

LAPORAN KERJA PRAKTIK

DESAIN ARSITEKTUR SERVER GOOGLE CLOUD UNTUK

MENGOPTIMALKAN KINERJA APLIKASI DAILY CLOUD

DALAM PEMANTAUAN KESEHATAN MENTAL

DI GOOGLE INDONESIA

Diajukan untuk memenuhi persyaratan kelulusan
Mata kuliah FTI335 - Kerja Praktik

oleh:

Fahrul Zaman / 301200020



PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
2024

LEMBAR PENGESAHAN

PROGRAM STUDI TEKNIK INFORMATIKA

DESAIN ARSITEKTUR SERVER GOOGLE CLOUD UNTUK

MENGOPTIMALKAN KINERJA APLIKASI DAILY CLOUD

DALAM PEMANTAUAN KESEHATAN MENTAL

DI GOOGLE INDONESIA

oleh :

Fahrul Zaman / 301200020

disetujui dan disahkan sebagai
Laporan Kerja Praktik

Bandung, 11 Januari 2024

Ketua Program Studi

Yusuf Muharam, S.Kom., M.Kom.

NIK. 0410482000

LEMBAR PENGESAHAN

PEMBIMBING KERJA PRAKTIK

DESAIN ARSITEKTUR SERVER GOOGLE CLOUD UNTUK

MENGOPTIMALKAN KINERJA APLIKASI DAILY CLOUD

DALAM PEMANTAUAN KESEHATAN MENTAL

DI GOOGLE INDONESIA

oleh :

Fahrul Zaman / 301200020

disetujui dan disahkan sebagai
Laporan Kerja Praktik

Bandung, 11 Januari 2024
Pembimbing Kerja Praktik

Yudi Herdiana, S.T., M.T.

NIK. 04104808008

LEMBAR PENGESAHAN
GOOGLE INDONESIA

DESAIN ARSITEKTUR SERVER GOOGLE CLOUD UNTUK
MENGOPTIMALKAN KINERJA APLIKASI DAILY CLOUD
DALAM PEMANTAUAN KESEHATAN MENTAL
DI GOOGLE INDONESIA

oleh :

Fahrul Zaman / 301200020

disetujui dan disahkan sebagai
Laporan Kerja Praktik

Bandung, 11 Januari 2024

Pembimbing Perusahaan



Mohammad Tauchid

ID. 00403050

ABSTRAKSI

Menurut World Health Organization (WHO), kesehatan mental merupakan kondisi dimana individu memiliki kesejahteraan yang tampak dari dirinya yang mampu menyadari potensinya sendiri. Meskipun demikian, Indonesia masih menghadapi tantangan serius terkait masalah kesehatan mental, berdasarkan data dari Riset Kesehatan Dasar (Riskesdas) 2013 sebanyak 1,7% warga Indonesia menderita gangguan mental berat, pada 2018 lebih dari 19 juta penduduk dewasa mengalami gangguan mental emosional. Kerja Praktik yang dilaksanakan mulai tanggal 16 Februari 2023 sampai dengan 28 Juli 2023 di Google Indonesia melalui program Bangkit Academy. Bangkit Academy merupakan sebuah program penyiapan karir dibawah naungan Google Indonesia. Pada akhir program, setiap peserta Kerja Praktik membentuk tim dengan anggota sebanyak 6 orang untuk mengikuti proyek akhir. Dalam upaya mengatasi permasalahan kesehatan mental, penulis melalui pelaksanaan Kerja praktik. memilih tema "Human Healthcare and Living Well-beings". Penulis dan tim mengembangkan sebuah aplikasi bernama Daily Cloud yang bertujuan mengurangi masalah kesehatan mental dengan fitur utama mendeteksi kesehatan mental berdasarkan catatan harian dan ekspresi wajah, serta edukasi melalui artikel-artikel dari sumber terpercaya. Penulis sebagai tim Cloud Computing pada Kerja Praktik ini bertujuan membangun sebuah arsitektur server google cloud untuk aplikasi pemantauan kesehatan mental dengan. Hasil dari kerja praktik, penulis mengembangkan Google Cloud Run sebagai server Back-End, menggunakan Firestore sebagai layanan database yang scalable dan terintegrasi dengan fitur autentikasi dari Firebase Authentication, membangun Back-End REST API sebagai inti, serta melakukan deployment model machine learning menjadi sebuah REST API. Kesimpulan dari kerja praktik ini yaitu arsitektur cloud yang dirancang membawa pengaruh terhadap kinerja aplikasi menjadi optimal dalam pemantauan kesehatan mental.

Kata Kunci: **Bangkit Academy, Daily Cloud, Google Cloud, Kesehatan Mental**

KATA PENGANTAR

Puji syukur atas kehadirat Allah SWT atas rahmat dan hidayah-Nya yang melimpah, shalawat serta salam tidak lupa disampaikan kepada junjungan Nabi Muhammad SAW yang senantiasa memberikan teladan dan petunjuk yang luhur dalam setiap aspek kehidupan.

Ucapan terima kasih yang tulus disampaikan kepada kedua Orang Tua yang senantiasa memberikan kasih sayang, dukungan, dan doa yang tidak terbatas. Serta, ucapan terima kasih juga disampaikan kepada pihak yang telah membantu dalam penyusunan Laporan Kerja Praktik ini, di antaranya:

1. Bapak Dr. Ir. H. Ibrahim Danuwikarsa, M.S., selaku Rektor Universitas Bale Bandung.
2. Ibu Dr. Diana Silaswati, Dra., M.Pd., selaku Wakil Rektor I Universitas Bale Bandung.
3. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi yang juga merupakan Dosen Pembimbing Kerja Praktik.
4. Bapak Yusuf Muhamram, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika.
5. Ibu Dora Songco, selaku Pimpinan Program Bangkit Academy di Google Indonesia yang telah memberikan kesempatan berharga.
6. Mohammad Tauchid, selaku Pembimbing Perusahaan Kerja Praktik.
7. Rekan-rekan seperjuangan yang telah menjadi rekan yang luar biasa selama pelaksanaan Kerja Praktik.
8. Rekan kerja *Capstone Project* C23-PS277 yaitu Sherly Santiadi, Ahsan Firdaus, Kaisar Fredi Valentino , Arya Tri Putra Majiah, dan Maya Septiani Br Simbolon yang telah bekerja sama dalam mengembangkan aplikasi android bernama Daily Cloud.

Laporan Akhir ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan guna memperbaiki dan menyempurnakan laporan ini.

Akhir kata, semoga Laporan Kerja Praktik ini dapat bermanfaat, terutama bagi yang tertarik dalam memahami lebih dalam tentang Kerja Praktik di Google Indonesia melalui program Bangkit Academy.

Bandung, 18 Januari 2024

Penulis

DAFTAR ISI

BAB I PENDAHULUAN.....	1
I.1 Latar Belakang.....	1
I.2 Lingkup.....	2
I.3 Tujuan.....	3
BAB II LINGKUNGAN KERJA PRAKTIK.....	5
II.1 Struktur Organisasi.....	5
II.2 Lingkup Pekerjaan.....	7
II.3 Deskripsi Pekerjaan.....	7
II.4 Jadwal Kerja.....	10
BAB III TEORI PENUNJANG KERJA PRAKTIK.....	15
III.1 Teori Penunjang.....	15
III.2 Peralatan Perancangan Aplikasi.....	16
BAB IV PELAKSANAAN KERJA PRAKTIK.....	34
IV.1 Input.....	34
IV.2 Proses.....	35
IV.2.1 Pembuatan Jadwal Pengerjaan Proyek.....	36
IV.2.2 Perancangan Arsitektur Cloud.....	37
IV.2.3 Perancangan Back-End.....	39
IV.3 Pencapaian Hasil.....	40
BAB V PENUTUP.....	58
V.1 Kesimpulan dan Saran Mengenai Pelaksanaan.....	58
V.1.1 Kesimpulan Pelaksanaan Kerja Praktik.....	58
V.1.2 Saran Pelaksanaan Kerja Praktik.....	59
V.2 Kesimpulan dan Saran Mengenai Substansi.....	59
V.2.1 Kesimpulan.....	59
V.2.2 Saran.....	60

DAFTAR TABEL

Tabel II.1 Jadwal Kerja.....	14
Tabel IV.1 Kebutuhan Perangkat Keras.....	34
Tabel IV.2 Minimum Kebutuhan Perangkat Keras.....	35
Tabel IV.3 Kebutuhan Perangkat Lunak.....	35

DAFTAR GAMBAR

Gambar II.1 Struktur Organisasi Bangkit Academy.....	5
Gambar IV.1 Gantt Chart Project Timeline.....	36
Gambar IV.2 Diagram Perancangan Arsitektur Cloud.....	37
Gambar IV.3 Firestore Database Collection.....	38
Gambar IV.4 Statistik Server pada Resource Google Cloud Run.....	41
Gambar IV.5 Database Firestore.....	42
Gambar IV.6 Firebase Authentication.....	43
Gambar IV.7 Tampilan Login dan Register.....	54
Gambar IV.8 Tampilan Home dan History Activity.....	55
Gambar IV.9 Tampilan Input Story dan Capture Wajah.....	56
Gambar IV.10 Tampilan Menu Artikel.....	57

BAB I

PENDAHULUAN

I.1 Latar Belakang

Menurut World Health Organization (WHO), kesehatan mental merupakan kondisi dimana individu memiliki kesejahteraan yang tampak dari dirinya yang mampu menyadari potensinya sendiri, memiliki kemampuan untuk mengatasi tekanan hidup normal pada berbagai situasi dalam kehidupan, mampu bekerja secara produktif dan menghasilkan, serta mampu memberikan kontribusi kepada komunitasnya. Menurut data dari Riset Kesehatan Dasar (Riskesdas) 2013, sebanyak 1,7% warga Indonesia menderita gangguan mental berat. Hal ini berarti, 1-2 orang dari 1000 penduduk Indonesia mengalami gangguan jiwa berat (Viora dalam Ika, 2015). Sekitar 6% orang mengalami gangguan mental emosional yang tampak sebagai gejala kecemasan dan depresi. Pada saat 2013, pengobatan gangguan jiwa tercatat bahwa kurang dari 10% orang yang mengalami gangguan jiwa mendapatkan layanan terapi oleh petugas kesehatan. Menurut data dari Riset Kesehatan Dasar (Riskesdas) 2018, menunjukkan lebih dari 19 juta penduduk berusia lebih dari 15 tahun mengalami gangguan mental emosional, dan lebih dari 12 juta penduduk berusia lebih dari 15 tahun mengalami depresi (Rokom, 2021). Data tersebut menggambarkan bahwa Indonesia belum berhasil mengatasi masalah kesehatan mental dengan tepat, dan situasi pandemi justru memperburuk kondisi penderita gangguan jiwa.

Google Indonesia bersama dengan GoTo dan Traveloka bekerja sama menyelenggarakan program Bangkit Academy yang merupakan sebuah program yang dirancang untuk menyiapkan karir dan menghasilkan individu dengan kemampuan teknis yang unggul untuk perusahaan teknologi dan startup terkemuka di Indonesia. Visi Bangkit Academy Google Indonesia adalah memberikan program pelatihan yang terstruktur dan berkualitas tinggi untuk menghasilkan lulusan berkaliber tinggi bagi perusahaan teknologi dan perusahaan rintisan kelas dunia. Pada akhir program, setiap peserta kerja praktik di Bangkit Academy Google Indonesia harus membentuk tim dengan anggota sebanyak 5-6

orang untuk mengikuti proyek akhir (*Capstone Project*) sebagai salah satu persyaratan kelulusan. Penulis bersama dengan rekan kerja tim memilih tema "*Human Healthcare and Living Well-beings*" dengan fokus utama mengenai masalah kesehatan mental khususnya di Indonesia.

Dalam menghadapi tantangan kesehatan mental, penting untuk merancang strategi yang tidak hanya efektif tetapi juga terintegrasi secara menyeluruh. Strategi ini haruslah bersifat berkelanjutan, tidak hanya berfokus pada penanganan kondisi saat ini, tetapi juga pada pencegahan dan rehabilitasi. Upaya ini tidak hanya berlaku bagi individu secara langsung, tetapi juga melibatkan keluarga dan masyarakat secara luas. Sayangnya, masih banyak orang yang merasa tidak perlu memantau kesehatan mental mereka secara teratur. Adanya stigma terkait mencari bantuan psikologis seringkali membuat seseorang enggan untuk mencari perawatan yang diperlukan. Untuk mengatasi hambatan ini, kami bersama rekan kerja telah mengembangkan sebuah aplikasi inovatif yang bertujuan untuk mengurangi masalah kesehatan mental. Melalui aplikasi yang penulis bersama rekan kerja kembangkan diharapkan dapat mengurangi masalah kesehatan mental dengan melacak jurnal harian pengguna yang dilengkapi dengan fitur-fitur seperti deteksi kesehatan mental berdasarkan catatan harian, deteksi ekspresi wajah pengguna, menampilkan ringkasan riwayat deteksi, serta memberikan edukasi tentang kesehatan mental melalui artikel-artikel dari sumber terpercaya.

I.2 Lingkup

Dalam pelaksanaan kerja praktik di Google Indonesia pada program Bangkit Academy, lingkup kegiatan kerja praktik difokuskan pada pelaksanaan Proyek *Capstone* (*Capstone Project*). Fokus kegiatan kerja praktik ini secara khusus difokuskan untuk mengembangkan dan menerapkan solusi inovatif yang diharapkan mampu memberikan dampak positif dalam menangani permasalahan kesehatan mental di Indonesia. Proyek Capstone tersebut menjadi inti dari kegiatan kerja praktik ini, di mana tujuannya adalah untuk menciptakan solusi yang tidak hanya kreatif tetapi juga praktis dalam mengatasi tantangan seputar kesehatan mental di masyarakat Indonesia.

Lingkup dalam proyek *capstone* pada kerja praktik ini mencakup serangkaian tahapan dan aspek, antara lain:

1. Penetapan Tema dan Fokus Proyek:

- Memilih tema "*Human Healthcare and Living Well-beings*" dengan penekanan khusus pada masalah kesehatan mental di Indonesia sebagai fokus utama proyek.

2. Analisis Kebutuhan Pengguna:

- Melakukan analisis mendalam terhadap kebutuhan pengguna terkait aplikasi kesehatan mental.
- Meneliti preferensi dan tantangan yang dihadapi oleh masyarakat Indonesia.

3. Pengembangan Aplikasi Kesehatan Mental:

- Membangun aplikasi kesehatan mental yang memiliki kemampuan melacak jurnal harian pengguna.
- Mengintegrasikan fitur-fitur penting seperti deteksi kesehatan mental berdasarkan catatan harian dan ekspresi wajah pengguna.

4. Pengujian dan Evaluasi:

- Melakukan pengujian aplikasi secara menyeluruh untuk menilai keefektifan dan kehandalan fitur yang telah diimplementasikan.
- Melakukan evaluasi berkala guna memperbaiki dan meningkatkan kualitas aplikasi berdasarkan umpan balik dari pengujian dan pengguna.

5. Dokumentasi dan Presentasi:

- Membuat dokumentasi komprehensif yang mencakup seluruh proses pengembangan proyek, termasuk metodologi, desain, dan hasil pengujian.
- Menyusun presentasi proyek *Capstone* yang informatif dan persuasif untuk memaparkan hasil dan solusi kepada tim penguji dan *stakeholders*.

I.3 Tujuan

Tujuan utama pelaksanaan kerja praktik di Google Indonesia adalah membangun sebuah arsitektur server *google cloud* untuk aplikasi pemantauan kesehatan mental dengan integrasi teknologi.

1. Membangun arsitektur *server* yang optimal guna meningkatkan fungsionalitas dan performa aplikasi yang akan dibangun.
2. Membangun *REST API* sebagai *Back-End* dari aplikasi yang akan dibangun dengan beberapa fungsi yaitu pengelolaan data pengguna, pengelolaan jurnal pengguna untuk deteksi kesehatan mental, dan penyediaan artikel sebagai edukasi kepada pengguna.
3. Melakukan *deployment* model *machine learning* ke arsitektur *cloud* melalui *REST API* yang berfungsi untuk mendeteksi dini masalah kesehatan mental berdasarkan jurnal harian.

BAB II

LINGKUNGAN KERJA PRAKTIK

II.1 Struktur Organisasi



Gambar II.1 Struktur Organisasi Bangkit Academy

Struktur organisasi Bangkit Academy 2023 merupakan sebuah kerangka formal yang menunjukkan bagaimana tugas dan tanggung jawab didistribusikan di antara anggota perusahaan dan hubungan yang terjalin antara pihak dalam organisasi tersebut, tujuannya adalah untuk mencapai tujuan organisasi secara efektif. Struktur organisasi Bangkit Academy 2023 dapat dilihat pada Gambar II.1. Tugas pokok dan fungsi setiap posisi pada Struktur Organisasi Bangkit Academy 2023 adalah sebagai berikut:

1. Google Indonesia

Google Indonesia berperan sebagai perusahaan induk dari Bangkit Academy. Google Indonesia bertanggung jawab atas keseluruhan program Bangkit Academy, termasuk perencanaan, pelaksanaan, dan evaluasi. Google Indonesia selalu memastikan bahwa Bangkit Academy berjalan lancar dan sesuai dengan tujuan serta menyediakan sumber daya dan dukungan yang diperlukan untuk menjalankan program, termasuk anggaran, infrastruktur, dan tenaga kerja.

2. *Program Lead*

Program Lead bertanggung jawab atas pengelolaan keseluruhan program dengan fokus pada strategi, visi, misi, dan tujuan. *Program Lead* memiliki tugas dan tanggung jawab sebagai berikut:

- Memimpin pengembangan dan pelaksanaan program Bangkit Academy di Google Indonesia.
- Membangun dan menjaga hubungan dengan mitra program
- Mengelola anggaran dan sumber daya program
- Memantau dan mengevaluasi kinerja program

3. *ID Program Manager*

ID Program Manager bertanggung jawab atas pelaksanaan program Bangkit Academy di Indonesia. *ID Program Manager* bekerja sama dengan Google Indonesia, GoTo, Traveloka, dan DeepTech Foundation untuk memastikan program berjalan lancar dan sesuai dengan tujuan.

ID Program Manager memiliki tugas dan tanggung jawab sebagai berikut:

- Mengelola pelaksanaan program Bangkit Academy di Indonesia
- Membangun dan menjaga hubungan dengan mitra program di Indonesia
- Menyusun dan mengelola anggaran program di Indonesia
- Memantau dan mengevaluasi kinerja program di Indonesia

4. *Learning Support Manager*

Learning Support Manager bertanggung jawab atas dukungan pembelajaran bagi peserta program Bangkit Academy. *Learning Support Manager* menyediakan berbagai layanan, termasuk materi pembelajaran, bimbingan belajar, dan bantuan teknis.

Learning Support Manager memiliki tugas dan tanggung jawab sebagai berikut:

- Menyediakan materi pembelajaran yang berkualitas
- Memberikan bimbingan belajar kepada peserta kerja praktik.
- Menyediakan bantuan teknis kepada peserta kerja praktik.
- Melakukan evaluasi secara berkala terhadap kebutuhan pembelajaran peserta untuk menyesuaikan dan meningkatkan kualitas layanan pendukung.

