

# Penerapan *Clean Architecture* pada Aplikasi Pemesanan Makanan menggunakan Metode *Slope One Algorithm*

Moh. Hilmy Badrudduja<sup>1</sup>, Ricky Eka Putra<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

<sup>1</sup>[moh.17051204030@mhs.unesa.ac.id](mailto:moh.17051204030@mhs.unesa.ac.id)

<sup>2</sup>[rickyeka@unesa.ac.id](mailto:rickyeka@unesa.ac.id)

**Abstrak** – Aplikasi Pemesanan Makanan merupakan suatu terobosan teknologi yang dibuat supaya masyarakat pengguna Android dapat membeli makanan kapanpun dan dimanapun. Mengonsumsi Makanan yang bemutrisi sangatlah dibutuhkan oleh tubuh terutama pada masa *COVID-19* seperti saat ini. Oleh karena itu, masyarakat membutuhkan sebuah aplikasi yang didalamnya terdapat rekomendasi makanan sesuai nutrisi yang terkandung sehingga dapat menciptakan pola makan yang sehat dan dapat mencegah risiko beberapa penyakit. Aplikasi ini dibuat menggunakan kualitas kode yang *maintainability* dengan menerapkan prinsip *Clean Architecture*. Pengujian pada penditian ini dilakukan dengan membandingkan antara aplikasi yang menerapkan prinsip *Clean Architecture* dengan aplikasi yang menerapkan arsitektur *Model-View-Controller (MVC)*. Hasil pengujian membuktikan bahwa penerapan prinsip *Clean Architecture* mempunyai tingkat *maintainability* lebih besar daripada penerapan arsitektur *MVC* yaitu 78.2% dibanding 54.8%, dimana 78.2% dapat dikatakan *maintainability* karena mendapatkan nilai lebih dari 67%. Selain itu, penelitian ini juga melakukan pengujian Metode *Slope One Algorithm* untuk mengetahui rekomendasi makanan yang baik untuk beberapa kondisi kesehatan (seperti sehat, diabetes, jantung, kelelahan, obesitas, dan sembelit) menjadi tiga kategori makanan berdasarkan keadaan gizi menurut Indeks Antropometri, yaitu Buruk dikonsumsi apabila  $p^{SI} \leq 2.4$ , Kurang baik dikonsumsi apabila  $2.4 < p^{SI} \leq 3.2$ , dan Baik dikonsumsi apabila  $p^{SI} > 3.2$ . Berdasarkan hasil pengujian Metode *Slope One Algorithm*, terdapat beberapa menu makanan yang sebaiknya dihindari oleh kondisi Kesehatan tertentu contohnya pada menu Nasi Goreng *Seafood* dan Mie Goreng *Seafood* yang mempunyai kategori Buruk dikonsumsi oleh seseorang yang mengalami Kelelahan karena mendapatkan nilai  $p^{SI}$  sebesar 1.375 atau kurang dari 2.4.

**Kata Kunci** – Rekomendasi Makanan, *Maintainability*, *Clean Architecture*, *MVC*, *Slope One Algorithm*

## I. PENDAHULUAN

Pada Desember 2019, China pertama kali melaporkan kepada WHO (*World Health Organization*) tentang adanya kasus pneumonia yang berat di wilayah Kota Wuhan, Provinsi Hubei, China. Hal ini diduga berkaitan dengan pasar yang menjual ikan dan berbagai hewan laut yang ada di Wuhan [1]. Penyakit ini awalnya diberi nama sebagai *2019-nCoV* (*2019 novel coronavirus*). Selanjutnya pada 11 Februari 2020 WHO mengumumkan nama baru *2019-nCoV*, yaitu *COVID-19* (*Coronavirus Disease*) yang ditimbulkan oleh virus *SARS-CoV-*

*2* (*Severe Acute Respiratory Syndrome Coronavirus-2*) [2]. *COVID-19* dilaporkan di Indonesia pertama kali pada 2 Maret 2020 dengan sebanyak dua kasus. Data tanggal 31 Maret 2020 membuktikan bahwa kasus yang terkonfirmasi sebanyak 1.528 kasus serta kasus kematian sebanyak 136 kasus. Tingkat mortalitas *COVID-19* di Indonesia termasuk yang tertinggi di Asia Tenggara yaitu sebesar 8.9% [2]. Oleh karena itu, pemerintah menghimbau adanya pembatasan aktivitas di luar rumah yang hal tersebut dapat mengakibatkan berubahnya gaya hidup masyarakat seperti menurunnya aktivitas fisik, meningkatnya konsumsi makanan, dan waktu makan yang kurang teratur [3]. Kebiasaan makan yang salah dapat meningkatkan risiko terkena beberapa penyakit, seperti diabetes, hipertensi, obesitas, serta dislipidemia atau gangguan lemak darah. Sehingga membutuhkan sebuah sistem yang mendorong kebiasaan makan yang salah menjadi lebih baik.

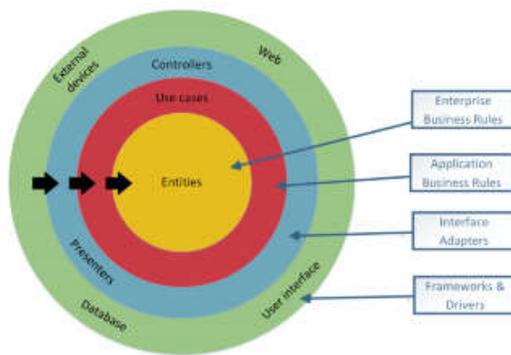
Selain itu, teknologi informasi yang terus berkembang juga telah memberi pengaruh yang cukup signifikan dalam kehidupan sehari-hari. Hal tersebut dapat dilihat dengan banyaknya perangkat *mobile* yang bisa membantu menyelesaikan tugas-tugas yang dikerjakan oleh manusia dalam kehidupan sehari-hari. Salah satu diantaranya dalam bidang kuliner, dimana saat ini kita dapat membeli makanan kapanpun dan dimanapun menggunakan perangkat *mobile* seperti Android.

