# FITUR *MACHINE LEARNING* UNTUK MENGIDENTIFIKASI KANDUNGAN GULA PADA MAKANAN: STUDI KASUS APLIKASI GLUCOFIT UNTUK PENCEGAHAN DIABETES DI INDONESIA




Disusun Oleh:

N a m a : Muhamad Rafli Fauzi
NIM : 21523042

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA**
**FAKULTAS TEKNOLOGI INDUSTRI**
**UNIVERSITAS ISLAM INDONESIA**
**2025**

**HALAMAN PENGESAHAN DOSEN PEMBIMBING**

# FITUR *MACHINE LEARNING* UNTUK MENGIDENTIFIKASI KANDUNGAN GULA PADA MAKANAN: STUDI KASUS APLIKASI GLUCOFIT UNTUK PENCEGAHAN DIABETES DI INDONESIA

**TUGAS AKHIR**

Disusun Oleh:

N a m a : Muhamad Rafli Fauzi

NIM : 21523042

Yogyakarta, 23 Juli 2025

Pembimbing,

( Izzati Muhimmah, S.T., M.Sc., Ph.D. )

**HALAMAN PENGESAHAN DOSEN PENGUJI**

# FITUR *MACHINE LEARNING* UNTUK MENGIDENTIFIKASI KANDUNGAN GULA PADA MAKANAN: STUDI KASUS APLIKASI GLUCOFIT UNTUK PENCEGAHAN DIABETES DI INDONESIA

## TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 6 Agustus 2025

Tim Penguji

Izzati Muhimmah, S.T., M.Sc., Ph.D

**Anggota 1**
Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs.

**Anggota 2**

Sri Mulyati, S.Kom., M.Kom.

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Muhamad Rafli Fauzi

NIM : 21523042

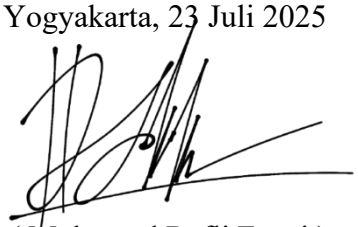
Tugas akhir dengan judul:

# FITUR *MACHINE LEARNING* UNTUK MENGIDENTIFIKASI KANDUNGAN GULA PADA MAKANAN: STUDI KASUS APLIKASI GLUCOFIT UNTUK PENCEGAHAN DIABETES DI INDONESIA

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 23 Juli 2025



( Muhamad Rafli Fauzi )

# HALAMAN PERSEMBAHAN

Kupersembahkan tugas akhir ini dengan penuh rasa syukur dan kerendahan hati:

Untuk Allah SWT,
Yang Maha Pengasih lagi Maha Penyayang, atas segala rahmat, taufik, dan hidayah-Nya yang tak terhingga. Tanpa izin dan pertolongan-Mu, tiada satu pun yang dapat kulakukan. Semoga skripsi ini menjadi amal yang Kau terima dan bermanfaat sebagai bentuk ibadah kepada-Mu.

Untuk Nabi Muhammad SAW,
Sang pembawa cahaya, teladan abadi dalam kesabaran, keikhlasan, dan keteguhan. Semoga shalawat dan salam selalu tercurah untukmu, ya Rasulullah. Aku berharap karya ini dapat mengikuti semangatmu dalam menebar ilmu dan kebaikan.

Kepada Ayahanda H. Fauzi Tambang, S.H., M.Si. dan Ibunda Hj. Iriana Rachmayanti,
Dua insan mulia yang tak kenal lelah memberikan doa, pengorbanan, dan dukungan tanpa batas. Setiap tetes keringat dan air mata kalian adalah energi yang menggerakkanku. Tak terhingga terima kasihku, semoga Allah membalas segala kebaikan dengan balasan yang berlipat ganda.

Untuk Saudari-Saudariku, Rasfia Wondayana Fauzi, dan Erinda Alfiani Fauzi
Yang selalu memberikan semangat, canda tawa, dan dukungan tanpa syarat. Terima kasih telah menjadi bagian dari perjalananku, menguatkanku di saat lelah, dan mengingatkanku untuk terus maju.

Untuk Sahabat dan Teman-Temanku, Agung, Bima, Fadhlan, Fikri, Ahsan, Luqman Yoangga, Maulana, Octaviano, Vahri, Rizqdwan, Wildan, Rizky PD, Wisnu, dan Nadiva
Terima kasih atas kebersamaan, diskusi panjang, dan bahu-membahu melewati suka dan duka selama ini. Keluarga kedua yang membuat perjalanan kuliah ini begitu berkesan dan menyenangkan.

Untuk Segenap Tenaga Pendidik di Universitas Islam Indonesia,
Bapak dan Ibu Dosen, serta staf pengajar yang dengan sabar membimbing, mengarahkan, dan memberikanku ilmu yang tak ternilai. Terima kasih atas dedikasi, nasihat, dan kesempatan yang diberikan. Semoga Allah membalas setiap jasa kalian dengan pahala yang berlipat.

Untuk Almamaterku, Universitas Islam Indonesia,
Terima kasih telah memberiku ruang untuk tumbuh, belajar, dan menemukan jati diri. Semoga kampus ini terus melahirkan generasi unggul yang bermanfaat bagi agama, bangsa, dan dunia.

Akhir kata, semoga skripsi ini dapat menjadi manfaat, setitik kontribusi bagi ilmu pengetahuan, dan menjadi amal jariyah yang terus mengalir.

# HALAMAN MOTO

"Sesungguhnya bersama kesulitan ada kemudahan."

(QS. Al-Insyirah: 5-6)

"Katakanlah: 'Sesungguhnya shalatku, ibadahku, hidupku, dan matiku hanyalah untuk Allah, Tuhan semesta alam.'"

(QS. Al-An'am: 162)

# KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya yang tak terhingga kepada seluruh hamba-Nya. Shalawat serta salam semoga senantiasa dilimpahkan kepada Nabi Muhammad SAW.

Berkat hidayah dan kemudahan dari Allah SWT, saya telah menyelesaikan tugas akhir yang berjudul "FITUR MACHINE LEARNING UNTUK MENGIDENTIFIKASI KANDUNGAN GULA PADA MAKANAN: STUDI KASUS APLIKASI GLUCOFIT UNTUK PENCEGAHAN DIABETES DI INDONESIA" sebagai salah satu syarat kelulusan program Strata-1 di Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

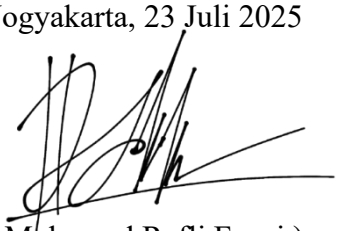
Proses penyusunan tugas akhir ini tidak terlepas dari dukungan, bimbingan, dan motivasi dari berbagai pihak. Oleh karena itu, dengan penuh keikhlasan, penulis mengucapkan terima kasih kepada:

1. Allah SWT, sumber segala petunjuk dan kemudahan, sehingga penulis dapat menyelesaikan karya ini dengan baik.
2. Bapak Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika UII, yang telah memberikan ilmu dan wawasan berharga.
3. Ibu Izzati Muhimmah, S.T., M.Sc., Ph.D., selaku dosen pembimbing yang senantiasa memberikan arahan, masukan, dan dukungan selama proses penelitian.
4. Kedua orang tua tercinta, atas segala kasih sayang, doa, dan pengorbanan yang tak ternilai.
5. Sahabat dan rekan-rekan, yang telah memberikan semangat dan kebersamaan dalam penyelesaian tugas akhir ini.
6. Semua pihak yang turut berkontribusi, baik secara langsung maupun tidak, yang tidak dapat disebutkan satu per satu.

Saya menyadari bahwa karya ini masih jauh dari sempurna. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

Akhir kata, semoga fitur aplikasi yang dikembangkan dapat memberikan manfaat, khususnya dalam meningkatkan efektivitas pencegahan penyakit Diabetes, serta turut berkontribusi pada pengembangan ilmu pengetahuan dan teknologi di bidang kesehatan.

Yogyakarta, 23 Juli 2025



( Muhamad Rafli Fauzi )

# SARI

Penelitian ini bertujuan mengembangkan fitur berbasis machine learning pada aplikasi GlucoFit untuk mengidentifikasi kandungan gula dalam makanan guna mendukung pencegahan diabetes di Indonesia. Diabetes melitus merupakan masalah kesehatan serius dengan prevalensi yang terus meningkat, salah satunya disebabkan oleh kurangnya kesadaran masyarakat terhadap asupan gula harian. Solusi yang ditawarkan adalah aplikasi mobile yang memanfaatkan teknologi Convolutional Neural Network (CNN) untuk mengenali jenis makanan dari foto dan memberikan estimasi kandungan gulanya, serta mengintegrasikan Generative AI API (Gemini) untuk menyajikan penjelasan ilmiah dan rekomendasi aktivitas fisik.

Metodologi penelitian mencakup pengumpulan data gambar makanan dari dataset Food-101, pra-pemrosesan data (cleaning, penyeimbangan kelas, augmentasi), pengembangan model CNN berbasis arsitektur EfficientNetB7, pelatihan model, serta pengujian akurasi. Model kemudian diintegrasikan ke aplikasi Android melalui API yang dideploy di Google Cloud Run. Hasil pengujian menunjukkan akurasi model sebesar 87% setelah optimasi, dengan kemampuan mengenali 60 jenis makanan populer di Indonesia. Evaluasi oleh delapan ahli gizi dan tenaga medis terhadap output Generative AI API Gemini memperoleh skor rata-rata 4.5 (skala 1-5) untuk aspek keterbacaan, relevansi, konsistensi, dan manfaat bagi pengguna.

Temuan penelitian ini mengindikasikan bahwa kombinasi teknologi machine learning dan generative AI dapat menjadi solusi inovatif untuk meningkatkan kesadaran masyarakat terhadap asupan gula serta mendukung pencegahan diabetes melalui pendekatan yang edukatif dan interaktif. Aplikasi GlucoFit diharapkan dapat berkontribusi dalam perubahan perilaku konsumsi makanan sehat di Indonesia.

Kata kunci: Diabetes melitus, pengenalan gambar, *Convolutional Neural Network (CNN)*, API, fitur aplikasi mobile.

# GLOSARIUM

| | |
|---|---|
| Akurasi | Metrik evaluasi yang menunjukkan persentase prediksi yang benar dari total prediksi yang dibuat oleh model. |
| Application Programming Interface (API) | Antarmuka pemrograman yang memungkinkan dua sistem atau aplikasi berkomunikasi dan bertukar data satu sama lain. |
| Convolutional Neural Network (*CNN*) | Jenis jaringan saraf tiruan yang digunakan terutama untuk analisis citra, mampu mengekstraksi fitur visual dari gambar. |
| Machine Learning | Cabang dari kecerdasan buatan (AI) yang memungkinkan sistem belajar dari data dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit. |
| Data | Kumpulan informasi yang digunakan dalam machine learning untuk melatih, memvalidasi, dan menguji model, termasuk gambar, teks, atau angka yang menjadi dasar pembelajaran pola. |
| Gemini API | API yang menyediakan akses ke model AI Gemini, yang memiliki kemampuan multimodal untuk memproses teks, gambar, audio, dll. |
| Kandungan Gula | Jumlah gula (karbohidrat sederhana) yang terdapat dalam suatu jenis makanan atau minuman. |
| Mobile Application | Aplikasi perangkat lunak yang dirancang untuk berjalan di perangkat mobile seperti smartphone atau tablet. |
| Nutrisi | Zat-zat yang terdapat dalam makanan yang diperlukan tubuh untuk pertumbuhan dan pemeliharaan fungsi tubuh. |

# DAFTAR ISI

HALAMAN J ..... i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ..... ii
HALAMAN PENGESAHAN DOSEN PENGUJI ..... iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR ..... iv
HALAMAN PERSEMBAHAN ..... v
HALAMAN MOTO ..... vi
KATA PENGANTAR ..... vii
SARI ..... ix
GLOSARIUM ..... x
DAFTAR ISI ..... xi
DAFTAR TABEL ..... xiii
DAFTAR GAMBAR ..... xiv
BAB I PENDAHULUAN ..... 1
1.1 Latar Belakang ..... 1
1.2 Rumusan Masalah ..... 6
1.3 Batasan Masalah ..... 6
1.4 Tujuan Penelitian ..... 7
1.5 Sistematika Penulisan ..... 7
BAB II KAJIAN PUSTAKA ..... 9
2.1 Kajian Teori ..... 9
2.1.1 Diabetes ..... 9
2.1.2 *Machine Learning* ..... 10
2.1.3 *Convolutional Neural Network (CNN)* ..... 11
2.1.4 AUC (Area Under the Curve) ..... 13
2.1.5 API (Application Programming Interface) ..... 13
2.2 Kajian Literatur ..... 14
BAB III METODOLOGI PENELITIAN ..... 19
3.1 Pengumpulan Data ..... 21
3.2 Pre-processing (Pengolahan Data) ..... 21
3.2.1 Cleaning Data ..... 21
3.2.2 Penyeimbangan Jumlah Data Antar Kelas ..... 22
3.2.3 Pemisahan Data Training, Validation, dan Testing ..... 22
3.2.4 Augmentasi Data ..... 22
3.3 Perancangan *CNN* ..... 23
3.4 Pelatihan Model ..... 23
3.5 Pengujian Model ..... 23
3.6 Pemasangan Model ke Aplikasi Android ..... 24
3.6.1 Upload Model ke *Cloud Storage* ..... 24
3.6.2 Deploy di *Cloud Run* ..... 24
3.6.3 API Dipakai oleh GlucoFit ..... 24
3.6.4 Pengujian Model pada Aplikasi ..... 24
3.7 Integrasi API pada Aplikasi (*Gemini API*) ..... 25
3.8 Evaluasi Hasil Penelitian ..... 25
BAB IV HASIL DAN PEMBAHASAN ..... 27
4.1 Pengumpulan Data ..... 27
4.2 *Pre-processing* ..... 29
4.2.1 Cleaning Data ..... 29

4.2.2 Penyeimbangan Jumlah Data Antar Kelas ..... 30
4.2.3 Pemisahan Data Training, Validation, dan Testing ..... 30
4.2.4 Augmentasi Data ..... 34
4.3 Perancangan *CNN* ..... 38
4.4 Pelatihan Model ..... 41
4.5 Pengujian Model ..... 43
4.6 Pemasangan Model ..... 49
4.6.1 Upload Model dan Data Gula Makanan ke *Cloud Storage* ..... 49
4.6.2 Deploy di *Cloud* Run ..... 49
4.6.3 *Cloud* Run Menghasilkan API ..... 50
4.6.4 API dipakai oleh GlucoFit ..... 50
4.6.5 Pengujian Model pada Aplikasi ..... 50
4.7 Integrasi API Gemini pada Aplikasi (*Android Studio*) ..... 51
*4.8* Penyesuaian Parameter & *Prompt Generative AI* API *Gemini* ..... 51
4.8.1 Penyesuaian Parameter *Generative AI* API Gemini ..... 51
4.8.2 Penyesuaian *Prompt Generative AI* API Gemini ..... 53
4.9 Penilaian Hasil *Generative AI* oleh Para Ahli ..... 57
BAB V KESIMPULAN DAN SARAN ..... 61
5.1 Kesimpulan ..... 61
5.2 Saran ..... 61
DAFTAR PUSTAKA ..... 63
LAMPIRAN ..... 67

# DAFTAR TABEL

Tabel 1.1 Jumlah Penderita Diabetes di Indonesia ..... 1
Tabel 2.1 Perbandingan Metode dalam Penjelasan Ilmiah ..... 14
Tabel 4.1 Daftar Kelas Makanan ..... 27
Tabel 4.2 Jumlah Data setelah Dibagi ..... 33
Tabel 4.3 Daftar Akurasi Setiap Kelas Makanan pada Pengujian Awal ..... 43
Tabel 4.4 Rentang Peningkatan Akurasi pada Setiap Kelas Makanan ..... 46
Tabel 4.5 pengaruh Parameter *Generatif AI* (*Gemini*) terhadap Karakteristik *Output* ..... 52
Tabel 4.6 Penilaian para ahli ..... 57

# DAFTAR GAMBAR

Gambar 3.1 Proses pembuatan model dan integrasi API.....21
Gambar 3.2 *Prompt Generative AI* pada Glucofit .....25
Gambar 4.1 Daftar Nama Makanan pada Klasifikasi Gambar dalam Kodingan.....31
Gambar 4.2 Pseudocode Struktur Direktori dan Pengumpulan Data Gambar.....32
Gambar 4.3 Pseudocode konversi data ke DataFrame.....32
Gambar 4.4 Pseudocode Pre-processing untuk Data Validasi.....33
Gambar 4. 5 Pseudocode Augmentasi Data Latih .....34
Gambar 4.6 Pseudocode Normalisasi pada Citra.....35
Gambar 4.7 Pseudocode Alur Generator Gambar dari DataFrame (Latih) .....36
Gambar 4.8 Pseudocode Alur Generator Gambar dari DataFrame (Validasi) .....36
Gambar 4.9 Pseudocode Alur Generator Gambar dari DataFrame (Test).....37
Gambar 4.10 Pseudocode Ekstraksi Fitur.....38
Gambar 4.11 Pseudocode Arsitektur Model.....39
Gambar 4.12 Pseudocode Fungsi define_compile_model() untuk Pembuatan dan Kompilasi Model.....40
Gambar 4.13 Pseudocode Inisialisasi Model dan Ringkasan Arsitektur dengan model.summary().....40
Gambar 4.14 Keseluruhan arsitektur model .....41
Gambar 4.15 Pseudocode Pelatihan Model dengan Metode model.fit() .....42
Gambar 4.16 Pseudocode Penyimpanan Model dalam Format .h5 Menggunakan model.save() .....42
Gambar 4.17 Hasil Akurasi pada Pengujian Awal .....43
Gambar 4.18 Akurasi pada Pengujian Akhir.....45
Gambar 4.19 Upload model dan data gula makanan di *Bucket GCP* .....49
Gambar 4.20 Pseudocode Konfigurasi Model *Generative AI* (Gemini) dengan Parameter Kustom.....51
Gambar 4.21 Pseudocode Konfigurasi Model *Generative AI* (Gemini) dengan Parameter Kustom.....52
Gambar 4.22 Pseudocode Fungsi generateScientificExplanation() untuk Pembuatan Prompt *Generative AI*.....53
Gambar 4.23 Ilustrasi deskripsi kelompok makanan bergula rendah (Ramen).....55
Gambar 4.24 Ilustrasi deskripsi kelompok makanan bergula tinggi (Chocolate Cake) .....56

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Diabetes melitus merupakan kondisi kronis yang ditandai dengan peningkatan kadar gula darah (glukosa) di atas batas normal. Penyakit ini tidak hanya memengaruhi kualitas hidup individu, tetapi juga memberikan beban yang signifikan terhadap sistem kesehatan dan ekonomi nasional. Prevalensi diabetes di Indonesia terus mengalami peningkatan dalam beberapa dekade terakhir, sejalan dengan perubahan gaya hidup, pola makan tidak sehat, kurangnya aktivitas fisik, serta peningkatan usia populasi rupiah (Laporan Tematik SKI, 2023).

Sebagai salah satu PTM utama, diabetes melitus kerap ditemukan bersamaan dengan penyakit lain seperti hipertensi, Kondisi ini semakin memperberat risiko komplikasi yang dapat menyebabkan kecacatan bahkan kematian dini. Oleh karena itu, pencegahan dan pengelolaan diabetes secara holistik sangat penting dilakukan untuk menekan angka kesakitan dan kematian akibat penyakit ini (Laporan Tematik SKI, 2023).

Prevalensi diabetes di Indonesia terus meningkat, baik secara nasional maupun global. Data dari International Diabetes Federation (IDF 2025) mengatakan bahwa Indonesia termasuk negara 5 besar dalam masyarakatnya yang pengidap diabetes. Sebagaimana tercantum dalam Tabel 1 , data penderita diabetes di Indonesia menunjukkan tren yang mengkhawatirkan. Pada tahun 2024, jumlah penderita diabetes pada kelompok usia 20–79 tahun mencapai 20,4 juta jiwa, dan diprediksi akan meningkat menjadi 28,6 juta jiwa pada tahun 2050, dengan kenaikan sebesar 8,2 juta jiwa. Kenaikan yang lebih signifikan terjadi pada kelompok usia lanjut (65–99 tahun), yaitu dari 3,4 juta jiwa pada tahun 2024 menjadi 7,8 juta jiwa pada tahun 2050, atau peningkatan sebanyak 4,4 juta jiwa (*International Diabetes Federation, 2025*).

Tabel 1.1 Jumlah Penderita Diabetes di Indonesia

| Kelompok Usia | Jumlah Penderita Diabetes di Indonesia (2024) | Prediksi Jumlah Penderita Diabetes di Indonesia (2050) | Kenaikan (juta jiwa) |
|---|---|---|---|
| 20–79 tahun | 20.4 juta | 28.6 juta | +8.2 juta |
| 65–99 tahun | 3.4 juta | 7.8 juta | +4.4 juta |

Sumber: International Diabetes Federation, 2025

Ditambah proporsi diabetes yang belum terdiagnosis sebanyak 73,2%, artinya dari 100 orang Indonesia yang sebenarnya punya diabetes, sekitar 73 orang belum sadar atau belum tahu kalau mereka sedang sakit, dan jumlah tepatnya adalah 15 juta jiwa yang belum menyadari penyakitnya (International Diabetes Federation, 2025). Hal ini menunjukkan bahwa banyak individu yang tidak menyadari kondisi mereka, sehingga menghambat upaya pencegahan dan pengobatan dini. Penyakit diabetes juga memberikan beban ekonomi yang besar pada negara. Pada tahun 2021, biaya pengobatan diabetes mencapai 17,92 triliun rupiah (Laporan Tematik SKI, 2023).

Faktor-faktor risiko seperti gaya hidup yang kurang aktif, pola makan tidak seimbang, usia yang semakin tua, obesitas, hipertensi, serta riwayat genetika merupakan beberapa penyebab utama meningkatnya jumlah kasus diabetes (Komiljonova, 2024; Oktora & Butar Butar, 2022). Selain itu, rendahnya kesadaran masyarakat akan kondisi kesehatannya juga turut menjadi penghambat dalam upaya pencegahan dan pengendalian penyakit diabetes.

Menjawab tantangan tersebut, penelitian ini bertujuan menghadirkan solusi inovatif yang mudah diakses oleh masyarakat guna mendukung pencegahan penyakit diabetes. Solusi yang ditawarkan berfokus pada dua faktor utama yang memengaruhi risiko terkena diabetes, yaitu pola makan yang tidak sehat dan gaya hidup dengan aktivitas fisik yang rendah Laporan Tematik SKI, 2023; Olimjonovna, 2024) .

Pola makan yang tidak sehat menjadi salah satu faktor utama penyebab diabetes, terutama diabetes tipe 2 (Olimjonovna, 2024.). Konsumsi makanan tinggi gula dan karbohidrat olahan secara terus-menerus dapat menyebabkan peningkatan kadar gula darah yang berlebihan, sehingga memicu resistensi insulin pada tubuh. Dengan mengedukasi masyarakat untuk lebih sadar akan asupan makanan—terutama kandungan gulanya—maka risiko terjadinya gangguan metabolisme tersebut dapat dikurangi secara signifikan. Oleh karena itu, solusi berupa aplikasi pemantau asupan gula seperti GlucoFit sangat relevan dalam memberikan informasi real-time tentang kandungan gula dalam makanan, serta mendorong masyarakat untuk membuat pilihan makanan yang lebih sehat.

Di sisi lain, gaya hidup dengan aktivitas fisik yang rendah juga turut meningkatkan risiko diabetes karena kurangnya penggunaan glukosa oleh tubuh sebagai sumber energi (Olimjonovna, 2024). Olahraga teratur dapat meningkatkan sensitivitas insulin, membantu menurunkan berat badan, serta memperbaiki metabolisme tubuh secara keseluruhan. Dengan memberikan rekomendasi aktivitas fisik yang sesuai kondisi pengguna, solusi ini tidak hanya

membantu mencegah diabetes tetapi juga meningkatkan kualitas hidup secara umum. Kombinasi antara perbaikan pola makan dan peningkatan aktivitas fisik menjadi pendekatan holistik yang efektif dalam upaya pencegahan diabetes, terutama di kalangan individu berisiko tinggi.

Salah satu strategi utama dalam penelitian ini adalah meningkatkan kesadaran masyarakat akan kandungan gula dalam makanan yang mereka konsumsi. Sebagai solusinya, dikembangkan sebuah fitur AI dalam aplikasi berbasis mobile yang memanfaatkan teknologi pengenalan citra (*image recognition*). Dengan fitur ini, pengguna dapat mengambil foto makanan menggunakan kamera ponsel, dan aplikasi secara otomatis akan menampilkan informasi mengenai kadar gula dari makanan tersebut. Selain itu, aplikasi juga menyediakan penjelasan ilmiah mengenai komposisi gula serta rekomendasi aktivitas fisik yang bisa dilakukan untuk membakar asupan gula yang telah dikonsumsi. Melalui pendekatan ini, diharapkan pengguna lebih sadar terhadap jumlah gula yang mereka konsumsi setiap hari, sehingga mampu membuat pilihan makanan yang lebih sehat dan mendukung pencegahan diabetes.

Salah satu pendekatan yang digunakan dalam penelitian ini adalah pemanfaatan teknologi *machine learning*. Sebelumnya, telah banyak penelitian yang menerapkan *machine learning* dalam bidang kesehatan, meskipun sebagian besar fokus pada diagnosis awal dan prediksi risiko penyakit. Beberapa contoh penerapan *machine learning* yang berhasil di bidang kesehatan antara lain prediksi risiko kematian pada pasien COVID-19, deteksi kesalahan resep obat, serta perkiraan efektivitas obat baru (Alanazi, 2022).

Bahkan, beberapa penelitian telah menerapkan *machine learning* dalam pengelolaan penyakit diabetes. Hasil penelitian sebelumnya menunjukkan bahwa teknologi *machine learning* memiliki kemampuan yang lebih baik dibandingkan metode konvensional dalam memprediksi risiko diabetes (Alanazi, 2022; Rayan et al., 2019). Selain meningkatkan ketepatan diagnosis, teknologi ini juga berperan dalam pengelolaan kesehatan dengan mendeteksi kemungkinan komplikasi sejak dini. Selain itu, *machine learning* dapat dijadikan sebagai platform edukasi yang membantu meningkatkan pemahaman dan kesadaran masyarakat terhadap faktor-faktor risiko penyakit diabetes (Jamal Khanam & Foo, 2021).

