

POLA DESAIN MODEL-VIEW-VIEW MODEL UNTUK PENGEMBANGAN APLIKASI PENGHITUNG KALORI YANG EFISIEN

Daniel Alexander Hijaya¹, Mohammad Fajar^{2*}, Hasniati³

^{1,2,3} Program Studi Teknik Informatika, STMIK Kharisma Makassar
e-mail: ¹danielalexander_17@kharisma.ac.id, ²fajar@kharisma.ac.id,
³hasniati@kharisma.ac.id

Abstrak

Penelitian ini bertujuan untuk mengimplementasikan pola arsitektur Model-View-ViewModel dalam perancangan aplikasi penghitung kalori (Calorie Counter) yang efisien. Penggunaan pola rancangan Model-View-ViewModel didasarkan pada usaha untuk memudahkan proses pemakaian kembali kode sumber (Reusability). Proses perancangan menggunakan flutter sebagai framework aplikasi. Penelitian ini diharapkan menjadi referensi mengenai Model-View-ViewModel, serta memberikan pengetahuan bagi peneliti mengenai kombinasi antara pola desain Model-View-ViewModel dan framework flutter pada platform android. Pengumpulan data dilakukan melalui studi literatur dan evaluasi terhadap rancangan sistem. Aplikasi dikembangkan menggunakan bahasa pemrograman Dart, sementara evaluasi penerapan Model-View-ViewModel terkait aspek reusability diukur menggunakan CK Metrics. Evaluasi menunjukkan rancangan aplikasi penghitung kalori yang menggunakan pola desain MVVM dapat dijalankan di atas platform Android menggunakan Bluestacks Android Emulator. Selain itu, derajat reusability rancangan aplikasi lebih baik dari aplikasi yang sama tanpa MVVM, dimana pada parameter WMC terjadi penurunan sebesar 4,5 poin, dan pada parameter DIT sebesar 0,3 poin, sementara pada parameter CBO sebanyak 0,4 poin, dan pada parameter LCOM sebesar 31,2 poin. Hal ini mengindikasikan bahwa aplikasi penghitung kalori yang menggunakan pola desain MVVM lebih mudah dalam perawatannya.

Kata kunci: Android, Model View ViewModel, CK Metrics, Reusability, Pola Desain.

Abstract

The aim of this study is to design an efficient calorie counter application using the Model-View-ViewModel (MVVM). The use of the Model-View-ViewModel design pattern is based on efforts to facilitate the process of reusing source code (Reusability). The design process uses flutter as an application framework. This research can be used as a reference regarding the Model-View-ViewModel, as well as providing new knowledge for researchers regarding the combination of the Model-View-ViewModel design pattern with the flutter framework on the android platform. Data collection in this study was carried out through literature studies and evaluation of the system design. The application was developed using the Dart programming language, while the evaluation of the implementation of MVVM design pattern is measured by using CK metrics. The evaluation shows that the MVVM calorie counter application can be run on Bluestacks Android Emulator successfully. In addition, the reusability degree of the application is better than the non-MVVM calorie counter application, where the WMC parameter decreased by 4.5 points, and the DIT parameter by 0.3 points, while the CBO parameter decreased by 0.4 points and the LCOM parameter by 31.2 points. The results indicate that the MVVM calorie counter application is easier to maintain.

Keywords : Android, Model View ViewModel, CK Metrics, Reusability, Design Pattern

* Corresponding author : Mohammad Fajar (fajar@kharisma.ac.id)

1. Pendahuluan

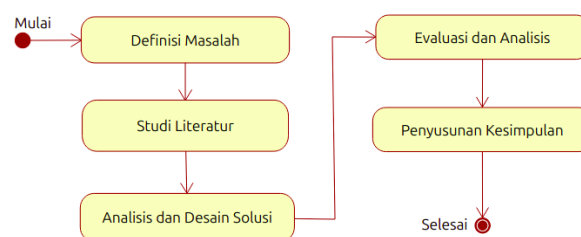
Pemanfaatan dan studi sistem-sistem berbasis komputer di bidang kesehatan terus-menerus dilakukan untuk meningkatkan efektifitas dan layanan kesehatan. Seperti sistem informasi rumah sakit atau klinik Kesehatan [1], sistem penunjang keputusan klinis [2], monitoring kondisi pasien berbasis teknologi IoT [3], [4], hingga pemanfaatan kecerdasan buatan untuk deteksi dan diagnosis potensi penyakit [5] [6]. Salah satu perangkat lunak aplikasi kesehatan dan kebugaran yang dianggap penting dan menarik banyak minat masyarakat untuk digunakan yaitu aplikasi monitoring konsumsi kalori dan pola hidup sehat atau *fitness* [7],[8]. Banyak studi telah dilakukan, dalam rangka memenuhi kebutuhan masyarakat akan aplikasi monitoring aktivitas diet atau kebutuhan kalori dan pola hidup sehat tersebut, seperti evaluasi dan ulasan kualitas serta efektivitas sejumlah aplikasi penghitung kalori dan *fitness* [9], [10]. Penelitian [11], mengkaji pengembangan aplikasi penghitung kalori berbasis *platform android*. Demikian juga penelitian Zhou, et al. [12] yang merancang aplikasi *fitness* untuk merekam aktivitas berupa jumlah langkah yang dilakukan setiap hari. Sementara studi [13] melakukan kajian terkait penambahan fitur pengenalan otomatis jenis makanan tradisional pada prototip aplikasi penghitung kalori menggunakan histogram warna dan jaringan syaraf tiruan. Akan tetapi, penelitian-penelitian tersebut belum membahas tentang penggunaan metode atau teknik perancangan aplikasinya. Menurut Sahly, Eiman M. & Sallabi, Omar M. [14], penerapan metode atau teknik desain tertentu seperti pola desain dapat meningkatkan kualitas sebuah perangkat lunak. Masalah dan tantangan dari sisi kebutuhan dan desain aplikasi juga diungkapkan oleh penelitian [15], dimana diperlukan studi lebih lanjut terkait aspek metode *agile* dan efisiensi rekayasa kebutuhan dari setiap pengembangan perangkat lunak.

