**LAPORAN KKI**

**IMPLEMENTASI MVP (*MODEL VIEW PRESENTER*) MENGGUNAKAN RETROFIT & RX JAVA PADA APLIKASI MADHANG BERBASIS ANDROID**

Diajukan untuk memenuhi salah satu syarat mengerjakan dan menempuh ujian

KKI



Disususn oleh :

Nama : Bona Deny Suryana

NIM : A11.2014.08381

Program Studi : Teknik Informatika

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

SEMARANG

2018

# **PERSETUJUAN SKRIPSI**

Nama : Bona Deny Suryana

NIM : A11.2014.08381

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Judul Tugas Akhir : Implementasi MVP (*Model View Presenter*)

Menggunakan Retrofit & Rx Java Pada Aplikasi

Madhang Berbasis Android

Tugas Akhir ini telah diperiksa dan disetujui,

Semarang, ……. Februari 2018

Menyetujui,

Pembimbing

Dr. Pulung Nurtantio A., S.T, M.Kom

NPP. 0686.11.2006.331

Mengetahui,

Dekan Fakultas Ilmu Komputer

Dr. Drs. Abdul Syukur, MM

NPP.0686.11.1992.017

# **PENGESAHAN DEWAN PENGUJI**

Nama : Bona Deny Suryana

NIM : A11.2014.08381

Program Studi : Teknik Informatika - S1

Fakultas : Ilmu Komputer

Judul Tugas Akhir : IMPLEMENTASI MVP (*MODEL VIEW PRESENTER*)

MENGGUNAKAN RETROFIT & RX JAVA PADA

APLIKASI MADHANG BERBASIS ANDROID

Tugas Akhir ini telah diujikan dan dipertahankan di hadapan Dewan Penguji pada Sidang tugas akhir pada ….. Februari 2018. Menurut pandangan kami, tugas akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugerahan gelar Sarjana Komputer (S.Kom).

Semarang, …… Februari 2018

Dewan Penguji :

Penguji 1

Nama Dosen

NPP.

Penguji 2

Nama Dosen

NPP

Ketua Penguji

Nama Dosen

NPP.

# **PERNYATAAN KEASLIAN SKRIPSI**

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan dibawah ini, saya :

Nama : Bona Deny Suryana

NIM : A11.2014.08381

Menyatakan bahwa karya ilmiah saya yang berjudul :

IMPLEMENTASI MVP (*MODEL VIEW PRESENTER*) MENGGUNAKAN RETROFIT & RX JAVA PADA APLIKASI MADHANG BERBASIS ANDROID

Merupakan karya asli saya (kecuali cuplikan dan ringkasan yang masing – masing telah saya jelaskan sumbernya dan perangkat pendukung. Apabila dikemudian hari, karya saya disinyalir bukan merupakan karya asli saya, dan disertai dengan bukti – bukti yang cukup, maka saya bersedia untuk dibatalkan gelar saya beserta hak dan kewajiban yang melekat pada gelar tersebut. Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada tanggal : …… Februari 2018

Yang menyatakan

(Bona Deny Suryana)

# **PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Universitas Dian Nuswantoro, yang bertanda tangan dibawah ini, saya :

Nama : Bona Deny Suryana

NIM : A11.2014.08381

Demi mengembangkan Ilmu Pengetahuan, menyetujui untuk memberikan kepada Universitas Dian Nuswantoro Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

**IMPLEMENTASI MVP (*MODEL VIEW PRESENTING*) MENGGUNAKAN RETROFIT & RX JAVA PADA APLIKASI MADHANG BERBASIS ANDROID**

Berserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Universitas Dian Nuswantoro berhak untuk menyimpan, meng-*copy*, ulang (memperbanyak), menggunakaan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya dan menampilkan/ mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Dian Nuswantoro, segala bentuk tuntutan hokum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian surat pernyataan saya ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada tanggal : ….. Februari 2018

Yang menyatakan

(Bona Deny Suryana)

# **UCAPAN TERIMA KASIH**

Alhamdulilah, puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya. Sholawat dan salam kepada Rasulullah Muhammad SAW, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul “IMPLEMENTASI MVP (*MODEL VIEW PRESENTER*) MENGGUNAKAN RETROFIT & RX JAVA” dengan baik tanpa suatu halanan yang berarti. Tugas Akhir ini disusun untuk memenuhi syarat kelulusan akademik.

Penulis menyadari bahwa tanpa bimbingan, bantuan, dan doa dari berbagai pihak, Tugas Akhir ini tidak akan dapat diselesaikan tepat pada waktunya. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar – besarnya kepada semua pihak yang telah membantu dalam proses penyusunan Tugas Akhir ini, yaitu kepada :

1. Allah SWT atas kehendak-Nya penulis dapat melaksanakan dan menyelesaikan Tugas Akhir ini.
2. Prof. Dr. Ir. Edi Noersasongko, M.Kom selaku Rektor Universitas Dian Nuswantoro.
3. Dr. Drs. Abdul Syukur, MM selaku Dekan Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
4. Heru Agus Santoso, Ph.D. selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
5. Dr. Pulung Nurtantio Andono, S.T, M.Kom selaku dosen pembimbing dan akademik yang dengan sabra selalu memberikan arahan, bimbingan dan motivasi dalam penyusunan Tugas Akhir ini.
6. Dosen pengampu Teknik Informatika Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang yang telah memberikan ilmu dan pengalamannya, sehingga penulis dapat mengimplementasikan ilmu yang telah disampaikan.
7. Kedua Orang Tua yang selalu saya cintai dan banggakan.
8. Teman – teman DOSCOM (Dinus Opensource Community).
9. Semua pihak yang namanya tidak dapat disebutkan satu per satu.

Akhir kata, penulis ingin menyampaikan bahwa penyusunan Tugas Akhir ini masih sangat jauh dari kesempurnaan. Oleh karena itu, penulis sangat mengharapkan berbagai masukan dari semua pihak, baik berupa saran maupun kritik yang sekiranya bias memperbaiki kekurangan – kekurangan yang ada dalam Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak. Aamiin.

Semarang, …. Februari 2018

Penulis

# **ABSTRAK**

Mobilitas sudah menjadi hal utama di era yang serba cepat ini, baik masalah makan, minum hingga pekerjaan sekalipun dituntut untuk cepat. Perangkat mobile kini menjadi kebutuhan pokok untuk setiap orang. Segala kegiatan mulai dari pekerjaan, sekolah, komunikasi, termasuk juga jual-beli dapat dilakukan melalui perangkat mobile atau smarthphone. Disi lain pengangguran semakin banyak, karena tidak banyak lowongan pekerjaan yang bisa dimasuki mengingat banyaknya pesaing, sehingga mengakibatkan sebagian wanita memilih untuk menjadi ibu rumah tangga. Untuk itu aplikasi madhang hadir dengan fasilitas yang dapat membantu ibu – ibu rumah tangga untuk meningkatkan perekonomian keluarga. Dalam pengembangan aplikasi madhang standarisasi dan dokumentasi adalah komponen paling penting dalam sebuah penggerjaan aplikasi. Dengan adanya standarisasi dan dokumentasi yang baik maka pengembangan aplikasi dapat diwariskan dan dipelajari oleh banyak orang, untuk diperbaiki dan dikembangkan lebih lanjut. Model View Presenter merupakan salah satu metode pola atau pattern pengembangan aplikasi yang popular digunakan dalam dunia pemrograman android. MVP merupakan model pattern pemrograman yang dihasilkan dari turunan MVC (Model View Controller). MVP lebih menjurus ke pemisahan antara view dengan logika pemrograman. Pemisahan dilakukan untuk mempermudah pengerjaan aplikasi baik dari efektifitas pengerjaan maupun testing sebelum di release. Selain itu pengerjaan aplikasi yang menggunakan standarisasi pemrograman dapat meminimalisir terjadinya bugs pada aplikasi.

**Kata kunci :** *model view presenter*, *rx java*, android, *retrofit*, *test driven development*

**DAFTAR ISI**

[**PERSETUJUAN SKRIPSI** ii](#_Toc507512268)

[**PENGESAHAN DEWAN PENGUJI** iii](#_Toc507512269)

[**PERNYATAAN KEASLIAN SKRIPSI** iv](#_Toc507512270)

[**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS** v](#_Toc507512271)

[**UCAPAN TERIMA KASIH** vi](#_Toc507512272)

[**ABSTRAK** viii](#_Toc507512273)

[**DAFTAR ISI** ix](#_Toc507512274)

[**DAFTAR TABEL** xiii](#_Toc507512275)

[**DAFTAR GAMBAR** xiv](#_Toc507512276)

[**BAB I** 1](#_Toc507512277)

[**PENDAHULUAN** 1](#_Toc507512278)

[**1.1** **Latarbelakang Masalah** 1](#_Toc507512279)

[**1.2** **Rumusan Masalah** 2](#_Toc507512280)

[**1.3** **Batasan Masalah** 2](#_Toc507512281)

[**1.4** **Tujuan Penelitian** 3](#_Toc507512282)

[**1.5** **Manfaat Penelitian** 3](#_Toc507512283)

[**1.5.1** **Manfaat Bagi Perusahaan** 3](#_Toc507512284)

[**1.5.2** **Manfaat Bagi Akademis** 3](#_Toc507512285)

[**1.6** **Manfaat Kuliah Kerja Industri** 3](#_Toc507512286)

[**1.6.1** **Manfaat yang didapat dari Kuliah Kerja Industri untuk Universitas Dian Nuswantoro yaitu :** 3](#_Toc507512287)

[**1.6.2** **Manfaat yang didapat dari Kuliah Kerja Industri untuk PT. Madhang Sak Indonesia yaitu :** 4](#_Toc507512288)

[**1.6.3** **Manfaat untuk mahasiswa yang melaksanakan Kuliah Kerja Industri di PT. Madhang Sak Indonesia yaitu :** 4](#_Toc507512289)

[**BAB II** 6](#_Toc507512290)

[**TINJAUAN PUSTAKA DAN LANDASAN TEORI** 6](#_Toc507512291)

[**2.1** **Tinjauan Objek Penelitian** 6](#_Toc507512292)

[**2.2** **State Of The Art** 8](#_Toc507512293)

[**2.3** **Landasan Teori** 10](#_Toc507512294)

[**2.3.1** **Android** 10](#_Toc507512295)

[**2.3.2** **Java** 11](#_Toc507512296)

[**2.3.3** **MVP (Model View Presenter)** 12](#_Toc507512297)

[**2.3.4** **ReactiveX Android (Rx Java)** 13](#_Toc507512298)

[**2.3.5** **Retrofit** 13](#_Toc507512299)