Android merupakan suatu sistem operasi *smartphone* yang mempunyai pangsa pasar paling besar di Indonesia. Pada Februari 2021, Pangsa pasar Android di Indonesia mencapai 92,03% [4]. Di *Google Play Store* terdapat sekitar 2.986.039 aplikasi Android [5]. Tingginya pangsa pasar Android di Indonesia dan banyaknya aplikasi Android di *Google Play Store* menunjukkan bahwa pengembang Android sangat dibutuhkan di Indonesia, sehingga profesi pengembang Android adalah salah satu profesi yang paling dicari oleh berbagai perusahaan di Indonesia [6]. Aplikasi yang telah dibuat akan terus berkembang mengikuti permintaan pasar, persaingan bisnis, teknologi, serta berbagai kepentingan perusahaan. Hal tersebut menuntut perusahaan untuk selalu melakukan eksplorasi terhadap aplikasi yang dimiliki supaya dapat bersaing dengan perusahaan lainnya. Hal tersebut juga didukung oleh survei tentang pengembangan aplikasi yang menemukan bahwa proses pemeliharaan atau *maintenance* membutuhkan 75-90% biaya dari biaya keseluruhan pengembangan aplikasi [6]. *Maintenance* adalah suatu aktivitas yang memerlukan banyak biaya, waktu, dan sumber daya manusia dalam pengembangan aplikasi yang tujuannya untuk menjaga dan memastikan suatu aplikasi dapat tetap berjalan dengan baik dalam jangka waktu yang lama. *Maintainability* dari suatu aplikasi merupakan tingkatan dari aplikasi tersebut agar bisa dipahami, diperbaiki, ataupun dikembangkan di kemudian hari, serta menjaga aplikasi agar

tetap berjalan baik. Meningkatkan kualitas kode yang rapi dan lebih bersih dapat meningkatkan tingkat *maintainability* dari suatu aplikasi [7].

*Clean Code* adalah konsep yang diperkenalkan oleh Robert C. Martin [8]. Prinsip *Clean Code* semakin populer di antara berbagai pengembang perangkat lunak di industri. Selain itu, bahkan universitas mempertimbangkan untuk mengubah pendekatan mereka tentang cara mengajarkan pengembangan perangkat lunak dengan memperkenalkan pengujian dan kode yang dapat dipelihara terlebih dahulu kepada siswa [8].

Arsitektur perangkat lunak meletakkan dasar dan menetapkan kerangka untuk perangkat lunak berkualitas. Selama beberapa dekade terakhir, banyak pendekatan dan pola desain yang berbeda dikembangkan dan diusulkan oleh para profesional untuk memfasilitasi ketahanan program melalui pola desain yang banyak digunakan dan dihargai oleh orang lain di seluruh dunia. Untuk memastikan sistem perangkat lunak yang dapat dipelihara dan kuat, kerangka yang digabungkan secara longgar, fleksibel dan dapat diperluas harus disediakan sebagai dasar yang siap untuk menerima persyaratan yang terus berubah dan baru. Untuk alasan ini, pendekatan menuju *Clean Architecture* sedang berkembang. *Clean Architecture* mempertimbangkan membangun arsitektur perangkat lunak yang dapat diskalakan dengan membangun di atas prinsip-prinsip *Clean Code* [8]. Berikut pada Gambar 1.1 merupakan Layer *Clean Architecture*.



Gambar 1.1. Layer *Clean Architecture* [8]

*Clean architecture* memberikan kemungkinan untuk mengubah dependensi eksternal atau detail implementasi dari lapisan data atau lapisan tampilan tanpa memengaruhi logika bisnis. Ini juga membuat aplikasi mudah untuk diuji serta mudah untuk menambahkan fitur baru jika implementasi fitur baru mengikuti aturan ketergantungan [9]. Untuk banyak proyek aplikasi, *Clean architecture* bisa sangat berharga. Mengikuti prinsip-prinsip *Clean architecture* menghasilkan aplikasi yang modular, fleksibel, serta dapat diuji [10]. Berdasarkan hasil pengujian yang dilakukan oleh Afiah Taqiu Sondha, dkk pada tahun 2020, "*Framework dan Code Generator Pengembangan Aplikasi Android dengan Menerapkan Prinsip Clean Architecture*" membuktikan bahwa aplikasi yang dikembangkan dengan menggunakan sistem dapat menghemat biaya dan waktu sebesar 42% daripada aplikasi yang dikembangkan dengan tidak menggunakan sistem. Pengujian ini dilakukan menggunakan matriks *McCall's Software Quality Model*. Hasil pengujian lain juga membuktikan bahwa pengembangan aplikasi dengan menggunakan sistem mempunyai tingkat *Maintainability* sebesar 81% atau 0,81 [6]. Penelitian lain juga dilaksanakan oleh

Nayab Akhtar dan Sana Ghafoor pada tahun 2020, "*Analysis of Architectural Patterns for Android Development*". Penelitian tersebut membandingkan antara *MVC*, *MVP*, *MVVM*, dan *Clean Architecture* dari segi *maintainability*, *testability*, *reusability* untuk pengembangan Android. Hasil pengujian menyimpulkan bahwa Arsitektur *MVC*, *MVP*, dan *MVVM* cukup untuk proyek-proyek kecil dan jangka waktu yang pendek, akan tetapi pada arsitektur ini memiliki beberapa kekurangan seperti perubahan atau penambahan fitur yang sulit dilakukan dan apabila basis kode menjadi besar maka pemisahan tanggung jawab dari setiap basis kode akan menjadi sulit. Sedangkan *Clean Architecture* membuat basis kode lebih bersih dan ringkas, serta dapat digunakan kembali dikarenakan *Clean Architecture* mudah dalam memisahkan tanggung jawab dari setiap basis kode serta dapat merangani operasi sebanyak mungkin dan meminimalkan logika dalam tampilan [11].

Dari permasalahan di atas diperlukan adanya Aplikasi pemesanan makanan yang didalamnya terdapat fitur rekomendasi makanan bernutrisi yang memiliki tingkat *maintainability* tinggi serta dapat dijalankan dengan mudah pada perangkat Android sehingga pengguna dapat membeli makanan dimanapun dan kapanpun. Pengertian Makanan bernutrisi adalah makanan yang dalam konteks dikonsumsi dan oleh seseorang yang mengkonsumsinya memberikan nutrisi yang bermanfaat (misalnya mineral, vitamin, asam amino esensial, asam lemak esensial, serta serat makanan) dan dapat meminimalkan kandungan yang berpotensi membahayakan (misalnya lemak jenuh dan gula) [12]. Berdasarkan Permenkes RI no.28 tahun 2019 mengenai Angka Kecukupan Gizi yang dianjurkan untuk masyarakat Indonesia dirangkum pada Gambar 1.2. berikut.

Usia	Energi (kcal)	Protein (gr)	Lemak (gr)	Karbohidrat (gr)	Serat (gr)	Air (ml)
Total	Omega 3	Omega 6				
4 - 6 th	1400	28	60	0.6	10	320
7 - 9 th	1800	40	60	0.9	10	280
10 - 12 th	2000	50	60	1.2	12	300
Perempuan 10 - 12 th	1900	50	60	1.0	1.0	280

Gambar 1.2. Angka Kecukupan Gizi yang dianjurkan [13]

Fitur rekomendasi makanan bernutrisi didapatkan berdasarkan referensi seputar nutrisi makanan baik berupa media cetak maupun media *online* serta hasil rekomendasi atau pendapat dari beberapa ahli Kesehatan khususnya dibidang makanan yang nantinya akan dikonversikan kedalam *variable* angka.

