

Aplikasi Pencatatan Kalori Harian Berbasis Android Dengan Arsitektur MVVM

Alfi Zia Ulhaq^{1*}, Abilawa Zulfiqar Adilukito², Sultan Muhamad Pascal Gadja Neru³, Muhammad Daffa Agisfio⁴

^{1,2,3,4}Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika
Jl. Kramat Raya No.98, Kwitang, Senen, Jakarta Pusat, 10450, Indonesia

e-mail: ¹alfiziaulhaq@gmail.com, ²abilawa0103@gmail.com, ³sultanmpgn@gmail.com,
⁴daffaforemail@gmail.com

(*) Corresponding Author

Artikel Info : Diterima : 24-05-2024 | Direvisi : 23-01-2025 | Disetujui : 30-01-2025

Abstrak - Ketidakseimbangan antara asupan dan pengeluaran kalori dianggap sebagai penyebab utama obesitas atau kelebihan berat badan. Maka dengan mengontrol asupan kalori yang seimbang dan sesuai dengan kebutuhannya akan dapat mencegah obesitas. Aplikasi pencatatan kalori harian dapat membantu seseorang dalam mencatat, mengontrol dan mendapat informasi asupan kalorinya. Penelitian ini membahas pengembangan aplikasi serupa yang diberi nama NutriZen dengan menggunakan metode *waterfall*, selama proses pengembangannya oleh tim CH2-PS076 dalam program Bangkit 2023 *batch 2*. Aplikasi yang dibuat merupakan aplikasi berbasis Android yang dibuat dengan bahasa pemrograman Kotlin dan arsitektur MVVM sebagai *design pattern* yang mudah dipelajari dan menjadikan kode mudah untuk dipahami dan dikelola. Pengujian aplikasi ini menggunakan metode *usability testing* dengan alat bantu *System Usability Scale* untuk mengetahui tingkat penerimaan *usability* pengguna. Hasil yang didapat termasuk dalam tingkat penerimaan *usability* marginal, sehingga diperlukan perbaikan agar aplikasi ini dapat lebih diterima dan diandalkan oleh masyarakat luas.

Kata Kunci : Obesitas, Kalori, MVVM, *Waterfall*, SUS

Abstracts - An imbalance between calorie intake and expenditure is considered the main cause of obesity or being overweight. So by controlling your calorie intake in a balanced manner and according to your needs you will be able to prevent obesity. The daily calorie recording application can help someone record, control and obtain information on their calorie intake. This article discusses the development of this type of application, named NutriZen, using the waterfall method, during the development process by the CH2-PS076 team in the Bangkit 2023 batch 2 program. The application created is an Android-based application created with the Kotlin programming language and MVVM architecture as a design pattern that is easy to learn and makes the code easy to understand and manage. Testing this application uses the usability test method with the System Usability Scale tool to determine the level of user acceptance of usability. The results obtained include a marginal level of usability acceptance, so improvements are needed so that this application can be more accepted and relied on by the wider community.

Keywords : Obesity, Calories, MVVM, *Waterfall*, SUS

PENDAHULUAN

Makanan cepat saji menjadi favorit banyak kalangan. Tetapi makanan itu cenderung tidak sehat dan mempunyai kalori yang besar. Akibatnya, banyak orang tidak dapat memperhatikan gizi dari apa yang dimakannya sehingga mengakibatkan pola makan yang tidak sehat. Pola makan yang tidak sehat merupakan faktor risiko utama berbagai penyakit yang berhubungan dengan kesehatan. Menurut Organisasi Kesehatan Dunia (WHO), 1 dari 3 orang dewasa di seluruh dunia memiliki pola makan yang tidak sehat. Akibatnya pada tahun 2016, 39% orang dewasa berusia 18 tahun ke atas mengalami kelebihan berat badan dan 13% mengalami obesitas. Sebagian besar penduduk yang kelebihan berat badan dan obesitas terdapat lebih banyak kematian orang dibandingkan yang kekurangan berat badan (WHO, 2024).

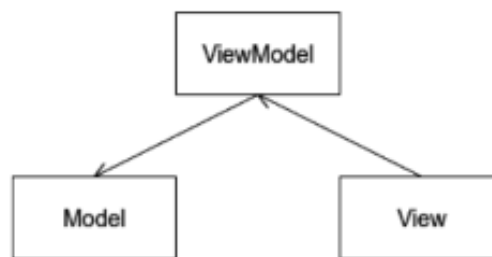
Ketika penyebab utama obesitas dianggap sebagai ketidakseimbangan antara asupan dan pengeluaran



kalori, maka penurunan asupan kalori diperkirakan akan langsung meringankan ketidakseimbangan energi dan membalikkan obesitas (Lee et al., 2021). Oleh karena itu dengan menghitung jumlah kebutuhan kalori dan asupan kalori harian merupakan langkah yang efektif untuk mencegah dan mengobati obesitas. Selain itu penting juga untuk mempertimbangkan pola makan sehat sebagai metode penurunan berat badan dan peningkatan kesehatan yang lebih dari sekadar penghitungan kalori (Bennett, 2022).

