



BANGKIT 2024
SURABAYA - ONLINE
(16 Februari 2024 - 30 Juni 2024)

Magang

HEALTHYLICIOUS
(Cloud Computing, Mobile Development, dan
Machine Learning)

AMMAL HAQI	NRP. 5024211003
SETYA HANDIKA	NRP. 5024211029
ARSENIUS AUDLEY WAHYU	NRP. 5024211030
DJATMIKO	
RAJAB AGUNG	NON TEKNIK KOMPUTER
FAISAL RIFKY NUGRAHA	NON TEKNIK KOMPUTER
MUHAMMAD ASRORUDDIN	NON TEKNIK KOMPUTER
ANAKTA RAFFAELL TAMBUNAN	NON TEKNIK KOMPUTER

DOSEN PEMBIMBING
Dr. Susi Juniastuti, S.T., M.Eng.
Ahmad Zaini, S.T., M.Sc.
Reza Fuad Rachmadi, S.T., M.T., Ph.D

DEPARTEMEN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2024

(Halaman ini sengaja dikosongkan)



ITS
Institut
Teknologi
Sepuluh Nopember

BANGKIT 2024
SURABAYA - ONLINE
(16 Februari 2024 - 30 Juni 2024)

Magang

HEALTHYLICIOUS
(Cloud Computing, Mobile Development, dan Machine Learning)

AMMAL HAQI	NRP. 5024211003
SETYA HANDIKA	NRP. 5024211029
ARSENIUS AUDLEY WAHYU DJATMIKO	NRP. 5024211030
RAJAB AGUNG	NON TEKNIK KOMPUTER
FAISAL RIFKY NUGRAHA	NON TEKNIK KOMPUTER
MUHAMMAD ASRORUDDIN	NON TEKNIK KOMPUTER
ANAKTA RAFFAELL TAMBUNAN	NON TEKNIK KOMPUTER

DOSEN PEMBIMBING
Dr. Susi Juniastuti, S.T., M.Eng.
Ahmad Zaini, S.T., M.Sc.
Reza Fuad Rachmadi, S.T., M.T., Ph.D

DEPARTEMEN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2024

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN I
HEALTHYLICIOUS

Laporan Magang ini disusun untuk memenuhi persyaratan akademik Departemen Teknik Komputer - Fakultas Teknologi Elektro dan Informatika Cerdas - Institut Teknologi Sepuluh Nopember

Menyetujui,
Pembimbing Magang



Ahmad Zaini, S.T., M.Sc.
NIP. 19750419200212 1 003

Menyetujui,
Pembimbing Magang



Reza Fuad Rachmadi, S.T., M.T., Ph.D
NIP. 19850403201212 1 001

Menyetujui,
Pembimbing Magang



Dr. Susi Juniastuti, S.T., M.Eng.
NIP. 19650618199903 2
001

Menyetujui,
Kepala Departemen Teknik Komputer ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
NIP.19700313 199512 1 001

SURABAYA

JUNI 2024

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN II
HEALTHYLICIOUS



BANGKIT 2024

Laporan Magang ini disusun untuk memenuhi persyaratan akademik Departemen Teknik Komputer - Fakultas Teknologi Elektro dan Informatika Cerdas - Institut Teknologi Sepuluh Nopember

Menyetujui,
Pembimbing Lapangan

Deti Anggraini Ekawati
NIP. 02022018017

BANDUNG
JUNI 2024

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa karena atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan laporan Magang di Google Bangkit 2024 yang dilaksanakan tanggal 16 Februari 2024 sampai dengan 30 Juni 2024 dengan judul: **HEALTHYLICIOUS**. Dalam penyelesaian Laporan Magang ini, kami mengucapkan terima kasih kepada berbagai pihak yang telah membantu terselesaikannya laporan ini hingga akhir:

- Allah SWT karena telah memberikan kelancaran serta keberkahan selama dilakukannya Magang.
- Bapak Dr. Supeno Mardi Susiki Nugroho, S.T., M.T. selaku Kepala Departemen Teknik Komputer FTEIC-ITS.
- Ibu Dr. Diah Puspito Wulandari S.T., M.Sc. selaku Koordinator Magang Departemen Departemen Teknik Komputer FTEIC-ITS
- Ibu Dr, Susi Juniastuti, S.T., M.Eng., Bapak Ahmad Zaini, S.T., M.Sc., dan Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D selaku Pembimbing Magang
- Mas Muhammad Rizki Syafapri, Mbak Elma Angelina Sondakh, dan Mbak Dwi Melly Aprilia Sari selaku Mentor Bangkit 2024 selama pelaksanaan Magang ini.
- Team Bangkit yang senantiasa membantu kelancaran berjalannya kegiatan Bangkit Academy 2024.
- Rekan-rekan satu tim serta semua pihak terkait yang tidak dapat kami sebutkan satu persatu.

Penulis menyampaikan permohonan maaf jika selama pelaksanaan Magang ini terdapat hal yang kurang berkenan dan apabila ada banyak kekurangan dalam perancangan dan pembuatan laporan Magang ini. Semoga buku laporan Magang ini dapat memberikan manfaat bagi para pembaca, khususnya bagi penulis sendiri.

Surabaya, Juni 2024

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN I	
LEMBAR PENGESAHAN II	
KATA PENGANTAR	
DAFTAR ISI	
DAFTAR GAMBAR	
DAFTAR DIAGRAM	
DAFTAR TABEL	
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERMASALAHAN	3
1.3 TUJUAN	3
1.4 BATASAN PERMASALAHAN	4
1.5 BENTUK KEGIATAN	5
1.6 TEMPAT DAN WAKTU PELAKSANAAN	7
1.7 METODE PENULISAN	8
1.8 SISTEMATIKA PENULISAN	8
BAB II TINJAUAN UMUM PERUSAHAAN	10
2.1 PROFIL SINGKAT	10
2.2 SEJARAH SINGKAT	11
2.3 VISI & MISI	12
BAB III TINJAUAN PUSTAKA	14
3.1 Machine Learning	14
3.2 Recommendation System	14
3.3 K-Nearest Neighbors	16
3.4 Python	17
3.5 NumPy	18
3.6 Pandas	18
3.7 Scikit-learn	19
3.8 Cloud Computing	20
3.8.1 On-demand self-service	21
3.8.2 Broad network access	21
3.8.3 Resource Pooling	21
3.8.4 Rapid elasticity	21
3.8.5 Measured service	21
3.8.6 Infrastructure as a Service	22

3.8.7	Platform as a service	22
3.8.8	Software as a service	22
3.9	Google Cloud Platform	23
3.9.1	Cloud Run	23
3.10	RESTful API	24
3.10.1	Client-server architecture	25
3.10.2	Stateless	25
3.10.3	Cacheability	26
3.10.4	Layered System	26
3.10.5	Uniform Interface	27
3.10.6	Code on Demand	27
3.11	Hapi.js	27
3.11.1	Routing yang Fleksibel	28
3.11.2	Pengelolaan Respons	28
3.11.3	Ekosistem Plugin	28
3.11.4	Keamanan	28
3.12	Firebase	28
3.13	Android Studio	29
3.14	Kotlin	30
3.15	Model View ViewModel	30
BAB IV	PEMBAHASAN	32
4.1	Rencana Pengembangan Produk	32
4.2	Tahap Pengembangan	34
4.2.1	Tim Mobile Development	35
4.2.1.1	Fething Data dari API	35
4.2.1.2	Pengolahan Data	37
4.2.1.3	Menampilkan Data pada Tampilan Aplikasi	40
4.2.1.4	User Interface Aplikasi	42
4.2.1.5	Analisis Survey Form	50
4.2.2	Tim Machine Learning	66
4.2.2.1	Data Gathering	66
4.2.2.2	Preprocessing and Cleaning Data	72
4.2.2.3	Modelling and Evaluation	76
4.2.2.4	Deployment	85
4.2.2.5	Analisis Performa Model	86
4.2.3	Tim Cloud Computing	87
4.2.3.1	API Authentication	87
4.2.3.2	API User Profile	88
4.2.3.3	API Features	89

4.2.3.4 Dokumentasi API	91
4.2.3.5 Kontainerisasi Dockerfile	94
4.2.3.6 Deploy dengan Cloud Run	97
4.2.3.7 Penyimpanan Data di Firebase	99
4.2.3.8 Analisis Request Latency	101
BAB V PENUTUP	106
5.1 KESIMPULAN	106
5.1.1 KESIMPULAN TEKNIS	106
5.1.2 KESIMPULAN UMUM	107
5.2 SARAN	107
5.2.1 SARAN TEKNIS	107
5.2.2 SARAN UMUM	109
DAFTAR PUSTAKA	111
LAMPIRAN	114

DAFTAR GAMBAR

Gambar 2.1.1 Logo Bangkit 2024	7
Gambar 3.7.1.1 GCP services berdasarkan tingkat manajemen penggunaanya	19
Gambar 4.1.1 Timeline Pengerjaan Magang	26
Gambar 4.2.1.1 Interface User Detail	40
Gambar 4.2.1.2 UI/UX Aplikasi Pantry	43
Gambar 4.2.1.3 UI/UX Aplikasi Home	44
Gambar 4.2.1.4 UI/UX Aplikasi Favourite	45
Gambar 4.2.1.5 UI/UX Aplikasi My Basket	46
Gambar 4.2.1.4 UI/UX Aplikasi Detailed Recipe	47
Gambar 4.2.1.5 UI/UX Aplikasi Browse Recipe	48
Gambar 4.2.1.6 UI/UX Aplikasi User Profile	49
Gambar 4.2.2.1 Pustaka recipe-scrapers	68
Gambar 4.2.2.2 Flowchart Cleaning Data	74
Gambar 4.2.2.5 Diagram alir TF-IDF vectorization dan cosine similarity untuk fitur “Get Recipes”	77
Gambar 4.2.2.6 Diagram alir KNN untuk fitur Recommended for You	78
Gambar 4.2.2.6 Hasil Generate Recipes	84
Gambar 4.2.2.7 Penerapan model dalam Aplikasi	85
Gambar 4.2.3.1 Layanan Artifact Registry	96
Gambar 4.2.3.2 Layanan Cloud Run	98
Gambar 4.2.3.3 Layanan Firebase	100
Gambar 4.2.3.4 Data Request Latency Cloud Run	102
Gambar 4.2.3.5 Data Cloud Trace	104

DAFTAR DIAGRAM

Diagram 4.2.1.1 Diagram Hasil Responsi 1	50
Diagram 4.2.1.2 Diagram Hasil Responsi 2	51
Diagram 4.2.1.3 Diagram Hasil Responsi 3	52
Diagram 4.2.1.4 Diagram Hasil Responsi 4	53
Diagram 4.2.1.5 Diagram Hasil Responsi 5	54
Diagram 4.2.1.6 Diagram Hasil Responsi 6	55
Diagram 4.2.1.7 Diagram Hasil Responsi 7	56
Diagram 4.2.1.8 Diagram Hasil Responsi 8	57
Diagram 4.2.1.9 Diagram Hasil Responsi 9	58
Diagram 4.2.1.10 Diagram Hasil Responsi 10	58
Diagram 4.2.1.11 Diagram Hasil Responsi 11	59
Diagram 4.2.1.12 Diagram Hasil Responsi 12	60
Diagram 4.2.1.13 Diagram Hasil Responsi 13	61
Diagram 4.2.1.14 Diagram Hasil Responsi 14	62
Diagram 4.2.1.15 Diagram Hasil Responsi 15	62
Diagram 4.2.1.16 Diagram Hasil Responsi 16	64
Diagram 4.2.1.17 Diagram Hasil Responsi 17	65

DAFTAR TABEL

Tabel 4.2.3.1 API Authentication	88
Tabel 4.2.3.2 API User Profile	89
Tabel 4.2.3.3 API Features	90

BAB I PENDAHULUAN

1.1 LATAR BELAKANG

Sejalan dengan tujuan utama dari Sustainable Development Goals (SDGs) poin nomor 3, yaitu Good Health and Well-Being, proyek kami berjudul “Health Innovation: Empowering Vulnerable Communities for Health and Well-being” dirancang untuk meningkatkan kesehatan dan kesejahteraan masyarakat, khususnya komunitas rentan di Indonesia. Indonesia, dengan beragam budaya dan tantangan geografisnya, sering kali melihat komunitas rentan menghadapi hambatan dalam mengakses layanan kesehatan yang penting. Oleh karena itu, proyek ini bertujuan untuk memanfaatkan teknologi, termasuk Machine Learning, AI, Cloud Computing, dan Pengembangan Aplikasi Seluler, guna menjembatani kesenjangan tersebut dan meningkatkan akses serta kualitas layanan kesehatan bagi mereka yang membutuhkan.

Kami mengamati bahwa saat ini banyak orang yang semakin peduli dan tertarik untuk menjalani gaya hidup sehat, khususnya dalam memilih dan mengkonsumsi makanan yang bergizi dan seimbang. Namun, tantangan umum yang dihadapi adalah keterbatasan pengetahuan tentang resep sehat yang dapat dibuat dari bahan-bahan yang ada di rumah. Kurangnya pengetahuan ini disebabkan oleh tidak adanya sarana yang memberikan rekomendasi resep makanan berdasarkan bahan yang tersedia serta mencantumkan nilai gizi dari setiap resep makanan sehat tersebut.

Sejalan dengan tren hidup sehat yang meningkat setelah pandemi COVID-19, banyak individu mulai kembali menjalankan kebiasaan hidup sehat untuk menjaga kesehatan mereka. Salah satu

cara untuk menjaga hidup sehat adalah dengan memilih makanan sehat yang dikonsumsi. Sehingga, tema kesehatan ini sangat relevan dan diharapkan dapat memberikan dampak positif bagi masyarakat.

Proyek Healthylicious hadir untuk menjawab kebutuhan yang semakin meningkat akan alat yang mudah diakses dan ramah pengguna guna memfasilitasi kebiasaan makan yang lebih sehat dalam gaya hidup yang serba cepat saat ini. Dengan meningkatnya perhatian terhadap nutrisi dan obesitas, ada permintaan yang besar akan solusi yang dapat memberdayakan individu untuk membuat pilihan makanan yang lebih baik dan terinformasi. Healthylicious bertujuan untuk mengisi kekosongan ini dengan menyediakan aplikasi seluler yang menawarkan rekomendasi personal untuk makanan bergizi berdasarkan preferensi diet pengguna dan bahan-bahan yang tersedia.

Healthylicious berupaya untuk memanfaatkan teknologi, termasuk Machine Learning, Cloud Computing, dan Pengembangan Aplikasi Seluler, untuk menganalisis pola diet dan merekomendasikan makanan yang seimbang. Melalui penelitian kami, kami menjawab pertanyaan seperti: Bagaimana integrasi machine learning, cloud computing, dan pengembangan mobile dapat memfasilitasi analisis efektif terhadap pola makan dan rekomendasi makanan seimbang? Dengan menangani masalah ini, tim kami berupaya memberdayakan individu untuk mengambil kendali atas kesehatan dan kesejahteraan mereka melalui pilihan nutrisi yang lebih baik, yang pada akhirnya berkontribusi pada masyarakat yang lebih sehat.

Dengan latar belakang tersebut, kami berinisiatif untuk mengembangkan sebuah platform aplikasi yang mendukung gaya hidup sehat dengan fokus pada makanan sehat, yang kami namakan

Healthylicious. Aplikasi ini memiliki fitur rekomendasi resep makanan berdasarkan bahan yang tersedia dengan mencantumkan informasi kalori pada setiap resep. Healthylicious juga dirancang untuk tidak menampilkan resep yang mengandung bahan yang tidak disukai atau yang dapat menyebabkan alergi pada pengguna. Kami berharap aplikasi Healthylicious dapat menjadi solusi praktis bagi masyarakat dalam menjalankan pola hidup sehat dengan memanfaatkan bahan-bahan yang sudah tersedia di rumah, sehingga dapat meningkatkan kualitas hidup dan kesehatan secara menyeluruh.

1.2 PERMASALAHAN

Dari latar belakang yang telah diuraikan, dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana mendapat rekomendasi resep makanan sehat berdasarkan bahan yang tersedia?
2. Bagaimana mendapatkan akses terhadap informasi nutrisi dari resep makanan yang akan dibuat?
3. Bagaimana mengurangi kesenjangan dalam akses terhadap informasi kesehatan khususnya di komunitas rentan?

1.3 TUJUAN

Adapun tujuan dari proyek Healthylicious dapat dilihat dari beberapa sudut pandang berikut:

- a. Tujuan Pengembangan Proyek
 1. Mengembangkan aplikasi yang dapat memberikan rekomendasi resep makanan sehat berdasarkan bahan yang tersedia di rumah pengguna.
 2. Meningkatkan akses masyarakat terhadap informasi nutrisi dan resep sehat melalui teknologi yang mudah digunakan.
 3. Memberikan solusi yang dapat membantu mengurangi kesenjangan dalam akses terhadap

informasi kesehatan, khususnya di komunitas rentan.

b. Tujuan Pembelajaran dan Pengembangan Kompetensi

1. Meningkatkan kemampuan berpikir kreatif, adaptif, konstruktif, dan kritis dalam menyelesaikan permasalahan, baik secara personal maupun dalam tim.
2. Menciptakan sinergi yang harmonis antara dunia pendidikan dan dunia kerja, sehingga dapat saling mendukung dan memperkaya satu sama lain.
3. Memperluas wawasan mahasiswa dalam pemahaman serta pengaplikasian Ilmu Pengetahuan dan Teknologi (IPTEK) di dunia kerja nyata.

c. Tujuan Akademis

1. Memenuhi persyaratan akademis berupa Satuan Kredit Semester (SKS) yang wajib diselesaikan oleh mahasiswa Departemen Teknik Komputer ITS.
2. Meningkatkan pengetahuan dan keterampilan mahasiswa melalui penerapan teknik dan ilmu yang telah dipelajari selama perkuliahan dalam konteks kerja nyata.
3. Mengembangkan profesionalisme dan kompetensi mahasiswa melalui pengalaman praktis yang diperoleh selama magang.

1.4 BATASAN PERMASALAHAN

Proyek Healthylicious menghadapi beberapa batasan yang perlu diperhatikan. Salah satunya adalah keterbatasan dalam jumlah dan kualitas dataset resep makanan yang tersedia, yang penting untuk menghasilkan rekomendasi yang akurat dan relevan. Kami juga menghadapi tantangan dalam mengintegrasikan berbagai teknologi seperti Machine Learning, Cloud Computing, dan Mobile Development secara efektif. Selain itu, personalisasi rekomendasi

resep berdasarkan preferensi diet pengguna dan bahan yang tersedia memerlukan algoritma yang kompleks dan data pengguna yang cukup.

1.5 BENTUK KEGIATAN

Program Bangkit 2024 merupakan bagian dari inisiatif Merdeka Belajar Kampus Merdeka (MBKM) yang bekerja sama dengan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia (Kemendikbud Ristek RI). Program ini bertujuan untuk membekali mahasiswa dengan keterampilan teknologi yang relevan dengan kebutuhan industri melalui pembelajaran interdisipliner yang mencakup Cloud Computing, Machine Learning, dan Mobile Development. Setiap peserta akan diberikan kurikulum sesuai dengan jalur pembelajaran interdisipliner yang mereka pilih, dimana mereka akan mengikuti kelas dalam format asinkron yang disediakan oleh Platform Dicoding Academy dan Coursera, dengan para pengajar profesional di bidangnya dengan harapan di akhir program ini, peserta akan dibekali dengan keahlian teknologi, soft skill, serta kemahiran bahasa Inggris yang diperlukan untuk bertransisi dari dunia akademis ke dunia kerja.

Capstone Project merupakan komponen inti dari Program Bangkit 2024, dimana peserta ditugaskan untuk menyelesaikan proyek-proyek yang relevan dengan topik yang telah diusulkan berkaitan dengan real-world case. Proyek ini dibagi menjadi tiga kategori utama: product-based project, company-based project, dan entrepreneur-based project.

Product-based Project:

- Peserta membentuk tim dan membuat Project Plan yang mencakup judul, pemilihan tema, abstrak, asal ide, cakupan proyek, jadwal proyek, daftar alat, dan penjelasan peran masing-masing Learning Path dalam proyek tersebut.
- Project Plan dan daftar anggota tim disubmit

melalui formulir yang disediakan oleh Bangkit 2024, kemudian menunggu konfirmasi penerimaan.

Company-based Project:

- Perusahaan mitra Bangkit 2024 mengusulkan beberapa judul atau topik permasalahan yang membutuhkan solusi. Anggota tim ditentukan berdasarkan kebutuhan Learning Path perusahaan tersebut.

Entrepreneur-based Project:

- Proyek ini berfokus pada pengembangan ide bisnis yang mencakup deskripsi bisnis, analisis pasar, strategi pemasaran, rencana keuangan, dan analisis risiko. Program ini juga menyediakan pendanaan inkubasi untuk membantu merealisasikan ide-ide inovatif menjadi startup yang berkelanjutan.

Berikut merupakan uraian dari teknis Kegiatan Magang yang dilakukan peserta Program Bangkit Academy 2024:

1. Setiap peserta dalam tim Capstone Project akan mengaplikasikan ilmu-ilmu yang telah dipelajari selama program untuk membuat aplikasi yang menyelesaikan masalah pada tema yang dipilih. Berikut adalah spesialisasi bidang yang akan diimplementasikan:
 - Cloud Computing: Mengelola infrastruktur server, database, dan jaringan yang diperlukan untuk aplikasi.
 - Machine Learning: Mengembangkan model pembelajaran mesin untuk memproses data dan membuat prediksi atau keputusan berdasarkan data tersebut.
 - Mobile Development: Merancang dan mengembangkan aplikasi mobile yang inovatif dan responsif.
2. Tim peserta akan dibimbing oleh seorang mentor

advisor yang memiliki keterampilan dan keahlian yang diperlukan untuk membantu pengembangan aplikasi tersebut.