[**2.3.6** **JSON (JavaScript Object Notation)** 14](#_Toc507512300)

[**2.3.7** **Firebase** 15](#_Toc507512301)

[**2.3.7.1** **Firebase User Authentification** 16](#_Toc507512302)

[**2.3.7.2** **Firebase Realtime Database** 17](#_Toc507512303)

[**2.4** **Deskripsi Tempat KKI** 17](#_Toc507512304)

[**2.5** **Logo dan Makna Tempat KKI** 18](#_Toc507512305)

[**2.6** **Struktur dan Job Deskripsi** 18](#_Toc507512306)

[**2.7** **Visi dan Misi Tempat KKI** 20](#_Toc507512307)

[**BAB III** 21](#_Toc507512308)

[**METODE PENELITIAN** 21](#_Toc507512309)

[**3.1** **Instrument Penelitian** 21](#_Toc507512310)

[**3.2** **Prosedur Pengambilan Data** 21](#_Toc507512311)

[**3.3** **Teknik Analisis Data** 21](#_Toc507512312)

[**3.4** **Metode** 22](#_Toc507512313)

[**3.4.1** **Pembuatan Test** 24](#_Toc507512314)

[**3.4.2** **Menjalankan Test** 27](#_Toc507512315)

[**3.4.3** **Penulisan Kode** 27](#_Toc507512316)

[**3.4.4** **Menjalankan Ulang Test** 28](#_Toc507512317)

[**3.4.5** **Refactoring** 28](#_Toc507512318)

[**3.5** **Pengujian** 28](#_Toc507512319)

[**BAB IV** 29](#_Toc507512320)

[**RANCANGAN SISTEM DAN IMPLEMENTASI** 29](#_Toc507512321)

[**4.1** **Gambaran Umum Sistem** 29](#_Toc507512322)

[**4.2** **Perancangan Kebutuhan Sistem** 29](#_Toc507512323)

[**4.2.1** **Kebutuhan Perangkat Lunak** 29](#_Toc507512324)

[**4.2.2** **Kebutuhan Perangkat Keras** 30](#_Toc507512325)

[**4.3** **Analisis Perancangan Sistem** 30](#_Toc507512326)

[**4.3.1** **Identifikasi Aktor** 30](#_Toc507512327)

[**4.3.2** ***Use Case Diagram*** 31](#_Toc507512328)

[**4.3.3** ***Activity Diagram*** 32](#_Toc507512329)

[**4.3.4** ***Sequensial Diagram*** 34](#_Toc507512330)

[**4.4** **Perancangan Antarmuka** 36](#_Toc507512331)

[**4.4.1** **Layout Halaman Login** 36](#_Toc507512332)

[**4.4.2** **Layout Halaman Registrasi** 37](#_Toc507512333)

[**4.4.3** **Layout Halaman Utama** 37](#_Toc507512334)

[**4.4.4** **Layout Halaman Detail Profil** 40](#_Toc507512335)

[**4.4.5** **Layout Halaman List Menu** 41](#_Toc507512336)

[**4.4.6** **Layout Tab Histori User** 41](#_Toc507512337)

[**4.4.7** **Layout Tab Chat User** 42](#_Toc507512338)

[**4.4.8** **Layout Tab Profile User** 43](#_Toc507512339)

[**4.4.9** **Layout Halaman List Pencarian Menu** 44](#_Toc507512340)

[**4.5** **Implementasi MVP** 44](#_Toc507512341)

[**4.5.1** **Struktur File MVP** 45](#_Toc507512342)

[**4.5.2** **MVP Halaman Login** 46](#_Toc507512343)

[**4.5.3** **MVP Halaman Form** 47](#_Toc507512344)

[**4.6** **Implementasi Rx Java** 47](#_Toc507512345)

[**4.6.1** **Rx Java Check Login** 47](#_Toc507512346)

[**4.6.2** **Rx Java Data User** 48](#_Toc507512347)

[**4.7** **Implementasi Retrofit** 49](#_Toc507512348)

[**4.7.1** **Retrofit Check Login** 49](#_Toc507512349)

[**4.7.2** **Retrofit Data User** 49](#_Toc507512350)

[**4.8** **Pengujian Aplikasi** 50](#_Toc507512351)

[**BAB V** 52](#_Toc507512352)

[**KESIMPULAN DAN SARAN** 52](#_Toc507512353)

[**5.1** **Kesimpulan** 52](#_Toc507512354)

[**5.2** **Saran** 52](#_Toc507512355)

[**DAFTAR PUSTAKA** 53](#_Toc507512356)

**DAFTAR TABEL**

[**Tabel 2. 1 Hasil Penelitian Bo Wu dan Xiaohu Yang 6**](#_Toc507512357)

[**Tabel 2. 2 Hasil Penelitian Artem Syromiatnikov dan Danny Weyns 8**](#_Toc507512358)

[**Tabel 2. 3 Tabel State Of The Art 8**](#_Toc507512359)

[**Tabel 4. 1 Identifikasi Aktor 30**](#_Toc507512360)

[**Tabel 4. 2 Tabel Tutorial Intro 38**](#_Toc507512361)

# **DAFTAR GAMBAR**

[**Gambar 2.1 Perbedaan Struktur MVP dan MVC 7**](#_Toc507512409)

[**Gambar 2.2 Skema Android 10**](#_Toc507512410)

[**Gambar 2.3 Alur Dan Aturan Model MVP 12**](#_Toc507512411)

[**Gambar 2.4 Contoh Struktur JSON (JavaScript Object Notation) 15**](#_Toc507512412)

[**Gambar 2.5 Logo PT Madhang Sak Indonesia 18**](#_Toc507512413)

[**Gambar 3.1 Alur Test Driven Development 22**](#_Toc507512416)

[**Gambar 3.2 Pembuatan Test 24**](#_Toc507512417)

[**Gambar 3.3 Pilih Method Test 25**](#_Toc507512418)

[**Gambar 3. 4 Pilih Structur Direktori Test 26**](#_Toc507512419)

[**Gambar 3.5 Hasil Generate Test 26**](#_Toc507512420)

[**Gambar 3.6 Contoh Test Success 27**](#_Toc507512421)

[**Gambar 3. 7 Contoh Test Gagal 27**](#_Toc507512422)

[**Gambar 4. 1 Use Case Diagram 31**](#_Toc507512423)

[**Gambar 4. 2 Activity Diagram Pembeli 32**](#_Toc507512424)

[**Gambar 4. 3 Activity Diagram Penjual 33**](#_Toc507512425)

[**Gambar 4. 4 Squensial Diagram Pembeli 34**](#_Toc507512426)

[**Gambar 4. 5 Squensial Diagram Penjual 35**](#_Toc507512427)

[**Gambar 4. 6 Halaman Login Aplikasi Madhang 36**](#_Toc507512428)

[**Gambar 4. 7 Halaman Form Data Diri 37**](#_Toc507512429)

[**Gambar 4. 8 Tab Halaman Utama 39**](#_Toc507512430)

[**Gambar 4. 9 Halaman Detail Profil 40**](#_Toc507512431)

[**Gambar 4. 10 Halaman List Menu 41**](#_Toc507512432)

[**Gambar 4. 11 Halaman Tab History 42**](#_Toc507512433)

[**Gambar 4. 12 Tab Chat User 43**](#_Toc507512434)

[**Gambar 4. 13 Tab Profil User 43**](#_Toc507512435)

[**Gambar 4. 14 Tab List Pencarian Menu 44**](#_Toc507512436)

[**Gambar 4. 15 Struktur File MVP (Model View Presenter) 45**](#_Toc507512437)

[**Gambar 4. 16 Potongan Source Code Rx Java Check Login 47**](#_Toc507512438)

[**Gambar 4. 17 Potongan Source Code Rx Java Data User 48**](#_Toc507512439)

[**Gambar 4. 18 Potongan Source Code Retrofit Check Login Pada File Java 49**](#_Toc507512440)

[**Gambar 4. 19 Potongan Source Code Abstraksi Retrofit Check Login 49**](#_Toc507512441)

[**Gambar 4. 20 Potongan Source Code Retrofit Data User Pada File Java 49**](#_Toc507512442)

[**Gambar 4. 21 Potongan Source Code Abstraksi Retrofit Data User 49**](#_Toc507512443)

[**Gambar 4. 22 Test Case Helper Facebook 50**](#_Toc507512444)

[**Gambar 4. 23 Hasil Test Case Helper Facebook 51**](#_Toc507512445)

# **BAB I**

# **PENDAHULUAN**

* 1. **Latarbelakang Masalah**

Di masa generasi Z saat ini, seluruh aktivitas manusia yang dilakukan sangat bergantung pada kemajuan teknologi. Segala sektor kehidupan modern manusia dilakukan secara digital, termasuk aktivitas jual beli makanan yang berbasis *marketplace*. Menurut Brunn, Jensen & Skovgaard [1], marketplace sendiri merupakan sebuah wadah komunitas bisnis interaktif secara elektronik yang menyediakan pasar dimana perusahaan dapat ambil andil dalam *B2B e-Commerce* dan atau kegiatan *e-Bussiness* lain. Di Indonesia sendiri baru sebagian kecil perusahaan berbasis aplikasi yang menyediakan layanan *marketplace* untuk menjual makanan ataupun masakan. Sebagai contoh Go-food & Grab Food, namun kedua aplikasi tersebut cenderung menyasar kepada usaha menengah keatas dan restoran.

Berdasarkan data hasil proyeksi penduduk oleh Badan Pusat Statistik (BPS, Populasi penduduk di Indonesia saat ini didominasi oleh kelompok usia produktif, yaitu usia antara 15-64 tahun) [2]. BPS sendiri membagi usia produktif kedalam dua kategori yaitu usia sangat produktif (15-49) tahun dan usia produktif (50-64) tahun. Berdasarkan hasil survey yang dilakukan oleh BPS perempuan usia sangat produktif (15-49) tahun diindonesia mencapai 69,4 Juta Jiwa, sedangkan untuk jumlah perempuan usia produktif (50-64) tahun mencapai 16,91 juta jiwa.

Menurut Deputi Bidang Statistik Sosial BPS M Sairi Hasbullah [3], tingkat pengangguran agak naik dari Agustus 2016 hingga Agustus 2017, hal tersebut terjadi dikarenakan banyaknya perempuan yang berhenti bekerja dan memilih menjadi ibu rumah tangga. Hal tersebut juga berdampak pada meningkatnya kemiskinan di Indoensia. Karena hal tersebut diindonesia muncul kelompok – kelompok usaha barus dibidang digital atau yang dikenal dengan startup mulai berlomba – lomba membuat aplikasi *marketplace* guna mendongkrak produktifitas dan meningkatkan perekonomian penduduk Indonesia terutama ibu rumah tangga. Salah satu perusahaan startup tersebut yaitu PT Madhang Sak Indonesia, dengan aplikasi madhang. PT Madhang Sak Indonesia merupakan startup yang bergerak dibidang kuliner atau jual beli masakan dan makanan, khususnya masakan ibu rumah tangga.