*Machine learning* telah terbukti efektif dalam penanganan diabetes, terutama dalam prediksi risiko dan deteksi dini penyakit. Dengan kemampuannya memproses data dalam jumlah besar, model atau algoritma seperti *Naïve Bayes, Random Forest, K-Nearest Neighbor*

*(KNN),* dan *Decision Tree* berhasil mengidentifikasi pasien dengan risiko tinggi secara akurat (Dagliati et al., 2017; Fatmawati et al., 2023; Lai et al., 2019; Pinandito, Wicaksono and Wijoyo, 2023; Sharma and Shah, 2021; Zou et al., 2018). Penggunaan parameter utama seperti kadar glukosa saja ternyata sudah dapat memberikan hasil yang cukup baik, meskipun performa model akan semakin optimal ketika dilengkapi dengan lebih banyak fitur. Sejalan dengan perkembangan teknologi, *machine learning* berpotensi memberikan solusi inovatif untuk meningkatkan pelayanan kesehatan serta mengurangi dampak jangka panjang penyakit diabetes. *Convolutional Neural Network* (CNN) sebagai salah satu pendekatan machine learning juga pernah dimanfaatkan dalam berbagai aplikasi kesehatan, termasuk analisis kandungan nutrisi makanan. Beberapa penelitian terdahulu telah berhasil mengimplementasikan CNN untuk estimasi nilai kalori dari citra makanan (Ardiansyah et al., 2025; Irawan, 2024; Pawening et al., 2023). Penerapan teknologi ini memungkinkan pengguna untuk memantau asupan nutrisi secara lebih akurat, sehingga dapat mendukung penerapan pola hidup sehat melalui pengaturan makan yang lebih terukur.

Meski telah banyak dimanfaatkan, penerapan machine learning dalam penelitian terdahulu masih memiliki beberapa keterbatasan dalam cakupan analisisnya. Mayoritas penelitian berfokus pada dua hal utama: (1) memprediksi status diabetes (mengategorikan penderita dan non-penderita) atau (2) mendeteksi kandungan kalori dari gambar makanan, tanpa memperkirakan kadar gula. Dengan demikian, meskipun teknologi ini mampu mengidentifikasi risiko diabetes dan menganalisis nutrisi dari gambar makanan, masih sedikit peneliti yang mengembangkan machine learning sebagai sarana untuk membantu masyarakat mencegah diabetes. Salah satu contoh nyata adalah sistem berbasis perangkat wearable yang dikombinasikan dengan machine learning—walaupun efektif dalam memantau kadar glukosa darah (Makroum et al., 2022), fungsinya masih terbatas pada pemantauan pasif.

Dalam penelitian ini, *CNN* dimanfaatkan untuk mengenali jenis makanan dari citra yang diunggah serta menampilkan kadar gula di dalamnya, sehingga membantu pengguna dalam memantau asupan gula sebagai langkah pencegahan dan pengelolaan diabetes. Pengguna hanya perlu mengunggah foto makanan, lalu sistem akan secara otomatis mengidentifikasi jenis makanan dan memberikan estimasi kandungan gulanya berdasarkan basis data nutrisi. *CNN* dipilih karena kemampuannya dalam mengekstrak fitur visual seperti warna, tekstur, dan bentuk makanan, yang sangat penting untuk mencapai akurasi tinggi dalam klasifikasi citra. Sebelumnya, telah ada penelitian yang menggunakan dataset yang sama dengan model

berbeda, namun hasilnya menunjukkan potensi besar dalam meningkatkan ketepatan identifikasi makanan dan estimasi nilai gizi melalui analisis citra (Salim et al., 2021).

Contoh lain pemanfaatan *CNN* dapat dilihat pada penelitian yang dilakukan oleh Hawibowo dan Muhimmah (2023), yang berhasil mencapai akurasi sebesar 96,97% dalam mengklasifikasikan tingkat kematangan buah pepaya (Sayyidin, H., & Muhimmah, 2024). Selain itu, penelitian oleh Azhari et al. (2023) juga memanfaatkan arsitektur berbasis *CNN*, yaitu YOLOv8, untuk mendeteksi trikoma pada daun kentang dengan tingkat presisi mencapai 98% (Fauzan Azhari et al., 2024). Pendekatan-pendekatan tersebut menunjukkan potensi besar dari teknologi *CNN* dalam pengenalan objek, yang dapat dimodifikasi dan diterapkan untuk mendeteksi serta menganalisis jenis makanan.

*API (Application Programming Interface)* memegang peran penting dalam mendukung interoperabilitas antar berbagai sistem kesehatan. Berdasarkan studi yang dilakukan oleh konen dan Rudin, penggunaan API tidak hanya mampu meningkatkan efisiensi operasional, tetapi juga mempermudah pertukaran data secara aman dan akurat di antara platform kesehatan yang berbeda (Gordon & Rudin, 2022). Dalam aplikasi GlucoFit, *API Gemini* dimanfaatkan untuk memberikan penjelasan ilmiah mengenai proses pencernaan makanan dan pengaruhnya terhadap kadar gula darah. Pemanfaatan teknologi ini sejalan dengan perkembangan global yang semakin fokus pada personalisasi layanan kesehatan melalui integrasi API.

Pemanfaatan *API* dalam sistem kesehatan bukan hanya sekadar teori, tetapi telah terbukti memberikan dampak nyata di lapangan. Berdasarkan hasil penelitian Koneni (2024), pasien yang menggunakan layanan berbasis API menunjukkan tingkat kepatuhan terhadap rencana pengobatan 45% lebih tinggi dibandingkan dengan yang tidak menggunakan, menunjukkan bahwa integrasi teknologi ini mampu meningkatkan keterlibatan dan kesadaran pengguna terhadap kesehatan mereka. Selain itu, studi oleh Aarsæther (2024) juga menegaskan bahwa API efektif meningkatkan efisiensi dalam pelayanan kesehatan modern melalui pertukaran data yang cepat dan akurat. Dalam konteks penelitian ini, API tidak hanya berfungsi sebagai jembatan antara model *machine learning* dengan sumber data eksternal untuk mengidentifikasi kandungan gula dalam makanan, tetapi juga sebagai alat untuk memberikan rekomendasi aktivitas fisik yang personal dan relevan. Untuk memperkaya informasi yang diberikan kepada pengguna, aplikasi GlucoFit mengintegrasikan API *LLM (Large Language Model*), yaitu *Google Gemini*, yang mampu menghasilkan penjelasan faktual dengan tingkat akurasi tinggi (Panchal et al., 2024). Penggunaan *API Gemini* ini memungkinkan sistem untuk

menyampaikan informasi secara interaktif, edukatif, dan mudah dipahami oleh masyarakat awam—sejalan dengan temuan Hau, Hassan & Zhou (2025) yang menyebutkan peran *API LLM* dalam meningkatkan interaktivitas aplikasi melalui pembuatan konten otomatis. Dengan demikian, integrasi *API* pada sistem berbasis *machine learning* seperti GlucoFit bukan hanya menjawab kebutuhan teknis, tetapi juga menjadi kunci untuk menciptakan solusi kesehatan yang lebih efektif, personal, dan mudah diakses oleh masyarakat luas.

Dengan infrastruktur API dan *machine learning* yang solid, langkah selanjutnya adalah memastikan bahwa fitur utama aplikasi, seperti prediksi jenis makanan dan estimasi kandungan gula, dapat berjalan secara akurat dan responsif sesuai harapan pengguna. pendekatan baru seperti aplikasi GlucoFit yang mengintegrasikan *machine learning* untuk memantau asupan gula secara real-time menjadi sangat penting. Secara keseluruhan, penelitian ini diawali dengan identifikasi masalah terkait kesadaran masyarakat terhadap asupan gula, dilanjutkan dengan pengumpulan data, pengembangan model *machine learning*, integrasi *API*, hingga pengujian fitur aplikasi. Sehubungan dengan hal tersebut, maka ditetapkan judul penelitian dalam skripsi ini adalah "Fitur Machine Learning Untuk Mengidentifikasi Kandungan Gula Pada Makanan: Studi Kasus Aplikasi Glucofit Untuk Pencegahan Diabetes di Indonesia"

## 1.2 Rumusan Masalah

Dengan merujuk pada konteks latar belakang yang telah diuraikan di atas, hal-hal berikut akan menjadi fokus utama dalam penelitian ini:

1. Bagaimana mengimplementasikan fitur *mobile* untuk menampilkan kandungan gula dalam makanan dan memberikan penjelasan ilmiah tentang kandungan gula tersebut?
2. Bagaimana fitur berbasis *Machine Learning* dapat membantu masyarakat dan penderita diabetes di Indonesia agar bisa mencegah penyakit diabetes?
3. Apakah penggunaan *API Gemini* dalam menghasilkan penjelasan mendapat apresiasi positif dari para ahli di bidang yang kompeten pada penyakit diabetes?

## 1.3 Batasan Masalah

Untuk menjaga fokus penelitian dan mencegah penyimpangan dari tujuan utama, penelitian ini akan dibatasi dengan hal-hal berikut:

1. Objek makanan yang digunakan dalam penelitian ini terbatas pada jenis-jenis makanan yang umum dikonsumsi oleh masyarakat Indonesia, dengan pengecualian terhadap makanan tradisional tertentu seperti gudeg.

2. Data yang digunakan dalam penelitian ini diambil dari sumber terbuka di *Kaggle* dengan dataset berjudul *Food-101*.
3. Data yang akan dianalisis diperoleh melalui kuisioner yang diberikan kepada tenaga kesehatan yang berkompeten dalam bidang diabetes.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengembangkan model berbasis citra (*image-based*) yang mampu memprediksi jenis makanan, kemudian mengintegrasikan hasil prediksi tersebut dengan database informasi gizi (khususnya kandungan gula) yang telah dikumpulkan sebelumnya.
2. Memberikan penjelasan ilmiah mengenai dampak kandungan gula yang terdeteksi terhadap kesehatan, rekomendasi aktivitas fisik/olahraga, dan penyajian informasi melalui pendekatan komunikasi kesehatan yang persuasive.
3. Mengevaluasi performa model dan kualitas informasi yang dihasilkan melalui pengujian akurasi serta penilaian oleh ahli gizi dan tenaga medis.

## 1.5 Sistematika Penulisan

Struktur penulisan yang digunakan dalam penyusunan tugas akhir ini dapat dijelaskan sebagai berikut:

**BAB I PENDAHULUAN**

Bagian pertama ini menjelaskan dengan rinci latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta panduan penulisan.

**BAB II KAJIAN PUSTAKA**

Bab ini membahas berbagai teori yang menjadi landasan penelitian, termasuk konsep-konsep terkait dengan Diabetes, Gula, *API*, serta penerapan *Machine Learning* dalam bidang kesehatan. Kajian pustaka ini berperan sebagai dasar teoritis yang memperkuat argumentasi dan analisis yang dilakukan dalam penelitian ini.

**BAB III METODOLOGI**

Bab ini menjelaskan pendekatan yang diterapkan dalam penelitian, mencakup proses pengumpulan data, analisis data, metode pengembangan model serta implementasi *API*. Uraian ini mencakup tahapan yang diambil dari awal hingga akhir penelitian, serta perangkat dan teknik yang digunakan.

**BAB IV HASIL DAN PEMBAHASAN**

Bab ini menjelaskan tahapan pembuatan fitur berupa implementasi model & API yang telah dirancang dalam penelitian ini. Selain itu, bab ini juga mencakup pengujian terhadap model yang telah diterapkan, termasuk penilaian terhadap keberhasilan implementasi dan analisis hasil pengujian yang dilakukan.

**BAB IV KESIMPULAN DAN SARAN**

Bab ini menyajikan kesimpulan dari keseluruhan penelitian yang telah dilakukan, yang mencakup jawaban atas pertanyaan penelitian dan pencapaian tujuan yang telah ditetapkan. Selain itu, bab ini juga memberikan rekomendasi yang dapat digunakan untuk penelitian selanjutnya atau penerapan fitur yang telah dikembangkan.

# BAB II
# KAJIAN PUSTAKA

## 2.1 Kajian Teori

Penelitian ini bertujuan untuk mengembangkan sebuah fitur dalam aplikasi Android yang mampu mendeteksi berbagai jenis makanan dan memberikan informasi mengenai kadar gula yang terkandung di dalamnya. Informasi tersebut dilengkapi dengan penjelasan ilmiah mengenai asal-usul kandungan gula serta saran aktivitas fisik yang dapat membantu menurunkan kadar gula dalam tubuh. Proses deteksi jenis makanan dilakukan melalui pendekatan *Convolutional Neural Network (CNN)*, yang telah terbukti efektif dalam pengolahan citra digital. Fitur ini memiliki potensi besar untuk diaplikasikan dalam bidang teknologi kesehatan, memanfaatkan kecerdasan buatan guna menyediakan informasi yang cepat, akurat, dan mudah dipahami oleh masyarakat awam. Dengan adanya fitur ini, diharapkan masyarakat dapat lebih sadar akan asupan gula harian mereka sehingga dapat menerapkan pola hidup sehat dan mencegah risiko penyakit yang terkait dengan konsumsi gula berlebih, seperti diabetes melitus. Melalui inovasi ini, diharapkan dapat tercipta perubahan positif dalam kebiasaan konsumsi makanan masyarakat serta memberikan kontribusi signifikan terhadap peningkatan kualitas kesehatan secara nasional.

### 2.1.1 Diabetes

Diabetes melitus merupakan gangguan metabolik kronis yang ditandai dengan peningkatan kadar glukosa darah secara persisten (hiperglikemia). Kondisi ini terjadi akibat kerusakan fungsi insulin, baik dari segi sekresi maupun aktivitas biologisnya. Insulin adalah hormon yang diproduksi oleh sel beta pankreas dan berperan penting dalam pengaturan metabolisme karbohidrat, lemak, dan protein di dalam tubuh (Olimjonovna, 2024).

Berdasarkan mekanisme patofisiologisnya, diabetes melitus dibagi menjadi dua jenis utama, yaitu:

1. Diabetes Tipe 1:

   Merupakan bentuk diabetes autoimun di mana sistem imun tubuh menyerang dan menghancurkan sel beta pankreas yang bertugas memproduksi insulin. Hal ini menyebabkan tubuh tidak mampu menghasilkan insulin. Diabetes tipe 1 biasanya didiagnosis pada masa anak-anak atau remaja, meskipun dapat juga terjadi pada usia dewasa (Olimjonovna, 2024).

2. Diabetes Tipe 2:

   Bentuk diabetes ini lebih umum ditemukan dan biasanya berkaitan erat dengan gaya

hidup serta faktor lingkungan. Pada diabetes tipe 2, tubuh mengalami resistensi terhadap insulin atau tidak mampu memproduksi cukup insulin untuk mempertahankan kadar gula darah normal. Penyakit ini umumnya terjadi pada individu dewasa di atas usia 45 tahun, tetapi semakin banyak kasus yang ditemukan pada kelompok usia lebih muda karena meningkatnya prevalensi obesitas dan kurangnya aktivitas fisik (Olimjonovna, 2024).

Penyebab utama diabetes melitus bervariasi tergantung pada jenisnya.

1. Pada diabetes tipe 1, penyebab utamanya adalah kerusakan autoimun pada sel beta pankreas yang menghasilkan insulin. Faktor genetik dan paparan virus tertentu diduga memicu respons autoimun tersebut (Olimjonovna, 2024).
2. Pada diabetes tipe 2, penyebab utamanya adalah kombinasi antara resistensi insulin dan defisiensi relatif produksi insulin. Resistensi insulin sering kali dipengaruhi oleh penumpukan lemak visceral, pola makan tinggi gula dan lemak jenuh, serta rendah aktivitas fisik (Olimjonovna, 2024).

### 2.1.2 *Machine Learning*

*Machine learning* merupakan salah satu cabang dari *Artificial Intelligence (AI)* yang bertujuan untuk membangun model komputer yang mampu belajar secara mandiri serta membuat prediksi atau keputusan berdasarkan data yang diberikan. Model-model *machine learning* terus meningkatkan akurasi hasilnya melalui proses pembelajaran dari data yang tersedia, tanpa perlu diprogram secara eksplisit untuk menyelesaikan suatu tugas tertentu (Kufel et al., 2023).

Istilah "*machine learning*" pertama kali diperkenalkan oleh Arthur Samuel pada tahun 1959 sebagai kemampuan komputer untuk belajar tanpa harus diprogram ulang secara langsung untuk setiap keterampilan baru. Dengan menggunakan dataset yang tersedia, algoritma *machine learning* yang didukung oleh model matematis dapat menghasilkan prediksi atau keputusan spesifik (Kufel et al., 2023).

Secara umum, terdapat dua jenis utama dalam *machine learning*, yaitu:

**a.** Pembelajaran Terawasi **(***Supervised Learning***)**

Pada pembelajaran terawasi, model dilatih menggunakan dataset yang sudah dilengkapi dengan label atau nilai output yang sesuai. Dengan demikian, model dapat mempelajari hubungan antara data input dan output, sehingga mampu melakukan klasifikasi terhadap data baru atau membuat prediksi (Kufel et al., 2023). Pembelajaran terawasi dibagi menjadi dua metode utama, yaitu:

1. Klasifikasi: digunakan untuk memprediksi kategori atau kelas dari suatu data. Contohnya adalah pengenalan tulisan tangan, di mana sistem mencocokkan karakter tulisan tangan (data input) dengan huruf cetak (kelas output).
2. Regresi: digunakan untuk memprediksi nilai kontinu berdasarkan data input. Misalnya, dalam bidang kesehatan, regresi dapat digunakan untuk memperkirakan biaya pengobatan berdasarkan harga obat, biaya alat medis, dan gaji tenaga medis.

**b.** Pembelajaran Tak Terawasi **(***Unsupervised Learning***)**

Berbeda dengan pembelajaran terawasi, pembelajaran tak terawasi tidak menggunakan data yang telah diberi label. Model belajar hanya dari data input tanpa ada nilai keluaran yang diharapkan. Tujuan utamanya adalah mengidentifikasi pola atau struktur tersembunyi dalam data (Kufel et al., 2023). Pembelajaran tak terawasi terbagi menjadi dua jenis utama, yaitu:

1. *Clustering*: membagi data ke dalam kelompok-kelompok berdasarkan kesamaan atau perbedaan karakteristik. Contohnya adalah pengelompokan hewan berdasarkan ciri-ciri visual yang diekstraksi oleh model.
2. *Association*: menganalisis hubungan antar elemen dalam sebuah dataset. Metode ini sering digunakan dalam strategi pemasaran, seperti menemukan asosiasi antara produk yang sering dibeli bersama.

### 2.1.3 *Convolutional Neural Network (CNN)*

*Convolutional Neural Network (CNN)* merupakan salah satu jenis utama jaringan saraf tiruan (*artificial neural network*) dalam subset *machine learning*, yang dirancang secara khusus untuk memproses data dengan struktur spasial seperti gambar atau citra. Sebagai bagian dari pendekatan *deep learning*, CNN memanfaatkan arsitektur berlapis untuk mengekstraksi fitur visual secara otomatis, menjadikannya sangat efektif dalam tugas-tugas pengenalan pola berbasis citra. *CNN* digunakan secara luas dalam bidang visi komputer, terutama untuk tugas pengenalan objek, klasifikasi gambar, dan deteksi pola visual. *CNN* mampu belajar representasi fitur secara otomatis dari data mentah melalui proses hierarkis. Dengan demikian, *CNN* tidak hanya meningkatkan efisiensi dalam ekstraksi fitur, tetapi juga meningkatkan akurasi sistem klasifikasi atau deteksi. *CNN* bekerja dengan cara mengaplikasikan operasi konvolusi pada citra masukan menggunakan filter atau kernel yang dipelajari selama proses pelatihan. Proses ini memungkinkan lapisan-lapisan *CNN* untuk mengekstrak fitur-fitur yang semakin kompleks

pada tingkat yang lebih tinggi, sehingga model dapat mengenali pola bahkan dalam kondisi adanya distorsi atau pergeseran posisi objek (Taye, 2023).

Arsitektur dasar *CNN* terdiri dari beberapa lapisan penting, yaitu lapisan konvolusi, lapisan *pooling*, fungsi aktivasi non-linear, dan lapisan terhubung penuh (*fully connected layer*). Lapisan konvolusi bertugas untuk mengekstrak fitur dari gambar dengan menggunakan filter yang diaplikasikan secara berulang-ulang. Setelah itu, lapisan pooling digunakan untuk mereduksi dimensi data sekaligus menjaga invariansi terhadap translasi. Fungsi aktivasi seperti *ReLU* diterapkan untuk memperkenalkan nonlinearitas ke dalam model, sehingga model dapat menangkap hubungan yang lebih kompleks antara fitur input dan output. Di bagian akhir, lapisan fully connected digunakan untuk melakukan klasifikasi berdasarkan fitur yang telah diekstrak sebelumnya. Selain itu, teknik seperti padding dan stride digunakan untuk mengontrol ukuran output serta mengurangi hilangnya informasi tepi gambar selama proses konvolusi (Taye, 2023).

Salah satu kelebihan utama *CNN* dibandingkan jaringan saraf tiruan biasa adalah kemampuannya dalam mempertahankan struktur spasial dari data gambar. Hal ini membuat *CNN* sangat efektif dalam menangani data visual berdimensi tinggi seperti foto, video, maupun citra medis. Selain itu, *CNN* menggunakan konsep weight sharing, di mana setiap neuron dalam lapisan konvolusi menggunakan parameter yang sama untuk seluruh area gambar, sehingga jumlah parameter yang harus dipelajari menjadi lebih sedikit. Konsep ini tidak hanya mempercepat proses pelatihan, tetapi juga membantu mengurangi risiko *overfitting*. Teknik regularisasi seperti *dropout*, *batch normalization*, dan *data augmentation* juga sering kali diterapkan untuk meningkatkan generalisasi model *CNN* (Taye, 2023).

*CNN* telah banyak digunakan dalam berbagai aplikasi nyata, termasuk pengenalan wajah, analisis citra medis, sistem pengawasan keamanan, pengenalan karakter optik (OCR), hingga segmentasi gambar. Beberapa arsitektur *CNN* populer seperti *LeNet*, *AlexNet*, *VGG*, *GoogLeNet*, *ResNet*, dan *DenseNet* telah dikembangkan untuk meningkatkan performa dalam berbagai tugas visi komputer. Dengan perkembangan teknologi komputasi dan tersedianya dataset besar yang dilabeli, *CNN* terus menjadi salah satu metode paling efektif dalam bidang pembelajaran mendalam (*deep learning*). Meskipun begitu, tantangan seperti kebutuhan akan sumber daya komputasi yang tinggi dan kompleksitas model masih menjadi fokus penelitian untuk pengembangan *CNN* yang lebih efisien dan ringkas di masa depan (Taye, 2023).

### 2.1.4 AUC (Area Under the Curve)

AUC atau Area Under the Curve adalah salah satu metrik evaluasi yang digunakan untuk mengukur seberapa baik suatu model klasifikasi mampu membedakan antara dua kelas, misalnya antara kelas positif dan kelas negatif AUC dihitung berdasarkan luas di bawah kurva ROC (Receiver Operating Characteristic), yang menunjukkan hubungan antara tingkat deteksi positif (True Positive Rate) dan tingkat kesalahan positif (False Positive Rate) pada berbagai ambang batas prediksi. Nilai AUC berkisar dari 0 hingga 1, di mana semakin mendekati 1, semakin baik kemampuan model dalam membedakan kedua kelas tersebut. Kelebihan utama AUC adalah bahwa ia tidak terlalu dipengaruhi oleh ketidakseimbangan jumlah data antar kelas, sehingga sangat berguna dalam kasus seperti deteksi penyakit langka atau penipuan. Selain itu, AUC juga tidak bergantung pada satu ambang batas tertentu, melainkan menilai kinerja model secara keseluruhan di semua kemungkinan ambang batas, sehingga memberikan gambaran yang lebih menyeluruh tentang kualitas model. (Yang & Ying, 2022)

### 2.1.5 API (Application Programming Interface)

API atau *Application Programming Interface* dapat didefinisikan sebagai sebuah antarmuka yang memungkinkan komunikasi antara aplikasi atau sistem secara programatis. Sebagai lapisan abstraksi, API bertindak sebagai penghubung yang memfasilitasi interaksi antara suatu perangkat lunak dengan pihak lain, baik itu aplikasi lain maupun komponen system (Pedro, 2024).

Fungsi utama API adalah untuk menyederhanakan dan memstandardisasi cara suatu aplikasi diakses oleh entitas eksternal. Dengan demikian, API menjadi sarana penting dalam proses integrasi antarsistem. Integrasi tersebut memungkinkan pengembang untuk menyusun suatu produk dengan memanfaatkan bagian-bagian fungsional yang telah tersedia, tanpa harus selalu membangun segala sesuatunya dari nol (Pedro, 2024).

Alih-alih membangun seluruh fitur dari awal, pengembang dapat menggunakan fungsionalitas yang sudah tersedia dalam bentuk API. Hal ini menjadikan API sebagai alat yang sangat powerful dalam pengembangan perangkat lunak modern. Pemanfaatan API memberikan keuntungan besar dalam hal efisiensi waktu dan sumber daya pengembangan. Fungsi yang ditawarkan melalui API bisa mencakup keseluruhan fitur aplikasi, satu modul tertentu, atau kombinasi sejumlah fungsi yang dipilih sesuai kebutuhan pengembangan (Pedro, 2024).

## 2.2 Kajian Literatur

Kajian literatur ini bertujuan untuk membandingkan berbagai metode yang digunakan dalam memberikan penjelasan ilmiah, dengan merujuk pada sejumlah penelitian sebelumnya yang relevan. Melalui perbandingan tersebut, diharapkan dapat diperoleh gambaran mengenai pendekatan-pendekatan yang telah dikembangkan dan diterapkan dalam konteks serupa, sehingga memberikan dasar teoritis serta praktis bagi penelitian saat ini.