5. *Cohort Manager*

Cohort Manager bertanggung jawab atas pengelolaan setiap kelompok peserta Bangkit Academy. *Cohort Manager* bertugas untuk memastikan bahwa peserta mendapatkan pengalaman belajar yang maksimal. Penulis bekerja dibawah naungan *Cohort Manager*. *Cohort Manager* memiliki tugas dan tanggung jawab sebagai berikut:

- Mengelola kegiatan belajar mengajar dalam kelompok
- Membantu peserta program dalam menyelesaikan tugas dan proyek
- Membina hubungan antar peserta

6. *Curriculum Manager*

Curriculum Manager bertanggung jawab atas pengembangan dan pemeliharaan kurikulum Bangkit Academy. *Curriculum Manager* memastikan kurikulum sesuai dengan kebutuhan industri dan memenuhi standar kualitas yang tinggi. *Curriculum Manager* memiliki tugas dan tanggung jawab sebagai berikut:

- Mengembangkan kurikulum yang sesuai dengan kebutuhan industri
- Memelihara kurikulum agar tetap relevan dengan perkembangan teknologi
- Melakukan evaluasi kurikulum secara berkala

II.2 Lingkup Pekerjaan

Dalam pelaksanaan Cloud Computing Learning Path Bangkit Academy 2023 terdapat beberapa kegiatan, di antaranya:

1. Pembelajaran Mandiri secara Daring (*Online Self-paced Learning*)
2. Pembelajaran Serentak secara Daring (*Online Synchronous Sessions*)
3. Refleksi dan Konsultasi (*Reflection and Consultation Sessions*)
4. Bangkit Students Team Meeting
5. Proyek Capstone (*Capstone Project*)

II.3 Deskripsi Pekerjaan

1. Pembelajaran Mandiri secara Daring (*Online Self-paced Learning*)

Peserta dapat mengikuti pembelajaran secara mandiri (*asynchronous*) melalui platform Dicoding, Coursera, dan Google Cloud Skill Boost. Setelah

berhasil menyelesaikan *course* pembelajaran, peserta akan memperoleh sertifikat kompetensi yang sebagai tanda keberhasilan dalam menyelesaikan berbagai *course* terkait *Cloud Computing*.

Berikut adalah beberapa materi pembelajaran yang disediakan oleh Bangkit Academy untuk *Cloud Computing Learning Path*:

1. Belajar Dasar Pemrograman JavaScript
2. Belajar Dasar Pemrograman Web
3. Google IT Support Professional Certificate - The Bits and Bytes of Computer Networking
4. Google Cloud Computing Foundations
5. Belajar Membuat Aplikasi Back-End untuk Pemula dengan Google Cloud
6. Google IT Support Professional Certificate - System Administration and IT Infrastructure Services
7. Cloud Engineer Learning Path
8. Menjadi Google Cloud Engineer
9. Google Cloud Skills Boost Quest
10. Preparing for Associate Cloud Engineer Certification
11. Simulasi Ujian Associate Cloud Engineer

2. Pembelajaran Serentak secara Daring (*Online Synchronous Sessions*)

Peserta mengikuti pembelajaran secara serentak (*synchronous*) dengan instruktur atau *Instructor-led Training* (ILT) bersama dengan instruktur yang *expert*. Pembelajaran ILT meliputi beberapa materi tentang *Technical Skills*, *Soft Skills*, dan *English Skills* bersama *native speaker* yang dilaksanakan pada platform Google Meet dengan durasi 120 menit untuk *ILT Technical* dan *ILT Soft Skills* serta 90 menit untuk *ILT English* yang dijadwalkan satu kali dalam satu minggu.

Berikut adalah daftar sesi *Instructor-led Training* untuk *Technical Skills*:

1. *Introduction to Google Cloud*
2. *JavaScript for Server-Side Development*
3. *Build Great Solutions with Compute and Network Services in Google Cloud*
4. *Store Your Data Without Worry in Google Cloud*

5. *Simplify Your Life with Automation and Operation Services in Google Cloud*
6. *Manage Your Cloud Resources Securely*
7. *Prepare Yourself Before Taking the ACE Certification*

Berikut adalah daftar sesi *Instructor-led Training* untuk *Soft Skills*:

1. *Growth Mindset and The Power of Feedback*
2. *Time Management*
3. *Critical Thinking and Problem Solving*
4. *Adaptability and Resilience*
5. *Project Management*
6. *Professional Communication and Networking*
7. *Digital Branding and Interview Communication*

Berikut adalah daftar sesi *Instructor-led Training* untuk *English Skills*:

1. *Spoken Correspondence*
 2. *Expressing Opinion*
 3. *Business Presentation*
- 3. Refleksi dan Konsultasi dengan Mentor (*Reflection and Consultation Sessions*)**

Peserta dapat merefleksikan pembelajarannya dan melakukan konsultasi terkait akademik maupun non-akademik dalam sesi konsultasi mingguan (*Weekly Consultation*) bersama dengan mentor dan rekan satu kelas yang telah ditentukan oleh Bangkit Academy pada awal program. Sesi ini dilaksanakan pada *platform Google Meet* dengan durasi 60 menit dan dijadwalkan dilaksanakan satu kali dalam satu minggu.

4. Bangkit Students Team Meeting

Bangkit Academy sangat berkomitmen untuk memastikan pengalaman pembelajaran yang optimal bagi setiap peserta programnya, oleh karena itu pihak Bangkit Academy secara rutin mengadakan pertemuan dengan peserta program setiap bulan, dengan tujuan memberikan informasi tentang pembelajaran yang akan dilakukan pada bulan berikutnya. Selain itu, pertemuan tersebut juga

bertujuan untuk memberikan rangkuman dan evaluasi hasil pembelajaran peserta selama satu bulan sebelumnya.

5. Proyek *Capstone* (*Capstone Project*)

Peserta kerja praktik program Bangkit Academy di Google Indonesia akan bekerja dalam tim proyek berskala nasional untuk menyelesaikan masalah yang relevan dalam lingkungan sekitar. Setiap tim proyek akan terdiri dari 5-6 orang yang mewakili ketiga jalur pembelajaran yang ada dalam program. Peserta memiliki kebebasan untuk membentuk tim dan memilih tema proyek. Waktu penggerjaan proyek akan berlangsung selama 1 bulan.

Setelah tahap penggerjaan proyek, 15 tim terbaik akan dipilih dan mendapatkan kesempatan untuk mengikuti program inkubasi. Program ini akan memberikan pendanaan lebih lanjut sebesar 10.000 USD serta dukungan dan bimbingan dalam mengembangkan solusi peserta menjadi produk atau layanan yang lebih matang dan siap untuk diluncurkan ke pasar.

II.4 Jadwal Kerja

Tanggal Kegiatan	Kegiatan		
	<i>Soft skills</i>	<i>English</i>	<i>Cloud</i>
16 Februari - 17 Februari 2023	<i>Preread SS 1 "Growth Mindset and The Power of Feedback"</i>		<i>Matriculation</i>
20 Februari - 24 Februari 2023	ILT SS 1		
27 Februari - 03 Maret 2023	<i>Assignment SS 1</i>	<i>English Pre-Test</i>	<i>ILT Tech 1 Introduction to Google Cloud</i>
			<i>The Bits and Bytes of Computer Networking by Coursera</i>
	<i>Preread SS 2 "Time Management"</i>		<i>Google Cloud Computing Foundations: Cloud Computing by Google SkillBoost</i>

			<i>Google Cloud Computing Foundations: Infrastructure in Google Cloud by Google SkillBoost</i>
			<i>Google Cloud Computing Foundations: Networking and Security by Google SkillBoost</i>
			<i>Google Cloud Computing Foundations: Data, ML, and AI in Google by Google SkillBoost</i>
			<i>Create and Manage Cloud Resources (Quest)</i>
			<i>Perform Foundational Infrastructure Tasks in Google Cloud (Quest)</i>
			<i>Build and Secure Networks in Google Cloud (Quest)</i>
06 Maret - 10 Maret 2023	ILT SS 2		<i>Perform Foundational Data, ML, and AI Tasks in Google Cloud (Quest)</i> Belajar Dasar Pemrograman Web (<i>up to done</i>) by Dicoding
13 Maret - 17 Maret 2023	<i>Assignment SS 2</i>		<i>ILT Tech 2 JavaScript for Server-Side Development</i> Belajar Dasar Pemrograman JavaScript (<i>up to done</i>) by Dicoding
	<i>Preread SS 3 "Critical Thinking and Problem Solving"</i>		Belajar Membuat Aplikasi <i>Back-End</i> untuk Pemula dengan <i>Google Cloud</i> (<i>up to Kuis Pengenalan Back-End</i>) by Dicoding
20 Maret - 24 Maret 2023	ILT SS 3		Belajar Membuat Aplikasi <i>Back-End</i> untuk Pemula dengan

			Google Cloud (<i>up to done</i>) by Dicoding
27 Maret - 31 Maret 2023	<i>Assignment SS 3</i>		<i>ILT Tech 3</i> <i>Build Great Solutions with Compute and Network Services in Google Cloud</i>
			<i>System Administration and IT Infrastructure Services by Coursera</i>
			<i>Preparing for your Associate Cloud Engineer Journey by Google SkillBoost</i>
03 April - 07 April 2023	<i>ILT SS 4</i>		<i>Google Cloud Fundamentals: Core Infrastructure by Google SkillBoost</i>
			<i>Essential Google Cloud Infrastructure: Foundation by Google SkillBoost</i>
			<i>Essential Google Cloud Infrastructure: Core Services by Google SkillBoost</i>
10 April - 14 April 2023	<i>Assignment SS 4</i>		<i>ILT Tech 4</i> <i>Store Your Data Without Worry in Google Cloud</i>
			<i>Elastic Google Cloud Infrastructure: Scaling and Automation by Google SkillBoost</i>
			<i>Getting Started with Google Kubernetes Engine by Google SkillBoost</i>
17 April - 21 April 2023			<i>Logging, Monitoring and Observability in Google Cloud by Google SkillBoost</i>
			<i>Getting Started with Terraform for</i>

			<i>Google Cloud by Google SkillBoost</i>
Cuti Idul Fitri 20 April - 26 April 2023			
24 April - 28 April 2023	<i>Preread SS 5 "Project Management"</i>		<i>Set Up and Configure a Cloud Environment in Google Cloud (Quest)</i>
			<i>Automating Infrastructure on Google Cloud with Terraform (Quest)</i>
01 Mei - 05 Mei 2023	ILT SS 5		Menjadi <i>Google Cloud Engineer (up to Kuis Monitoring dan Logging) by Dicoding</i>
08 Mei - 12 Mei 2023	<i>Assignment SS 5</i>		<i>ILT Tech 5 Simplify Your Life with Automation and Operation Services in Google Cloud</i>
	<i>Preread SS 6 "Professional Communication and Networking"</i>		Menjadi <i>Google Cloud Engineer (up to done) by Dicoding</i>
			<i>Application Development with Cloud Run</i>
15 Mei - 19 Mei 2023	ILT SS 6		<i>Hybrid Cloud Modernizing Applications with Anthos by Google SkillBoost</i>
22 Mei - 26 Mei 2023	<i>Assignment SS 6</i>		<i>ILT Tech 6 Manage Your Cloud Resources Securely</i>
	<i>Preread SS 7 "Personal Branding and Interview Communication"</i>		<i>Serverless Cloud Run Development (Quest)</i>
			<i>Deploy to Kubernetes in Google Cloud (Quest)</i>
29 Mei - 02 Juni 2023	ILT SS 7		<i>CAPSTONE PROJECT</i>

05 Juni - 09 Juni 2023	<i>Assignment SS 7</i>		
12 Juni - 16 Juni 2023			
19 Juni - 23 Juni 2023			
26 Juni - 30 Juni 2023			<i>Judging - Target: 15 Best teams</i> <i>Tech: Certification Prep (ACE Simulation Course Dicoding)</i>
03 Juli - 07 Juli 2023		English Post-Test	<i>ILT Tech 7</i> <i>Prepare Yourself Before Taking the ACE Certification</i>
10 Juli - 14 Juli 2023			<i>Announcement & Incubation Offering</i> <i>Tech: Certification Prep & ILT Tech 7 (Cloud Explorations + ACE Examination Practice)</i>
17 Juli - 21 Juli 2023			<i>End of Learning, Certification Offering, Merchandise</i>
24 Juli - 28 Juli 2023			<i>Transcript & Administration</i>
			<i>Clarification, Legal & Letters, Closing.</i>

Tabel II.1 Jadwal Kerja

BAB III

TEORI PENUNJANG KERJA PRAKTIK

III.1 Teori Penunjang

Pada pelaksanaan kerja Praktik pada program Bangkit Academy di Google Indonesia, peserta kerja Praktik tidak hanya diberikan tugas-tugas teknis, tetapi juga diharapkan memiliki landasan pengetahuan teoritis yang kuat. Adanya pengetahuan dan teori yang diperoleh selama masa perkuliahan menjadi elemen kunci yang mendukung peserta kerja Praktik dalam menyelesaikan tugas-tugasnya secara efektif.

Berikut adalah beberapa bidang pengetahuan dan teori yang menjadi dasar bagi peserta kerja Praktik:

1. Teori Basis Data (Mata kuliah TIF310 Basis Data dan TIF311 Sistem Basis Data)

Pengetahuan dan pemahaman terkait Basis Data dan Sistem Basis Data sangat penting dalam mengelola informasi secara efisien. Mata kuliah TIF310 dan TIF311 memberikan dasar yang kokoh untuk implementasi dan manajemen data, termasuk konsep-konsep RDBMS (*Relational Database Management System*).

2. Teori Arsitektur Komputer (Mata Kuliah TIF305 Organisasi dan Arsitektur Komputer)

Pemahaman tentang arsitektur komputer, termasuk komponen seperti RAM dan Processor, memberikan landasan yang kritis untuk pengembangan dan pemahaman sistem komputer. Peserta kerja Praktik memperoleh pengetahuan ini melalui mata kuliah TIF305 Organisasi dan Arsitektur Komputer.

3. Teori Jaringan Komputer (Mata kuliah TIF317 Jaringan Komputer)

Bidang jaringan komputer adalah elemen penting dalam dunia teknologi informasi modern. Mata kuliah TIF317 Jaringan Komputer memberikan pemahaman mendalam tentang konsep-konsep jaringan, termasuk IP Address, Routing, dan Model jaringan client-server, yang merupakan dasar penting dalam pengembangan aplikasi yang melibatkan komunikasi data.

4. Teori Mobile Computing (Mata kuliah TIF333 *Wireless/Mobile Computing*)

Dalam era mobile computing, pemahaman tentang teknologi nirkabel dan mobile menjadi semakin krusial. Mata kuliah TIF333 *Wireless/Mobile Computing* menyediakan landasan bagi peserta kerja Praktik untuk memahami dan mengembangkan aplikasi yang berfokus pada lingkungan mobile.

5. Teori Sistem Operasi (Mata kuliah TIF306 Sistem Operasi)

Pemahaman tentang sistem operasi merupakan hal yang mendasar dalam memahami cara sebuah sistem beroperasi. Mata kuliah TIF306 Sistem Operasi memberikan pengetahuan dan wawasan yang diperlukan bagi peserta kerja Praktik untuk berinteraksi dengan sistem secara efektif.

III.2 Peralatan Perancangan Aplikasi

Selama pelaksanaan kerja Praktik pada Program Bangkit Academy di Google Indonesia, terdapat beberapa peralatan utama dalam mendukung jalannya pelaksanaan kerja praktik dan perancangan aplikasi pada saat *Capstone Project*.

A. Software dan Tools

Berikut adalah daftar beberapa *software* dan *tools* yang digunakan:

1. Google Cloud Platform

Google Cloud Platform (GCP) merupakan serangkaian layanan komputasi awan yang disediakan oleh Google, yang mencakup berbagai layanan seperti *compute*, *storage*, dan lainnya. GCP berjalan pada infrastruktur yang sama dengan infrastruktur yang digunakan oleh Google itu sendiri. Berikut adalah beberapa layanan utama yang tersedia di *Google Cloud Platform* (GCP):

a. *Compute Services*

Google Cloud Platform (GCP) menyediakan layanan komputasi yang dapat diandalkan melalui *Compute Services*. Ini mencakup Google Compute Engine (GCE) untuk menyediakan mesin virtual (VM) dan Google Kubernetes Engine (GKE) untuk manajemen kontainer. Dengan Compute Services, pengguna dapat menjalankan aplikasi, mengelola beban kerja, dan mengalokasikan sumber daya komputasi sesuai kebutuhan pelanggan.

b. *Storage Services*

Layanan Penyimpanan Google Cloud memungkinkan pengguna untuk menyimpan dan mengelola data dengan berbagai cara. Google Cloud Storage menawarkan penyimpanan objek yang aman dan scalable, sementara Google Cloud SQL memberikan solusi penyimpanan database relasional. Dengan *Storage Services*, pengguna dapat mengelola data pelanggan dengan efisien dan aman di lingkungan cloud.

c. *Networking Services*

Google Cloud Platform menawarkan *Networking Services* yang memungkinkan pengguna mengelola dan mengoptimalkan lalu lintas data di dalam dan keluar dari cloud. Layanan ini mencakup Google Cloud Load Balancing untuk mendistribusikan lalu lintas dengan seimbang dan Virtual Private Cloud (VPC) untuk membuat jaringan pribadi yang dapat dikonfigurasi sesuai kebutuhan.

d. *Data Analytics*

Untuk memproses dan menganalisis data besar, GCP menyediakan layanan *Data Analytics* seperti Google BigQuery untuk analisis data real-time dan Google Cloud Dataprep untuk persiapan data. Layanan ini membantu pengguna mendapatkan wawasan berharga dari informasi yang tersimpan dalam skala besar.

e. *Machine Learning and AI*

Dalam bidang Machine Learning dan Kecerdasan Buatan, GCP menyediakan alat dan lingkungan untuk pengembangan, pelatihan, dan implementasi model. Contohnya termasuk Google Cloud AI Platform untuk pelatihan model dan Cloud Vision API untuk pengenalan gambar. Layanan ini membantu organisasi memanfaatkan kecerdasan buatan dalam aplikasi dan layanan pengguna.

f. *Big Data Processing*

Layanan *Big Data Processing* di GCP, seperti Google Cloud Dataflow layanan yang memungkinkan pengguna untuk melakukan pemrosesan data

secara terdistribusi dan adaptif, ini dapat digunakan untuk menganalisis dan mengolah aliran data secara real-time atau batch. Serta Google Cloud Pub/Sub untuk sistem antrian pesan, memungkinkan pengguna untuk menangani pemrosesan dan analisis data skala besar dengan efisiensi tinggi. Ini mendukung pengolahan data yang dinamis dan adaptif.

g. *Application Development*

Untuk mendukung pengembangan, pengujian, dan implementasi aplikasi, Google Cloud Platform menyediakan layanan *Application Development*. Google App Engine memungkinkan platform pemrograman aplikasi, sementara Cloud Functions memberikan lingkungan untuk mengelola fungsi *serverless*. Layanan ini membantu pengembang membangun dan menjalankan aplikasi pengguna dengan mudah di lingkungan cloud.

2. Docker

Docker adalah sebuah perangkat lunak yang bersifat *free* dan *open source* yang merupakan salah satu dari implementasi *Container Manager*. *Container* merupakan sebuah *environment* terisolasi yang berjalan di atas sistem operasi yang ada dan memungkinkan aplikasi dan proses untuk berjalan dengan cara yang terisolasi dari *environment* lainnya. Docker sebagai teknologi container, memberikan solusi untuk mengemas, mendistribusikan, dan menjalankan aplikasi dengan konsisten di berbagai lingkungan.

Teknologi *Docker Container* pertama kali diperkenalkan pada tahun 2013 sebagai *open source docker engine* yang berarti kode sumbernya dapat diakses dan dimodifikasi oleh siapa saja sesuai dengan lisensi yang diberikan. Docker memanfaatkan konsep komputasi yang sudah ada seputar *container*, khususnya di dunia *Linux* yang dikenal sebagai cgroups dan namespaces. Keunikan teknologi Docker terletak pada fokusnya terhadap kebutuhan para pengembang dan operator sistem untuk memisahkan dependensi aplikasi dari infrastruktur.