Menurut Paul Graham [4], “*Startup is a company designed to grow* fast”, sedangakan menurut Eric Ries [5], “*Startup is a human institution design that create something new under condition extream and* serenity”. Dari pendapat tersebut dapat disimpulkan bahwa startup adalah sebuah perusahaan yang belum lama terbentuk, dan biasanya masih dalam proses pengembangan. Untuk dapat bersaing perusahaan startup harus bergerak secara cepat dan terencana dengan jangka panjang. Dalam proses riset dan pengembangan banyak pembenahan dan perbaikan sistem aplikasi yang dilakukan agar dapat berjalan dengan baik. Untuk itu penulis menggunakan *pattern* MVP (Model View Presenter) dengan library Retrofit dan RxJava dalam membangun aplikasi madhang pada platform android.

* 1. **Rumusan Masalah**

Berdasarkan latarbelakang tersebut, rumusan masalah yang didapat peneliti dalam penelitian ini adalah bagaimana membangun aplikasi madhang agar mampu bertahan dengan perubahan yang cepat menggunakan *pattern* MVP (*Model View Presenter*) dengan menggunakan library Retrofit dan RxJava.

* 1. **Batasan Masalah**

Adapun batasan masalah yang didapat agar bahasan dan hasil yang didapat tidak menyimpang adalah sebagai berikut.

1. Perancangan aplikasi android madhang ini menggunakan *pattern* MVP (Model View Presenter)
2. Pengambilan data dari api menggunakan Retrofit dan Rx Java
3. Perancangan aplikasi menggunakan Bahasa Pemrograman Java
   1. **Tujuan Penelitian**

Membangun aplikasi madhang agar mampu bertahan dengan perubahan yang cepat menggunakan *pattern* MVP (Model View Presenter) dengan menggunakan library Retrofit dan RxJava.

* 1. **Manfaat Penelitian**

Penelitian ini diharapkan dapat bermanfaat untuk pihak – pihak terkait, diantaranya adalah sebagai berikut.

* + 1. **Manfaat Bagi Perusahaan**

1. Menghasilkan aplikasi yang mampu bertahan dengan perubahan – perubahan yang ada dari sisi *backend*.
2. Menjadikan aplikasi lebih *test-friendly* dan mudah untuk di *maintain*.
3. Dokumentasi yang dihasilkan lebih jelas dan mudah dipahami.
   * 1. **Manfaat Bagi Akademis**
4. Dapat dijadikan sebagai tolak ukur sejauh mana mahasiswa dapat menerima dan menerapkan materi yang disampaikan oleh dosen guna menyelesaikan sebuah permasalahan.
5. Dapat dijadikan sebagai acuan untuk pengembangan lebih lanjut oleh mahasiswa lainnya pada kasus yang sama.
   1. **Manfaat Kuliah Kerja Industri**
      1. **Manfaat yang didapat dari Kuliah Kerja Industri untuk Universitas Dian Nuswantoro yaitu :**
6. Univerisitas Dian Nuswantoro dapat menjalin hubungan kerja sama bilateral saling menguntungkan dengan PT. Madhang Sak Indonesia selaku industri start-up yang berbasis di Semarang.
7. Meningkatkan kualitas lulusan karena lulusan yang bersangkutan telah menempuh dan menyelesaikan proses pembelajaran dan pembuatan aplikasi yang digunakan pada dunia nyata dan telah di publikasikan oleh PT Madhang Sak Indonesia.
8. Universitas Dian Nuswantoro dapat lebih dikenal dalam dunia industri informatika dan start-up, dikarenakan PT Madhang Sak Indonesia telah bekerjasama dengan perusahaan – perusahaan ternama seperti Grab dan PayTren.
   * 1. **Manfaat yang didapat dari Kuliah Kerja Industri untuk PT. Madhang Sak Indonesia yaitu :**
9. Membantu PT Madhang Sak Indonesia dalam pengembangan aplikasi dengan metode – metode yang aplikatif sehingga dapat mempermudah programmer untuk melakukan pengembangan lebih lanjut.
10. PT Madhang Sak Indonesia dapat lebih dikenal di lingkungan universitas melalui mahasiswa yang ditempatkan di perusahaan tersebut.
    * 1. **Manfaat untuk mahasiswa yang melaksanakan Kuliah Kerja Industri di PT. Madhang Sak Indonesia yaitu :**
11. Mahasiswa mempelajari dan mematangkan bahasa pemrograman Java yang digunakan pada platform berbasis android selama 24 minggu secara intensif.
12. Mahasiswa mendapatkan pengalaman baru dalam membangun aplikasi dari berbagai sisi pengembangan, mulai dari metode pengembangan yang digunakan (*pattern*) seperti MVC, MVP, MVVM, *dependency*, hingga optimasi aplikasi.
13. Selama proses pengerjaan dan pembelajaran, mahasiswa dibimbing oleh setidaknya satu penyelia atau leader project dari PT Madhang Sak Indonesia.
14. Mahasiswa mendapat kesempatan untuk mengimplementasikan *Model View Presenter*, Rx Java, dan Retrofit pada saat proses pembuatan aplikasi, sehingga mempercepat pengembangan dan mempermudah untuk dilakukan *maintenance* dikemudian hari.
15. Mahasiswa lebih dituntut pada praktik daripada teori, sehingga memiliki banyak waktu untuk menyelesaikan problem yang ada di lapangan secara langsung.
16. Mahasiswa bebas menggunakan *dependency* dan *library*  yang akan diterapkan pada apilkasi madhang untuk menyelesaikan masalah yang telah dilakuan analisa sebelumnya.
17. Mahasiswa memiliki kesempatan untuk bekerja di PT Madhang Sak Indonesia atau perusahaan yang sudah berkerja sama dengan PT Madhang Sak Indonesia.
18. Dapat membangun karakter dan mental mahasiswa untuk bekerja secara professional, karena dalam dunia industri terdapat beberapa aturan dan tanggungjawab yang harus dipenuhi.

**BAB II**

**TINJAUAN PUSTAKA DAN LANDASAN TEORI**

* 1. **Tinjauan Objek Penelitian**

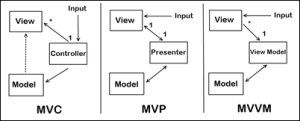
Pengembangan aplikasi dapat dilakukan tanpa menggunakan arsitektur seperti MVP (Model View Presenter). Aplikasi akan tetap berjalan dengan baik tanpa kendala apapun, namun dapat dibayangkan ketika aplikasi yang dikembangkan satu atau dua tahun yang lalu terjadi perubahan atau perbaikan pada beberapa komponen atau elemen, tentu hal tersebut akan menjadi masalah bila pengembang atau developer telah berganti. Untuk itulah standarisasi dengan menggunakan model arsitektur diperlukan. Ada beberapa model standarisasi yang popular digunakan dalam membangun sebuah aplikasi, diantaranya yaitu MVC (Model View Controller) dan MVP (Model View Presenter).

Menurut penelitian yang dilakukan oleh Bo Wu dan Xiaohu Yang dari Fakultas Ilmu Komputer dan Teknologi Universitas Zhejiang didapatkan bahwa MVC lebih optimal digunakan untuk platform web dimana terdapat banyak tampilan yang bias digunakan untuk berulang – ulang. Sedangkan untuk MVP baik digunakan untuk “Fat Client” dengan tampilan yang kompleks dan logika yang simple, lebih tepatnya baik untuk sistem seperti *financial system* [6].

Tabel 2. 1 Hasil Penelitian Bo Wu dan Xiaohu Yang

|  |  |
| --- | --- |
| Architecture | Better for using |
| MVC | Web Application multiple and frequently-redirecting screen. |
| MVP | “Fat-Client”, Win Form Application with complex screen logics but fewer screen redirecting. |

Perbedaan mendasar lainnya antara MVC (Model View Controller) dan MVP (Model View Presenter) yaitu terdapat rule pada MVP untuk Model tidak dapat berkomunikasi langsung dengan View, sehingga harus dilakukan dengan perantara atau contract berupa presenter. Dalam hal ini merupakan kelebihan MVP dibandingkan dengan MVC, dikarenakan state UI benar – benar terpisah. Hal tersebut memudahkan developer atau pengembang untuk memahami bagian antara logic dan view secara terpisah.

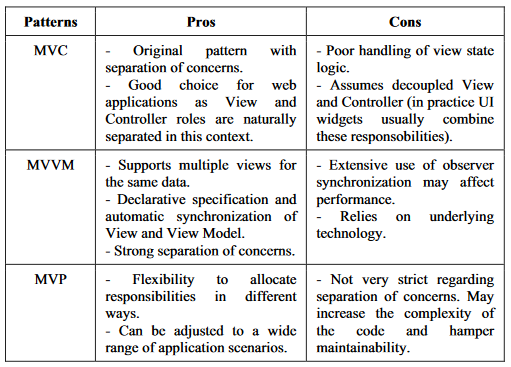


Gambar 2.1 Perbedaan Struktur MVP dan MVC

Menurut penelitian yang dilakukan oleh Artem Syromiatnikov dan Danny Weyns dari Fakultas Ilmu Komputer, Universitas Linnaeus, Vaxjo, Sweden. Bahwa setiap pola atau model arsitektur pembangunan sistem merupakan solusi untuk *reusable* atau penggunaan kembali yang dapat di singkronisasi antara *user interface* dengan domain data atau dalam hal ini adalah Model [7].