Tabel 2.1 Perbandingan Metode dalam Penjelasan Ilmiah

| Metode | Kelebihan | Kekurangan | Referensi |
|---|---|---|---|
| **Penjelasan Manual/*Predefined*** | 1.Tidak bergantung pada koneksi internet. | 1. Waktu pengumpulan dan pembaruan data bisa sangat memakan waktu.<br>2.Perlu banyak pemeliharaan jika data baru ditemukan.<br>3.Rentan terhadap kesalahan manusia.<br>4. biaya jangka panjang seperti tenaga kerja, pelatihan, dan koreksi kesalahan akan lebih mahal. | Martinez, J. (2025). *Manual data entry vs automated data entry: Key differences*. DocuClipper. https://www.docuclipper.com/blog/manual-data-entry-vs-automated-data-entry/#:~:text=Automated%20systems%20offer%20long%2Dterm,and%20less%20efficient%20over%20time |
| ***Knowledge Graph* (Graf Pengetahuan)** | 1. Dapat memberikan penjelasan yang kontekstual dan berbasis hubungan antar elemen makanan.<br>2. Dapat menjelaskan aspek lain selain gula, seperti kalori, protein, dll. | 1. Pembuatannya rumit dan memerlukan pemahaman mendalam.<br>2. Dapat membutuhkan waktu dan sumber daya yang cukup besar.<br>3. Membutuhkan pemeliharaan dan pembaruan berkala. | Hogan A, Blomqvist E, Cochez M, D'amato C, Melo G de, Gutierrez C, et al. Knowledge Graphs. ACM Comput Surv [Internet]. 2021 Jul;54(4). Available from: https://doi.org/10.1145/3447772 |

| | | | |
|---|---|---|---|
| | 3. Memudahkan penelusuran data terkait secara dinamis. | | |
| **Sistem Pakar (*Expert System*)** | 1. Dapat memberikan penjelasan berdasarkan pengetahuan ahli yang terstruktur.<br>2. Dapat menangani penjelasan kompleks yang melibatkan banyak faktor.<br>3. Dapat dibuat lebih interaktif. | 1. Membutuhkan pengumpulan dan pemrograman pengetahuan ahli.<br>2. Memerlukan pembaruan berkelanjutan untuk mencakup pengetahuan terbaru.<br>3. Implementasi bisa memakan waktu dan mahal. | Adani, M. R. (2021). Sistem pakar. Sekawan Media. https://www.sekawanmedia.co.id/blog/sistem-pakar/ |
| **Asisten Virtual (*Chatbot*)** | 1. Bisa memberikan penjelasan secara interaktif dengan pengguna<br>2. Dapat diintegrasikan dengan aplikasi secara langsung.<br>3. Memberikan pengalaman berbicara yang lebih alami. | 1. Membutuhkan pengembangan dan pemrograman chatbot yang rumit.<br>2. Pengguna mungkin tidak selalu puas dengan jawaban chatbot yang terbatas.<br>3. Perlu pengawasan dan pembaruan untuk mempertahankan kualitas jawaban. | Adamopoulou E, Moussiades L. An Overview of Chatbot Technology. In: Maglogiannis I, Iliadis L, Pimenidis E, editors. Artificial Intelligence Applications and Innovations. Cham: Springer International Publishing; 2020. p. 373–83. |

| | | | |
|---|---|---|---|
| **Menggunakan API Eksternal (Gemini API)** | 1. Memungkinkan penjelasan berbasis data yang lebih kaya dan terpercaya dari sumber ilmiah<br>2. Mudah diintegrasikan dengan aplikasi.<br>3. Mempercepat proses pengembangan. | 1. Bergantung pada koneksi internet.<br>2. Biaya berlangganan API (jika ada).<br>3. Tidak sepenuhnya fleksibel dalam hal penyesuaian penjelasan. | Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, A., Chen, J., Pitler, E., Lillicrap, T., Lazaridou, A., … Vinyals, O. (2025). *Gemini: A Family of Highly Capable Multimodal Models*. https://arxiv.org/abs/2312.11805<br>Gordon WJ, Rudin RS. Why APIs? Anticipated value, barriers, and opportunities for standards-based application programming interfaces in healthcare: perspectives of US thought leaders. JAMIA Open [Internet]. 2022 Jul 1;5(2):ooac023.<br>Kufel, J., Bargieł-Łączek, K., Kocot, S., Koźlik, M., Bartnikowska, W., Janik, M., Czogalik, Ł., Dudek, P., Magiera, M., Lis, A., Paszkiewicz, I., Nawrat, Z., Cebula, M., & Gruszczyńska, K. (2023). What Is *Machine Learning*, Artificial Neural Networks and *Deep Learning*?—Examples of Practical Applications in Medicine. Diagnostics, 13(15), 2582. |

| | | | |
|---|---|---|---|
| | | | https://doi.org/10.3390/diagnostics13152582 |

Penggunaan API (*Application Programming Interface*) seperti *Gemini* dapat menutupi kekurangan-kekurangan dari metode lain yang telah dijabarkan di tabel 2.1. terutama dalam konteks interaksi dan pemrosesan informasi secara efektif. API *Gemini* menawarkan kemampuan untuk mengakses kecerdasan buatan yang sangat canggih, yang memungkinkan untuk memberikan jawaban interaktif dan kontekstual secara dinamis, mengatasi beberapa keterbatasan yang ada pada metode lain.

Salah satu kelemahan utama pada metode manual atau predefined adalah kebutuhan untuk pengumpulan dan pembaruan data yang memakan waktu serta rentannya terhadap kesalahan manusia. Dengan menggunakan API, pembaruan data dapat dilakukan secara otomatis dan real-time, mengurangi beban kerja manual dan meminimalkan risiko kesalahan. API dapat langsung terhubung dengan sumber data terbaru, sehingga proses pembaruan informasi menjadi lebih efisien dan terkini, tanpa memerlukan pemeliharaan data yang rumit.

Pada metode **Knowledge Graph**, meskipun menawarkan kemampuan untuk memberikan penjelasan berbasis hubungan antar elemen dan data terkait, pembuatan dan pemeliharaan graf ini memerlukan pemahaman mendalam dan sumber daya yang cukup besar. API *Gemini*, di sisi lain, dapat memberikan penjelasan kontekstual yang berbasis hubungan antar elemen tanpa memerlukan pembangunan graf yang rumit. Dengan integrasi API, informasi dapat ditarik dan disampaikan dengan lebih mudah, mengurangi kompleksitas pengelolaan graf pengetahuan yang cenderung memakan waktu dan biaya. Selain itu, API juga dapat secara otomatis meng-

update data dan hubungan yang relevan, menghindari masalah pembaruan berkala yang menjadi kelemahan utama dalam *Knowledge Graph*.

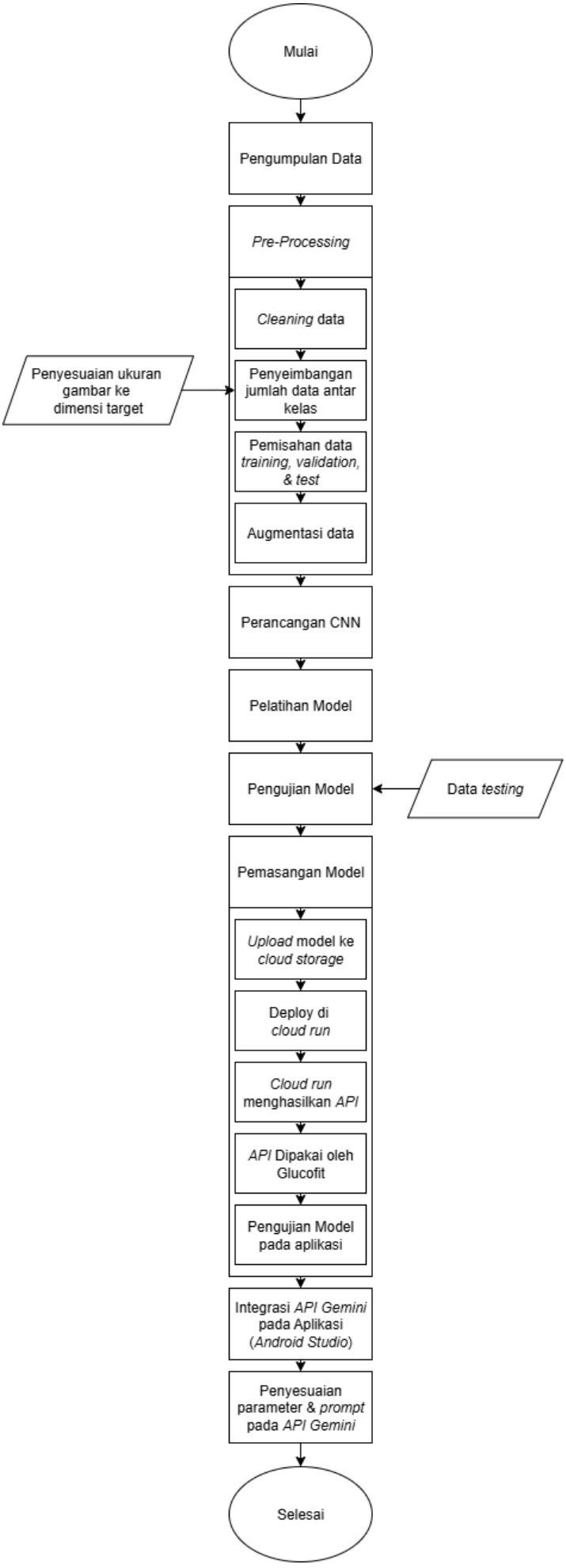
Dalam konteks **Sistem Pakar**, meskipun sistem ini dapat memberikan penjelasan berbasis pengetahuan ahli yang terstruktur, pengumpulan dan pemrograman pengetahuan ahli membutuhkan waktu dan biaya yang tidak sedikit. API seperti Gemini dapat menyerap dan mengadaptasi berbagai informasi dari berbagai sumber tanpa perlu mengumpulkan data ahli secara manual. Dengan menggunakan API, proses pengolahan data dan pengetahuan dapat diotomatisasi, memungkinkan penyajian penjelasan yang lebih cepat dan efisien, sambil tetap mempertahankan akurasi yang tinggi. Hal ini sangat berguna dalam situasi di mana pengetahuan baru terus berkembang, dan pembaruan sistem pakar secara manual akan memakan waktu dan sumber daya yang lebih banyak.

Metode **Asisten Virtual (*Chatbot*)**, yang menawarkan interaksi langsung dengan pengguna, memiliki tantangan dalam hal pengembangan dan pemrograman yang rumit, serta keterbatasan dalam memberikan jawaban yang lebih mendalam. API *Gemini* dapat menutupi kekurangan ini dengan menyediakan kecerdasan buatan yang dapat mengolah pertanyaan lebih kompleks dan memberikan jawaban yang lebih alami dan dinamis. API memungkinkan chatbot untuk belajar dari interaksi sebelumnya, meningkatkan kualitas respons tanpa memerlukan pengembangan sistem yang rumit. Selain itu, API juga memfasilitasi pengawasan dan pembaruan secara otomatis, yang memastikan bahwa kualitas jawaban tetap terjaga.

Secara keseluruhan, penggunaan API seperti Gemini menawarkan solusi untuk masalah-masalah utama yang ditemukan dalam metode lain, termasuk pembaruan data otomatis, interaksi dinamis, pengurangan kompleksitas pengelolaan data, dan pemrograman yang lebih efisien. Dengan menggunakan API, sistem dapat lebih adaptif dan responsif terhadap kebutuhan pengguna, mengoptimalkan penyajian informasi dan penjelasan dalam waktu yang lebih singkat dan dengan lebih sedikit sumber daya yang dibutuhkan.

# BAB III
# METODOLOGI PENELITIAN

Penelitian ini akan difokuskan pada pengembangan dan pengujian fitur scan makanan pada aplikasi GlucoFit, yang memanfaatkan model *machine learning* berbasis *Convolutional Neural Network (CNN)* untuk mengidentifikasi kandungan gula pada makanan. Untuk memberikan penjelasan ilmiah mengenai kepada pengguna, integrasi API *Gemini* yang berfungsi menyampaikan informasi terkait asal bahan dan proses pengolahan yang dapat memengaruhi kadar gula. Metodologi eksperimental digunakan untuk menguji performa integrasi API dalam menghasilkan informasi yang relevan dan mudah dipahami oleh pengguna. Rinciannya dapat diamati melalui representasi visual pada Gambar 3.1.


Mulai
Pengumpulan Data
Pre-Processing
Cleaning data
Penyesuaian ukuran gambar ke dimensi target
Penyeimbangan jumlah data antar kelas
Pemisahan data training, validation, & test
Augmentasi data
Perancangan CNN
Pelatihan Model
Pengujian Model
Data testing
Pemasangan Model
Upload model ke cloud storage
Deploy di cloud run
Cloud run menghasilkan API
API Dipakai oleh Glucofit
Pengujian Model pada aplikasi
Integrasi API Gemini pada Aplikasi (Android Studio)
Penyesuaian parameter & prompt pada API Gemini
Selesai

Gambar 3.1 Proses pembuatan model dan integrasi API

Pada Gambar 3.1 menguraikan rangkaian metodologi yang akan diterapkan mulai dari tahap awal hingga tahap akhir dalam jalur penelitian ini. Setiap tahapan dalam gambar tersebut akan dijabarkan secara rinci dalam sub bab dibawah ini.

## 3.1 Pengumpulan Data

Tahap awal pembangunan model *machine learning* melibatkan pengumpulan data berupa citra atau foto makanan. Data utama diperoleh dari dataset "Food-101" yang diunduh dari Kaggle. Dataset ini mencakup 60 jenis makanan populer di Indonesia, dengan total 60.000 gambar (1.000 gambar untuk setiap kelas makanan). Pemilihan dataset ini didasarkan pada kesesuaian dengan kebutuhan proyek, jumlah data yang memadai untuk pelatihan model *CNN*, serta relevansi varian kelas makanan dengan selera masyarakat Indonesia, seperti sushi, ramen, dan burger. Setelah dikumpulkan, gambar-gambar tersebut diproses untuk memastikan kualitas dan konsistensi dataset sebelum digunakan dalam pelatihan model.

Sebagai pelengkap, data tambahan berupa estimasi kandungan gula untuk setiap jenis makanan diperoleh secara manual melalui internet, khususnya website tentang nutrisi makanan yang terpercaya.

## 3.2 Pre-processing (Pengolahan Data)

Setelah dataset berhasil dikumpulkan, langkah berikutnya adalah mempersiapkan data tersebut agar siap digunakan untuk melatih model. Tahap persiapan ini sangat penting karena menentukan seberapa baik model nantinya dapat belajar mengenali gambar dengan akurat. Setelah pengumpulan data gambar selesai, dilakukan persiapan data untuk pelatihan model seperti cleaning model, penyamaan jumlah dataset agar tidak bias, kemudian langkah-langkah yang memerlukan kodingan sebagai berikut:

### 3.2.1 Cleaning Data

Tahap *cleaning data* bertujuan untuk membersihkan dataset dari gambar yang tidak relevan, rusak, atau tidak sesuai dengan kategori makanan yang ditentukan. Pada tahap ini, dilakukan pemeriksaan manual maupun otomatis terhadap seluruh file gambar untuk memastikan formatnya sesuai (misalnya JPEG atau PNG), ukuran file memadai, serta tidak ada duplikasi. Proses ini sangat penting untuk meningkatkan kualitas dataset sehingga model dapat belajar secara lebih akurat dan efisien.

### 3.2.2 Penyeimbangan Jumlah Data Antar Kelas

Dataset yang tidak seimbang antar kelas dapat menyebabkan model cenderung bias terhadap kelas mayoritas. Untuk mengatasi hal ini, dilakukan penyeimbangan jumlah data pada setiap kelas makanan. Teknik yang digunakan meliputi penghapusan sampel berlebih pada kelas yang terlalu banyak (*undersampling*) atau penambahan variasi gambar pada kelas yang minim menggunakan augmentasi (*oversampling*). Dengan demikian, model diharapkan memiliki kemampuan klasifikasi yang merata dan tidak hanya unggul pada kelas tertentu saja.

### 3.2.3 Pemisahan Data Training, Validation, dan Testing

Dataset yang telah dikumpulkan kemudian dibagi menjadi tiga bagian utama: data latih, data validasi, dan data uji. Pembagian ini dilakukan untuk mendukung proses pelatihan dan evaluasi model secara lebih efektif. Sebagian besar data digunakan sebagai data latih, yaitu bagian dari dataset yang akan dipelajari oleh model untuk mengenali pola atau ciri-ciri dari gambar makanan. Setelah itu, sebagian kecil dari dataset dialokasikan sebagai data validasi yang berfungsi untuk mengevaluasi kinerja model selama proses pelatihan serta membantu menyetel parameter model agar tidak hanya hafal pada data latih, tetapi juga mampu mengenali data baru dengan baik. Sementara itu, data uji digunakan setelah model selesai dilatih dan disetel, bertindak sebagai simulasi kondisi nyata untuk memberikan penilaian akhir terhadap kemampuan model dalam memprediksi data yang belum pernah dilihat sebelumnya.

Pada tahap yang sama, semua gambar juga diubah ukurannya menjadi dimensi yang seragam agar sesuai dengan format input model *CNN*. Proses ini memastikan bahwa seluruh gambar memiliki ukuran yang kompatibel dengan arsitektur model yang digunakan. Dengan pendekatan pembagian data yang sistematis dan penyesuaian format gambar yang tepat, model diharapkan mampu memberikan hasil yang akurat dan konsisten saat digunakan dalam kondisi nyata.

### 3.2.4 Augmentasi Data

Augmentasi data dilakukan untuk meningkatkan keragaman dataset secara virtual tanpa mengubah label asli dari gambar. Teknik yang digunakan meliputi rotasi, pergeseran, zoom, pembalikan horizontal, dan shear. Proses ini membantu model dalam mempelajari variasi objek dari berbagai sudut dan kondisi, sehingga meningkatkan kemampuan generalisasi model. Augmentasi hanya diterapkan pada data latih agar evaluasi pada data validasi dan uji tetap objektif.

### 3.3 Perancangan *CNN*

Dalam penelitian ini, model *Convolutional Neural Network (CNN)* dipilih sebagai metode utama untuk membangun sistem klasifikasi gambar makanan. Pemilihan model ini didasarkan pada kemampuannya dalam mengenali pola visual pada citra secara efektif, terutama pada gambar yang memiliki struktur grid seperti foto makanan. *CNN* dirancang khusus untuk menangani data berbentuk gambar, sehingga cocok digunakan dalam tugas pengenalan objek berbasis citra.

Salah satu keunggulan utama dari *CNN* adalah kemampuan model untuk mengekstrak fitur penting dari gambar secara otomatis melalui beberapa lapisan pemrosesan. Dengan demikian, tidak diperlukan proses ekstraksi fitur manual yang rumit dan rentan kesalahan. Selain itu, untuk meningkatkan efisiensi pelatihan sekaligus hasil akhir model, digunakan teknik *transfer learning* yang memanfaatkan model *CNN* yang telah dilatih sebelumnya. Teknik ini membantu mempercepat proses pembelajaran dan meningkatkan kemampuan model dalam mengenali gambar baru secara akurat.

Pendekatan ini memungkinkan model mencapai performa yang tinggi meskipun menggunakan dataset yang relatif terbatas. Dengan kombinasi arsitektur *CNN* dan pemanfaatan *transfer learning*, model diharapkan dapat memberikan hasil yang optimal dalam mengklasifikasikan berbagai jenis makanan berdasarkan gambar input.

### 3.4 Pelatihan Model

Model dilatih menggunakan dataset yang telah disiapkan untuk mempelajari ciri-ciri visual setiap jenis makanan. Selama proses ini, dilakukan penyetelan parameter secara optimal sambil menggunakan data validasi untuk mencegah model menghafal data latihan (*overfitting*). Kinerja model diukur menggunakan akurasi, yaitu persentase prediksi yang benar dibandingkan seluruh prediksi yang dibuat. Dengan pendekatan ini, model dapat mengenali berbagai jenis makanan dengan tingkat keakuratan yang tinggi.

### 3.5 Pengujian Model

Model yang telah dilatih kemudian diuji menggunakan dua sumber data: dataset testing yang sudah dipisahkan sebelumnya dan gambar makanan tambahan dari internet. Pengujian ini bertujuan untuk mengevaluasi kemampuan model dalam mengenali jenis makanan dengan benar. Hasil pengujian kemudian dianalisis secara mendalam untuk mengidentifikasi kelebihan

dan kekurangan model, sehingga bisa dilakukan penyempurnaan lebih lanjut guna meningkatkan performanya.

## 3.6 Pemasangan Model ke Aplikasi Android

Langkah pertama dalam pemasangan model *machine learning* ke dalam aplikasi Android melibatkan beberapa tahapan yang teliti dan esensial. Proses ini dirinci sebagai berikut:

### 3.6.1 Upload Model ke *Cloud Storage*

Model *machine learning* yang telah selesai dilatih disimpan dalam bentuk file, seperti .tflite atau .h5. Agar dapat diakses oleh aplikasi secara online, model ini diunggah ke layanan penyimpanan awan (*cloud storage* ), seperti Google *Cloud*. Langkah ini memastikan bahwa model tersedia kapanpun dan dimanapun saat dibutuhkan oleh aplikasi.

### 3.6.2 Deploy di *Cloud Run*

Setelah model tersimpan di *cloud*, langkah selanjutnya adalah menyediakan layanan untuk menjalankan model tersebut. Ini dilakukan dengan mendeploy *backend* (server) berupa API menggunakan layanan seperti Google *Cloud Run*. Dengan demikian, aplikasi bisa mengirimkan data gambar ke server, lalu server akan memproses prediksi menggunakan model *machine learning* dan mengembalikan hasilnya.

### 3.6.3 API Dipakai oleh GlucoFit

GlucoFit adalah nama aplikasi Android yang dikembangkan dalam penelitian ini. Setelah API siap di *cloud*, aplikasi GlucoFit terhubung dengannya untuk mengakses model *machine learning*. Saat pengguna mengambil atau mengunggah foto makanan, aplikasi mengirimkan gambar tersebut melalui API ke server, dan menampilkan hasil prediksi jenis makanan beserta informasi gizinya.

### 3.6.4 Pengujian Model pada Aplikasi

Sebelum aplikasi digunakan secara luas, model *machine learning* dan integrasinya dengan aplikasi diuji coba. Uji coba dilakukan untuk memastikan bahwa gambar dari kamera atau galeri benar diproses, komunikasi antara aplikasi dan server berjalan lancar, serta prediksi akhir sesuai dengan harapan. Proses ini juga membantu menemukan dan memperbaiki potensi masalah sebelum aplikasi dirilis.

### 3.7 Integrasi API pada Aplikasi (*Gemini API*)

API digunakan untuk memberikan penjelasan ilmiah tentang kandungan gula dalam makanan yang terdeteksi. Tahapannya meliputi:

a) **Panggilan API**:

1. Aplikasi mengirimkan jenis makanan dan estimasi kandungan gula ke **API Gemini**.
2. Permintaan mencakup informasi tentang **asal bahan, proses pengolahan, dan saran aktivitas** untuk mengurangi dampak gula. Berikut adalah seluruh perintahnya:

```
"Jelaskan dengan singkat, pertanyaan persuasif, & menarik tentang asal kandungan gula sebanyak $sugarContent gram pada $foodName tanpa mengulangi lagi perintah ini.

Fokuskan pada proses pembuatan, analogi kreatif tanpa mengurangi selera makan, dan bagaimana mempengaruhi penyakit diabetes dalam bentuk yang mudah dipahami & persuasif. Berikan saran singkat aktivitas/olahraga untuk mengurangi jumlah kadar gula tersebut! "
```

Gambar 3.2 *Prompt Generative AI* pada Glucofit

b) **Pengolahan Data**:

1. API menghasilkan respons berupa:
   I. **Asal Usul Bahan Makanan** (sumber bahan yang memengaruhi kandungan gula).
   II. **Proses Pengolahan Makanan** (metode memasak yang memengaruhi gula).
   III. **Saran Kesehatan** (rekomendasi aktivitas untuk menyeimbangkan asupan gula).

c) **Integrasi Respons ke Aplikasi**:

1. Hasil API ditampilkan dalam format yang mudah dipahami pengguna.

### 3.8 Evaluasi Hasil Penelitian

Penelitian ini melakukan evaluasi melalui dua tahap utama untuk mengukur efektivitas sistem yang dikembangkan, yaitu Uji Akurasi Model dan Uji Integrasi API. Hasil lengkap dari

kedua pengujian tersebut telah didokumentasikan dalam laporan terpisah. Berikut penjelasan detail mengenai metode evaluasi yang dilakukan.

1. Uji Akurasi Model

Evaluasi terhadap model *machine learning* dilakukan untuk mengukur performanya dalam mengenali jenis makanan dan memprediksi kandungan gula. Pengujian ini menggunakan dataset yang telah disiapkan sebelumnya dengan akurasi sebagai metrik utama. Hasil pengujian menunjukkan kemampuan model dalam melakukan identifikasi makanan secara akurat, dengan analisis detail yang tercantum dalam lampiran laporan.

2. Uji Integrasi API

Pengujian integrasi sistem menggunakan pendekatan *black box testing* yang melibatkan penilaian ahli di bidang diabetes. Evaluasi dilakukan dengan skala Likert untuk mengukur:

a. Daya tarik dan kemudahan pemahaman konten
b. Tingkat relevansi output dengan input makanan
c. Konsistensi informasi dengan referensi ilmiah terkait bahan makanan dan proses pengolahannya

Tujuan pengujian ini adalah untuk memverifikasi bahwa sistem mampu memberikan informasi yang akurat dan bermanfaat mengenai kandungan gula, asal bahan, serta proses pengolahan makanan. Penilaian dilakukan secara subjektif oleh ahli berdasarkan kriteria yang telah ditetapkan, dengan hasil evaluasi yang telah didokumentasikan secara terpisah.

Kedua metode evaluasi ini saling melengkapi dalam menilai aspek teknis dan fungsional sistem secara komprehensif, sekaligus menjadi dasar untuk pengembangan lebih lanjut.