3. Repositori GitHub dibuat untuk menyimpan perubahan dan sebagai bukti dalam laporan kemajuan proyek.
4. Pada minggu kedua, akan diadakan mid-checkpoint di mana setiap tim wajib membuat laporan yang berisi perkembangan proyek dari masing-masing bidang, serta bukti-bukti pendukung, dan mengirimkannya melalui Google Form yang disediakan oleh Bangkit 2024.
5. Di minggu terakhir, setiap tim akan merekam presentasi yang mencakup latar belakang proyek, tampilan aplikasi, target pemasaran, perbandingan dengan aplikasi serupa, timeline pengembangan, dan perencanaan anggaran (budgeting).

Dengan pendekatan ini, Program Bangkit 2024 diharapkan dapat memberikan pengalaman praktis yang komprehensif bagi para peserta dalam mengaplikasikan pengetahuan dan keterampilan yang telah diperoleh selama program berlangsung

1.6 TEMPAT DAN WAKTU PELAKSANAAN

Adapun waktu dan tempat pelaksanaan dari Magang ini adalah sebagai berikut:

Waktu : 16 Februari – 30 Juni 2024

Tempat : Surabaya - Online

1.7 METODE PENULISAN

Metode penulisan yang digunakan dalam pelaksanaan magang ini meliputi beberapa pendekatan yang sistematis dan terstruktur untuk memastikan akurasi dan relevansi data yang diperoleh. Adapun metode-metode yang digunakan adalah sebagai berikut:

1. Pengumpulan data dilakukan melalui pengamatan langsung dan partisipasi aktif dalam kegiatan Capstone Project Bangkit 2024. Pengamatan ini mencakup berbagai aspek kegiatan, mulai dari proses perencanaan proyek, pelaksanaan tugas-tugas teknis, hingga evaluasi hasil akhir.
2. Untuk memperkaya dan mendukung data yang diperoleh dari pengamatan, metode literatur digunakan sebagai sumber referensi tambahan. Sumber literatur ini mencakup buku, jurnal, artikel ilmiah, website, modul pembelajaran, dan referensi lain yang relevan dengan topik yang dibahas dalam proyek.
3. Diskusi dengan pembimbing lapangan, yaitu Mas Rizki, Mbak Elma, dan Mbak Melly selaku mentor dari Bangkit 2024, menjadi bagian penting dalam metode penulisan ini. Melalui diskusi, diperoleh berbagai wawasan, saran, dan masukan yang berharga terkait dengan pelaksanaan proyek.

1.8 SISTEMATIKA PENULISAN

Laporan magang ini disusun secara sistematis dan terstruktur, dibagi menjadi lima bab utama yang saling berkaitan untuk memberikan pemahaman yang komprehensif mengenai pelaksanaan dan hasil magang. Berikut adalah sistematika penulisan laporan ini:

a. BAB I : PENDAHULUAN

Bab ini memberikan gambaran umum mengenai laporan magang, dimulai dengan latar belakang yang menjelaskan alasan dan konteks di balik pelaksanaan proyek Healthylicious. Selanjutnya, bab ini mengidentifikasi batasan permasalahan yang dihadapi selama pelaksanaan proyek, serta tujuan dari pelaksanaan magang ini. Bentuk kegiatan yang dilakukan selama magang, waktu dan tempat pelaksanaan, metode penulisan yang digunakan untuk mengumpulkan dan

menganalisis data, serta sistematika penulisan laporan ini juga dijelaskan secara rinci dalam bab ini.

b. BAB II : TINJAUAN UMUM PERUSAHAAN

Bab ini membahas profil dan gambaran umum mengenai kegiatan Capstone Project Bangkit 2024. Bab ini dimulai dengan sejarah dan latar belakang program Bangkit, yang menjelaskan asal mula dan tujuan dari program ini. Struktur program yang mencakup berbagai komponen utama dan kegiatan yang dilaksanakan dalam program Bangkit 2024 juga dijelaskan secara rinci.

c. BAB III : TINJAUAN PUSTAKA

Bab ini mencakup teori-teori penunjang yang relevan dengan perancangan aplikasi Healthylicious. Penjelasan dimulai dengan konsep-konsep dasar mengenai teknologi yang digunakan dalam proyek, serta literatur yang mendukung pengembangan aplikasi Healthylicious terkait dengan Machine Learning, Cloud Computing, dan Mobile Development.

d. BAB IV : PEMBAHASAN

Bab ini menguraikan proses dan hasil rancangan aplikasi Healthylicious. Deskripsi dimulai dengan tahapan pengembangan aplikasi, mulai dari perencanaan hingga implementasi. Bab ini memberikan gambaran lengkap mengenai seluruh proses yang dilalui dalam pengembangan aplikasi Healthylicious.

e. BAB V : PENUTUP

Bab ini menyajikan kesimpulan dan saran dari hasil pelaksanaan magang. Kesimpulan mencakup ringkasan mengenai hasil dan temuan utama dari proyek Healthylicious, serta rekomendasi untuk pengembangan lebih lanjut dan perbaikan di masa mendatang.

f. DAFTAR PUSTAKA

Bagian ini berisi daftar sumber referensi yang digunakan dalam penulisan laporan magang ini. Sumber-sumber ini mencakup buku, jurnal, artikel, dan sumber lainnya yang relevan dengan topik yang dibahas, memberikan dasar ilmiah yang kuat untuk analisis dan temuan yang disajikan dalam laporan.

g. LAMPIRAN

Bagian lampiran menyertakan dokumen tambahan yang melengkapi laporan magang ini, seperti dokumentasi proyek dalam bentuk gambar, diagram, dan dokumen pendukung lainnya.

(Halaman ini sengaja dikosongkan)

BAB II TINJAUAN UMUM PERUSAHAAN

2.1 PROFIL SINGKAT



Gambar 2.1.1 Logo Bangkit 2024

Bangkit adalah program kesiapan karir yang dirancang oleh Google dengan dukungan penuh dari GoTo, Traveloka, dan Deeptech Foundation. Program ini juga merupakan bagian dari inisiatif Kampus Merdeka - Studi Independen Bersertifikat. Selain itu, Bangkit bekerja sama dengan Universitas Stanford melalui program University Innovation Fellow. Program Bangkit 2024 dirancang untuk mengembangkan kompetensi mahasiswa agar siap berkarir di dunia teknologi. Program ini merupakan hasil kemitraan antara Dirjen Pendidikan Tinggi Kemendikbud Ristek, Google, GoTo, Traveloka, dan mitra perguruan tinggi lainnya.

Bangkit 2024 menawarkan kurikulum interdisipliner yang dirancang untuk membekali siswa dengan keterampilan yang dibutuhkan di industri teknologi. Program ini mencakup tiga jalur pembelajaran utama: Machine Learning, Mobile Development, dan Cloud Computing. Setiap jalur pembelajaran dilengkapi dengan materi yang mendalam, pelatihan praktis, dan proyek-proyek nyata yang relevan dengan kebutuhan industri saat ini.

Di akhir program, peserta akan dibekali dengan keahlian teknologi yang komprehensif, termasuk keterampilan soft skill seperti kemampuan berpikir kritis, komunikasi, dan kolaborasi, serta kecakapan bahasa Inggris. Hal ini bertujuan untuk mempersiapkan peserta agar dapat bertransisi dari dunia akademis ke dunia kerja

dengan lancar dan sukses di perusahaan-perusahaan terkemuka.

Bangkit 2024 juga menawarkan berbagai kegiatan tambahan yang mendukung pengembangan profesional peserta. Peserta akan memiliki kesempatan untuk berpartisipasi dalam proyek Capstone yang menantang, di mana mereka dapat mengaplikasikan pengetahuan yang telah diperoleh dalam proyek nyata yang relevan dengan industri. Selain itu, peserta akan mendapatkan bimbingan dari mentor yang berpengalaman, yang akan membantu mereka dalam mengembangkan proyek dan mempersiapkan diri untuk karir di bidang teknologi.

Dengan berbagai fitur dan dukungan yang ditawarkan, Bangkit 2024 bertujuan untuk menjadi program yang holistik dan komprehensif dalam mempersiapkan talenta muda Indonesia untuk bersaing di pasar kerja global. Program ini tidak hanya fokus pada pengembangan keterampilan teknis tetapi juga pada pengembangan karakter dan profesionalisme peserta, sehingga mereka siap untuk menghadapi tantangan di dunia kerja dan berkontribusi secara signifikan dalam industri teknologi.

2.2 SEJARAH SINGKAT

Indonesia telah menjadi kekuatan digital terbesar di Asia Tenggara, dengan nilai pasar bruto (GMV) diprediksi akan berlipat ganda pada tahun 2030 dibandingkan dengan nilai saat ini. Namun, meskipun pertumbuhan pesat di sektor teknologi, terdapat tantangan besar dalam memenuhi kebutuhan akan talenta digital yang berkualitas. Keterampilan tambahan yang spesifik sangat diperlukan untuk memasuki industri teknologi yang terus berkembang.

Bangkit pertama kali diluncurkan pada tahun 2020 sebagai respons terhadap kebutuhan akan pengembangan talenta digital. Program ini didorong oleh Google dan dilaksanakan bekerja sama dengan Gojek, Tokopedia, dan Traveloka sebagai mitra pendiri. Sejak

itu, Bangkit telah melatih lebih dari 6.000 orang Indonesia di bidang machine learning, mobile development, dan cloud computing, dengan banyak lulusan yang berhasil berkarir di startup, perusahaan nasional, hingga unicorn.

Pada tahun 2021, Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi merangkul Bangkit sebagai bagian dari Kampus Merdeka. Dukungan ini memungkinkan program untuk memperluas jangkauannya hingga sepuluh kali lipat dari jumlah peserta sebelumnya, memberikan dampak yang signifikan bagi semua lulusan.

Pada tahun 2024, Bangkit terus berkembang dengan memperkenalkan kurikulum yang lebih disempurnakan dan berfokus pada tiga jalur pembelajaran utama: Machine Learning, Mobile Development, dan Cloud Computing. Program ini dirancang untuk mempersiapkan siswa dengan keterampilan teknologi yang relevan, dilengkapi dengan pelatihan soft skills dan kecakapan bahasa Inggris. Di akhir program, peserta akan memiliki keahlian yang diperlukan untuk sukses di dunia kerja dan berkontribusi secara signifikan di perusahaan-perusahaan teknologi terkemuka.

2.3 VISI & MISI

Bangkit memiliki visi untuk menjadi program pengembangan talenta digital terdepan di Indonesia, yang mempersiapkan lulusan untuk berkontribusi signifikan di sektor teknologi global. Program ini bertujuan untuk menyampaikan pelatihan yang terstruktur dan berkualitas tinggi guna menghasilkan lulusan berkaliber tinggi yang siap berkarir di perusahaan teknologi dan startup kelas dunia.

Untuk mewujudkan visi tersebut, Bangkit menjalankan misinya melalui tiga prinsip utama: *industry-led*, *immersive*, dan *Interdisciplinary*. Dengan bimbingan dari para ahli industri, peserta terlibat dalam proyek nyata dan pelatihan praktis yang menggunakan

teknologi terbaru. Program ini menawarkan tiga jalur pembelajaran utama, yaitu Machine Learning, Mobile Development, dan Cloud Computing yang saling melengkapi dan memberikan peserta keahlian yang komprehensif. Melalui kombinasi prinsip-prinsip ini, Bangkit berkomitmen untuk membekali peserta dengan keterampilan teknis, soft skills, dan kecakapan bahasa Inggris yang diperlukan untuk sukses di dunia kerja dan beradaptasi dalam lingkungan industri yang dinamis

BAB III TINJAUAN PUSTAKA

3.1 Machine Learning

Machine learning (ML) adalah cabang dari ilmu komputer yang bertujuan untuk memungkinkan komputer “belajar” tanpa perlu diprogram secara eksplisit. Asal-usulnya dapat ditelusuri kembali ke gerakan kecerdasan buatan (AI) pada tahun 1950-an, dengan fokus pada tujuan praktis dan aplikasi, terutama prediksi dan optimisasi [1]. Dalam machine learning, komputer meningkatkan kinerja tugas tertentu melalui “pengalaman” yang dalam praktiknya berarti menyesuaikan diri dengan data.

Machine learning sering dianggap sebagai subset dari kecerdasan buatan (AI), di mana perangkat lunak komputer menggunakan algoritma untuk mengenali pola dalam data dan membuat keputusan atau rekomendasi secara otonom [11]. Algoritma ini dirancang untuk belajar dan memperbaiki analisis mereka melalui penggunaan data input dan output dalam jumlah besar. Melalui repetisi yang cukup dan modifikasi algoritma, sistem ML dapat memprediksi hasil yang akurat berdasarkan data baru yang diberikan.

Machine learning telah diadopsi di berbagai industri karena kemampuannya untuk menganalisis data dalam jumlah besar dan menghasilkan prediksi yang akurat [1]. Dalam bidang kesehatan, misalnya, ML digunakan untuk menganalisis data medis dan membantu dalam diagnosis penyakit, rekomendasi perawatan, dan analisis gambar medis.

3.2 Recommendation System

Recommendation system adalah salah satu aplikasi utama dari data mining dan machine learning yang bertujuan untuk memahami preferensi pengguna dan memberikan rekomendasi yang sesuai. Sistem ini

bekerja dengan menggunakan data eksplisit dan implisit dari pengguna, seperti riwayat interaksi dan umpan balik, untuk memperkirakan preferensi pengguna melalui berbagai teknik data mining [12]. Berdasarkan analisis data ini, sistem rekomendasi menyaring informasi produk atau layanan untuk menawarkan rekomendasi yang disesuaikan dengan selera atau kebutuhan pengguna. Penelitian di bidang ini terus berkembang untuk memperbaiki kinerja sistem rekomendasi, dengan mengembangkan model yang lebih canggih dan teknik evaluasi korelasi antara pengguna untuk memberikan rekomendasi yang lebih akurat.

Sistem rekomendasi memainkan peran penting dalam berbagai aplikasi online, mulai dari e-commerce, streaming media, hingga jaringan sosial. Internet memungkinkan pengawasan berbagai aktivitas pengguna tanpa mengganggu aktivitas mereka, mengumpulkan data dari kelompok sosial yang beragam di seluruh dunia. Tantangan utama dalam sistem berbasis data mining adalah analisis basis informasi yang sangat besar. Teknik machine learning, seperti analisis regresi dan korelasi, neural network, dan logika fuzzy, membantu mengatasi masalah ini dengan menciptakan algoritma rekomendasi yang dapat mengolah data secara efektif [15]. Sistem ini sangat berharga dalam menghadapi banjir informasi yang tersedia di internet, membantu pengguna menemukan konten yang relevan di antara ribuan film, jutaan artikel ilmiah, dan jumlah musik yang tak terhitung.

Dengan perkembangan teknologi internet dan machine learning, sistem rekomendasi telah menjadi semakin populer dan penting. Penelitian di bidang ini didorong oleh kebutuhan untuk mengatasi jumlah informasi yang besar dan memberikan rekomendasi yang relevan kepada pengguna. Penggunaan algoritma yang kompleks dan data besar memungkinkan sistem ini untuk memberikan saran yang tepat, yang tidak hanya membantu pengguna dalam membuat keputusan yang lebih baik tetapi juga meningkatkan pengalaman

pengguna secara keseluruhan [15].

3.3 K-Nearest Neighbors

K-Nearest Neighbors (KNN) adalah algoritma machine learning yang digunakan untuk klasifikasi dan regresi. Algoritma ini bekerja dengan mengidentifikasi k sampel pelatihan terdekat (tetangga) untuk menentukan label atau nilai dari titik data yang tidak berlabel. KNN dikategorikan sebagai metode pembelajaran non-parametrik dan berbasis instansi, yang berarti ia tidak membuat asumsi apa pun tentang distribusi data yang mendasarinya dan menyimpan seluruh dataset pelatihan untuk melakukan klasifikasi atau prediksi baru [20]. Proses perhitungan dilakukan ketika klasifikasi atau prediksi dilakukan, yang membuatnya dikenal sebagai model "lazy learning".

Untuk menentukan tetangga terdekat, KNN menggunakan metrik jarak, dengan Euclidean distance sebagai metode yang paling umum digunakan untuk variabel kontinu. Alternatif lain seperti Manhattan distance, Minkowski distance, dan Hamming distance dapat digunakan tergantung pada jenis variabel dan konteks data [20]. Nilai k, yang mewakili jumlah tetangga yang dipertimbangkan, merupakan parameter penting dalam algoritma KNN. Pemilihan nilai k yang tepat sangat penting untuk keseimbangan antara overfitting dan underfitting. Nilai k yang kecil dapat menyebabkan model overfit, sementara nilai k yang besar dapat menyebabkan underfit. Teknik cross-validation sering digunakan untuk menentukan nilai k yang optimal.

KNN memiliki berbagai aplikasi, termasuk sistem rekomendasi, pengenalan pola, prediksi pasar keuangan, deteksi intrusi, dan diagnosis medis. Dalam sistem rekomendasi, KNN menggunakan data clickstream untuk memberikan rekomendasi otomatis kepada pengguna berdasarkan perilaku pengguna lain yang serupa. Meski KNN sederhana dan mudah

diimplementasikan, ia memiliki keterbatasan dalam hal skalabilitas dan kinerja dengan dataset besar, yang sering kali memerlukan teknik tambahan untuk meningkatkan efisiensi komputasi.

3.4 Python

Python adalah bahasa pemrograman tingkat tinggi yang ditafsirkan dan berorientasi objek dengan semantik dinamis. Bahasa ini dikenal karena sintaksnya yang sederhana dan mudah dipelajari, yang menekankan keterbacaan kode dan mengurangi biaya pemeliharaan program. Struktur data built-in tingkat tinggi, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuat Python sangat menarik untuk pengembangan aplikasi cepat serta untuk digunakan sebagai bahasa scripting atau penghubung untuk mengintegrasikan komponen yang ada. Python mendukung modularitas program dan penggunaan ulang kode melalui dukungan untuk modul dan paket [19].

Python digunakan secara luas dalam berbagai aplikasi mulai dari pengembangan web, ilmu data, hingga kecerdasan buatan dan pembelajaran mesin. Dukungan komunitas yang kuat dan pengembangan berkelanjutan oleh Python Software Foundation memastikan bahwa Python tetap menjadi pilihan utama bagi pengembang di seluruh dunia. Python menyediakan koleksi pustaka standar yang luas, mencakup berbagai tugas seperti manipulasi string, operasi file, dan pengembangan aplikasi web. Pustaka-pustaka populer seperti NumPy, SciPy, Pandas, dan Matplotlib memperkaya ekosistem Python, menjadikannya alat yang sangat diperlukan bagi ilmuwan data dan insinyur perangkat lunak [19].

3.5 NumPy

NumPy adalah pustaka open source yang fundamental untuk komputasi ilmiah di Python. Dibuat pada tahun 2005, NumPy menggantikan dua pustaka

sebelumnya, Numeric dan Numarray, dengan menggabungkan keunggulan keduanya untuk memberikan kinerja yang lebih baik dan antarmuka yang lebih kaya. NumPy menyediakan objek array multidimensi yang homogen serta berbagai fungsi untuk operasi cepat pada array, termasuk operasi matematika, logika, manipulasi bentuk, pengurutan, pemilihan, I/O, transformasi Fourier, aljabar linear dasar, operasi statistik dasar, dan simulasi acak [10].

Inti dari NumPy adalah objek ndarray, yang memungkinkan pengelolaan array n-dimensi dari tipe data homogen. NumPy terkenal karena efisiensinya dalam mengelola operasi elemen demi elemen melalui kode yang telah dikompilasi sebelumnya, memberikan kinerja mendekati C dengan kesederhanaan kode Python. Ini membuat NumPy sangat berguna dalam aplikasi ilmiah dan numerik, termasuk dalam bidang data science, pembelajaran mesin, dan komputasi ilmiah [10].

NumPy juga merupakan basis bagi banyak pustaka Python lainnya yang digunakan dalam komputasi ilmiah dan analisis data, seperti SciPy untuk operasi ilmiah tambahan, Matplotlib untuk visualisasi data, Pandas untuk analisis data, dan Scikit-learn untuk pembelajaran mesin. Dengan komunitas yang besar dan aktif, NumPy terus dikembangkan dan diperbarui untuk mendukung kebutuhan teknologi modern, termasuk integrasi dengan GPU dan TPU melalui pustaka seperti CuPy dan TensorFlow untuk komputasi yang lebih cepat dan lebih efisien.

3.6 Pandas

Pandas adalah pustaka open-source yang sangat kuat untuk analisis dan manipulasi data dalam Python. Diperkenalkan oleh Wes McKinney pada tahun 2008, Pandas menyediakan struktur data cepat, fleksibel, dan ekspresif yang dirancang untuk membuat pekerjaan dengan data "relasional" atau "berlabel" menjadi mudah

dan intuitif. Pandas berfungsi sebagai blok bangunan tingkat tinggi yang esensial untuk analisis data praktis di Python dan bertujuan untuk menjadi alat analisis/manipulasi data open-source yang paling kuat dan fleksibel yang tersedia dalam bahasa apapun [3].