Berikut merupakan table hasil penelitian yang telah dilakukan oleh Artem Syromiatnikov dan Danny Weyns :

Tabel 2. 2 Hasil Penelitian Artem Syromiatnikov dan Danny Weyns



* 1. **State Of The Art**

Tabel 2. 3 Tabel State Of The Art

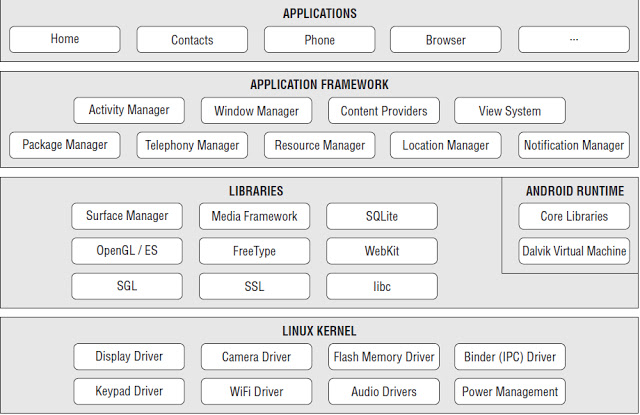
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Peneliti | Tahun | Judul | Masalah | Metode | Hasil |
| 1 | Bo Wu & Yang Xiaohu | 2009 | Application of MVP Architect  ure in Reengineering of  Legacy Financial System | Beberapa logika dideklarasikan secara berulang di beberapa file | MVP (Model View Presenter) dengan SubPresenter | Peningktan efisiensi pada proses test aplikasi |
| 2 | Yang Zhang & Yanjing Luo | 2010 | An  Architecture  and  Implement  Model  for  Model-View-Presenter  Pattern | Testing pada View dan Bussiness Logic | MVP (Model View Presenter) | Efisiensi dan mempermudah testing untuk View dan Bussiness Logic |
| 3 | Artem Syromiatnikov & Danny Weyns | 2014 | A Journey Through the Land of  Model-View-\* Design Patterns | Integrasi antara UI dengan domain aplikasi atau *model* | MVP (Model View Presenter), MVC (Model View Controller), MVVM (Model View View Model) | Tiap model memiliki ciri khas tersendiri |
| 4 | Redouane Esbai & Mohammed Erramdani | 2015 | Model-To-Model transformation in approach by modeling: | Proyeksi dari UML ke MVP (Model View Presenter) | MVP (Model View Presenter) | MVP berhasil diterapkan |

Kesimpulan :

Arsitektur MVP (Model View Presenter) telah di uji coba dan diterpakan pada berbagai kepentingan, yang membedakan penelitian ini dengan penelitian sebelumnya adalah pada implementasi penggunaannya. Pada penelitian ini aplikasi akan diterapkan pada aplikasi Madhang berbasis android untuk jual beli makanan secara online.

* 1. **Landasan Teori**
     1. **Android**

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet.



Gambar 2.2 Skema Android

Pada mulanya Android Inc merupakan pengembang android, Andrid Inc sendiri adalah sebuah perusahaan baru yang melakukan pengembangan *software* yang ditujukan untuk perangkat bergerak atau *mobile* dan kemudian dibeli oleh Google Inc. Pada saat proses pengembangan *Open Handset Alliance* (OHA dibentuk dari 34 perusahaan terkemuka software, hardware maupun *telecommunication* yang termasuk Qualcom, HTC, Motorola, Intel, serta Google. Untuk saat ini *Operating System* android sudah mencapai versi 8.0 yaitu *Oreo* dengan API 26, dan dikabarkan juga Android juga akan merilis update untuk Android P. Secara urutan android memiliki beberapa versi sebelumnya diantaranya yaitu android *Cupcake* (versi 1.5), Donut (versi 1.6), (versi 2.0/2.1) Éclair, Frozen Yogurt (versi 2.2) atau biasa disebut dengan Froyo, *Ginggerbread* (versi 2.3), veris 3.0/3.1/3.2 *Honeycomb*, *Ice Cream Sandwich* (versi 4.0), versi 4.1/4.2/4.3 *Jelly Beans*, *Kitkat* (versi 4.4), versi 5.0/5.1 *Lolipop*, *Marshmallow* (versi 6.0), dan versi 7.0/7.1 *Nougat*. Untuk pengembangan aplikasi android, Google mengeluarkan Android Studio dimana Android Studio merupakan resmi dari Google. Android Studio merupakan IDE yang dibuat atas kerjasama Google dengan JetBrains. Saat ini ada 2 bahasa yang didukung penuh oleh Google dalam pengembangan aplikasi android. Aplikasi android dapat dikembangkan menggunakan Bahasa pemrograman Java dan Kotlin (bahasa yang didukung oleh Android Studio). Namun ada bahasa lain yang juga dapat digunakan untuk membangun perangkat mobile Android yaitu C# dengan Xamarin, HTML5 dengan Intel XDK ataupun Ionic.

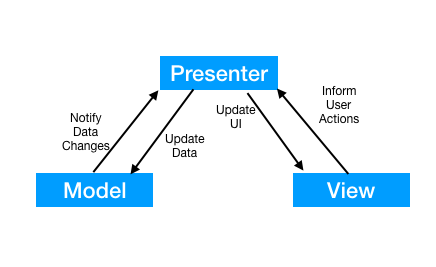
* + 1. **Java**

Java merupakan bahasa pemrograman yang dapat dikatakan memiliki multifungsi lebih dari bahasa pemrograman lainnya dengan bentuk yang lebih sederhana. Pada awalnya *sun Microsystem* melakukan pengembangan bahasa pemrograman Java pada tahun 1991 dengan nama Oak, dikarenakan nama Oak dinilai kurang menjual, maka nama tersebut diganti dengan nama Java pada Januari 1995. Java sendiri merupakan perpaduan dari bahasa pemrograman SmallTalk, C++, C, Object-C dan Common LISP yang dilengkapi juga dengan unsur keamanan.

Keunggulan terpenting dari Java sendiri yaitu perubahan paradigm pemrograman yang lebih sederhana, missal bahasa C++ atau C menggunakan pointer untuk pemanggilan ulangnya yang dirasa sulit, maka pada bahasa pemrograman Java telah ditinggalkan sehingga diperoleh kemudahan untuk penggunaannya [8]. Peran bahasa pemrograman Java sendiri sudah teruji di banyak media seperti penggunaan aplikasi client-server, *World Web Wide* (www) *web* *protocol*. Bahasa pemrograman Java memiliki sifat tidak bergantung pada *platform* atau portabilitas, yang artinya Java bias dijalankan di berbagai komputer maupun berbagai sistem operasi. Tingkat portabilitas atau platform tidak hanya ada di *source code* (kode sumber)., akan tetapi juga ada pada tingkat *bytecode* (binernya), sehingga kompilasi program Java dapat dijalankan pada perangkat yang berbeda tanpa harus melakukan kompilasi ulang *bytecode*. Java juga memiliki *interpreter* yang dapat dimengerti oleh semua *platform*.

* + 1. **MVP (Model View Presenter)**

MVP merupakan salah satu pattern arsitektur yang dapat digunakan untuk membangun sebuah sistem. MVP adalah hasil penurunan dari MVC (*Model View Controller*) dan paling sering digunakan untuk membangun aplikasi yang mengancalkan user interface (*frontend*). MVP (*Model View* Presenter) pertama kali muncul di IBM dan lebih terlihat di Taligent the 1990's [9]. Arsitektur ini sama seperti pola desain lainnya, yang memungkinkan beberapa pengembang bekerja dan menguji secara bersamaan.



Gambar 2.3 Alur Dan Aturan Model MVP

Agar aplikasi mudah dikembangkan dan pengembang yang mudah dipelihara perlu mendefinisikan lapisan yang terpisah dengan baik. Jadi dengan menggunakan MVP :

1. Tugas latar belakang (*logic, business model*) dipisahkan dari *activity* / *view* / *fragment* agar tidak terlepas dari sebagian dari *life-cycle*.
2. Tugas kompleks dipecah menjadi yang lebih sederhana dan mudah dipecahkan bila ada masalah (mudah untuk di*maintenance*).
3. Membuat *View* (*Activity, Fragment, View*) menjadi independen dari sumber.
4. Memfasilitasi pengujian unit / *unit test* secara otomoatis.
   * 1. **ReactiveX Android (Rx Java)**

Reactive Programming merupakan sebuah paradigm pemrograman yang berorientasi pada aliran data (*data flows*) dan penyebaran dari perubahan data [10]. RectiveX merupakan sebuah kamus / pustaka / *library* untuk menggabungkan *asynchronus* dan *event-base* program dengan menggunakan *observable sequences* [11]. Jadi Rx java merupakan sebuah *dependency* atau *library* yang digunakan untuk mengimplementasikan reactive programming (*asyncrhronus and event-base*) [12].

* + 1. **Retrofit**

Retrofit merupakan rest client library yang menggunakan komponen OkHttp untuk android yang di kembangkan oleh Squere. Library ini bekerja dengan baik untuk otentikasi dan berinteraksi dengan API. Retrofit adalah kelas di mana antarmuka API diubah menjadi objek yang dapat dipanggil. Secara default, retrofit akan memberi default memungkinkan untuk platform, namun memungkinkan penyesuaian.

* + 1. **JSON (JavaScript Object Notation)**

JSON (JavaScript Object Notation) merupakan format pertukaran data yang ringan, mudah dibaca, ditulis oleh manusia serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 Desember 1999 [13]. JSON memiliki beberapa kelebihan diantaranya sebagai berikut :

1. Mempunyai banyak tipe data.
2. Tipe Data JSON diantaranya : *number, string, boolean, array*.
3. Dapat diakses dengan mudah sebagai *object* JSON.
4. *Value* dapat diambil dengan mudah.
5. Semua browser telah *support* JSON.
6. API yang simple.
7. Di *JavaScript*, *object* cepat direalisasi.
8. *AJAX Toolkits support* dengan JSON.
9. Sepenuhnya otomatis *serialize object JavaScript*.

JSON merupakan format text yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan bahasa pemrogaman yang umum digunakan oleh keluarga programmer C, termasuk juga C++, C#, C, Objective C, Java, JavaScript, Python, Perl, dan lainnya. Maka dari itu sifat tersebut menjadikan JSON ideal sebagai bahasa pertukaran data.

JSON sendiri terbuat dari dua struktur, diantaranya sebagai berikut :

1. Kumpulan pasangan *name* dan *value*. Untuk beberapa bahasa, hal tersebut disebut sebagai *Object* (Objek), *record* (rekaman), *struct* (Struktur), *dictionary* (Kamus), *HashTable* (Tabel Hash), *Key List* (List Kunci), atau *assosiative array*.
2. Daftar nilai yang terurut (*an ordered list of value*). Untuk beberapa bahasa, hal tersebut dinyatakan sebagai *array* atau larik, *vector* (Vektor), *sequence* (urutan), dan juga *list* (daftar).