Dalam pengembangan perangkat lunak aplikasi kesehatan, salah satu parameter penting yang perlu dipertimbangkan dan diperhatikan oleh tim pengembang yaitu *reusability*. Aspek *reusability* atau dapat digunakan ulang adalah potensi atau kemampuan dari aset-aset sebuah perangkat lunak untuk digunakan kembali selama siklus pengembangan [16] [17]. Aset perangkat lunak yang dapat digunakan kembali ini dapat berupa kode sumber, komponen pustaka, API, *framework*, rancangan, hingga dokumentasi. Sebuah perangkat lunak dengan tingkat *reusability* yang baik berdampak pada kualitas, produktivitas dan konsistensi untuk dapat berfungsi dalam jangka waktu yang lama. Selain itu, tidak kalah pentingnya juga berdampak pada penghematan biaya pengembangan, waktu implementasi dan kemudahan perawatan. Salah satu pendekatan yang cukup efektif untuk meningkatkan aspek *reusability* perangkat lunak, yaitu pemakaian prinsip pemisahan kepentingan menggunakan pola desain (*design pattern*) [18], [19]. Sehingga dalam penelitian ini dikaji pemakaian pola desain *Model-View-View-Model* (MVVM) untuk pengembangan aplikasi penghitung kalori. Studi oleh Sun, Chen & Yu [20], memperlihatkan bahwa keuntungan perancangan perangkat lunak menggunakan MVVM yaitu mampu meningkatkan reusabilitas dan menurunkan biaya pengembangan *perangkat lunak*.

Pola desain telah banyak dibahas dalam literatur. Diantaranya [21] yang merancang aplikasi web menggunakan framework *model view controller* (MVC). dan [22], yang menerapkan prinsip MVC pada sistem sistem informasi. Demikian pula penelitian [23], menerapkan pola desain MVVM pada aplikasi Bank Sampah Digital. Meskipun telah memanfaatkan pola desain, tetapi studi-studi ini belum menyajikan ataupun menjelaskan struktur implementasi sistem aplikasi dimana pola desain MVC ataupun MVVM digunakan. Di sisi lain, studi [24], mengkaji penerapan pola desain MVVM dalam pengembangan aplikasi pembelajaran *mobile serta menyajikan desain* dan implementasi MVVM sebagaimana yang diusulkan dalam penelitian ini. Akan Tetapi evaluasi yang dilakukan masih terbatas pada metode *black-box* dimana tingkat *reusability* aplikasi belum dapat diukur. Pola desain, seperti MVC dan MVVM berdampak terhadap kualitas sistem atau aplikasi, khususnya aspek *reusability*. Sehingga perlu dipertimbangkan evaluasi pengukuran aspek tersebut. Oleh karena itu, penelitian ini bertujuan untuk merancang aplikasi penghitung kalori yang lebih efisien menggunakan Model-View-View-Model. Diharapkan studi ini memberikan informasi penting terkait pemakaian pola desain MVVM dalam meningkatkan aspek *reusability* perangkat lunak termasuk metrik untuk evaluasinya.

2. Metode Penelitian

Penelitian diawali dengan identifikasi dan definisi masalah terkait aspek *reusability* pada aplikasi penghitung kalori, selanjutnya dilakukan studi literatur dengan mengumpulkan data dan informasi terkait solusi dari masalah *reusability* tersebut, dalam hal ini pemakaian pola desain, dilanjutkan ke tahapan analisis kebutuhan aplikasi dan desain solusi-nya. Evaluasi dilakukan untuk validasi desain solusi yang diusulkan, hasil evaluasi dianalisis untuk kemudian dijadikan sebagai dasar penyusunan kesimpulan penelitian. Gambar 1 menyajikan tahapan-tahapan dalam penelitian ini.