# BAB IV
# HASIL DAN PEMBAHASAN

## 4.1 Pengumpulan Data

Untuk memastikan model yang dikembangkan sesuai dengan kebiasaan makan masyarakat Indonesia, proses kurasi terhadap dataset Food-101 dari Kaggle perlu dilakukan. Dataset awal yang berisi 101.000 gambar dari 101 kategori makanan diseleksi berdasarkan relevansinya dengan pola konsumsi lokal. Proses seleksi ini dilakukan dengan cara:

1. Mengidentifikasi makanan yang umum ditemukan di warung makan, kafe, atau restoran di Indonesia
2. Berkonsultasi dengan keluarga dan teman tentang makanan apa saja yang sering mereka temui atau konsumsi sehari-hari
3. Menghilangkan 41 kategori makanan yang dianggap kurang familiar atau jarang dikonsumsi di Indonesia

Hasil seleksi menghasilkan 60 kategori makanan yang populer, dengan total 60.000 gambar. Pendekatan ini memastikan model dapat mengenali makanan-makanan yang benar-benar relevan dengan konteks penggunaan sehari-hari di Indonesia. Berikut adalah daftar makanannya:

Tabel 4.1 Daftar Kelas Makanan

| No. | Kelas Makanan | No | Kelas Makanan | No | Kelas Makanan |
|---|---|---|---|---|---|
| **1** | club_sandwich | **21** | dumplings | **41** | apple_pie |
| **2** | baklava | **22** | fish_and_chips | **42** | cannoli |
| **3** | beignets | **23** | spaghetti_carbonara | **43** | samosa |
| **4** | bibimbap | **24** | prime_rib | **44** | grilled_cheese_sandwich |
| **5** | caesar_salad | **25** | garlic_bread | **45** | french_toast |
| **6** | strawberry_shortcake | **26** | donuts | **46** | steak |
| **7** | red_velvet_cake | **27** | cheesecake | **47** | caprese_salad |
| **8** | waffles | **28** | pizza | **48** | chocolate_mousse |
| **9** | spaghetti_bolognese | **29** | macarons | **49** | baby_back_ribs |
| **10** | tiramisu | **30** | creme_brulee | **50** | grilled_salmon |
| **11** | tacos | **31** | pancakes | **51** | onion_rings |
| **12** | sushi | **32** | fried_calamari | **52** | filet_mignon |
| **13** | chocolate_cake | **33** | miso_soup | **53** | chicken_curry |

| | | | | | |
|---|---|---|---|---|---|
| **14** | fried_rice | **34** | churros | **54** | hot_dog |
| **15** | french_fries | **35** | sashimi | **55** | chicken_wings |
| **16** | hamburger | **36** | gyoza | **56** | lasagna |
| **17** | greek_salad | **37** | takoyaki | **57** | cheese_plate |
| **18** | macaroni_and_cheese | **38** | ramen | **58** | ice_cream |
| **19** | falafel | **39** | frozen_yogurt | **59** | breakfast_burrito |
| **20** | cup_cakes | **40** | nachos | **60** | omelette |

Kemudian setelah dicocokan dengan data estimasi kandungan gula makanan secara manual seperti pada lampiran 2, kita bisa melihat perbedaan antara kelompok makanan manis dengan kelompok makanan asin. Berdasarkan analisis data kandungan gula pada berbagai jenis makanan, terlihat pemisahan yang sangat jelas antara kelompok makanan manis dan asin atau gurih. Kelompok makanan manis, yang didominasi oleh hidangan penutup seperti kue, pai, dan es krim, menunjukkan rata-rata kandungan gula yang sangat tinggi, yaitu sekitar 24,6 gram—hampir delapan kali lipat lebih tinggi dibandingkan kelompok asin. Rentang kandungan gulanya juga sangat lebar (37,7 gram), menandakan variasi yang besar dalam tingkat kemanisan, dengan chocolate cake sebagai makanan tertinggi (44 gram). Sebaliknya, kelompok makanan asin atau gurih—yang terdiri atas hidangan utama seperti daging, ikan, salad gurih, dan makanan berbasis gurih—memiliki rata-rata kandungan gula hanya 3,2 gram, dengan sebagian besar item bahkan mendekati nol gram. Meskipun demikian, terdapat beberapa outlier dalam kelompok asin, seperti spaghetti bolognese (14 gram) dan lasagna (11 gram), yang kandungan gulanya meningkat signifikan kemungkinan akibat penggunaan saus tomat atau bumbu tambahan.

Perbedaan ini tidak hanya terletak pada angka, tetapi juga pada sumber dan fungsi makanan. Gula pada kelompok manis umumnya berasal dari tambahan gula rafinasi seperti gula pasir, sirup, atau cokelat, dan berperan sebagai pemuas selera atau hidangan penutup. Sementara itu, pada kelompok asin, gula hadir dalam jumlah kecil dan biasanya berasal dari bahan alami (seperti laktosa dalam keju atau fruktosa dalam tomat) atau sebagai penyeimbang rasa dalam bumbu. Implikasinya, konsumen yang ingin membatasi asupan gula disarankan untuk lebih memilih hidangan dari kelompok asin yang kandungan gulanya rendah, seperti steak, atau salad sayuran tanpa dressing manis.

## 4.2 *Pre-processing*

Setelah proses pengumpulan data selesai, dilakukan serangkaian langkah praproses untuk memastikan dataset siap dan optimal digunakan dalam pembuatan model *machine learning*. Langkah pertama yang dilakukan adalah data cleaning, yaitu proses pembersihan data guna memastikan bahwa setiap citra masih relevan dengan kelasnya. Dalam kasus ini, terdapat beberapa data (foto) yang ternyata bukan merupakan gambar makanan dan perlu dikeluarkan dari dataset.

Setelah proses pembersihan selesai, langkah berikutnya adalah menyeimbangkan jumlah data pada setiap kelas agar tidak terjadi bias selama pelatihan model. Hal ini penting untuk menjaga akurasi dan keadilan model dalam memprediksi setiap kategori. Setelah data seimbang, tahap selanjutnya adalah pre-processing yang melibatkan penggunaan kode program.

### 4.2.1 Cleaning Data

Tahap *cleaning data* merupakan langkah awal dalam proses *pre-processing* yang bertujuan untuk memastikan dataset berada dalam kondisi yang bersih dan layak digunakan untuk pelatihan model. Dalam konteks penelitian ini, dataset terdiri dari kumpulan gambar makanan yang dikumpulkan dari berbagai sumber. Oleh karena itu, sebelum digunakan lebih lanjut, diperlukan proses pembersihan manual untuk menjaga kualitas data.

Proses *cleaning* dilakukan secara cermat dengan langkah-langkah sebagai berikut:

1. Memeriksa satu per satu gambar secara manual untuk memastikan bahwa gambar tersebut benar-benar merupakan foto makanan sesuai dengan kategori yang dimaksud. Beberapa gambar dihapus jika ternyata tidak relevan atau bukan gambar makanan.
2. Mengecek keberadaan duplikasi gambar, baik dalam satu kelas maupun antar kelas. Proses ini dilakukan secara visual dengan melihat tampilan gambar satu per satu.
3. Menghapus gambar yang tidak dapat dibuka atau memiliki format yang tidak didukung oleh sistem. *File* rusak atau tidak kompatibel bisa menyebabkan error saat proses pelatihan model.
4. Memvalidasi kesesuaian label kelas dengan isi gambar, sehingga tidak terjadi kesalahan pelabelan (*mislabeling*). Misalnya, gambar mie goreng yang masuk ke dalam folder "nasi goreng" akan dipindahkan atau dihapus.

Langkah *cleaning data* yang dilakukan secara manual ini sangat penting untuk menjaga akurasi dan keandalan model. Selain itu, proses ini membantu menghindari bias pada model

akibat data yang tidak relevan atau salah label, serta meningkatkan efisiensi komputasi selama pelatihan model.

### 4.2.2 Penyeimbangan Jumlah Data Antar Kelas

Pada tahap ini, dilakukan penyeimbangan jumlah data untuk setiap kelas agar tidak terjadi bias pada model akibat ketidakseimbangan distribusi data. Penyeimbangan dilakukan dengan cara menyamakan jumlah data pada setiap kelas berdasarkan jumlah data paling sedikit yang dimiliki oleh salah satu kelas. Dengan demikian, semua kelas akan memiliki jumlah data yang sama, sehingga model dapat belajar secara proporsional dari masing-masing kelas.

Berdasarkan analisis awal, jumlah data minimal yang ditemukan dalam satu kelas adalah sebanyak 807 data . Oleh karena itu, seluruh kelas lainnya disamakan atau dibatasi menjadi 807 data per kelas. Proses ini membantu mengurangi risiko *overfitting* pada kelas mayoritas dan meningkatkan kemampuan generalisasi model dalam mengenali setiap kategori makanan secara lebih akurat dan adil.

### 4.2.3 Pemisahan Data Training, Validation, dan Testing

Proses pemisahan dataset menjadi data latih (*training* ), validasi (*validation* ), dan uji (*testing* ) merupakan tahap penting dalam pengembangan model klasifikasi gambar makanan. Tujuannya adalah untuk memastikan bahwa model tidak hanya belajar dengan baik dari data latih, tetapi juga mampu menggeneralisasi pada data baru. Proses ini dilakukan setelah seluruh data berhasil dikumpulkan, dikelompokkan, dan disimpan dalam format *DataFrame*. Berikut penjelasan rinci dari setiap langkahnya:

#### 4.2.3.1 Struktur Direktori dan Klasifikasi Data

Data gambar dikelompokkan berdasarkan 60 kategori makanan yang telah ditentukan sebelumnya. Setiap kategori direpresentasikan dalam variabel food_names yang berisi daftar nama makanan seperti "apple_pie", "beef_carpaccio", hingga "waffles". Struktur direktori dibuat dengan pola data/[nama_makanan] agar proses pembacaan dan pengelolaan data dapat dilakukan secara otomatis dan terstruktur seperti yang tertera pada gambar 4.1.

```
    # Daftar 60 kategori makanan yang dipilih
food_names =
["apple_pie", "baby_back_ribs", "baklava", "beef_carpaccio", "beet_salad",
"beignets", "bibimbap", "breakfast_burrito", "caesar_salad", "cannoli",
"caprese_salad", "cheesecake", "cheese_plate", "chicken_curry",
```

```
"chicken_wings", "chocolate_cake", "chocolate_mousse", "churros",
"club_sandwich", "creme_brulee", "cup_cakes", "donuts", "dumplings",
"falafel", "filet_mignon", "fish_and_chips", "french_fries",
"french_toast", "fried_calamari", "fried_rice", "frozen_yogurt",
"garlic_bread",  "greek_salad", "grilled_cheese_sandwich",
"grilled_salmon", "gyoza", "hamburger", "hot_dog", "ice_cream", "lasagna",
"macaroni_and_cheese", "macarons", "miso_soup", "nachos", "omelette",
"onion_rings",  "pancakes", "panna_cotta", "peking_duck", "pho", "pizza",
"pork_chop", "poutine", "prime_rib", "ramen", "red_velvet_cake", "samosa",
"sashimi", "spaghetti_bolognese" "spaghetti_carbonara", "steak",
"strawberry_shortcake", "sushi", "tacos", "takoyaki", "tiramisu","waffles"
]
```

Gambar 4.1 Daftar Nama Makanan pada Klasifikasi Gambar dalam Kodingan

#### 4.2.3.2 Pembacaan dan Penyimpanan Path Gambar

Untuk mengelola data secara efisien, path file gambar dan label kategori masing-masing dimuat ke dalam sebuah *list* menggunakan modul os dari *Python*. Selama proses ini, sistem memverifikasi eksistensi direktori tiap kategori makanan untuk memastikan semua data tersedia sebelum dilanjutkan ke tahap berikutnya.

```
    PROGRAM KumpulkanDataGambarMakanan
    MULAI

        BUAT base_dir = 'data'
        BUAT food_dirs = MAP food -> GABUNGKAN_PATH(base_dir, food) UNTUK
SETIAP food DALAM food_names

        BUAT data = LIST_KOSONG

        UNTUK SETIAP food DALAM food_names:
            BUAT food_dir = GABUNGKAN_PATH(base_dir, food)

            JIKA DIREKTORI_ADA(food_dir) MAKA
                UNTUK SETIAP img_file DALAM LIST_FILE(food_dir):
                    BUAT img_path = GABUNGKAN_PATH(food_dir, img_file)
                    TAMBAHKAN (img_path, food) KE data
            SELAIN ITU:
                CETAK "Peringatan: Direktori " + food + " tidak ditemukan"
        SELESAI
```

Gambar 4.2 Pseudocode Struktur Direktori dan Pengumpulan Data Gambar.

Pada Gambar 4.2 ditunjukkan contoh kode program yang digunakan untuk mengumpulkan data gambar makanan beserta labelnya. Kode tersebut pertama-tama mendefinisikan direktori utama (base_dir) sebagai folder 'data', kemudian membuat struktur path untuk setiap kategori makanan menggunakan dictionary comprehension. Selanjutnya, dilakukan iterasi melalui setiap kategori makanan untuk mengumpulkan path gambar dan labelnya. Jika direktori suatu kategori makanan tidak ditemukan, program akan menampilkan pesan peringatan. Setiap gambar yang valid beserta labelnya disimpan dalam sebuah list bernama data yang berisi *tuple (img_path, food*), di mana *img_path* merupakan lokasi file gambar dan food adalah nama kategori makanan yang sesuai. Kode ini menjadi langkah penting dalam persiapan data sebelum proses pelatihan model, karena memastikan bahwa semua gambar telah terorganisir dengan benar dan siap untuk diproses lebih lanjut.

#### 4.2.3.3 Konversi ke DataFrame

Setelah seluruh path gambar dan label berhasil dikumpulkan, data kemudian dikonversi ke dalam bentuk DataFrame menggunakan pustaka pandas seperti yang ditunjukkan pada Gambar 4.3.

```
BUAT DataFrame df DENGAN data DAN kolom ['image_path', 'label']
```

Gambar 4.3 Pseudocode konversi data ke DataFrame

Struktur DataFrame yang dihasilkan terdiri dari dua kolom utama: 'image_path' yang menyimpan lokasi file gambar, dan 'label' yang berisi kategori makanan terkait. Penggunaan format DataFrame ini memberikan beberapa keunggulan, antara lain mempermudah proses manipulasi data, memungkinkan pengecekan konsistensi data secara efisien, serta menyediakan struktur yang kompatibel untuk tahapan preprocessing berikutnya dalam pipeline pembangunan model. Konversi ke *DataFrame* merupakan langkah penting dalam memastikan data terorganisir dengan baik sebelum masuk ke proses ekstraksi fitur dan pelatihan model.

#### 4.2.3.4 Pemisahan Data

Tahap selanjutnya adalah pemisahan data menjadi tiga bagian: data latih, data validasi, dan data uji.

```
    PROGRAM BagiData
    MULAI
        BUAT train_df, temp_df = BAGI_DATA(df, ukuran_latih=0.85,
random_state=1)

        BUAT val_df, test_df = BAGI_DATA(temp_df, ukuran_test=0.5,
random_state=1)
    SELESAI
```

Gambar 4.4 Pseudocode Pre-processing untuk Data Validasi

Proses pemisahan dataset ditunjukkan pada Gambar 4.4, dimana dilakukan pembagian proporsional menggunakan fungsi *train_test_split* dari *scikit-learn*. Tahap pertama membagi dataset awal menjadi dua bagian, dengan 85% data dialokasikan sebagai data latih dan 15% sisanya sebagai data sementara (*temp_df*). Pada tahap kedua, data sementara tersebut dipisahkan secara merata menjadi data validasi dan data uji, masing-masing sebesar 7.5% dari total dataset awal. Pembagian ini menggunakan parameter *random_state=1* untuk memastikan hasil yang dapat direproduksi. Strategi pembagian seperti ini memungkinkan model untuk memiliki data yang cukup untuk proses pelatihan, sekaligus menyediakan data validasi untuk tuning parameter dan data uji yang independen untuk evaluasi akhir performa model.

Tabel 4.2 Jumlah Data setelah Dibagi

| Bagian | Training 85% | Validation 7.5% | Test 7.5% |
|---|---|---|---|
| 1 kelas | 685 | 61 | 61 |
| 60 kelas | 41100 | 3660 | 3660 |

Distribusi data sebelum proses pelatihan model ditunjukkan secara lengkap pada Tabel 4.2. Dataset dibagi secara proporsional menjadi tiga bagian utama, dimana 85% dialokasikan sebagai data training (41.100 sampel), sedangkan 7,5% masing-masing untuk data validasi dan data testing (3.660 sampel per bagian). Pembagian ini konsisten untuk semua kelas, dengan setiap kelas individu mendapatkan alokasi 685 sampel untuk training dan 61 sampel masing-masing untuk validasi dan *testing*. Pembagian yang seimbang ini memastikan bahwa model mendapatkan cukup data untuk belajar pola klasifikasi, sekaligus menyediakan data validasi

yang memadai untuk proses *tuning* dan data *testing* yang representatif untuk evaluasi akhir performa model.

### 4.2.4 Augmentasi Data

Augmentasi data merupakan tahap penting dalam *preprocessing* dataset gambar untuk meningkatkan variasi dan jumlah data latih secara virtual, sehingga membantu model belajar lebih baik tanpa menambah jumlah *file* asli. Proses ini dilakukan hanya pada data latih untuk memastikan bahwa evaluasi model tetap objektif dan tidak dipengaruhi oleh gambar hasil modifikasi.

```
PROGRAM KonfigurasiAugmentasiGambar
MULAI
    NORMALISASI=1/255,
    ROTASI=40,
    GESER_LEBAR=0.2,
    GESER_TINGGI=0.2,
    SHEAR=0.2,
    PERBESAR=0.2,
    PUTAR_HORIZONTAL=YA,
    MODE_ISI='nearest'
SELESAISELESAI
```

Gambar 4. 5 Pseudocode Augmentasi Data Latih

Konfigurasi augmentasi yang diterapkan meliputi normalisasi nilai piksel dengan *rescale*=1./255 (mengubah rentang nilai piksel dari 0-255 menjadi 0-1), rotasi gambar hingga 40 derajat (*rotation_range=40*), pergeseran horizontal dan vertikal sebesar 20% dari lebar/tinggi gambar (*width_shift_range* dan *height_shift_range=0.2*), transformasi *shear* dengan sudut 20 derajat (shear_range=0.2), zoom acak antara 80-120% (zoom_range=0.2), serta *flipping horizontal* (*horizontal_flip=True*). Parameter *fill_mode='nearest'* digunakan untuk mengisi piksel kosong yang muncul setelah transformasi dengan nilai piksel terdekat. Augmentasi ini menghasilkan variasi gambar yang lebih beragam dari data asli seperti terlihat pada Gambar 4.5, dimana terlihat contoh hasil transformasi gambar dari satu sampel input yang telah mengalami berbagai operasi geometris dan intensitas. Teknik ini secara signifikan meningkatkan generalisasi model dengan memperkenalkan variasi artifisial yang menyerupai kondisi nyata.

Transformasi ini membantu model dalam mengenali ciri-ciri visual makanan dari berbagai sudut dan kondisi, bukan hanya menghafal gambar-gambar tertentu. Dengan demikian, kemampuan generalisasi model diharapkan meningkat secara signifikan. Sebaliknya, data validasi dan uji tidak melalui proses augmentasi, namun hanya dinormalisasi nilainya agar sesuai dengan skema input model seperti yang ditunjukkan pada gambar 4.6. Hal ini memastikan bahwa evaluasi model dilakukan terhadap data yang alami dan representatif.

```
PROGRAM SiapkanDataTesting
MULAI
    BUAT pengolah_gambar_uji = GENERATOR_GAMBAR(
        normalisasi=1.0/255
    )
SELESAI
```

Gambar 4.6 Pseudocode Normalisasi pada Citra

#### 4.2.4.1 Pembentukan Generator Data dengan flow_from_dataframe

Proses selanjutnya adalah pembentukan Generator data yang mana masih merupakan proses dari augmentasi data. Membentuk generator data akan mengelola aliran gambar secara efisien selama pelatihan dan evaluasi model. Proses ini dilakukan menggunakan fungsi *flow_from_dataframe* dari objek *ImageDataGenerator*, yang memungkinkan pembacaan gambar langsung dari path yang tersimpan dalam *DataFrame (train_df, val_df, dan test_df).*

```
PROGRAM SiapkanDataLatihan
MULAI
    BUAT gambar_latihan = AMBIL_GAMBAR(
        data_asli=train_df,
        kolom_gambar='image_path',
        kolom_kategori='label',
        ukuran=(224, 224),
        warna='rgb',
        format_kategori='categorical',
        jumlah_per_kelompok=32,
        acak=YA,
        kode_acak=42
    )
SELESAI
```

Gambar 4.7 Pseudocode Alur Generator Gambar dari DataFrame (Latih)

Kode ini digunakan untuk mempersiapkan data gambar yang akan dipakai untuk melatih model. Dataframe (tabel data) akan berisi dua kolom penting: 'image_path' (lokasi file gambar) dan 'label' (kategori gambar). Setiap gambar akan diubah ukurannya menjadi 224x224 piksel dan menggunakan warna standar *RGB*. Karena model akan melakukan klasifikasi multi-kelas, label diubah ke format *one-hot encoding*. Data diambil per kelompok (*batch*) yang masing-masing berisi 32 gambar, dan diacak agar model belajar lebih baik. Angka seed=42 memastikan pengacakan ini bisa diulang dengan hasil sama. Seperti terlihat pada Gambar 4.7, proses ini memungkinkan komputer membaca dan memproses gambar secara efisien selama pelatihan model.

```
PROGRAM SiapkanDataUntukTesModel
MULAI
    BUAT gambar_untuk_tes = MENGAMBIL_GAMBAR_DARI_TABEL(
        tabel_data=test_df,
        kolom_gambar='image_path',
        kolom_label='label',
        ukuran_gambar=(224, 224),
        mode_warna='rgb',
        mode_label='categorical',
        ukuran_grup=32,
        acak_gambar=TIDAK
    )
SELESAI
```

Gambar 4.8 Pseudocode Alur Generator Gambar dari DataFrame (Validasi)

Kode ini digunakan untuk mempersiapkan data validasi yang akan dipakai untuk menguji performa model selama pelatihan. *Dataframe* validasi (val_df) akan berisi lokasi gambar (x_col=*'image_path'*) dan labelnya (y_col='label'). Sama seperti data latih, gambar diubah ukurannya menjadi 224x224 piksel dan menggunakan format warna RGB. Label diubah ke format one-hot encoding karena ini adalah masalah klasifikasi multi-kelas. Berbeda dengan data latih, data validasi tidak diacak (*shuffle=False*) agar evaluasi model lebih konsisten. Data diproses per kelompok (*batch*) yang masing-masing berisi 32 gambar. Seperti terlihat pada

Gambar 4.8, proses ini memungkinkan model untuk diuji dengan data baru yang belum pernah dilihat selama pelatihan, sehingga bisa mengetahui seberapa baik model benar-benar belajar.

```
PROGRAM MemuatGambarValidasi
MULAI
    BUAT gambar_valid DENGAN aliran_dari_dataframe(
        Data_asli = df_valid,
        kolom_x = 'path_gambar',
        kolom_y = 'label',
        ukuran_target = (224, 224),
        mode_warna = 'rgb',
        mode_label = 'kategorikal',
        ukuran_batch = 32,
        acak = SALAH
    )
SELESAI
```

Gambar 4.9 Pseudocode Alur Generator Gambar dari DataFrame (Test)

Kode ini berfungsi untuk mempersiapkan data testing yang akan digunakan untuk mengevaluasi performa akhir model. Data diambil *dari dataframe test_df* yang berisi dua kolom utama: *'image_path'* (menunjukkan lokasi penyimpanan file gambar) dan 'label' (kategori klasifikasi gambar). Setiap gambar akan diresize menjadi ukuran 224x224 piksel dan menggunakan mode warna RGB untuk konsistensi dengan data training. Parameter *class_mode='categorical'* digunakan karena ini merupakan masalah klasifikasi multi-kelas, dimana label akan dikonversi ke format one-hot encoding. Data *testing* diproses dalam batch berisi 32 gambar sekaligus untuk efisiensi komputasi. Berbeda dengan data training, data testing tidak diacak (*shuffle=False*) agar evaluasi hasilnya konsisten dan dapat direproduksi. Seperti yang ditunjukkan pada Gambar 4.9, generator ini akan secara otomatis membaca dan memproses gambar-gambar testing sesuai dengan konfigurasi yang telah ditentukan, sehingga memudahkan proses evaluasi akhir model.

Dengan pendekatan ini, seluruh dataset — baik latih, validasi, maupun uji — siap diproses secara otomatis dalam bentuk batch selama pelatihan dan evaluasi model *CNN*.

## 4.3 Perancangan *CNN*

Perancangan *Convolutional Neural Network (CNN)* yang akan dilakukan bertujuan untuk membuat model yang dapat mengenali dan mengklasifikasikan gambar dengan akurat. *CNN* dipilih karena kemampuannya yang baik dalam memproses data gambar secara otomatis. Arsitektur yang digunakan terdiri dari beberapa lapisan utama. Dalam implementasinya, akan digunakan teknik *transfer learning* dengan memanfaatkan model *EfficientNetB7* yang sudah pretrained untuk mempercepat proses pelatihan dan meningkatkan akurasi. Model ini kemudian akan dimodifikasi dengan menambahkan lapisan-lapisan khusus yang disesuaikan dengan kebutuhan klasifikasi pada penelitian ini.

```
PROGRAM PengambilFiturGambar
MULAI
    FUNGSI ambil_fitur(masukan):
        fitur = EfficientNetB7(
            ukuran_masukan=(224, 224, 3),
            sertakan_klasifikasi_akhir=False,
            bobot='imagenet'
        )(masukan)

        KEMBALIKAN fitur // Hasilnya gambar yang sudah diekstrak
SELESAI
```

Gambar 4.10 Pseudocode Ekstraksi Fitur

Kode ini membuat fungsi untuk mengambil ciri-ciri penting dari gambar menggunakan model *EfficientNetB7* yang sudah dilatih sebelumnya. Model ini menerima input gambar berukuran 224x224 piksel dengan 3 saluran warna (RGB). Parameter *include_top=False* berarti hanya menggunakan bagian untuk mengekstrak ciri-ciri, tanpa lapisan akhir untuk klasifikasi. Dengan *weights='imagenet'*, model menggunakan pengetahuan yang sudah dipelajari dari dataset ImageNet yang besar. Seperti terlihat pada Gambar 4.10, fungsi ini akan mengubah gambar input menjadi representasi numerik (fitur/hasil) yang lebih padat namun mengandung informasi penting, sehingga memudahkan model untuk mempelajari pola-pola dalam gambar. Pendekatan ini sangat efisien karena memanfaatkan pengetahuan yang sudah dimiliki model, alih-alih memulai dari nol.

```
PROGRAM KlasifikasiGambar
MULAI
    FUNGSI klasifikasikan(masukan):
       x = RataRataGlobal2D()(masukan)
       x = Ratakan()(x)
       x = LapisanPadat(1024, aktivasi="relu")(x)
       x = Dropout(0.2)(x)
       x = LapisanPadat(101, aktivasi="softmax", nama="klasifikasi")(x)

        KEMBALIKAN x
SELESAI
```

Gambar 4.11 Pseudocode Arsitektur Model

Kode ini mendefinisikan bagian classifier yang bertugas melakukan klasifikasi berdasarkan fitur yang telah diekstraksi. Pertama, *GlobalAveragePooling2D* digunakan untuk mengurangi dimensi fitur dengan mengambil rata-rata nilai setiap feature map, sehingga menghasilkan vektor yang lebih ringkas. Selanjutnya, Flatten meratakan data menjadi bentuk 1D agar dapat diproses oleh lapisan fully connected. Lapisan *Dense* dengan 1024 neuron dan aktivasi *ReLU* berfungsi untuk mempelajari pola kompleks dari data. *Dropout* dengan rate 0.2 diterapkan untuk mencegah *overfitting* dengan cara mematikan 20% neuron secara acak selama pelatihan. Terakhir, lapisan *Dense* dengan 101 neuron dan aktivasi *softmax* menghasilkan probabilitas kelas untuk 101 kategori yang berbeda, seperti terlihat pada Gambar 4.11. Arsitektur ini dirancang untuk mengoptimalkan akurasi klasifikasi sekaligus menjaga model agar tidak terlalu kompleks.

```
PROGRAM BuatDanLatihModel
MULAI
    FUNGSI buat_model_terlatih():
        masukan = Input(ukuran=(224, 224, 3))
        ciri_gambar = ambil_fitur(masukan)
        hasil_klasifikasi = pengklasifikasi(ciri_gambar)
        model = Model(masukan=masukan, keluaran=hasil_klasifikasi)
        model.siapkan(
            pengoptimal=Adam(laju_pembelajaran=0.001),
            kerugian='entropi_silang_kategorikal',
            metrik=['akurasi']
        )
```

```
        KEMBALIKAN model
SELESAI
```

Gambar 4.12 Pseudocode Fungsi define_compile_model() untuk Pembuatan dan Kompilasi Model

Kode ini digunakan untuk membuat dan menyiapkan model *deep learning* siap pelatihan. Pertama harus menentukan menentukan input berupa gambar berukuran 224x224 piksel dengan 3 saluran warna (RGB). Kemudian model memproses gambar melalui dua bagian utama: (1) *feature_extractor* yang bertugas mengambil ciri-ciri penting dari gambar menggunakan *EfficientNetB7*, dan (2) *classifier* yang melakukan klasifikasi berdasarkan ciri-ciri tersebut. Kedua bagian ini digabungkan menjadi satu model utuh. Model kemudian dikompilasi dengan optimizer Adam yang membantu proses pembelajaran dengan learning rate 0.001, menggunakan *loss function categorical_crossentropy* yang cocok untuk masalah klasifikasi multi-kelas, serta *metric accuracy* untuk mengukur performa. Seperti terlihat pada Gambar 4.12, seluruh proses ini menghasilkan model siap latih yang dapat mengenali dan mengklasifikasikan gambar dengan efektif.