3. Google Cloud Run

Google Cloud Run adalah salah satu layanan *serverless* di Google Cloud Platform memungkinkan untuk menjalankan *container* berbasis Docker secara

otomatis dan dengan *automatic scaling* tanpa perlu mengelola infrastruktur atau server secara langsung. Dengan Google Cloud Run, pengguna dapat fokus pada pengembangan dan implementasi aplikasi tanpa perlu khawatir tentang manajemen infrastruktur.

Berikut adalah beberapa fitur yang tersedia dalam layanan Google Cloud Run mencakup:

- *Serverless*: Google Cloud Run merupakan layanan *serverless*, yang berarti pengguna tidak perlu memikirkan tentang infrastruktur otomatisasi dan *scaling*. Layanan ini akan menangani otomatisasi dan *scaling* sesuai kebutuhan.
- *Containerized Applications*: Google Cloud Run mendukung *container* Docker, sehingga pengguna dapat mengemas dan menjalankan aplikasi dalam *container*. Hal ini memberikan fleksibilitas dan portabilitas dalam mengelola aplikasi.
- *Auto-scaling*: Layanan ini otomatis mengelola *scaling* dari aplikasi sesuai dengan jumlah permintaan yang diterima. *Container* akan dihentikan secara otomatis jika tidak ada permintaan, dan akan menambah *scale* aplikasi jika permintaan tinggi.
- Integrasi dengan Google Cloud *Services*: Google Cloud Run terintegrasi dengan layanan Google Cloud lainnya seperti Cloud Monitoring, Cloud Logging, dan Cloud Build untuk memperoleh informasi dan mengelola aplikasi dengan lebih baik.
- Biaya berdasarkan Penggunaan: Harga Google Cloud Run didasarkan pada penggunaan aktual. Pengguna hanya membayar untuk sumber daya yang digunakan selama eksekusi aplikasi

4. JavaScript

JavaScript (JS) adalah bahasa pemrograman tingkat tinggi yang digunakan untuk mengembangkan aplikasi web interaktif dan dinamis yang pertama kali diperkenalkan oleh Netscape pada tahun 1995. Sejak saat itu, JavaScript telah

menjadi bahasa pemrograman yang sangat populer dan digunakan secara luas untuk mengembangkan aplikasi web interaktif dan dinamis.

Bahasa ini memiliki popularitas yang tinggi di internet dan dapat beroperasi di sebagian besar penjelajah web terkemuka, termasuk Google Chrome, Internet Explorer (IE), Mozilla Firefox, Netscape, dan Opera. Kode JavaScript dapat disematkan dalam halaman web menggunakan tag script. Sebagai salah satu teknologi inti World Wide Web bersama dengan HTML dan CSS, JavaScript memainkan peran penting dalam menghadirkan interaktivitas pada halaman web dan merupakan komponen esensial dari aplikasi web modern. JavaScript memiliki beberapa konsep dasar yang membentuk dasar pemahaman untuk mengembangkan aplikasi web. Berikut adalah beberapa konsep kunci dalam JavaScript:

- Variabel dan Tipe Data: JavaScript menggunakan variabel untuk menyimpan dan merujuk pada data. Tipe data umum meliputi string, number, boolean, object, dan lainnya.
- Operator: JavaScript memiliki berbagai jenis operator, termasuk operator aritmatika seperti penjumlahan dan pengurangan, operator perbandingan sama dengan dan tidak sama dengan, serta operator logika seperti AND, OR, dan NOT.
- Kondisional (*if-else*): Digunakan untuk mengambil keputusan berdasarkan kondisi tertentu. Contohnya, pernyataan if akan dieksekusi jika kondisi yang diberikan bernilai benar.
- Perulangan (*loop*): Digunakan untuk mengeksekusi serangkaian pernyataan secara berulang. *Loop for* digunakan untuk iterasi dengan jumlah iterasi yang diketahui, sementara *loop while* digunakan untuk iterasi selama kondisi tertentu benar.
- Fungsi: Fungsi adalah blok kode yang dapat dipanggil untuk mengeksekusi tugas tertentu. JavaScript mendukung pembuatan fungsi dan fungsi anonim.
- *Array*: *Array* adalah struktur data yang digunakan untuk menyimpan sejumlah nilai dalam satu variabel. Nilai-nilai dalam array diindeks, dan

pengelolaan data menggunakan array memungkinkan manipulasi data yang lebih efisien.

- **Objek:** Objek adalah struktur data kompleks yang dapat menyimpan data dan fungsi dalam satu unit. Objek memiliki properti (variabel) dan metode (fungsi) yang dapat diakses dan dimanipulasi. Objek memungkinkan representasi yang lebih terstruktur dan terorganisir dari informasi dalam program JavaScript.
- **Event Handling:** JavaScript memungkinkan pengembangan aplikasi yang responsif dengan merespons peristiwa (*events*) seperti klik tombol, input pengguna, atau perubahan status halaman. Ini dilakukan dengan menggunakan *event listeners* yang dapat menangkap dan menanggapi interaksi pengguna.
- **DOM Manipulation:** *Document Object Model (DOM)* adalah representasi struktur dokumen HTML yang dapat dimanipulasi oleh JavaScript. Melalui *DOM manipulation*, pengembang dapat membuat perubahan dinamis pada elemen-elemen HTML dan CSS, seperti menambah, mengubah, atau menghapus elemen, membuat animasi, dan lainnya.
- **Asynchronous JavaScript (Callback, Promise, Async/Await):** JavaScript mendukung pemrograman asinkron untuk menangani tugas-tugas non-blocking. Ini melibatkan penggunaan callback functions, promises, dan async/await.
- **Closure:** *Closure* terjadi ketika sebuah fungsi memiliki akses ke variabel-variabel di luar cakupannya. Ini memungkinkan pembuatan fungsi yang mempertahankan referensi ke lingkungan di mana dibuat, bahkan setelah fungsi tersebut telah selesai dieksekusi.
- **Prototype Inheritance:** JavaScript menggunakan konsep pewarisan prototipe, di mana suatu objek dapat mewarisi properti dan metode dari objek prototipenya. Ini memungkinkan pembuatan hierarki objek dan berbagi implementasi di antara objek-objek yang terkait.
- **Module System:** Modul memungkinkan pengorganisasian kode JavaScript ke dalam bagian-bagian terpisah, membantu dalam pembangunan dan

pemeliharaan kode yang terstruktur dan terkelola. Ini meningkatkan modularitas dan memudahkan pengelolaan dependensi.

- Ajax (*Asynchronous JavaScript and XML*): Teknik yang memungkinkan pengiriman dan penerimaan data dari server secara asinkron tanpa memuat ulang halaman web. Ini meningkatkan responsivitas aplikasi dengan mempercepat pertukaran data antara browser dan server.
- ES6+ Features: JavaScript terus berkembang dengan penambahan fitur-fitur baru. ECMAScript 6 (ES6) dan versi selanjutnya membawa banyak peningkatan, termasuk *let/const*, *arrow functions*, *template literals*, *destructuring*, dan lainnya.

5. Node.js

Node.js merupakan sebuah lingkungan runtime *JavaScript* yang pertama kali diperkenalkan pada tahun 2009 oleh Ryan Dahl, yang memungkinkan *developer* untuk menulis kode *JavaScript* di mana pun, tidak hanya terbatas pada lingkungan *browser*. Node.js berjalan pada *V8 JavaScript Engine* yang merupakan inti dari browser *Google Chrome*, hal ini memungkinkan Node.js memiliki performa yang tinggi.

Node.js adalah proyek *open source*, yang berarti kode sumbernya dapat diakses, dimodifikasi, dan didistribusikan secara bebas. Selain itu, Node.js bersifat *cross-platform*, artinya dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, dan Linux. Aplikasi Node.js berjalan pada *single process*, namun dapat menangani banyak koneksi secara bersamaan. Ini dicapai dengan menggunakan model *non-blocking* dan *asynchronous I/O*. Saat menunggu operasi *I/O*, Node.js tidak memblokir proses lain, sehingga aplikasi dapat terus berjalan tanpa harus menunggu proses selesai.

Node.js memanfaatkan pendekatan *asynchronous*, yang memungkinkan eksekusi operasi-operasi seperti *I/O*, file system access, atau panggilan ke database tanpa memblokir thread utama. Hal ini membuat aplikasi Node.js menjadi lebih responsif dan efisien dalam menangani banyak koneksi secara simultan.

6. Node Package Manager (npm)

Node Package Manager (npm) adalah manajer paket untuk lingkungan runtime JavaScript Node.js. Node package manager adalah bagian dari ekosistem Node.js yang digunakan oleh pengembang untuk mengelola dependensi (*library* dan *modules* JavaScript) dalam sebuah proyek. Berikut adalah beberapa fitur utama terkait dengan Node Package Manager:

- Manajemen Paket: NPM memungkinkan pengguna untuk mencari, menginstal, dan mengelola paket JavaScript dari *repository* NPM. Paket-paket ini dapat berupa modul-modul, *library*, atau alat-alat yang dapat digunakan dalam proyek pengembangan perangkat lunak.
- Instalasi Paket: pengguna dapat dengan mudah menginstal paket JavaScript yang diperlukan untuk proyek. Perintah *npm install* digunakan untuk mengunduh dan menginstal paket dari *repository* NPM ke dalam proyek.
- Manajemen Dependensi: NPM secara otomatis melacak dan mencatat dependensi proyek, yaitu paket-paket yang diperlukan oleh proyek tersebut agar dapat berfungsi. NPM membuat daftar dependensi ini dalam file *package.json* sehingga dapat diandalkan untuk pengelolaan dependensi.
- NPM *Script*: NPM memungkinkan pengguna untuk mendefinisikan dan menjalankan skrip-skrip khusus dalam proyek melalui file *package.json*. Skrip-skrip ini dapat mencakup tugas-tugas seperti pengujian, pembuatan distribusi, dan lainnya.
- *Repository* Publik dan Pribadi: NPM menyediakan *repository* publik yang umumnya digunakan untuk paket-paket open source. Selain itu, NPM juga mendukung penggunaan *repository* pribadi untuk menyimpan paket-paket yang khusus untuk suatu organisasi atau proyek.

7. Express.js

Express.js adalah kerangka kerja (*framework*) web yang dibangun diatas Node.js yang berguna untuk memudahkan pembuatan aplikasi web. Express.js menyediakan seperangkat alat dan fitur yang memungkinkan pengembang untuk dengan cepat membuat aplikasi web dengan *JavaScript* di sisi *server*. Express.js

dirancang untuk menjadi ringkas dan efisien. Ini memungkinkan pengembang untuk dengan cepat membuat aplikasi web yang tangguh tanpa harus menangani banyak detail teknis. Express.js juga merupakan salah satu kerangka kerja web paling populer untuk Node.js. Kepopulerannya didukung oleh komunitas pengembang yang aktif, menyediakan berbagai sumber daya, tutorial, dan modul tambahan.

Salah satu keunggulan Express.js terletak pada kemampuannya untuk menyederhanakan penanganan rute (*routing*) dalam aplikasi. Rute-rute dapat dengan mudah ditentukan, memungkinkan pengembang menentukan *response* yang sesuai dengan permintaan yang masuk dengan jelas dan terstruktur. Integrasi dengan berbagai jenis middleware, memungkinkan eksekusi tugas tambahan di berbagai tahap proses *request-response*. Express.js juga menonjol dalam manajemen permintaan dan respons HTTP. Melalui API yang tersedia, pengembang dapat dengan mudah mengelola dan memanipulasi data permintaan serta merancang respons yang diberikan oleh server. Selain itu, kemampuannya dalam pengaturan server memberikan kontrol penuh terhadap konfigurasi dan penyesuaian lingkungan server aplikasi.

8. Python

Python adalah bahasa pemrograman tingkat tinggi yang populer, serbaguna, dan mudah dipelajari. diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python telah berkembang menjadi salah satu bahasa pemrograman yang paling banyak digunakan di dunia. Salah satu kelebihan utama Python adalah kesanggupannya menangani berbagai tipe pemrograman, mulai dari pengembangan web, ilmu data, kecerdasan buatan, hingga pengembangan aplikasi desktop.

Karena desainnya yang bersifat ekspresif, Python memungkinkan pengembang untuk menulis kode dengan lebih sedikit baris dibandingkan dengan bahasa pemrograman lainnya. Hal ini membuatnya ideal untuk proyek-proyek besar yang membutuhkan produktivitas tinggi.

Berikut beberapa fitur utama Python:

- Sintaks yang mudah dibaca: Python dirancang dengan sintaks yang mudah dibaca, hal ini membuat python menjadi pilihan yang baik untuk pemula.
- Interpretasi dan Kompilasi: Python dapat dijalankan sebagai bahasa yang diinterpretasikan atau dikompilasi. Ini memberikan fleksibilitas dalam pengembangan dan eksekusi kode.
- Tipe data dinamis: Python bersifat dinamis, yang berarti variabel tidak perlu dideklarasikan dengan tipe tertentu. Tipe data pada Python dapat berubah dinamis selama waktu eksekusi.
- Modularitas dan Pemrograman Berorientasi Objek: Python mendukung pemrograman berorientasi objek dan modularitas, memungkinkan pengembang untuk mengorganisir kode ke dalam objek dan modul yang dapat digunakan kembali.
- *Library* dan *Framework* yang beragam: Python memiliki berbagai perpustakaan dan *framework* yang beragam, seperti NumPy untuk komputasi numerik, Pandas untuk analisis data, Django untuk pengembangan web, dan TensorFlow untuk pembelajaran mesin
- *Cross-platform*: Python dapat dijalankan di berbagai sistem operasi termasuk Windows, macOS, dan Linux, sehingga memungkinkan portabilitas aplikasi.
- Implementasi yang luas: Python digunakan dalam berbagai bidang termasuk pengembangan web, ilmu data, kecerdasan buatan, pemrosesan bahasa alami, pengembangan game, dan masih banyak lagi
- *Open Source*: Python bersifat *open source*, yang berarti kode sumbernya dapat diakses dan dimodifikasi oleh siapa saja sesuai dengan lisensi yang diberikan.

9. Flask

Flask adalah sebuah *framework* web ringan yang ditulis dalam bahasa pemrograman Python. Diciptakan oleh Armin Ronacher, Flask dirancang untuk menjadi sederhana, fleksibel, dan mudah dipahami, memungkinkan pengembang untuk membuat aplikasi web dengan cepat dan efisien. Berikut adalah beberapa fitur Flask:

- *Microframework*: Flask adalah *microframework*, yang berarti flask menyediakan fungsionalitas dasar yang diperlukan untuk membangun aplikasi web, tetapi tetap sederhana dan fleksibel. Hal ini memungkinkan pengembang untuk memilih dan menggunakan ekstensi atau *library* eksternal sesuai kebutuhan.
- *Routing*: Flask menyediakan sistem *routing* yang mudah digunakan, memungkinkan pengembang untuk menentukan bagaimana URL permintaan dari klien akan dilayani oleh aplikasi.
- *Templating*: Flask mendukung *templating engine* Jinja2 yang memungkinkan pengembang membuat tampilan yang dinamis dan fleksibel untuk aplikasi web.
- *Werkzeug*: Flask menggunakan Werkzeug sebagai salah satu pilar utamanya. Werkzeug adalah toolkit WSGI (*Web Server Gateway Interface*) untuk mengelola permintaan HTTP dan tanggapan
- *Integrasi dengan Jinja2*: Flask terintegrasi secara erat dengan mesin template Jinja2, yang memungkinkan pengembang untuk membuat tampilan web yang dinamis dengan menggunakan sintaks template yang kuat.
- *RESTful*: Flask mendukung pembuatan aplikasi web RESTful, memungkinkan pengembang untuk merancang API yang bersih dan terstruktur.

10. Firebase

Firebase merupakan layanan dari Google yang menawarkan solusi komprehensif bagi pengembang aplikasi dengan tujuan mempermudah dan mempercepat proses pengembangan. Dikenal juga sebagai *Back-End as a Service* (BaaS), Firebase menyediakan berbagai fitur yang dapat digunakan pengembang untuk mengelola aspek *Back-End* dari aplikasi pengguna.

Firebase dimulai pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk pertama Firebase adalah Realtime Database, yang memungkinkan pengembang untuk menyimpan data dan melakukan sinkronisasi dengan banyak pengguna secara *real-time*. Seiring berjalannya waktu, Firebase berkembang

menjadi platform layanan pengembangan aplikasi. Pada Oktober 2014, Google mengakuisisi perusahaan tersebut.

11. Firebase Authentication

Firebase Authentication merupakan salah satu komponen utama dari Firebase, menawarkan lebih dari sekadar layanan autentikasi untuk aplikasi Android dan iOS yang dilengkapi dengan SDK yang *user-friendly* dan antarmuka siap pakai. Firebase Authentication memberikan pengalaman autentikasi yang andal dan terintegrasi dengan berbagai metode autentikasi. Mulai dari opsi nomor telepon hingga sandi, hingga dukungan penuh terhadap penyedia identitas populer seperti Google dan Facebook, layanan ini memberikan fleksibilitas kepada pengembang untuk memilih opsi autentikasi yang paling sesuai dengan kebutuhan aplikasi yang dibangun.

Sebagai bagian dari ekosistem Firebase, layanan ini terhubung secara sinergis dengan berbagai fitur lain yang ditawarkan oleh Firebase. Dengan demikian, pengguna dapat merasakan manfaat integrasi yang mulus antara autentikasi dan berbagai fungsi lain seperti penyimpanan data, analisis, dan cloud messaging. Dalam hal standar keamanan, Firebase Authentication mengandalkan kerangka kerja industri seperti OAuth 2.0 dan OpenID Connect, memudahkan pengembang untuk mengintegrasikan autentikasi Firebase dengan Back-End kustom yang dikelola. Sebagai solusi autentikasi yang holistik, Firebase Authentication bukan hanya sekadar pintu masuk ke aplikasi, melainkan fondasi yang kokoh untuk membangun pengalaman pengguna yang aman dan terpercaya.

12. Cloud Firestore

Cloud Firestore merupakan layanan basis data non-relasional (NoSQL) yang memiliki pondasi yang elastis dan dapat ditingkatkan untuk pengembangan aplikasi di berbagai perangkat, termasuk *mobile*, *web*, dan *server*. Firestore dapat diintegrasikan dalam ekosistem Firebase dan Google Cloud Platform. Keunggulan Cloud Firestore tidak hanya terletak pada struktur basis datanya, tetapi juga pada kemampuan untuk menjaga konsistensi data di seluruh aplikasi pengguna.

Satu fitur utama yang membedakan Cloud Firestore adalah kemampuannya dalam memberikan informasi secara *real-time*, memastikan setiap perubahan data langsung terdistribusi ke seluruh aplikasi tanpa *delay*. Sehingga pendistribusian dapat responsif dan dinamis, terutama ketika bekerja dengan aplikasi yang memerlukan pembaruan data secara cepat dan akurat.

Cloud Firestore juga menyediakan layanan dalam mode offline untuk aplikasi di perangkat seluler dan web. Ini berarti pengguna tetap dapat mengakses dan memanipulasi data bahkan tanpa koneksi internet, dan ketika kembali terhubung, perubahan yang dilakukan secara offline akan secara otomatis disinkronkan.

Secara keseluruhan, Cloud Firestore memenuhi kebutuhan pengembangan aplikasi modern dengan menyediakan basis data yang responsif, fleksibel, dan handal. Firestore dapat menjadi solusi yang tepat untuk membangun aplikasi yang membutuhkan konsistensi data dan pengalaman pengguna yang optimal di berbagai platform.

13. Google BigQuery

BigQuery merupakan salah satu layanan Google Cloud Platform yang merupakan sistem pengelolaan data warehouse. BigQuery sepenuhnya dirancang untuk membantu organisasi mengelola dan menganalisis data dengan efisiensi tinggi. Dengan berbagai fitur bawaan seperti *machine learning*, analisis *geospasial*, dan *business intelligence*, BigQuery menjadi sebuah platform yang komprehensif untuk mendukung kebutuhan analisis data yang kompleks.