Aplikasi pemesanan makanan adalah aplikasi yang dapat dijalankan pada perangkat Android dengan menerapkan prinsip *Clean Architecture* dan menggunakan metode *Slope One Algorithm* untuk menentukan rekomendasi makanan bernutrisi serta hanya terbatas menu makanan yang terdapat pada Waun g Apung 2 Kota Kediri. *Slope one Algorithm* merupakan salah satu algoritma untuk membuat sistem rekomendasi. *Slope One Algorithm* memberi prediksi menurut nilai hasil pencarian dari item-item yang dibandingkan. Kelebihan dari *Slope One Algorithm* dibandingkan dengan algoritma rekomendasi lainnya yaitu *Slope One Algorithm* mudah untuk diimplementasi, efisien dalam melakukan *query*, tidak membutuhkan banyak

requirement dikarenakan rekomendasi menurut rating dari tiap item, serta cukup akurat [14]. Berdasarkan Penelitian yang dilaksanakan oleh Dhama Pratama dan Seng Hansun pada tahun 2017, "Aplikasi Rekomendasi Tempat Makan Menggunakan Algoritma Slope One pada Platform Android". Penelitian tersebut bertujuan untuk mengetahui tingkat kepuasan pengguna saat menggunakan aplikasi menurut tiga kategori, diantaranya 77.11% untuk tingkat kegunaan aplikasi, 73% untuk kualitas informasi yang diberikan, serta 72.67% untuk kualitas tampilan antarmuka aplikasi. Secara keseluruhan, pengguna merasa puas saat menggunakan aplikasi sistem rekomendasi tempat makan Taste dengan nilai sebesar 74,26% yang termasuk dalam kategori positif [15]. Penelitian juga dilakukan oleh Arief Fadillah pada tahun 2019, "Pemanfaatan Algoritma Slope One dengan Metode Collaborative Filtering dalam Mengetahui Makanan Populer". Penelitian ini hanya untuk mengetahui nilai rating dari data menu produk yang ada pada Restoran Gulo Klopo [16].

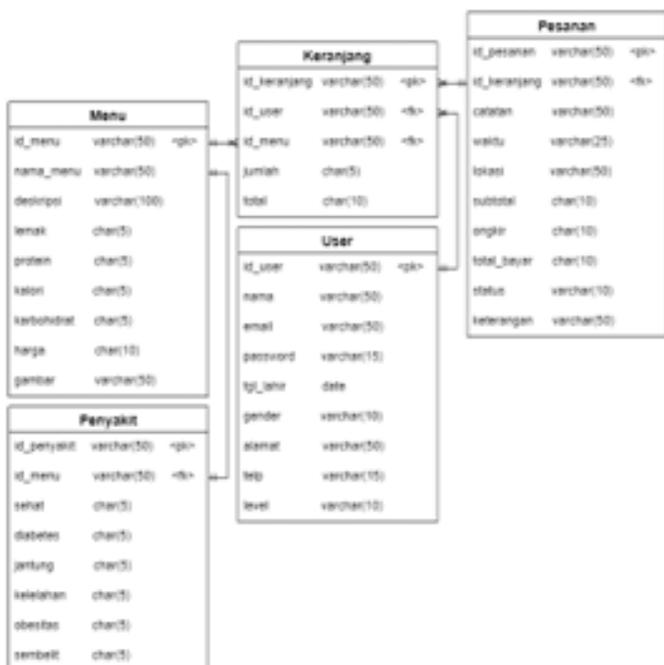
Dengan adanya Aplikasi ini diharapkan agar nantinya masyarakat pengguna *smartphone* berbasis Android dapat membeli makanan kapanpun dan dimanapun yang didalamnya terdapat rekomendasi makanan sesuai nutrisi yang terkandung serta dapat mengetahui kandungan yang terdapat pada makanan tersebut sehingga dapat menciptakan pola makan yang sehat dan dapat mencegah risiko beberapa penyakit.

## II. METODOLOGI PENELITIAN

Pada Metodologi Penelitian berikut ini akan dibahas mengenai tahapan dari perancangan aplikasi pemesanan makanan yang akan dibuat dalam penelitian ini.

### A. Perancangan Database

Perancangan Database merupakan proses untuk menentukan isi dan pengaturan data yang diperlukan dalam mendukung berbagai rancangan aplikasi atau sistem. Perancangan Database bertujuan untuk memenuhi informasi yang berisikan berbagai kebutuhan pengguna secara khusus beserta aplikasinya [17].



Perancangan Database ini dibuat menggunakan *Flowchart Maker* dan untuk pembuatan Database menggunakan *Firestore*. Berikut Gambar 2.1. adalah desain Database dari aplikasi yang akan dibuat.

Gambar 2.1. Desain Database Aplikasi

### B. Perancangan Aplikasi

Untuk melanjutkan dalam pembuatan aplikasi ini maka penulis membutuhkan perancangan sistem aplikasi sebagai berikut.



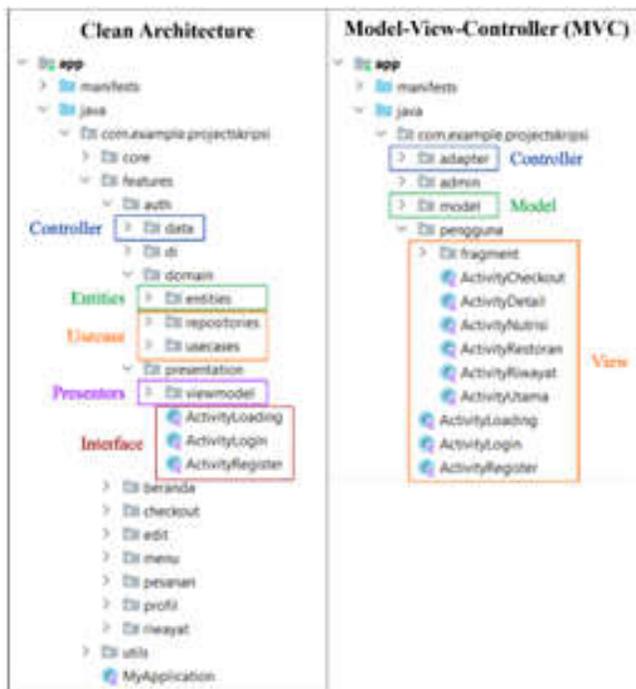
Gambar 2.2. Flowchart atau Diagram Alir Aplikasi

Pada Gambar 2.2. menjelaskan runtutan secara detail bagaimana aktivitas pengguna ketika aplikasi dijalankan. Pertama pengguna diharuskan melakukan *login* atau *register* terlebih dahulu sebelum masuk ke tampilan utama dari aplikasi tersebut. Setelah *login* berhasil, aplikasi akan menampilkan tampilan utama sesuai dengan *email* dan *password* yang pengguna masukkan tadi. Apabila *email* dan *password* yang dimasukkan sebagai admin, maka akan terdapat 4 fitur utama, yaitu : melihat daftar menu makanan, menambahkan mengedit dan menghapus menu makanan, menerima pesanan dari pelanggan, serta melihat status pesanan yang telah masuk. Sedangkan Apabila *email* dan *password* yang dimasukkan sebagai pelanggan, maka akan terdapat 4 fitur utama juga, yaitu : melihat daftar menu makanan, memesan makanan yang diinginkan, melakukan *checkout*, serta melihat status pesanan yang pernah dipesan sebelumnya. Pada Aplikasi ini juga terdapat menu *logout* apabila ingin keluar dari akun pengguna.