Model akhir kemudian dibuat dengan memanggil fungsi *define_compile_model* dan *summary* model ditampilkan untuk memberikan gambaran menyeluruh tentang arsitektur yang dibangun, termasuk jumlah parameter yang dapat dilatih pada setiap lapisan.

```
PROGRAM DefinisikanDanKompilasiModel
MULAI
    model = definisikan_dan_kompilasi_model()
    CETAK model.ringkasan()
SELESAI
```

Gambar 4.13 Pseudocode Inisialisasi Model dan Ringkasan Arsitektur dengan model.summary()

Model akhir kemudian dibuat dengan memanggil fungsi define_compile_model dan *summary model* ditampilkan untuk memberikan gambaran menyeluruh tentang arsitektur yang dibangun, termasuk jumlah parameter yang dapat dilatih pada setiap lapisan.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_layer (InputLayer) | (None, 224, 224, 3) | 0 |
| efficientnetb7 (Functional) | (None, 7, 7, 2560) | 64,097,687 |
| global_average_pooling2d (GlobalAveragePooling2D) | (None, 2560) | 0 |
| flatten (Flatten) | (None, 2560) | 0 |
| dense (Dense) | (None, 1024) | 2,622,464 |
| dropout (Dropout) | (None, 1024) | 0 |
| classification (Dense) | (None, 60) | 61,500 |

Gambar 4.14 Keseluruhan arsitektur model

Model ini seperti pabrik pengolah gambar yang terdiri dari beberapa bagian penting. Pertama ada pintu masuk (input layer) yang menerima foto berukuran 224x224 piksel dengan warna standar (RGB). Kemudian ada mesin canggih bernama *EfficientNetB7* - ini seperti ahli seni yang bisa mengenali 64 juta ciri-ciri khusus dalam gambar, dan menghasilkan analisis detail berukuran 7x7 dengan 2560 karakteristik unik.

Selanjutnya ada proses penyederhanaan (*GlobalAveragePooling* dan *Flatten*) yang mengubah analisis tadi menjadi daftar angka panjang (2560 angka). Dua lapisan pemikir (*Dense*) kemudian bekerja: lapisan pertama punya 1024 "otak kecil" untuk memahami pola, dan lapisan akhir punya 60 "keputusan akhir" yang menentukan kategori gambar. Ada juga sistem pengaman (*Dropout*) yang sengaja mematikan sebagian "otak kecil" secara acak saat latihan agar model tidak terlalu ngotot mengingat gambar latihan.

Seperti terlihat di Gambar 4.14, seluruh sistem ini dirancang untuk bisa belajar mengenali dan mengelompokkan gambar dengan tepat, sambil memanfaatkan pengetahuan yang sudah dipelajari sebelumnya oleh *EfficientNetB7*. Model ini cukup pintar tapi tidak terlalu rumit, sehingga bisa bekerja dengan efisien.

## 4.4 Pelatihan Model

Setelah arsitektur model selesai dirancang, tahap selanjutnya adalah pelatihan model menggunakan data yang telah dipersiapkan. Model dikompilasi dengan *optimizer Adam* yang dikenal efisien dalam pelatihan *deep learning* karena kemampuannya menyesuaikan learning rate secara otomatis untuk setiap parameter. *Optimizer* ini juga efektif mengatasi masalah vanishing gradient yang sering terjadi pada jaringan neural yang dalam.

Fungsi *loss* yang digunakan adalah *Sparse Categorical Crossentropy*, pilihan yang tepat untuk masalah klasifikasi multi-kelas seperti pengenalan jenis makanan ini. Penggunaan parameter *from_logits=True* menunjukkan output model belum melalui fungsi aktivasi *softmax,* sehingga perhitungan dilakukan secara numerik yang lebih stabil oleh *TensorFlow*. Akurasi dipilih sebagai metrik evaluasi utama karena memberikan gambaran intuitif tentang performa model dalam memprediksi kelas yang benar.

```
PROGRAM LatihanModel
MULAI
    riwayat = model.latih(
        data_latih = gambar_latihan,
        data_validasi = gambar_validasi,
        jumlah_epoch = 20
    )
SELESAI
```

Gambar 4.15 Pseudocode Pelatihan Model dengan Metode model.fit()

Proses pelatihan model dilakukan menggunakan fungsi *fit* seperti yang ditunjukkan pada Gambar 4.15. Kode ini akan melatih model menggunakan data latih (train_images) yang telah disiapkan sebelumnya, sekaligus memvalidasi performa model menggunakan data validasi (*val_images*) di setiap *epoch*. Pelatihan dijalankan selama 20 *epoch*, artinya seluruh dataset akan diproses sebanyak 20 kali oleh model. Pada setiap *epoch*, model akan belajar menyesuaikan bobotnya untuk meningkatkan akurasi prediksi. Data validasi berfungsi sebagai penguji independen untuk memantau apakah model mengalami *overfitting* (terlalu menghafal data latih) selama proses pelatihan berlangsung. Proses ini menghasilkan objek history yang menyimpan rekaman perkembangan metrik akurasi dan *loss* selama pelatihan, yang nantinya dapat digunakan untuk analisis lebih lanjut.

```
PROGRAM SimpanModel
MULAI
    MODEL.SIMPAN(Kaggle-EfficientNeB7_Model_food_classifier_model.h5')
SELESAI
```

Gambar 4.16 Pseudocode Penyimpanan Model dalam Format .h5 Menggunakan model.save()

Setelah pelatihan selesai, model yang telah dilatih disimpan dalam format HDF5 (.h5) untuk penggunaan di masa mendatang. Format ini memungkinkan penyimpanan seluruh arsitektur model, bobot yang telah dilatih, serta informasi kompilasi dalam satu file.

## 4.5 Pengujian Model

Model diuji menggunakan dataset eksternal yang dikumpulkan secara terpisah dari data pelatihan untuk memvalidasi keakuratan model. Dataset uji terdiri dari 3.000 gambar dengan distribusi merata, yaitu 50 sampel per kelas.

```
Overall Analysis for prediction_results.csv:
----------------------------------------
Total Images: 3000
Correct Predictions: 2390
Incorrect Predictions: 610
Overall Accuracy: 79.67%
```

Gambar 4.17 Hasil Akurasi pada Pengujian Awal

Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 4.17, model mencapai akurasi keseluruhan sebesar 79,67%. Secara rinci:

a. Jumlah prediksi benar: 2.390 gambar
b. Jumlah prediksi salah: 310 gambar

Hasil ini menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam mengklasifikasikan data uji, meskipun masih terdapat ruang untuk peningkatan.

Tabel 4.3 Daftar Akurasi Setiap Kelas Makanan pada Pengujian Awal

| no | Kelas Makanan | Akurasi % | no | Kelas Makanan | Akurasi % |
|---|---|---|---|---|---|
| **1** | club_sandwich | 100 % | **31** | creme_brulee | 86 % |
| **2** | baklava | 98 % | **32** | pancakes | 84 % |
| **3** | beignets | 98 % | **33** | fried_calamari | 84 % |
| **4** | bibimbap | 98 % | **34** | miso_soup | 82 % |
| **5** | caesar_salad | 98 % | **35** | churros | 82 % |
| **6** | strawberry_shortcake | 98 % | **36** | sashimi | 82 % |
| **7** | red_velvet_cake | 98 % | **37** | gyoza | 80 % |
| **8** | waffles | 96 % | **38** | takoyaki | 80 % |
| **9** | spaghetti_bolognese | 96 % | **39** | ramen | 80 % |
| **10** | tiramisu | 94 % | **40** | frozen_yogurt | 78 % |
| **11** | tacos | 94 % | **41** | nachos | 76 % |

| | | | | | |
|---|---|---|---|---|---|
| **12** | sushi | 94 % | **42** | cannoli | 74 % |
| **13** | chocolate_cake | 94 % | **43** | samosa | 74 % |
| **14** | fried_rice | 94 % | **44** | grilled_cheese_sandwich | 72 % |
| **15** | french_fries | 94 % | **45** | french_toast | 70 % |
| **16** | hamburger | 92 % | **46** | steak | 70 % |
| **17** | greek_salad | 92 % | **47** | caprese_salad | 70 % |
| **18** | macaroni_and_cheese | 90 % | **48** | chocolate_mousse | 68 % |
| **19** | falafel | 90 % | **49** | baby_back_ribs | 66 % |
| **20** | cup_cakes | 90 % | **50** | grilled_salad | 62 % |
| **21** | apple_pie | 90 % | **51** | onion_rings | 60 % |
| **22** | dumplings | 88 % | **52** | filet_mignon | 58 % |
| **23** | fish_and_chips | 88 % | **53** | chicken_curry | 58 % |
| **24** | spaghetti_carbonara | 88 % | **54** | hot_dog | 58 % |
| **25** | prime_rib | 88 % | **55** | chicken_wings | 58 % |
| **26** | garlic_bread | 88 % | **56** | lasagna | 54 % |
| **27** | donuts | 86 % | **57** | cheese_plate | 48 % |
| **28** | cheesecake | 86 % | **58** | ice_cream | 42 % |
| **29** | pizza | 86 % | **59** | breakfast_burrito | 30 % |
| **30** | macarons | 86 % | **60** | omelette | 22 % |

Berdasarkan hasil evaluasi model klasifikasi makanan pada tabel 4.3 menggunakan dataset uji pertama, diperoleh tingkat akurasi yang bervariasi untuk setiap kelas makanan. Model menunjukkan performa terbaik pada kelas *club_sandwich* dengan akurasi sempurna sebesar 100%. Selain itu, beberapa kelas lain seperti *baklava*, *beignets*, *bibimbap*, *caesar_salad*, *strawberry_shortcake* , dan *red_velvet_cake* juga mencatatkan akurasi cukup tinggi sebesar 98%, menunjukkan bahwa model mampu mengklasifikasikan makanan-makanan tersebut dengan sangat baik.

Secara umum, model memberikan hasil yang stabil untuk kelas-kelas makanan populer atau memiliki ciri visual yang jelas dan mudah dibedakan. Namun, terdapat beberapa kelas makanan yang memiliki akurasi relatif rendah, seperti *omelette* (22%), *breakfast_burrito* (30%), *ice_cream* (42%), dan *cheese_plate* (48%). Rendahnya akurasi pada kelas-kelas ini kemungkinan disebabkan oleh beberapa faktor, antara lain:

a. Kemiripan visual dengan kelas lain: Beberapa makanan seperti *omelette* atau *ice_cream* mungkin memiliki penampilan yang mirip dengan makanan lain dalam dataset.

b. Jumlah data pelatihan yang tidak seimbang: Jika jumlah sampel untuk kelas tertentu lebih sedikit, maka model cenderung kurang optimal dalam mempelajari fitur unik dari kelas tersebut.

c. Variasi bentuk, warna, atau penyajian yang besar dalam satu kelas, sehingga menyulitkan model untuk mengenali pola secara konsisten.

Data (foto) yang awalnya salah diprediksi kemudian diuji kembali melalui serangkaian transformasi yang mensimulasikan kondisi riil penggunaan ponsel pintar, seperti penskalaan ulang (resizing), rotasi (rotation), penyesuaian pencahayaan (brightness adjustment), dan cropping. Transformasi ini bertujuan untuk meningkatkan robustnes model terhadap variasi input gambar yang mungkin terjadi di lapangan. Efektivitas optimasi ini diukur menggunakan metrik evaluasi berikut:

$$\frac{\text{Jumlah prediksi Benar Awal} + \text{Jumlah Prediksi Benar Setelah Optimisasi}}{\text{Total Jumlah Prediksi}} . 100\% \qquad (4.1)$$

Rumus ini menghitung peningkatan persentase akurasi secara komprehensif dengan membandingkan akumulasi prediksi benar sebelum dan setelah optimasi terhadap total dataset uji. Pendekatan ini tidak hanya mengukur peningkatan kinerja model akibat transformasi, tetapi juga memastikan bahwa perubahan yang dilakukan bersifat signifikan secara statistik. Hasil perhitungan akan menunjukkan seberapa besar model dapat diandalkan dalam menangani variasi gambar di lingkungan aplikasi nyata.

```
Combined Results:
----------------------------------------
Total Images: 3000
Updated Correct Predictions: 2614
Updated Incorrect Predictions: 386
Updated Overall Accuracy: 87.13%
```

Gambar 4.18 Akurasi pada Pengujian Akhir

Berdasarkan hasil pengujian, model mengalami peningkatan akurasi sebesar 7%, dengan total akurasi mencapai 87%. Berikut adalah detail peningkatan akurasi pada setiap kelas makanan pada tabel 4.4:

Tabel 4.4 Rentang Peningkatan Akurasi pada Setiap Kelas Makanan

| **Kelas Makanan** | **Akurasi %** | **Rentang Peningkatan Akurasi** |
|---|---|---|
| omelette | 51 | +29 |
| breakfast_burrito | 56 | +29 |
| ice_cream | 63 | +21 |
| cheese_plate | 67 | +19 |
| lasagna | 71 | +17 |
| onion_rings | 75 | +15 |
| filet_mignon | 73 | +15 |
| chicken_curry | 73 | +15 |
| hot_dog | 73 | +15 |
| chicken_wings | 73 | +15 |
| grilled_salmon | 76 | +14 |
| chocolate_mousse | 80 | +12 |
| baby_back_ribs | 78 | +12 |
| french_toast | 81 | +11 |
| steak | 81 | +11 |
| caprese_salad | 81 | +11 |
| grilled_cheese_sandwich | 82 | +10 |
| cannoli | 84 | +10 |
| samosa | 84 | +10 |
| frozen_yogurt | 86 | +8 |
| miso_soup | 89 | +7 |
| churros | 89 | +7 |
| sashimi | 89 | +7 |
| gyoza | 87 | +7 |
| takoyaki | 87 | +7 |
| ramen | 87 | +7 |
| pancakes | 90 | +6 |
| fried_calamari | 90 | +6 |
| donuts | 91 | +5 |
| cheesecake | 91 | +5 |

| | | |
|---|---|---|
| pizza | 91 | +5 |
| macarons | 91 | +5 |
| creme_brulee | 91 | +5 |
| dumplings | 92 | +4 |
| fish_and_chips | 92 | +4 |
| spaghetti_carbonara | 92 | +4 |
| prime_rib | 92 | +4 |
| garlic_bread | 92 | +4 |
| macaroni_and_cheese | 94 | +4 |
| falafel | 94 | +4 |
| cup_cakes | 94 | +4 |
| apple_pie | 94 | +4 |
| hamburger | 95 | +3 |
| greek_salad | 95 | +3 |
| tiramisu | 96 | +2 |
| tacos | 96 | +2 |
| sushi | 96 | +2 |
| chocolate_cake | 96 | +2 |
| fried_rice | 96 | +2 |
| french_fries | 96 | +2 |
| waffles | 97 | +1 |
| spaghetti_bolognese | 97 | +1 |
| baklava | 99 | +1 |
| beignets | 99 | +1 |
| bibimbap | 99 | +1 |
| caesar_salad | 99 | +1 |
| strawberry_shortcake | 99 | +1 |
| red_velvet_cake | 99 | +1 |
| club_sandwich | 100 | 0 |

Hasil tersebut menunjukkan bahwa penerapan transformasi pra-pemrosesan mampu membantu model mengenali gambar dengan lebih baik, terutama untuk kelas makanan yang sebelumnya sulit dikenali.

Peningkatan akurasi tertinggi terjadi pada kelas *omelette* dan *breakfast_burrito*, yaitu masing-masing naik 29% dan 26%, menandakan bahwa transformasi yang dilakukan cukup efektif untuk meningkatkan kemampuan model dalam mengklasifikasikan makanan-makanan tersebut. Kelas-kelas lain seperti *ice_cream*, *cheese_plate*, dan *lasagna* juga menunjukkan perbaikan signifikan. Meskipun beberapa kelas sudah memiliki akurasi tinggi dan tidak berubah, seperti *club_sandwich* atau *baklava*, kebanyakan kelas mendapat manfaat dari proses transformasi ini.

Pengujian juga dilakukan menggunakan metrik AUC (Area Under the Curve) untuk mengevaluasi kemampuan model dalam membedakan setiap kelas makanan secara komprehensif. Nilai macro-average AUC sebesar 0.9297 (pada kelas yang valid) pada lampiran 4 mengindikasikan bahwa model secara rata-rata memiliki akurasi 92.97% dalam membedakan semua kelas makanan dengan hampir sempurna. Skor AUC yang mendekati 1.0 ini menegaskan bahwa model jarang melakukan kesalahan klasifikasi antar kelas.

Beberapa jenis makanan bahkan mencapai skor AUC sempurna (1.0000), seperti strawberry_shortcake dan red_velvet_cake, yang berarti model tidak pernah salah mengidentifikasi kedua makanan ini. Kelas lain seperti miso_soup (0.9999), spaghetti_carbonara (0.9993), dan creme_brulee (0.9992) juga mendekati sempurna, dengan hanya sedikit kasus yang mungkin terlewat. Namun, terdapat kelas club_sandwich yang tidak memiliki nilai AUC (NaN) karena jumlah data validasinya tidak mencukupi untuk perhitungan metrik ini. Meskipun demikian, berdasarkan hasil sebelumnya, kelas ini memiliki akurasi 100%, sehingga tetap dapat diandalkan meskipun evaluasi AUC tidak tersedia.

Dengan hasil AUC yang tinggi secara keseluruhan, model CNN ini terbukti handal dalam mengenali berbagai jenis makanan, yang menjadi dasar penting untuk estimasi kandungan gula yang akurat dalam aplikasi GlucoFit. Performa ini juga menunjukkan bahwa model mampu mengatasi variasi visual dalam gambar makanan, sehingga layak digunakan dalam skenario dunia nyata.

Secara keseluruhan, rangkaian pengujian ini membuktikan ketahanan (robustness) model dalam menghadapi berbagai kondisi pengambilan gambar di dunia nyata — seperti variasi ukuran, sudut pandang, pencahayaan, dan orientasi — berkat strategi pra-pemrosesan dan augmentasi data yang diterapkan. Hasil macro-average AUC sebesar 0.9297 dan peningkatan akurasi akhir menjadi 87% mengonfirmasi bahwa model tidak hanya memberikan hasil yang lebih baik dan stabil, tetapi juga memiliki kemampuan yang sangat andal (mendekati sempurna) dalam membedakan antar kelas makanan. Temuan ini semakin menguatkan relevansi dan kesiapan model untuk diterapkan dalam skenario aplikasi berbasis mobile seperti GlucoFit, di mana pengguna sering kali mengambil foto makanan dalam kondisi yang tidak terkontrol dan kurang ideal.

## 4.6 Pemasangan Model

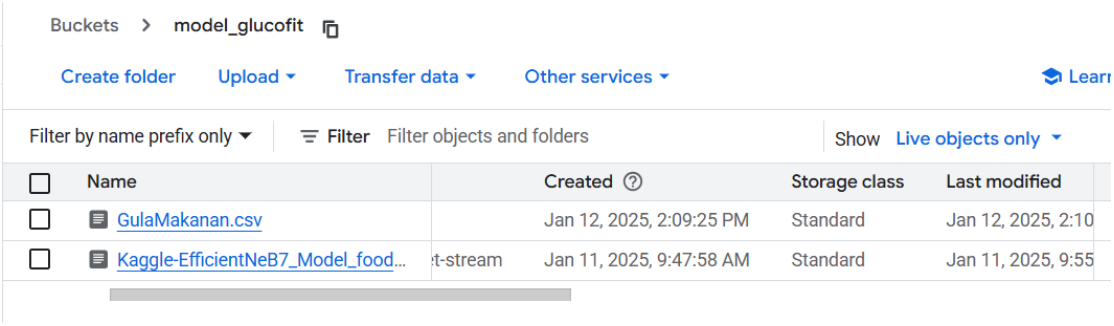
Biasanya, model *machine learning* dikonversi terlebih dahulu ke dalam format *TensorFlow Lite (.tflite*) karena format ini dirancang khusus untuk menjalankan model secara efisien pada perangkat mobile. Namun, dalam kasus tertentu, metode konversi tersebut tidak dapat diterapkan secara langsung karena ukuran model masih terlalu besar meskipun sudah dikonversi. Hal ini menyebabkan kendala saat model diintegrasikan ke dalam lingkungan pengembangan aplikasi seperti *Android Studio*.

Sebagai solusi alternatif, model tidak disimpan secara langsung dalam proyek aplikasi, melainkan disimpan pada penyimpanan eksternal atau layanan *cloud*. Selanjutnya, model dipanggil melalui sebuah API agar hasil prediksi dapat dikirimkan ke aplikasi mobile. Pendekatan ini memungkinkan penggunaan model *machine learning* berukuran besar tanpa mengorbankan kinerja aplikasi di perangkat mobile.

### 4.6.1 Upload Model dan Data Gula Makanan ke *Cloud Storage*

Langkah awal dalam pemasangan model *machine learning* pada aplikasi *mobile* adalah melakukan unggah (upload ) model  yang berformat .h5 (model Keras) dan data gula makanan (csv) ke layanan penyimpanan berbasis *cloud* seperti yang tertera pada gambar 4.19. Dalam penelitian ini, layanan yang digunakan adalah *Google Cloud Console* , dengan memanfaatkan fitur *Cloud Storage* Bucket yang dikhususkan untuk proyek GlucoFit .

Pemilihan Google *Cloud Storage* dimaksudkan untuk memastikan model dan data tersimpan secara aman dan mudah diakses saat dibutuhkan untuk proses inferensi melalui aplikasi *mobile*.




Gambar 4.19 Upload model dan data gula makanan di *Bucket GCP*

### 4.6.2 Deploy di *Cloud* Run

Setelah model dan data gula makanan berhasil diunggah ke Google *Cloud Storage*, langkah selanjutnya adalah melakukan *deployment* logika pemrosesan model ke layanan Google *Cloud* Run . Pada tahap ini, aplikasi *backend* yang dibangun menggunakan framework seperti Flask di*deploy* agar dapat menerima permintaan (*request*) dari aplikasi *mobile* untuk melakukan inferensi.

### 4.6.3 *Cloud* Run Menghasilkan API

Setelah proses *deployment* selesai, Google *Cloud* Run akan menghasilkan sebuah API (Application Programming Interface) berbasis protokol HTTP. API ini berupa alamat URL yang digunakan sebagai antarmuka komunikasi antara aplikasi *mobile* dan model *machine learning* yang berjalan di server *cloud*.

### 4.6.4 API dipakai oleh GlucoFit

Aplikasi *GlucoFit* yang dikembangkan dalam lingkungan Android Studio kemudian memanfaatkan API tersebut untuk mengirimkan data masukan (*input data* ) ke model di *cloud*. Hasil prediksi dari model diterima oleh aplikasi dalam bentuk respons API dan ditampilkan kepada pengguna dalam format yang mudah dipahami.

### 4.6.5 Pengujian Model pada Aplikasi

Pengujian model bertujuan untuk memastikan bahwa model *machine learning* yang telah dideploy di lingkungan *cloud* dapat berfungsi secara optimal ketika diintegrasikan ke dalam aplikasi mobile. Beberapa aspek utama yang menjadi fokus dalam pengujian ini antara lain:

1. Akurasi hasil prediksi:
   Model diuji menggunakan sejumlah data uji untuk memastikan output prediksi sesuai dengan harapan dan memiliki tingkat akurasi yang memadai.
2. Kemampuan pemanggilan API:
   Diperiksa apakah aplikasi dapat memanggil API model secara konsisten tanpa mengalami error.
3. Waktu eksekusi (latency):
   Diukur durasi dari saat aplikasi mengirimkan permintaan hingga menerima respons prediksi dari model. Waktu eksekusi rata-rata berada dalam rentang 2–5 detik, tergantung pada beban server dan kecepatan koneksi internet.

Hasil pengujian menunjukkan bahwa model dapat diakses secara stabil melalui API dan memberikan hasil prediksi yang akurat sesuai dengan data uji. Meskipun terdapat sedikit variasi waktu respons, performa model dinilai memadai untuk mendukung fitur prediktif dalam aplikasi mobile GlucoFit.

## 4.7 Integrasi API Gemini pada Aplikasi (*Android Studio*)

Pada tahap pengembangan aplikasi, salah satu komponen penting adalah penerapan fitur berbasis kecerdasan buatan (AI), yaitu dengan mengintegrasikan model generatif dari Google Gemini. Proses integrasi dilakukan melalui *Google AI API*, khususnya menggunakan *library GenerativeMode*l dari *Google AI Kotlin SDK.*

```
PROGRAM PasangModelGeneratif
MULAI
    model = ModelGeneratif(
        namaModel = "gemini-exp-1206",
        kunciAPI = Konstanta.kunciAPI,
        konfigurasi = buatKonfigurasi() {
            …
        }
    )
SELESAI
```

Gambar 4.20 Pseudocode Konfigurasi Model *Generative AI* (Gemini) dengan Parameter Kustom

Kode pada Gambar 4.20 menunjukkan konfigurasi model generatif Gemini untuk keperluan penelitian. Model diinisialisasi dengan nama "gemini-exp-1206" menggunakan API *key* yang disimpan di *Constants.apikey.*

## *4.8* Penyesuaian Parameter & *Prompt Generative AI* API *Gemini*

Pada tahap ini dilakukan integrasi dan optimasi parameter untuk *Generative AI* API *Gemini* guna mendapatkan hasil yang optimal sesuai kebutuhan penelitian.

### 4.8.1 Penyesuaian Parameter *Generative AI* API Gemini

```
            PROGRAM PengaturanGenerasi
MULAI
    pengaturanGenerasi = {
        FOKUS_JAWABAN = 2.0
        topK = 200
        topP = 0.9
        maksTokenKeluaran = 700
```

```
            jenisRespons = "text/plain"
        }
    SELESAI
```

Gambar 4.21 Pseudocode Konfigurasi Model *Generative AI* (Gemini) dengan Parameter Kustom

Gambar 4.21 mengilustrasikan konfigurasi teknis yang diterapkan pada *Generative AI* API Gemini, khususnya pada komponen generationConfig. Kode ini dirancang untuk memodifikasi dan mengoptimalkan proses generasi teks otomatis sesuai kebutuhan pengguna.