Salah satu keunggulan utama BigQuery adalah arsitektur serverless-nya. Dengan pendekatan ini, pengguna dapat menggunakan query SQL tanpa perlu mengelola infrastruktur di belakangnya. Mesin analisis terdistribusi yang *scalable* dan terdistribusi pada BigQuery memungkinkan pembuatan *query* dengan ukuran *terabyte* dalam hitungan detik, sementara untuk data sebesar *petabyte*, hasil *query* dapat diperoleh dalam hitungan menit. Hal ini menciptakan lingkungan yang optimal untuk mengatasi beban analisis data berskala besar.

Antarmuka BigQuery meliputi antarmuka Google Cloud Console dan *command-line-tools* BigQuery. Developer dan data scientist dapat menggunakan library klien dengan bahasa pemrograman termasuk Python, Java, JavaScript, dan Go, serta REST API dan RPC API BigQuery untuk mengubah dan mengelola data. Driver ODBC dan JDBC menyediakan interaksi dengan aplikasi yang ada, termasuk alat dan utilitas pihak ketiga.

14. Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah *text editor open source* yang dikembangkan oleh Microsoft, dirancang untuk beroperasi pada komputer desktop dan kompatibel dengan sistem operasi Windows, macOS, dan Linux. Kelebihan utama dari VS Code adalah fitur-fitur unggulannya yang dirancang untuk meningkatkan efisiensi dan mempercepat proses penulisan kode. Sebagai text editor yang sangat fleksibel, Visual Studio Code mendukung berbagai bahasa pemrograman utama, seperti Java, C++, Python, HTML, CSS, JavaScript, Go, Rust, dan banyak lainnya. Kemampuannya untuk kompatibilitas dengan banyak bahasa pemrograman membuatnya menjadi pilihan yang sangat fleksibel untuk berbagai jenis proyek pengembangan perangkat lunak.

Salah satu fitur kunci dari VS Code adalah kemampuannya untuk mengakomodasi berbagai ekstensi. Pengguna dapat menambahkan ekstensi sesuai kebutuhan, termasuk Debuggers, Linters, dan dukungan untuk berbagai bahasa pemrograman dan framework. Dengan ekstensi ini, pengembang dapat mengkustomisasi pengalaman pengkodean sesuai dengan kebutuhan dan preferensi spesifik. Dengan antarmuka pengguna yang intuitif dan dukungan untuk berbagai fitur produktivitas, Visual Studio Code menjadi pilihan populer di kalangan pengembang perangkat lunak.

15. Git

Git adalah sistem kontrol versi (*version control system*) yang bersifat terdistribusi, dirancang untuk mengelola dan melacak perubahan dalam kode sumber selama pengembangan perangkat lunak. Git memungkinkan beberapa pengembang bekerja pada proyek yang sama secara bersamaan, melacak

perubahan, dan mengintegrasikan kode dengan lebih efisien. Sistem ini diciptakan oleh Linus Torvalds pada tahun 2005 dan sejak itu telah menjadi standar industri dalam manajemen kode sumber.

Git menggunakan model terdistribusi, yang berarti setiap pengembang memiliki salinan penuh dari seluruh proyek dan dapat bekerja secara mandiri tanpa koneksi internet. Ini memungkinkan tim pengembang untuk bekerja secara kolaboratif dan menggabungkan perubahan secara fleksibel. Beberapa fitur utama Git meliputi *branch*, *commit*, *merge*, dan *pull request*. *Branch* memungkinkan pengembang membuat cabang terpisah dari kode untuk bekerja pada fitur atau perbaikan bug tanpa mempengaruhi versi utama. *Commit* merekam perubahan kode, sementara *merge* menggabungkan cabang atau perubahan ke dalam versi utama. *Pull request* memungkinkan pengembang mengajukan perubahan untuk ditinjau dan disetujui oleh rekan tim sebelum disatukan ke dalam proyek.

16. GitHub

GitHub merupakan sebuah *platform* pengembangan perangkat lunak yang berbasis web, memainkan peran sentral dalam ekosistem pengembangan perangkat lunak modern. Didirikan oleh Tom Preston-Werner, Chris Wanstrath, dan PJ Hyett, GitHub menjadi tempat bagi banyak proyek *open source* dan perusahaan yang melibatkan jutaan pengembang di seluruh dunia.

Salah satu fitur utama GitHub adalah menyediakan *repository* yang memungkinkan pengguna menyimpan dan mengelola proyek perangkat lunak menggunakan sistem kontrol versi (*version control system*) Git. Git memungkinkan pengembang untuk melacak perubahan dalam kode sumber, bekerja sama secara efisien, dan mengelola versi dari perangkat lunak yang dikembangkan.

GitHub menyediakan *repository* untuk menyimpan dan mengelola proyek perangkat lunak, serta berbagai fitur kolaborasi untuk tim *developer*. Berikut beberapa fitur utama GitHub:

- **Manajemen Source Code:** GitHub memungkinkan pengguna membuat *repository* (repo) untuk menyimpan dan mengelola kode sebuah proyek.

GitHub menggunakan Git sebagai *version control system* untuk melacak perubahan dalam kode.

- Kolaborasi Tim: Platform ini dirancang untuk mendukung kolaborasi tim dalam pengembangan perangkat lunak. Tim dapat berkontribusi pada proyek, membuat cabang (*branch*), dan menggabungkan perubahan (*merge*) dengan mudah.
- Issue Tracking: GitHub menyediakan fitur pelacakan isu (issue tracking) untuk memudahkan diskusi, pemantauan, dan penyelesaian masalah atau peningkatan dalam proyek.
- Pull Requests: Pull request adalah cara untuk mengusulkan perubahan kode ke *repository* utama. Hal ini memfasilitasi diskusi, tinjauan kode, dan integrasi perubahan ke dalam kode utama.
- Wiki dan Dokumentasi: Tim pengembang dapat membuat halaman Wiki untuk dokumentasi proyek dan menyertakan panduan penggunaan, informasi kontribusi, dan detail teknis lainnya.
- GitHub Actions: GitHub Actions adalah layanan otomatisasi yang memungkinkan pengguna membuat dan menjalankan alur kerja otomatis, seperti uji coba (testing) atau penyebaran (deployment), langsung di GitHub.
- Gists: Gists adalah cara cepat untuk berbagi potongan kode atau teks dengan orang lain. Gists dapat digunakan untuk membagikan contoh kode atau bahkan menyusun catatan sementara.
- GitHub Pages: GitHub Pages memungkinkan pengguna untuk menerbitkan situs web statis langsung dari *repository* GitHub. Ini adalah cara sederhana untuk membuat halaman web proyek atau dokumentasi.
- Lisensi *Open Source*: Banyak proyek di GitHub adalah proyek *open source*, yang berarti kode sumbernya dapat diakses, digunakan, dan dimodifikasi oleh siapa saja sesuai dengan lisensi yang diberikan oleh pengembang.

17. Postman

Postman adalah sebuah platform kolaboratif yang menyediakan alat untuk menguji, mengelola, dan mengembangkan API (*Application Programming Interface*). Postman diciptakan oleh Postdot Technologies, Postman menyediakan

berbagai fitur yang mempermudah pengembangan dan pengujian API, baik untuk pengembang perangkat lunak maupun tim pengembangan. Berikut adalah beberapa fitur utama Postman:

- Pengujian API: Postman memungkinkan pengguna untuk membuat dan mengirim permintaan HTTP ke berbagai endpoint API, menguji *response*, dan memvalidasi apakah API berfungsi sesuai yang diharapkan.
- *Request Collection*: Pengguna dapat mengelompokkan setiap request API ke dalam sebuah *collection*, memudahkan pengorganisasian dan pengelolaan permintaan terkait.
- *Shared Collection*: Pengguna dapat berbagi *request collection* dengan tim atau komunitas melalui sebuah tautan, memungkinkan tim dapat berkolaborasi menggunakan *collection* yang sama.
- Environment Variables: Fitur ini memungkinkan pengguna untuk menyimpan nilai-nilai yang dapat digunakan secara berulang, seperti URL atau token otentikasi, dalam suatu lingkungan. Ini membantu pengujian API di berbagai lingkungan (misalnya *development* dan *production*).
- Otomatisasi dan Scripting: Postman menyediakan otomatisasi dengan menulis skrip menggunakan JavaScript.
- Pengelolaan API: Postman menyertakan fitur untuk membuat, memodifikasi, dan membagikan dokumentasi API. Hal ini dapat membantu pengguna dalam merinci dan menjelaskan fungsi-fungsi API secara terperinci.

B. *Hardware*

Hardware atau perangkat keras yang digunakan dalam mendukung pelaksanaan kerja praktik antara lain:

1. Laptop

Laptop digunakan sebagai alat utama untuk menghadiri pembelajaran, mengerjakan assignment dari setiap kelas, serta penggerjaan *capstone project*. Laptop menjadi platform utama bagi untuk mengeksplorasi dan mengembangkan solusi yang diperlukan dalam konteks kerja praktik.

2. Perangkat *Mobile* Android

Perangkat *mobile* berbasis Android juga menjadi komponen penting dalam kerja praktik ini. Perangkat ini digunakan untuk menjalankan aplikasi hasil dari *capstone project*, memungkinkan untuk menguji dan mengevaluasi langsung performa aplikasi pada platform yang relevan, sehingga memastikan kesesuaian dan kualitas aplikasi dengan lingkungan pengguna akhir.

BAB IV

PELAKSANAAN KERJA PRAKTIK

IV.1 Input

Pada desain arsitektur server *Google Cloud* untuk aplikasi *Daily Cloud*, terdapat beberapa tahapan dalam pembangunan arsitektur *Google Cloud* diantaranya:

- a. Perancangan Arsitektur Cloud
- b. Autentikasi dengan *Firebase Authentication*
- c. Database *Firestore*
- d. Server dengan *Google Cloud Run*
- e. Back-End API dengan *Node.js* dan *Express.js*
- f. Prediction API dengan *Python* dan *Flask*
- g. Pengolahan data dengan Google BigQuery

1. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk desain arsitektur server *Google Cloud* untuk aplikasi *Daily Cloud* di Google Indonesia, penulis menggunakan laptop dengan spesifikasi:

No	Item	Spesifikasi
1	<i>Processor</i>	Kecepatan 3.10GHz
2	<i>Hard Disk</i>	500 GB
3	<i>Memory</i>	8 GB
4	VGA	1 GB
5	<i>Monitor</i>	Resolusi 1920x1080

Tabel IV.1 Kebutuhan Perangkat Keras

2. Minimum Kebutuhan Perangkat Keras

Minimum spesifikasi laptop atau komputer yang harus digunakan agar dapat melakukan desain arsitektur server *Google Cloud* untuk aplikasi *Daily Cloud* di Google Indonesia adalah sebagai berikut:

No	Item	Spesifikasi
1	<i>Processor</i>	Kecepatan 2.10GHz
2	<i>Hard Disk</i>	500 GB
3	<i>Memory</i>	4 GB
4	VGA	256 MB
5	<i>Monitor</i>	Resolusi 1366x768

Tabel IV.2 Minimum Kebutuhan Perangkat Keras

3. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam melakukan desain arsitektur server *Google Cloud* untuk aplikasi *Daily Cloud* di Google Indonesia adalah sebagai berikut:

No	Item	Spesifikasi
1	Sistem Operasi	<i>Windows 10 Pro 64-bit</i>
2	<i>Web Browser</i>	<i>Google Chrome</i>
3	<i>Code Editor</i>	<i>Visual Studio Code</i>
4	Bahasa Pemrograman	<i>JavaScript dan Python</i>

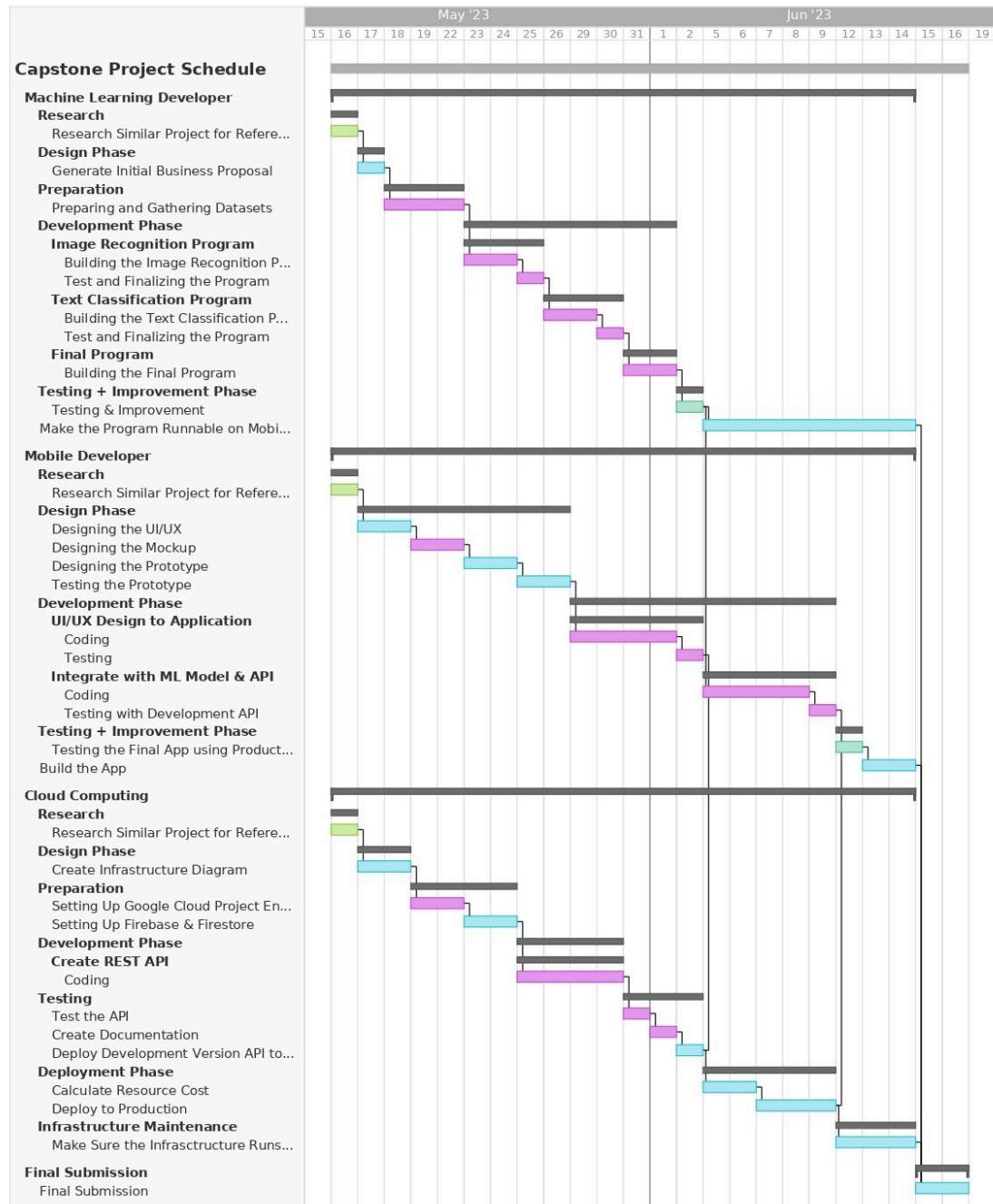
Tabel IV.3 Kebutuhan Perangkat Lunak

IV.2 Proses

Pada pengembangan aplikasi Daily Cloud oleh tim C23-PS277, tim *Cloud Computing* memiliki peran penting yang mencakup berbagai aspek yaitu membuat REST API untuk mengelola data pengguna, mengolah data jurnal yang telah ditulis oleh pengguna pada sisi *client*, melakukan prediksi dari jurnal ke Model *Machine Learning* untuk mendapatkan prediksi, dan mengelola artikel yang akan digunakan oleh aplikasi. Selain itu, tim *Cloud Computing* juga bertanggung jawab dalam pembuatan arsitektur *google cloud* yang akan digunakan sebagai *server* dari proyek Daily Cloud, yang diantaranya pembuatan *server Cloud Run*, Firestore sebagai *database*, autentikasi dengan *Firebase Authentication*, *Google*

Container Registry untuk mengelola *docker images*, dan *Google BigQuery* untuk pengolahan data jurnal yang akan digunakan sebagai dataset.

IV.2.1 Pembuatan Jadwal Pengerjaan Proyek



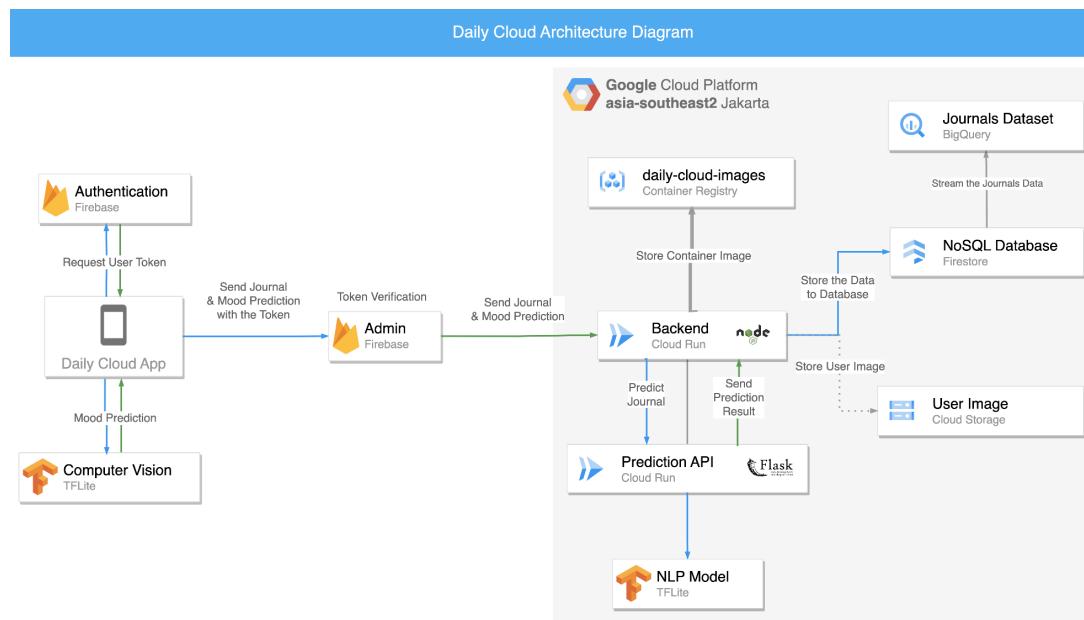
Gambar IV.1 Gantt Chart Project Timeline

Tahap pertama adalah menentukan jadwal pengerjaan Capstone Project menggunakan *Gantt Chart* seperti pada Gambar IV.1 untuk merencanakan dan

memvisualisasikan waktu penggeraan setiap tugas atau fase proyek. Dengan tim yang terdiri dari 6 orang, penting untuk menyusun jadwal yang efisien dan memanfaatkan keahlian dan ketersediaan masing-masing anggota tim. Tugas tim *Cloud Computing* bertugas untuk *Research* arsitektur cloud, membuat diagram arsitektur *cloud*, melakukan persiapan resource pada *project Google Cloud*, mengembangkan *Back-End API*, melakukan *testing*, *deployment API* ke *server*, serta melakukan *maintenance* arsitektur cloud.

IV.2.2 Perancangan Arsitektur *Cloud*

Tahap selanjutnya dalam penggeraan *Capstone Project* khususnya pada *Learning Path Cloud Computing*, adalah perancangan arsitektur cloud. Proses ini melibatkan pembuatan sebuah diagram arsitektur *cloud* yang merinci komponen-komponen utama, layanan, dan interaksi di dalam lingkungan *cloud* yang akan digunakan dalam proyek.