### C. Penerapan Prinsip Clean Architecture

Dalam pengembangan aplikasi berbasis Android terdapat beberapa arsitektur seperti *Model-View-Controller (MVC)*, *Clean Architecture*, dan lain sebagainya. Arsitektur *MVC* mempunyai pola desain yang cukup sederhana dikarenakan arsitektur ini membagi semua objek ke dalam tiga *layer* yaitu *Model*, *View*, dan *Controller*. Aplikasi yang menerapkan arsitektur *MVC* tidak menyertakan *business logic* dan *user flows* yang rumit. Kelemahan dari arsitektur *MVC* adalah arsitektur ini tidak dapat memberikan fleksibilitas dan tidak dapat digunakan kembali untuk proyek berskala besar sehingga menyebabkan kekacauan dalam proses pengembangan suatu aplikasi [18].

Sedangkan, Penerapan prinsip *Clean Architecture* dalam suatu aplikasi dilakukan dengan cara memisahkan antar *layer* pada aplikasi tersebut. Suatu aplikasi mempunyai beberapa *layer* yaitu *entities*, *usecases*, *interface adapters*, dan *framework and drivers*. Penerapan prinsip *Clean Architecture* dapat meningkatkan tingkat *maintainability* dari suatu aplikasi. Hal tersebut dikarenakan adanya pemisahan antar komponen pada suatu aplikasi menjadi komponen-komponen yang independen dan lebih modular sehingga apabila terjadi *bug* atau *error* pada suatu komponen, pengembang hanya fokus memperbaiki *bug* atau *error* pada komponen tersebut tanpa memengaruhi komponen lainnya. Penambahan sebuah fitur pada aplikasi yang menerapkan prinsip *Clean Architecture* juga sangatlah mudah dilakukan [6]. Oleh karena itu, arsitektur yang diterapkan dalam aplikasi yang telah dirancang sesuai Gambar 2.2. tersebut menggunakan prinsip *Clean Architecture*. Berikut ini pada Gambar 2.3. menunjukkan perbedaan antara *layer Clean Architecture* dan *layer MVC*.



Gambar 2.3. Perbedaan Layer Clean Architecture dan MVC

Masing-masing *layer* pada aplikasi yang menerapkan prinsip *Clean Architecture* dinamakan *single actionable idea*. Interaksi antar *single actionable idea* harus saling berurutan, tidak boleh melewati *single actionable idea* lainnya. Prinsip *dependency*

*inversion* dapat digunakan dalam komunikasi dan pengiriman data antar *single actionable idea*. Prinsip *dependency inversion* juga dapat mengatasi permasalahan ketergantungan antar komponen dalam suatu aplikasi. Pengiriman data pada tiap-tiap *single actionable idea* harus berupa tipe data primitif atau struktur data sederhana. Pengiriman data antar *single actionable idea* berupa objek akan melanggar prinsip *dependency inversion* dalam pengembangan suatu aplikasi [6].

### D. Klasifikasi Metode Slope One Algorithm

Setelah menentukan perancangan aplikasi, dilanjutkan menghitung klasifikasi Metode *Slope One Algorithm* untuk dapat menentukan rekomendasi makanan yang bernutrisi sesuai dengan kondisi Kesehatan pengguna. *Slope one Algorithm* merupakan salah satu algoritma untuk membuat sistem rekomendasi. *Slope One Algorithm* memberikan prediksi menurut nilai hasil pencarian dari item-item yang dibandingkan. *Slope One Algorithm* melakukan perhitungan menurut hubungan linear dari *weight* atau nilai preferensi dari *item-item* dibandingkan. Estimasi umum dari dasar perhitungan *Slope One Algorithm* yaitu fungsi linear  $y = mx + b$ , dengan asumsi *gradient*  $m = 1$ , sehingga fungsi tersebut menjadi  $b = y - x$ . Cara kerja *Slope One Algorithm* adalah dengan mencari selisih dari suatu *item* dengan *item-item* lain yang dibandingkan [15].

Perhitungan *Slope One Algorithm* dapat diformulasikan untuk pencarian selisih menggunakan persamaan (1).

$$dev_{j,i} = \sum_{u \in S_{j,i}(x)} \frac{u_j - u_i}{card(S_{j,i}(x))} \quad (1)$$

dimana  $dev_{j,i}$  = rata-rata dari selisih *rating item j* dan *i*

$u_j$  = *rating item j*

$u_i$  = *rating item i*

$card(S_{j,i}(x))$  = jumlah elemen yang dibandingkan

Apabila telah mendapatkan selisih, maka selanjutnya dilakukan perhitungan rekomendasi untuk *item j* yang dapat diformulasikan menggunakan persamaan (2).

$$p^{SI}(u)_j = dev_{j,i} + u_j \quad (2)$$

dimana  $p^{SI}(u)_j$  = nilai rekomendasi untuk *item j*

$dev_{j,i}$  = rata-rata dari selisih *rating item j* dan *i*

$u_j$  = *rating item j*

Berdasarkan persamaan-persamaan di atas, *Slope One Algorithm* memberikan rekomendasi dengan melakukan perhitungan selisih dari tiap-tiap *item*. Selisih yang didapatkan akan dirata-rata setiap *item* yang selanjutnya akan dijumlahkan dengan *value* dari tiap-tiap *item*. *Value* yang telah dijumlahkan dengan rata-rata selisihnya akan digunakan sebagai *point* untuk memberikan rekomendasi [15].

Pada penelitian ini dataset yang akan dipakai merupakan dataset dari kandungan nutrisi dari tiap-tiap menu makanan. Menu makanan yang terdapat pada dataset tersebut sesuai dengan menu yang ada pada objek penelitian, yaitu Warung Apung 2 Kota Kediri. Sedangkan kandungan nutrisi pada dataset didapatkan dari berbagai sumber referensi mengenai nutrisi makanan baik berupa media cetak maupun media *online*. Berikut adalah dataset nutrisi makanan yang dipakai dalam penelitian yang dibutuhkan untuk proses rekomendasi dapat dilihat pada Tabel I.