Pustaka ini terkenal karena dua struktur data utamanya: Series (array 1-dimensi berlabel) dan DataFrame (struktur tabular 2-dimensi berlabel, yang ukuran kolomnya dapat diubah). Pandas dibangun di atas pustaka NumPy dan dirancang untuk berintegrasi dengan baik dalam lingkungan komputasi ilmiah bersama pustaka pihak ketiga lainnya. Hal ini memungkinkan penanganan data yang efisien dan mendukung berbagai operasi, seperti pengelolaan data yang hilang, pengelompokan data, penggabungan dan penggabungan data set, serta manipulasi data deret waktu.

Pandas telah menjadi bagian integral dari ekosistem data science, digunakan secara luas dalam keuangan, statistik, ilmu sosial, dan berbagai bidang rekayasa. Pustaka ini memungkinkan peneliti dan analis untuk memuat data dari berbagai format (seperti CSV, Excel, SQL, dan HDF5), melakukan transformasi data yang kompleks, dan menghasilkan analisis yang siap untuk visualisasi dan pelaporan [3]. Dengan komunitas yang aktif dan dukungan dari berbagai perusahaan dan lembaga, Pandas terus berkembang untuk memenuhi kebutuhan teknologi modern dalam analisis data.

3.7 Scikit-learn

Scikit-learn adalah pustaka open source untuk machine learning yang ditulis dalam Python. Pustaka ini memungkinkan integrasi metode machine learning secara mudah dan cepat ke dalam kode Python. Scikit-learn menyediakan berbagai metode untuk klasifikasi, regresi, estimasi matriks kovarians, pengurangan dimensi, praproses data, dan pembuatan masalah benchmark [2]. Pustaka ini dirancang agar mudah

diakses oleh semua orang dan dapat digunakan dalam berbagai konteks, baik untuk penelitian maupun aplikasi komersial.

Scikit-learn mencakup fungsi tambahan yang integral dalam pipeline machine learning, seperti langkah praproses data, teknik resampling data, parameter evaluasi, dan antarmuka pencarian untuk pengoptimalan performa algoritma. Pustaka ini dibangun di atas NumPy, SciPy, dan Matplotlib, yang membuatnya kompatibel dengan ekosistem Python ilmiah lainnya [2]. Scikit-learn adalah proyek yang didorong oleh komunitas dengan dukungan finansial dari berbagai institusi dan grant pribadi, memastikan keberlanjutan dan pengembangannya.

Pustaka ini sangat populer dan banyak digunakan dalam berbagai proyek komersial dan penelitian. Scikit-learn terus dikembangkan dan diperbarui, dengan rilis terbaru yang mencakup berbagai perbaikan dan penambahan fitur baru. Dengan menyediakan alat yang efisien untuk analisis data prediktif dan antarmuka yang standar untuk implementasi algoritma machine learning, scikit-learn telah menjadi alat yang esensial bagi para peneliti dan praktisi di bidang data science dan machine learning [2].

3.8 Cloud Computing

Cloud Computing atau komputasi awan merupakan model pengiriman layanan melalui internet yang memungkinkan akses fleksibel dan cepat terhadap sumber daya komputasi seperti server, penyimpanan, basis data, jaringan, dan aplikasi [17].

National Institute of Standards and Technology (NIST) [14] menyebutkan bahwa komputasi awan memiliki lima karakteristik yang mendasar yaitu:

3.8.1 *On-demand self-service*

On-demand self-service memungkinkan pengguna layanan komputasi awan dapat

menyediakan kemampuan komputasi seperti server, jaringan, dan penyimpanan sesuai dengan kebutuhan secara otomatis tanpa memerlukan interaksi manusia dengan penyedia layanan awan

3.8.2 *Broad network access*

Broad network access memungkinkan layanan tersedia dan dapat diakses melalui jaringan internet dan dari berbagai macam jenis platform yang digunakan user seperti ponsel, tablet, PC, dan sebagainya.

3.8.3 *Resource pooling*

Penyedia layanan komputasi awan menyediakan kumpulan sumber daya komputasi yang dapat digunakan untuk banyak konsumen dengan menggunakan model *multi-tenant*, dengan sumber daya fisik dan virtual yang ditugaskan dan dipindahkan secara dinamis sesuai dengan permintaan pengguna.

3.8.4 *Rapid elasticity*

Karakteristik *rapid elasticity* memiliki kemampuan untuk menyediakan dan menghentikan sumber daya secara fleksibel baik secara horizontal maupun vertikal sesuai dengan permintaan

3.8.5 *Measured service*

Sistem komputasi awan secara otomatis dapat mengontrol dan melakukan optimasi penggunaan sumber daya dengan memanfaatkan kemampuan pengukuran pada beberapa tingkat abstraksi yang sesuai dengan jenis layanan. Penggunaan sumber daya bisa dipantau, dikendalikan, dan dilaporkan, sehingga memberikan transparansi bagi penyedia dan pengguna layanan.

Penyedia komputasi awan menawarkan "layanan" mereka menurut model yang berbeda, di mana tiga

model standar menurut NIST adalah *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS), dan *Software as a Service* (SaaS) [14]. Model-model ini menawarkan peningkatan abstraksi, karena itu model-model ini sering digambarkan sebagai lapisan dalam sebuah tumpukan, infrastruktur, *platform*, dan *software* sebagai layanan, tetapi hal ini tidak perlu saling berkaitan.

3.8.6 *Infrastructure as a Service*

"*Infrastructure as a Service*" (IaaS) merupakan layanan yang fleksibel untuk pengguna layanan dalam membangun dan mengatur infrastrukturnya sendiri. Dengan menggunakan IaaS, pengguna dapat dengan bebas menentukan sumber daya yang diperlukan dalam membangun infrastrukturnya, sementara penyedia layanan awan bertanggung jawab atas manajemen perangkat lunak dan perawatannya.

3.8.7 *Platform as a service*

"*Platform as a service*" (PaaS) adalah layanan yang menyediakan lingkungan untuk *testing* dan *deployment aplikasi*. Infrastruktur komputasi awan telah diatur dan menjadi tanggung jawab dari penyedia layanan sehingga pengguna dapat fokus dalam mengembangkan aplikasi.

3.8.8 *Software as a service*

"*Software as a service*" merupakan layanan yang disediakan oleh penyedia layanan kepada pengguna berupa aplikasi yang berjalan di atas infrastruktur komputasi awan. Pada SaaS, pengguna tidak memiliki tanggung jawab dan kontrol terhadap infrastruktur yang ada seperti server, jaringan, sistem operasi dan sebagainya.

3.9 Google Cloud Platform

Google Cloud Platform (GCP) adalah layanan

komputasi awan publik yang ditawarkan oleh Google. Layanan ini mencakup berbagai aspek seperti komputasi, penyimpanan, analisis data, pembelajaran mesin, dan pengelolaan aplikasi. GCP memungkinkan pengembang dan perusahaan untuk membangun, menguji, dan menerapkan aplikasi dalam skala global menggunakan infrastruktur yang sama dengan yang digunakan oleh Google untuk produk-produk Google secara internal seperti Google Search, Google Drive, Gmail, dan Youtube [6].

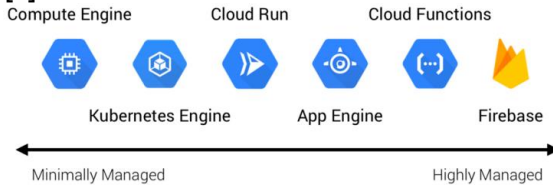
Google Cloud Platform adalah bagian dari Google Cloud, yang mencakup infrastruktur cloud publik Google Cloud Platform, serta Google Workspace (G Suite), versi perusahaan Android dan Chrome OS, dan *Application Programming Interface* (API) untuk *Machine Learning* dan layanan pemetaan perusahaan.

Pada 2019, Google mengumumkan Cloud Run, sebuah platform untuk mengembangkan dan menghosting aplikasi web di *data center* yang dikelola Google, yang merupakan layanan komputasi awan pertama dari perusahaan. Layanan ini tersedia secara umum pada November 2011. Sejak diluncurkannya App Engine, Google menambahkan beberapa layanan cloud ke GCP.

3.9.1 Cloud Run

Google Cloud Run adalah salah satu layanan komputasi tanpa server (*serverless*) yang mengkombinasikan kemudahan penggunaan dari *Platform as a Service* (Paas) dengan fleksibilitas dan portabilitas yang mengotomatisasi penempatan dan skalabilitas aplikasi container. Layanan Cloud Run memungkinkan pengembang untuk menjalankan aplikasi berbasis container yang ditulis dalam bahasa pemrograman apapun, dengan menggunakan kerangka kerja (*framework*) atau pustaka yang dipilih tanpa perlu mengelola infrastruktur server. Cloud Run didasarkan pada

teknologi open-source Knative, yang menyediakan antarmuka yang seragam untuk menjalankan aplikasi container di berbagai lingkungan komputasi awan dan *on-premises* [7].



Gambar 3.7.1.1 GCP services berdasarkan tingkat manajemen penggunaannya

3.10 RESTful API

Representational state transfer (REST) merupakan gaya arsitektur perangkat lunak yang digunakan untuk mendesain layanan web seperti World Wide Web (WWW) yang memungkinkan komunikasi antara klien dan server melalui protokol HTTP. REST memanfaatkan HTTP *methods* seperti GET, POST, PUT, dan DELETE untuk melakukan operasi pada sumber daya yang diidentifikasi oleh URL (*Uniform Resource Locator*). REST merupakan model arsitektur yang bertujuan untuk meningkatkan kinerja, skalabilitas, kesederhanaan, dan portabilitas komunikasi antara sistem terdistribusi. REST memungkinkan pengembang untuk membangun sistem yang dapat diskalakan, efisien dan mudah dipelihara [5].

Gaya arsitektur REST terdiri dari enam karakteristik utama. Ketika karakteristik ini diaplikasikan, sistem dapat memperoleh properti non-fungsional seperti kinerja, skalabilitas, kesederhanaan, visibilitas, portabilitas, kemampuan dimodifikasi, dan keandalan. Sebuah sistem memenuhi beberapa atau semua karakteristik ini dapat disebut sebagai RESTful. Karakteristik REST API adalah sebagai berikut [5]:

3.10.1 *Client-server architecture*

Arsitektur client server menerapkan prinsip pemisahan masalah dengan memisahkan antarmuka pengguna dari penyimpanan data. Hal ini meningkatkan portabilitas antarmuka pengguna. Dalam lingkup web, browser web dikembangkan untuk berbagai platform tanpa memerlukan pengetahuan tentang implementasi server yang mendasarinya. Pemisahan ini juga menyederhanakan komponen server, meningkatkan skalabilitas, dan memungkinkan komponen berkembang secara independen, yang sangat penting dalam lingkungan berskala besar yang melibatkan banyak domain organisasi.

3.10.2 *Stateless*

Stateless merupakan pembatas dimana setiap permintaan dari klien ke server harus berisi semua informasi yang diperlukan untuk memproses permintaan tersebut tanpa menyimpan informasi di sisi server untuk referensi di masa mendatang. Klien bertanggung jawab untuk menyimpan dan mengelola informasi terkait sesi. Jika diperlukan informasi dari permintaan sebelumnya, klien akan menyertakannya dalam permintaan saat ini.

3.10.3 *Cacheability*

Ketika konsumen meminta representasi sumber daya, permintaan tersebut melewati cache atau serangkaian cache (cache local, proxy, atau cache balik) menuju layanan yang menghosting sumber daya. Jika salah satu cache di sepanjang jalur permintaan memiliki salinan baru dari representasi yang diminta, salinan tersebut digunakan untuk memenuhi permintaan. Jika tidak, permintaan dikirim ke layanan asal. Melalui header HTTP, server asal

menentukan apakah respons dapat di-cache, oleh siapa, dan untuk berapa lama. Cache di sepanjang jalur respons dapat menyimpan salinan respons, selama metadata caching mengizinkannya.

3.10.4 Layered System

REST memungkinkan penggunaan arsitektur sistem berlapis di mana API dapat diterapkan di server A, data disimpan di server B, dan autentikasi dilakukan di server C. Klien biasanya tidak dapat mengetahui apakah mereka terhubung langsung ke server akhir atau melalui perantara. Proxy atau load balancer yang ditempatkan antara klien dan server tidak mempengaruhi komunikasi mereka dan tidak memerlukan pembaruan kode klien atau server. Server perantara dapat meningkatkan skalabilitas sistem dengan mengaktifkan load balancing dan menyediakan cache bersama. Keamanan dapat ditambahkan sebagai lapisan terpisah, memisahkan logika bisnis dari logika keamanan, dan memberlakukan kebijakan keamanan. Server perantara juga dapat memanggil beberapa server lain untuk memberikan respons ke klien.

3.10.5 Uniform Interface

Uniform Interface memungkinkan klien dan server berkomunikasi dalam satu bahasa, terlepas dari arsitektur backend mereka. Antarmuka ini menyediakan sarana komunikasi standar yang konsisten antara klien dan server, seperti penggunaan HTTP dengan sumber daya URI, operasi CRUD (Create, Read, Update, Delete), dan format data seperti JSON.

3.10.6 Code On Demand

Code on Demand (COD) adalah satu-satunya

kendala opsional dalam REST. Ini memungkinkan klien untuk meningkatkan fleksibilitas mereka karena server menentukan bagaimana tugas tertentu dilakukan. Misalnya, klien dapat mengunduh kode seperti JavaScript, applet Java, atau aplikasi Flash untuk mengenkripsi komunikasi, sehingga server tidak mengetahui rutinitas atau kunci enkripsi yang digunakan. Namun, penggunaan COD mengurangi visibilitas, sehingga batasan ini bersifat opsional. Tidak semua API membutuhkan fleksibilitas semacam ini.

3.11 Hapi.js

Hapi.js merupakan kerangka kerja (*framework*) aplikasi web untuk Node.js yang dirancang untuk membangun aplikasi dan layanan yang skalabel dan dapat dipelihara dengan mudah. *Framework* Hapi dikenal dengan kemudahan konfigurasi dan ekosistem plugin yang kaya sehingga memudahkan pengembang untuk menambahkan fitur dan fungsionalitas [9].

Hapi memfasilitasi pembuatan REST API dengan menyediakan alat dan struktur yang memungkinkan pengembang untuk dengan mudah mendefinisikan rute, menangani permintaan dan mengelola respons. Berikut adalah beberapa cara Hapi untuk mendukung implementasi REST API:

3.11.1. Routing yang Fleksibel

Hapi.js menyediakan sistem routing yang kuat yang memungkinkan pengembang untuk mendefinisikan rute RESTful dengan mudah. Rute ini dapat dikonfigurasi untuk menangani berbagai metode HTTP seperti GET, POST, PUT, dan DELETE. Pengembang dapat menentukan handler untuk setiap rute yang akan memproses permintaan dan mengembalikan respons yang sesuai.

3.11.2. Pengelolaan Respons

Hapi.js memudahkan pengelolaan respons HTTP dengan menyediakan berbagai metode untuk mengirim respons ke klien. Pengembang dapat mengatur status kode HTTP, mengirim data dalam format JSON, dan menambahkan header HTTP yang diperlukan.

3.11.3. Ekosistem Plugin

Hapi.js memiliki ekosistem plugin yang kaya, yang memungkinkan pengembang untuk dengan mudah menambahkan fungsionalitas tambahan seperti autentikasi, logging, dan manajemen sesi. Plugin seperti hapi-auth-jwt2 dapat digunakan untuk menambahkan otentikasi berbasis token ke REST API.

3.11.4. Keamanan

Hapi.js menyediakan berbagai fitur keamanan bawaan yang dapat digunakan untuk melindungi REST API dari serangan umum seperti CSRF, XSS, dan injeksi SQL. Pengembang dapat menggunakan alat dan konfigurasi yang disediakan oleh Hapi.js untuk memastikan API mereka aman.

3.12 Firebase

Firebase adalah platform pengembangan aplikasi yang disediakan oleh Google, yang menawarkan berbagai alat dan infrastruktur untuk membangun, meningkatkan, dan mengelola aplikasi. Firebase menyediakan berbagai layanan seperti database real-time, otentikasi, analitik, hosting, dan notifikasi push, yang memungkinkan pengembang untuk fokus pada pengembangan fitur aplikasi tanpa harus khawatir tentang manajemen infrastruktur. Google memperluas Firebase dengan menambahkan berbagai fitur dan integrasi dengan layanan Google lainnya, menjadikannya platform komprehensif untuk pengembangan aplikasi mobile dan web. Salah satu layanan yang disediakan firebase adalah Firebase

Realtime Database yang merupakan database noSQL yang menyimpan data dalam format JSON dan memungkinkan sinkronisasi data secara real-time antara klien dan server [8].

3.13 Android Studio

Android Studio adalah Integrated Development Environment (IDE) yang digunakan untuk membangun aplikasi Android, berdasarkan IntelliJ IDEA. IntelliJ IDEA sendiri merupakan Java Integrated Development Environment yang dikembangkan oleh JetBrains untuk mengembangkan perangkat lunak komputer. IntelliJ IDEA berfungsi untuk membantu dalam pembuatan coding dari segi navigasi, mendukung produktivitas, hingga menyediakan kode editor yang cerdas [16].

Kelebihan Android Studio meliputi iterasi peng-codingan yang cepat berkat IntelliJ IDEA yang memungkinkan coding dan running workflow dengan cepat. Android Studio memiliki konfigurasi pembangunan tanpa batas dengan struktur proyek dan pembangunan berbasis Gradle yang fleksibel, memungkinkan pembuatan APK untuk berbagai jenis perangkat. Fitur-fitur Android Studio membantu pengembang meng-coding dengan percaya diri untuk menghasilkan kode terbaik. Selain itu, Android Studio mempermudah pembuatan aplikasi yang kaya dan terkoneksi, serta menyediakan alat GUI yang mempermudah perancangan tampilan aplikasi, sehingga menghilangkan tugas-tugas yang melelahkan [4].

3.14 Kotlin

Kotlin adalah bahasa pemrograman statically-typed yang menggabungkan prinsip-prinsip object-oriented dengan fitur-fitur fungsional dan berjalan di atas Java Virtual Machine (JVM) [18]. Bahasa pemrograman ini dikembangkan oleh JetBrains sejak 2011 dan secara resmi didukung oleh Google untuk pengembangan aplikasi Android pada Mei 2017, seperti

diumumkan pada acara Google I/O 2017. Sejak event tersebut, popularitas Kotlin meningkat secara signifikan. Bahasa pemrograman ini dapat digunakan untuk pengembangan aplikasi Android, server-side, dan client-side. Kotlin dirancang untuk memiliki interoperabilitas penuh dengan Java, sehingga Kotlin dapat digabungkan dalam satu proyek aplikasi dengan bahasa Java.

Salah satu masalah umum yang sering ditemukan dalam bahasa pemrograman Java adalah `NullPointerException` (NPE). Kotlin didesain dengan `null-safety`, sehingga masalah tersebut tidak lagi ditemui dalam Kotlin. Pada bahasa Kotlin, NPE sudah dapat diketahui pada saat `compile time`, berbeda dengan Java yang melakukan pengecekan NPE pada saat `runtime`. Kotlin juga didukung dengan fitur `functional programming`, seperti penggunaan `lambda expression`, `higher-order function`, `lazy evaluation`, dan beberapa metode pada `collections` seperti `filtering`, `mapping`, `ordering`, dan lain-lain.

3.15 Model View ViewModel

Model View ViewModel (MVVM) adalah pola desain arsitektur yang memecah aplikasi menjadi tiga komponen inti: Model, View, dan ViewModel. Model berisi data mentah dan logika bisnis, termasuk kelas-kelas yang memodelkan data dan logika untuk mengakses serta memanipulasinya. Model bekerja secara independen dari View dan ViewModel, tanpa informasi tentang cara data ditampilkan di layar. View adalah lapisan antarmuka pengguna (UI) yang tidak mengandung logika bisnis, hanya fokus pada elemen visual yang dilihat oleh pengguna. View berkomunikasi dengan ViewModel, menampilkan informasi sesuai dengan cara yang disediakan oleh ViewModel, dan mengirimkan event pengguna (seperti klik tombol) ke ViewModel. ViewModel, sebagai jembatan antara Model dan View, mentransformasikan data dari Model ke

bentuk yang lebih mudah untuk ditampilkan oleh View, menangani semua logika presentasi dan input dari pengguna, sehingga View tetap ramping dan hanya menerima data siap tampil [13].

Pola desain MVVM memiliki beberapa kelebihan, termasuk pemisahan kode yang jelas yang membuat kode lebih terstruktur dan mudah dikelola, serta memudahkan kolaborasi antara designer UI dan developer dengan memisahkan View dan Model. Selain itu, MVVM memudahkan testing terhadap logika bisnis tanpa bergantung pada tampilan UI, dan memungkinkan perubahan tampilan tanpa harus mengubah logika bisnis. MVVM cocok digunakan dalam aplikasi dengan banyak logika dan UI yang kompleks, memberikan modularity dalam pengembangan, dan mendukung pengembangan kolaboratif. MVVM sangat populer dalam pengembangan aplikasi dengan kerangka kerja seperti WPF, Silverlight, aplikasi Windows Store, serta platform pengembangan seperti Xamarin dan React Native, karena membantu membangun aplikasi yang lebih bersih, mudah dipelihara, dan scalable.