Gambar IV.2 Diagram Perancangan Arsitektur *Cloud*

Seperti yang ditunjukkan dalam Arsitektur Google Cloud pada Gambar IV.1, pada Arsitektur Google Cloud menunjukkan penggunaan beberapa sumber daya Google Cloud Platform yang secara integral terlibat dalam pembangunan

arsitektur cloud untuk aplikasi Daily Cloud. Beberapa sumber daya kunci yang digunakan adalah sebagai berikut:

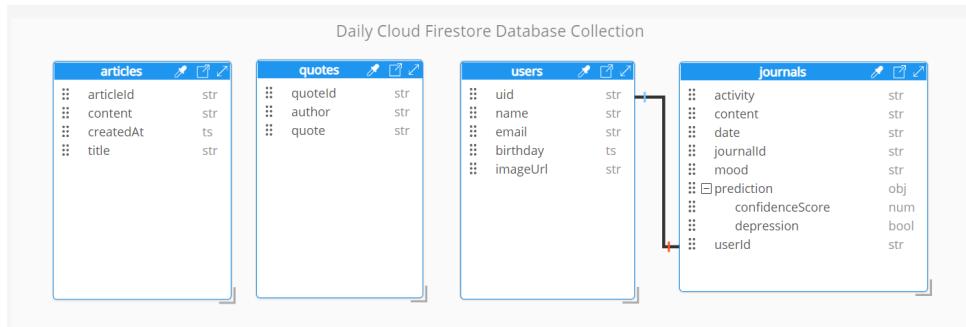
1. Google Cloud Run

Google Cloud Run digunakan sebagai platform serverless untuk REST API. Cloud Run memungkinkan pengembang menjalankan kontainer Docker secara otomatis dan menyediakan layanan yang dapat diakses melalui permintaan HTTP. Ini memberikan skalabilitas otomatis dan tidak memerlukan manajemen infrastruktur.

2. Google Cloud Build dan Container Registry.

Google Cloud Build dan Container Registry, Cloud Build digunakan sebagai alat untuk membangun docker images dan menyimpannya ke dalam *Container Registry*. Sedangkan Container Registry berfungsi sebagai *repository* penyimpanan untuk menyimpan dan mengelola *container* Docker.

3. Firestore



Gambar IV.3 Firestore Database Collection

Firestore digunakan sebagai penyimpanan basis data untuk keperluan aplikasi, termasuk data jurnal, akun pengguna, artikel, dan kutipan hari ini (quotes of the day). Firestore adalah basis data dokumen yang menyediakan skema fleksibel dan dapat diakses melalui API. Untuk skema *collection* pada *database* Firestore dapat dilihat pada Gambar IV.2.

4. BigQuery

BigQuery digunakan sebagai layanan pengolahan data jurnal dan berperan sebagai *Training Data* untuk *Machine Learning*. BigQuery adalah

layanan penyimpanan dan analisis data yang dapat menangani kuantitas data yang besar dan kompleks dengan cepat. Ini memungkinkan analisis SQL interaktif dan pemrosesan data paralel.

IV.2.3 Perancangan *Back-End*

Pada perancangan *Back-End* aplikasi *Daily Cloud*, terdapat dua *Rest API* yang berperan krusial dalam mendukung fungsionalitas aplikasi, yaitu *Main API* dan *Prediction API*. Pada *Main API*, bahasa pemrograman *JavaScript* digunakan dengan memanfaatkan *framework ExpressJS*, sementara pada *Prediction API*, bahasa pemrograman yang diterapkan adalah *Python* dengan dukungan dari *framework Flask*.

1. Main API

Pengembangan *Back-End* aplikasi *Daily Cloud*, bahasa pemrograman yang dipilih untuk *Main API* adalah *JavaScript*, dengan menggunakan platform *Node.js*, serta memanfaatkan *framework ExpressJS*. *Framework ExpressJS*, sebagai salah satu *framework* terkemuka untuk *Node.js*, membantu mempercepat proses pengembangan dan menyediakan struktur yang terorganisir.

Main API memiliki beragam fungsi utama yang membentuk inti fungsionalitas aplikasi. Pertama, layanan autentikasi seperti *login*, *register*, dan verifikasi *token* menjadi fitur utama dalam menyediakan pengalaman pengguna yang aman dan terpercaya. Selanjutnya, API ini memberikan layanan untuk menyimpan dan mendapatkan jurnal pengguna, memungkinkan pengguna untuk mencatat dan mengakses riwayat aktivitas harian mereka. Tidak hanya itu, *Main API* juga menawarkan layanan untuk mendapatkan artikel, memperkaya konten aplikasi dengan informasi tambahan yang relevan. Terakhir, API ini menyediakan layanan untuk mendapatkan *quotes of the day*, memberikan inspirasi dan motivasi kepada pengguna.

2. Prediction API

Pada pengembangan *Prediction API*, bahasa pemrograman yang diadopsi adalah *Python*, dengan dukungan dari *framework Flask*. Python

dipilih karena keandalannya dalam pemrosesan data dan integrasi yang baik dengan berbagai *library machine learning*. *Framework Flask*, yang ringan dan dapat diimplementasikan dengan mudah, memastikan pengembangan API yang efisien dan *scalable*.

Fungsi utama *Prediction API* melibatkan integrasi *Back-End* dengan model *machine learning*. API ini menggunakan *TensorFlow*, salah satu *library machine learning* yang sangat populer, untuk melakukan prediksi kondisi pengguna berdasarkan jurnal yang telah ditulis. Proses ini memberikan dimensi tambahan pada aplikasi, memungkinkan pengguna untuk menerima rekomendasi atau analisis berdasarkan data yang mereka bagikan.

Adapun beberapa alat (tools) penting turut berperan dalam pengembangan aplikasi ini. GitHub berperan sebagai remote repository untuk menyimpan kode program, Visual Studio Code digunakan sebagai teks editor dalam proses pengembangan, dan Postman menjadi alat pengujian keseluruhan API. Seluruh kode program pengembangan aplikasi Daily Cloud tersedia untuk akses melalui GitHub Repository, yang dapat diakses pada Lampiran C.1. *Repository GitHub Daily Cloud* juga dapat diakses secara langsung melalui link <https://github.com/daily-cloud>, menciptakan transparansi dan keterbukaan dalam proses pengembangan aplikasi ini. Dengan demikian, perancangan Back-End tidak hanya mencakup struktur teknis, tetapi juga memastikan bahwa seluruh tim pengembang dapat bekerja secara kolaboratif dengan efisien, transparan, dan terorganisir.

IV.3 Pencapaian Hasil

Hasil dari proyek *Product-Based Capstone* dari tim C23-PS277 adalah sebuah aplikasi bernama Daily Cloud yang telah berhasil dikembangkan terlampir sebuah *banner* aplikasi sebagai cover branding pada Lampiran C.2. Aplikasi ini memiliki fitur utama yaitu memprediksi kesehatan mental pengguna berdasarkan jurnal dan ekspresi wajah pengguna yang telah dijelaskan sebelumnya dalam ruang lingkup proyek. Berikut adalah beberapa pencapaian hasil dari tim *Cloud*

Computing sebagai *Cloud Engineer* dan *Back-End Developer* dari tim C23-PS277:

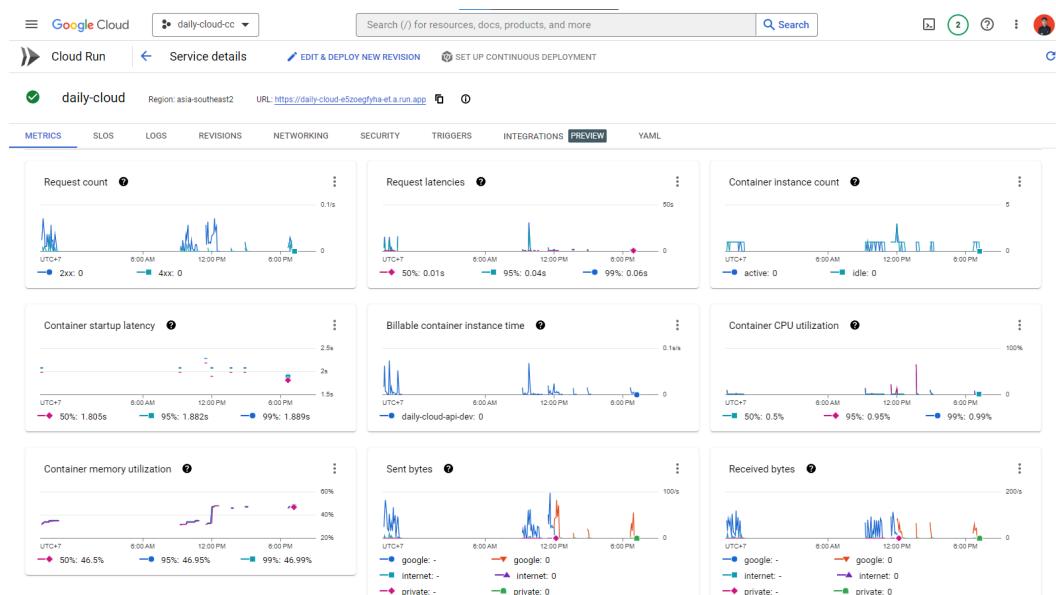
1. Cloud Computing

Pengembangan Arsitektur Aplikasi Daily Cloud mencakup beberapa aspek penting dalam arsitektur *cloud* yang berhasil dibangun menggunakan Google Cloud Platform (GCP). Berikut adalah beberapa poin utama pada hasil pengembangan *server cloud* pada GCP:

1) Arsitektur *Cloud* di Google Cloud Platform (GCP):

Arsitektur *cloud* telah berhasil dibangun dengan menggunakan Google Cloud Platform. GCP menyediakan beragam layanan dan sumber daya cloud yang mendukung kebutuhan aplikasi Daily Cloud, mulai dari *server*, penyimpanan data, *monitoring* kinerja *server*, serta melakukan analisis pada data yang dikirimkan oleh *client*.

2) Google Cloud Run sebagai *Platform Serverless*



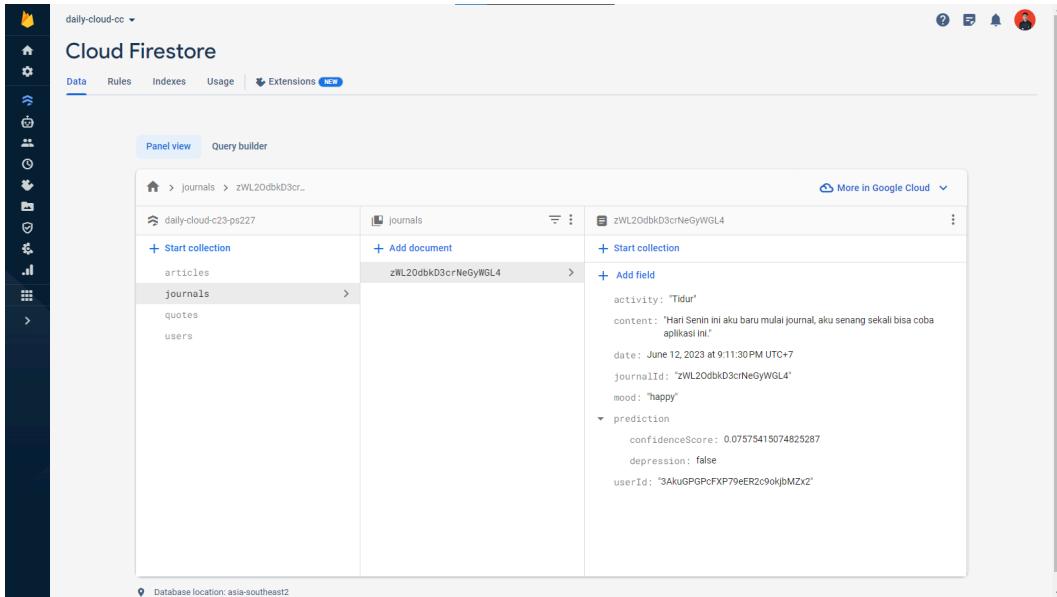
Gambar IV.4 Statistik Server pada Resource Google Cloud Run

Google Cloud Run digunakan sebagai platform serverless untuk melakukan *deployment* REST API dari *Back-End*. Cloud Run memungkinkan

aplikasi beradaptasi secara dinamis dengan permintaan pengguna, sambil memberikan fleksibilitas dan skalabilitas tinggi.

Seperti pada Gambar IV.4 yang menampilkan statistik Google Cloud Run, yang mencakup kinerja, jumlah instance, waktu response, dan parameter lainnya, menjadi kunci dalam pemantauan dan optimalisasi server.

3) Firestore sebagai Layanan Database



The screenshot shows the Google Cloud Platform Cloud Firestore interface. On the left, there's a sidebar with icons for Home, Project Overview, Cloud Functions, Cloud Storage, and Cloud Firestore. The main area has tabs for Data, Rules, Indexes, Usage, and Extensions. The Data tab is selected, showing a hierarchical view of collections: 'journals' under 'daily-cloud-c23-ps227'. A specific document, 'zWL20dbkD3crNeGyWGL4', is selected and expanded. The document contains fields: 'activity' (Tidur), 'content' (Hari Senin ini aku baru mulai jurnal, aku senang sekali bisa coba aplikasi ini.), 'date' (June 12, 2023 at 9:11:30 PM UTC-7), 'journalId' (zWL20dbkD3crNeGyWGL4), 'mood' (happy), 'prediction' (confidenceScore: 0.07575415074825287, depression: false), and 'userId' (3AkU6GPGPcXP79eER2c9okjbMzx2). There are also buttons for '+ Start collection', '+ Add document', and '+ Add field'.

Gambar IV.5 Database Firestore

Firestore digunakan sebagai layanan database dari aplikasi Daily Cloud. Ini mencakup penyimpanan data pengguna, jurnal, artikel, dan kutipan dengan cara yang terstruktur, skalabel, dan aman. Pada Lampiran C.5 terdapat beberapa data yang dalam *database* Firestore pada masa pengembangan aplikasi Daily Cloud.

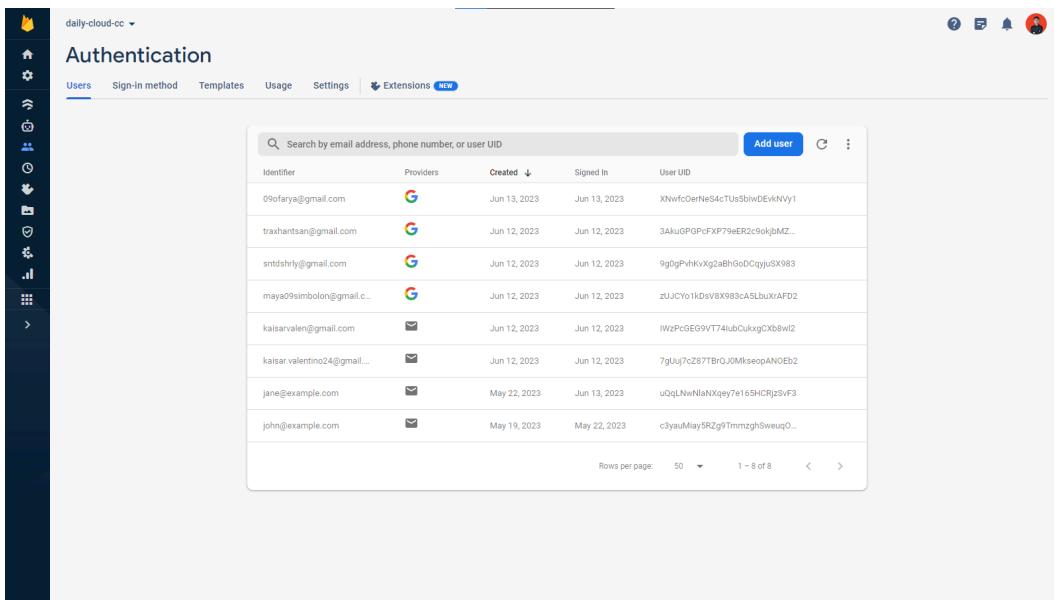
4) BigQuery untuk Pengolahan Data dan Analitik

BigQuery digunakan untuk menganalisis data jurnal pengguna. Data ini dapat diolah dan disusun menjadi dataset yang mencakup berbagai aspek dari aktivitas harian pengguna. Proses analisis data di BigQuery memberikan kesempatan untuk memahami pola perilaku pengguna, mengeksplorasi tren,

dan mendapatkan wawasan yang mendalam tentang preferensi atau kebutuhan pengguna.

Dataset hasil analisis di BigQuery kemudian menjadi sumber informasi yang berharga bagi aplikasi Daily Cloud. Data ini dapat dimanfaatkan untuk memberikan rekomendasi yang lebih personal, merancang fitur yang lebih sesuai dengan kebutuhan pengguna, atau bahkan untuk keperluan analisis bisnis dan pengambilan keputusan strategis.

5) Firebase Authentication sebagai layanan autentikasi



The screenshot shows the Firebase Authentication interface for a project named 'daily-cloud-cc'. The 'Users' tab is selected. A search bar at the top allows searching by email address, phone number, or user UID. Below the search bar is a table with columns: Identifier, Providers, Created, Signed In, and User UID. The table lists eight users, each with their email, provider (Google), creation date (Jun 13, 2023 for the first user), sign-in date (Jun 13, 2023 for the first user), and a long User UID. The interface includes standard navigation buttons like 'Add user' and a 'Rows per page' dropdown set to 50.

Identifier	Providers	Created	Signed In	User UID
09ofarya@gmail.com		Jun 13, 2023	Jun 13, 2023	XNivfcOerNe54cTUs5biwDEvkNVy1
traxhantsan@gmail.com		Jun 12, 2023	Jun 12, 2023	3AkluPGPcXP79eER2c9okjbMZ...
sntdshly@gmail.com		Jun 12, 2023	Jun 12, 2023	9g0gPvhKXg2aBhGoDcqjuSX983
maya09simbolon@gmail.c...		Jun 12, 2023	Jun 12, 2023	zUJCYo1kDsVBX983cA5LbuXIAFD2
kaisarvalen@gmail.com		Jun 12, 2023	Jun 12, 2023	lWzPcGE9VT74lubCukxgCxb8wl2
kaisar.valentino24@gmail.c...		Jun 12, 2023	Jun 12, 2023	7g0uj7cZ87TBQJ0MjkeepaANOEBz
jane@example.com		May 22, 2023	Jun 13, 2023	uQqLNvNlaNXqey7e165ICRjzSVF3
john@example.com		May 19, 2023	May 22, 2023	c3yauMiy5RZg9TmmzghSweuq0...

Gambar IV.6 Firebase Authentication

Firebase Authentication diterapkan sebagai layanan keamanan pengguna. Ini memungkinkan implementasi sistem login, registrasi, dan verifikasi token dengan mudah, menjaga keamanan dan integritas informasi pengguna yang diakses oleh aplikasi. Pada Lampiran C.6 menunjukkan beberapa *user* yang telah terdaftar selama masa pengembangan.