Tabel I. Dataset Nutrisi dari Menu Makanan [19]

Menu	Lemak (g)	Protein (g)	Kalori (kal)	Karbo (g)
Paket Ikan Gurami	6.68	23.05	350	48.54
Paket Ikan Nila	2.51	25.91	318.6	48.54
Paket Ikan Lele	12.99	18.75	417	55.8
Paket Ayam Kampung	8.21	32.78	404	48.54
Paket Ayam Negeri	9.89	24.67	385	48.54
Paket Tahu Tempe Penyet	10.44	12.28	351	54.84
Nasi Goreng	9	9.37	302	45.83
Nasi Goreng Seafood	8.73	8.24	296	45.87
Nasi Goreng Spesial	9.28	9.39	305	45.74
Mie Goreng	9	9.37	302	45.83
Mie Goreng Seafood	8.73	8.24	296	45.87
Mie Goreng Spesial	9.28	9.39	305	45.74

Selain dataset dari kandungan nutrisi dari tiap-tiap menu makanan seperti pada Tabel I, terdapat juga dataset dari beberapa kondisi Kesehatan yang dibutuhkan untuk agar pengguna dapat memesan menu makanan yang sesuai dengan kondisi kesehatannya. Dataset tersebut berisikan sampel beberapa kondisi Kesehatan yang umum diakibatkan karena pola makan yang salah beserta indikator kandungan nutrisi yang sebaiknya dihindari oleh yang mengalami kondisi kesehatan tersebut. Berikut adalah dataset dari beberapa kondisi Kesehatan yang ditunjukkan pada Tabel II

Tabel II. Dataset Nutrisi yang sebaiknya dihindari oleh beberapa Kondisi Kesehatan [20]

Kondisi	Lemak (g)	Protein (g)	Kalori (kal)	Karbo (g)
Sihat	-	-	-	-
Diabetes (K1)	Diatas 25	Diatas 22	Diatas 750	Diatas 75
Jantung (K2)	Diatas 25	Dibawah 88	Diatas 750	Diatas 75
Kelelahan (K3)	Dibawah 10	Dibawah 88	Dibawah 300	Dibawah 30
Obesitas (K4)	Diatas 25	Diatas 22	Diatas 750	Diatas 75
Sembelit (K5)	Diatas 25	Diatas 22	Diatas 750	Diatas 75

Langkah selanjutnya adalah menentukan *variable* yang dibutuhkan sebelum melakukan pengujian metode *Slope One Algorithm*. Penentuan *variable* dilakukan dengan cara mengubah nilai kandungan nutrisi dari menu makanan pada Tabel I sesuai dengan nutrisi yang sebaiknya dihindari oleh beberapa kondisi kesehatan pada Tabel II menjadi beberapa *variable* yaitu 0 apabila nilainya sesuai dan 1 apabila nilainya tidak sesuai. Berikut daftar *variable* yang pada pengujian metode *Slope One Algorithm* yang dirangkum pada Tabel III.

Tabel III. Daftar *Variable* pada Pengujian Metode *Slope One Algorithm*

Menu	Nutrisi	Sihat	K1	K2	K3	K4	K5
Paket Ikan Gurami	Lemak	1	1	1	0	1	1
	Protein	1	0	1	1	0	0
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Paket Ikan Nila	Lemak	1	1	1	0	1	1
	Protein	1	0	1	1	0	0
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1

Paket Ikan Lele	Lemak	1	1	1	1	1	1
	Protein	1	1	1	1	1	1
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Paket Ayam Kampung	Lemak	1	1	1	0	1	1
	Protein	1	0	1	1	0	0
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Paket Ayam Negeri	Lemak	1	1	1	0	1	1
	Protein	1	0	1	1	0	0
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Paket Tahu Tempe Penyet	Lemak	1	1	1	1	1	1
	Protein	1	1	1	1	1	1
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1

Nasi Goreng	Lemak	1	1	1	0	1	1
	Protein	1	1	1	1	1	1
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Nasi Goreng Seafood	Lemak	1	1	1	0	1	1
	Protein	1	1	0	0	1	1
	Kalori	1	1	1	0	1	1
	Karbo	1	1	1	1	1	1
Nasi Goreng Spesial	Lemak	1	1	1	0	1	1
	Protein	1	1	1	1	1	1
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Mie Goreng	Lemak	1	1	1	0	1	1
	Protein	1	1	1	1	1	1
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1
Mie Goreng Seafood	Lemak	1	1	1	0	1	1
	Protein	1	1	0	0	1	1
	Kalori	1	1	1	0	1	1
	Karbo	1	1	1	1	1	1
Mie Goreng Spesial	Lemak	1	1	1	0	1	1
	Protein	1	1	1	1	1	1
	Kalori	1	1	1	1	1	1
	Karbo	1	1	1	1	1	1

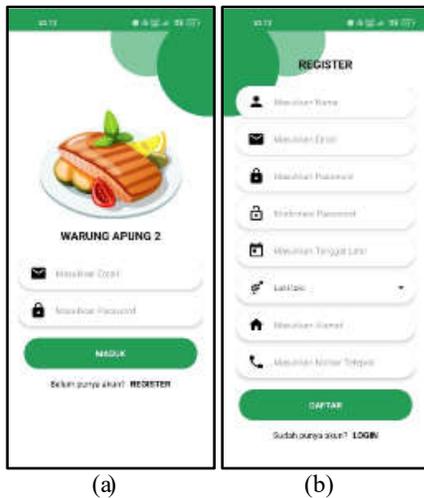
### III. HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan berikut ini akan dibahas mengenai implementasi dari perancangan aplikasi yang telah dijelaskan serta pengujian dari penerapan prinsip *Clean Architecture* dan metode *Slope One Algorithm* pada aplikasi.

#### A. Implementasi Aplikasi

##### 1. Login dan Register

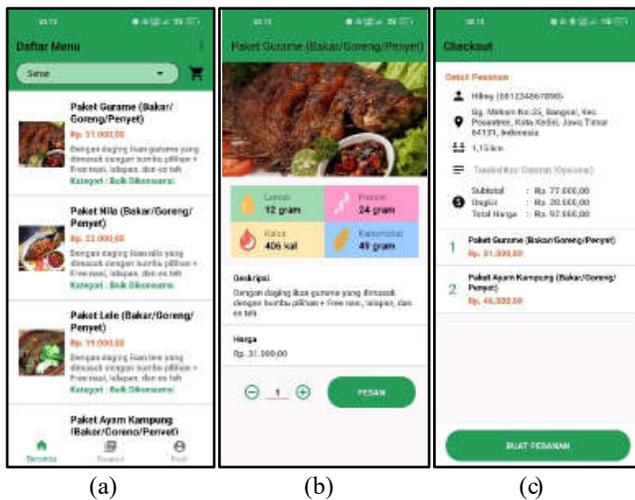
Pada tampilan ini berisikan *login* dan *register* untuk pelanggan dan admin. Disini pengguna tinggal memasukkan *email* dan *password* pada tampilan *login* apabila sudah mempunyai akun, sedangkan yang belum mempunyai akun dapat melakukan registrasi dengan melengkapi data yang ada pada tampilan *register*. Berikut Gambar 3.1. (a) merupakan tampilan *login* dan Gambar 3.1. (b) merupakan tampilan *register*.