BAB IV PEMBAHASAN

4.1 Rencana Pengembangan Produk

Dalam proses Magang di Bangkit 2024, kami berinisiatif untuk membuat sebuah platform aplikasi yang dapat mendukung gaya hidup sehat yang berkelanjutan di bidang makanan yang bernama Healthylicious yang memiliki fitur dengan menawarkan resep yang dipersonalisasi dan mudah diikuti menggunakan bahan-bahan rumah tangga yang umum ditemukan. Dengan mengintegrasikan pembelajaran mesin, komputasi awan, dan pengembangan seluler, Healthylicious bertujuan untuk menyediakan alat bagi pengguna untuk menjaga pola makan yang lebih sehat dengan mudah. Healthylicious memberikan panduan dan sumber daya untuk mendukung perjalanan menuju hidup sehat. Pada tahapan awal pembuatan, tim memulai dengan membuat *timeline* pengerjaan dengan gambaran umum sebagai berikut:

No	Task Schedule							Week 1							Week 2							Week 3							Week 4											
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7					
1	Preparation and Machine Learning Planning																																							
	Prepare dataset																																							
	Data wrangling																																							
	EDA & Pre-processing data																																							
	Designing model architecture (trial&error)																																							
	Model training																																							
	Model optimization																																							
	Model evaluation																																							
	Deployment																																							
2	Android Application Development																																							
	Build a list feature																																							
	UI Design																																							
	Develop application																																							
	User Testing																																							
3	Cloud Computing Development																																							
	Cloud Architecture Design Diagram																																							
	Build database																																							
	Build API																																							
	Deployment																																							
4	Testing																																							
	Alpha testing																																							
	Fix and solving problem																																							
5	Report																																							

Gambar 4.1.1 Timeline Pengerjaan Magang

Setelah membuat timeline, tim membagi tugas kepada seluruh anggota. Adapun untuk masing-masing

tim yang bekerja memiliki alur proses tersendiri dan akan dijelaskan pada poin selanjutnya.

4.2 Tahap Pengembangan

Pengembangan aplikasi Healthylicious dilakukan dengan kolaborasi dari tiga tim yaitu Tim Mobile Development sebagai pengembang aplikasi android Healthylicious, Tim Machine Learning sebagai pengembang algoritma pembelajaran mesin untuk fitur rekomendasi dan fitur *generate* resep, dan Tim Cloud Computing sebagai pengembang *backend* dan melakukan implementasi Google Cloud dari aplikasi Healthylicious.

4.2.1 Tim Mobile Development

Dalam tim Mobile Development, tugas kami adalah sebagai pembuatan *User Interface* atau tampilan utama aplikasi dan pengimplementasian fungsinya. Dengan memanfaatkan *Figma* untuk membuat model aplikasi dan *Android Studio* untuk pembuatan dan implementasi fungsi dari aplikasi.

Salah satu bagian dari pekerjaan tim Mobile Development adalah untuk mengaplikasikan data API yaitu *fetching* dan menampilkan datanya ke tampilan aplikasi. Pekerjaan saya fokus utamanya adalah untuk mengimplementasikan fungsi dan interaksi yang ada pada aplikasi. Sebagai team, teman saya satu tim Mobile Development akan membuat sebuah model *User Interface* kemudian tugas saya adalah memasukan model tersebut ke dalam *Android Studio* dan mengimplementasikan fungsinya dan tampilannya. Maka berikut akan dijelaskan salah satu contoh proses untuk menampilkan sebuah data dari API yaitu data profile.

4.2.1.1 Fetching Data dari API

Fetching data dari API dilakukan dengan cara membuat sebuah kelas bernama “*ApiConfig*” dimana kelas tersebut berisikan alamat URL dari *Healthylicious*: “<https://healthylicious-backend-z5ldsdy5xq-et.a.run.app/>”, kode nya adalah sebagai berikut:

```
class ApiConfig {
    companion object {
        fun getApiService(token: String): Api {
            val loggingInterceptor =
                HttpLoggingInterceptor().apply {
                    level =
```

```

HttpLoggingInterceptor.Level.BODY
    }
    val authInterceptor = Interceptor {
chain ->
    val req = chain.request()
    val requestHeaders =
req.newBuilder()
        .addHeader("Authorization",
"Bearer $token")
        .build()
        chain.proceed(requestHeaders)
    }
    val client = OkHttpClient.Builder()
        .addInterceptor(loggingInterceptor)
        .addInterceptor(authInterceptor)
        .build()
    val retrofit = Retrofit.Builder()
        .baseUrl("https://healthylicious-
backend-z5ldsdy5xq-et.a.run.app/")

.addConverterFactory(GsonConverterFactory.cr
eate())

.addCallAdapterFactory(CoroutineCallAdapterFa
ctory())

        .client(client)
        .build()
    return retrofit.create(Api::class.java)
    }
}
}

```

Pada kode tersebut juga terdapat token. Token disini memiliki peran penting yaitu sebagai kunci unik yang dimiliki user, dan berbeda setiap kali user melakukan login, yang digunakan untuk mengambil data API.

Setelah pembuatan kelas ini, maka akan

dituliskan *interface* Api, contohnya pada data profile, digunakan metode “GET(“user/profile/{uid}” dengan kode sebagai berikut:

```
@GET("user/profile/{uid}")
fun getUserProfile(
    @Path("uid") uid: String
): Call<ProfileResponse>
```

Pada kode tersebut dibuat sebuah fungsi “getUserProfile” untuk mengambil data dengan metode GET dan datanya diterima dalam bentuk “Profile Response”. Maka disini profile response adalah sebagai berikut:

```
data class ProfileResponse(
    val userProfile: UserProfile? = null
)

data class UserProfile(
    val username: String,
    val age: String,
    val weight: String, //in kg
    val height: String //in cm
)
```

Maka setelah data tersebut diterima, data akan diolah lebih lanjut pada View Model.

4.2.1.2 Pengolahan Data

Data yang sudah diterima dalam bentuk “Profile Response” akan disimpan ke dalam ViewModel. Awalnya dibuat sebuah repository dan didalamnya dibuat fungsi untuk mengimplementasikan fetching datanya:

```

fun getUserProfile(uid: String):
Call<ProfileResponse> {
    return apiService.getUserProfile(uid)
}

```

Setelah diambil dari repository, dibuat pula sebuah View Model. View Model disini berfungsi sebagai model data aplikasi yang ditampilkan dalam tampilan. Didalamnya akan ada fungsi sebagai berikut:

```

fun getUserProfile(uid: String){
    repository.getUserProfile(uid).enqueue(object:
    Callback<ProfileResponse> {
        override fun onResponse(
            call: Call<ProfileResponse>,
            response: Response<ProfileResponse>
        ) {
            if(response.isSuccessful &&
            response.body() != null){
                val profileResponse =
                response.body()
                if (profileResponse != null) {
                    saveUserProfile(profileResponse.userProfile!!)
                    _profileResult.value =
                    profileResponse
                    println("Saved profileResult")
                }
            } else {
                //TODO Error
                println("Response body error")
                val nullProfile =
                ProfileResponse(UserProfile("", "", "", ""))
                _profileResult.value = nullProfile
            }
        }
    })
}

```



```

        }
    }

    override fun onFailure(call:
    Call<ProfileResponse>, t: Throwable) {
        TODO("Not yet implemented")
    }
}
}
}

```

Fungsi ini nantinya akan dipanggil dan disimpan dalam local user preference. Local user preference adalah sebuah penyimpanan lokal yang disimpan pada device pengguna. Berikut adalah kode untuk menyimpan user profile ke local user preference:

```

suspend fun saveUserProfile(user: UserProfile) {
    datastore.edit { preferences ->
        preferences[USERNAME_KEY] =
        user.username
        preferences[AGE_KEY] = user.age
        preferences[WEIGHT_KEY] =
        user.weight
        preferences[HEIGHT_KEY] = user.height
    }
}
}

```

4.2.1.3 Menampilkan Data pada Tampilan Aplikasi

Kemudian proses selanjutnya adalah pengolahan data, dimana setelah data disimpan pada local user preference, data akan dipanggil untuk ditampilkan. Maka sebelumnya, akan dibuat sebuah *user interface* pada activity supaya data memiliki wadah atau tampilan. Berikut merupakan gambar interface user detail profile.



Gambar 4.2.1.1 Interface User Detail

Pada penampilan activity, agar interface dapat menampilkan data dengan baik, maka sebelum ditampilkan, akan dilakukan penggabungan data dengan interface. Karena setiap aset dalam tampilan aplikasi memiliki ID, maka dapat disambungkan dengan datanya. Contoh proses penyambungan adalah sebagai berikut:

```
val emailTextView: TextView =
view.findViewById(R.id.profile_email_value)
val usernameTextView: TextView =
view.findViewById(R.id.profile_username_value
)
val ageTextView: TextView =
view.findViewById(R.id.profile_age_value)
val weightTextView: TextView =
view.findViewById(R.id.profile_weight_value)
val heightTextView: TextView =
view.findViewById(R.id.profile_height_value)
val logoutButton: Button =
view.findViewById(R.id.button_logout)
val resetButton: Button =
view.findViewById(R.id.button_clear)
val dislikeButton: Button =
view.findViewById(R.id.button_update_dislike)
val profileDislikesValue: TextView =
view.findViewById(R.id.profile_dislikes_value)

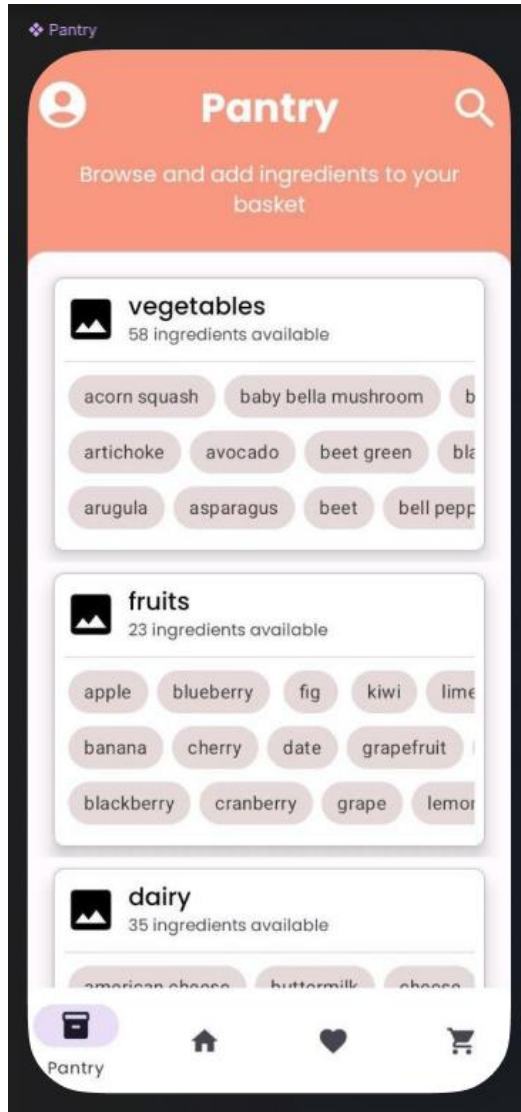
viewModel.getSession().observe(viewLifecycleO
wner) { user ->
    emailTextView.text = user.email
    usernameTextView.text =
user.extralnfo.username
    ageTextView.text = user.extralnfo.age
    weightTextView.text =
```

```
user.extrInfo.weight + " kg"  
    heightTextView.text =  
user.extrInfo.height + " cm"  
}
```

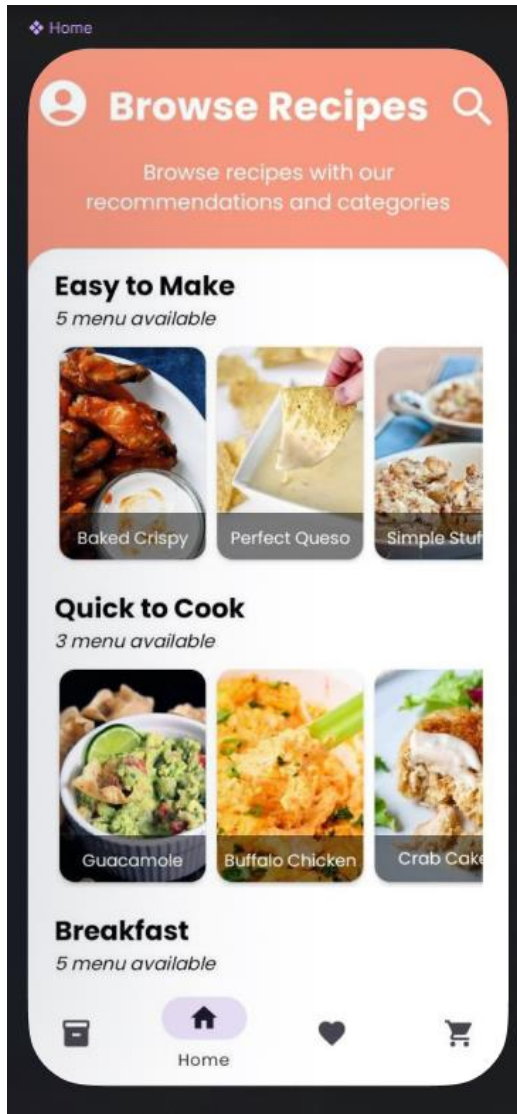
Maka setelah disambungkan, data akan otomatis tampil pada aplikasi.

4.2.1.4 User Interface Aplikasi

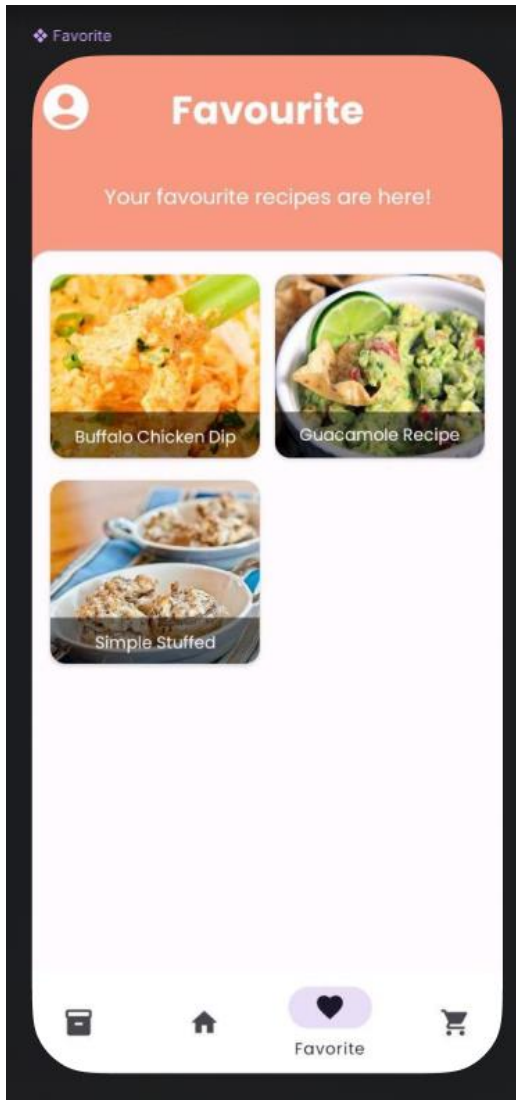
Maka dari contoh penerapan pengolahan data pada sub bab sebelumnya, saya melakukan hal yang hampir sama untuk berbagai fungsi yang dimiliki oleh aplikasi. Fungsi-fungsi aplikasi adalah *login*, *register*, *fetching user profile data*, menambahkan dan menghapus item dari favorit, menambahkan sebuah item untuk generate resep, generate resep, menampilkan list resep, menampilkan list rekomendasi resep, menampilkan detail resep, fitur search ingredient, fitur search resep, *fetching recipe data*, mengirimkan rating resep dari user, mengimplementasikan tombol *logout* dan implementasi tombol untuk menghapus data local. Gambar berikut mewakili UI/UX yang sudah dibuat oleh tim dari Mobile Development untuk *User Interface* dari aplikasi Healthylicious.



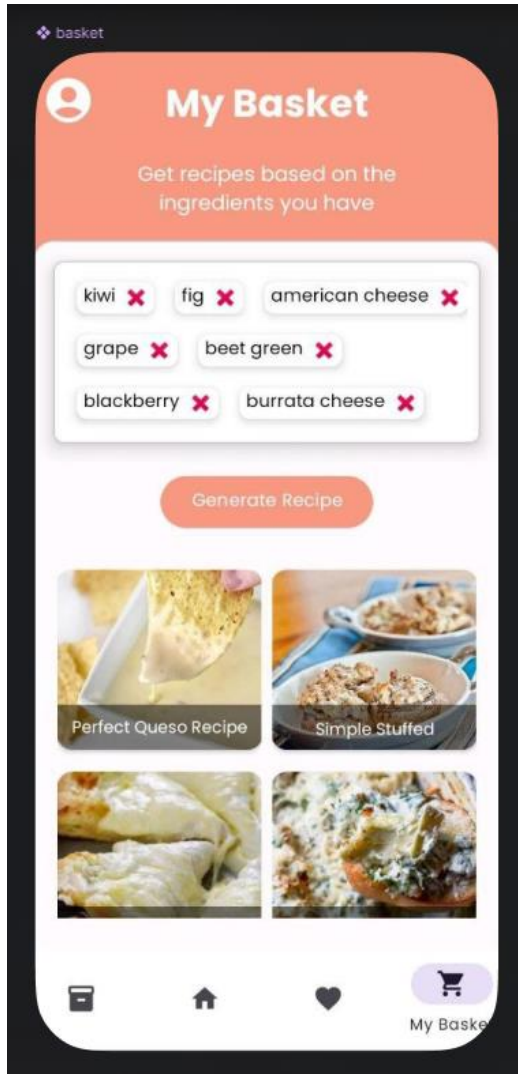
Gambar 4.2.1.2 UI/UX Aplikasi Pantry



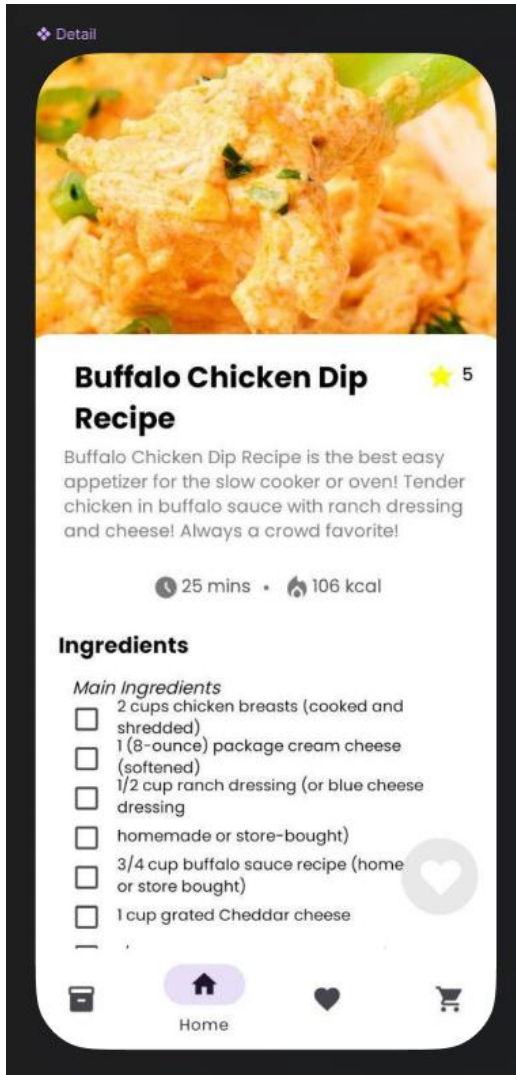
Gambar 4.2.1.3 UI/UX Aplikasi Home



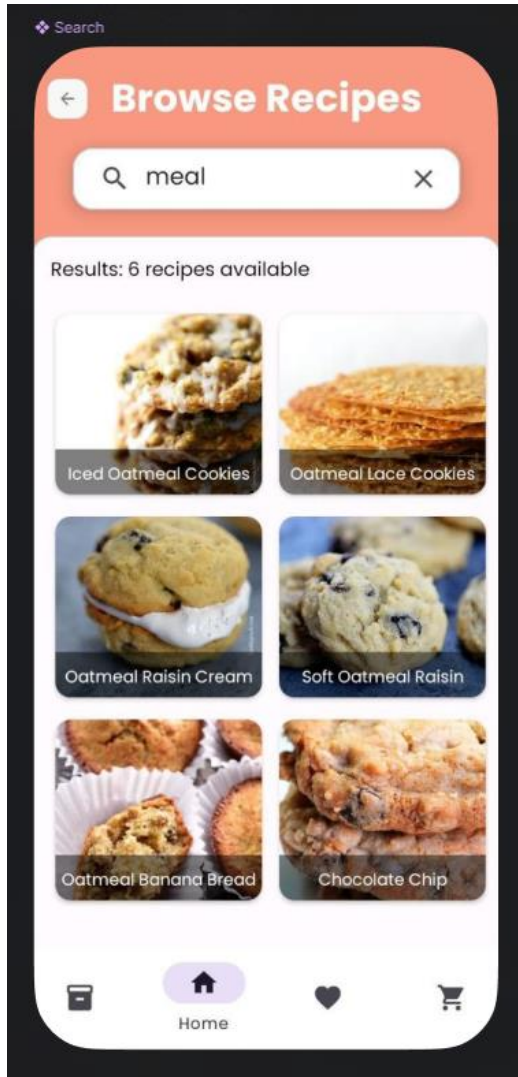
Gambar 4.2.1.4 UI/UX Aplikasi Favourite



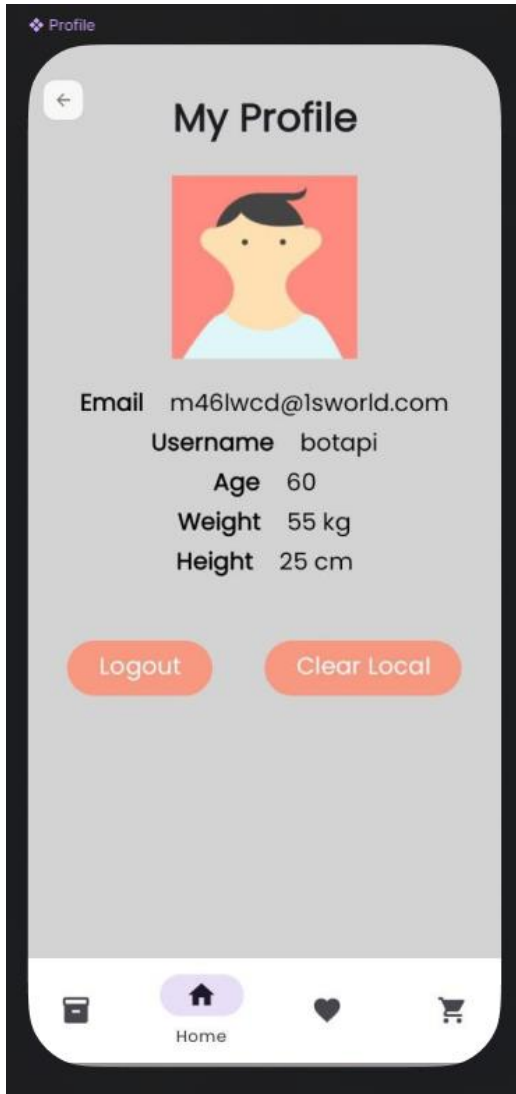
Gambar 4.2.1.5 UI/UX Aplikasi My Basket



Gambar 4.2.1.4 UI/UX Aplikasi Detailed Recipe



Gambar 4.2.1.5 UI/UX Aplikasi Browse Recipe



Gambar 4.2.1.6 UI/UX Aplikasi User Profile

4.2.1.5 Analisis Survey Form

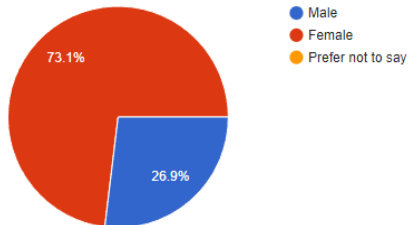
Dalam survey ini, akan diberikan 19 pertanyaan terdiri dari 4 *user information*, 4 relevansi fitur, 5 pertanyaan *user interface*, dan 6 pertanyaan feedback aplikasi. Survei ini akan diberikan kepada 52 responden calon user.