Tabel 4.5 pengaruh Parameter *Generatif AI* (*Gemini*) terhadap Karakteristik *Output*

| **Parameter** | **Nilai Rendah** | **Nilai Tinggi** | **Efek yang Terjadi Pada Nilai Rendah** | **Efek yang Terjadi Pada Nilai Tinggi** |
|---|---|---|---|---|
| temperature | 0.1-0.5 (10-50%) | 1.0-2.0 (100-200%) | Jawaban lebih pasti dan konsisten | Jawaban lebih bervariasi dan kreatif |
| topK | 20-50 token | 100-200 token | Pilihan kata lebih terbatas dan aman | Pilihan kata lebih beragam |
| topP | 0.5-0.7 (50-70%) | 0.8-0.9 (80-90%) | Hanya pakai kata-kata yang paling mungkin | Boleh pakai kata yang kurang umum |
| maxOutputTokens | 100-300 token | 500-700 token | Jawaban lebih singkat | Jawaban lebih panjang. |

Konfigurasi parameter pada *Generative AI* API Gemini pada tabel 4.5 dirancang secara khusus untuk menghasilkan respons yang optimal. Nilai *temperature* sebesar 2f sengaja dipilih tinggi untuk meningkatkan kreativitas dan variasi output, meskipun berpotensi mengurangi konsistensi - nilai yang lebih rendah akan menghasilkan jawaban yang lebih fokus. Parameter *topK* diatur ke 200 token, memberikan model cukup ruang untuk mempertimbangkan berbagai kemungkinan kata terbaik dalam setiap langkah generasi, sementara *topP* 0.9f (90%) berfungsi sebagai batasan probabilitas kumulatif untuk menjaga relevansi output. Pembatasan *maxOutputTokens* sebesar 700 token memastikan respons memiliki kedalaman yang cukup tanpa menjadi bertele-tele. Format *responseMimeType "text/plain"* dipilih untuk kesederhanaan pemrosesan data. Kombinasi parameter ini menghasilkan keseimbangan antara: (1) kreativitas respons yang menjaga *engagement* pengguna, (2) variasi *output* yang mencegah

kebosanan, (3) relevansi konten yang tetap terjaga, dan (4) panjang *respon* yang optimal. Nilai-nilai tersebut merupakan hasil iterasi pengujian untuk mencapai titik optimal dimana model mampu menghasilkan dialog yang alami, informatif, sekaligus menarik tanpa mengorbankan akurasi konten.

### 4.8.2 Penyesuaian *Prompt Generative AI* API Gemini

```
    PROGRAM BuatPenjelasanIlmiah
    MULAI
        FUNGSI buatPenjelasan(namaMakanan: Teks?, kadarGula: Teks?)
            prompt = """
                Buat penjelasan singkat, menarik dan meyakinkan tentang:
                1. Asal usul kadar gula sebanyak [kadarGula] gram dalam
[namaMakanan]
                2. Proses pembuatannya (gunakan perumpamaan kreatif yang
tidak mengurangi selera)
                3. Pengaruhnya terhadap penyakit diabetes (sederhana dan
mudah dimengerti)
                4. Saran aktivitas/olahraga singkat untuk membakar gula
tersebut

                Catatan:
                - Jangan ulangi instruksi ini
                - Hindari contoh menggunakan semut
                - Buat semenarik mungkin!
            """
            coba {
                hasilPenjelasan = mintaPenjelasanDariAI(prompt)
                tampilkan(hasilPenjelasan)
            }
            jika_error {
                tampilkan("Maaf, terjadi kesalahan saat membuat
penjelasan")
            }
        SELESAI
    SELESAI
```

Gambar 4.22 Pseudocode Fungsi generateScientificExplanation() untuk Pembuatan Prompt *Generative AI*

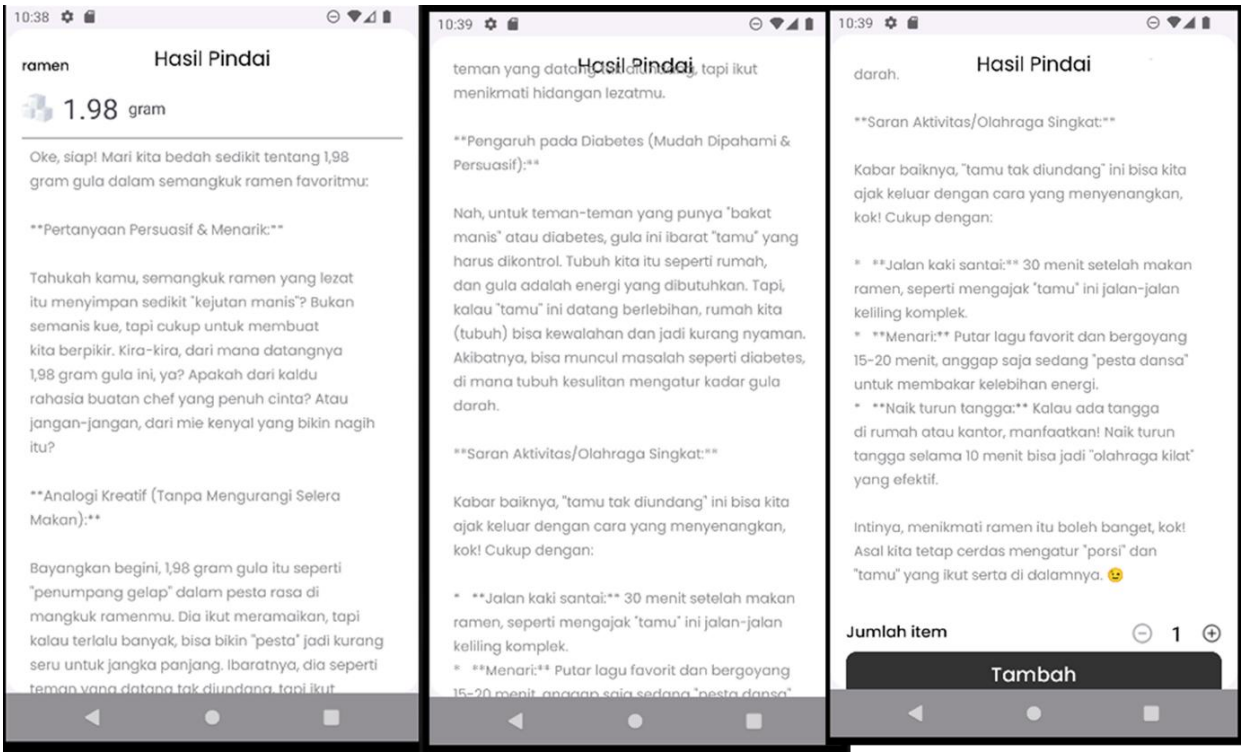
Fungsi *generateScientificExplanation* dirancang untuk menghasilkan penjelasan ilmiah yang informatif dan menarik terkait kandungan gula pada makanan, seperti ditunjukkan pada Gambar 4.8. Fungsi ini membutuhkan dua input, yaitu *foodName* (nama makanan) dan *sugarContent* (jumlah gula). Prompt yang digunakan menginstruksikan AI untuk memberikan penjelasan singkat namun persuasif tentang sumber gula dalam makanan tersebut, dengan penekanan pada proses produksi dan dampaknya terhadap diabetes.

Pembatasan khusus diterapkan pada penggunaan analogi—terutama menghindari penyebutan semut—karena pengalaman menunjukkan bahwa API cenderung menggunakan analogi ini secara berulang, yang berpotensi mengurangi selera makan pengguna. Selain penjelasan, prompt juga meminta rekomendasi aktivitas fisik sederhana untuk membantu menetralisir asupan gula. Output diharapkan disajikan dalam bahasa yang mudah dipahami dan bersifat mengajak.

Fungsi ini dijalankan dalam *viewModelScope* untuk memastikan eksekusi berjalan lancar di latar belakang tanpa mengganggu antarmuka pengguna. Pendekatan ini memadukan aspek edukasi kesehatan dengan pertimbangan psikologis pengguna.

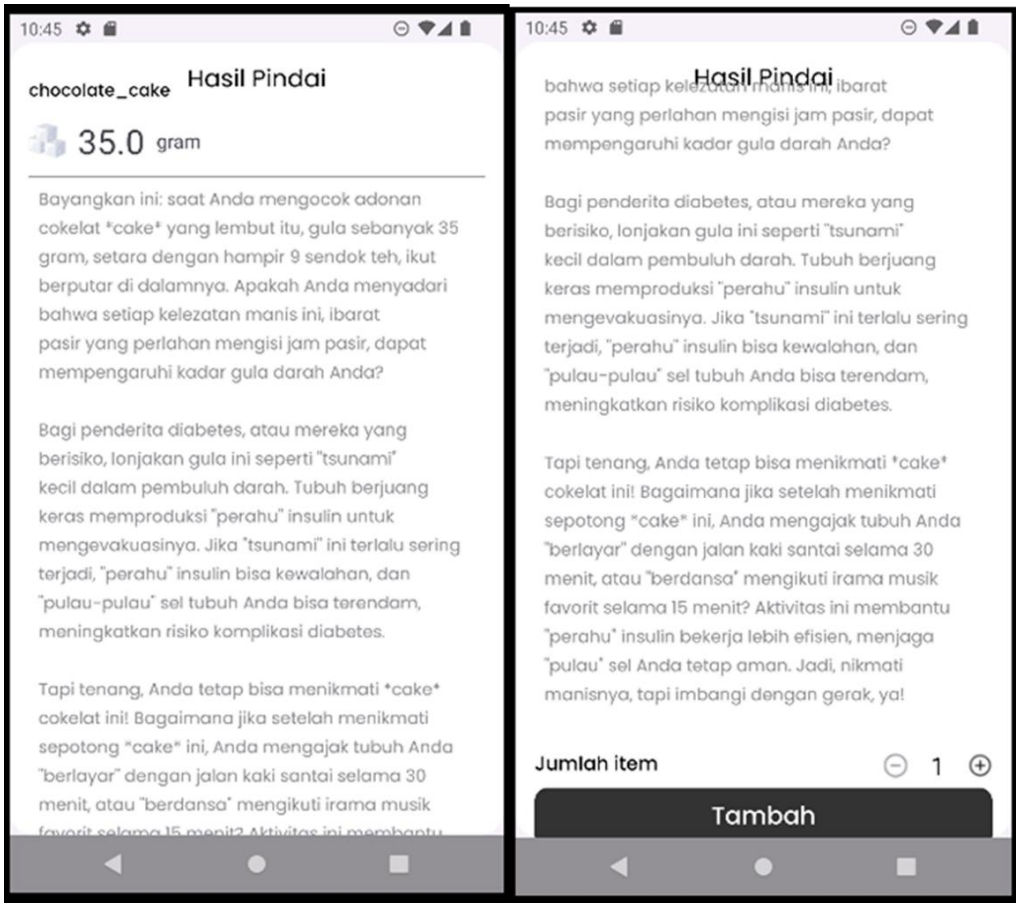
Berikut laporan hasil pengujian fitur pasca instalasi model dan integrasi API. Perlu diperhatikan bahwa angka kandungan gula yang ditampilkan pada gambar 4.23 dan 4.24 berbeda dengan data aktual, karena hasil tersebut diperoleh dari pengujian fitur menggunakan data *dummy/dimmy* sebagai bahan percobaan sistem. Perbedaan ini terjadi dalam rangka uji coba fungsi sebelum diterapkan pada data sebenarnya. Secara fundamental, kandungan gula pada produk ramen memang tergolong rendah, sementara pada chocolate cake secara alami memiliki kadar gula yang tinggi, sehingga tetap mencerminkan karakteristik sebenarnya dari masing-masing produk makanan.

Gambar 4.23 Ilustrasi deskripsi kelompok makanan bergula rendah (Ramen)




Penjelasan mengenai jenis kadar gula rendah terkesan kreatif dan persuasif, berhasil menarik minat pengguna untuk membacanya. Kontennya juga efektif dalam memaparkan bahaya yang dihasilkan oleh ramen. Namun, terdapat kebocoran instruksi seperti kalimat "Pertanyaan Persuasif & menarik" dan "analogi kreatif (Tanpa Mengurangi Selera Makan)", yang dapat mengurangi kesan profesional dari fitur ini. Hal ini terjadi karena keterbatasan kualitas model *generative AI* yang gratis.

Gambar 4.24 Ilustrasi deskripsi kelompok makanan bergula tinggi (Chocolate Cake)




Untuk *Chocolate Cake*, meskipun tidak dijelaskan sumber perhitungan kadar gulanya, penggunaan analogi seperti takaran sendok teh berhasil membuat informasi lebih mudah dipahami oleh orang awam dibandingkan angka mentah. Pada penjelasan ini tidak ada kebocoran perintah seperti ramen. Namun, sayangnya saran aktivitas fisik yang diberikan terkesan generik—hampir identik dengan rekomendasi untuk makanan lain seperti ramen—sehingga kurang ada penyesuaian khusus untuk setiap jenis hidangan. Keterbatasan ini muncul karena kualitas API Gemini yang masih gratis, sehingga hasilnya belum optimal dalam memberikan personalisasi mendetail.

### 4.9 Penilaian Hasil *Generative AI* oleh Para Ahli

Evaluasi kualitas *output Generative AI* API Gemini dilakukan melalui penilaian oleh delapan ahli dengan latar belakang multidisiplin, terdiri atas satu ahli gizi, dua apoteker, dua nutrisionis, satu peneliti kesehatan, satu dosen bidang gizi/kesehatan, dan satu dokter umum.

Dengan melibatkan multidisiplin ilmu, penilaian ini tidak hanya mengukur kualitas teknis *output AI*, tetapi juga memastikan bahwa informasi yang dihasilkan bermanfaat secara klinis, mudah dipahami, dan aman bagi pengguna aplikasi kesehatan seperti GlucoFit.

Tabel 4.6 Penilaian para ahli

| No | Bagian | Aspek Penilaian | Skor Rata-rata | Penilaian |
|---|---|---|---|---|
| **1** | A. Keterbacaan dan Daya Tarik | Informasi yang diberikan oleh API Gemini menarik dan menarik perhatian saya | 4.5 | Sangat Setuju |
| **2** | A. Keterbacaan dan Daya Tarik | Informasi yang diberikan mudah dipahami oleh pengguna awam | 4.6 | Sangat Setuju |
| **3** | A. Keterbacaan dan Daya Tarik | Informasi disajikan dengan cara yang kreatif dan inovatif | 4.5 | Sangat Setuju |
| **4** | B. Relevansi Tinggi | Informasi yang dihasilkan relevan dengan jenis makanan yang diinput | 4.5 | Sangat Setuju |
| **5** | B. Relevansi Tinggi | Informasi sesuai dengan kandungan gula pada makanan tersebut | 4.1 | Mayoritas Setuju |
| **6** | B. Relevansi Tinggi | Informasi membantu memahami hubungan antara makanan dan risiko diabetes | 4.5 | Sangat Setuju |
| **7** | B. Relevansi Tinggi | Informasi mencakup asal bahan dan proses pengolahan makanan | 4.0 | Mayoritas Setuju |
| **8** | C. Konsistensi Informasi | Informasi yang diberikan konsisten dengan sumber ilmiah terpercaya tentang kandungan gula | 4.1 | Mayoritas Setuju |
| **9** | C. Konsistensi Informasi | Informasi yang diberikan tidak bertentangan dengan pengetahuan gizi yang berlaku umum | 4.4 | Sangat Setuju |
| **10** | C. Konsistensi Informasi | Informasi yang diberikan akurat dalam menjelaskan dampak makanan terhadap kadar gula darah | 4.1 | Mayoritas Setuju |
| **11** | D. Manfaat untuk Pengguna | Informasi yang diberikan membantu pengguna membuat keputusan diet yang lebih sehat | 4.6 | Sangat Setuju |
| **12** | D. Manfaat untuk Pengguna | Saran aktivitas/olahraga yang diberikan relevan untuk mengurangi kadar gula darah | 4.0 | Mayoritas Setuju |
| **13** | D. Manfaat untuk Pengguna | Informasi yang diberikan mendukung pencegahan dan manajemen diabetes | 4.4 | Sangat Setuju |
| **14** | D. Manfaat untuk Pengguna | Informasi yang dihasilkan sudah sesuai dengan standar tata laksana ahli gizi atau rekomendasi profesi terkait dalam pencegahan dan manajemen diabetes | 4.0 | Mayoritas Setuju |

| | | | | |
|---|---|---|---|---|
| **15** | E. Umpan Balik Tambahan | Secara keseluruhan, saya puas dengan kualitas informasi yang dihasilkan oleh API Gemini | 4.5 | Sangat Setuju |
| **16** | E. Umpan Balik Tambahan | Saya merekomendasikan penggunaan API Gemini dalam aplikasi kesehatan seperti GlucoFit | 4.5 | Sangat Setuju |

Penilaian terhadap kualitas teks generatif dari API Gemini dilakukan oleh delapan pakar yang terdiri dari ahli gizi dan tenaga medis spesialis diabetes. Evaluasi *menggunakan skala Likert* 1-5 (1=Tidak Setuju, 5=Sangat Setuju) dengan hasil komprehensif yang disajikan dalam Tabel 4.6 Penilaian ini dilakukan untuk memastikan bahwa informasi yang dihasilkan dapat membantu pencegahan dan pengelolaan diabetes secara efektif. Berdasarkan hasil penilaian, sebagian besar ahli menilai bahwa informasi yang diberikan menarik perhatian (50% pada skor 4 dan 5), mudah dipahami oleh masyarakat awam (62,5% pada skor 5), serta disampaikan secara kreatif dan inovatif melalui analogi atau ilustrasi yang relevan (50% pada skor 4 dan 5). Dari segi relevansi, mayoritas ahli menyatakan bahwa informasi yang diberikan sesuai dengan jenis makanan dan kandungan gulanya, serta membantu pengguna memahami hubungannya dengan risiko diabetes. Selain itu, informasi juga dinilai mencakup aspek-aspek penting seperti asal bahan dan proses pengolahan makanan.

Dalam hal konsistensi, sebagian besar ahli menyatakan bahwa informasi selaras dengan sumber ilmiah terpercaya dan tidak bertentangan dengan prinsip-prinsip gizi yang umum berlaku. Meskipun ada beberapa masukan mengenai perlunya penjelasan lebih rinci tentang jenis-jenis gula (seperti glukosa, fruktosa, sukrosa) dan referensi ilmiah yang digunakan, secara umum kualitas informasi dianggap akurat, terutama dalam menjelaskan dampak makanan terhadap kadar gula darah. Dari sisi manfaat bagi pengguna, *API Gemini* dinilai membantu dalam membuat keputusan diet yang lebih sehat, memberikan saran aktivitas fisik yang relevan, serta mendukung upaya pencegahan dan pengelolaan diabetes. Sebagian besar ahli juga menyatakan puas dengan kualitas informasi dan bersedia merekomendasikan penggunaan *API Gemini* dalam aplikasi kesehatan seperti GlucoFit.

Hasil penilaian pada Tabel 4.6 mengungkapkan bahwa beberapa indikator API Gemini memperoleh nilai rata-rata di bawah 4,3. Nilai tersebut tetap tergolong baik, namun menandakan perlunya penyempurnaan untuk mencapai kinerja yang lebih optimal. Khusus untuk peningkatan kualitas layanan informasi. Aspek pertama dengan skor 4,1 adalah "informasi sesuai dengan kandungan gula pada makanan tersebut". Hal ini mengindikasikan bahwa meskipun mayoritas ahli menyetujui, masih terdapat keraguan mengenai akurasi data

kandungan gula. Kemungkinan penyebabnya meliputi variasi produk makanan, keterbatasan basis data, atau kurangnya detail mengenai jenis gula spesifik.

Aspek kedua yang memerlukan perbaikan adalah "informasi mencakup asal bahan dan proses pengolahan makanan tersebut" dengan skor 4,0. Skor ini menunjukkan bahwa informasi yang diberikan belum cukup komprehensif dalam menjelaskan asal bahan baku dan pengaruh proses pengolahan terhadap kadar gula. Padahal, informasi tersebut sangat penting untuk menilai dampak makanan secara lebih menyeluruh.

Pada aspek "informasi yang diberikan konsisten dengan sumber ilmiah terpercaya tentang kandungan gula" yang memperoleh skor 4,1, terlihat perlunya penjelasan lebih rinci mengenai sumber referensi. Validasi data berdasarkan literatur ilmiah terbaru juga perlu ditingkatkan untuk memastikan keakuratan informasi.

Aspek akurasi dalam menjelaskan dampak makanan terhadap kadar gula darah juga mendapat skor 4,1. Skor ini mencerminkan bahwa informasi yang diberikan masih terlalu umum dan belum mempertimbangkan faktor individu seperti indeks glikemik atau profil kesehatan spesifik pengguna.

Untuk aspek rekomendasi aktivitas fisik, skor 4,0 menunjukkan bahwa saran olahraga yang diberikan belum sepenuhnya spesifik dan disesuaikan dengan kebutuhan individual pengguna. Demikian pula dengan aspek kesesuaian dengan standar tata laksana diabetes yang juga memperoleh skor 4,0, mengindikasikan perlunya validasi lebih lanjut oleh tenaga kesehatan profesional agar rekomendasi sesuai dengan pedoman medis terkini.

Selain penilaian kuantitatif, para ahli memberikan berbagai masukan konstruktif pada bagian komentar terbuka untuk penyempurnaan parameter dan *prompt API Gemini*. Beberapa saran perbaikan mencakup penyebutan jenis gula spesifik (glukosa, fruktosa, sukrosa, laktosa) beserta dampaknya terhadap gula darah, penyertaan referensi ilmiah, konversi takaran ke satuan rumah tangga (sdm/sdt), penyederhanaan teks agar lebih ringkas, serta penambahan informasi nutrisi seperti kalori dan protein. Para ahli juga merekomendasikan inklusi konten tambahan seperti jenis gula yang aman untuk diabetes, panduan aktivitas fisik sesuai standar Kemenkes RI, dan visualisasi proses metabolisme melalui ilustrasi/animasi – khususnya untuk pengguna lansia dengan penyesuaian ukuran font dan warna.

Di sisi lain, umpan balik positif menyoroti kualitas informasi yang sudah informatif dan bahasa yang persuasif, sehingga API Gemini dinilai layak diintegrasikan dalam aplikasi kesehatan seperti *GlucoFit*. Kombinasi antara apresiasi terhadap kelebihan sistem dan rekomendasi spesifik ini memberikan peta jalan yang jelas untuk peningkatan akurasi,

aksesibilitas, dan kedalaman konten guna optimalisasi manfaat bagi pengguna, terutama dalam konteks pencegahan dan manajemen diabetes.

Berdasarkan analisis terhadap nilai dan komentar para ahli, beberapa keterbatasan pada output generative AI yang digunakan dapat diatasi melalui peningkatan model generatif yang lebih mutakhir. Implementasi model generatif terkini seperti OpenAI GPT-4, OpenAI GPT-4 Turbo, Claude 3 Opus, atau Gemini 1.5 yang memiliki kapabilitas lebih canggih diharapkan mampu mengatasi berbagai kekurangan yang diidentifikasi oleh para ahli.

# BAB V
# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

a. Model *Convolutional Neural Network (CNN)* dengan arsitektur *EfficientNetB7* berhasil dikembangkan untuk mengidentifikasi jenis makanan dari foto input pengguna. Setelah melalui proses *preprocessing* data, pelatihan, dan evaluasi, model mencapai akurasi rata-rata sebesar 87% setelah penerapan transformasi gambar tambahan. Selain itu, untuk mendukung informasi edukatif, aplikasi mengintegrasikan Google Gemini API guna menghasilkan penjelasan ilmiah terkait kandungan gula, dampaknya terhadap diabetes, serta saran aktivitas fisik.

b. Dengan kemampuan aplikasi dalam memprediksi kandungan gula pada makanan dan menyampaikan informasi edukatif secara real-time, pengguna dapat membuat keputusan makanan yang lebih tepat dan sadar akan risiko konsumsi gula berlebihan. Integrasi rekomendasi aktivitas fisik juga memperkuat aspek preventif aplikasi, karena aktivitas fisik merupakan faktor penting dalam menurunkan risiko diabetes tipe 2. Pendekatan ini sangat relevan dalam konteks pencegahan dini penyakit metabolik seperti diabetes, terutama di kalangan individu dengan risiko tinggi.

c. Hasil penilaian oleh delapan tenaga ahli menunjukkan bahwa *output* teks yang dihasilkan oleh API Gemini memiliki tingkat keterbacaan tinggi, relevansi baik, konsistensi informasi dengan referensi ilmiah, serta manfaat signifikan bagi pengguna. Mayoritas ahli memberikan respon positif terhadap penggunaan teknologi ini dalam konteks edukasi kesehatan, khususnya dalam meningkatkan pemahaman masyarakat terhadap pola makan dan pencegahan diabetes. Hal ini membuktikan bahwa integrasi *LLM (Large Language Model*) dalam sistem edukasi kesehatan memiliki potensi besar untuk dikembangkan lebih lanjut.

## 5.2 Saran

a. Pengembangan Model *Machine Learning* dengan Kelas Makanan yang Lebih Variatif: Untuk pengembangan selanjutnya, disarankan memperluas variasi makanan dalam model dengan menambahkan lebih banyak makanan tradisional Indonesia seperti gudeg, rendang, atau nasi padang. Data dapat dikumpulkan melalui metode pengambilan gambar otomatis dari internet (web scraping) dari berbagai sumber resep dan konten kuliner online, kemudian diseleksi secara

manual untuk memastikan kualitas dan relevansinya. Setiap jenis makanan sebaiknya memiliki jumlah data yang seimbang, idealnya sekitar 1.000 gambar per kategori, untuk menghindari ketimpangan data yang dapat mempengaruhi akurasi model. Pendekatan ini akan membuat model lebih representatif dalam mengenali keragaman makanan khas Indonesia sekaligus meningkatkan keakuratannya.

b. Penambahan Visualisasi Alat Rumah Tangga untuk Edukasi Kandungan Gula: Sebagai tambahan edukasi bagi pengguna, disarankan untuk menyertakan visualisasi alat rumah tangga seperti sendok makan, sendok teh, atau gelas dalam penyajian informasi kandungan gula. Visualisasi ini akan memudahkan pemahaman masyarakat awam terhadap jumlah gula dalam suatu hidangan, serta membantu mereka dalam memperkirakan asupan harian secara lebih intuitif.
c. Pemanfaatan API Generatif dengan Kemampuan Output yang Lebih Informatif dan Efisien: Untuk meningkatkan kualitas informasi yang diberikan kepada pengguna, disarankan untuk melakukan optimasi pada pemanfaatan API generatif. Pendekatan yang dapat dilakukan meliputi penggunaan API yang lebih canggih atau penerapan teknik fine-tuning dan prompt-engineering yang lebih matang. Hal ini bertujuan untuk menghasilkan output yang lebih informatif dengan menyertakan detail penting seperti konversi kandungan gula dari satuan gram ke dalam takaran porsi atau ukuran rumah tangga (misalnya: "X gram gula setara dengan Y sendok makan per Z gram sajian"). Selain itu, perlu dilakukan pengoptimalan waktu respons API untuk memastikan pengalaman pengguna tetap optimal tanpa menunggu terlalu lama, sekaligus menjaga kualitas output agar tetap kreatif, akurat, dan mudah dipahami dalam penyajian informasinya. Penyempurnaan ini akan membuat fitur penjelasan kandungan gula menjadi lebih bermanfaat sekaligus meningkatkan nilai edukasi bagi pengguna.