Karena itu peneliti bersama tim CH2-PS076 melalui program Bangkit 2023 *batch 2* mengembangkan aplikasi pencatatan kalori harian yang bernama NutriZen yang bertujuan untuk mengatasi masalah pola makan dan nutrisi dengan memberikan informasi kalori berdasarkan foto makanan pengguna atau memasukkannya secara manual. Pengembangan aplikasi ini terinspirasi dari aplikasi Fatsecret yang sudah populer untuk *capstone project*. Peneliti membuat aplikasi ini secara keseluruhan dari sisi *Front-End* Android. Aplikasi ini memungkinkan pengguna untuk mencatat konsumsi kalori harian dan memilih dari tiga program diet, yaitu menambah berat badan, menurunkan berat badan, dan menjaga berat badan.

Aplikasi ini dibuat dengan bahasa pemrograman Kotlin untuk Android Native dan menggunakan pola arsitektur *Model-View-ViewModel* (MVVM). Bahasa Kotlin dipilih karena memiliki sintaks yang lebih ringkas dan lebih aman dibanding Java dengan fitur *null safety* serta direkomendasikan secara resmi oleh Google. Selain MVVM ada juga arsitektur *Model-View-Presenter* (MVP) dan *Model-View-Controller* (MVC) yang biasa digunakan sebagai *design pattern* dalam pengembangan aplikasi. Meskipun MVC dan MVP mungkin lebih mudah untuk diasimilasi dan diimplementasikan, keunggulan MVVM dibandingkan MVC dan MVP adalah waktu yang diperlukan untuk mempelajari pola arsitektur MVVM lebih cepat (Arponen, 2023). Selain itu MVVM dan MVP juga lebih baik dibandingkan dengan MVC pada metrik *memory usage*, *execution time*, dan *coupling level* yang lebih rendah (Tetiana et al., 2023).



Sumber: Dumbravan (2022)

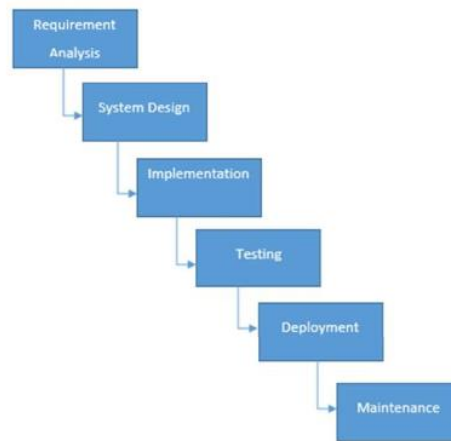
Gambar 1. Interaksi komponen MVVM

Dalam MVVM, View merepresentasikan *activities* dan *fragments*, Model sebagai manajemen data dan ViewModel adalah perantara keduanya ketika permintaan data terjadi dari Model saat View membutuhkannya (Dumbravan, 2022). Hal ini bertujuan agar kode pada aplikasi menjadi terorganisir, mudah dibaca, dipahami, dan dirawat (Fajri & Rani, 2022).

Aplikasi pencatatan kalori harian ini diharapkan menjadi perangkat lunak yang memberikan manfaat kepada Masyarakat luas yang peduli terhadap kesehatan dan membantu mengurangi risiko terhadap obesitas dengan menjaga asupan kebutuhan kalori. Selain itu NutriZen juga bisa dipakai oleh banyak kalangan muda yang mengikuti kegiatan *Fitness* atau *Gym* untuk membantu program diet mereka.

METODE PENELITIAN

Metode yang digunakan dalam pengembangan aplikasi pencatatan kalori harian dalam penelitian ini adalah metode *waterfall*. Para pengembang harus menyelesaikan sebuah fase dalam metodologi ini untuk menyelesaikan langkah berikutnya (Saeed et al., 2019). Diperkenalkan oleh Winston Royce pada tahun 1970. Ini adalah salah satu Model *Systems Development Life Cycle* (SDLC) yang tertua, namun tidak banyak digunakan dalam beberapa tahun terakhir. Pendekatan *waterfall* tidak memungkinkan proses untuk kembali ke fase sebelumnya dan memungkinkan adanya perubahan di dalamnya. Model ini digunakan untuk proyek-proyek kecil, karena hanya ada sedikit ruang untuk revisi setelah suatu tahapan selesai (Gurung et al., 2020). Fase-fase dalam model *waterfall* dapat meliputi fase-fase seperti *requirement analysis*, *system design*, *implementation*, *testing*, *deployment*, dan *maintenance* (Pargaonkar, 2023).