4 pertanyaan *User Information* mengandung pertanyaan yang mengidentifikasi identitas responden, seperti usia dan jenis kelamin. Berikut merupakan hasil dari kedua pertanyaan pertama dari *User Information*.

Gender

 Copy

52 responses

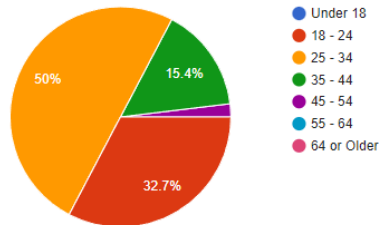


4.2.1.1 Diagram Hasil Responsi 1

Age

 Copy

52 responses



4.2.1.2 Diagram Hasil Responsi 2

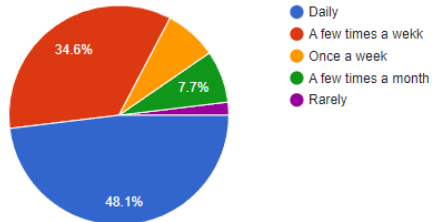
Dari kedua hasil survei tersebut, didapatkan bahwa dari 52 orang yang mengisi survei, sebanyak 38 responden berjenis kelamin wanita dan 14 berjenis kelamin pria. Kemudian dari pertanyaan kedua, mayoritas umur adalah sebanyak 26 orang berusia 64 tahun atau lebih, kemudian 17 orang berusia 18-24 tahun, kemudian 8 orang berusia 35-44 tahun, dan yang terakhir 1 orang berusia 45-54 tahun.

Kemudian pertanyaan selanjutnya, yaitu *“How often do you cook at home?”*. Pertanyaan ini melambangkan seberapa seringnya responden memasak di rumah. Berikut merupakan hasilnya.

How often do you cook at home?



52 responses



4.2.1.3 Diagram Hasil Responsi 3

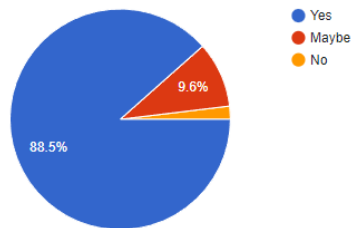
Hasil responsi dari pertanyaan ini, 25 orang menyatakan bahwa setiap hari mereka memasak di rumah, kemudian 18 orang menyatakan bahwa mereka memasak beberapa kali dalam seminggu, dan sisanya sebanyak 9 orang berada di kategori sekali seminggu, beberapa kali sebulan, dan jarang. Maka dari tabel ini dapat disimpulkan bahwa mayoritas dari responden lebih sering memasak di rumah.

Kemudian pertanyaan selanjutnya *“Would you be interested in a Recipe/Cooking Application to help you navigate the recipes and track your calories?”* untuk menggambarkan ketertarikan responden terhadap aplikasi ini. Berikut adalah hasil dari survei pertanyaan tersebut.

Would you be interested in a Recipe/Cooking Application to help you navigate the recipes and track your calories?



52 responses



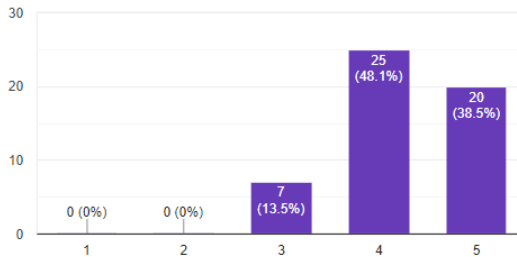
4.2.1.4 Diagram Hasil Responsi 4

Dari hasil survei tersebut, 46 orang menjawab ya, sedangkan 5 orang menjawab mungkin, dan 1 orang menjawab tidak tertarik. Hasil ini menggambarkan bahwa mayoritas dari responden akan tertarik terhadap aplikasi ini.

Kemudian selanjutnya adalah 4 pertanyaan relevansi fitur. Pertanyaan pertama adalah untuk menilai relevansi sebuah fitur dari skala 1 sampai 5. Fitur pertama yang dinilai adalah *Generate Recipe: you can generate recipes based on the ingredients you have. For example, you input the ingredient: Apple, and then you click on the 'generate recipe' button and then there will be displayed the recipes with Apple as an ingredient.* Dan berikut adalah hasil surveinya.

What do you think of this feature's relevance/importance on a Cooking App?

52 responses



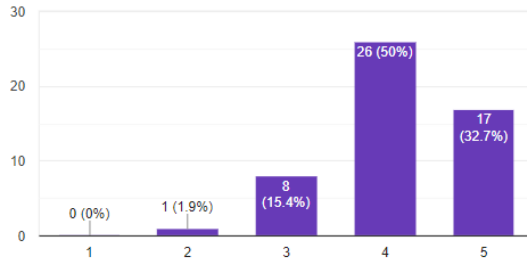
4.2.1.5 Diagram Hasil Responsi 5

Dari hasil survei didapati bahwa 7 orang memberikan nilai 3 dari 5, 25 orang memberikan nilai 4 dari 5, dan 20 orang memberikan nilai 5 dari 5. Dari hasil tersebut dapat disimpulkan bahwa mayoritas dari orang menganggap bahwa fitur *Generate Recipe* merupakan fitur yang relevan pada aplikasi ini.

Kemudian untuk fitur yang dinilai selanjutnya adalah fitur *Browse Recipe: this feature displays the recommended recipes for you. You can also search for a recipe.* Kemudian berikut merupakan hasil dari survei.

What do you think of this feature's relevance/importance on a Cooking App?

52 responses



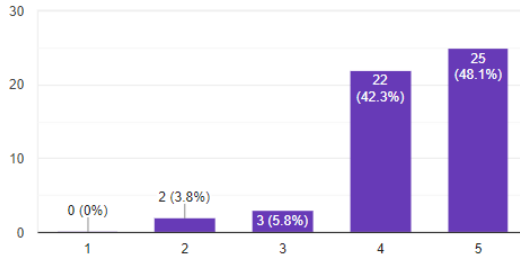
4.2.1.6 Diagram Hasil Responsi 6

Dari hasil survei tersebut, didapati bahwa 1 orang memberikan nilai 2 dari 5, 8 orang memberikan nilai 3 dari 5, 26 orang memberikan nilai 4 dari 5, dan 17 orang memberikan nilai 5 dari 5. Dari hasil tersebut, dapat disimpulkan bahwa fitur *Browse Recipe* merupakan fitur yang relevan pada aplikasi ini.

Kemudian fitur yang akan dinilai selanjutnya adalah fitur *Favourite Recipe: this feature displays the recipes you added as a favourite*. Kemudian berikut adalah hasil dari surveinya.

What do you think of this feature's
relevance/importance on a Cooking App?

52 responses



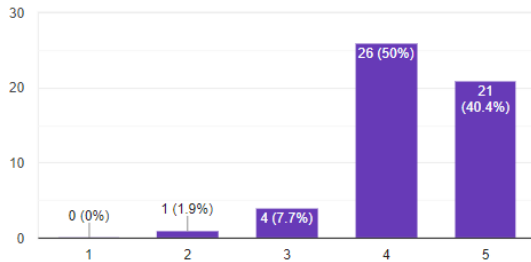
4.2.1.7 Diagram Hasil Responsi 7

Dari hasil survei yang didapat, 2 orang memberikan nilai 2 dari 5, 3 orang memberikan nilai 3 dari 5, 22 orang memberikan nilai 4 dari 5, dan 25 orang memberikan nilai 5 dari 5. Dari data tersebut, dapat disimpulkan bahwa fitur *Favourite Recipe* merupakan fitur yang relevan pada aplikasi ini.

Kemudian pada pertanyaan keempat, akan dinilai fitur *Dislike Recipe: this feature helps you remove all the related recipe with certain ingredients. This feature helps you so you do not have to worry if you have any allergies or disliking a certain ingredient.* Kemudian berikut adalah hasil dari survei tersebut.

What do you think of this feature's
relevance/importance on a Cooking App?

52 responses

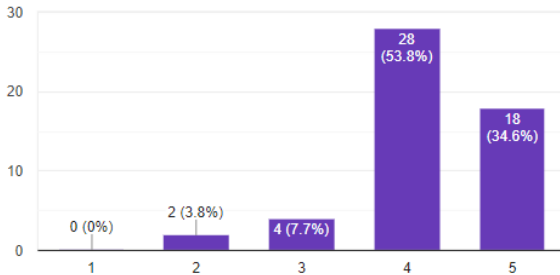


4.2.1.8 Diagram Hasil Responsi 8

Dari data yang didapatkan, 1 orang memberikan nilai 2 dari 5, 4 orang memberikan nilai 3 dari 5, 26 orang memberikan nilai 4 dari 5, dan 21 orang memberikan nilai 5 dari 5. Maka dari data yang didapat, dapat disimpulkan bahwa fitur *Dislike Recipe* merupakan fitur yang relevan pada aplikasi ini.

Kemudian selanjutnya akan ditunjukkan sebuah video demo dari aplikasi kemudian akan responden akan diberikan 5 pertanyaan dengan jawaban berupa skala 1-5 untuk menilai sebuah aspek dari aplikasi tersebut.

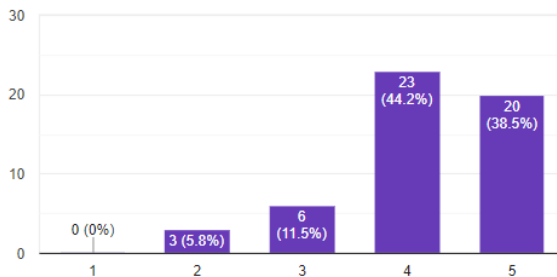
Pertanyaan pertama adalah *“How would you rate the User Interface of the application?”*. Berikut adalah hasil dari survei tersebut.



4.2.1.9 Diagram Hasil Responsi 9

Dari data yang didapatkan, 2 orang memberikan nilai 2 dari 5, 4 orang memberikan nilai 3 dari 5, 28 orang memberikan nilai 4 dari 5, dan 18 orang memberikan nilai 5 dari 5. Maka dari data tersebut, dapat disimpulkan bahwa mayoritas dari responden memberikan respon positif terhadap *User Interface* dari aplikasi dengan nilai rata-rata 4,2.

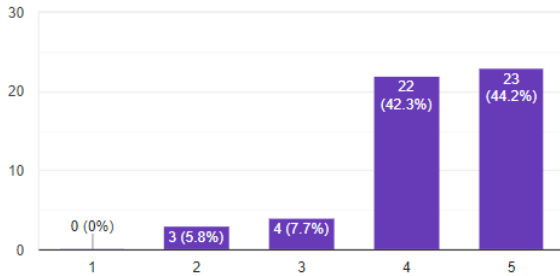
Kemudian selanjutnya terdapat pertanyaan “*How would you rate the User Friendly aspect (easy to use and navigate) of the application?*”. Berikut merupakan hasil dari survei.



4.2.1.10 Diagram Hasil Responsi 10

Dari data tersebut didapati bahwa 3 orang memberikan nilai 2 dari 5, 6 orang memberikan nilai 3 dari 5, 23 orang memberikan nilai 4 dari 5, dan 20 orang memberikan nilai 5 dari 5. Maka dari data tersebut, dapat disimpulkan bahwa untuk aspek *User Friendly* pada aplikasi, responden memberikan respon positif dengan nilai rata-rata 4,1.

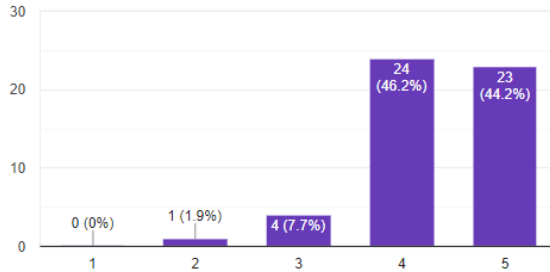
Kemudian selanjutnya terdapat pertanyaan “*How would you rate the color choice of our application?*”. Berikut merupakan hasil dari survei.



4.2.1.11 Diagram Hasil Responsi 11

Dari data yang didapat, 3 orang memberikan nilai 2 dari 5, 4 orang memberikan nilai 3 dari 5, 22 orang memberikan nilai 4 dari 5, dan 23 orang memberikan nilai 5 dari 5. Dari hasil data tersebut, didapati bahwa responden memiliki respon positif terhadap pilihan warna aplikasi dengan nilai rata-rata 4,2.

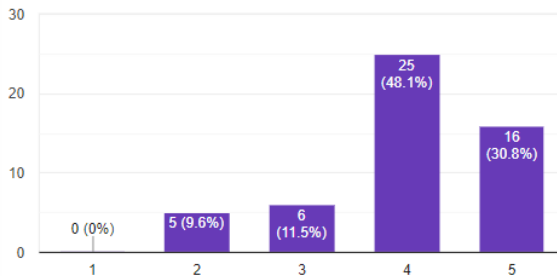
Kemudian untuk pertanyaan selanjutnya adalah “*How would you rate the information displayed in our application?*”. Kemudian didapati hasil survei sebagai berikut.



4.2.1.12 Diagram Hasil Responsi 12

Dari hasil data tersebut, didapati bahwa 1 orang memberikan nilai 2 dari 5, 4 orang memberikan nilai 3 dari 5, 24 orang memberikan nilai 4 dari 5, dan 23 orang memberikan nilai 5 dari 5. Maka dapat disimpulkan bahwa responden memberikan respon positif terhadap desain display information dengan nilai rata-rata 4,3.

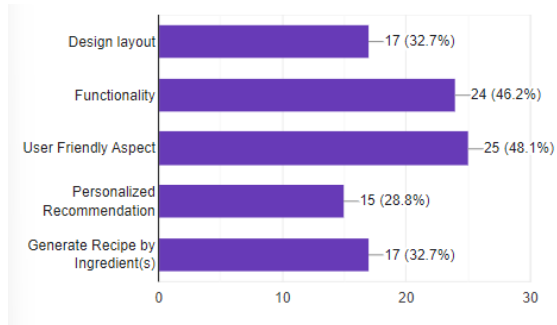
Kemudian selanjutnya terdapat pertanyaan “*How would you rate the logo of our application?*”. Kemudian berikut merupakan hasil dari survei.



4.2.1.13 Diagram Hasil Responsi 13

Dari hasil data tersebut, didapati 5 orang memberikan nilai 2 dari 5, 6 orang memberikan nilai 3 dari 5, 25 orang memberikan nilai 4 dari 5, dan 16 orang memberikan nilai 5 dari 5. Dari data tersebut, dapat disimpulkan bahwa responden memberikan respon positif terhadap logo aplikasi dengan nilai rata-rata 4.

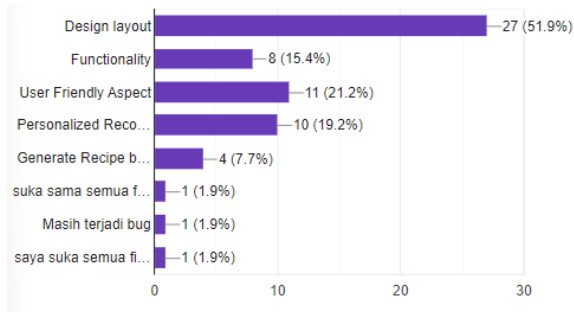
Untuk bagian selanjutnya merupakan bagian feedback aplikasi. Pertanyaan pertama adalah “*What do you like most about Healthylicious?*”. Berikut merupakan hasil dari survei.



4.2.1.14 Diagram Hasil Responsi 14

Dari hasil responsi, terhadap pertanyaan hal yang paling disukai, 17 orang menjawab *Design Layout*, 24 orang menjawab *Functionality*, 25 orang menjawab *User Friendly Aspect*, 15 orang menjawab *Personalized Recommendation*, dan 17 orang menjawab *Generate Recipe by Ingredient(s)*. Dari data yang didapat, dapat disimpulkan bahwa hal yang disukai oleh responden terbagi merata.

Kemudian terdapat pertanyaan “*What do you like least about Healthylicious?*”. Berikut adalah hasil survei.



4.2.1.15 Diagram Hasil Responsi 15

Dari data yang didapatkan, jawaban untuk hal yang paling tidak disukai, 27 orang menjawab *Design Layout*, 8 orang menjawab *Functionality*, 11 orang menjawab *User Friendly Aspect*, 10 orang menjawab *Personalized Recommendation*, 4 orang menjawab *Generate Recipe*, 2 orang menjawab suka semua, dan 1 orang menjawab masih terdapat bug. Maka dari responsi ini dapat disimpulkan bahwa aplikasi tersebut masih terdapat banyak yang harus dikembangkan lagi, terutama pada bagian design layout, user friendly aspect, dan personalized recommendation.

Kemudian pertanyaan “*What features would you like to see added or improved in future updates?*”. Dari jawaban responden, didapatkan poin sebagai berikut::

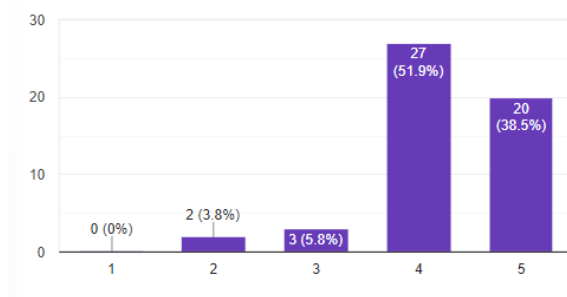
- Menambahkan fitur video tutorial
- Memperbaiki desain layout
- Mengintegrasikan dengan retail market
- Offline mode
- Login dengan sidik jari
- Video tutorial
- Menyediakan tips and trick

- Customizable serving size
- Prediksi harga makanan
- Collection theme meals
- Tambahkan komentar resep

Kemudian selanjutnya terdapat pertanyaan *“Any other suggestions or comments?”*. Jawaban dari pertanyaan ini beragam pula seperti pertanyaan sebelumnya, namun dapat disimpulkan dari poin berikut:

- Peningkatan kualitas resep
- Peningkatan kelancaran UI
- Peningkatan desain visual
- Perbanyak fitur
- Tambahkan community feature
- Perbaiki logo

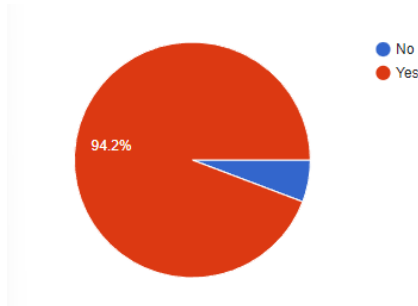
Lalu pada bagian terakhir yaitu bagian *Overall Satisfaction*, pertanyaan pertama adalah *“Overall, how satisfied are you with Healthylicious?”* dan responden menjawab dengan skala 1-5. Berikut merupakan hasil survei.



4.2.1.16 Diagram Hasil Responsi 16

Dari diagram diatas, disimpulkan bahwa 2 orang memberikan nilai 2 dari 5, 3 orang memberikan nilai 3 dari 5, 27 orang memberikan nilai 4 dari 5, dan 20 orang memberikan nilai 5 dari 5. Dari data tersebut, dapat disimpulkan bahwa responden memberikan respon positif terhadap nilai kepuasan aplikasi tersebut, dengan nilai rata-rata sebesar 4,2.