Gambar 3.1. (a) Tampilan Login dan (b) Tampilan Register

## 2. Menu Utama untuk Pelanggan

Pada Menu Utama pelanggan dapat melihat daftar menu makanan, harga makanan, kandungan yang terdapat pada makanan tersebut, serta dapat membuat pesanan makanan sesuai dengan yang diinginkan.

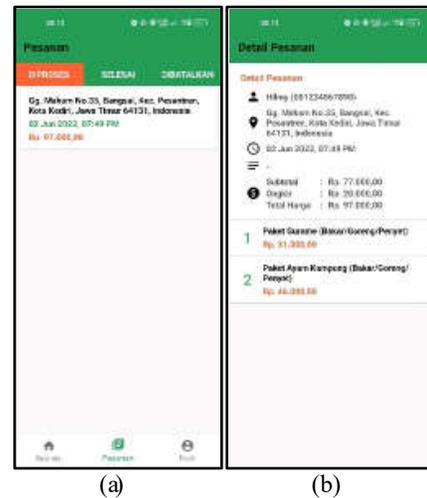


Gambar 3.2. (a) Tampilan Beranda, (b) Tampilan Detail Makanan, dan (c) Tampilan Checkout

Gambar 3.2. (a) merupakan tampilan beranda untuk pelanggan yang berisikan daftar menu makanan dan terdapat fitur rekomendasi untuk kategori makanan sesuai dengan kondisi kesehatan yang dipilih pelanggan. Gambar 3.2. (b) menunjukkan detail menu makanan berupa gambar makanan, kandungan yang terdapat pada makanan tersebut, deskripsi dan harga makanan, serta dapat membuat pesanan sesuai yang diinginkan. Sedangkan Gambar 3.2. (c) merupakan tampilan dimana pelanggan dapat melihat detail pesanan serta dapat melakukan checkout apabila pesanan sudah sesuai yang diinginkan.

## 3. Menu Pesanan untuk Pelanggan

Setelah memesan makanan, pelanggan dapat melihat status dan detail pesanan yang telah dibuat sebelumnya.

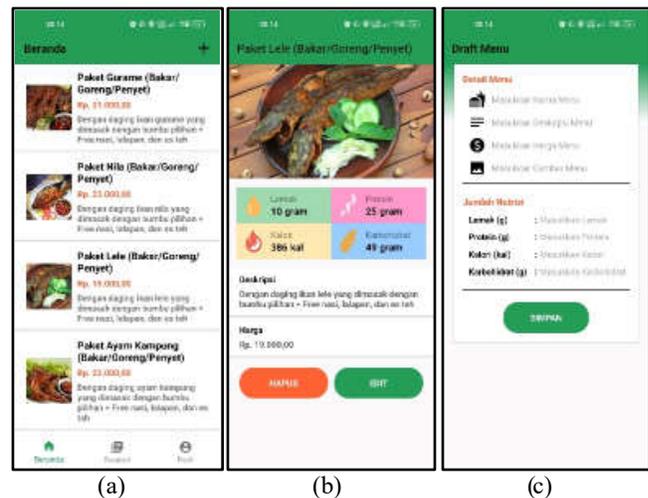


Gambar 3.3. (a) Tampilan Status Pesanan dan (b) Tampilan Detail Pesanan

Pada Gambar 3.3. (a) merupakan tampilan status pesanan untuk pelanggan yang terdapat 3 macam status yaitu diproses apabila pesanan sedang dibuat dan diantar ke lokasi tujuan, selesai apabila pesanan sudah diterima oleh pelanggan, serta dibatalkan apabila menu makanan kosong atau terdapat kendala lainnya. Sedangkan Gambar 3.3. (b) merupakan tampilan detail dari pesanan tersebut yang berisikan identitas dan lokasi pemesan, waktu pesanan, total harga, serta daftar menu yang telah dipesan.

## 4. Menu Utama untuk Admin

Pada Menu Utama admin dapat melihat daftar menu makanan, melihat kandungan yang terdapat pada makanan tersebut, serta dapat menambahkan atau mengedit menu makanan.



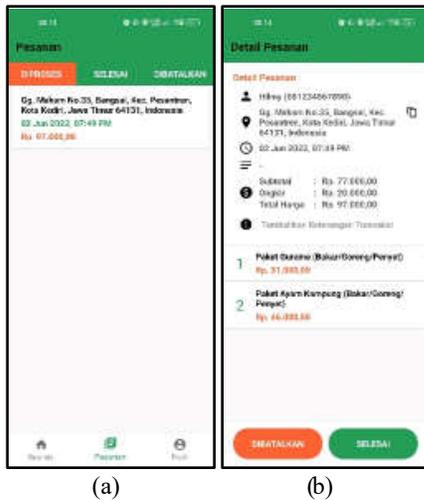
Gambar 3.4. (a) Tampilan Beranda, (b) Tampilan Detail Makanan, dan (c) Tampilan Tambah atau Edit Menu Makanan

Gambar 3.4. (a) merupakan tampilan beranda untuk admin yang berisikan daftar menu makanan. Gambar 3.4. (b) menunjukkan detail menu makanan berupa gambar makanan, kandungan yang terdapat pada makanan tersebut, deskripsi dan harga makanan, serta dapat menghapus atau mengedit menu makanan tersebut. Sedangkan Gambar 3.4. (c) merupakan tampilan dimana admin

dapat menambahkan atau mengedit menu makanan lalu menyimpannya ke database.

### 5. Menu Pesanan untuk Admin

Admin dapat melihat pesanan yang masuk, serta menerima dan membatalkan pesanan apabila menu makanan lagi kosong atau terdapat kendala lainnya.

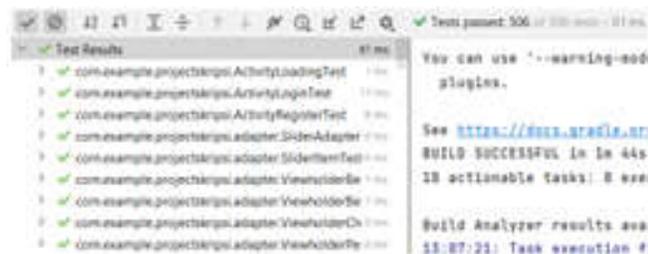


Gambar 3.5. (a) Tampilan Status Pesanan dan (b) Tampilan Detail Pesanan

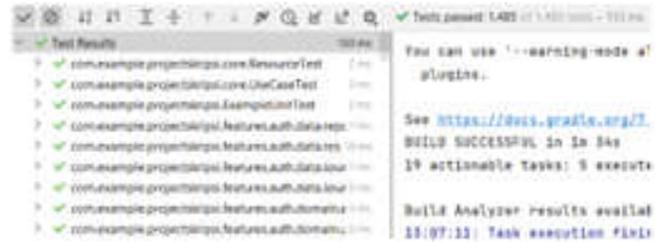
Pada Gambar 3.5. (a) merupakan tampilan status pesanan untuk admin yang terdapat tiga macam status yaitu diproses apabila terdapat pesanan masuk, selesai apabila pesanan sudah diterima oleh pelanggan, serta dibatalkan apabila menu makanan kosong atau terdapat kendala lainnya. Sedangkan Gambar 3.5. (b) merupakan tampilan detail dari pesanan tersebut yang berisikan identitas dan lokasi pemesan, waktu pesanan, total harga, daftar menu yang telah dipesan, serta admin dapat mengakhiri dan membatalkan pesanan tersebut.