Sumber: Gurung et al. (2020)

Gambar 2. Fase pada metode *waterfall*

Dalam penelitian ini hanya sebagian fase dari metode *waterfall* yang diterapkan karena artikel ini berfokus pada pengembangan aplikasi pencatatan kalori harian dari sisi *Front-End* Android serta integrasinya dengan *endpoint* API. Metode ini dipilih karena lebih mudah diimplementasikan untuk proyek skala kecil dengan kebutuhan yang sudah terdefinisi dengan jelas sejak awal.

1. Analisa kebutuhan

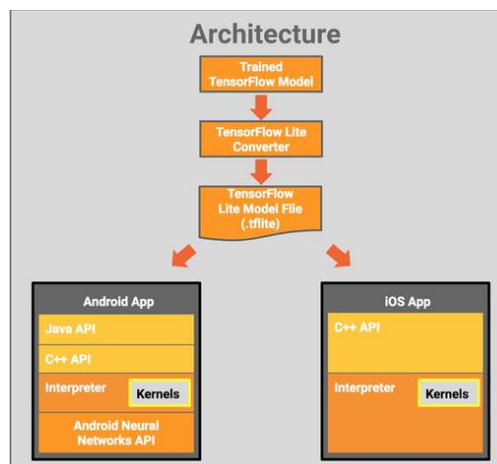
Pada fase ini menjelaskan proses pengumpulan kebutuhan *software* sehingga mudah dipahami dan disesuaikan dengan kebutuhan pengguna (Indah, 2022). Analisa fungsional akan dirancang dengan membuat ruang lingkup yang memuat fitur minimum aplikasi.

2. Desain sistem

Dalam merancang desain suatu sistem akan di buat *class diagram* dan *application flowchart*. Suatu *class diagram* menunjukkan struktur statis dari kelas-kelas inti yang membangun sistem (Riza et al., 2020). Dan *application flowchart* digunakan untuk menunjukkan alur penggunaan aplikasi. Fungsi *flowchart* adalah untuk mengetahui bagaimana proses aliran data dan menerjemahkan proses berjalannya sebuah program sehingga akan mempermudah tahapan pembuatan sistem serta agar lebih mudah untuk dipahami (Dwi Saputro & Frans Kuddi, 2022).

3. Implementasi

Implementasi pengkodean menggunakan bahasa pemrograman Kotlin untuk Android Native dan menggunakan arsitektur MVVM sebagai *design pattern*. Salah satu fitur unggulan yang dihadirkan pada aplikasi ini adalah fitur deteksi kalori makanan yang menggunakan *machine learning* untuk klasifikasi gambar. Untuk itu implementasi model *machine learning* menggunakan TensorFlow Lite (TFLite) yang di *deploy* pada perangkat *mobile* agar menghindari *latency response* klasifikasi gambar jika model di *deploy* ke dalam *cloud*. TFLite merupakan *framework deeplearning opensource* untuk inferensi *on-device* yang dapat dikompresi dan di *deploy* ke dalam aplikasi *mobile* atau sejenisnya (Taufiq et al., 2021).



Sumber: Jasrotia (2020)

Gambar 3. Arsitektur TFLite *model deployment* pada perangkat *mobile* Android dan IOS

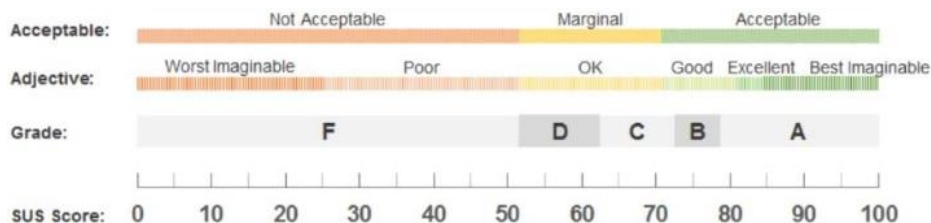
Keunggulannya adalah mampu dieksekusi lebih cepat pada perangkat *mobile* atau perangkat tertanam dan menyediakan *model maker* yang dapat dieksekusi oleh Tensorflow Lite Interpreter (Permana & Budayawan, 2020). Tetapi untuk memperolehnya harus dibuat terlebih dahulu model Tensorflow yang telah dilatih kemudian di konversi menjadi format .tflite (Jasrotia, 2020).