Gambar 1. Tahapan Penelitian

2.1. Teknik Pengumpulan Data

Pengumpulan data penelitian dilakukan melalui studi literatur terkait pola desain MVVM, dan pengujian rancangan aplikasi diatas *platform Bluestacks Android Emulator* Profil ONEPLUS A3010, Android 7.1/Nougat serta perhitungan aspek *reusability* menggunakan CK metrics untuk

aplikasi penghitung kalori. Hasil pengujian dan perhitungan ini dijadikan dasar penyusunan kesimpulan.

2.2. Model View ViewModel (MVVM)

Menurut Sorensen & Mihailescu [25] *Model-View-ViewModel* adalah sebuah pola yang dikemukakan oleh John Gossman pada tahun 2005, dimana pola tersebut merupakan variasi dari pola *Presentation Model* (PM) yang dipopulerkan oleh Martin Fowler pada tahun 2004. MVVM adalah sebuah pola desain yang memisahkan *domain logic* dari *presentation logic* menjadi 3 bagian [26], yaitu pertama terdapat *View (Presentation Layer)*, yang merupakan komponen yang bertugas untuk berinteraksi dengan pengguna, dengan cara menampilkan *output* dan menerima *input*. *Input* yang terjadi akan direkam oleh *View*, dan diteruskan ke *ViewModel*. Kedua terdapat *ViewModel* yang merupakan komponen yang bertugas menangani komunikasi antara *View* dan *Model*. *View* meneruskan data ke *ViewModel*, lalu *ViewModel* akan mengubah *event* dan data tersebut dalam format yang dapat diproses oleh *Model* sebelum diteruskan ke *Model*. *ViewModel* juga bertugas untuk menerjemahkan bahasa dari *Model*, agar bisa diproses dan ditampilkan oleh *View*. Lapisan atau layer terakhir yaitu *Model*, merupakan komponen yang bertugas untuk menangani dan memproses data. *Model* menangani bagian komputasi dan logika dalam pola MVVM.

2.3. Evaluasi Menggunakan Chidamber and Kemerer Metrics (CK Metrics)

CK Metrics pertama kali diusulkan oleh Chidamber dan Kemerer [27] yang bertujuan untuk mengembangkan dan mengimplementasikan sebuah instrumen atau metrik baru untuk desain sistem berorientasi objek. CK Metrics adalah metrik yang ditujukan untuk mengevaluasi perangkat lunak atau sistem berorientasi objek.

Penelitian ini berfokus kepada pemakaian ulang aset dari aplikasi yang dibuat, khususnya kode sumbernya. Sehingga metrik berfungsi mengevaluasi aspek *reusability* dari aplikasi penghitung kalori, baik yang menggunakan pola desain MVVM maupun aplikasi penghitung kalori tanpa MVVM. Menurut penelitian yang dilakukan oleh [28], beberapa metrik yang dapat mengevaluasi tingkat *reusability* yaitu:

- A. *Weighted Methods Per Class* (WMC) - WMC adalah jumlah metode yang terdapat didalam sebuah *class*. Semakin banyak metode yang terdapat dalam sebuah *class*, maka akan semakin banyak waktu yang digunakan untuk mengembangkan kembali dan untuk melakukan kegiatan perawatan. Jika nilai WMC kecil, maka akan semakin mudah penggunaan kembali *class* tersebut.
- B. *Depth of Inheritance Tree* (DIT) - DIT adalah panjang dari hirarki inheritansi yang ada pada sebuah *class*, yang dapat dihitung dengan menghitung hirarki *parent* dari *class* tersebut. Semakin rendah posisi dalam hirarki, maka semakin banyak juga metode yang diwarisi sehingga membuat *class* tersebut menjadi semakin kompleks, dan sensitif terhadap perubahan

class lain.

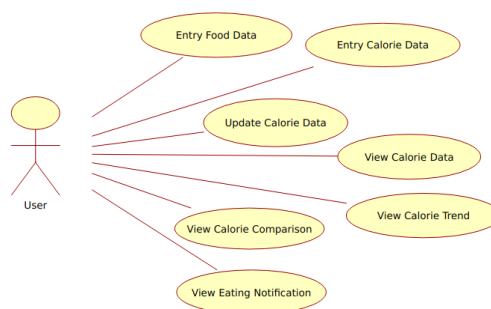
- C. *Coupling Between Objects* (CBO) – CBO adalah jumlah dari *class* yang saling terhubung dengan *class* lain (kebergantungan *class*). Nilai CBO sebuah *class* bergantung pada jumlah *class* yang terhubung dengan *class* yang sedang dihitung. Sebuah *class* dapat dikatakan terhubung jika terdapat sebuah nilai yang dipertukarkan antar *class* (seperti pengiriman parameter), dan atau jika sebuah *class* menggunakan metode yang dideklarasikan di *class* lain. Semakin banyak *coupling* yang terjadi, akan membuat *class* semakin bergantung antar satu sama lain, dan membuat penggunaan kembali *class* tersebut menjadi semakin sulit dikarenakan *class* tersebut sulit untuk diubah tanpa mempengaruhi *class* lain. *Class* yang dirancang sedapat mungkin lebih independen.
- D. *Lack of Cohesion in Methods* (LCOM) – LCOM adalah perhitungan untuk kohesi yang terjadi pada *class*. Kohesi yang dimaksudkan yaitu pemakaian sebuah atribut/*field* data yang sama oleh metode-metode yang terdapat didalam sebuah *class*. Nilai kohesi yang rendah, dapat meningkatkan kompleksitas sebuah program dan meningkatkan kemungkinan terjadinya error pada proses pengembangan. Semakin tinggi nilai LCOM, maka kohesi program akan semakin rendah. *Class-class* yang dirancang sedapat mungkin memiliki tugas yang spesifik dan tunggal, sehingga lebih independen.