2. Back-End

Pengembangan Back-End Aplikasi Daily Cloud mencakup beberapa elemen kunci yang berhasil diimplementasikan. Berikut adalah beberapa poin penting dalam pengembangan Back-End pada aplikasi Daily Cloud:

1) Main API

Main API telah berhasil dibuat sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan sebelumnya. API ini mencakup beberapa fungsi inti, termasuk User API untuk manajemen pengguna, Journal API untuk menyimpan dan mengambil jurnal pengguna, Article API untuk mengakses artikel, dan Quotes of The Day API untuk memberikan kutipan motivasi harian. Berikut adalah beberapa kode program untuk Main API.

a. *Package.json* sebagai identitas program

```
{
  "name": "daily-cloud-be",
  "version": "0.5.1",
  "description": "Application Programming Interface (API) used by the Daily Cloud Mobile App",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "node .",
    "dev": "nodemon ."
  },
  "keywords": [
    "api"
  ],
  "author": "daily-cloud",
  "repository": {
    "type": "git",
    "url": "git+https://github.com/daily-cloud/daily-cloud-be.git"
  },
  "license": "ISC",
  "dependencies": {
    "@google-cloud/storage": "^6.10.1",
    "cors": "^2.8.5",
    "dotenv": "^16.0.3",
    "express": "^4.18.2",
    "firebase-admin": "^11.8.0",
    "multer": "1.4.5-lts.1"
  },
  "devDependencies": {
    "eslint": "^8.40.0",
    "nodemon": "^2.0.22"
  }
}
```

File package.json berfungsi sebagai identitas program, menyediakan informasi penting seperti nama program, versi, deskripsi, skrip yang dapat dijalankan, dan dependensi yang dibutuhkan. Beberapa poin penting dari package.json ini mencakup:

- Dependencies: Mendefinisikan paket-paket atau *library* yang dibutuhkan oleh program.
- Scripts: Menentukan skrip-skrip yang dapat dijalankan dengan perintah npm. Skrip "start" digunakan untuk menjalankan aplikasi, sedangkan "dev" menggunakan nodemon untuk memonitor perubahan pada kode dan secara otomatis *reload* server.

b. *Register User*

```
async signUpUser() {
  const birthday = Timestamp.fromDate(new
Date(this.data.birthday));
  const user = {
    uid: this.data.uid,
    email: this.data.email,
    name: this.data.name,
    birthday,
    imageUrl: this.data.imageUrl || defaultUserImageUrl,
  };
  const userRef =
  firestore.collection('users').doc(this.data.uid);
  await userRef.set(user);
  return userRef.id;
}
```

Fungsi *signUpUser* digunakan untuk mendaftarkan pengguna baru ke aplikasi. Data seperti email, nama, dan tanggal lahir diambil dari input dan disimpan dalam database Firestore.

c. Verifikasi Token User ketika Login

```
async function verifyToken(req, res, next) {
  const authorizationHeader = req.headers.authorization;
  if (!authorizationHeader) {
    res.status(401);
    res.send({
      message: 'Unauthenticated',
    });
  }
}
```

```

        error: 'Access Denied. No token provided',
    );
    return;
}
const userToken = authorizationHeader.split(' ')[1];
try {
    const decodedToken = await
admin.auth().verifyIdToken(userToken);
    req.user = decodedToken;
    next();
} catch (err) {
    console.error(`Token verification failed : ${err}`);
    res.status(401);
    res.send({ message: 'Unauthenticated', error: 'Token
is not valid' });
}
}

```

Fungsi *verifyToken* berperan sebagai middleware untuk memastikan bahwa token yang dikirim oleh pengguna saat login valid. Hal ini dilakukan dengan memverifikasi token menggunakan Firebase Admin SDK.

d. Melihat detail *user*

```

async getUserDetails() {
    try {
        const userRef =
firestore.collection('users').doc(this.data.uid);
        const snapshot = await userRef.get();
        const user = snapshot.data();
        return user;
    } catch (err) {
        throw new Error(`Failed to get user details:
${err}`);
    }
}

```

Fungsi *getUserDetails* memiliki tujuan utama untuk mengambil dan menyajikan informasi terperinci mengenai seorang pengguna berdasarkan ID pengguna tertentu. Fungsi ini mengakses database Firestore, yang merupakan penyimpanan cloud dari Firebase, untuk mengambil data yang diperlukan.

e. Melakukan *update* detail *user*

```
async updateUserDetails() {
  try {
    const userRef =
      firestore.collection('users').doc(this.data.uid);
    const snapshot = await userRef.get();
    const user = snapshot.data();
    const name = this.data.name;
    const updatedData = {
      ...user, // keep the old data
      name, // update the new data
    };
    await userRef.set(updatedData);
    return this.getUserDetails();
  } catch (err) {
    throw new Error(`Failed to update user details:
${err}`);
  }
}
```

Fungsi *updateUserDetails* memperbarui detail pengguna seperti nama. Fungsi ini mengambil data pengguna yang sudah ada, memperbarui data yang diperlukan, dan menyimpannya kembali ke Firestore.

f. Menambah jurnal

```
async addNewJournal(data) {
  const newJournalRef = await this.journalsRef.doc();
  const todayTimestamp = Date.now();
  // using local date (Jakarta)
  const date = Timestamp.fromMillis(todayTimestamp);
  // Predict journal content from cloud
  const prediction = await
  fetchPrediction(data.content);
  const newJournal = {
    ...data,
    date,
    journalId: newJournalRef.id,
    prediction,
  };

  await newJournalRef.set(newJournal);
  return newJournalRef.id;
}
```

Fungsi *addNewJournal* digunakan untuk menambahkan jurnal baru ke database Firestore. Selain menyimpan konten jurnal, fungsi ini juga mengambil prediksi dari layanan eksternal yaitu *Prediction API* berdasarkan konten jurnal yang ditulis.

g. Melihat semua jurnal

```
async getAllJournals(uid, filter) {
  let snapshot = this.journalsRef;
  const { month, year } = filter;
  const journals = [];
  // initial query to get all journals filter by user id
  snapshot = await this.journalsRef.where('userId',
  '==', uid);
  // get journal filter by month and year
  if (month && year) {
    // month is 0 based
    const startTimestamp = new Date(year, month - 1, 1);
    // 1st day of month
    const endTimestamp = new Date(year, month, 1); // 1st day of next month
    // filter by >= startTimestamp and <= endTimestamp
    snapshot = await snapshot
      .where('date', '>=',
      Timestamp.fromDate(startTimestamp))
      .where('date', '<=',
      Timestamp.fromDate(endTimestamp))
      .get();
  } else {
    snapshot = await snapshot.get();
  }
  // push the journals to journals array
  snapshot.forEach((doc) => {
    journals.push(doc.data());
  });
  return journals;
}
```

Fungsi *getAllJournals* digunakan untuk mengambil semua jurnal pengguna berdasarkan filter tertentu, seperti bulan dan tahun. Hasilnya akan digunakan untuk menampilkan riwayat jurnal pengguna sesuai dengan kriteria yang dibutuhkan. Dengan menggunakan filter bulan dan tahun, pengguna dapat dengan mudah melihat *progress* atau perubahan dalam pemikiran dan pengalaman mereka dari waktu ke waktu.

h. Melihat jurnal berdasarkan ID

```
async getJournalById(journalId) {
  const snapshot = await
this.journalsRef.doc(journalId);
  const doc = await snapshot.get();
  if (!doc.exists) {
    return null;
  }
  return doc.data();
}
```

Fungsi *getJournalById* memungkinkan aplikasi untuk mengambil detail jurnal berdasarkan ID tertentu dari *database Firestore*.

i. Melakukan *check* jurnal hari ini

```
async checkJournalToday(uid) {
  const date = new Date();
  const startTimestamp = date.setHours(0, 0, 0, 0); // set time 00:00:00
  const endTimestamp = date.setHours(23, 59, 59, 999); // set time 23:59:59
  const snapshot = await this.journalsRef
    .where('userId', '==', uid)
    .where('date', '>=',
  Timestamp.fromMillis(startTimestamp))
    .where('date', '<=',
  Timestamp.fromMillis(endTimestamp))
    .get();
  const journal = [];
  const result = {};
  if (snapshot.empty) {
    result.status = false;
    result.journal = null;
  } else {
    snapshot.forEach((doc) => {
      journal.push(doc.data());
    });
    result.status = true;
    result.journal = journal[0];
  }
  return result;
}
```

Fungsi *checkJournalToday* digunakan untuk memeriksa apakah pengguna telah membuat jurnal pada hari yang sama. Hasilnya memberikan status apakah jurnal sudah dibuat hari ini atau belum.

j. Kode *Dockerfile* untuk deploy ke *Google Cloud Run*

```
FROM node:lts-alpine
WORKDIR /usr/src/app
COPY package*.json .
RUN npm install
COPY . .
# EXPOSE 8080
CMD [ "node", "index" ]
```

File *Dockerfile* digunakan untuk mengkonfigurasi *container Docker* yang akan digunakan untuk menjalankan aplikasi di *Google Cloud Run*. Pada contoh tersebut, *container* dibangun menggunakan *node:lts-alpine* sebagai *base image*, mengatur direktori kerja, menyalin file *package.json*, melakukan instalasi dependensi dengan *npm install*, menyalin seluruh proyek, dan menjalankan aplikasi dengan perintah "*node index*". File *Dockerfile* ini penting untuk memastikan aplikasi dapat di-deploy dengan benar di lingkungan *Google Cloud Run*.

2) *Prediction API*

Prediction API merupakan elemen penting dalam Back-End, berfungsi sebagai penghubung antara Main API dengan model machine learning. API ini dirancang khusus untuk memprediksi kesehatan mental pengguna berdasarkan data jurnal yang dikirimkan. Berikut adalah beberapa kode program untuk *Prediction API*.

a. *Load Machine Learning Model*

```
def load_model():
    interpreter =
        lite.Interpreter(model_path=path.abspath("app/keras_model/
        /model_nlp.tflite"))
    interpreter.allocate_tensors()
    return interpreter
```

Fungsi *load_model* digunakan untuk memuat model machine learning yang akan digunakan dalam prediksi kesehatan mental. Pada contoh ini, model *machine learning* disimpan dalam format *TensorFlow Lite* (.tflite). *Interpreter TensorFlow Lite* digunakan untuk memproses model dan mengalokasikan tensor yang diperlukan.

b. Prediksi ke *Machine Learning Model*

```

def preprocess_text(text):
    text = word_tokenize(text.lower())
    text = [t for t in text if t not in stop_words]
    text = [stemmer.stem(t) for t in text]
    text = [t if not t.isdigit() else num2words(int(t))
for t in text]
    text = " ".join(text)
    return text

def predict_list_text(list_text):
    interpreter = load_model()
    data = pd.DataFrame(list_text, columns=["text"])
    data["nlp_text"] = data["text"].apply(lambda x:
preprocess_text(x))
    one_hot_representation = [one_hot(words, 14233) for
words in data["nlp_text"]]
    embedded_docs = pad_sequences(
        one_hot_representation, padding="post",
        truncating="post", maxlen=1366
    )
    embedded_docs_float32 = np.array(embedded_docs,
dtype=np.float32)
    input_details = interpreter.get_input_details()
    result = []
    for doc in embedded_docs_float32:
        input_data = np.expand_dims(
            doc, axis=0
        )
        interpreter.set_tensor(input_details[0]["index"],
input_data)
        interpreter.invoke()
        output_details = interpreter.get_output_details()
        output_tensor =
interpreter.get_tensor(output_details[0]["index"])
        result.append(output_tensor[0].tolist())
    return result

def predict_depression(text):

```

```

list_text = [text]
confidence_score = predict_list_text(list_text)[0][0]
depression = confidence_score >= 0.5
result = {
    "depression": depression,
    "confidence_score": confidence_score,
}
return result

```

- Fungsi *preprocess_text* melakukan pra-pemrosesan pada teks yang akan diprediksi. Proses ini mencakup tokenisasi kata, penghapusan kata stop (stop words), stemming, dan mengubah angka menjadi representasi kata.
- Fungsi *predict_list_text* melakukan prediksi kesehatan mental untuk daftar teks yang diberikan. Pra-pemrosesan dilakukan pada setiap teks, kemudian teks diubah menjadi representasi one-hot dan di-pad sesuai dengan panjang maksimum. Proses prediksi dilakukan menggunakan model TensorFlow Lite.
- Fungsi *predict_depression* merupakan antarmuka lebih tinggi yang digunakan untuk memprediksi kesehatan mental untuk satu teks. Hasil prediksi mencakup nilai kepercayaan (*confidence score*) dan kategori kesehatan mental (*depression* atau tidak).

c. API Routes untuk *Prediction*

```

@predict_depression_route.route("/depression",
methods=["GET", "POST"])
def predict_text():
    if request.method == "GET":
        return jsonify({"message": "Predict Depression
API"}), 200
    elif request.method == "POST":
        try:
            request_data = request.form
            text = request_data["text"]
            if not text:
                raise Exception("No text provided!")
            predicted_data = predict_depression(text)
            depression = predicted_data["depression"]
            confidence_score =
predicted_data["confidence_score"]

```

```

        return (
            jsonify(
                {
                    "status": "success",
                    "message": "Text predicted
successfully!",
                    "data": {
                        "text": text,
                        "depression": depression,
                        "confidenceScore": confidence_score,
                    },
                }
            ),
            200,
        )
    except Exception as e:
        return (
            jsonify(
                {
                    "status": "error",
                    "message": "No text provided!, please
provide text string in the form of 'text'",
                    "error": str(e),
                }
            ),
            500,
        )
    
```

Endpoint API diatur dalam fungsi *predict_text*. Pada metode *GET*, API memberikan respons yang menyatakan bahwa ini adalah API untuk memprediksi kesehatan mental. Pada metode *POST*, API menerima teks dari permintaan dan memprediksi kesehatan mental dengan menggunakan fungsi *predict_depression*. Hasil prediksi, bersama dengan teks input dan nilai kepercayaan, dikirimkan sebagai respon JSON.

- d. Kode *Dockerfile* untuk *deploy* ke *Google Cloud Run*

```

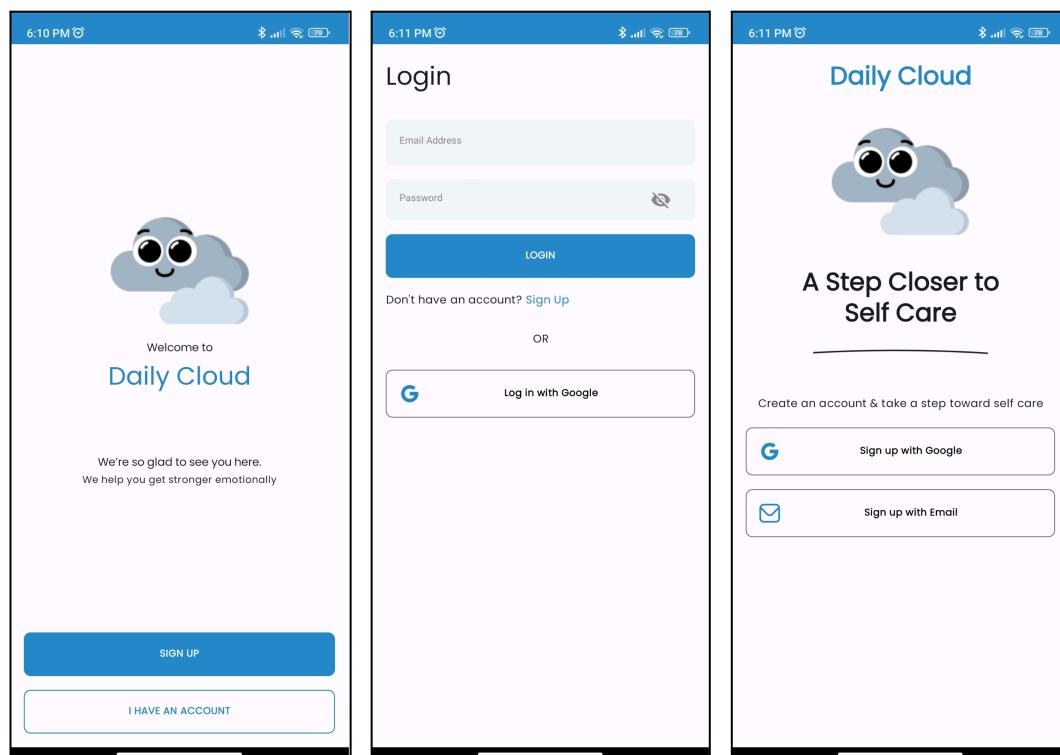
FROM tensorflow/tensorflow:latest
ENV PYTHONUNBUFFERED True
ENV APP_HOME /app
WORKDIR $APP_HOME
COPY . .
RUN pip install --no-cache-dir -r requirements.txt
CMD exec gunicorn --workers 1 --threads 8 --timeout 0
main:app
    
```

File *Dockerfile* ini menyusun konfigurasi untuk *container Docker* yang akan digunakan untuk menjalankan *Prediction API* di *Google Cloud Run*. Pada contoh ini, *TensorFlow Lite* dan *gunicorn* digunakan sebagai *server* aplikasi. Proses ini melibatkan instalasi dependensi dari file *requirements.txt* dan menjalankan *gunicorn* sebagai *server* aplikasi dengan satu *workers* dan delapan *thread*. File *Dockerfile* penting untuk memastikan bahwa lingkungan runtime di dalam *container* sesuai dengan kebutuhan aplikasi dan dapat dijalankan di *Google Cloud Run*.

3. User Interface Aplikasi Daily Cloud

Berikut adalah tampilan (*User Interface*) Aplikasi yang telah dibuat yaitu Daily Cloud dapat dilihat sebagai berikut:

1) Tampilan *Login* dan *Register*

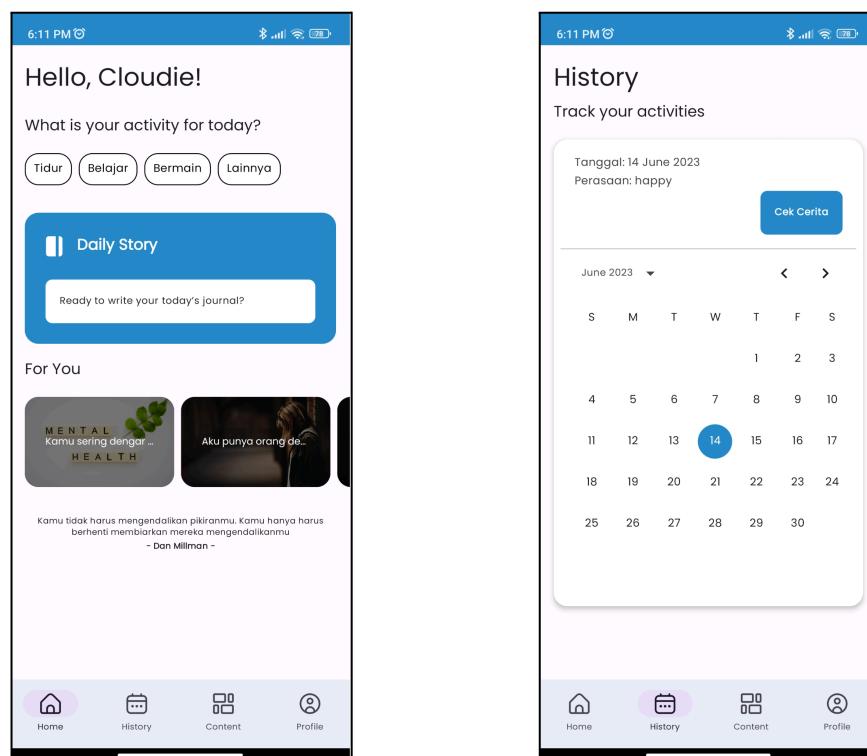


Gambar IV.7 Tampilan Login dan Register

Tampilan ini merupakan pintu gerbang ke aplikasi, di mana pengguna dapat masuk ke akun yang sudah ada atau mendaftar untuk membuat akun

baru. Desainnya dirancang untuk memberikan pengalaman pengguna yang intuitif, dengan formulir input yang jelas dan panduan yang membantu pengguna dalam proses login atau pendaftaran. Memiliki menu yang mudah diakses dan instruksi yang jelas adalah kunci dalam memastikan bahwa pengguna dapat dengan mudah mengakses layanan aplikasi.