4. Pengujian

Jenis pengujian yang dilakukan adalah *usability testing* dengan alat bantu *System Usability Scale* (SUS). Metode ini merupakan alat untuk *usability testing* yang serbaguna dan sangat kuat. Metode *testing* ini terdiri dari 10 pertanyaan yang akan dijawab pada rentang 5 poin mulai dari sangat tidak setuju hingga sangat setuju (Guralnick et al., 2023). Kemudian untuk mendapatkan nilai skor rata-rata terdapat aturan sebagai berikut:

- a. Terdapat 10 pertanyaan sistematis yang diberikan kepada responden dengan detail pertanyaan sebagai berikut:
 - 1) Saya berpikir akan sering menggunakan sistem ini.
 - 2) Menurut saya sistem ini rumit untuk digunakan
 - 3) Menurut saya sistem ini mudah untuk digunakan
 - 4) Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
 - 5) Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
 - 6) Saya merasa akan ada banyak hal yang tidak konsisten dalam sistem ini
 - 7) Menurut saya orang lain akan memahami cara menggunakan sistem ini dengan cepat
 - 8) Menurut saya sistem ini membingungkan
 - 9) Saya merasa sangat percaya diri dalam menggunakan sistem ini
 - 10) Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini
- b. Setiap pertanyaan bernomor ganjil, yaitu 1, 3, 5, 7, dan 9. Skor setiap pertanyaan tersebut didapat dari poin pertanyaan yang di jawab dikurangi satu.
- c. Setiap pertanyaan bernomor genap, yaitu 2, 4, 6, 8, dan 10. Skor setiap pertanyaan tersebut didapat dari lima poin dikurangi poin pertanyaan yang dijawab.
- d. Hasil skor dari seluruh pertanyaan akan dijumlahkan kemudian dikali 2,5. Maka didapat skor untuk satu responden.
- e. Untuk mendapatkan rata-rata nilai skor SUS. Maka nilai skor seluruh responden dijumlahkan kemudian dibagi banyaknya responden (Saputra, 2019).

Setelah itu rentang hasil skor SUS yang didapat untuk penelitian ini dapat diterjemahkan menggunakan *adjective scale* dan *grade scale* (Muqoddas et al., 2020) dengan rentang skor seperti pada gambar 4.



Sumber: (Muqoddas et al., 2020)

Gambar 4. *SUS score range*

HASIL DAN PEMBAHASAN

1. Analisa kebutuhan

Kebutuhan pengguna yang paling utama adalah pengguna dapat mengetahui berapa jumlah kebutuhan kalori hariannya dan dapat menghitung masukan kalori setiap kali dia makan serta mendapat peringatan berapa sisa jatah kalori yang harus dipenuhi. Sehingga aplikasi ini di rancang memiliki fitur minimum, sebagai berikut:

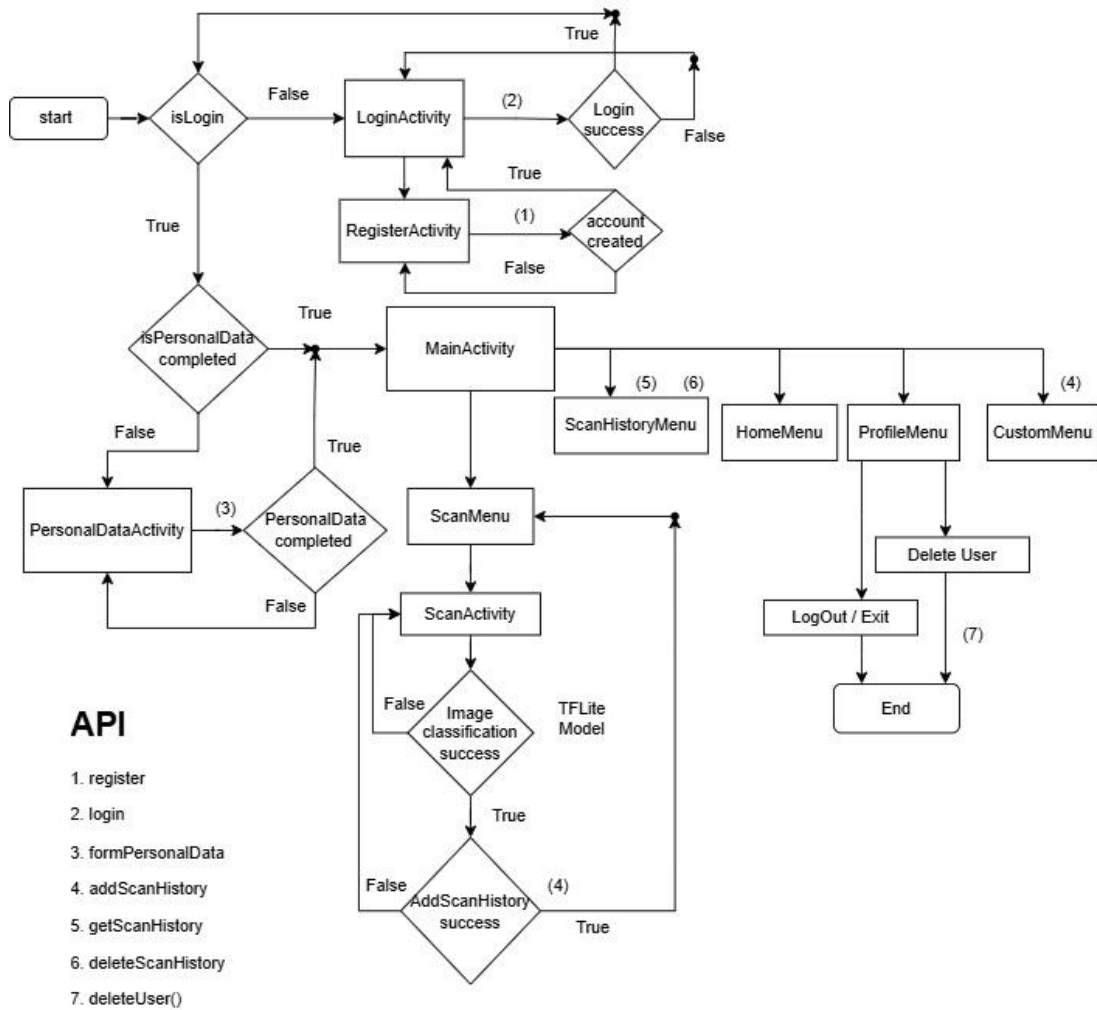
- a. Tersedia 3 program diet untuk personalisasi pengguna, yaitu:
 - 1) *Gain weight*, bagi pengguna yang ingin menambah berat badan
 - 2) *Lose weight*, bagi pengguna yang ingin menurunkan berat badan
 - 3) *Maintain weight*, bagi pengguna yang ingin mempertahankan berat badan
- b. Terdapat *dashboard* yang memberikan informasi target kalori yang dipersonalisasi berdasarkan profil pengguna, dan persentase konsumsi kalori harian
- c. Pengguna dapat mengubah data profil atau personalisasi akun.
- d. Pengguna bisa melacak asupan kalori dan nutrisi dari waktu ke waktu melalui riwayat buku harian makanan (*Food diary history*)