3. Hasil dan Pembahasan

Bagian ini membahas hasil analisis dan desain aplikasi serta evaluasinya.

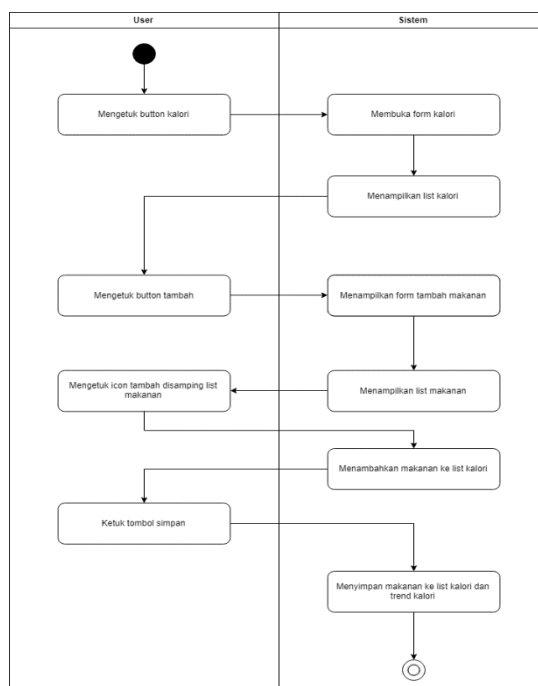
3.1. Analisis dan Desain Aplikasi

Pada awal pengembangan aplikasi penghitung kalori, belum digunakan pendekatan ataupun pola desain tertentu. Aplikasi masih dikembangkan secara *ad-hoc*. Hal ini membuat pembagian peran pada struktur aplikasi menjadi tidak jelas, dan berdampak pada reusabilitas sistem secara keseluruhan. Tidak ada pembagian spesifik antara antarmuka pengguna (UI), proses menampilkan data dan juga proses menyimpan datanya. Sehingga aplikasi dikembangkan kembali menggunakan pola desain MVVM agar strukturnya menjadi lebih jelas. Dalam proses analisis dan spesifikasi kebutuhan (*requirement*), sejumlah fitur diidentifikasi dan dianalisis seperti: Lihat *trend* kalori, lihat perbandingan kalori, dan Lihat notifikasi jam makan. Pengguna aplikasi (*User*) sebagai aktornya dapat berinteraksi langsung dengan setiap fitur aplikasi. Hasil analisis kebutuhan aplikasi kalori *counter* divisualisasikan melalui *diagram Use-case* yang disajikan pada Gambar 2.



Gambar 2. Use Case Diagram Aplikasi Penghitung Kalori

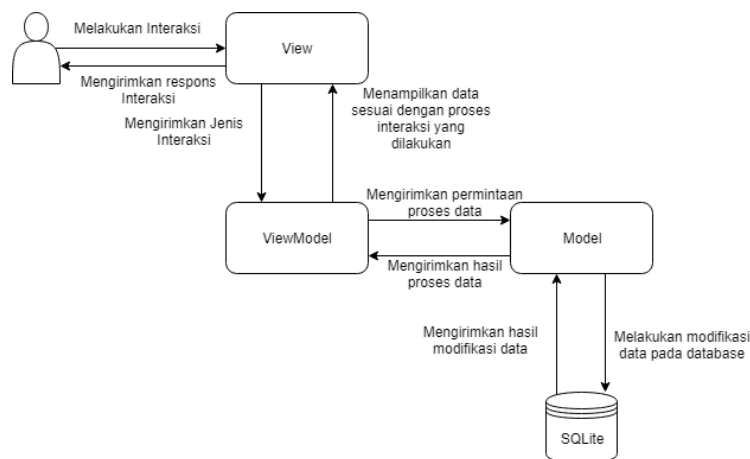
Selanjutnya, masing-masing *use case* dianalisis untuk memahami alur kerja atau skenario yang terjadi menggunakan diagram aktivitas (*activity diagram*). Gambar 3 menyajikan salah satu contoh analisis proses memasukkan data kalori yang dikonsumsi menggunakan diagram aktivitas.