Kemudian pertanyaan terakhir yaitu *“Would you download Healthylicious?”* dan disediakan dua opsi yaitu *“Yes”* dan *“No”*. Kemudian berikut adalah hasil dari survei.



4.2.1.17 Diagram Hasil Responsi 17

Dari data yang diambil, 49 orang menjawab *“Yes”* dan 3 menjawab *“No”*. Hasil responsi ini menggambarkan bahwa mayoritas responden berjumlah 49 tertarik untuk *men-download* aplikasi ini.

4.2.2 Tim Machine Learning

Projek Healthylicious adalah upaya kolaboratif dari tim machine learning yang bertujuan untuk menciptakan aplikasi yang dapat memberikan rekomendasi resep berdasarkan bahan-bahan yang dimiliki pengguna dan preferensi pribadi mereka. Proses pengembangan projek ini melalui beberapa tahap, mulai dari pengumpulan data, praproses dan pembersihan data, pemodelan dan evaluasi, hingga deployment. Berikut ini adalah penjelasan mendetail tentang setiap tahap yang dilakukan oleh tim machine learning dalam pengembangan projek Healthylicious.

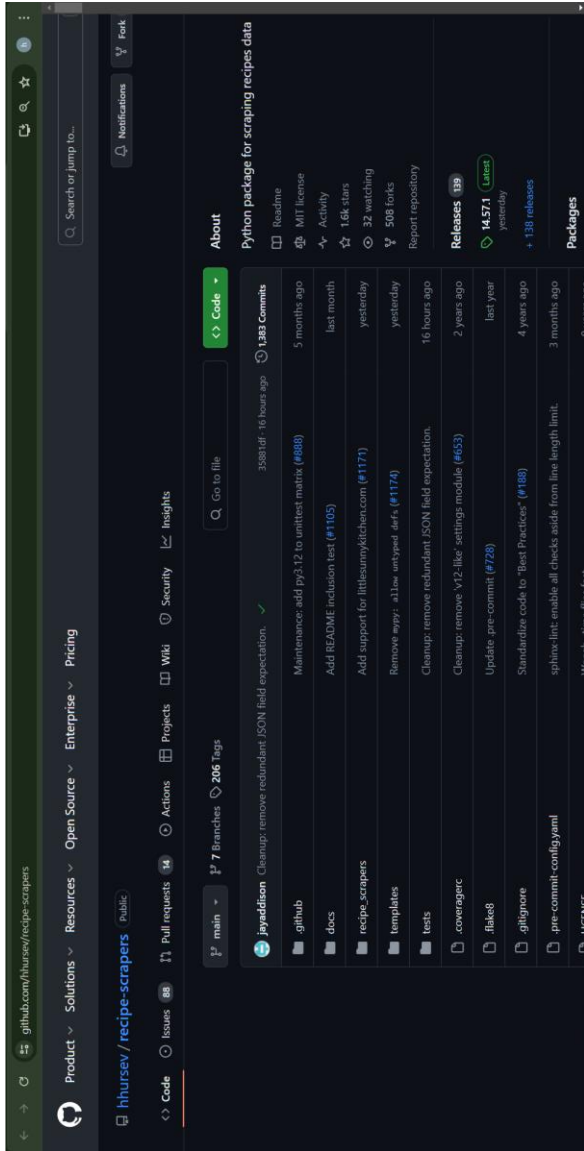
4.2.2.1 Data Gathering

Tahap pertama dalam pengembangan projek Healthylicious adalah pengumpulan data melalui web scraping. Tim menggunakan pustaka `recipe-scrapers` untuk mengumpulkan data resep dari berbagai situs web. Fungsi `scrape_recipe` yang dibuat bertujuan untuk mengambil berbagai atribut resep seperti kategori, judul, waktu total, bahan-bahan, kelompok bahan, instruksi, nutrisi, hasil, deskripsi, gambar, dan penilaian. Proses ini dilengkapi dengan mekanisme penanganan kesalahan yang memastikan data yang diambil lengkap dan akurat. Setelah data terkumpul, tim melakukan pemeriksaan untuk duplikasi URL guna memastikan integritas data yang diambil. Data yang berhasil dikumpulkan kemudian disimpan dalam format `DataFrame` menggunakan pustaka `Pandas` untuk memudahkan langkah berikutnya dalam proses pengembangan.

Gambar-gambar berikut memberikan ilustrasi lebih lanjut tentang proses pengumpulan data:

Gambar 4.2.2.1 berikut menunjukkan pustaka yang

kami gunakan dalam scraping data, berasal dariGitHub: “<https://github.com/hhursev/recipe-scrapers>”



Gambar 4.2.2.1 Pustaka recipe-scrappers

Berikut menunjukkan cuplikan kode dari fungsi `scrape_recipe`, yang digunakan untuk mengambil berbagai atribut resep secara otomatis dari situs web.

```
# Function to scrape data from a single URL  
using recipe-scrapers  
  
def scrape_recipe(url):  
    scraper = scrape_me(url)  
  
    def safe_scrape(scrape_func, default='N/A'):  
        try:  
            result = scrape_func()  
            if result is None:  
                raise  
                ValueError(f"{scrape_func.__name__} returned  
                None")  
            return result  
        except (SchemaOrgException, ValueError)  
    as e:  
        print(f"Error getting  
{scrape_func.__name__} for {url}: {e}")  
        return default  
  
    category = safe_scrape(scraper.category)  
    title = safe_scrape(scraper.title)  
    total_time = safe_scrape(scraper.total_time)  
    ingredients= ',  
' .join(safe_scrape(scraper.ingredients,  
default=[]))  
    ingredient_groups =  
safe_scrape(scraper.ingredient_groups,  
default=[])  
    ingredient_groups= ', ' .join([str(group) for  
group in ingredient_groups])  
    instructions =  
safe_scrape(scraper.instructions)
```

```

nutrients = safe_scrape(scraper.nutrients)
yields = safe_scrape(scraper.yields)
description =
safe_scrape(scraper.description)
image = safe_scrape(scraper.image)
ratings = safe_scrape(scraper.ratings)
cuisine = safe_scrape(scraper.cuisine)

# Check for missing ingredients
if 'N/A' in ingredient_groups:
    status = 'Incomplete'
else:
    status = 'Complete'

return {
    'Category': category,
    'Title': title,
    'Total Time': total_time,
    'All Ingredients': ingredients,
    'Ingredient Groups': ingredient_groups,
    'Instructions': instructions,
    'Nutrition': nutrients,
    'Cuisine': cuisine,
    'Yields': yields,
    'Image': image,
    'Ratings': ratings,
    'Description': description,
    'Status': status
}

```

Dan berikut memperlihatkan salah satu hasil akhir dari proses scraping dalam bentuk DataFrame json, yang memuat berbagai atribut resep yang berhasil dikumpulkan

```

[
  ...,
  {
    "recipeId": 3,
    "title": "Guacamole Recipe",
    "ingredient_groups": [
      {
        "ingredients": [
          "2 avocados (halved",
          "seeded",
          "and peeled)",
          "1 tablespoon lime juice",
          "1/2 teaspoon sea salt",
          "1 clove garlic (minced)",
          "2 Roma tomatoes (seeded and
diced)",
          "1 onion (diced)",
          "1 tablespoon cilantro",
          "pinch ground cumin (optional)"
        ],
        "purpose": "Main Ingredients"
      }
    ],
    "instructions": "Mash avocado with a fork
or potato masher in a bowl until the consistency
you prefer. Add lime juice and salt and mix well
with fork. Stir in tomatoes, onion, and chopped
cilantro. If using cumin, add at this
point.\r\n\r\nPour into a serving bowl and serve.",
    "instructions_list": [
      "Mash avocado with a fork or potato
masher in a bowl until the consistency you
prefer. Add lime juice and salt and mix well
with fork. Stir in tomatoes, onion, and chopped
cilantro. If using cumin, add at this point.",
      "Pour into a serving bowl and serve."
    ],
    "category": "Appetizer",

```



```

"yields": "6 servings",
"description": "This easy Guacamole
recipe, made with fresh ingredients like
avocados, lime juice, cilantro, onions, and
tomatoes is authentic and delicious! Tasty
appetizer and side dish!",
"total_time": 5,
"nutrients": {
  "carbohydrateContent": "7 g",
  "proteinContent": "2 g",
  "fatContent": "10 g",
  "saturatedFatContent": "1 g",
  "sodiumContent": "200 mg",
  "fiberContent": "5 g",
  "sugarContent": "1 g",
  "calories": "112 kcal",
  "servingSize": "1 serving"
},
"image": "https://addapinch.com/wp-
content/uploads/2013/01/guacamole-
DSC_1685-3.jpg"
},
...
]

```

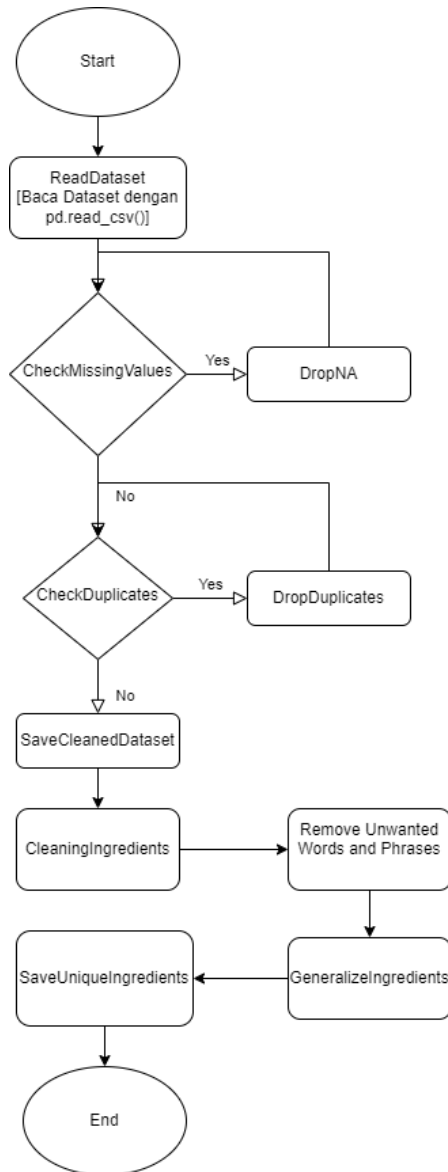
4.2.2.2 Preprocessing and Cleaning Data

Langkah selanjutnya adalah praproses dan pembersihan data. Dataset yang diperoleh dari proses scraping diimpor dan diperiksa untuk nilai yang hilang dan duplikat. Baris yang mengandung nilai yang hilang dihapus untuk memastikan kelengkapan data. Tim juga menghapus duplikat baris berdasarkan kolom judul untuk menghindari data yang redundan. Untuk bahan-bahan dalam resep, dilakukan proses pembersihan lanjutan yang

melibatkan normalisasi kata, penghapusan karakter yang tidak diinginkan, serta penyatuan bentuk jamak menjadi bentuk tunggal. Semua proses ini bertujuan untuk memastikan bahwa data yang akan digunakan untuk pemodelan adalah data yang bersih dan konsisten.

Berikut adalah gambar yang menjelaskan proses ini lebih lanjut:

Gambar 4.2.2.2 menunjukkan diagram alir yang menggambarkan langkah-langkah pra proses dan pembersihan data, termasuk penghapusan nilai yang hilang dan normalisasi bahan-bahan.



Gambar 4.2.2.2 Diagram Alir Cleaning Data

Berikut memperlihatkan kondisi awal DataFrame pada bagian ingredients sebelum dilakukan pembersihan dan generalisasi.

```
{
  ...,
  "ingredient_groups": [
    {
      "ingredients": [
        "2 avocados (halved",
        "seeded",
        "and peeled)",
        "1 tablespoon lime juice",
        "1/2 teaspoon sea salt",
        "1 clove garlic (minced)",
        "2 Roma tomatoes (seeded and
diced)",
        "1 onion (diced)",
        "1 tablespoon cilantro",
        "pinch ground cumin (optional)"
      ],
      "purpose": "Main Ingredients"
    }
  ],
  ...
}
```

Dan berikut memperlihatkan kondisi akhir DataFrame pada bagian ingredients setelah dilakukan pembersihan dan generalisasi.

```
{
  ...,
  "ingredient_groups": [
    {
      "ingredients": [
```

```

        "avocado",
        "lime",
        "sea salt",
        "garlic",
        "roma tomato",
        "onion",
        "cilantro",
        "cumin"
    ],
    "purpose": "Main Ingredients"
},
...
}

```

Maka proses cleaning data telah berhasil menangani data NaN, duplikat, serta menggeneralisir data untuk memudahkan proses modelling.

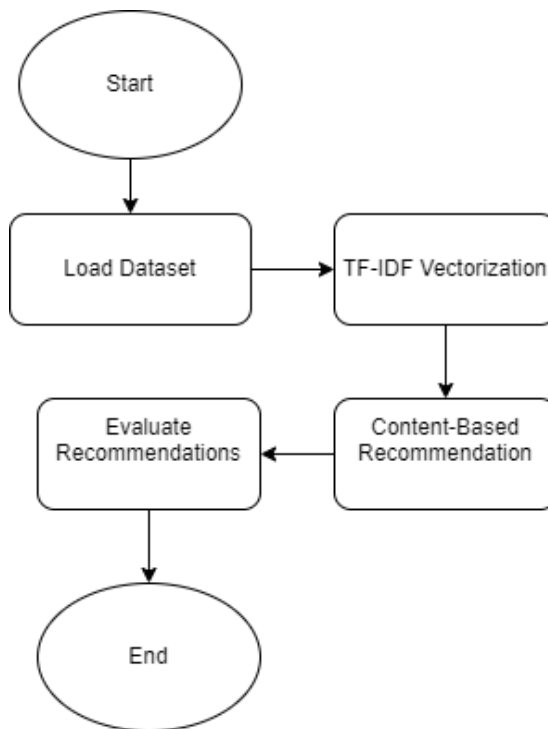
4.2.2.3 Modelling and Evaluation

Tahap pemodelan dimulai dengan membangun dua fitur utama: "Recommended for you" dan "Get Recipes". Untuk fitur "Recommended for you", tim menggunakan algoritma K-Nearest Neighbors (KNN) yang menggabungkan kesamaan bahan, kategori, dan waktu memasak untuk memberikan rekomendasi resep. Algoritma ini diimplementasikan dengan memanfaatkan TF-IDF vectorization dan cosine similarity untuk menghitung skor kesamaan antara resep. Untuk fitur "Get Recipes", model berbasis konten dibangun dengan menggunakan TF-IDF vectorization untuk memberikan rekomendasi berdasarkan bahan yang dimiliki pengguna. Kedua model dievaluasi menggunakan metrik seperti RMSE, MAE, precision, dan recall melalui teknik

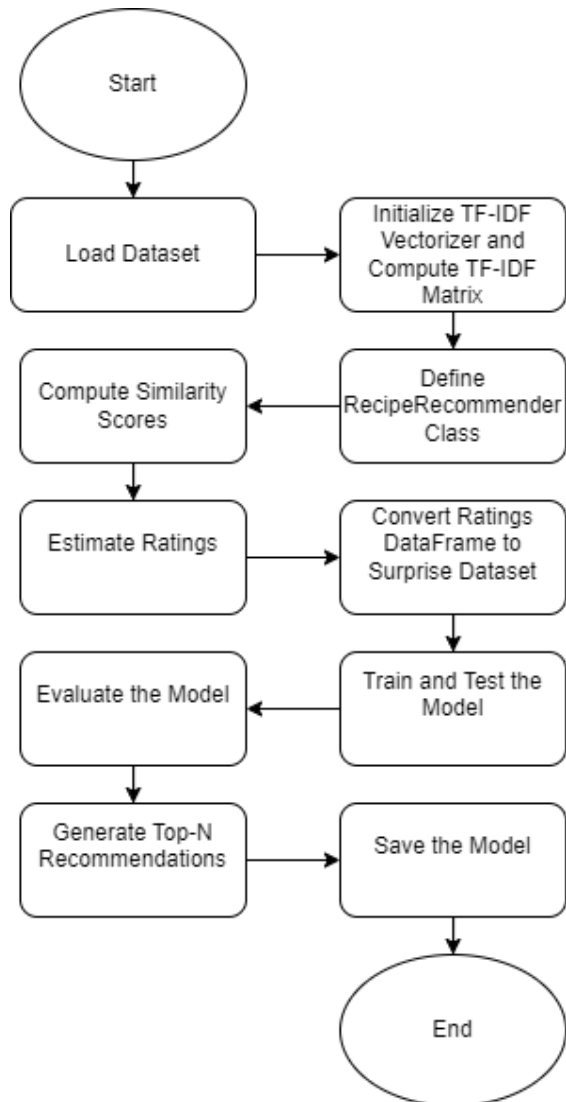
cross-validation. Hasil evaluasi menunjukkan performa model yang baik dan siap untuk tahap deployment.

Gambar berikut memberikan ilustrasi lebih lanjut:

Gambar 4.2.2.3 dan 4.2.2.4 menggambarkan diagram alir dari proses pemodelan dan evaluasi, menunjukkan langkah-langkah dari pembangunan model hingga evaluasi performa.



Gambar 4.2.2.3 Diagram Alir TF-IDF Vectorization dan Cosine Similarity untuk fitur “Get Recipes”



Gambar 4.2.2.4 Diagram Alir KNN untuk Fitur Recommended for You

Berikut merupakan cuplikan class KNN yang digunakan untuk menghasilkan rekomendasi:

```
class RecipeRecommender(AlgoBase):
    def __init__(self, num_neighbors=40,
sim_options={}):
        AlgoBase.__init__(self)
        self.num_neighbors = num_neighbors

    def fit(self, trainset):
        AlgoBase.fit(self, trainset)
        self.recipe_data = {row['recipeId']: row for
_, row in recipe_details_df.iterrows()}
        self.recipe_index_map = {row['recipeId']: i
for i, row in recipe_details_df.iterrows()}

        logging.info("Calculating similarity
matrix...")
        self.similarity_matrix =
np.zeros((self.trainset.n_items,
self.trainset.n_items))
        for i in range(self.trainset.n_items):
            if (i % 50 == 0):
                logging.info(f"Processing {i} of
{self.trainset.n_items}")
                for j in range(i + 1,
self.trainset.n_items):
                    recipe1_id =
int(self.trainset.to_raw_iid(i))
                    recipe2_id =
int(self.trainset.to_raw_iid(j))
                    similarity_score =
self.compute_similarity(recipe1_id, recipe2_id)
                    self.similarity_matrix[i, j] =
similarity_score
                    self.similarity_matrix[j, i] =
similarity_score
                logging.info("---- calculating --- is ---")
```



```

done.")
    return self

    def compute_similarity(self, recipe1_id,
recipe2_id):
        recipe1 = self.recipe_data[recipe1_id]
        recipe2 = self.recipe_data[recipe2_id]
        recipe1_index =
self.recipe_index_map[recipe1_id]
        recipe2_index =
self.recipe_index_map[recipe2_id]
        ingredient_similarity_score =
cosine_similarity(ingredient_tfidf_matrix[recipe
1_index],
ingredient_tfidf_matrix[recipe2_index]).flatten(
)[0]
        category_similarity_score = 1 if
recipe1['Category'] == recipe2['Category'] else 0
        time_difference = abs(recipe1['Total
Time'] - recipe2['Total Time'])
        time_similarity_score = np.exp(-
time_difference / 10.0)
        total_similarity_score = 0.2 *
ingredient_similarity_score + 0.6 *
category_similarity_score + 0.2 *
time_similarity_score
        return total_similarity_score

    def estimate(self, user, item):
        if not (self.trainset.knows_user(user) and
self.trainset.knows_item(item)):
            raise PredictionImpossible('User and/or
item is unknown.')
        neighbors = []
        for rating in self.trainset.ur[user]:
            similarity_score =
self.similarity_matrix[item, rating[0]]
            neighbors.append((similarity_score,

```

```

rating[1]))
    top_k_neighbors =
heapq.nlargest(self.num_neighbors, neighbors,
key=lambda t: t[0])
    if not top_k_neighbors:
        raise PredictionImpossible('No similar
neighbors.')
    sim_total = weighted_sum = 0
    for (sim_score, rating) in top_k_neighbors:
        if sim_score > 0:
            sim_total += sim_score
            weighted_sum += sim_score * rating
        if sim_total == 0:
            raise PredictionImpossible('No similar
neighbors.')
    predicted_rating = weighted_sum /
sim_total
    return predicted_rating

```

dan Gambar 4.2.2.5 berikut menunjukkan hasil rekomendasi dari algoritma KNN yang dihasilkan untuk salah satu user

```

# Display top-10 recommendations for another user
user_id = 2 # Change this to a different user ID
top_n_recommendations = generate_top_n_recommendations(predictions, n=10)
if user_id in top_n_recommendations:
    print(f"Top-10 recommendations for user {user_id}:")
    for recipe_id, estimated_rating in top_n_recommendations[user_id]:
        recipe_title = recipe_details_df[recipe_details_df['recipeid'] == recipe_id]['title'].values[0]
        print(f"{recipe_title} (estimated rating: {estimated_rating})")
else:
    print(f"No recommendations for user {user_id}")

Top-10 recommendations for user 2:
Classic Caesar Dressing Recipe (estimated rating: 3.9913576177216714)
Roasted Sweet Potato Wedges Recipe (estimated rating: 3.6837830326182326)
Skinny No-Bake Chocolate Cookie Recipe (estimated rating: 3.6387948073964926)
Fresh Pineapple Sorbet Recipe (estimated rating: 3.605132845294333)
Chocolate Espresso Ganache Pie Recipe (estimated rating: 3.478857015263494)
Perfect Whipped Cream Recipe (estimated rating: 3.44811303861189)
Lemon Buttermilk Waffles Recipe (estimated rating: 3.4312400119345443)
Cranberry Salsa over Cream Cheese Recipe (estimated rating: 3.3568863750981897)
Peanut Butter Sheet Cake Recipe (estimated rating: 3.309578833390095)
Tofffee Apple Pie Cookies Recipe (estimated rating: 3.3024148959730297)