2) Tampilan *Home* dan *History Activity*

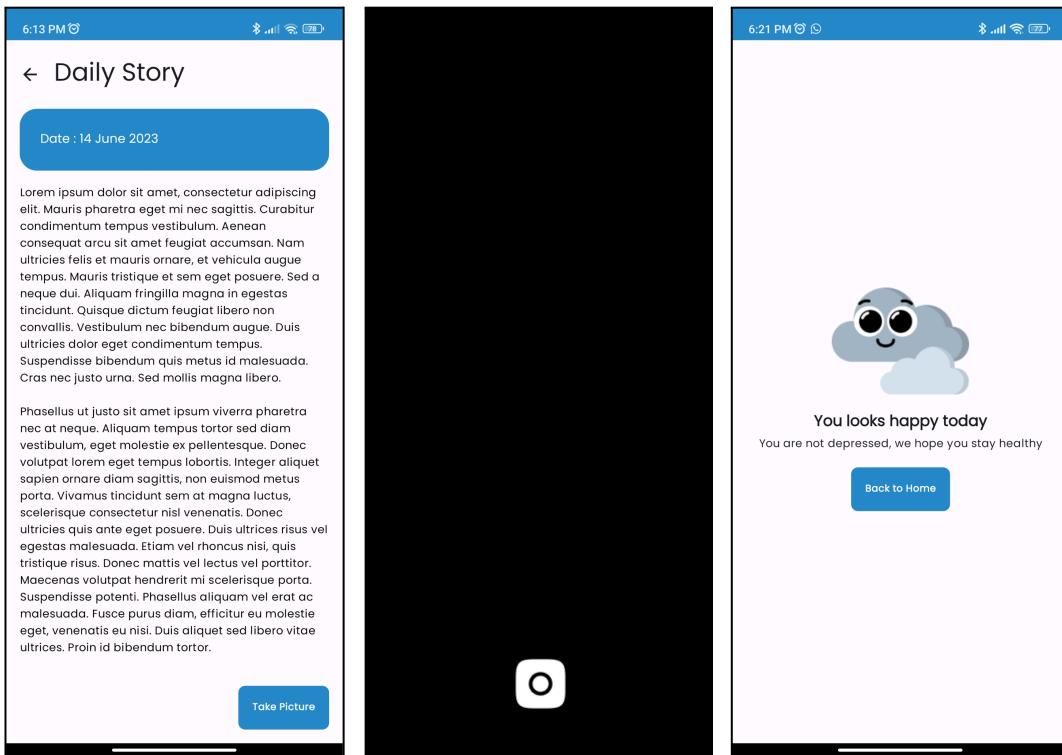


Gambar IV.8 Tampilan Home dan History Activity

Tampilan Home adalah pusat kontrol dari aplikasi, di mana pengguna dapat melihat informasi penting seputar kesehatan mental mereka. Ini bisa berupa ringkasan harian, statistik perkembangan kesehatan mental, atau berita terkini seputar topik kesehatan mental.

Sementara itu, Tampilan History Activity memberikan gambaran tentang aktivitas dan perkembangan pengguna dari waktu ke waktu. Melacak perubahan dan kemajuan adalah bagian integral dari perawatan kesehatan mental, dan desain yang bersih dan informatif membantu pengguna untuk melihat dengan jelas bagaimana mereka telah berkembang.

3) Tampilan *Input Story* dan *Capture Wajah*



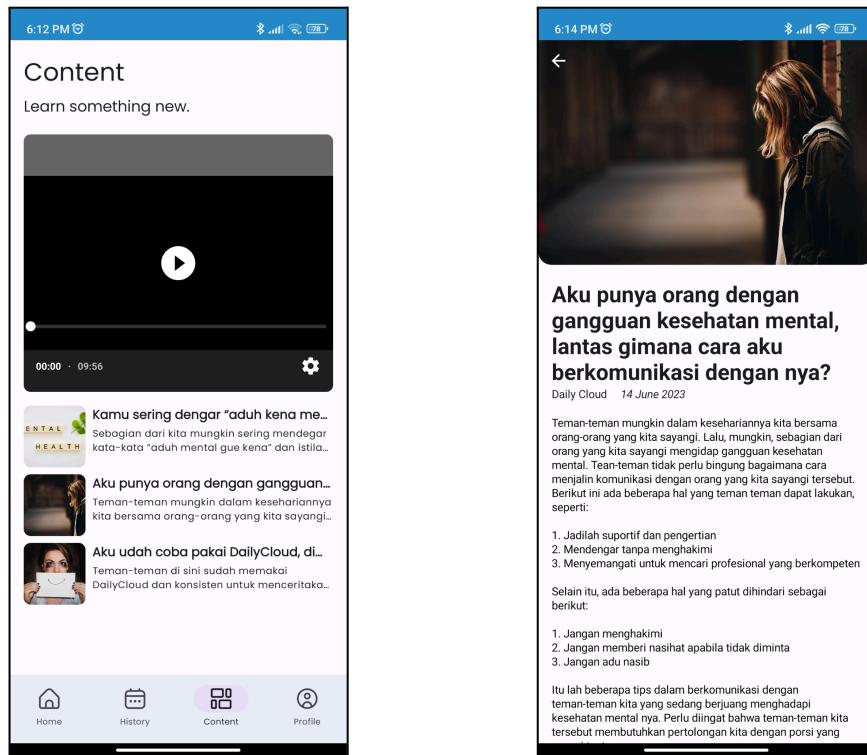
Gambar IV.9 Tampilan Input Story dan Capture Wajah

Tampilan Input Story adalah tempat di mana pengguna dapat mengekspresikan diri mereka dengan menuliskan cerita atau catatan harian seputar perasaan dan pengalaman pengguna. Desainnya harus memberikan pengalaman menulis yang nyaman dan memberikan pengguna kebebasan untuk mengekspresikan diri mereka dengan bebas.

Sementara itu, Tampilan Capture Wajah memungkinkan pengguna untuk menggunakan kamera ponsel mereka untuk merekam ekspresi wajah mereka. Fitur ini dapat memberikan wawasan tambahan tentang perasaan dan emosi pengguna, serta membantu mereka dalam memantau perubahan dalam kondisi emosional mereka dari waktu ke waktu.

Setelah pengguna mengisi catatan harian dan melakukan capture wajah, langkah berikutnya melibatkan pengiriman data tersebut ke server untuk dilakukan prediksi yang bertujuan melakukan monitoring keadaan mental dari pengguna.

4) Tampilan Menu Artikel



Gambar IV.10 Tampilan Menu Artikel

Tampilan ini berfungsi sebagai sumber informasi dan panduan seputar kesehatan mental. Pengguna dapat menjelajahi berbagai artikel dan sumber daya yang tersedia untuk mendapatkan pemahaman yang lebih baik tentang topik-topik seperti manajemen stres, keseimbangan emosional, dan strategi pemulihan. Desainnya harus menarik dan mudah dinavigasi, dengan fitur pencarian yang kuat dan kategori yang terorganisir dengan baik untuk memastikan pengguna dapat dengan mudah menemukan konten yang mereka butuhkan.

Setelah menyelesaikan *Capstone Project* oleh Bangkit Academy ini, banyak ilmu yang diperoleh terkait dengan pembangunan arsitektur *cloud* khususnya *Google Cloud*, proses pembuatan API, Firestore *database* dan proses *deployment* API di ke layanan Serverless Google Cloud Run.

BAB V

PENUTUP

V.1 Kesimpulan dan Saran Mengenai Pelaksanaan

Setelah menyelesaikan kerja praktik, berikut adalah kesimpulan dan saran yang dapat penulis sampaikan mengenai pelaksanaan kerja praktik:

V.1.1 Kesimpulan Pelaksanaan Kerja Praktik

Kesimpulan yang diperoleh selama melaksanakan kerja praktik adalah sebagai berikut:

1. Mahasiswa dapat mengimplementasikan ilmu dan keterampilan yang diperoleh selama perkuliahan untuk menyelesaikan permasalahan di dunia nyata.
2. Mahasiswa dapat mengetahui ilmu dan keterampilan yang dibutuhkan untuk memasuki dunia kerja, seperti:
 - a. Keterampilan berkomunikasi dan bekerja sama dengan tim.
 - b. Keterampilan menganalisis permasalahan kemudian dicari solusinya.
 - c. Keterampilan mempelajari hal baru dalam waktu yang relatif singkat.
3. Mahasiswa mendapat pengalaman serta pengetahuan mengenai pentingnya etos kerja yang baik, disiplin, dan tanggung jawab dalam menyelesaikan suatu pekerjaan. Kerja praktik dapat melatih mahasiswa untuk bekerja sama dalam suatu tim, baik antar rekan kerja praktik maupun dengan pihak Instansi.
4. Mahasiswa memperoleh ilmu baru yang tidak didapatkan selama perkuliahan. Pada kerja praktik di Bangkit Academy, pengetahuan yang diperoleh mengenai:
 - a. Konsep *Cloud Computing* dan penggunaannya dengan Google Cloud Platform.
 - b. Ilmu tentang *soft-skills* yang diajarkan di Bangkit Academy Google Indonesia.
 - c. Berkolaborasi dengan tim dari berbagai *learning path* dalam pengajaran *Capstone Project*.

V.1.2 Saran Pelaksanaan Kerja Praktik

Adapun saran mengenai pelaksanaan kerja praktik antara lain:

1. Selain keterampilan teknis, perlu ditekankan pengembangan soft skills seperti kemampuan komunikasi, kerjasama tim, dan kepemimpinan. Penguasaan soft skills akan membantu mahasiswa dalam berinteraksi secara efektif di lingkungan kerja dan meningkatkan daya saing peserta di pasar tenaga kerja.
2. Tekankan studi kasus sebuah proyek pada materi pembelajaran agar lebih relevan dengan perkembangan industri. Hal ini akan membantu mahasiswa memahami tren industri dan bagaimana teori yang dipelajari dapat diaplikasikan dalam situasi kerja yang sebenarnya.
3. Sediakan informasi dan pengenalan terhadap praktik terbaik di industri teknologi dan pengembangan perangkat lunak. Dengan demikian, mahasiswa dapat memahami praktik terkini dalam lingkungan pekerjaan dan mengetahui standar yang diharapkan dari mereka
4. Perlu adanya bimbingan secara lebih intensif bagi mahasiswa kerja praktik agar pelaksanaannya lebih terarah. Hal ini dapat dilakukan dengan meningkatkan frekuensi pertemuan dan komunikasi antara dosen pembimbing dan mahasiswa, serta menyediakan platform online untuk diskusi dan tanya jawab.

V.2 Kesimpulan dan Saran Mengenai Substansi

Berikut adalah kesimpulan dan saran mengenai substansi yang dapat disampaikan selama kerja praktik di Google Indonesia:

V.2.1 Kesimpulan

Setelah menyelesaikan kerja praktik di Google Indonesia, Aplikasi Daily Cloud berhasil dibangun oleh penulis bersama dengan rekan tim. Aplikasi ini memiliki fitur utama yaitu memprediksi kesehatan mental pengguna berdasarkan jurnal dan ekspresi wajah pengguna serta menyediakan artikel untuk edukasi pengguna. Berikut adalah beberapa hasil kerja praktik penulis yang berperan sebagai *Cloud Computing* berhasil membangun server arsitektur *cloud*

menggunakan *Google Cloud Platform* sehingga kinerja aplikasi Daily Cloud dapat berjalan optimal:

1. Menggunakan *Google Cloud Run* sebagai server untuk *deployment REST API* dari Back-End. *Cloud Run* memungkinkan aplikasi beradaptasi secara dinamis dengan permintaan pengguna, memberikan fleksibilitas dan skalabilitas tinggi.
2. Menggunakan *Firebase* sebagai layanan *database* yang *scalable* dan terintegrasi dengan fitur autentikasi dari *Firebase Authentication*.
3. Membangun *Back-End Main API* sebagai *REST API* inti dari aplikasi *Daily Cloud* dengan beberapa fitur yaitu *User API* untuk manajemen pengguna, *Journal API* untuk menyimpan dan mengambil jurnal pengguna, *Article API* untuk mengakses artikel, dan *Quotes of The Day API* untuk memberikan kutipan motivasi harian.
4. Melakukan *deployment* model *machine learning* menjadi sebuah *REST API* bernama *Prediction API* yang berfungsi untuk memprediksi kesehatan mental pengguna berdasarkan data jurnal yang dikirimkan.

V.2.2 Saran

Dari hasil kerja praktik khususnya Arsitektur *Cloud* dengan *Google Cloud*, terdapat beberapa saran yang dapat diajukan, antara lain:

1. Optimalkan penggunaan layanan *Google Cloud*, bandingkan penggunaan layanan yang telah digunakan dengan layanan serupa lainnya untuk mendapatkan efisiensi dan performa yang maksimal.
2. Tingkatkan keamanan *server* dengan mengimplementasikan layanan keamanan *Google Cloud* untuk mencegah potensi ancaman keamanan terhadap aplikasi *Daily Cloud*.
3. Implementasikan alat-alat *monitoring* dan analisis kinerja seperti *Google Cloud Monitoring* dan *Google Cloud Logging* untuk memantau kinerja aplikasi secara *real-time* dan mengidentifikasi masalah dengan cepat.
4. Optimalkan *deployment* model *machine learning* menjadi sebuah layanan yang berdiri sendiri agar performa prediksi menjadi maksimal.

DAFTAR PUSTAKA

- Ika. (2015). Minim Psikolog, Ribuan Penderita Gangguan Jiwa Belum Tertangani. Diakses pada tanggal 26 Juni 2023, dari <https://ugm.ac.id/id/berita/9715-minim.psikolog.ribuan.penderita.gangguan.jiwa.belum.tertangani>
- Kementerian Kesehatan Republik Indonesia. (2022). Mengenal Pentingnya Kesehatan Mental pada Remaja. Diakses pada tanggal 26 Juni 2023, dari https://yankes.kemkes.go.id/view_artikel/362/mengenal-pentingnya-kesehatan-mental-pada-remaja
- Rokom. (2021). Kemenkes Beberkan Masalah Permasalahan Kesehatan Jiwa di Indonesia. Sehat Negeriku. Diakses pada tanggal 26 Juni 2023, dari <https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20211007/1338675/kemenkes-beberkan-masalah-permasalahan-kesehatan-jiwa-di-indonesia/>
- Sardjito. (2022). Minimnya Kesadaran Masyarakat terhadap Mental Health. Diakses pada tanggal 26 Juni 2023, dari <https://sardjito.co.id/2022/03/09/minimnya-kesadaran-masyarakat-terhadap-mental-health/>
- Wijaya, Y. D. (2019). Kesehatan Mental di Indonesia: Kini dan Nanti. Buletin Jagaddhita, 1(1), 1-4.
- World Health Organization. (2022). Mental health: Strengthening our response. Diakses pada tanggal 26 Juni 2023, dari <https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-responses>

LAMPIRAN A

TOR

Perencanaan yang matang telah dilakukan untuk merancang kurikulum program ini dari awal hingga akhir. Pada akhir program, semua peserta yang memenuhi kriteria kelulusan akan dianggap sebagai Lulusan Bangkit dan diberikan sertifikat kelulusan serta transkrip nilai lengkap. Bagi peserta yang tidak memenuhi kriteria kelulusan Bangkit, akan mendapatkan Sertifikat Kehadiran dan transkrip nilai sebagian. Lulusan Bangkit juga akan mendapatkan voucher untuk mengikuti ujian sertifikasi di Jalur Pembelajaran masing-masing. Persyaratan kelulusan Bangkit 2023 adalah sebagai berikut:

- **Hadir dan aktif berpartisipasi dalam sesi wajib** yang termasuk, namun tidak terbatas pada:
 - Sesi Pembukaan Bangkit 2023 & *Technical Briefing*
 - 80% kehadiran untuk sesi *Instructor-led Training Technical**
 - 80% kehadiran untuk sesi *Instructor-led Training Soft Skills**
 - 90% dari sesi tamu/sesi khusus yang wajib
 - Dan sesi wajib lainnya yang ditambahkan atas kebijakan Tim Bangkit.
 - **Maksimal 5 sesi** absen atau mengerjakan abstrak*

Sesi akan diumumkan setidaknya 3 hari sebelum kelas dimulai.
Dimohon untuk periksa kalender setiap hari.

* Peserta dapat absen untuk tidak mengikuti sesi karena keadaan darurat dengan mengisi formulir ini (maksimal 1 hari sebelum sesi). Sesi yang terlewatkan harus diganti dengan mengikuti sesi kelompok lain atau menonton rekaman dan mengirimkan abstrak.

- Untuk sesi mandiri, Anda hanya perlu menyelesaikannya dalam minggu yang sama. Jika Anda memiliki hal lain yang harus dilakukan pada saat alokasi waktu belajar mandiri, Anda tidak perlu mengisi formulir. Cukup alokasikan waktu lain di luar waktu yang dialokasikan Bangkit untuk belajar disesuaikan waktunya.

- **Kirimkan karya Anda sendiri untuk tugas dan proyek.** Bangkit merupakan bagian dari program Kampus Merdeka dimana integritas akademik dijunjung tinggi. Anda harus menunjukkan dan menjunjung tinggi integritas dan kejujuran tertinggi dalam semua pekerjaan akademik yang Anda lakukan. Plagiarisme tidak diperbolehkan, dan nilai untuk tugas yang telah dikerjakan akan dibatalkan jika karya Anda terindikasi melakukan plagiat. Platform pembelajaran mitra kami dapat melarang atau menonaktifkan akun Anda jika Anda melakukan plagiat atau tidak jujur berdasarkan kebijakan mereka yang mutlak.
- **Menyelesaikan tugas-tugas resmi Bangkit** (termasuk classroom dan mitra platform pembelajaran kami - Dicoding, Google Cloud Skills Boost, Coursera) sesuai dengan standar masing-masing. Keterlambatan pengumpulan tugas akan diterima, tetapi akan mengurangi skor tugas yang bersangkutan.
- **Berkontribusi pada Bangkit Capstone Project.** Ini akan dinilai oleh Komite Bangkit dan anggota tim Anda dan termasuk kehadiran Anda dalam presentasi proyek akhir.
- **Mematuhi Kode Etik Bangkit.** Antara lain :
 - Integritas
 - Jujur
 - Bermanfaat
 - Responsif
 - Bertindak Cepat
 - Menaati peraturan
 - Saling mendukung satu sama lain
 - Dilarang keras melakukan Pelecehan, Diskriminasi, dan Penindasan

LAMPIRAN B
LOG ACTIVITY

Minggu/Tgl	Kegiatan	Hasil
Ke-0 / 16 Feb - 17 Feb 2023	Students Technical Briefing	Membahas tentang perkenalan program Bangkit Academy 2023
	Life Path Session	Membahas tentang perjalanan hidup, hal-hal yang mempengaruhi jalan hidup, serta tips dalam merefleksikannya.
	Matriculation	Memahami tentang dasar-dasar pengembangan software, logika pemrograman, git dan github.
	Preread SS 1 "Growth Mindset and The Power of Feedback"	Mendapatkan modul pembelajaran Growth Mindset and The Power of Feedback untuk persiapan sesi ILT-Softskills
Ke-1 / 20 Feb - 24 Feb 2023	Bangkit 2023 Opening Session	Membahas tentang pembukaan program Bangkit Academy 2023
	ILT SS 01	Memahami tentang konsep Growth Mindset dan manfaatnya, serta memahami cara memberikan Feedback yang efektif.
Ke-2 / 27 Feb - 03 Mar 2023	Assignment SS 1	Penugasan menjawab soal untuk pemahaman lebih lanjut mengenai Growth Mindset and The Power of Feedback
	ILT-Tech-01: Introduction to Google Cloud	Memahami konsep dan cara kerja Cloud Computing, perkenalan Google Cloud Platform seperti cara kerja dan layanan yang tersedia.
	Preread SS 2 "Time Management"	Mendapatkan modul pembelajaran Time Management untuk persiapan sesi ILT-Softskills
	English Pre-Test	Mengukur kemampuan Bahasa Inggris sebelum mengikuti sesi ILT-English
	The Bits and Bytes of Computer Networking by Coursera	Memahami tentang Jaringan Komputer seperti TCP/IP Five-Layer Network Model yaitu Physical, Data Link, Network, Transport, and Application layers.
	Google Cloud Computing Foundations: Cloud Computing by Google SkillBoost	Memahami dasar Cloud Computing, jenis layanan pada Cloud Computing, cara menggunakan Google Cloud Platform melalui hands-on labs.
	Google Cloud Computing Foundations: Infrastructure in Google Cloud by Google SkillBoost	Memahami layanan storage, database, API, dan keamanan pada Google Cloud Platform. Serta praktiknya melalui hands-on labs.