Gambar 3. Diagram Aktivitas Tambah Kalori

Pada rancangan alur dari arsitektur sistem, pengguna melakukan interaksi dengan komponen *view*. Interaksi yang dilakukan diteruskan ke komponen *viewmodel*, dan kemudian *viewmodel* akan meminta perubahan data kepada komponen *model*. *Model* akan memproses

data, lalu modifikasi pada *database* sesuai dengan permintaan dilakukan. *Model* kemudian memberikan umpan balik data yang telah dimodifikasi ke komponen *viewmodel*, lalu *viewmodel* memproses data tersebut sehingga ditampilkan di *view*, sesuai dengan interaksi yang dilakukan dengan *user*. Gambar 4 menyajikan rancangan arsitektur aplikasi penghitung kalori yang menggunakan pola desain MVVM yang diusulkan dalam penelitian ini.



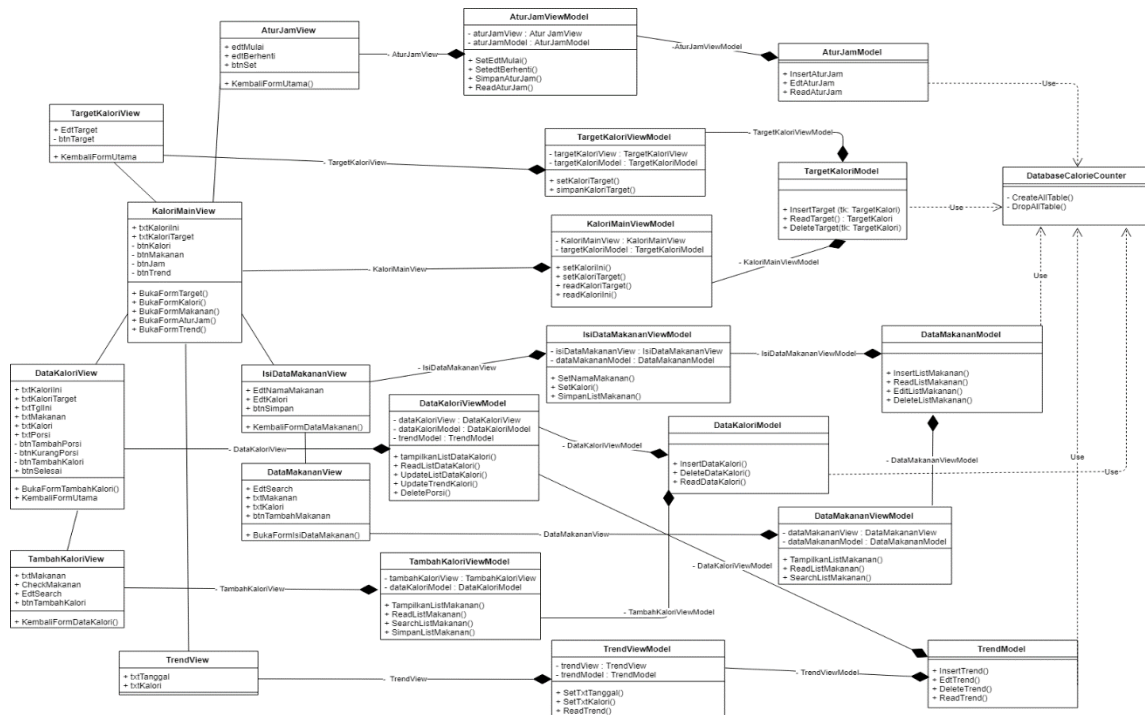
Gambar 4. Desain Arsitektur Aplikasi

Selanjutnya dirancang struktur sistem yang menerapkan MVVM menggunakan *class diagram*. Dalam *class diagram* tersebut, digambarkan relasi di dalam aplikasi penghitung kalori. Dibandingkan dari sistem sebelumnya yang belum menggunakan MVVM, rancangan *class diagram* untuk sistem penghitung kalori yang menerapkan MVVM mengalami perubahan yang cukup signifikan. Setiap form memiliki arsitektur *View*, *ViewModel*, dan *Model*. *View* bertugas sebagai *output* untuk menampilkan formulir di layar, sekaligus menerima *input* dari pengguna atau dapat disebut juga sebagai *Presentation Layer*. Terdapat 8 formulir pada penghitung kalori, sehingga terdapat pula 8 *view* yang bertugas untuk menampilkan formulir, yaitu *KaloriMainView*, *TargetKaloriView*, *DataKaloriView*, *TambahKaloriView*, *IsiDataMakananView*, *DataMakananView*, *AturJamView*, dan *TrendView*.