- e. Fitur pindai makanan memungkinkan pengguna mengambil gambar makanan dan menerima perkiraan kandungan kalori dan nutrisi makanan mereka dengan jumlah jenis makanan yang terbatas, kemudian menambahkannya sebagai *item* konsumsi makanan ke *Food diary history*
- f. Pengguna juga dapat memasukkan item konsumsi makanan secara manual ke *Food diary history* dan dapat mengoreksi salah *input* dalam *Food diary history* dengan menyediakan fitur *delete item* konsumsi makanan

2. Desain sistem

a. *Flowchart*

Untuk mempermudah pengembangan dan memahami alur pemakaian aplikasi maka dibuatlah *flowchart*. *Flowchart* merupakan gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut (Zalukhu et al., 2023).

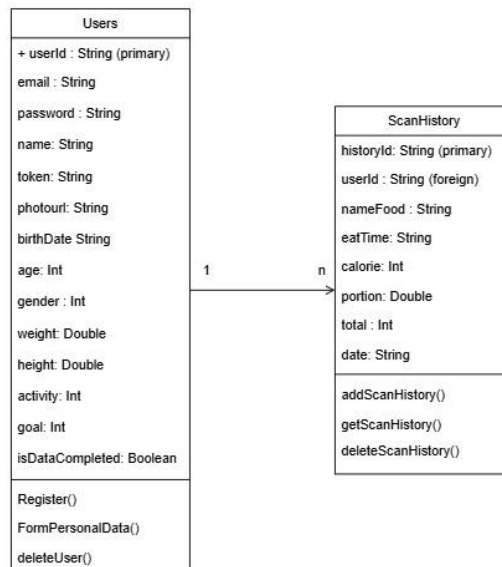


Sumber: Hasil Penelitian (2024)

Gambar 5. *Flowchart* aplikasi

b. *Class Diagram*

Class Diagram digunakan untuk mempermudah perancangan dan konfigurasi pada *database* yang menyajikan hubungan antar entitas yang disertai metode didalamnya. Pada aplikasi ini *database* memiliki entitas *users* untuk fitur otentikasi, otorisasi dan personalisasi serta entitas *ScanHistory* yang di gunakan dalam pencatatan riwayat konsumsi kalori harian.

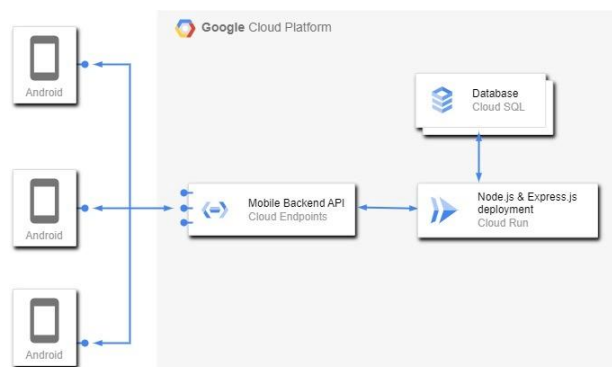


Sumber: Hasil Penelitian (2024)

Gambar 6. Class diagram aplikasi

c. Cloud Architecture Diagram

Data akun pengguna akan disimpan di *cloud* agar dapat disimpan dengan rapi dan bisa diakses dari mana pun, jika pengguna mengganti perangkat seluler mereka. *Back-End* diimplementasikan dengan *deployment* Node.js dan Express.js ke Google Cloud Platform. Dan untuk mengakses *endpoint* API akan menggunakan *library* Retrofit Android.