### B. Pengujian Prinsip Clean Architecture

Pengujian pada Prinsip *Clean Architecture* ini menggunakan *IDE Android Studio* untuk menilai sebuah aplikasi dari segi *maintainable*. Pengujian ini dilakukan dengan membandingkan Prinsip *Clean Architecture* dengan arsitektur perangkat lunak lainnya, yaitu *Model-View-Controller (MVC)*. Berikut ini Gambar 3.6. dan Gambar 3.7. dari pengujian kedua arsitektur perangkat lunak tersebut.



Gambar 3.6. Hasil Pengujian menggunakan MVC



Gambar 3.7. Hasil Pengujian menggunakan Clean Architecture

Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 3.6. dan gambar 3.7., dihasilkan nilai dari beberapa perhitungan *Quality Factor (M)*. Hasil nilai perhitungan *Quality Factor (M)* tersebut meliputi *Conciseness*, *Consistency*, *Instrumentation*, *Modularity*, *Self-Documentation*, dan *Simplicity*. Berikut hasil nilai dari perhitungan *Quality Factor (M)* dapat dirangkum pada Tabel IV.

Tabel IV. Hasil Pengujian Tingkat Maintainability

Quality Factor (M)	Positive Test dan Test Case	Clean Architecture		MVC	
		Nilai	Hasil	Nilai	Hasil
Conciseness	Banyaknya Class	142	0.017	31	0.01
	LLOC	8535		2991	
Consistency	Fitur sesuai desain	19	1	19	1
	Fitur yang didesain	19		19	
Instrumentation	Fitur punya Instrumen	18	0.947	18	0.947
	Fitur yang seharusnya punya instrumen	19		19	
Modularity	Class yang loose coupling	111	0.782	17	0.548
	Banyaknya Class	142		31	
Self-Documentation	Class yang clean code	103	0.725	7	0.226
	Banyaknya Class	142		31	
Simplicity	Fungsi yang clean code	1421	0.957	502	0.992
	Banyaknya Fungsi	1485		506	

Hasil dari pengujian pada Tabel III membuktikan bahwa pengembangan aplikasi dengan menggunakan prinsip *Clean Architecture* mempunyai tingkat *maintainability* lebih besar daripada aplikasi yang dikembangkan menggunakan arsitektur lainnya, yaitu *Model-View-Controller (MVC)*, yaitu 0.782 dibanding 0.548 atau dalam persentase 78,2% dibanding 54,8%. Nilai *modularity* pada *pseudocode* yang dikembangkan lebih dari nilai ambang batas tingkat *maintainability* yaitu sebesar 67%, sehingga hal ini menunjukkan bahwa aplikasi berhasil menerapkan prinsip *Clean Architecture*. Sistem tersebut berhasil memisahkan antar layer pada *pseudocode*. Tiap-tiap layer berkomunikasi menggunakan struktur data sederhana sehingga tidak melanggar prinsip *dependency inversion*.

### C. Pengujian Metode Slope One Algorithm

Pengujian ini bertujuan untuk mengetahui menu makanan apa saja yang baik dikonsumsi oleh pengguna dengan kondisi Kesehatan tertentu seperti sehat, diabetes, jantung, kelelahan, obesitas, dan sembelit. Pengujian ini dilakukan dengan cara

menghitung *variable-variable* pada Tabel III menggunakan persamaan (1) dan (2) metode *Slope One Algorithm*. Hasil dari Pengujian terdapat tiga kategori makanan berdasarkan keadaan gizi menurut Indeks Antropometri, antara lain : Buruk dikonsumsi apabila mendapatkan nilai  $p^{SI} \leq 2.4$ , Kurang baik dikonsumsi apabila mendapatkan nilai  $2.4 < p^{SI} \leq 3.2$ , serta Baik dikonsumsi apabila mendapatkan nilai  $p^{SI} > 3.2$ . Berikut ini Tabel V adalah tabel nilai  $p^{SI}$  dari hasil pengujian Metode *Slope One Algorithm*.

Tabel V. Nilai  $p^{SI}$  dari Pengujian Metode *Slope One Algorithm*

Menu	Sehat	K1	K2	K3	K4	K5
Paket Ikan Gurami	4	3.125	4	3.125	3.125	3.125
Paket Ikan Nila	4	3.125	4	3.125	3.125	3.125
Paket Ikan Lele	4	4	4	4	4	4
Paket Ayam Kampung	4	3.125	4	3.125	3.125	3.125
Paket Ayam Negeri	4	3.125	4	3.125	3.125	3.125
Paket Tahu Tempe Penyet	4	4	4	4	4	4
Nasi Goreng	4	4	4	3.125	4	4
Nasi Goreng Seafood	4	4	3.125	1.375	4	4
Nasi Goreng Spesial	4	4	3.125	4	4	4
Mie Goreng	4	4	4	3.125	4	4
Mie Goreng Seafood	4	4	3.125	1.375	4	4
Mie Goreng Spesial	4	4	3.125	4	4	4

Dari hasil pengujian *Slope One Algorithm* pada Tabel IV, dapat diambil kesimpulan bahwa untuk terdapat beberapa menu Makanan yang sebaiknya dihindari oleh kondisi Kesehatan tertentu contohnya pada menu Nasi Goreng *Seafood* dan Mie Goreng *Seafood* yang mempunyai kategori Buruk dikonsumsi oleh seseorang yang mengalami Kelelahan karena mendapatkan nilai  $p^{SI}$  sebesar 1.375 atau dibawah 2.4 yang merupakan kategori Buruk dikonsumsi.