ViewModel bertugas sebagai perantara antara *View* dan *Model*. Interaksi yang membutuhkan pengambilan data dari *database*, dan juga interaksi yang melakukan perubahan pada *database*. *ViewModel* merupakan lapisan bisnis yang bertugas untuk menentukan data apa saja yang perlu ditampilkan di *View*. *ViewModel* dalam aplikasi ini adalah, *KaloriMainViewModel*, *TargetKaloriViewModel*, *DataKaloriViewModel*, *TambahKaloriViewModel*, *IsiDataMakananViewModel*, *DataMakananViewModel*, *AturJamViewModel*, dan *TrendViewModel*.

Model bertugas sebagai *Data Access Layer*, dimana semua data yang berhubungan dengan *database* dikelola dan diproses. *Model* memproses data, sesuai dengan permintaan dari

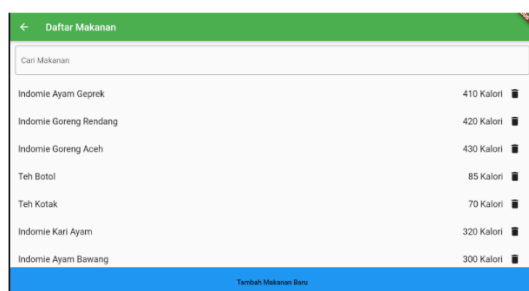
ViewModel, lalu memberikan hasilnya kembali kepada *ViewModel* untuk ditampilkan pada *View*. Komponen *Model* dalam aplikasi ini yaitu, *TargetKaloriModel*, *DataKaloriModel*, *DataMakananModel*, *AturJamModel*, dan *TrendModel*. Gambar 5 menyajikan *Class Diagram* hasil rancangan.



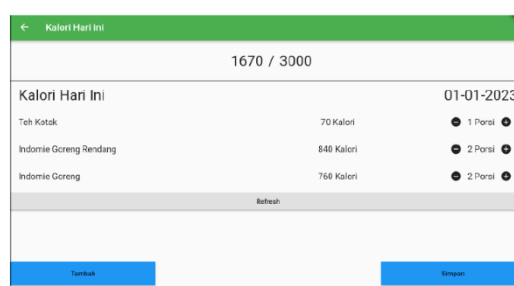
Gambar 5. Desain struktur MVVM pada aplikasi menggunakan *class diagram*

3.2. Evaluasi Aplikasi

Rancangan sistem diimplementasikan menggunakan platform *Android Studio* versi 4.11. dengan *Bluestacks Android Emulator* (Profil ONEPLUS A3010, Android 7.1/Nougat). Gambar 5 dan Gambar 6 memperlihatkan fitur *Entry Food Data* dan *Entry Calorie Data* yang berhasil dijalankan. Halaman Fitur *Entry Food Data* digunakan untuk menambahkan data makanan ke kalori kedalam daftar kalori dengan mengetuk ikon tambah, kemudian *user* memasukkan nama makanan dan data kalori per porsi-nya. Sementara halaman fitur *Entry Calorie Data* berfungsi menampilkan kalori yang sudah diinput oleh pengguna pada hari yang sama. Pengguna dapat mengubah porsi, dimana jika porsi telah mencapai angka 0, maka makanan tersebut akan otomatis terhapus dari daftar kalori. Tombol tambah akan membuka formulir tambah kalori, dan simpan akan meng-*update* dan mengembalikan pengguna ke halaman utama.



Gambar 6. Entry Food Data



Gambar 7. Entry Calorie Data

Untuk evaluasi *reusability* diukur menggunakan CK Metrics terhadap rancangan aplikasi penghitung kalori yang belum menerapkan MVVM dan aplikasi yang telah menerapkan MVVM. Evaluasi melibatkan empat parameter yang berfungsi untuk mengukur *reusability*, yaitu *Weighted Method per Class (WMC)*, *Depth of Inheritance Tree (DIT)*, *Coupling Between Object (CBO)*, dan *Lack of Cohesion in Method (LCOM)*. Keseluruhan class dan metode yang ada pada kode, termasuk mekanisme pewarisan kedua aplikasi perlu dihitung. Dari hasil evaluasi diketahui terdapat penurunan nilai pada parameter WMC, DIT, CBO dan juga LCOM. Penurunan tersebut menandakan tingkat *reusability* yang semakin baik, sehingga dalam penelitian ini dapat diketahui bahwa penerapan MVVM mampu meningkatkan aspek *reusability* pada aplikasi yang menggunakan MVVM. Tabel 1 menyajikan data evaluasi CK Metrics secara keseluruhan untuk dua kasus aplikasi penghitung kalori.