# DAFTAR PUSTAKA

Irawan, P. N. C. (2024). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) DALAM KLASIFIKASI GAMBAR MAKANAN UNTUK MENENTUKAN KANDUNGAN KALORI PADA MAKANAN.

Aarsaether, A. N. (2024). *Open* APIs *for Healthcare System Interoperability*.

Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), *Artificial Intelligence Applications and Innovations* (pp. 373–383). Springer International Publishing.

Alanazi, A. (2022). Using *machine learning* for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, *30*, 100924. https://doi.org/https://doi.org/10.1016/j.imu.2022.100924

Ardiansyah, R., Fajarulloh, A. C., & Putra, B. A. S. (2025.). Penggunaan CNN untuk menentukan jumlah kalori pada sayuran dan buah menggunakan image processing. Seminar Nasional Teknologi & Sains, 4(1), 571-579. https://doi.org/10.29407/5dzfn808

Dagliati, A., Marini, S., Sacchi, L., Cogni, G., Teliti, M., Tibollo, V., de Cata, P., Chiovato, L., & Bellazzi, R. (2017). *Machine Learning* Methods to Predict Diabetes Complications. *Journal of Diabetes Science and Technology*, *12*(2), 295–302. https://doi.org/10.1177/1932296817706375

Dritsas, E., & Trigka, M. (2022). Data-Driven Machine-Learning Methods for Diabetes Risk Prediction. *Sensors*, *22*(14). https://doi.org/10.3390/s22145304

Gordon, W. J., & Rudin, R. S. (2022). Why APIs? Anticipated value, barriers, and opportunities for standards-based application programming interfaces in healthcare: perspectives of US thought leaders. *JAMIA Open*, *5*(2), ooac023. https://doi.org/10.1093/jamiaopen/ooac023

Guan, Z., Li, H., Liu, R., Cai, C., Liu, Y., Li, J., Wang, X., Huang, S., Wu, L., Liu, D., Yu, S., Wang, Z., Shu, J., Hou, X., Yang, X., Jia, W., & Sheng, B. (2023a). Artificial intelligence in diabetes management: Advancements, opportunities, and challenges. In *Cell Reports Medicine* (Vol. 4, Issue 10). Cell Press. https://doi.org/10.1016/j.xcrm.2023.101213

Hau, K., Hassan, S., & Zhou, S. (2025). *LLMs in Mobile Apps: Practices, Challenges, and Opportunities*. http://arxiv.org/abs/2502.15908

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. de, Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M.,

Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge Graphs. *ACM Comput. Surv.*, *54*(4). https://doi.org/10.1145/3447772

Irawan, P. N. C. (2024). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) DALAM KLASIFIKASI GAMBAR MAKANAN UNTUK MENENTUKAN KANDUNGAN KALORI PADA MAKANAN.

Khanam, J. J., & Foo, S. Y. (2021). A comparison of *machine learning* algorithms for diabetes prediction. *ICT Express*, *7*(4), 432–439. https://doi.org/10.1016/j.icte.2021.02.004

Koneni, V. (n.d.). Article ID: IJCET_15_04_019 Cite this Article: Venugopal Koneni, Revolutionizing Healthcare Services Through API Integration: A Detailed Case Study. *International Journal of Computer Engineering and Technology (IJCET)*, *15*(4), 226–237. https://doi.org/10.5281/zenodo.13150848

Kufel, J., Bargieł-Łączek, K., Kocot, S., Koźlik, M., Bartnikowska, W., Janik, M., Czogalik, Ł., Dudek, P., Magiera, M., Lis, A., Paszkiewicz, I., Nawrat, Z., Cebula, M., & Gruszczyńska, K. (2023). What Is *Machine Learning*, Artificial Neural Networks and *Deep Learning*?—Examples of Practical Applications in Medicine. Diagnostics, 13(15), 2582. https://doi.org/10.3390/diagnostics13152582

Lai, H., Huang, H., Keshavjee, K., Guergachi, A., & Gao, X. (2019). Predictive models for diabetes mellitus using *machine learning* techniques. *BMC Endocrine Disorders*, *19*(1), 101. https://doi.org/10.1186/s12902-019-0436-6

*Laporan SKI TEMATIK 2023*. (n.d.). https://www.badankebijakan.kemkes.go.id/laporan-tematik-ski/

Makroum, M. A., Adda, M., Bouzouane, A., & Ibrahim, H. (2022). *Machine Learning* and Smart Devices for Diabetes Management: Systematic Review. In *Sensors* (Vol. 22, Issue 5). MDPI. https://doi.org/10.3390/s22051843

*million*. (n.d.).

Martinez, J. (2025). Manual data entry vs automated data entry: Key differences. DocuClipper. https://www.docuclipper.com/blog/manual-data-entry-vs-automated-data-entry/#:~:text=Automated%20systems%20offer%20long%2Dterm,and%20less%20efficient%20over%20time

Oktora, S. I., & Butar, D. B. (2022). Determinants of Diabetes Mellitus Prevalence in Indonesia. *Kemas*, *18*(2), 266–273. https://doi.org/10.15294/kemas.v18i2.31880

Olimjonovna, K. O. (2024). UNDERSTANDING THE CAUSES AND RISK FACTORS OF DIABETES.

Panchal, R. K., Bhimrao Bhoite Vishwanath Karad, S., Pande, A., Patil, R., Mukkemwar, R., Panchal, R., & Bhoite, S. (2024). *Comprehensive Study of Google Gemini and Text Generating Models: Understanding Capabilities and Performance*. https://www.researchgate.net/publication/386101230

Pedro, B. (2024). Building an API Product: Design, implement, release, and maintain API products that meet user needs. Packt Publishing Ltd.

Ratri Enggar Pawening, Meliana Eka Puteri, Agmelita Dwi Jianika, & Fitriyah Hidayati. (2023). Deteksi Objek pada Citra Makanan Sebagai Rekomendasi Diet Menggunakan Metode Mask R-CNN. Jurnal Ilmiah Teknik Informatika Dan Komunikasi, 4(1), 87–97. https://doi.org/10.55606/juitik.v4i1.733

Rayan, Z., Alfonse, M., & Salem, A.-B. M. (2019). *Machine Learning* Approaches in Smart Health. *Procedia Computer Science*, *154*, 361–368. https://doi.org/10.1016/j.procs.2019.06.052

Salim, N. O. M., Zeebaree, S. R. M., Sadeeq, M. A. M., Radie, A. H., Shukur, H. M., & Rashid, Z. N. (2021). Study for Food Recognition System Using *Deep Learning*. *Journal of Physics: Conference Series*, *1963*(1). https://doi.org/10.1088/1742-6596/1963/1/012014

Sekawan Media. (2021). *Sistem pakar: Pengertian, tujuan, struktur, dan metode* . https://www.sekawanmedia.co.id/blog/sistem-pakar/

Sharma, T., & Shah, M. (2021). A comprehensive review of *machine learning* techniques on diabetes detection. *Visual Computing for Industry, Biomedicine, and Art*, *4*(1), 30. https://doi.org/10.1186/s42492-021-00097-7

Team, G., Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, A., Chen, J., Pitler, E., Lillicrap, T., Lazaridou, A., … Vinyals, O. (2025). *Gemini: A Family of Highly Capable Multimodal Models*. https://arxiv.org/abs/2312.11805

Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. Computation, 11(3), 52. https://doi.org/10.3390/computation11030052

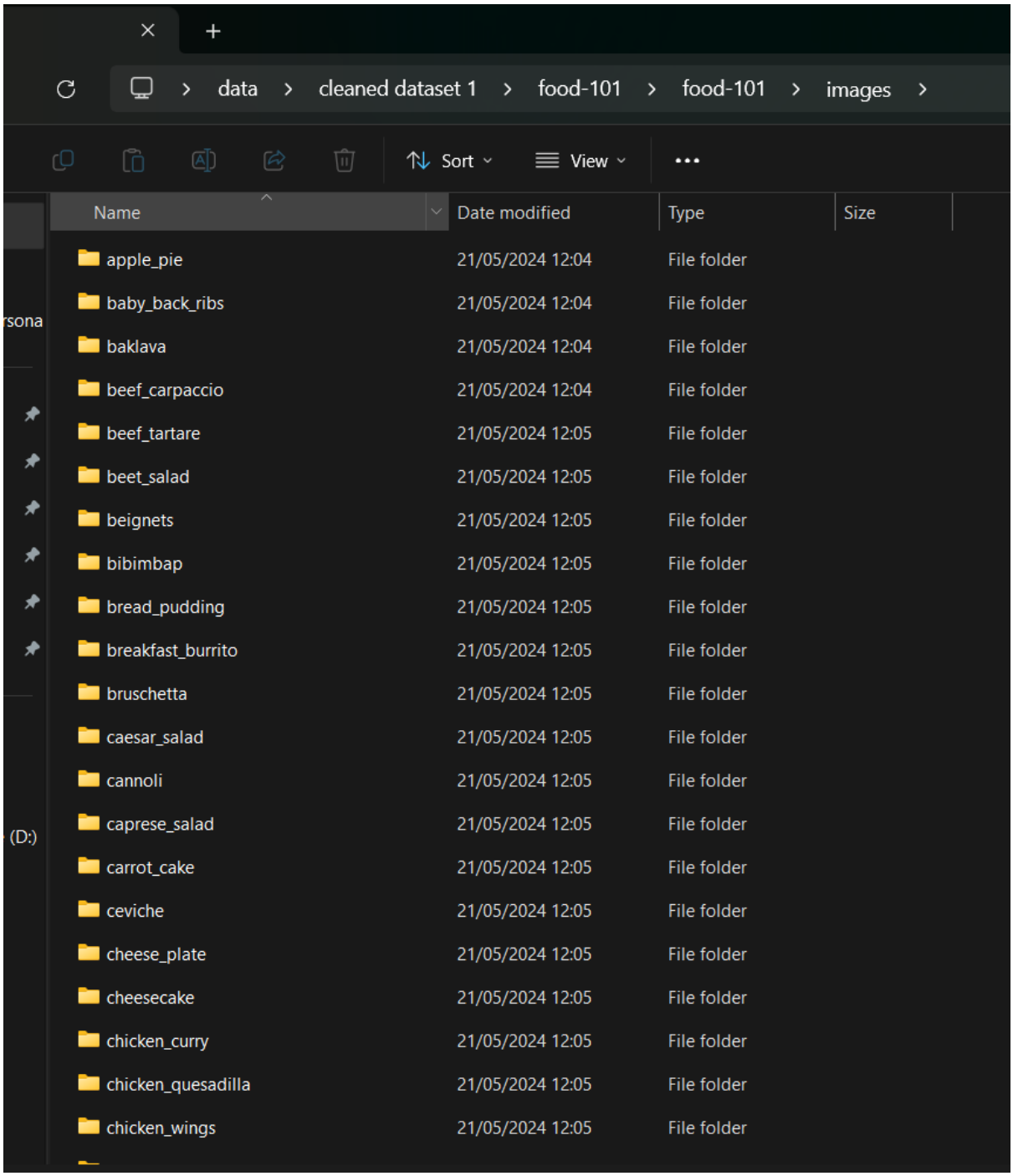
Ulfa, M., Setyonugroho, W., Lestari, T., Widiasih, E., & Nguyen Quoc, A. (2022). Nutrition-Related Mobile Application for Daily Dietary Self-Monitoring. In *Journal of Nutrition and Metabolism* (Vol. 2022). Hindawi Limited. https://doi.org/10.1155/2022/2476367

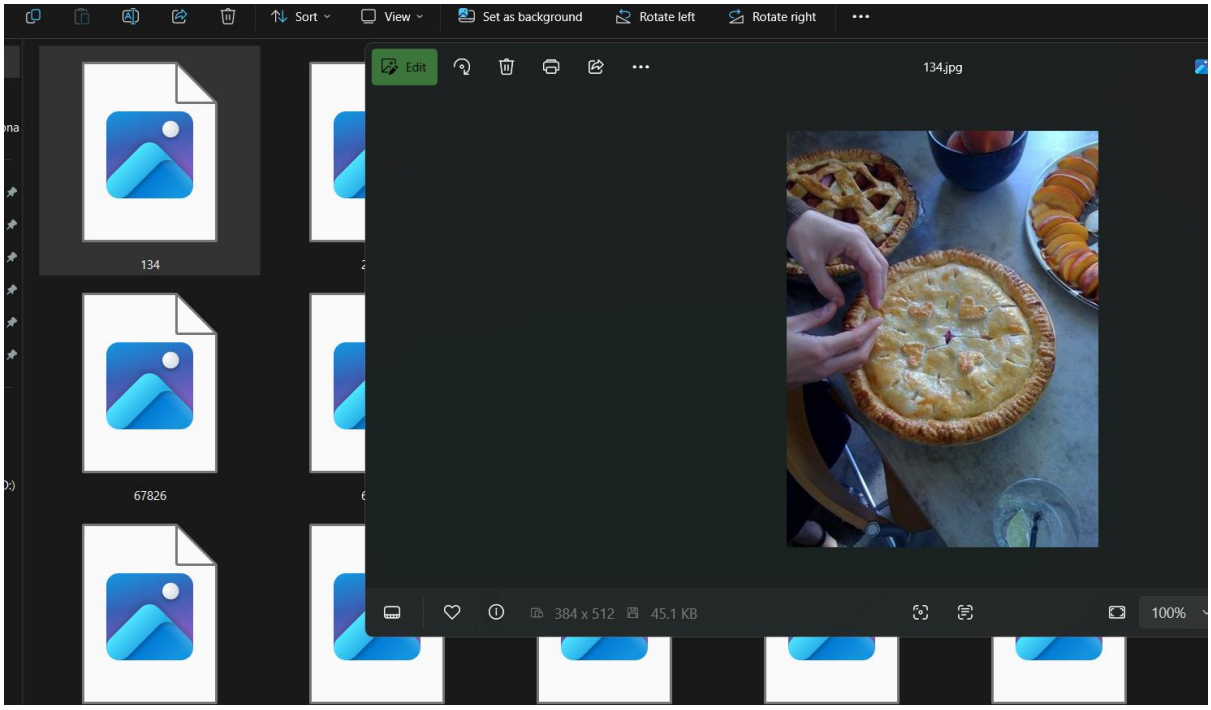
Yang, T., & Ying, Y. (2022). AUC Maximization in the Era of Big Data and AI: A Survey. ACM Comput. Surv., 55(8). https://doi.org/10.1145/3554729

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting Diabetes Mellitus With *Machine Learning* Techniques. *Frontiers in Genetics*, *9*. https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2018.00515

# LAMPIRAN

1. Dataset yang telah dibersihkan dan siap dilatih.

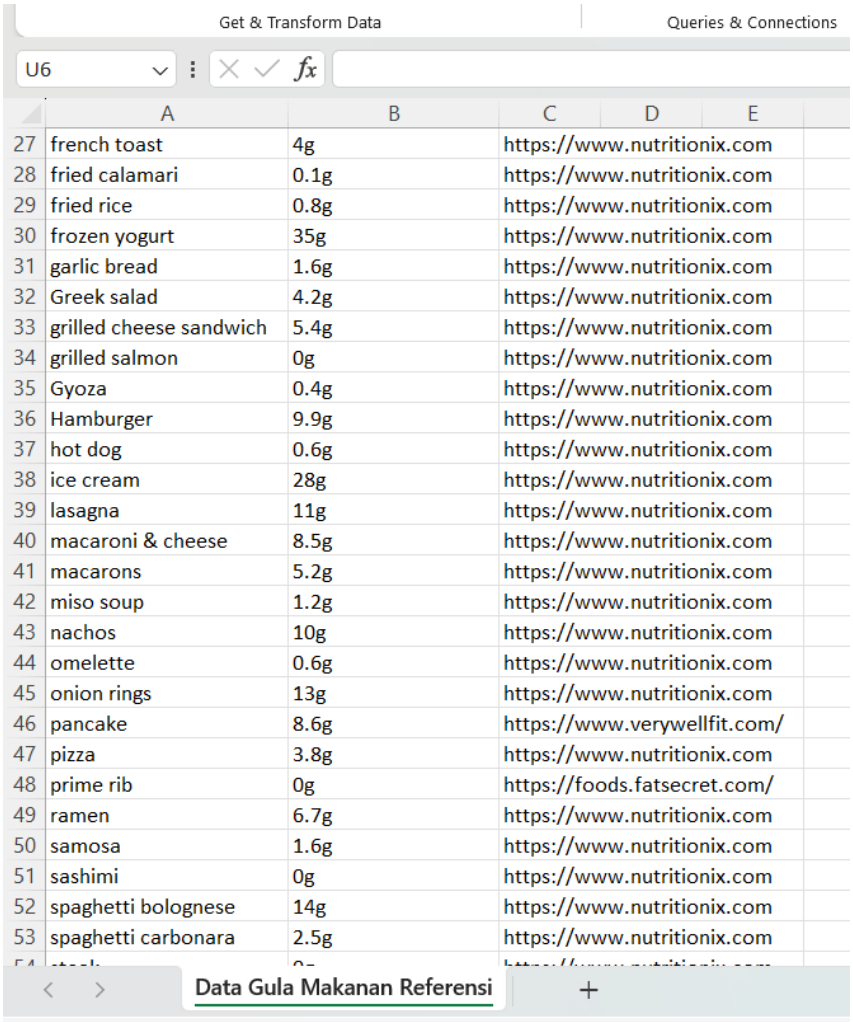




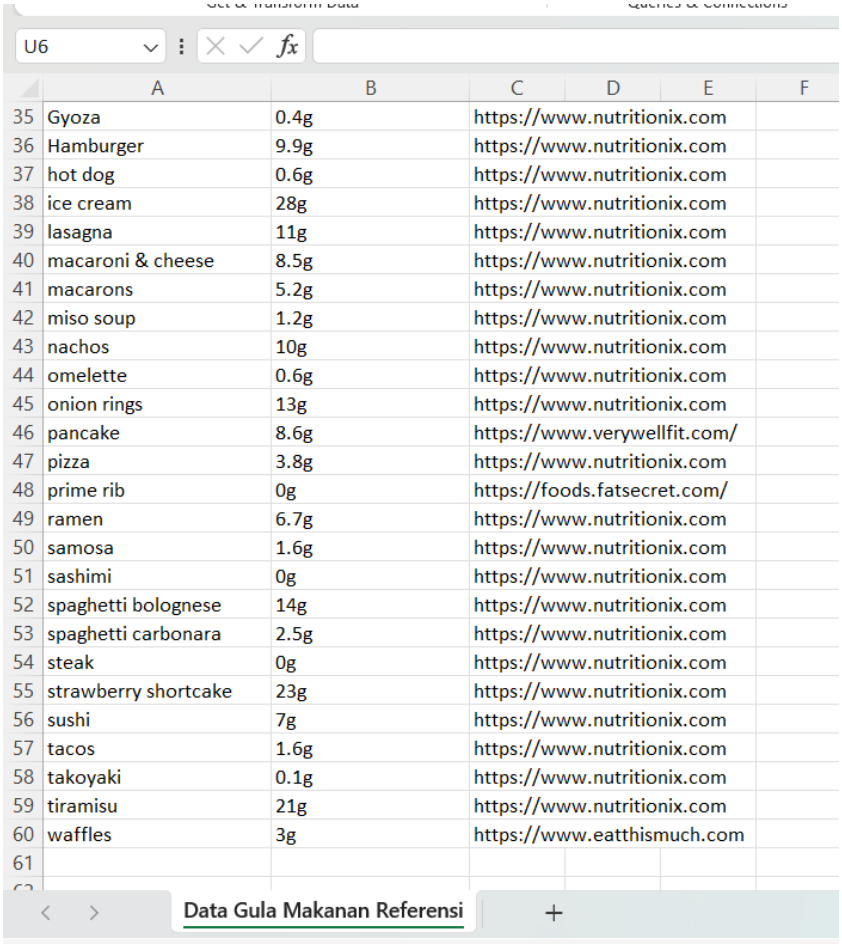

2. Data gula makanan serta referensinya.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | food | sugar(g) | Referensi Link | | | | |
| 2 | apple pie | 16g | https://www.nutritionix.com | | | | |
| 3 | baby back ribs | 1.8g | https://www.nutritionix.com/food/baby-back-ribs | | | | |
| 4 | Baklava | 13g | https://www.nutritionix.com | | | | |
| 5 | Beet Salad | 18g | https://www.nutritionix.com/food/beet-salad | | | | |
| 6 | beignets | 31.09g | https://www.nutritionix.com | | | | |
| 7 | bibimbap | 1.52g | https://www.nutritionix.com | | | | |
| 8 | breakfast burrito | 1.4g | https://www.nutritionix.com | | | | |
| 9 | caesar salad | 4.9g | https://www.nutritionix.com | | | | |
| 10 | caprese salad | 2.8g | https://www.nutritionix.com | | | | |
| 11 | Cannoli | 6.3g | https://www.nutritionix.com | | | | |
| 12 | cheesecake | 27g | https://www.nutritionix.com | | | | |
| 13 | cheese Plate | 0.5g | https://www.nutritionix.com | | | | |
| 14 | chicken curry | 2g | https://www.nutritionix.com | | | | |
| 15 | chicken wing | 0.1g | https://www.nutritionix.com | | | | |
| 16 | chocolate cake | 44g | https://www.nutritionix.com | | | | |
| 17 | Chocolate mousse | 30g | https://www.nutritionix.com | | | | |
| 18 | churros | 4.8g | https://www.nutritionix.com | | | | |
| 19 | club sandwich | 7.1g | https://www.nutritionix.com | | | | |
| 20 | Creme Brulee | 30g | https://www.nutritionix.com | | | | |
| 21 | cupcake | 30g | https://www.nutritionix.com | | | | |
| 22 | donuts | 14g | https://www.nutritionix.com | | | | |
| 23 | dumplings | 0.5g | https://www.nutritionix.com | | | | |
| 24 | falafel | 0.6g | https://www.nutritionix.com | | | | |
| 25 | fish and chips | 2g | https://www.nutritionix.com | | | | |
| 26 | french fries | 2g | https://www.nutritionix.com | | | | |
| 27 | french toast | 4g | https://www.nutritionix.com | | | | |

Data Gula Makanan Referensi



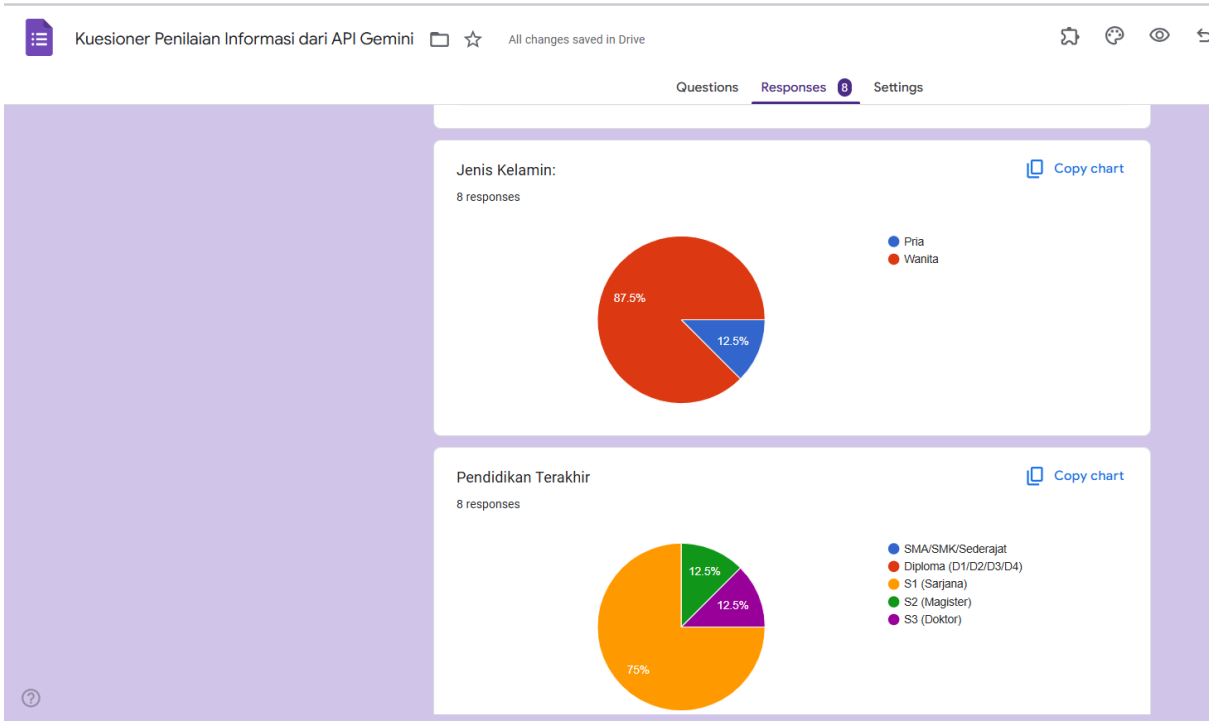

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 27 | french toast | 4g | https://www.nutritionix.com | | |
| 28 | fried calamari | 0.1g | https://www.nutritionix.com | | |
| 29 | fried rice | 0.8g | https://www.nutritionix.com | | |
| 30 | frozen yogurt | 35g | https://www.nutritionix.com | | |
| 31 | garlic bread | 1.6g | https://www.nutritionix.com | | |
| 32 | Greek salad | 4.2g | https://www.nutritionix.com | | |
| 33 | grilled cheese sandwich | 5.4g | https://www.nutritionix.com | | |
| 34 | grilled salmon | 0g | https://www.nutritionix.com | | |
| 35 | Gyoza | 0.4g | https://www.nutritionix.com | | |
| 36 | Hamburger | 9.9g | https://www.nutritionix.com | | |
| 37 | hot dog | 0.6g | https://www.nutritionix.com | | |
| 38 | ice cream | 28g | https://www.nutritionix.com | | |
| 39 | lasagna | 11g | https://www.nutritionix.com | | |
| 40 | macaroni & cheese | 8.5g | https://www.nutritionix.com | | |
| 41 | macarons | 5.2g | https://www.nutritionix.com | | |
| 42 | miso soup | 1.2g | https://www.nutritionix.com | | |
| 43 | nachos | 10g | https://www.nutritionix.com | | |
| 44 | omelette | 0.6g | https://www.nutritionix.com | | |
| 45 | onion rings | 13g | https://www.nutritionix.com | | |
| 46 | pancake | 8.6g | https://www.verywellfit.com/ | | |
| 47 | pizza | 3.8g | https://www.nutritionix.com | | |
| 48 | prime rib | 0g | https://foods.fatsecret.com/ | | |
| 49 | ramen | 6.7g | https://www.nutritionix.com | | |
| 50 | samosa | 1.6g | https://www.nutritionix.com | | |
| 51 | sashimi | 0g | https://www.nutritionix.com | | |
| 52 | spaghetti bolognese | 14g | https://www.nutritionix.com | | |
| 53 | spaghetti carbonara | 2.5g | https://www.nutritionix.com | | |



| | A | B | C |
|---|---|---|---|
| 35 | Gyoza | 0.4g | https://www.nutritionix.com |
| 36 | Hamburger | 9.9g | https://www.nutritionix.com |
| 37 | hot dog | 0.6g | https://www.nutritionix.com |
| 38 | ice cream | 28g | https://www.nutritionix.com |
| 39 | lasagna | 11g | https://www.nutritionix.com |
| 40 | macaroni & cheese | 8.5g | https://www.nutritionix.com |
| 41 | macarons | 5.2g | https://www.nutritionix.com |
| 42 | miso soup | 1.2g | https://www.nutritionix.com |
| 43 | nachos | 10g | https://www.nutritionix.com |
| 44 | omelette | 0.6g | https://www.nutritionix.com |
| 45 | onion rings | 13g | https://www.nutritionix.com |
| 46 | pancake | 8.6g | https://www.verywellfit.com/ |
| 47 | pizza | 3.8g | https://www.nutritionix.com |
| 48 | prime rib | 0g | https://foods.fatsecret.com/ |
| 49 | ramen | 6.7g | https://www.nutritionix.com |
| 50 | samosa | 1.6g | https://www.nutritionix.com |
| 51 | sashimi | 0g | https://www.nutritionix.com |
| 52 | spaghetti bolognese | 14g | https://www.nutritionix.com |
| 53 | spaghetti carbonara | 2.5g | https://www.nutritionix.com |
| 54 | steak | 0g | https://www.nutritionix.com |
| 55 | strawberry shortcake | 23g | https://www.nutritionix.com |
| 56 | sushi | 7g | https://www.nutritionix.com |
| 57 | tacos | 1.6g | https://www.nutritionix.com |
| 58 | takoyaki | 0.1g | https://www.nutritionix.com |
| 59 | tiramisu | 21g | https://www.nutritionix.com |
| 60 | waffles | 3g | https://www.eatthismuch.com |
| 61 | | | |

