Journal of Engineering Technology*, 9(1), 13–30. <https://doi.org/10.24003/emitter.v9i1.558>
- [3] Liem, I., Keahlian, K., Rekayasa, D. &, & Lunak, P. (2008). *DIKTAT KULIAH DASAR PEMROGRAMAN Bagian : Pemrograman Fungsional*.
- [4] Soni, A., & Ranga, V. (2019). API features individualizing of web services: REST and SOAP. *International Journal of Innovative Technology and Exploring Engineering*, 8(9 Special Issue), 664–671. <https://doi.org/10.35940/ijitee.I1107.0789S19>
- [5] Rizki, K., & Adil, A. (2018). Implementasi Google Maps API Berbasis Android untuk Lokasi Fasilitas Umum di Kabupaten Sumbawa. *Jurnal MATRIK*, 17(2), 34–44. <https://doi.org/10.30812/matrik.v17i2.87>
- [6] Sommerville, Ian., “Software Engineering Rekayasa Perangkat Lunak”, Erlangga, Jakarta, 2018.
- [7] *Penggunaan Flask untuk Pemula*. (n.d.).
- [8] Izzati, M. D., & Kartikasari, M. D. (2022). Implementasi Metode Perhitungan Aktuaria Program Dana Pensiun Menggunakan Flask. *Jambura Journal of Mathematics*, 4(2), 247–264. <https://doi.org/10.34312/jjom.v4i2.12954>