Tabel 1. Hasil Pengukuran CK Metrics Terhadap Aplikasi

Kasus Uji	WMC	DIT	CBO	LCOM
Aplikasi tanpa MVVM	8,2	0,8	2,5	35,9
Aplikasi dengan MVVM	3,681818	0,454545	2,136364	4,727273
Selisih	4,518182	0,345455	0,363636	31,17273

4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan, disimpulkan bahwa rancangan aplikasi penghitung kalori yang menggunakan pola desain MVVM berhasil dijalankan di atas *platform Android* menggunakan *Bluestacks Android Emulator*. Hasil pengukuran, memperlihatkan aplikasi memiliki tingkat *reusability* yang lebih baik dimana pada parameter WMC terjadi penurunan sebesar 4,5 poin, dan pada parameter DIT sebesar 0,3 poin, sementara pada parameter CBO sebanyak 0,4 poin, dan pada parameter LCOM sebesar 31,2 poin. Hal ini

mengindikasikan bahwa aplikasi penghitung kalori yang menggunakan pola desain MVVM lebih mudah dalam perawatannya dibanding aplikasi yang sama yang tidak menggunakan MVVM.

Daftar Pustaka

- [1] J. Sligo, R. Gauld, V. Roberts, and L. Villa, "A literature review for large-scale health information system project planning, implementation and evaluation," *Int J Med Inform*, vol. 97, pp. 86–97, 2017, doi: <https://doi.org/10.1016/j.ijmedinf.2016.09.007>.
- [2] R. T. Sutton, D. Pincock, D. C. Baumgart, D. C. Sadowski, R. N. Fedorak, and K. I. Kroeker, "An overview of clinical decision support systems: benefits, risks, and strategies for success," *NPJ Digit Med*, vol. 3, no. 1, p. 17, 2020, doi: [10.1038/s41746-020-0221-y](https://doi.org/10.1038/s41746-020-0221-y).
- [3] Y. Yuhefizar, A. Nasution, R. Putra, E. Asri, and D. Satria, "Alat Monitoring Detak Jantung Untuk Pasien Beresiko Berbasis IoT Memanfaatkan Aplikasi OpenSID berbasis Web," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 2, pp. 265–270, Aug. 2019, doi: [10.29207/resti.v3i2.974](https://doi.org/10.29207/resti.v3i2.974).
- [4] M. A. Akkaş, R. SOKULLU, and H. Ertürk Çetin, "Healthcare and patient monitoring using IoT," *Internet of Things*, vol. 11, p. 100173, 2020, doi: <https://doi.org/10.1016/j.iot.2020.100173>.
- [5] Y. Muhammad, M. Tahir, M. Hayat, and K. T. Chong, "Early and accurate detection and diagnosis of heart disease using intelligent computational model," *Sci Rep*, vol. 10, no. 1, p. 19747, 2020, doi: [10.1038/s41598-020-76635-9](https://doi.org/10.1038/s41598-020-76635-9).
- [6] A. Singh, H. Mahapatra, A. K. Biswal, M. Mahapatra, D. Singh, and M. Samantaray, "Heart Disease Detection Using Machine Learning Models," *Procedia Comput Sci*, vol. 235, pp. 937–947, 2024, doi: <https://doi.org/10.1016/j.procs.2024.04.089>.
- [7] P. Jones, V. Ratten, and T. Hayduk, "Sport, fitness, and lifestyle entrepreneurship," *International Entrepreneurship and Management Journal*, vol. 16, no. 3, pp. 783–793, 2020, doi: [10.1007/s11365-020-00666-x](https://doi.org/10.1007/s11365-020-00666-x).
- [8] S. Angosto, J. García-Fernández, and M. Grimaldi-Puyana, "A systematic review of intention to use fitness apps (2020–2023)," *Humanit Soc Sci Commun*, vol. 10, no. 1, p. 512, 2023, doi: [10.1057/s41599-023-02011-3](https://doi.org/10.1057/s41599-023-02011-3).
- [9] P. Banerjee, V. V. R. Mendu, D. Korrapati, and S. R. M. Gavaravarapu, "Calorie counting smart phone apps: Effectiveness in nutritional awareness, lifestyle modification and weight management among young Indian adults," *Health Informatics J*, vol. 26, no. 2, pp. 816–828, Jun. 2020, doi: [10.1177/1460458219852531](https://doi.org/10.1177/1460458219852531).
- [10] I. Khaghani-Far, S. Nikitina, M. Báez, E. A. Taran, and F. Casati, "Fitness Applications for Home-Based Training," *IEEE Pervasive Comput*, vol. 15, no. 4, pp. 56–65, 2016, doi: [10.1109/MPRV.2016.76](https://doi.org/10.1109/MPRV.2016.76).
- [11] A. Adewumi, G. Olatunde, S. Misra, R. Maskeliūnas, and R. Damaševičius, "Developing a Calorie Counter Fitness App for Smartphones," in *Information Technology Science*, T. Antipova and Á. Rocha, Eds., Cham: Springer International Publishing, 2018, pp. 23–33.
- [12] M. Zhou et al., "Personalizing Mobile Fitness Apps using Reinforcement Learning HHS Public Access," in *CEUR workshop proceedings*, A. Said and T. Komatsu, Eds., Tokyo, Mar. 2018. Accessed: Dec. 14, 2024. [Online]. Available: <http://ceur-ws.org/Vol-2068/humanize7.pdf>