Sumber: Hasil Penelitian (2024)

Gambar 7. Cloud Architecture Diagram untuk back-end aplikasi

3. Implementasi

Aplikasi berhasil dibuat dengan bahasa Kotlin dan MVVM *design pattern* serta memenuhi kebutuhan fitur minimum yang telah dibuat dalam tahap analisis kebutuhan. Menggunakan *library* dan *framework* seperti Jetpack compose, Retrofit, Navigation, Kotlin Coroutines, LiveData, dan Coil. Serta *Machine Learning Model Deployment* dengan TFLite. Hasil dari implementasi dirangkum ke dalam poin-poin berikut:

- a. Pengguna masuk ke halaman *dashboard* yang menampilkan skor personal BMI (*Body Mass Index*) dan ringkasan target kalori harian dengan BMR (*Basal Metabolic Rate*) yang dihitung dengan formula Mifflin-St Jeor, berdasarkan data personalisasi pengguna saat mengisi formulir personalisasi setelah pendaftaran akun seperti pada gambar 10. Adapun persamaan BMI, BMR untuk laki-laki dan BMR untuk perempuan dijabarkan dalam persamaan (1), (2) dan (3) (Abreu, 2023).

$$BMI = \frac{w}{H^2} \tag{1}$$

$$BMR \text{ (Untuk laki-laki)} = (10 \times w) + (6,25 \times h) - (5 \times a) + 5 \tag{2}$$

$$BMR \text{ (Untuk perempuan)} = (10 \times w) + (6,25 \times h) - (5 \times a) - 161 \tag{3}$$

Keterangan:

w = Berat badan (kg)

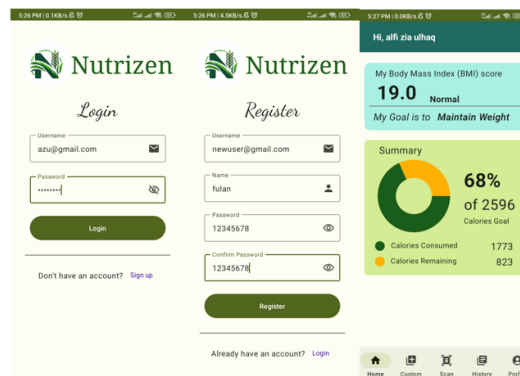
H = Tinggi badan (m)

h = Tinggi badan (cm)

a = Umur dalam tahun

y = Konstanta faktor aktivitas, dengan nilai berdasarkan kategori berikut:

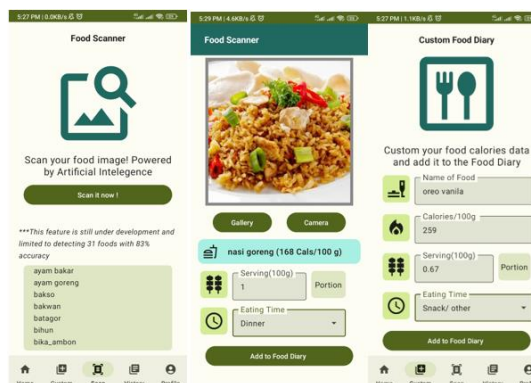
- 1) *Sedentary* = 1,2 (sedikit berolahraga)
- 2) *Light Activity* = 1.375 (berolahraga 1 sampai 3 hari setiap minggu)
- 3) *Moderate Activity* = 1,55 (berolahraga 3 sampai 5 hari setiap minggu)
- 4) *Very Active* = 1.725 (berolahraga 6 sampai 7 hari setiap minggu)
- 5) *Extra Active* = 1,9 (berolahraga 2 kali sesi setiap hari)



Sumber: Hasil Penelitian (2024)

Gambar 8. Screenshots halaman login, register dan dashboard

- b. Pemindai makanan untuk menilai kalori makanan dengan model pembelajaran mesin TFLite dapat memindai gambar dari galeri atau kamera. Setelah model menghasilkan *output* label jenis makanan tertentu kemudian hasil kalori akan keluar dan dapat ditambahkan ke riwayat buku harian makanan. Alat ini terbatas dalam mendeteksi 31 jenis makanan dengan akurasi 83%. Tersedia juga formulir *input* kalori manual untuk menambahkan *item* ke buku harian makanan.