Hasil Pengujian tersebut selanjutnya dapat diimplementasikan kedalam *pseudocode* program aplikasi yang hasil tampilannya ditunjukkan pada Gambar 3.2. (a) Berikut ini *pseudocode* dari Metode *Slope One Algorithm* yang diterapkan dalam aplikasi penelitian ini.

```

//Slope One Algorithm Diabetes
val devK1 = (((lemakK1-lemakSehat)+
    (proteinK1-proteinSehat)+(kaloriK1-
    kaloriSehat)+(karbohidratK1-
    karbohidratSehat)).absoluteValue)
    .toDouble()/jumlahData.toDouble()
val pK1 = devK1+(lemakK1+proteinK1+kaloriK1+
    karbohidratK1).toDouble()
    
```

#### IV. KESIMPULAN

Dalam penelitian ini diuji tingkat *maintainability* dari kode yang dihasilkan dalam suatu aplikasi dengan menggunakan *IDE Android Studio*. Pengujian dilakukan dengan membandingkan antara aplikasi yang menerapkan prinsip *Clean Architecture* dengan aplikasi yang menerapkan arsitektur lainnya, yaitu *Model-View-Controller (MVC)*. Hasil dari pengujian ini menunjukkan bahwa pengembangan aplikasi dengan menerapkan prinsip *Clean Architecture* mempunyai tingkat *maintainability* lebih besar daripada pengembangan aplikasi dengan menerapkan arsitektur *Model-View-Controller (MVC)* yaitu 0.782 dibanding 0.548 atau dalam persentase 78,2% dibanding 54,8%, dimana 0.782 atau 78.2% dapat dikatakan *maintainability* karena mendapatkan nilai lebih dari 67%.

Selain pengujian tingkat *maintainability* dari kode yang dihasilkan dalam aplikasi tersebut, penelitian ini juga menguji Metode *Slope One Algorithm* untuk mengetahui rekomendasi menu makanan yang baik untuk beberapa kondisi kesehatan (seperti sehat, diabetes, jantung, kelelahan, obesitas, dan sembelit) menjadi tiga kategori makanan berdasarkan keadaan gizi menurut Indeks Antropometri, yaitu Buruk dikonsumsi apabila mendapatkan nilai  $p^{SI} \leq 2.4$ , Kurang baik dikonsumsi apabila mendapatkan nilai  $2.4 < p^{SI} \leq 3.2$ , serta Baik dikonsumsi apabila mendapatkan nilai  $p^{SI} > 3.2$ . Berdasarkan hasil pengujian Metode *Slope One Algorithm*, dapat diambil kesimpulan bahwa untuk terdapat beberapa menu Makanan yang sebaiknya dihindari oleh kondisi Kesehatan tertentu contohnya pada menu Nasi Goreng *Seafood* dan Mie Goreng *Seafood* yang mempunyai kategori Buruk dikonsumsi oleh seseorang yang mengalami Kelelahan karena mendapatkan nilai  $p^{SI}$  sebesar 1.375 atau kurang dari 2.4.

#### V. SARAN

Penelitian pada implementasi program aplikasi yang sudah dibuat masih bisa dikembangkan lagi menjadi aplikasi yang lebih kompleks, misalnya penambahan menu makanan dan tidak terbatas pada Warung Apung 2 Kota Kediri agar pengguna dapat memilih dan memesan lebih banyak varian makanan di tempat makan yang diinginkan, penambahan kondisi kesehatan agar pengguna yang mengalami kondisi kesehatan lainnya dapat juga mengetahui menu makanan yang baik dikonsumsi dirinya, serta meningkatkan desain tampilan aplikasi.

#### UCAPAN TERIMA KASIH

Penelitian ini dapat dilaksanakan dengan baik berkat adanya bantuan dari beberapa pihak. Oleh karena itu, peneliti menyampaikan ucapan terimakasih kepada pihak Warung Apung 2 Kota Kediri yang sudah memberikan kerjasama yang baik dalam penelitian ini, orang tua yang sudah memberikan dukungan dan fasilitas penunjang, serta dosen pembimbing yang dengan sabar membimbing penelitian ini hingga selesai.

#### DAFTAR PUSTAKA

[1] Rothan, H. A., dan Byrareddy, S. N. 2020. The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak *Journal of autoimmunity*. 109 (102433): 1-4.

- [2] Susilo, Adityo, dkk. 2020. Coronavirus disease 2019: Tinjauan literatur terkini. *Jurnal Penyakit Dalam Indonesia*. 7 (1) : 45-67.
- [3] Kemenkes RI. 2020. Pedoman Pencegahan dan Pengendalian COVID-19. Direktorat Jenderal Pencegahan dan Pengendalian Penyakit (P2P). Jakarta.
- [4] (2021) “StatCounter GlobalStats,” [Online], <https://gs.statcounter.com/os-market-share/mobile/indonesia/>, diakses pada tanggal 29 Maret 2021.
- [5] (2021) “AppBrain” [Online], <https://www.appbrain.com/stats/numberof-android-apps/>, diakses pada tanggal 29 Maret 2021.
- [6] Taqiu Sondha, Aflah, dkk. 2020. Framework dan Code Generator Pengembangan Aplikasi Android dengan Menerapkan Prinsip Clean Architecture. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*. 9 (4) : 327-335.
- [7] Malhotra, Ruchika, dan Chug, Anuradha. (2016). Software Maintainability: Systematic Literature Review and Current Trends. *International Journal of Software Engineering and Knowledge Engineering*. 26 (8) : 1221-1253.
- [8] Ivanics, Peter. 2017. An Introduction to Clean Software Architecture.
- [9] Bui Duy, Tung. 2017. Reactive Programming and Clean Architecture in Android Development.
- [10] Knill, Matthew. 2019. Refactoring To Clean Architecture.
- [11] Akhtar, Nayab dan Ghafoor, Sana. 2020. Analysis of Architectural Patterns for Android Development.
- [12] Neufeld, Lynnette M., dkk. 2021. A healthy diet. A definition for the United Nations Food System Summit 2021. *A paper for the Scientific Group of the UN Food System Summit*. 1-11.
- [13] <https://lifestyle.bisnis.com/read/20200508/106/1238079/berikut-kebutuhan-gizi-per-hari-yang-dibutuhkan-manusia>, diakses pada tanggal 10 Juni 2021.
- [14] Jiang, T. dan Lu, W. 2013. Improved Slope One Algorithm Based on Time Weight. *Applied Mechanics and Materials*. 347-350 : 2365-2368.
- [15] Pratama, Dharma, dan Hansun, Seng. 2017. Aplikasi Rekomendasi Tempat Makan Menggunakan Algoritma Slope One pada Platform Android. *IJCS*. 11 (1) : 11-20.
- [16] Fadillah, Arief. 2019. Pemanfaatan Algoritma Slope One Dengan Metode Collaborative Filtering Dalam Mengetahui Makanan Populer.
- [17] <http://www.ocw.upj.ac.id/files/Slide-INF106-INF106-Slide-05.pptx>, diakses pada tanggal 26 Mei 2022.
- [18] Koval, Andrii. 2021. Comparative Analysis of Modern iOS Architecture in Different Development Stages.
- [19] <https://www.fatsecret.co.id/kalori-gizi/>, diakses pada tanggal 10 Juni 2022.
- [20] <https://halosehat.com/gizi-nutrisi/panduan-gizi/>, diakses pada tanggal 10 Juni 2022.