	Google Cloud Computing Foundations: Networking and Security by Google SkillBoost	Memahami layanan networking pada Google Cloud Platform, seperti VPC, Firewall, dan Load Balancing. Serta praktiknya melalui hands-on labs.
	Google Cloud Computing Foundations: Data, ML, and AI in Google by Google SkillBoost	Memahami layanan data seperti BigQuery, Dataproc, dan lain-lain, machine learning seperti AutoML, dan artificial intelligence seperti VertexAI pada Google Cloud Platform.
	Create and Manage Cloud Resources (Quest)	Praktik GCP tentang membuat Virtual Machine pada Compute Engine, mengelola Kubernetes Engine, dan membuat HTTP Load Balancer.
	Perform Foundational Infrastructure Tasks in Google Cloud (Quest)	Praktik GCP tentang membuat Cloud Storage Bucket, Cloud IAM, Cloud Monitoring, Cloud Functions, dan Google Pub/sub.
	Build and Secure Networks in Google Cloud (Quest)	Praktik GCP tentang membuat Multiple VCP Networks, membuat Nginx Server dan mengelola akses External HTTP ke server dengan Firewall, dan membuat Internal Load Balancer.
Ke-3 / 06 Mar - 10 Mar 2023	ILT SS 02	Memahami konsep dan pentingnya Time Management, serta 5 pilar tentang Time Management.
	Perform Foundational Data, ML, and AI Tasks in Google Cloud (Quest)	Praktik GCP yaitu penggunaan Vertex AI, Dataprep, Dataflow, Dataproc, Cloud Natural Language API, Google Cloud Speech API, dan Video Intelligence.
	Belajar Dasar Pemrograman Web (up to done) by Dicoding	Mempelajari Dasar HTML dan CSS, serta mengerjakan tugas yaitu membuat website dengan tema bebas berdasarkan ketentuan.
Ke-4 / 13 Mar - 17 Mar 2023	Bangkit 2023 H1 - Student Team Meeting 1	Membahas tentang tips belajar di Bangkit, ILT English update, reward system, dan student dashboard.
	Assignment SS 2	Penugasan menjawab soal untuk pemahaman lebih lanjut mengenai Time Management
	Preread SS 3 "Critical Thinking and Problem Solving"	Mendapatkan modul pembelajaran Critical Thinking and Problem Solving untuk persiapan sesi ILT-Softskills
	ILT Tech 2 JavaScript for Server-Side Development	Memahami dasar Server-Side application, dasar REST API, dasar Node.js untuk server, membuat HTTP Server, dan testing menggunakan Postman.
	ILT English 1 Spoken Correspondence	Memahami tentang panduan berkomunikasi dengan telepon dan video konferensi pada kehidupan bisnis.
	Belajar Dasar Pemrograman JavaScript (up to done) by Dicoding	Memahami fundamental JavaScript, struktur data, functions, OOP, Modules, NPM, Javascript Testing, dan lain-lain.
	Belajar Membuat	Memahami dasar Back-End, seperti server, web

	Aplikasi Back-End untuk Pemula dengan Google Cloud (up to Kuis Pengenalan Back-End) by Dicoding	service, komunikasi client-server, REST web service, HTTP Verbs dan Response, dan URL Design.
Ke-5 / 20 Mar - 24 Mar 2023	ILT SS 3	Memahami pentingnya Critical Thinking sebagai dasar bagi Problem Solving yang efektif
	Belajar Membuat Aplikasi Back-End untuk Pemula dengan Google Cloud (up to done) by Dicoding	Memahami cara membangun Web Service dengan JavaScript menggunakan framework Hapi dan testing API dengan software Postman. Serta mengerjakan proyek akhir Bookshelf API.
Ke-6 / 27 Mar - 31 Mar 2023	ILT Tech 3 Build Great Solutions with Compute and Network Services in Google Cloud	Memahami layanan komputasi pada GCP seperti Compute Engine, Cloud Run, GKE, dan lainnya. Serta layanan jaringan seperti VPC, Cloud DNS, Cloud VPN, dan lainnya.
	System Administration and IT Infrastructure Services by Coursera	Memahami tentang pentingnya System Administation dan Infrastructure Services dalam perusahaan.
	Preparing for your Associate Cloud Engineer Journey by Google SkillBoost	Memahami cara mengatur persiapan untuk mengikuti ujian Associate Cloud Engineer.
	Assignment SS 3	Penugasan menjawab soal untuk pemahaman lebih lanjut mengenai Critical Thinking and Problem Solving
	Preread SS 4 "Adaptability and Resilience"	Mendapatkan modul pembelajaran Adaptability and Resilience untuk persiapan sesi ILT-Softskills
Ke-7 / 03 Apr - 07 Apr 2023	ILT SS 4	Memahami pentingnya adaptabilitas dalam kehidupan, cara-cara untuk beradaptasi, dan bagaimana membangun ketangguhan secara efektif.
	Google Cloud Fundamentals: Core Infrastructure by Google SkillBoost	Memahami lebih dalam konsep dan terminologi dalam Google Cloud Platform.
	Essential Google Cloud Infrastructure: Foundation by Google SkillBoost	Memahami lebih dalam konsep infrastruktur dalam Google Cloud Platform, seperti VPC dan Virtual Machines.
	Essential Google Cloud Infrastructure: Core Services by Google	Memahami lebih dalam mengenai Identity Access Management (IAM) untuk pengelolaan anggota dalam perusahaan, layanan Storage, serta monitoring.

	SkillBoost	
Ke-8 / 10 Apr - 14 Apr 2023	Bangkit 2023 H1 - Student Team Meeting 2	Rangkuman pembelajaran bulan sebelumnya, milestone berikutnya, grading system update, dan pengenalan Capstone.
	Capstone Project Team Creating	Bergabung ke tim Capstone.
	ILT Tech 4 Store Your Data Without Worry in Google Cloud	Memahami beberapa layanan terkait manajemen data dan solusi database di Google Cloud seperti Storage, Database, dan Big Data.
	Elastic Google Cloud Infrastructure: Scaling and Automation by Google SkillBoost	Memahami layanan infrastruktur dan platform yang komprehensif dan fleksibel yang disediakan oleh Google Cloud.
	Getting Started with Google Kubernetes Engine by Google SkillBoost	Memahami konsep Container, dasar-dasar kubernetes, arsitektur kubernetes, dan kubernetes workloads.
	Assignment SS 4	Penugasan membuat LinkedIn post berjudul "5 Ways of Adaptability" untuk pemahaman lebih lanjut mengenai Adaptability and Resilience
Ke-9 / 17 Apr - 21 Apr 2023	Logging, Monitoring and Observability in Google Cloud by Google SkillBoost	Memahami bagaimana memantau, menangani masalah, dan meningkatkan kinerja infrastruktur dan aplikasi dengan prinsip Site Reliability Engineering (SRE).
	Getting Started with Terraform for Google Cloud by Google SkillBoost	Memahami Terraform untuk Google Cloud dalam penerapan infrastruktur sebagai kode (IaC).
Libur Lebaran 20 April - 26 April 2023		
Ke-10 / 24 Apr - 28 Apr 2023	Set Up and Configure a Cloud Environment in Google Cloud (Quest)	Praktik GCP tentang penggunaan IAM, membuat VPC networks, deploy dan monitor aplikasi pada Compute Engine, menggunakan Cloud SQL, serta mengelola Google Kubernetes Engine.
	Automating Infrastructure on Google Cloud with Terraform (Quest)	Praktik GCP tentang membangun infrastruktur GCP dengan Kode menggunakan Terraform, serta penggunaan modul terraform.
	Preread SS 5 "Project Management"	Mendapatkan modul pembelajaran Project Management untuk persiapan sesi ILT-Softskills
Ke-11 / 01 Mei - 05 Mei 2023	ILT SS 5	Memahami manajemen proyek dan tahapannya, serta aspek-aspek penting dalam kesuksesan proyek.
	Menjadi Google Cloud Engineer (up to Kuis	Memahami tentang pengenalan Google Cloud, layanan komputasi, data, jaringan, monitoring dan logging di

	Monitoring dan Logging) by Dicoding	Google Cloud. Serta mengerjakan proyek pertama mendeploy aplikasi Profile pada App Engine.
	Capstone Project Plan Submission	Mengirim Capstone Project plan kepada pihak Bangkit. Dengan keputusan rencana membuat Aplikasi mengenai Mental Health.
Ke-12 / 08 Mei - 12 Mei 2023	ILT Tech 5 Simplify Your Life with Automation and Operation Services in Google Cloud	Memahami tentang layanan Google Cloud Marketplace, konsep Infrastructure as Code (IaC), dan layanan pada Google Cloud Operation Suite.
	Menjadi Google Cloud Engineer (up to done) by Dicoding	Memahami layanan Cloud IAM, deploy dan implementasi solusi cloud. Serta mengerjakan proyek akhir yaitu deploy aplikasi Money Tracker App di Compute Engine dan App Engine.
	Application Development with Cloud Run	Memahami layanan serverless Cloud Run, konsep dan cara kerjanya.
	Assignment SS 5	Penugasan menjawab soal untuk pemahaman lebih lanjut mengenai Project Management
	Preread SS 6 "Professional Communication and Networking"	Mendapatkan modul pembelajaran Professional Communication and Networking untuk persiapan sesi ILT-Softskills
Ke-13 / 15 Mei - 19 Mei 2023	Bangkit 2023 H1 - Student Team Meeting 3	Membahas tentang persiapan Bangkit Career Fair, Google Certification, Capstone Update, dan Cohort Update.
	ILT English 2 Expressing Opinions	Memahami cara meminta dan memberikan pendapat, serta memberikan tanggapan dengan cara yang sopan dan tepat.
	ILT SS 6	Memahami cara berkomunikasi yang efektif, memahami SOSIAL STYLE, dan prinsip Networking dalam kehidupan.
	Hybrid Cloud Modernizing Applications with Anthos by Google SkillBoost	Memahami konsep modernisasi hybrid cloud, migrasi aplikasi dari server on-premise ke Google Cloud dengan Anthos.
	Capstone Project - Week 1	Persiapan Github repository, spesifikasi API, Firebase Admin, dan setup ExpressJS project untuk pembuatan REST API, mempersiapkan GCP Project untuk Infrastruktur Google Cloud, serta Firestore untuk Database.
Ke-14 / 22 Mei - 26 Mei 2023	ILT Tech 6 Manage Your Cloud Resources Securely	Memahami konsep-konsep keamanan Google Cloud, serta pengelolaan Identity Access Management (IAM) di Google Cloud.

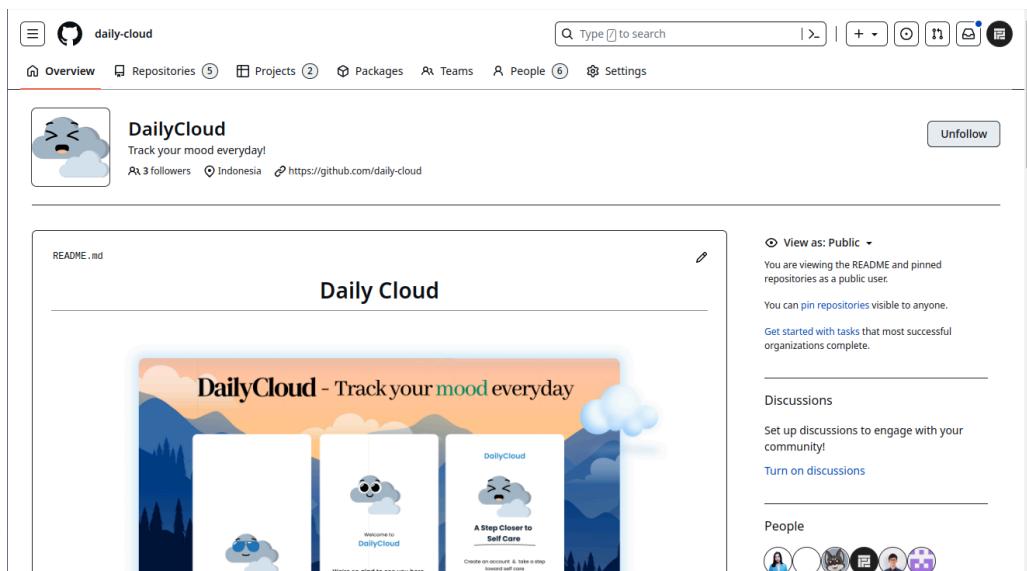
	Serverless Cloud Run Development (Quest)	Praktik GCP tentang membangun aplikasi serverless menggunakan Cloud Run.
	Deploy to Kubernetes in Google Cloud (Quest)	Praktik GCP tentang membuat Docker Image dengan Dockerfile, serta menggunakan image tersebut untuk deploy aplikasi ke Google Kubernetes Engine.
	Assignment SS 6	Penugasan membuat video tentang "Learning experiences" dengan 7C of Communication untuk pemahaman lebih lanjut mengenai Professional Communication and Networking
	Preread SS 7 "Personal Branding and Interview Communication"	Mendapatkan modul pembelajaran Personal Branding and Interview Communication untuk persiapan sesi ILT-Softskills
	Capstone Project - Week 2	Mengerjakan User API untuk signup, melihat detail user, dan mengubah data user serta menyiapkan Cloud Storage sebagai tempat menyimpan User Image, serta deployment ke Cloud Run.
	Capstone Project Mentoring Tech #1 With Kak Caesar Irawan	Memahami tentang Business Proposal, program/tools untuk kolaborasi tim, serta saran dari mentor untuk deploy ML model ke Cloud Run.
	Capstone Project Mentoring Project Management #1 With Kak Muhammad Fikri Kawakibi Huda	Memahami tentang Business Model, seperti ads, advertorials, psychologist consultation, paywall, dan subscription, serta saran dari mentor untuk meningkatkan user flow dari aplikasi.
Ke-15 / 29 Mei - 02 Jun 2023	ILT SS 7	Memahami bagaimana membuat Personal Branding yang kuat dan bagaimana cara berkomunikasi secara efektif selama wawancara.
	Assignment SS 7	Penugasan membuat video tentang menjawab Interview untuk pemahaman lebih lanjut mengenai Personal Branding and Interview Communication
	Capstone Project Mid Checkpoint	Mengirim progress Capstone Project ke pada tim Bangkit dan harus memenuhi kriteria 50% progress.
	Capstone Project Week 3	Menyiapkan Cloud Build untuk CI/CD serta memperbarui Dokumentasi dari API.
Ke-16 / 05 Jun - 09 Jun 2023	Capstone Project Week 4	Membuat Article API untuk mendapatkan semua artikel, mendapatkan artikel berdasarkan ID, dan Quotes of The Day API untuk mendapatkan random quotes, serta membuat API baru menggunakan Python dengan Flask untuk memprediksi jurnal ke ML model.
Ke-17 / 12 Jun - 16 Jun 2023	Bangkit 2023 H1 - Student Team Meeting 4	Membahas tentang pelaksanaan Bangkit Career Fair 2023, Presentasi untuk Capstone Project, Cohort Update, Rewards Update, dan Laporan Akhir MSIB.

	ILT English 3 Business Presentation	Memahami tentang bagaimana membuat dan menyampaikan presentasi yang efektif dan menarik.
	Capstone Project Week 5	Membuat Journal API untuk menambah, mendapatkan semua jurnal, melihat jurnal berdasarkan ID, dan memperbaiki beberapa bug pada User API ketika menambahkan user baru, serta deploy Main API dan Prediction API (Flask) ke Google Cloud Run untuk production.
	Capstone Project Final Deliverable	Tim mengumpulkan Final Deliverables Capstone Project, yang berisi tentang 10 menit video presentasi dari Aplikasi Daily Cloud.
Ke-18 / 19 Jun - 23 Jun 2023	Product-based Capstone Peer Review	Presentasi hasil product-based capstone, yaitu aplikasi Daily Cloud
Ke-19 / 26 Jun - 30 Jun 2023	Judging - Target: 15 Best teams Tech: Certification Prep (ACE Simulation Course Dicoding)	Mengerjakan simulasi ujian Associate Cloud Engineer (ACE) pada platform Dicoding.
Ke-20 / 03 Jul - 07 Jul 2023	ILT Tech 7 Prepare Yourself Before Taking the ACE Certification	Memahami lebih lanjut tentang ACE Certification, persiapan materi, soal-soal, dan pembahasan soal yang dapat membantu peserta sebelum mengikuti ACE Certification.
	Announcement & Incubation Offering Tech: Certification Prep & ILT Tech 7 (Cloud Explorations + ACE Examination Practice)	Persiapan sertifikasi, Eksplorasi Cloud, dan Praktik Ujian ACE
	English Post-Test	Mengukur kemampuan Bahasa Inggris setelah mengikuti sesi ILT-English
Ke-21 / 10 Jul - 14 Jul 2023	End of Learning, Certification Offering, Merchandise	Mendapatkan Token untuk ACE Certification, Merchandise dan Transkrip sementara
Ke-22 / 17 Jul - 21 Jul 2023	Transcript & Administration	Transkrip Nilai
Ke-23 / 24 Jul - 28 Jul 2023	Clarification, Legal & Letters, Closing.	Penutupan Program Bangkit

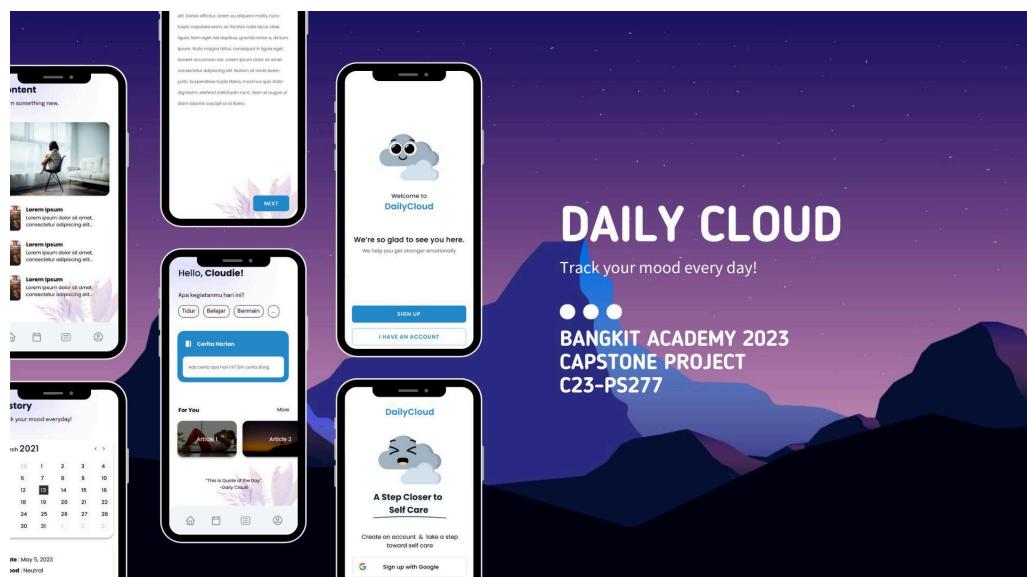
LAMPIRAN C

DOKUMEN TEKNIK

Pada lampiran ini, terlampir beberapa dokumen teknik sebagai bagian dari kerja praktik di Bangkit Academy 2023. Dokumen teknik ini mencakup informasi mendalam dan rinci terkait implementasi, desain, pengembangan arsitektur server *Google Cloud*, dan hasil dari *Capstone Project*.



Lampiran C.1 GitHub Capstone Project Daily Cloud



Lampiran C.2 Banner Aplikasi Daily Cloud