Data Gula Makanan Referensi

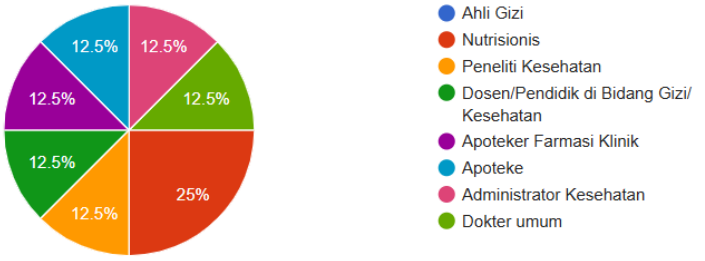
3. Penilaian para ahli untuk hasil generative AI



Profesi/Jabatan Saat Ini:

Copy chart

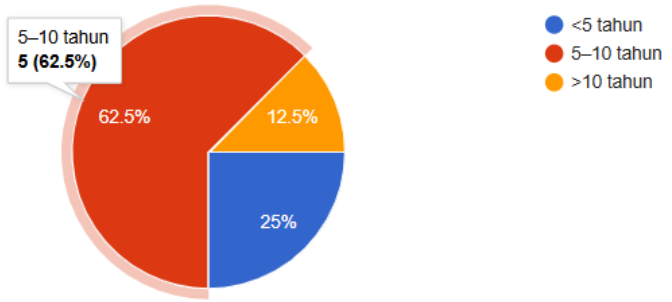
8 responses




Pengalaman Kerja di Bidang Gizi/Kesehatan (dalam tahun):

Copy chart

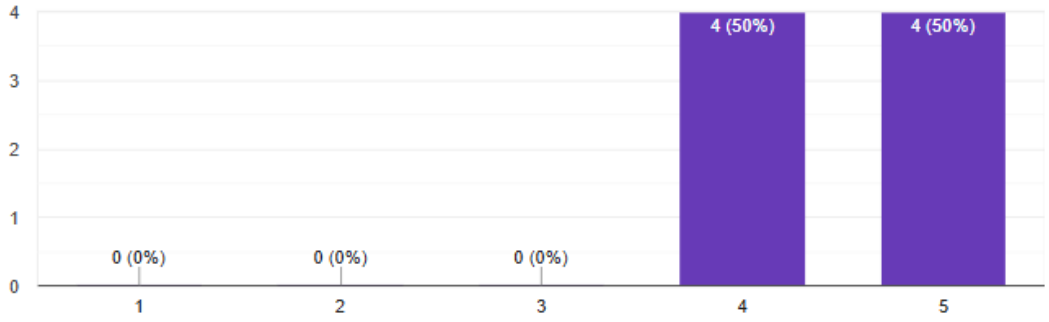
8 responses



Informasi yang diberikan oleh API Gemini menarik dan menarik perhatian saya.

Copy chart

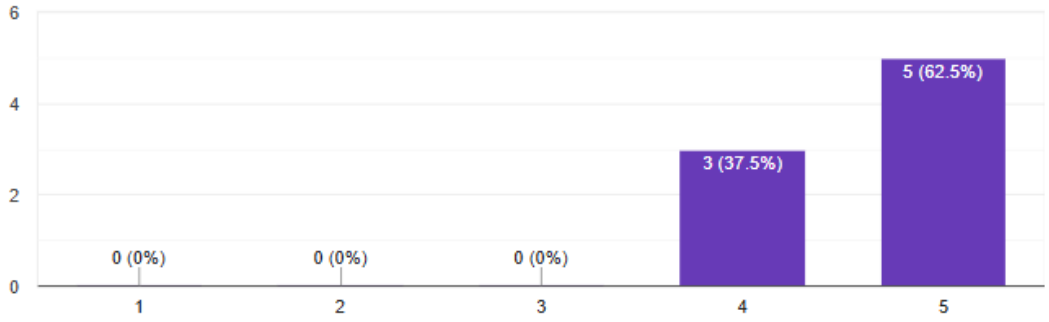
8 responses




Informasi yang diberikan mudah dipahami oleh pengguna awam.

Copy chart

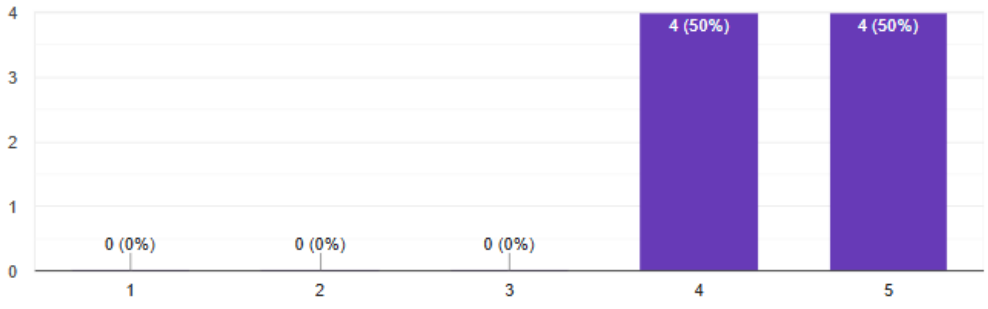
8 responses




Informasi disajikan dengan cara yang kreatif dan inovatif (misalnya, melalui analogi atau ilustrasi).

Copy chart

8 responses

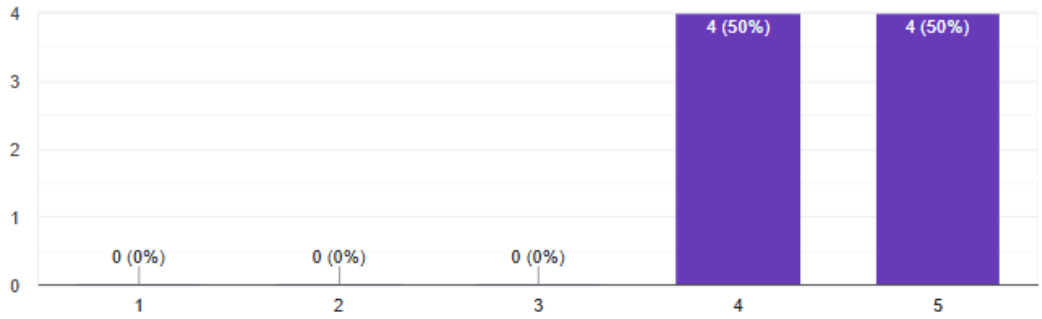


**B. Relevansi Tinggi**

Informasi yang dihasilkan relevan dengan jenis makanan yang diinput.

Copy chart

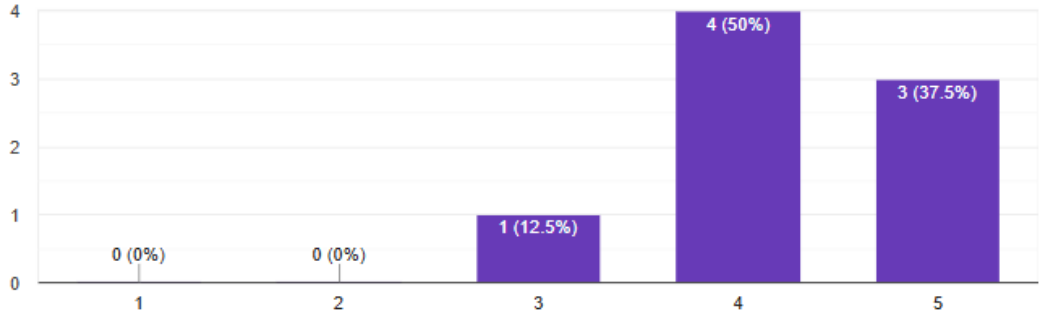
8 responses




Informasi yang diberikan sesuai dengan kandungan gula pada makanan tersebut.

Copy chart

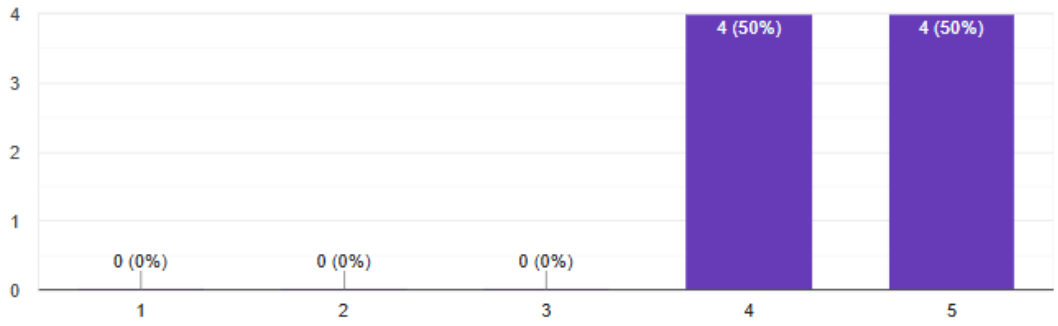
8 responses



Informasi membantu pengguna memahami hubungan antara makanan dan risiko diabetes.

Copy chart

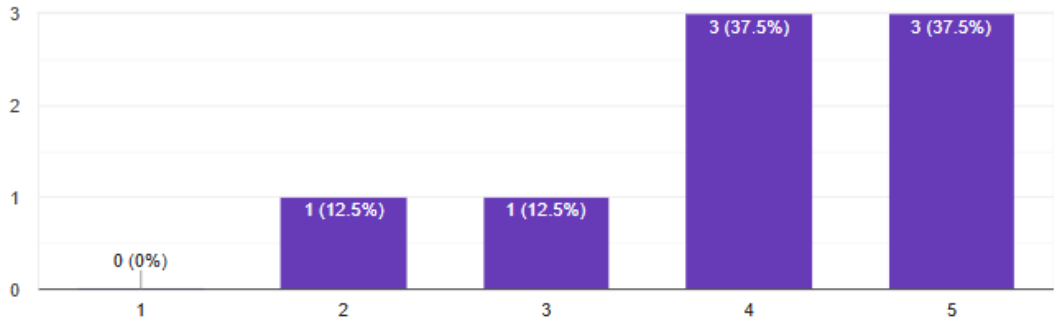
8 responses




Informasi mencakup aspek-aspek penting seperti asal bahan dan proses pengolahan makanan.

Copy chart

8 responses

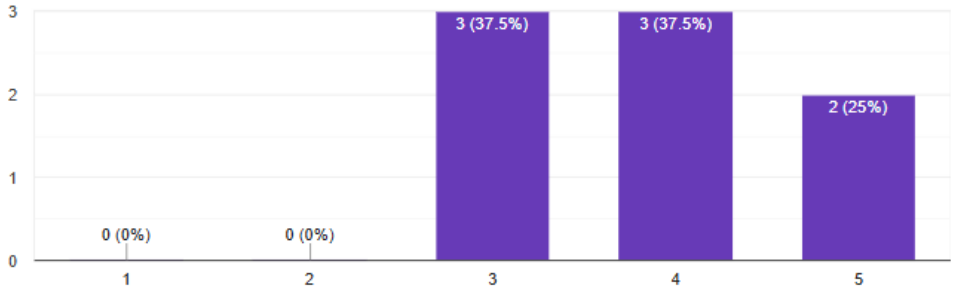


## C. Konsistensi Informasi

Informasi yang diberikan konsisten dengan sumber ilmiah terpercaya tentang kandungan gula.

Copy chart

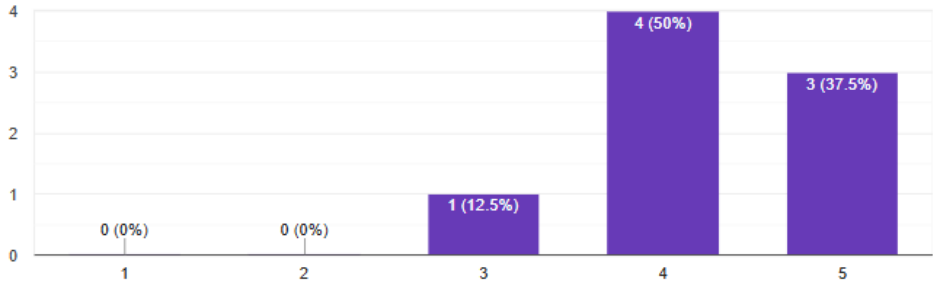
8 responses




Informasi yang diberikan tidak bertentangan dengan pengetahuan gizi yang berlaku umum.

Copy chart

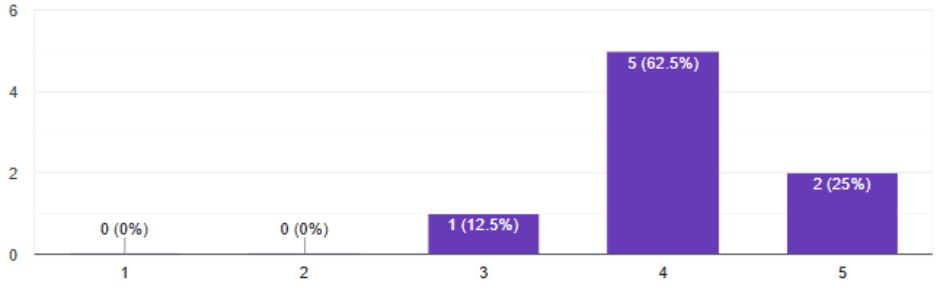
8 responses




Informasi yang diberikan akurat dalam menjelaskan dampak makanan terhadap kadar gula darah.

Copy chart

8 responses

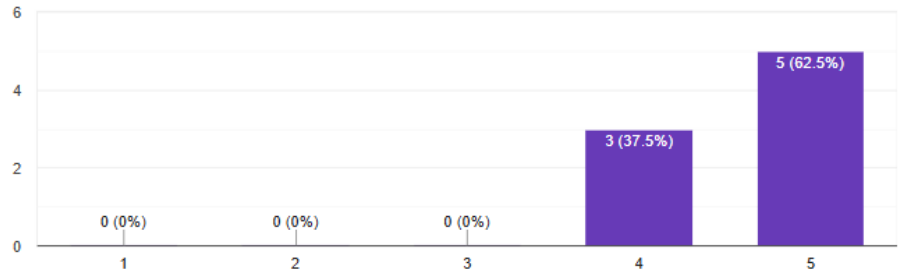


**D. Manfaat untuk Pengguna**

Informasi yang diberikan membantu pengguna membuat keputusan diet yang lebih sehat.

Copy chart

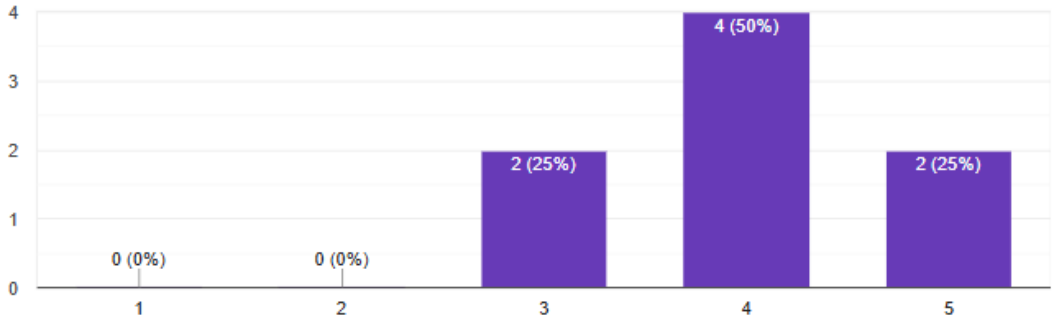
8 responses




Saran aktivitas/olahraga yang diberikan relevan untuk mengurangi kadar gula darah.

Copy chart

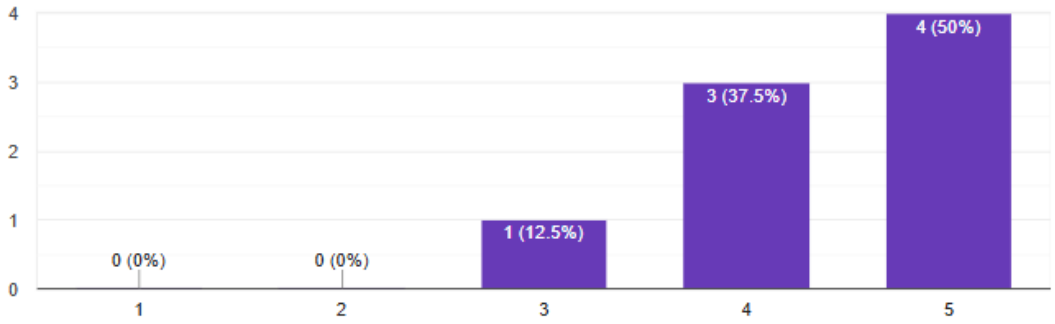
8 responses




Informasi yang diberikan mendukung pencegahan dan manajemen diabetes.

Copy chart

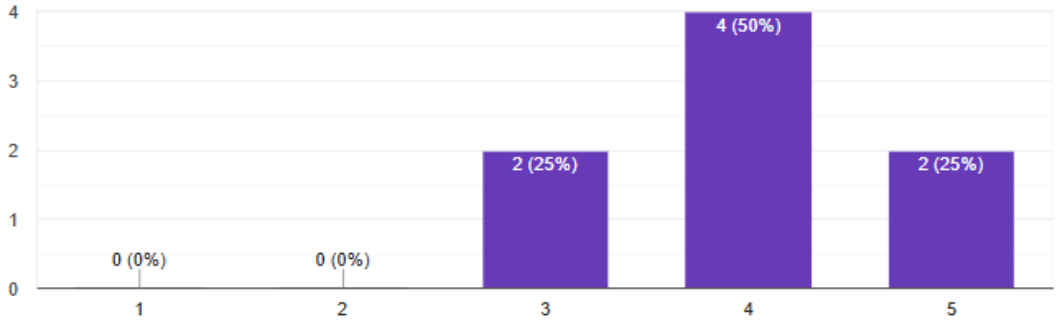
8 responses



Informasi yang dihasilkan sudah sesuai dengan standar tata laksana ahli gizi atau rekomendasi profesi terkait dalam pencegahan dan manajemen diabetes.

Copy chart

8 responses

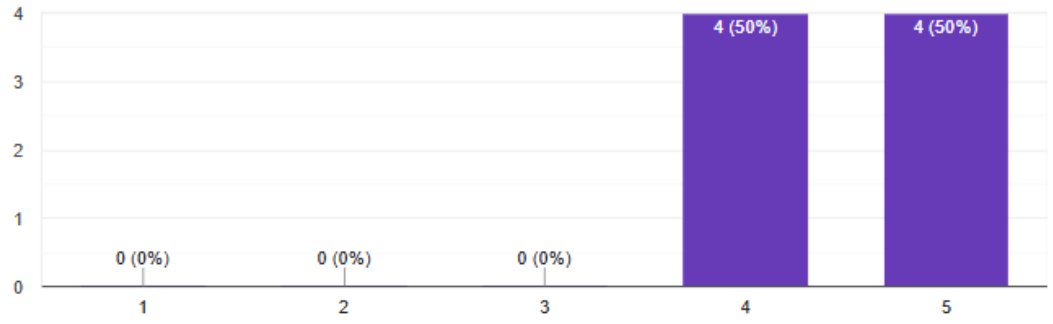



**E. Umpan Balik Tambahan**

Secara keseluruhan, saya puas dengan kualitas informasi yang dihasilkan oleh API Gemini.

Copy chart

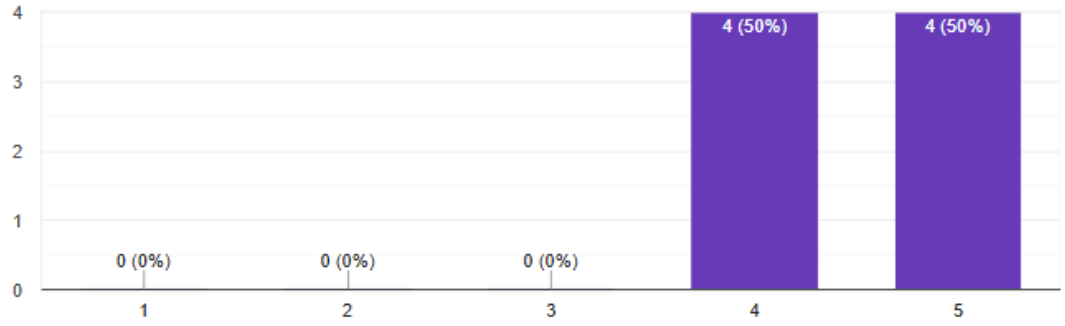
8 responses



Saya merekomendasikan penggunaan API Gemini dalam aplikasi kesehatan seperti GlucoFit.

Copy chart

8 responses




**Bagian Komentar Terbuka**

Apakah ada aspek tertentu dari informasi yang dihasilkan oleh API Gemini yang perlu diperbaiki? Jika ya, jelaskan.

8 responses

Tidak

Cantumkan URT gula 1 sdm dan kandungan semua dlm makanan

Tidak ada

Tidak, aplikasinya sudah sangat informatif dan edukatif.

Perlu mencantumkan referensi yang digunakan untuk menghasilkan informasi

Iya, gula terdiri dari banyak jenis. Glukosa, sukrosa, laktosa, fruktosa. Untuk beberapa jenis gula di metabolisme di hepar sehingga jarang mempengaruhi gula dalam darah yg merupakan pilihan aman untuk pengguna diabetes melitus. Untuk saat ini informasi yg diberikan hanya gula secara umum. Sehingga mungkin untuk jenis gula dalam makanan tertentu bila memungkinkan diberi informasi apakah gula tsb termasuk gula yg aman untuk pasien DM (seperti pada tropicana) atau yg sebaiknya memang dilakukan pembatasan harian dalam konsumsinya

Pengurangan jumlah tulisan

Apakah ada informasi tambahan yang seharusnya disertakan oleh API Gemini untuk mendukung pengguna dalam memantau asupan gula mereka? Jika ada, jelaskan.

8 responses

Tidak

Sebutkan jenis keturunan gula. Glukosa frukotosa brpa%

Cukup baik

Berat satuan penukar dirumah tangga seperti sdm,sdt, dll yang direkomendasikan untuk kadar gula yang aman dalam satu kali asupan.

Jenis gula, misalnya glukosa, sukrosa, atau fruktosa

Saya sarankan ada informasi gerakan masyarakat yang di keluarkan oleh kemenkes yang bisa juga digunakan untuk memperbaiki pola hidup pasien diabetes melitus. Disana juga tertera olahraga yang bagaimana yg tepat untuk pasien DM, dan ada warning terkait gejala2 komplikasi yg sebaiknya di waspadai oleh pasien² DM

Informasi tambahan kebutuhan kalori, jumlah kalori dan gizi lainnya (protein dll)

Saran lain untuk meningkatkan kualitas informasi yang dihasilkan oleh API Gemini:

8 responses

Tidak ada sudah baik

Tidak

Tingkatkan lagi

Lanjutkan

Tidak ada. Sukses selalu ya.

Membatasi sumber dari referensi tertentu agar informasi lebih relevan, misalnya menggunakan referensi dari kemenkes untuk rekomendasi olahraga atau langkah prevensi diabetes yang lain

Karena pasien diabetes melitus biasanya bekisaran usia 40-70 tahun (Lansia) . Tolong font yg dibuat agak besar agar bisa terbaca oleh mata² lansia, bahasa dan aplikasi sebaiknya dibuat se simple mungkin agar bisa di gunakan oleh lansia yg biasanya kurang pandai dalam penggunaan aplikasi. Ilustrasinya sudah bagus tp dgn gambar biasanya lebih bisa di tangkap oleh pasien dripada sekedar kalimat² (banyak sumber di yutub yg punya ilustrasi perjalanan penyakit DM) sehingga menarik juga untuk di tonton.

4. Hasil Pengujian Model CNN dengan metrik AUC

```
Macro-average AUC (valid classes only): 0.9297

Top Performing Classes:
                        AUC Score
strawberry_shortcake     1.000000
red_velvet_cake          1.000000
miso_soup                0.999877
spaghetti_carbonara      0.999267
creme_brulee             0.999170

Classes with insufficient data:
                 AUC Score
club_sandwich          NaN
```