Struktur – struktur data tersebut disebut sebagai struktur data Universal. Semua bahasa pemrograman modern pada dasarnya mendukung struktur data tersebut dalam bentuk yang sama maupun bentuk yang berbeda. Hal itu pantas disebut demikian dikarenakan format data yang mudah ditukar dengan bahasa – bahasa pemrograman lain yang juga berdasar pada struktur tersebut [14]. Berikut merupakan contoh format JSON :



Gambar 2.4 Contoh Struktur JSON (JavaScript Object Notation)

* + 1. **Firebase**

Firebase adalah sebuah layanan infrastruktur *backend-as-a-service* (BaaS) yang diakuisisi oleh Google pada bulan oktober tahun 2014 lalu [15]. Firebase menawarkan kemudahan untuk para pengembang perangkat lunak dalam membangun aplikasi yang lebih baik serta mengembangkan bisnis yang sukses melalui seluruh fitur komplementernya. Untuk saat ini firebase sudah memiliki berbagai fitur, diantaranya seperti *firebase cloud messaging, firebase realtime database, firebase authentification, crash reporting, test lab, hosting, storage, notification, remote config, app indexing, dynamic link, invite, adword, admob* dan masih banyak lagi. Semua fitur tersebut dikemas dalam sebuah SDK (*Software Development Kit*) Firebase tunggal, sehingga dengan kemudahan yang ditawarkan para pengembang perangkat lunak dapat focus memecahkan masalah *customer* melalui perangkat lunak yang dibuatnya sehingga tidak menghabiskan banyak waktu dalam membangung infrastruktur yang kompleks [16]. Firebase merupakan sebuah platform yang disediakan oleh Google dan dapat digunakan untuk beberapa platform seperti Web dan Mobile (Android dan iOS).

* + - 1. **Firebase User Authentification**

Firebase User Authentification merupakan sebuah layanan *instant* atau siap pakai yang ada pada firebase SDK. Firebase User Authentification memungkinkan aplikasi untuk melakukan *user authentification* menggunakan berbagaimacam metode autentifikasi seperti *federate identity provider integration* (*authentifikasi* menggunakan Facebook, Google , Twitter maupun Github), *email and password based authentification*, *custom authentification system integration*, hingga *anonymouse* *authentification*. Firebase User Authentification berkerja dengan cara mengirimkan *server response* dari Firebase Server berdasarkan *cresidential* yang dikirimkan oleh *client* ke Firebase Server. *Cresidential* tersebut dapat berupa *email* dan *password* maupun juga dapat berupa *token* OAuth dari sebuah *federated identity provider*. Melalui *server response* yang diterima oleh Firebase Server, aplikasi dapat mengakses informasi dasar profile pengguna dan mengontrol akses pengguna terhadap produk atau layanan Firebase yang terdapat pada aplikasi.

* + - 1. **Firebase Realtime Database**

Firebase Realtime Database adalah database yang di-*host* di *cloud*. Data disimpan sebagai JSON (JavaScript Object Notation) dan disingkronkan secara realtime ke setiap klien yang terhubung, apabila ada perubahan pada data yang tersimpan, maka setiap user yang terhubung akan menerima *update* data secara otomati [17]. Format waktu yang digunakan pada Firebase adalah TimeStamp (*time since the Unix epoch*) dalam *miliseconds*.

* 1. **Deskripsi Tempat KKI**

PT Madhang Sak Indonesia merupakan sebuah startup atau perusahaan rintisan awal yang berfokusi pada teknologi informasi penjualan atau dengan katalain e-commerce yang berfokus pada bidang kuliner dengan pangsa pasar utamanya adalah ibu – ibu rumah tangga.

Informasi tentang PT Madhang Sak Indonesia :

Alamat : Komplek Universitas Dian Nuswantoro, Gedung D, Lantai ,

Jl. Nakula 1 Semarang, Jawa Tengah, Indonesia.

Telepon : +62811-2610-201

Email : [layanan@madhang.id](mailto:layanan@madhang.id)

Website : www.madhang.id / www.madhang.com

* 1. **Logo dan Makna Tempat KKI**

****

Gambar 2.5 Logo PT Madhang Sak Indonesia

Logo PT Madhang Sak Indonesia memiliki warna dasar orang, yang melambangkan keceriaan, dengan berbentuk bulat dan bergambar sendok dan garpu ditengahnya yang juga memiliki makna silaturahim melalui hidangan makan bersama, sesuai dengan visi dan misi perusahaan yaitu melestarikan masakan (resep) warisan keluarga dengan budaya silaturahim.

* 1. **Struktur dan Job Deskripsi**

Struktur organisasi pada PT Madhang Sak Indonesia

1. Chief

Bertanggung jawab dalam menetapkan visi misi perusahaan dan mengkomunikasikan kepada seluruh anggota organisasi. Bertanggung jawab terhadap seluruh aktivitas dalam perusahaan PT Madhang Sak Indonesia, menjamin segala kegiatan bisnis di PT Madhang Sak Indonesia bergerak efektif dan bersinergi. Menjadi jembatan antar divisi dan berperan sebagai *main decision maker*, memimpin dan memfasilitasi seluruh suara dalam organisasi.

1. Advisor

Bertanggung jawab untuk mengelola dan mengatur keuangan perusahaan dengan memberikan nasihat tentang berbagai masalah keuangan seperti investasi, asuransi, pajak, perencanaan, gaji karyawan dan masalah bisnis lainnya yang berhubungan dengan keuangan.

1. Lead Technology

Bertanggung jawab penuh atas segala kegiatan teknologin dan informasi yang ada dalam organisasi PT Madhang Sak Indonesia. Menjadi wakil direktur utama dalam pengambilan keputusan yang berkaitan dengan Teknologi Informasi yang ada dalam PT Madhang sak Indonesia.

1. Lead Marketing

Bertanggung jawab membantu direktur utama memimpin divisi marketing dan menangani berbagai perihal tentang marketing atau pemasaran didalam PT Madhang Sak Indonesia, serta membantu direktur utama dalam memimpin perusahaan dari dari sisi penjualan dan akuisisi customer / pelanggan.

1. Secretary

Bertanggung jawab membantu direktur utama untuk kepentingan PT Madhang Sak Indonesia yang berkaitan dengan administari surat menyurat, berkas kerjasama, MoU, dan kepentingan administrasi lainnya dalam PT Madhang Sak Indonesia.

1. Head Of Android

Bertanggung jawab kepada Lead Technology atas segala kegiatan pengembangan aplikasi madhang yang berbasis Android. Head Of Android memiliki peranan untuk melakukan riset, pengembangan, perbaikan, dan testing atas aplikasi madhang berbasis android yang dihasilkan di PT Madhang Sak Indonesia.

1. Head Of iOS

Bertanggung jawab kepada Lead Technology atas segala kegiatan pengembangan aplikasi madhang yang berbasis iOS. Head Of iOS memiliki peranan untuk melakukan riset, pengembangan, perbaikan, dan testing atas aplikasi madhang berbasis iOS yang dihasilkan di PT Madhang Sak Indonesia.

1. Designer / Creative Team

Memiliki peranan dan tanggung jawab atas segala konten publikasi dan kegiatan PT Madhang Sak Indonesia berupa acara. Kontent yang dimaksud meliputi ide – ide kreatif yang dituangkan dalam bentuk animasi maupun grafis yang kemudian di publikasi dalam akun social media dan website PT Madhang Sak Indonesia.

* 1. **Visi dan Misi Tempat KKI**

Visi : Di awali dengan memberitahu kepada semua orang bahwa kita adalah startup karya anak bangsa Indonesia. Kami ingin meningkatkan kesejahteraan taraf hidup keluarga. Melalui teknologi kami berusaha menyebarkan dampak sosial yaitu kehidupan yang lebih baik untuk ibu rumah tangga dan keluarganya dengan meningkatkan jumlah penghasilan mereka. Karena setiap ibu rumah tangga memiliki resep makanan yang khas dan berbeda. Dengan itu kita ingin mengubah resep keluarga menjadi nilai ekonomi yang lebih.

Misi :

* Mengubah resep keluarga mempunyai nilai ekonomi lebih
* Membantu meningkatkan taraf kehidupan keluarga..
* Mengangkat kearifan lokal untuk melestarikan masakan warisan dengan budaya silaturahmi

**BAB III**

**METODE PENELITIAN**

* 1. **Instrument Penelitian**

Penyusunan suatu proyek penelitian, memerlukan data – data yang sesuai dengan permasalahan pokok yang dihadapi. Data dapat dikatakan baik apabila data tersebut dapat mewakili objek yang sedang diteliti dan untuk mendapatkan data yang baik diperlukan metode atau serangkaian cara yang sesuai dengan kebutuhan penelitian

* 1. **Prosedur Pengambilan Data**

Untuk mendapatkan data yang benar, akurat dan relevan serta sesuai dengan sumber data dan tujuan penyusunan Laporan Kuliah Kerja Industri ini, maka penulis dalam pengumpulan data menggunakan beberapa teknik antara lain adalah sebagai berikut :

* Oberservasi

Teknik pengumpulan data dengan cara melakukan pengamatan secara langsung pada sistem pengembangan yang telah ada di dalam Tim Pengembang Aplikasi Madhang, PT Madhang Sak Indonesia.

* FGD (Focus Group Discussion)

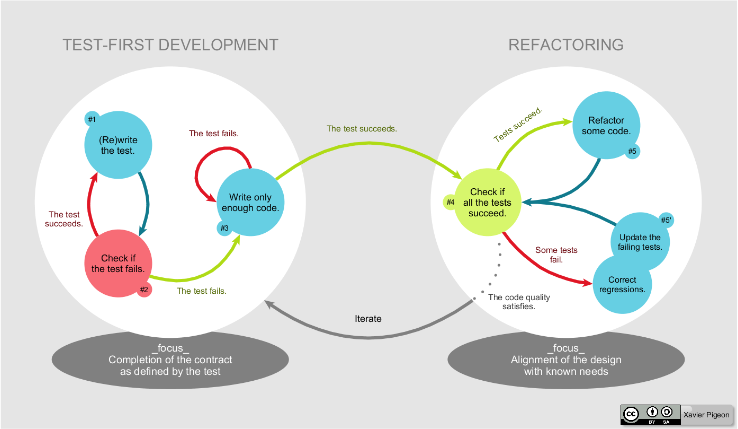
Teknik pengumpulan data dengan cara membuat sebuah forum group diskusi, yang berisikan para pengembang aplikasi. Peneliti membuka diskusi pada group chat tim android Madhang, guna mengetahui masalah yang sebenarnya dialami oleh tim pengembang.

* 1. **Teknik Analisis Data**

Penulis dalam penelitian ini menggunakan analisis pendekatan yang bersifat kuantitatif, dimana setelah semua data terkumpul kemudian data di analisa dengan teknik statistik. Data yang diperoleh dari FGD dan Observasi akan diolah sesuai dengan bagian – bagian dari sistem aplikasi madhang. Dari hasil analisa yang diapatkan kemudian dilakukan analisis masalah dengan pengembangan fitur yang cocok sebagai *problem solver* masalah tersebut, sehingga diaharpakn fitur baru yang dibuat akan mampu menyelesaikan masalah yang ada dan berfungsi dengan baik.