```

Gambar 4.2.2.5 Hasil Rekomendasi KNN

Kemudian berikut merupakan cuplikan fungsi untuk untuk fitur generate recipes menggunakan TF-IDF Vectorization dan cosine similarity

```
# TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english',
ngram_range=(1, 2))

tfidf_matrix =
tfidf_vectorizer.fit_transform(recipes_df['Ingredients'])

def
get_content_based_recommendations(user_ingredients,
top_n=10):

    user_tfidf =
tfidf_vectorizer.transform([user_ingredients])
    cosine_sim = cosine_similarity(user_tfidf,
tfidf_matrix)
    similar_indices = cosine_sim.argsort().flatten()[::-
1][:top_n]
    return recipes_df.iloc[similar_indices]
```

Terakhir Gambar 4.2.2.6 menunjukkan hasil dari generate recipes berdasarkan bahan-bahan yang dimiliki pengguna menggunakan TF-IDF Vectorization dan cosine similarity

```
# Example usage
user_ingredients_str = "cranberry, orange, butter, sugar"
content_based_recommendations = get_content_based_recommendations(user_ingredients_str)
print(content_based_recommendations)

✓ 0.0s
79 Warm Brie with Honeyed Cranberry Walnut Fruit ...
246 Cranberry Orange Muffins Recipe
936 Cranberry Orange Glazed Ham Recipe
1016 Cranberry Sauce Recipe
49 Sugared Cranberries Recipe
61 Cranberry Orange Rolls Recipe
9 Cranberry Salsa over Cream Cheese Recipe
71 Orange Cranberry Salsa Recipe
490 Heritage Frosting Recipe
653 Citrus Sorbet with Meyer Lemons
```

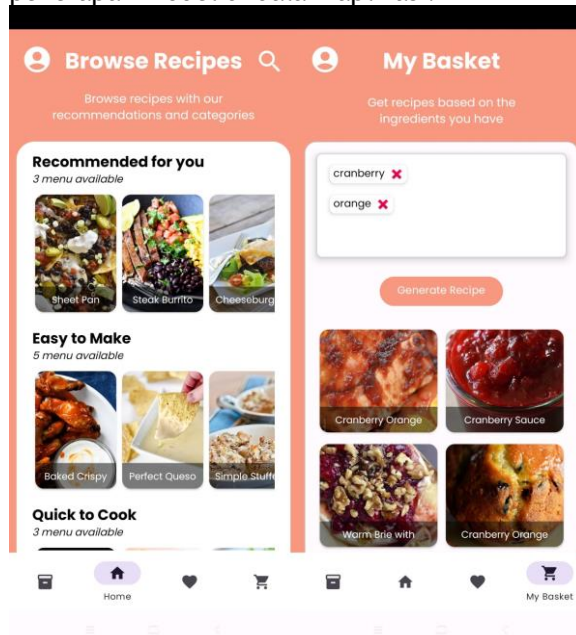
Gambar 4.2.2.6 Hasil Generate Recipes
Dengan demikian model dinyatakan siap untuk

proses deployment.

4.2.2.4 Deployment

Setelah model dievaluasi dan dinyatakan siap, tahap terakhir adalah deployment. Model ini kemudian diintegrasikan langsung ke dalam arsitektur cloud yang kemudian dapat dipanggil oleh aplikasi lewat API. Dengan demikian, tim berhasil menciptakan solusi yang memanfaatkan teknologi machine learning untuk mendukung gaya hidup sehat melalui rekomendasi resep yang dipersonalisasi.

Berikut merupakan gambar yang menunjukkan penerapan model di dalam aplikasi:



Gambar 4.2.2.7 Penerapan Model dalam Aplikasi

4.2.2.5 Analisis Performa Model

Model telah dievaluasi menggunakan dua metrik utama yaitu Root Mean Squared Error (RMSE) dan Mean Absolute Error (MAE). Hasil evaluasi menunjukkan bahwa model memiliki nilai RMSE sebesar 1.020 dan nilai MAE sebesar 0.795.

- RMSE (Root Mean Squared Error): Metrik ini mengukur rata-rata kesalahan antara nilai prediksi dan nilai aktual, dengan memperhatikan besarnya kesalahan. Nilai RMSE yang lebih rendah menunjukkan bahwa model mampu membuat prediksi yang lebih dekat dengan nilai aktual. Dalam konteks ini, RMSE 1.02 menunjukkan bahwa prediksi model rata-rata memiliki kesalahan sekitar satu unit penilaian
- MAE (Mean Absolute Error): Metrik ini mengukur rata-rata kesalahan absolut antara nilai prediksi dan nilai aktual. MAE memberikan gambaran seberapa besar rata-rata kesalahan prediksi tanpa memperhatikan arah kesalahan (positif atau negatif). Nilai MAE sebesar 0.795 menunjukkan bahwa, rata-rata, prediksi model memiliki kesalahan absolut kurang dari satu unit penilaian.

Pada sistem rekomendasi, metrik akurasi tidak cocok digunakan untuk mengukur performa model. Hal ini dikarenakan:

- Data Imbalance: Pada umumnya, dataset dalam sistem rekomendasi memiliki ketidakseimbangan data yang tinggi, dimana hanya sebagian kecil item yang relevan bagi pengguna tertentu. Metrik akurasi dapat memberikan gambaran yang menyesatkan karena dapat menghasilkan nilai tinggi hanya dengan memprediksi semua item sebagai tidak relevan.

- Konteks Relevansi: Dalam sistem rekomendasi, yang lebih penting adalah seberapa baik model dapat merekomendasikan item yang relevan (precision) dan seberapa banyak item relevan yang dapat ditemukan (recall), bukan hanya seberapa banyak prediksi yang benar secara keseluruhan.
- Nilai Prediksi yang Beragam: Sistem rekomendasi sering kali berurusan dengan nilai prediksi yang beragam (misalnya, penilaian dari 1 hingga 5 bintang). Akurasi tidak mempertimbangkan seberapa jauh nilai prediksi dari nilai sebenarnya, sementara metrik seperti RMSE dan MAE memberikan informasi tersebut.

4.2.3 Tim Cloud Computing

Tim cloud computing berperan untuk membuat API yang diperlukan aplikasi serta mengimplementasikannya dengan menggunakan layanan Google Cloud Platform. Secara umum, pembuatan API dibagi menjadi tiga bagian yaitu bagian Authentication, User Profile, dan Features.

4.2.3.1 API Authentication

API Authentication berhubungan dengan autentikasi pengguna dalam menggunakan aplikasi Healthylicious. Dengan API ini, pengguna dapat melakukan registrasi akun, masuk ke aplikasi, dan melakukan perubahan kata sandi jika diperlukan. Berikut adalah endpoint yang dibuat untuk Authentication dari Healthylicious:

Endpoint	Method	Keterangan
/register	POST	Membuat akun Healthylicious

/login	POST	Masuk ke aplikasi Healthylicious
/login/google	POST	Masuk menggunakan Google
/request-reset-password	POST	Reset kata sandi

Tabel 4.2.3.1 API Authentication

4.2.3.2 API User Profile

API User Profile berhubungan dengan profil dan preferensi dari pengguna aplikasi Healthylicious. Profil pengguna ini berisi data username, usia, berat badan, dan tinggi badan yang nantinya dapat digunakan untuk perhitungan kalori yang dibutuhkan. Sedangkan preferensi dalam User Profile dibuat agar pengguna dapat menghindari bahan makanan yang mengandung alergi dan juga bahan makanan yang tidak disukai. Berikut adalah endpoint dari User Profile:

Endpoint	Method	Keterangan
/user/profile	POST	Menambahkan detail user profile
/user/profile	PUT	Memperbaharui user profile
/user/profile/{uid}	GET	Melihat detail user profile
/user/dislikes	POST	Menambahkan

		bahan makanan yang tidak disukai atau alergi
/user/dislikes	GET	Melihat bahan makanan yang tidak disukai atau alergi

Tabel 4.2.3.2 API User Profile

4.2.3.3 API Features

API Features berisi semua fitur utama yang ada di aplikasi Healthylicious yang mencakup *ingredients*, *recipes*, *recommendations*, dan *recipe generation*. API ini akan menampilkan pemilihan bahan, resep yang tersedia, resep rekomendasi, dan juga fitur untuk melakukan rekomendasi resep berdasarkan bahan yang diinputkan. Berikut adalah endpoint dari Features:

Endpoint	Method	Keterangan
/ingredients	GET	Menampilkan semua <i>ingredients</i> yang dikelompokkan berdasar kategori
/ingredients/search	GET	Pencarian <i>ingredients</i>
/recipes	GET	Menampilkan semua judul dan gambar dari resep
/recipes/search	GET	Pencarian resep

/recipes/{title}	GET	Menampilkan resep secara detail
/recommendations	GET	Menampilkan rekomendasi resep berdasarkan kategori : Easy to make, Quick to Cook, Breakfast, Dessert, dan Snack
/rate	POST	Memberikan rating pada resep
/rate/recommendations	GET	Menampilkan rekomendasi resep berdasarkan rating
/rating	GET	Menampilkan rating pada setiap resep
/generate	POST	Melakukan <i>generate</i> resep dari input <i>ingredient</i>

Tabel 4.2.3.3 API Features

4.2.3.4 Dokumentasi API

Dokumentasi API digunakan untuk mengetahui permintaan yang diberikan pada setiap endpoint. Berikut adalah dokumentasi dari API Healthylicious

POST /register

- Method: POST
- Endpoint: /register
- Body parameters:

```
{  
    "email": "<your_email>",  
    "password": "<your_password>"  
}
```

POST /login

- Method: POST
- Endpoint: /login
- Body parameters:

```
{  
    "email": "<your_email>",  
    "password": "<your_password>"  
}
```

POST /request-reset-password

- Method: POST
- Endpoint: /request-reset-password
- Body parameters:

```
{  
    "email": "<your_email>",  
}
```

POST /user/profile

- Method: POST
- Endpoint: /user/profile
- Header : Authorization: Bearer "token"
- Body parameters:

```
{  
    "username": "Name",  
}
```

```
"age": 21,  
"weight": 50,  
"height": 175  
}
```

PUT /user/profile

- Method: PUT
- Endpoint: /user/profile
- Header : Authorization: Bearer "token"
- Body parameters:

```
{  
  "username": "New Name",  
  "age": 21,  
  "weight": 50,  
  "height": 175  
}
```

GET /user/profile

- Method: GET
- Endpoint: /user/profile{uid}
- Header : Authorization: Bearer "token"
- Body parameters: -

POST /user/dislikes

- Method: POST
- Endpoint: /user/dislikes
- Header : Authorization: Bearer "token"
- Body parameters:

```
["beef","corn","bacon","potato,crab"]
```

GET /user/dislikes

- Method: GET
- Endpoint: /user/dislikes
- Header : Authorization: Bearer "token"
- Body parameters: -

GET /ingredients

- Method: GET
- Endpoint: /ingredients
- Body parameters: -

GET /ingredients/search

- Method: GET
- Endpoint: /ingredients/search
- Body parameters: -- Key : query -- Value : chili
(your ingredient search)

GET /recipes

- Method: GET
- Endpoint: /ingredients
- Body parameters: -

GET /recipes/search

- Method: GET
- Endpoint: /ingredients/search
- Body parameters: -- Key : query -- Value : chili
(your recipes key search)

GET /recipes/search

- Method: GET
- Endpoint: /ingredients/{title}
- Body parameters: -

GET /recommendations

- Method: GET
- Endpoint: /recommendations
- Body parameters: -

POST /rate

- Method: POST
- Endpoint: /rate
- Header : Authorization: Bearer "token"
- Body parameters:

```
{
  "recipeld": "15",
  "rating": 3
}
```

GET /rate/recommendations

- Method: GET
- Endpoint: /rate/recommendations
- Header : Authorization: Bearer "token"
- Body parameters: -

GET /rating

- Method: GET
- Endpoint: /rating
- Body parameters: -

POST /generate

- Method: POST
- Endpoint: /generate
- Header : Authorization: Bearer "token"
- Body parameters:

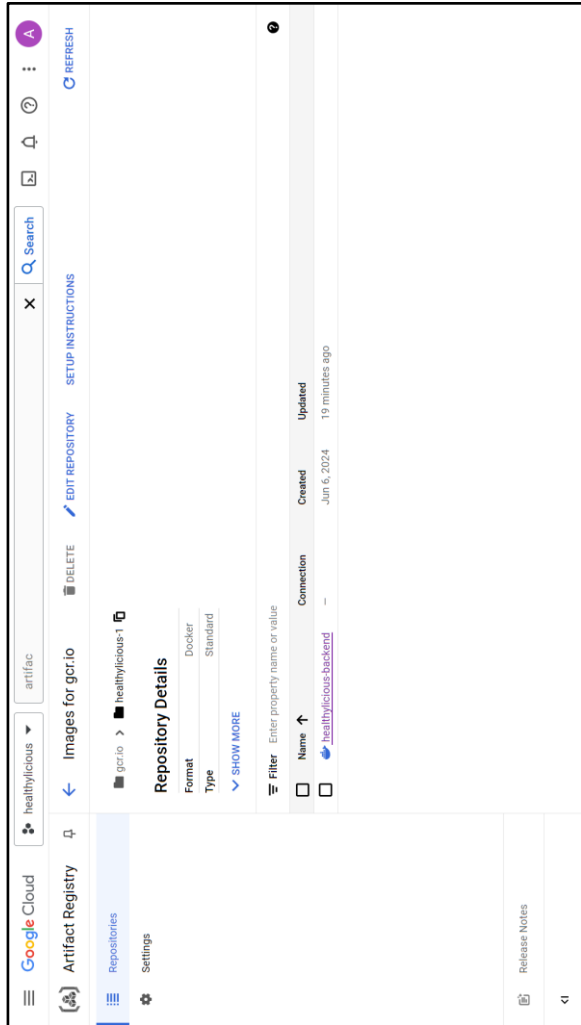
```
{
  "ingredients": "beef, butter, salt"
}
```

4.2.3.5 Kontainerisasi Dockerfile

Sebelum melakukan deploy backend, kode yang sudah dibuat di kontainerisasi ke Google Artifact Registry dengan perintah sebagai berikut:

```
gcloud builds submit --tag gcr.io
/healthylicious-1/healthylicious-backend .
```

Perintah tersebut memanfaatkan Google Cloud Build untuk membangun (*build*) aplikasi backend yang sudah dibuat. *Docker image* yang sudah dibangun akan disimpan dalam gcr.io di Artifact Registry dengan nama healthylicious-backend.



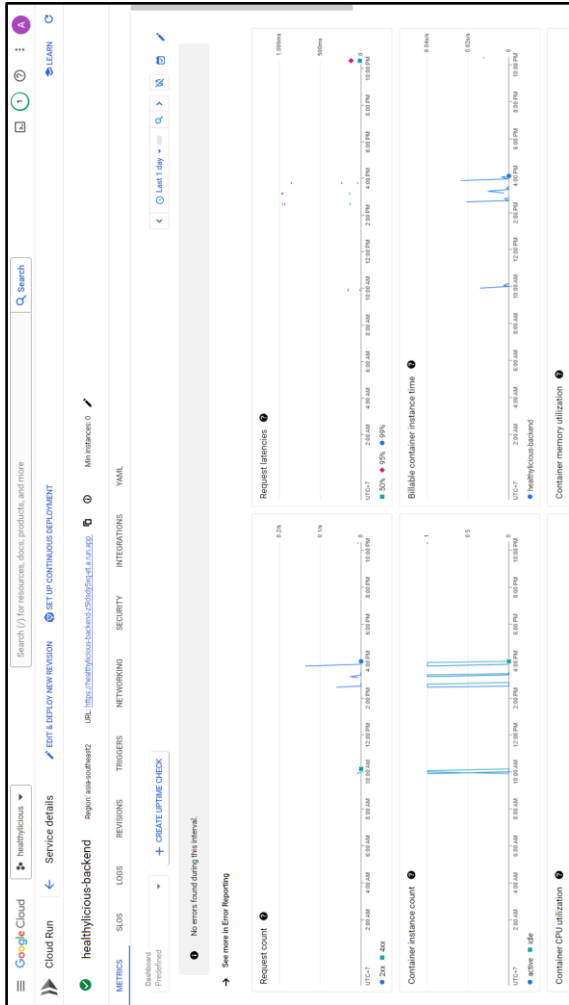
Gambar 4.2.3.1 Layanan Artifact Registry

4.2.3.6 Deploy dengan Cloud Run

Cloud Run menjalankan aplikasi berbasis kontainer dalam lingkungan *serverless*. Untuk melakukan *deploying* docker image healthylicious-backend, maka digunakan perintah berikut:

```
gcloud run deploy healthylicious-backend
--image gcr.io /healthylicious -1 /
healthylicious-backend
--platform managed
--region asia-southeast2
--allow-unauthenticated
```

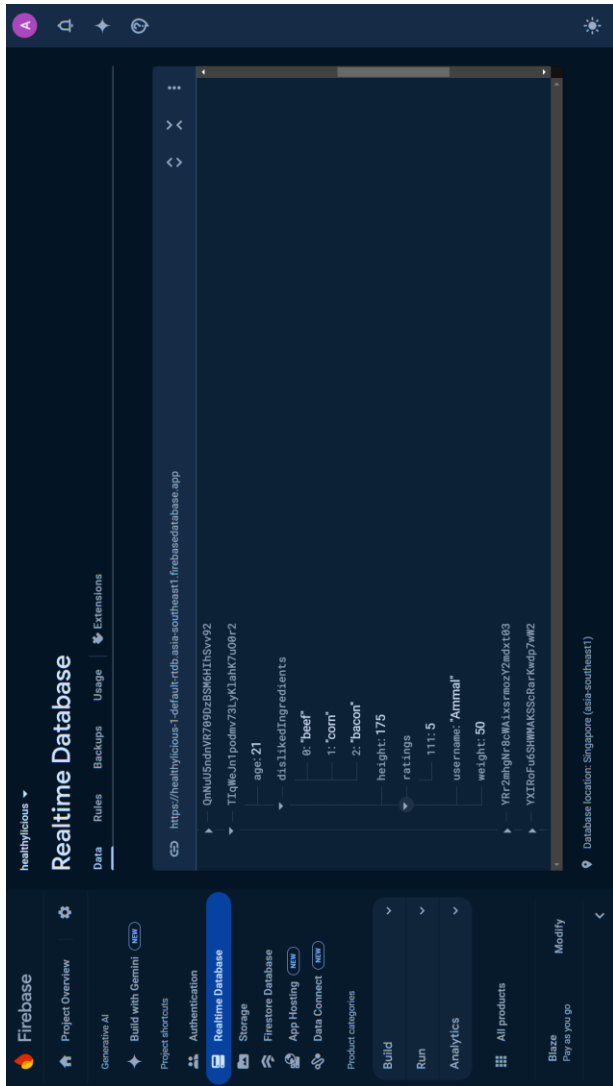
Perintah tersebut akan melakukan *deploy* aplikasi backend ke Cloud Run dengan nama layanan healthylicious-backend. Image yang akan di *deploy* berada pada direktori **gcr.io/healthylicious-1/healthylicious-backend**. Sesuai dengan karakteristik dari Cloud Run yaitu *serverless*, perintah **platform managed** menunjukkan bahwa layanan akan berjalan di platform Cloud Run yang dikelola sepenuhnya oleh Google yang dengan mengelola infrastruktur dan skala layanan secara otomatis. Wilayah tempat layanan akan di *deploy* adalah **asia-southeast 2** untuk mengurangi latensi bagi pengguna di wilayah tersebut. Kemudian aplikasi backend yang sudah di *deploy* dapat diakses secara langsung dengan adanya perintah **allow-unauthenticated**.



Gambar 4.2.3.2 Layanan Cloud Run

4.2.3.7 Penyimpanan Data di Firebase

Data pengguna Healthylicious termasuk data autentikasi, data profil pengguna, preferensi pengguna, dan data rating disimpan dalam Firebase. Firebase Authentication mengelola data pengguna yang telah melakukan registrasi dan login dengan menyimpan email yang digunakan dan UID yang dimiliki setiap pengguna. Untuk profil pengguna, preferensi, dan rating yang diberikan oleh pengguna, data tersebut disimpan dalam Firebase Realtime Database untuk melakukan sinkronisasi data antara pengguna secara *real-time*. Data disimpan dalam format JSON yang memudahkan strukturisasi data.