Sumber: Hasil Penelitian (2024)

Gambar 9. Screenshots fitur pemindai kalori makanan dan *input* kalori manual

- c. Riwayat buku harian makanan dengan ringkasan dapat di cari berdasarkan tanggal dengan *item* yang dapat di hapus. Dan terdapat formulir data pribadi pada menu profil untuk data personalisasi akun yang dapat diperbaharui sehingga informasi pada *dashboard* juga diperbaharui secara otomatis.



Sumber: Hasil Penelitian (2024)

Gambar 10. Screenshots riwayat buku harian makanan, menu profil dan formulir data personalisasi

4. Pengujian

Dengan menggunakan alat pengujian *System Usability Scale* (SUS) untuk metode *usability testing* telah dilakukan survei terhadap 10 responden. Survei ini mengukur tingkat kepuasan atau penerimaan user dalam menggunakan aplikasi. Karena Secara umum metode *usability testing* menggunakan kuesioner atau daftar pertanyaan yang diberikan kepada responden untuk melakukan perhitungan (Saputra, 2019). Hasil survei dari para responden kemudian dihitung berdasarkan aturan SUS untuk mencari nilai rata-ratanya.

Tabel 1. Hasil survei SUS aplikasi pencatatan kalori harian, NutriZen

No	Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Skor
1	R1	4	2	5	1	4	3	4	2	4	3	75
2	R2	4	2	4	2	5	2	4	2	4	3	75
3	R3	4	1	4	1	3	4	5	1	5	4	75
4	R4	4	2	5	1	5	1	5	1	1	3	80
5	R5	3	3	3	3	3	3	3	3	3	3	50
6	R6	4	4	5	2	4	1	1	5	3	5	50
7	R7	5	1	5	4	5	1	5	1	5	2	90
8	R8	3	3	3	3	3	3	3	3	3	3	50
9	R9	4	3	4	1	4	3	4	2	3	5	62,5
10	R10	3	3	3	2	4	3	3	2	3	3	57,5
Rata-rata skor											66,5	

Sumber: Hasil Penelitian (2024)

Berdasarkan hasil survei yang dilakukan sebagaimana pada tabel 1. Hasil rata-rata skor SUS yang didapat adalah 66,5. Jika diterjemahkan dengan *adjective scale*, skornya masuk dalam *range* OK dan Good. Dan jika menggunakan *grade scale*, maka skornya mendapat nilai C. Dengan nilai skor yang didapat, maka tingkat penerimaan (*acceptability*) *usability testing* masih tergolong marginal.

KESIMPULAN

Aplikasi pencatatan kalori harian, NutriZen dikembangkan menggunakan bahasa pemrograman Kotlin untuk Android Native dengan arsitektur *Model-View-ViewModel* (MVVM). Metode *waterfall* dipilih sebagai *Systems Development Life Cycle* (SDLC) karena proyek aplikasi ini masih tergolong skala kecil. Semua fitur minimum yang dirancang dalam analisa kebutuhan berhasil di buat dengan keunggulan MVVM sebagai *design pattern* yang mudah dipelajari, diorganisir, dibaca, dipahami dan dikelola. Setelah aplikasi melewati pengujian *usability testing* dengan alat uji *System Usability Scale* diperoleh hasil tingkat penerimaan yang masih tergolong marginal. Hal ini dapat menjadi perbaikan bagi peneliti dan pengembang selanjutnya agar dapat merancang desain aplikasi yang lebih ramah pengguna dengan desain *user experience* (UX) yang tidak rumit dan *user interface* (UI) yang lebih menarik. Harapannya aplikasi ini dapat terus dikembangkan dan lebih diterima masyarakat luas dalam membantu mencegah obesitas, dan menjalankan program diet dan gaya hidup sehat.