* 1. **Metode**

Dalam penelitian ini tim pengembang dari PT Madhang Sak Indonesia dan peneliti mengembangkan dengan metode pengembangan Extream Programming, yaitu Test Driven Development (TDD) atau yang sering disebut dengan Unit Test-first Development, yaitu dengan membuat test case sebelum membuat fungsional code. Berikut ini merupakan 5 tahapan atau fase dalam penerapan metode ini :



Gambar 3.1 Alur Test Driven Development

1. Pembuatan Test

Langkah pertama yang dilakukan yaitu dengan membuat *unit test* untuk setiap fitur yang akan dibuat. Dalam pembuatan *test* membutuhkan spesifikasi, deskripsi fitur yang jelas, hingga kebutuhan input sampai dengan output yang dihasilkan.

1. Menjalankan Test

Ketika akan menjalankan *test* pastikan *test* yang baru saja dibuat mengalami kegagalan atau tidak memenugi hasil yang diharapkan. Untuk tahapan ini sering disebut dengan *red-phase*.

1. Penulisan Kode

Penulisan kode program pada fase ini memiliki tujuan utama yaitu membuat program lulus dari *test*. Untuk bagian ini kode mungkin masih kurang elegan atau berantakan, hal tersebut dikarenakan untuk focus utama mengarah pada kelulusan *test* dan tidak memerlukan pembuatan fungsi tambahan diluar meluluskan *test*. Hal tersebut dikarenakan akan dilakukan *refactoring* pada tahapan kelima.

1. Menjalankan Ulang Test

Pastikan seluruh *test* yang dilakukan telah lolo, jika belum lolos, maka perlu dilakukan iterasi pada kode program hingga semua lolos *test*. Dengan menggunakan iterasi secara terus – menerus, *programmer* dapat dengan mudah memantau kinerja dari aplikasi sekaligus meminimalisir kesalahan kode setelah penambahan fitur baru.

1. Refactoring

Pada tahapan inilah kode menjadi berantakan atau kurang elegan, duplikasi, dan tidak efisien, dengan adanya iterasi *testing* juga sekaligus untuk memastikan perubahan yang dilakukan menghasilkan hal yang sama sebelum di *refactoring*. Seiring dengan perkembangan aplikasi kode yang banyak, Test Driven Development (TDD) membantu membuat kode menjadi lebih mudah dipelihara dan dibaca kembali oleh programmer, baik programmer tersebut maupun programmer lain. Karena setiap fungsi yang dibuat telah diberi deskripsi, contoh input dan ouput, sehingga kode tersebut dapat menjelaskan dengan sendirinya.

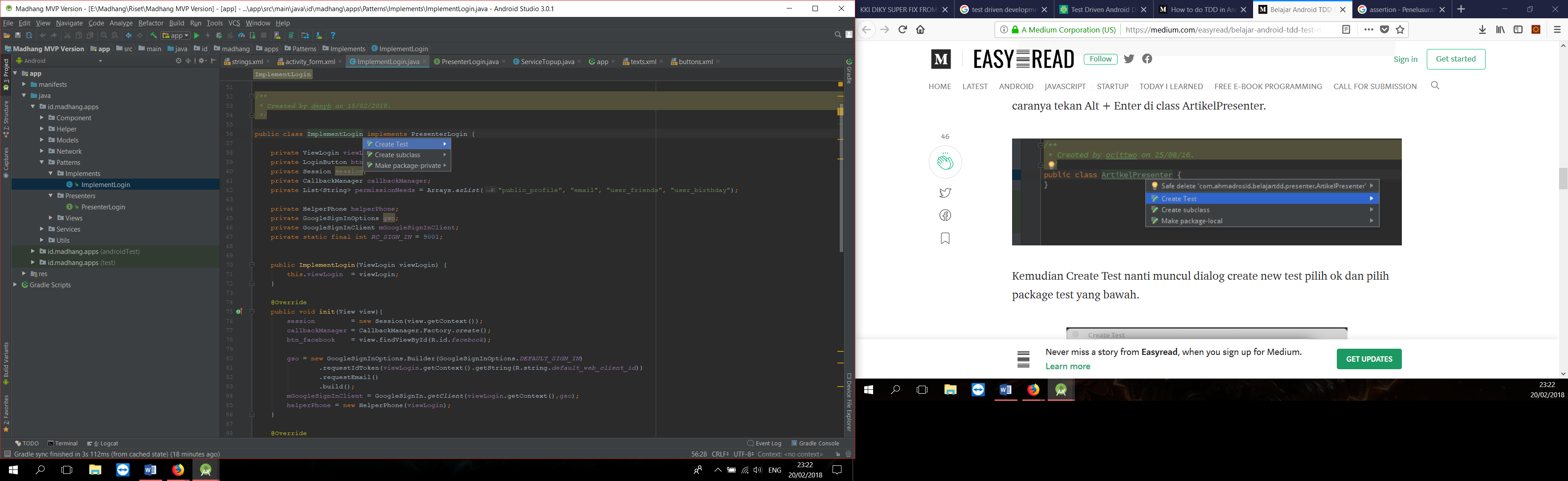
* + 1. **Pembuatan Test**

Sebelum melakukan pembuatan test, peneliti menganalisa setiap fitur yang akan dibuat pada aplikasi berdasarkan *use case* dan aktifitas diagram sehingga peneliti mengetahui alur setiap fitur dan beserta data yang diperlukan. Terdapat beberapa fitur utama yang wajib tersedia didalam aplikasi ini, diantaranya adalah sebagai berikut :

* Fitur *authentification* yang meliputi sub-fitur *login – register* dan *logout*.
* Fitur pembelian yang meliputi sub-fitur pencarian dan menampilkan makanan
* Fitur dapur yang meliputi sub-fitur membuat, menghapus dan menampilkan
* Fitur makanan / menu yang meliputi sub-fitur membuat, menghapus dan menampilka.
* Fitur chat yang meliputu sub-fitur membuat *room*, dan menampilkan.

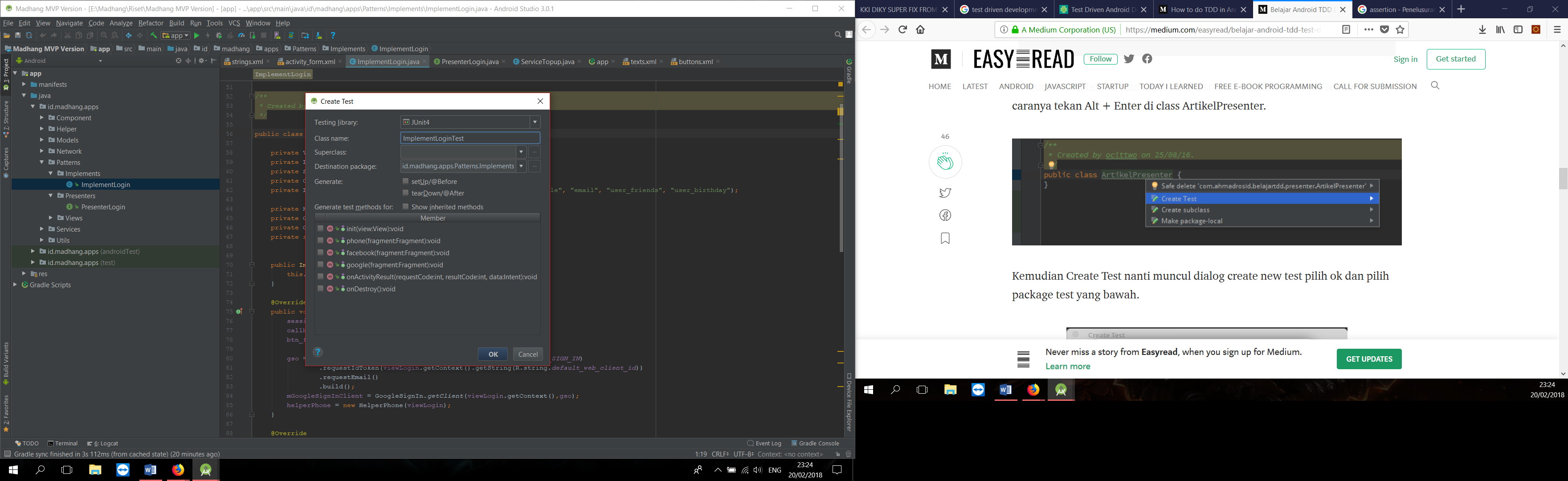
Setiap fitur yang dibuat penulis akan membuat sebuat *file*  yang digunakan untuk menampung seluruh *test* yang berkaitan dengan fitur tersebut, dengan ketentuan bahwa setiap sub-fitur akan dibuatkan setidaknya dua kondisi *test* yang mencakup positif *test* dan negative *test*, sehingga dapat dipastikan bahwa fungsi yang dibuat dapan menangani ketika seluruh parameter yang diperlukan benar dan dapat mengantisipasi jika parameter yang dikirim ada yang salah.

Berikut merupakan contoh pembuatan test :



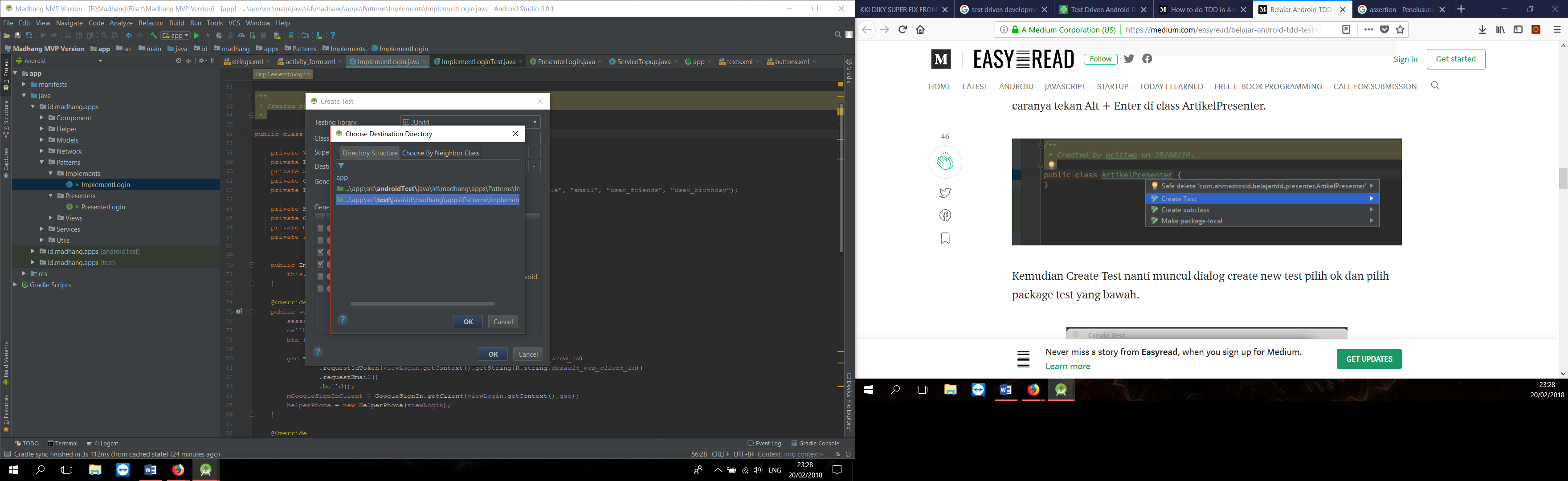
Gambar 3.2 Pembuatan Test

Pertama tentukan file *presenter* yang akan di test, kemudian tekan alt + enter lalu pilih create test pada fitur yang diinginkan.