- [13] N. A. A. N. Muhammad, C. P. Lee, K. M. Lim, and S. F. A. Razak, "Malaysian food recognition and calorie counter application," in 2017 IEEE 15th Student Conference on Research and Development (SCORED), 2017, pp. 445–450. doi: 10.1109/SCORED.2017.8305442.
- [14] E. M. Sahly and O. M. Sallabi, "Design pattern selection: A solution strategy method," in 2012 International Conference on Computer Systems and Industrial Informatics, 2012, pp. 1–6. doi: 10.1109/ICCSII.2012.6454337.
- [15] B. Maqbool and S. Herold, "Challenges in Developing Software for the Swedish Healthcare Sector." [Online]. Available: <https://drive.google.com/file/d/>
- [16] B. Mehboob, C. Y. Chong, S. P. Lee, and J. M.-Y. Lim, "Reusability affecting factors and software metrics for reusability: A systematic literature review," *Softw Pract Exp*, vol. 51, pp. 1416–1458, 2021, [Online]. Available: <https://api.semanticscholar.org/CorpusID:233478223>
- [17] M. Burgin, H. K. Lee, and N. Debnath, "Software technological roles, usability, and reusability," in Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004., 2004, pp. 210–214. doi: 10.1109/IRI.2004.1431462.
- [18] F. E. Shahbudin, "Design Patterns for Developing High Efficiency Mobile Application," *J Inf Technol Softw Eng*, vol. 03, no. 03, 2013, doi: 10.4172/2165-7866.1000122.
- [19] L. A. T. Nguyen, T. S. Huynh, D. T. Tran, and Q. H. Vu, "Design and Implementation of Web Application Based on MVC Laravel Architecture," *European Journal of Electrical Engineering and Computer Science*, vol. 6, no. 4, pp. 23–29, Aug. 2022, doi: 10.24018/ejece.2022.6.4.448.
- [20] W. Sun, H. Chen, and W. Yu, "The Exploration and Practice of MVVM Pattern on Android Platform," in Proceedings of the 2016 4th International Conference on Machinery, Materials and Information Technology Applications, Atlantis Press, Jan. 2017. doi: 10.2991/icmmita-16.2016.205.
- [21] L. Yosephine, H. Arfandy, and B. Zaman, "RANCANG BANGUN APLIKASI WEB UNTUK PELELANGAN PONSEL BEKAS MENGGUNAKAN FRAMEWORK MODEL VIEW CONTROLLER," *JTRISTE*, vol. 8, no. 1, pp. 21–28, 2021, [Online]. Available: <https://jurnal.kharisma.ac.id/jtriste/article/view/164>
- [22] R. F. Al Fikri and E. Hernawati, "PENERAPAN PRINSIP MODEL VIEW CONTROLLER PADA SISTEM INFORMASI DOSEN BERBASIS WEBSITE," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 9, no. 1, pp. 69–78, Dec. 2023, doi: 10.36341/rabit.v9i1.4067.
- [23] M. Ibra Alfathar et al., "PENERAPAN MVVM (MODEL VIEW VIEWMODEL) PADA PENGEMBANGAN APLIKASI BANK SAMPAH DIGITAL," *J. Ris. dan Apl. Mhs. Inform.*, vol. 05, no. 02, pp. 406–414, 2024, [Online]. Available: <https://jim.unindra.ac.id/index.php/jrami/article/view/11071>
- [24] H. Muhammad Iqbal, V. Putra, and D. Muriyatmoko, "Penerapan Pola Arsitektur Model-View-ViewModel pada Aplikasi Pembelajaran Shorof Berbasis Mobile," in Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK), 2024, pp. 101–107. [Online]. Available: <https://seminar.iaii.or.id/index.php/SISFOTEK/article/view/452>
- [25] M. Mihailescu and E. Sorensen, "Model-View-ViewModel (MVVM) Design Pattern using Windows Presentation Foundation (WPF) Technology," *Megabyte Journal*, Oct. 2010.

-
- [26] J. Patrzyk, B. Patrzyk, K. Rycerz, and M. Bubak, "Towards A Novel Environment For Simulation Of Quantum Computing," *Computer Science*, vol. 16, p. 103, Jan. 2015, doi: 10.7494/csci.2015.16.1.103.
- [27] S. R. Chidamber and C. Kemerer, "Kemerer, C.F.: A metric suite for object oriented design. *IEEE Trans. Softw. Eng.* 20(6), 476-493," *Software Engineering, IEEE Transactions on*, vol. 20, pp. 476–493, Jul. 1994, doi: 10.1109/32.295895.
- [28] L. H. Rosenberg and L. E. Hyatt, "Software Quality Metrics for Object-Oriented Environments," 2002. [Online]. Available: <https://api.semanticscholar.org/CorpusID:59779685>