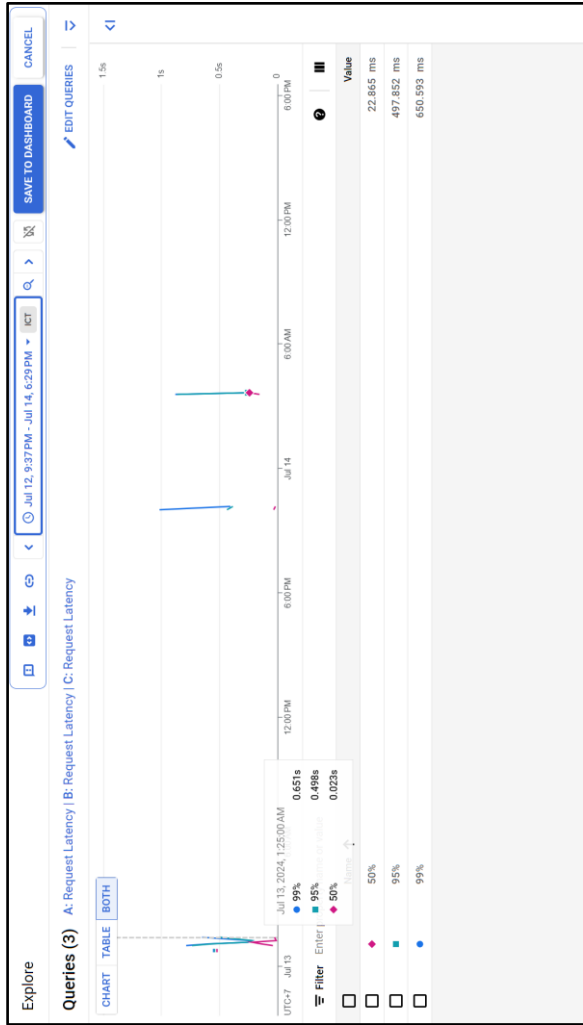


Gambar 4.2.3.3 Layanan Firebase

4.2.3.8 Analisis Request Latency

Request Latency dari *Backend* aplikasi Healthylicious dapat dianalisis melalui tiga pendekatan yaitu:

- 50th percentile : Merupakan angka median yang memberikan gambaran umum tentang kinerja aplikasi secara keseluruhan. Namun tidak mencakup permintaan yang mungkin memiliki latensi yang jauh lebih tinggi dari median.
- 95th percentile : Memberikan gambaran tentang bagaimana aplikasi menangani sebagian besar beban kerja, termasuk beban yang lebih tinggi daripada rata-rata.
- 99th percentile : Memberikan gambaran tentang skenario terburuk yang dialami oleh pengguna.



Gambar 4.2.3.4 Data Request Latency Cloud Run

Data *Request Latency* yang didapat ketika user menggunakan aplikasi Healthylicious adalah sebagai berikut:

- 50th percentile : 22.865 ms
Berarti bahwa 50% dari semua permintaan yang dilakukan dalam aplikasi Healthylicious diproses dalam waktu 22.865 ms atau kurang. Hal ini menunjukkan bahwa setengah dari permintaan memiliki latensi yang sangat rendah dan responsif.
- 95th percentile : 497.852 ms
Berarti bahwa 95% dari semua permintaan yang dilakukan dalam aplikasi Healthylicious diproses dalam waktu 497.852 ms atau kurang. Hanya 5% dari permintaan yang memiliki latensi lebih tinggi dari 497.852 ms.
Terlihat bahwa terdapat peningkatan yang signifikan dari latensi aplikasi Healthylicious untuk sebagian kecil permintaan.
- 99th percentile : 650.593 ms
Berarti bahwa 99% dari semua permintaan diproses dalam waktu 650.593 ms atau kurang. Hanya 1% dari permintaan yang memiliki latensi lebih tinggi dari 650.593 ms.
Peningkatan latensi yang lebih tinggi menunjukkan bahwa dalam kondisi yang sangat jarang terjadi, latensi bisa meningkat hingga 650ms.

Untuk mengetahui penyebab lebih detail mengenai peningkatan latensi dari setiap permintaan yang dilakukan dalam aplikasi Healthylicious, maka dapat dilakukan pengamatan pada Google Cloud Trace

The screenshot displays the Google Cloud Trace interface for the project 'healthycious'. The main content area shows a list of traces with the following columns: Latency, HTTP Method, and URL. The selected trace has a latency of 835 ms and is for the URL '/rate'.

Latency	HTTP Method	URL	Time
835 ms	GET	/rate	3 minutes a...
14 ms	POST	/recommen...	4 minutes a...
153 ms	GET	/recipes/S...	12 minutes
11 ms	GET	/rate/recom...	12 minutes
227 ms	GET	/rating	13 minutes

The 'Trace details' section shows the following information:

- Trace ID: 7d61bc8e4ef9920e83f6451bc6089
- Start time (Asia/Jakarta): JM 13 at 1:21:39:995 AM
- Duration: 835ms
- Name: Collapses all
- Service: Service
- Trace ID: 7d61bc8e4ef9920e83f6451bc6089
- Path: /login

Gambar 4.2.3.5 Data Cloud Trace

Dari data tersebut terlihat bahwa untuk melakukan permintaan pada endpoint yang tidak memerlukan model machine learning, permintaan dapat diproses dengan latensi yang rendah dibawah 100 ms. Sedangkan untuk permintaan pada endpoint yang memerlukan model machine learning seperti generate dan recommendations, permintaan akan diproses dengan latensi antara 100 - 200 ms. Latensi tertinggi terjadi ketika user melakukan permintaan yang berhubungan dengan autentikasi seperti login, yang mana latensi mencapai angka 835 ms

BAB V PENUTUP

5.1 KESIMPULAN

5.1.1 KESIMPULAN TEKNIS

Berdasarkan aplikasi Healthylicious yang telah dikembangkan, dapat disimpulkan beberapa hal sebagai berikut untuk masing-masing divisi :

1. **Mobile Development:** *User Interface* sudah berjalan dengan baik secara fungsional dan desain. Namun masih ada aspek yang bisa dikembangkan lebih lanjut seperti penambahan fitur dan memperbaiki desain agar lebih interaktif.
2. **Machine Learning:** Model menunjukkan performa yang cukup baik dengan nilai RMSE dan MAE yang relatif rendah, menunjukkan bahwa prediksi model cukup akurat.
3. **Cloud Computing:** Berdasarkan analisis data dari Cloud Run dan Cloud Trace, terdapat adanya variasi yang signifikan dalam waktu respons (*latency*). Secara umum Healthylicious dapat dikatakan bekerja cukup responsif dalam keadaan normal. Namun peningkatan latensi yang signifikan terjadi dalam situasi tertentu yang mengindikasikan bahwa ada sebagian kecil permintaan yang mengalami lonjakan beban yang tidak terduga dan tidak optimal dalam distribusi beban.

5.1.2 KESIMPULAN UMUM

Berdasarkan kegiatan Magang yang telah dilaksanakan, dapat disimpulkan beberapa hal berikut :

1. Dalam jangka waktu satu bulan, tim Healthylicious telah berhasil mengembangkan sebuah platform yang memanfaatkan teknologi machine learning dan cloud computing, serta mengintegrasikan praktik terbaik dalam pengembangan aplikasi mobile. Aplikasi Healthylicious menawarkan solusi inovatif untuk membantu pengguna dalam memilih resep makanan sehat berdasarkan bahan yang tersedia di rumah, serta memperhitungkan preferensi dan alergi pengguna.
2. Aplikasi Healthylicious, yang dikembangkan menggunakan Android Studio, Kotlin, JavaScript, Python, dan berbagai pustaka machine learning seperti scikit-learn menawarkan fitur-fitur seperti rekomendasi resep berdasarkan bahan yang tersedia, informasi nutrisi, dan penilaian pengguna.
3. Potensi pengembangan aplikasi Healthylicious sangat luas. Aplikasi ini dapat dikembangkan lebih lanjut untuk mencakup lebih banyak resep dan variasi bahan, meningkatkan akurasi rekomendasi dengan dataset yang lebih besar dan lebih beragam.

5.2 SARAN

5.2.1 SARAN TEKNIS

Berdasarkan analisis yang telah dilakukan untuk aplikasi Healthylicious, terdapat beberapa saran yaitu :

1. Mobile Development: Untuk meningkatkan *user interface* dan *user experience* maka

dapat dilakukan langkah pengembangan sebagai berikut:

- a. Peningkatan kualitas *user interface*.
- b. Melakukan eksperimen untuk mencari *user interface* yang paling cocok untuk tema aplikasi.
- c. Melakukan penambahan fitur terhadap aplikasi agar aplikasi memiliki nilai jual yang lebih tinggi.
- d. Melakukan peningkatan kualitas *user experience* pada tampilan supaya aplikasi berjalan lebih lancar dan mencegah terjadinya crash.

2. Machine Learning: Untuk meningkatkan performa model dapat dilakukan dengan beberapa langkah pengembangan sebagai berikut.

- a. Hyperparameter tuning lebih lanjut, atau mencoba algoritma alternatif seperti SVD atau deep learning-based recommender systems.
- b. Menggunakan teknik feature engineering yang lebih canggih untuk menangkap lebih banyak informasi dari data resep dan pengguna, misalnya dengan menggunakan embedding dari neural network.
- c. Menambah dataset untuk dapat membantu model mengenali pola yang lebih kompleks dan peluang yang lebih baik untuk memahami preferensi pengguna dan menghasilkan rekomendasi yang lebih akurat.

3. Cloud Computing: Untuk mengatasi variasi kenaikan latensi yang signifikan dan memastikan kinerja aplikasi yang lebih

konsisten, dapat dilakukan dengan menerapkan beberapa langkah pengembangan yaitu:

- a. Konfigurasi pengelolaan beban dengan load balancer untuk menentukan distribusi beban yang merata dan menghindari distribusi beban yang merata dan menghindari overload.
- b. Konfigurasi sistem peringatan (alert) untuk mengetahui jika latensi mendekati atau melebihi ambang batas yang dapat diterima.
- c. Pengujian lebih lanjut dengan pengujian beban (load testing) untuk mensimulasikan kondisi yang menyebabkan peningkatan latensi dan pengujian stress (stress testing) untuk memahami batas maksimal aplikasi dan mengetahui respon aplikasi jika berada di bawah beban ekstrim.

5.2.2 SARAN UMUM

Berdasarkan kegiatan Magang yang telah dilaksanakan, terdapat beberapa saran yaitu :

1. Untuk Bangkit: Diharapkan di tahun mendatang, Tim Bangkit dapat meningkatkan pemerataan anggota kelompok untuk Project Capstone agar lebih optimal. Hal ini penting untuk memastikan bahwa setiap anggota tim memiliki peran yang signifikan dan berkontribusi secara maksimal dalam proyek.
2. Untuk pelajar yang akan melakukan Magang: Mempelajari berbagai tools dan framework yang relevan dengan materi Capstone Bangkit sebelum mengikuti program akan sangat membantu dalam proses pengerjaan magang. Selain itu, manajemen waktu yang

baik sangat penting untuk memastikan kelancaran proses magang dan pencapaian hasil yang optimal. Disarankan juga untuk aktif berpartisipasi dalam diskusi dan sesi mentoring yang disediakan oleh program untuk mendapatkan wawasan dan pengetahuan yang lebih mendalam.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

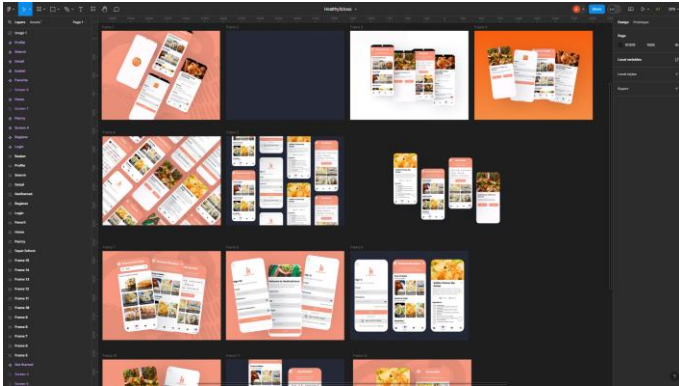
- [1] Bini, S. A. (2018). Artificial Intelligence, Machine Learning, Deep Learning, and Cognitive Computing: What Do These Terms Mean and How Will They Impact Health Care? *J Arthroplasty*, 33(8), 2358-2361. doi: 10.1016/j.arth.2018.02.067. Epub 2018 Feb 27. PMID: 29656964.
- [2] Bisong, E. (2019). Introduction to Scikit-learn. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_18
- [3] Bernard, J. (2016). Python Data Analysis with pandas. In *Python Recipes Handbook*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-0241-8_5
- [4] Felker, D. (2018). *Android Studio 3.0 Development Essentials - Android 8 Edition*. Payload Media.
- [5] Fielding, R. T. (2000). Chapter 5: Representational State Transfer (REST). *Architectural Styles and the Design of Network-based Software Architectures* (Ph.D.). University of California, Irvine.
- [6] Google. (n.d). Google Cloud Products. Retrieved from <https://cloud.google.com/products/>. Accessed 28 June 2024.
- [7] Google. (n.d). Cloud Run Documentation. Retrieved from <https://cloud.google.com/run/docs>. Accessed 27 June 2024.
- [8] Google Firebase. (n.d). Firebase Documentation. Retrieved from <https://firebase.google.com/docs>. Accessed 28 June 2024.
- [9] Hapi Dev. (2024). Hapi.js Documentation. Retrieved from <https://hapi.dev>. Accessed 28 June 2024.
- [10] Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. *Nature*, 585, 357-362. <https://doi.org/10.1038/s41586-020-2649-2>
- [11] Helm, J. M., Swiergosz, A. M., Haeberle, H. S., et al. (2020). *Machine Learning and Artificial Intelligence:*

- Definitions, Applications, and Future Directions. *Curr Rev Musculoskelet Med*, 13, 69-76.
<https://doi.org/10.1007/s12178-020-09600-8>
- [12] Ko, H., Lee, S., Park, Y., & Choi, A. (2022). A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*, 11(1), 141.
<https://doi.org/10.3390/electronics11010141>
- [13] Lagomarsino, J. (2018). *MVVM in Android: Clean Architecture for Elegant and Modern Android Apps*. Leanpub.
- [14] Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing (Technical report)*. National Institute of Standards and Technology: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145.
- [15] Nawrocka, A., Kot, A., & Nawrocki, M. (2018). Application of machine learning in recommendation systems. In *2018 19th International Carpathian Control Conference (ICCC)*, Szilvasvarad, Hungary, pp. 328-331. doi: 10.1109/CarpathianCC.2018.8399650.
- [16] Phillips, B., Stewart, C., Hardy, K., & Marsicano, M. (2018). *Android Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch.
- [17] Prasetyo, D., dkk. (2024). "Cloud Computing" Penamuda Media.
- [18] Samuel, B. (2019). *Kotlin Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch.
- [19] Srinath, K. R. (2017). Python-the fastest growing programming language. *International Research Journal of Engineering and Technology*, 4(12), 354-357.
- [20] Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Ann Transl Med*, 4(11), 218. doi: 10.21037/atm.2016.03.37. PMID: 27386492; PMCID: PMC4916348.

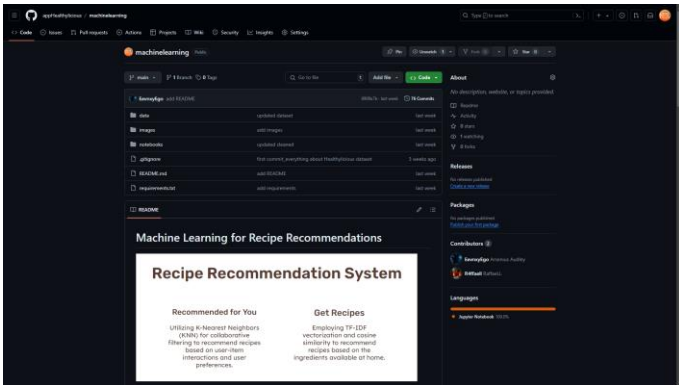
(Halaman ini sengaja dikosongkan)

LAMPIRAN

1. Figma :



2. Repositori untuk *Machine Learning*



Datasets

- `ingredients`: Contains recipe information including ingredients, instructions, and nutritional data.
- `preferences`: Contains user ratings for different recipes.

Machine Learning Models

Recommended for You

Utilizes `K-Nearest Neighbors (KNN)` for collaborative filtering to recommend recipes based on user item interactions and user preferences.

- User Interactions:**
 - Users choose disliked ingredients and allergies, which are stored in `Preferences`.
 - Users rate recipes, and these ratings are also stored in `Preferences`.
- Collaborative Filtering:**
 - Calculates the similarity between the data points (recipes) and all training data.
 - Recommends recipes with the highest similarity while considering the user's dislikes and preferences.

Project Structure

- `data/`: Contains datasets used for training and evaluation.
- `models/`: Jupyter notebooks with exploratory data analysis and model development.
- `utils/`: Specific files to be ignored by git.

Getting Started

Prerequisites

- Python 3.8+
- Jupyter Notebook
- Required Python packages listed in `requirements.txt`

Installation

- Clone the repository:


```
git clone https://github.com/realbadcat/recipe-recommender-gpt
```
- Clone the repository:


```
pip install -r requirements.txt
```
- Run the Jupyter Notebook:


```
jupyter notebook
```

Usage

- Open the notebooks in the `notebooks/` directory to explore the data and model training process.
- Modify and run the notebooks to train the model and generate recommendations based on your own input data.

Datasets

Generate Recipe

Get Recipes

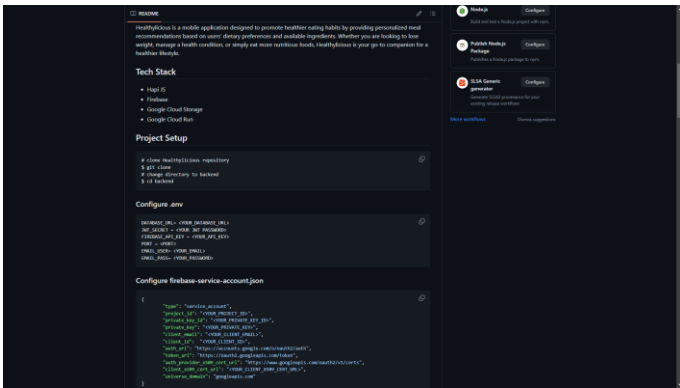
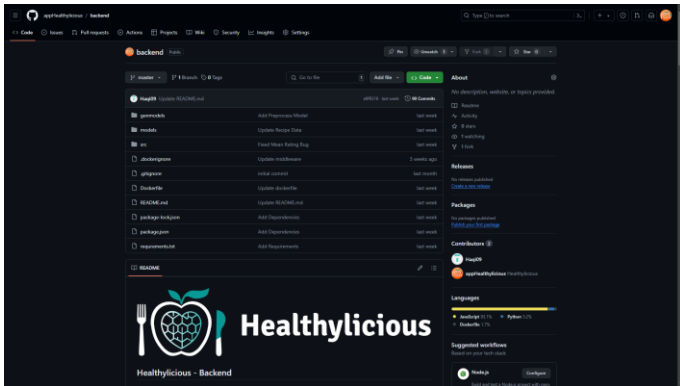
Employs TF-IDF vectorization and cosine similarity to recommend recipes based on the ingredients available at home.

- User Input:**
 - Users choose disliked ingredients and allergies, which are stored in `Preferences`.
 - Users input ingredients they have and request recipe generation.
- Content-Based Filtering:**
 - Retrieve ingredient dislikes and allergies from `Preferences`.
 - Apply TF-IDF vectorization on user input ingredients.
 - Calculate cosine similarity to find top N similar recipes.
 - Generate and recommend recipes that do not contain the disliked or allergic ingredients.

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.

3. Repositori untuk *Cloud Computing*




Endpoint	Method	Description
AUTH		
/register	POST	Create HealthFlow Account
/login	POST	Login to HealthFlow App
/login/google	POST	Login with Google
/request-recipe	POST	Request Recipe placement
USER PROFILE		
/user/profile	POST	Add user profile details
/user/profile	PUT	Update user profile details
/user/profile/details	GET	View user profile details
/user/dietplan	POST	Add dietplan / allergy ingredients
/user/dietplan	GET	View dietplan / allergy ingredients
RECIPES		
/ingredients	GET	View all ingredients sorted by categories
/ingredients/search	GET	Search ingredients
/recipes	GET	View all recipes title and image
/recipes/search	GET	Search recipes
/recipe/details	GET	View Detailed recipe
/recommendations	GET	View recipe grouped by categories: Easy to Make, Quick to Cook, Breakfast, Dinner, Snack
/rate	POST	Rate the recipe
/recommendations/rate	GET	View recommended recipe based on rating
/rating	GET	View all recipe ratings
/generate	POST	Generate recipe recommendation

API Documentation

4. Repositori Untuk *Mobile Development*

The screenshot displays the GitHub interface for the repository 'healthflowapp' on the 'mobile' branch. The file explorer on the left shows a directory structure including 'android', 'ios', 'api', 'utils', 'app', 'android/app/src/main/java', 'android/app/src/main/res', 'android/app/src/test/java', 'android/build.gradle', 'android/gradle.properties', 'android/gradlew.bat', 'android/settings.gradle', and 'README'. The main area shows the content of the 'README' file, which features a section titled 'Mobile Development' with an image of various food items.

Mobile Development



Healthycious - Mobile Development

Healthycious is a mobile application designed to promote healthier eating habits by providing personalized meal recommendations, based on users' dietary preferences and available ingredients. The app leverages advanced machine learning algorithms to analyze user input and suggest nutritious recipes that can be easily prepared at home. By integrating cloud computing and mobile development, Healthycious ensures a seamless and user-friendly experience, making it easier for individuals to make informed dietary choices and embrace a balanced diet. Whether you are looking to lose weight, manage a health condition, or simply eat more nutritious foods, Healthycious is your go-to companion for a healthier lifestyle.