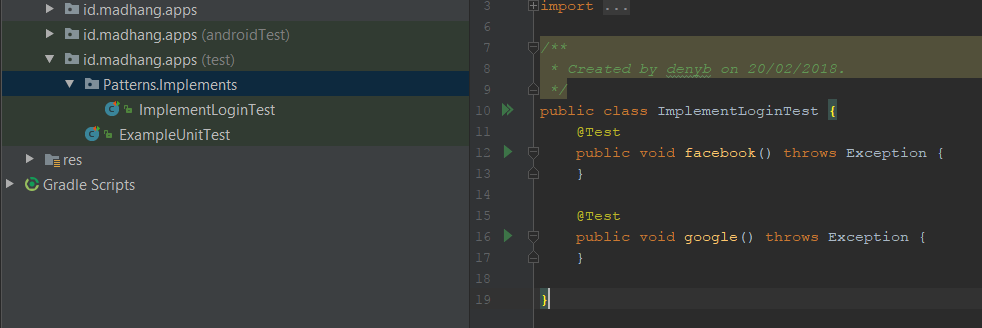
Gambar 3.3 Pilih Method Test

Selanjutnya akan muncul dialog *create test* pilih method yang akan ditest dan library yang akan digunakan, disini penulis menggunakan Junit4 sebagai library testing, lalu pilih ok.



Gambar 3. 4 Pilih Structur Direktori Test

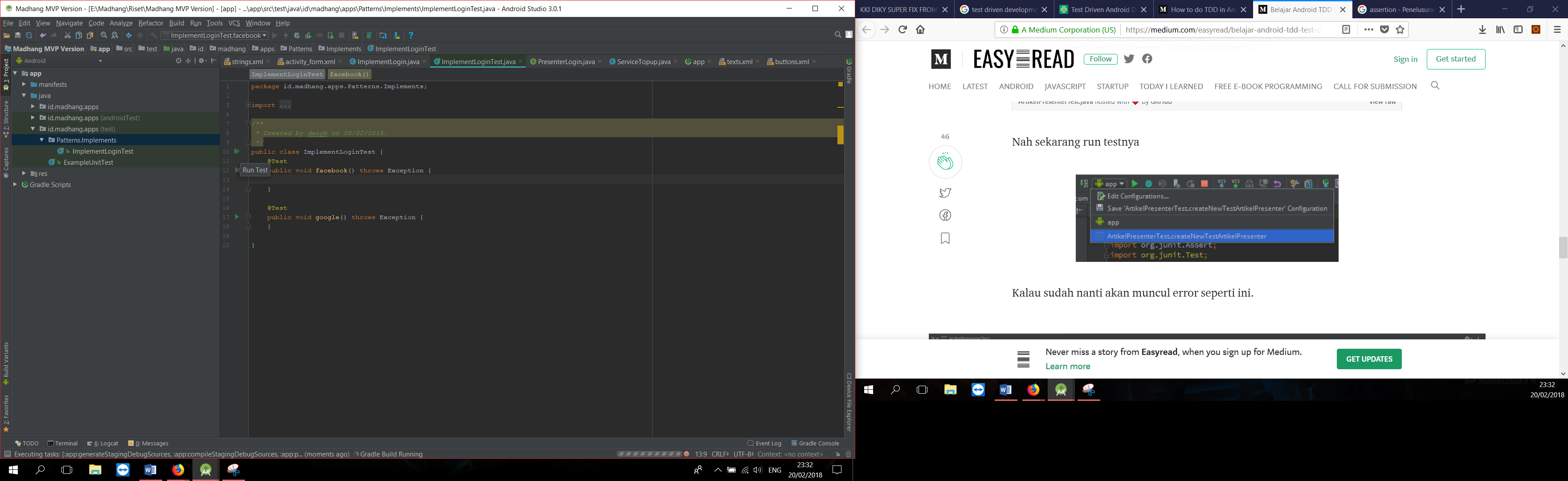
Selanjutnya pilih package test kemudian pilih ok lagi. Maka android studio akan meng-*generate* class test seperti berikut.

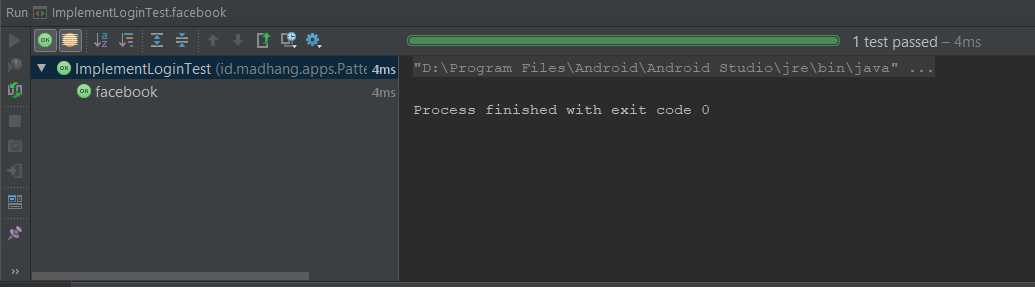


Gambar 3.5 Hasil Generate Test

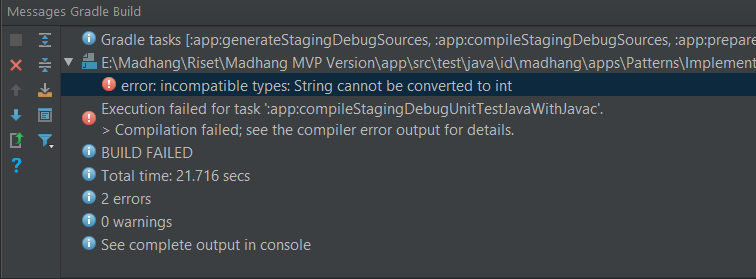
Setelah itu dapat dilakukan validasi test positif terlebih dahulu selanjutnya diikuti dengan test negative pada file baru yang telah digenerate.

* + 1. **Menjalankan Test**

Setiap kali selesai pembuatan *test case* untuk sebuah fitur, *test* runner akan berjalan sesuai dengan *file* yang akan dicek dengan menjalankan file test yang diinginkan dan menekan tombol tau icon run yang tersedia pada android studio.



Gambar 3.6 Contoh Test Success



Gambar 3. 7 Contoh Test Gagal

* + 1. **Penulisan Kode**

Penulisan kode aplikasi yang peneliti maksud pada fase ini adalahdengan membuat fungsi yang berfungsi untuk menangani problem atau bug yang didapat setelah dilakukan test sebelumnya. Setelah kode yang dibuat dirasa akan dapat memenuuhi *test case* maka barulah dilanutkan pada tahap selanjutnya.

* + 1. **Menjalankan Ulang Test**

Pada tahapan ini dilakukan dengan tujuan agar dapat memastikan bahwa kode yang ditulis pada tahapan sebelumnya dapat memenuhi persyaratan *assertion* pada *test case* yang diinginkan. Namun bila pada saat dijalankan nantinya masih belum lulu *test case*, maka perlu dilakukan penulisan ulang kode dan akan dilakukan iterasi terus pada tahapan ini hingga seluruhnya lulus *test case*.

* + 1. **Refactoring**

Untuk tahapan ini sebenarnya tidak harus dilakukan, dikarenakan aplikasi telah berjalan dengan semestinya dan telah diuji coba di tahap sebelumnya. Namun, akan lebih baik jika *refactoring*  tetap dilakukan agar kode mejadi lebih efisien dan rapi.

Setiap kali *refactoring* selesai dilakukan, maka akan dijalankan *test* ulang guna memastikan bahwa proses *refactoring* tidak merusak *function* yang telah memenuhi seluruh *test case* sebelumnya.

Jika pada tahapan ini membuat aplikasi yang lulus *test case*  menjadi tidak lulus, maka diperlukan iterasi ulang hingga seluruhnya memenuhi *test case* atau tahapan *refactoring* ini dapat dibatalkan atau tidak dilakukan.

* 1. **Pengujian**

Peneliti menerapkan metode Test Driven Development, dimana pengujian dibuat terlebih dahulu sebelum kode aplikasi ditulis. Sehingga, aplikasi yang dihasilkan telah teruji secara otomatis setelah penggerjaan selesai.

**BAB IV**

**RANCANGAN SISTEM DAN IMPLEMENTASI**

* 1. **Gambaran Umum Sistem**

Sistem yang akan dibuat merupakan aplikasi android Madhang. Aplikasi ini merupakan aplikasi yang jual-beli makanan via online dengan target khusus, yaitu memberdayakan ibu – ibu rumah tangga yang senantiasa ingin menjual masakannya ke khalayak umum. Aplikasi madhang dapat memfasilitasi UMKM menengah kebawah untuk memasarkan produk olahan makanan, minuman, jajanan, maupun resep khas keluarga yang diwariskan secara turun temurun. Platform android dipilih karena dapat memudahkan pengguna dalam melakukan transaksi, pencarian infokuliner yang selama ini belum diketahui. Mengingat pada era saat ini merupakan era mobile.

* 1. **Perancangan Kebutuhan Sistem**

Untuk membangun aplikasi ini, penulis membutuhkan beberapa komponen, diantaranya yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*).

* + 1. **Kebutuhan Perangkat Lunak**

Untuk spesifikasi kebutuhan *software* yang digunakan dibagi menjadi dua kategori, yaitu *development* dan *production*.