REFERENSI

- Abreu, M. (2023, May 3). *Mifflin-St. Jeor for nutrition professionals* - Nutrium Blog. <https://nutrium.com/blog/mifflin-st-jeor-for-nutrition-professionals/>
- Arponen, O. (2023). *Software architectural patterns and principles in Android development* [Metropolia University]. <http://www.theseus.fi/handle/10024/791289>
- Bennett, L. (2022). Relation between Calorie Intake and Obesity in Human Body. *Journal of Nutritional Disorders & Therapy*, 12(7), 1–2. <https://www.walshmedicalmedia.com/open-access/relation-between-calorie-intake-and-obesity-in-human-body-113456.html>
- Dumbravan, A. (2022). *Clean Android Architecture : Take a Layered Approach to Writing Clean, Testable, and Decoupled Android Applications*. Packt Publishing. https://www.google.co.id/books/edition/Clean_Android_Architecture/4QtxEAAAQBAJ?hl=en&gbpv=1
- Dwi Saputro, A., & Frans Kuddi, B. (2022). Pembuatan Aplikasi Try Out Cat (Computer Assisted Test) Penerimaan Pegawai Negeri Sipil Bidang Tes Intelegensi Umum Berbasis Desktop. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 6. <https://tunasbangsa.ac.id/ejurnal/index.php/jsakti/article/view/460>
- Fajri, A. R., & Rani, S. (2022). Penerapan Design Pattern MVVM dan Clean Architecture pada Pengembangan Aplikasi Android (Studi Kasus: Aplikasi Agree Partner). *AUTOMATA*, 3(2). <https://journal.uui.ac.id/AUTOMATA/article/view/24041>
- Guralnick, D., Auer, M. E., & Poce, A. (2023). *Creative approaches to technology-enhanced learning for the workplace and higher education: proceedings of 'The Learning Ideas Conference' 2023*. Springer. https://www.google.co.id/books/edition/Creative_Approaches_to_Technology_Enhanc/a9LYEAAAQBAJ?gbpv=1
- Gurung, G., Shah, R., & Jaiswal, D. P. (2020). Software Development Life Cycle Models-A Comparative Study. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 30–37. <https://doi.org/10.32628/CSEIT206410>
- Indah, N. N. (2022). *Analisis Dan Perancangan Sistem Informasi Pengelolaan Arus Kas Pada PT Graha Sentramulya*. Jurnal Ilmu Data. <http://ilmudata.org/index.php/ilmudata/article/view/78>
- Jasrotia, N. (2020). *Deploying Tenserflow Lite Model On Android*. <https://medium.com/bobble-engineering/deploying-tenserflow-lite-model-on-android-de4ac3c187b2>
- Lee, C. L., Liu, W. J., & Wang, J. S. (2021). Associations of low-carbohydrate and low-fat intakes with all-cause mortality in subjects with prediabetes with and without insulin resistance. *Clinical Nutrition (Edinburgh, Scotland)*, 40(5), 3601–3607. <https://doi.org/10.1016/J.CLNU.2020.12.019>
- Muqoddas, A., Yogananti, A. F., & Bastian, H. (2020). Usability User Interface Desain pada Aplikasi Ecommerce (Studi Komparasi Terhadap Pengalaman Pengguna Shopee, Lazada, dan Tokopedia). *ANDHARUPA: Jurnal Desain Komunikasi Visual & Multimedia*, 6(01). <https://doi.org/10.33633/ANDHARUPA.V6I1.3194>
- Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. *International Journal of Scientific and Research Publications*, 13(8), 120–124. <https://doi.org/10.29322/IJSRP.13.08.2023.P14015>
- Permana, A., & Budayawan, K. (2020). Aplikasi Android Pengklasifikasi Semantik Teks Menggunakan Tensorflow Lite Pada Ringkasan Karya Ilmiah. *Voteteknika (Vocational Teknik Elektronika Dan Informatika)*, 8(4), 128–135. <https://doi.org/10.24036/VOTETEKNIKA.V8I4.110349>
- Riza, A., Anshori, M. A. I., Arrazy, F., & Yaqin, M. A. (2020). Pengukuran Metrik Kompleksitas Web Service Sekolah. *Jurasik (Jurnal Riset Sistem Informasi Dan Teknik Informatika)*, 5(1), 147–153. <https://doi.org/10.30645/JURASIK.V5I1.179>
- Saeed, S., Jhanjhi, N. Z., Naqvi, M., & Humayun, M. (2019). Analysis of software development methodologies. *International Journal of Computing and Digital Systems*, 8(5). <https://doi.org/10.12785/IJCD/080502>
- Saputra, A. (2019). Penerapan Usability pada Aplikasi PENTAS Dengan Menggunakan Metode System Usability Scale (SUS). *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 1(3), 206–212. <https://doi.org/10.35746/JTIM.V1I3.50>
- Taufiq, A., Pratama, M., & Pratama, A. R. (2021). Rancang Bangun Aplikasi Android “Kuliah Apa?” Berbasis Flutter dan TensorFlow Lite. *AUTOMATA*, 2(1). <https://journal.uui.ac.id/AUTOMATA/article/view/17293>
- Tetiana, V., Kusumo, D. S., & Andrian, M. (2023). *Analisis Pengaruh Pola Arsitektur Model View View Model (MVVM) terhadap Kinerja Aplikasi Mobile dengan Menerapkan Application Programming Interface (API) Covid 19*. E-Proceeding of Engineering. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/20608>
- WHO. (2024, March 1). *Obesity and overweight*. WHO. <https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight>
- Zalukhu, A., Purba, S., & Darma, D. (2023). Perangkat Lunak Aplikasi Pembelajaran Flowchart. *Jurnal Teknologi, Informasi Dan Industri*, 4(1), 61–70. <https://ejurnal.istp.ac.id/index.php/jtii/article/view/351>