* Development
  + Sistem Operasi Windows 10 Pro 64 Bit
  + Android Studio Versi
  + Code Igniter (PHP Framework)
  + Postman
* Production
* CentOS
* Docker
* Cassandra DB
  + 1. **Kebutuhan Perangkat Keras**

Untuk spesifikasi kebutuhan *hardware* yang digunakan dibagi menjadi dua kategori, yaitu *development* dan *production*.

* Development
  + Intel Core i7
  + RAM 8GB
* Mobile
* Xiaomi Redmi Note 4X Prime
* RAM 4GB
* Processor Deca-core MediaTek Helio X20
* GPU Mali-T880
* Production
* Intel Xeon 2xE5-2620
* RAM 32 GB
* SSD 8TB
  1. **Analisis Perancangan Sistem**
     1. **Identifikasi Aktor**

Identifikasi aktor adalah pendeskripsian aktor yang terlibat pada sistem yang akan dibangun. Aktor yang terlibat dalam aplikasi ini terdiri atas dua aktor yaitu Mitra dan User. Berikut merupakan penjelasan dari tiap aktor yang terlibat dalam table 4.1.

Tabel 4. 1 Identifikasi Aktor

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Mitra | Mitra merupakan aktor yang dapat menjual atau menawarkan produk berupa makanan ataupun minuman di aplikasi madhang. |
| 2 | User | Merupakan aktor yang dapat melakukan transaksi pembelian makanan di aplikasi madhang. |

* + 1. ***Use Case Diagram***

****

Gambar 4. 1 Use Case Diagram

Pada use case diatas User (dalam hal ini adalah pembeli) dapat memilih makanan/minuman,melakukan pemesanan atau order, melakukan pembayaran dan memberikan review serta rating terhadap penjual dan makanan yang dipesannya. Disisi lain Mitra (dalam hal ini merupakan penjual) dapat melakukan posting makanan atau mengiklankan makanan/produknya, menerima informasi pesanan atau order, memberikan konfirmasi kepada pembeli yang telah melakukan order terhadap produknya, dan yang terakhir yaitu dapat mengambil pamasukan atau hasil pembayaran yang telah dilakukan oleh pembeli. Selain penjual dan pembeli, juga terdapat pemerain lainnya yaitu Server atau sistem madhang. Disini sisi server atau sistem madhang hanya menjembatani transaksi yang dilakukan antara penjual dan pembeli. Server melakukan pengiriman data informasi pesanan kepada penjual atau mitra pada saat pembeli melakukan pemesanan makanan kepada penjual dari aplikasi madhang. Pengiriman data yang dihasilkan berbentuk pesan notifikasi. Hal ini juga berlaku sebaliknya ketika penjual mengkonfirmasi pesanan pembeli.

* + 1. ***Activity Diagram***

1. Pembeli



Gambar 4. 2 Activity Diagram Pembeli

Diatas merupakan skema *activity diagram* dari sudut pembeli atau user yang menggunakan aplikasi madhang untuk melakukan transaksi order atau pembelian makanan hingga selesai.

1. Penjual / Mitra



Gambar 4. 3 Activity Diagram Penjual

Diata merupakan *activity diagram* yang menggambarkan peranan aplikasi madhang dari sudut penjual atau mitra. Mitra dapat melakukan login, sehingga sistem akan melakukan verivikasi terhadap akun penjual atau mita, jika gagal maka login akan diulangi atau masuk kembali kehalaman login, jika berhasil makan penjual/mitra akan langsung diarahkan ke halaman utama sehingga mitra dapat melakukan inputan makanan atau produk yang ingin dijual. Setelah penjual atau mitra melakukan inputan, sistem akan memproses data agar tersimpan didatabase, dan memvalidasinya, data yang tervalidasi akan langsung dipublish ke pilihan menu jual yang dapat diakses oleh pengguna atau pembeli.

* + 1. ***Sequensial Diagram***

1. Pembeli



Gambar 4. 4 Squensial Diagram Pembeli

Pada sequensial diagram diatas user dalam hal ini adalah pembeli melakukan login dengan memasukkan username & password, login yang dimaksud adalah login ketika user menggunakan login via facebook & google, pengguna akan diminta untuk memasukkan password dan username (email) dari akun facaebook ataupun google saat pertama kali login atau jika belum pernah ada aplikasi facebook atapun akun google pada smarthphone yang digunakan. Selanjutnya pembeli dapat memilih makanan lalu pembeli dapat memesannya, data pesanan kemudian akan diteruskan oleh sistem ke aplikasi madhang penjual, sehingga penjual tau makanan apa yang dipesan di tempatnya.

Penjual perlu melakukan konfirmasi pesanan, bila dirasa pesanan yang diminta dapat dipenuhi oleh penjual. Setelah penjual melakukan konfirmasi, data akan diteruskan oleh sistem ke aplikasi madhang pembeli, selanjutnya pembeli akan diminta untuk menyelesaikan proses order atau pemesanan dengan melakukan pembayaran.

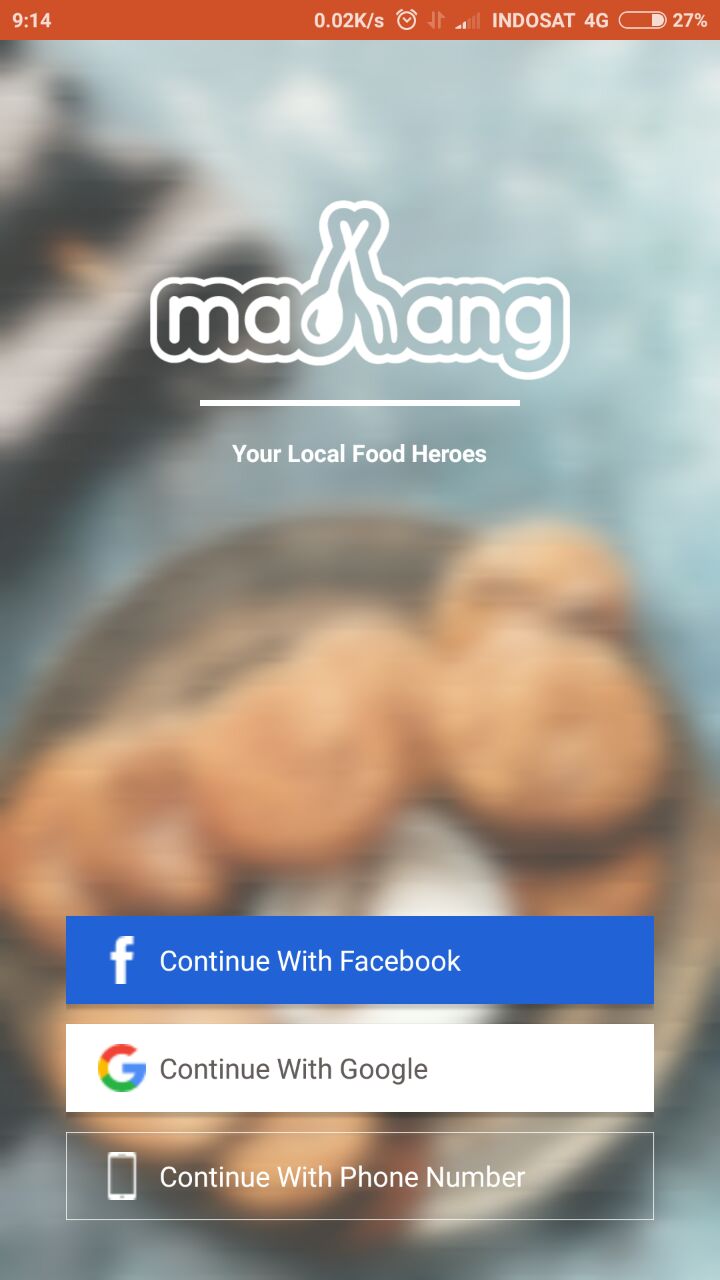
1. Penjual / Mitra



Gambar 4. 5 Squensial Diagram Penjual

Pada sequensial diagram diatas mitra dalam hal ini adalah penjual melakukan login dengan memasukkan username & password, login yang dimaksud adalah login ketika user menggunakan login via facebook & google, pengguna akan diminta untuk memasukkan password dan username (email) dari akun facaebook ataupun google saat pertama kali login atau jika belum pernah ada aplikasi facebook atapun akun google pada smarthphone yang digunakan. Selanjunya penjual dapat memasukan data makanan atau produk yang akan dijual, data tersebut kemudian akan diteruskan ke sistem, dan sistem akan menampilkan display makanan atau produk mitra kepada pembeli di fitur pembelian. Setelah data makanan atau produk telah terpublikasi maka penjual atau mitra akan mendapatkan notifikasi atau pemberitahuan bahwa produk makanan yang telah diinputkan telah terpublikasi. Selanjutanya proses yang ada pada sequensial diagram diatas sama seperti halnya sequensial diagram transaksi yang dilakukan oleh pembeli di tampilan sequensial diagram sebelumnya.

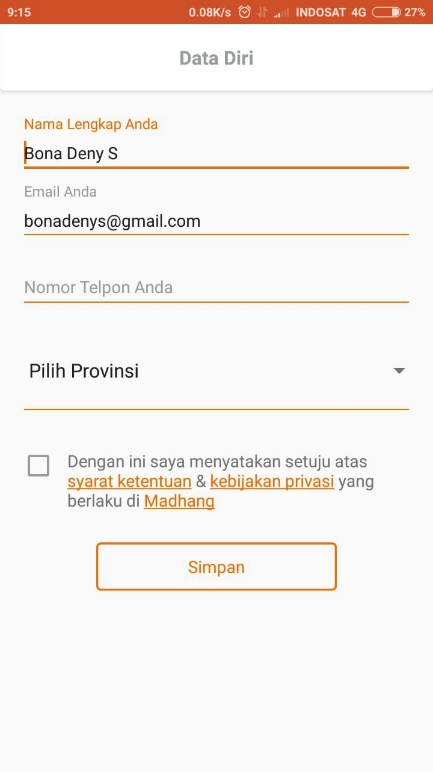
* 1. **Perancangan Antarmuka**
     1. **Layout Halaman Login**

****

Gambar 4. 6 Halaman Login Aplikasi Madhang

Aplikasi madhang menggunakan pihak ketiga seperti Facebook SDk, Facebook Akun Kit dan Google SDK untuk mendukung fitur login cepatnya, sehingga pengguna dapat dengan mudah mengintegrasaikan akun facebook, google maupun nomor telepon pengguna ke aplikasi Madhang.

* + 1. **Layout Halaman Registrasi**

****

Gambar 4. 7 Halaman Form Data Diri

Untuk login pertama kali pengguna akan disajikan halaman form registrasi untuk melengkapi informasi data diri pengguna. Namun untuk login setelahnya pengguna akan langsung menuju ke halaman utama.

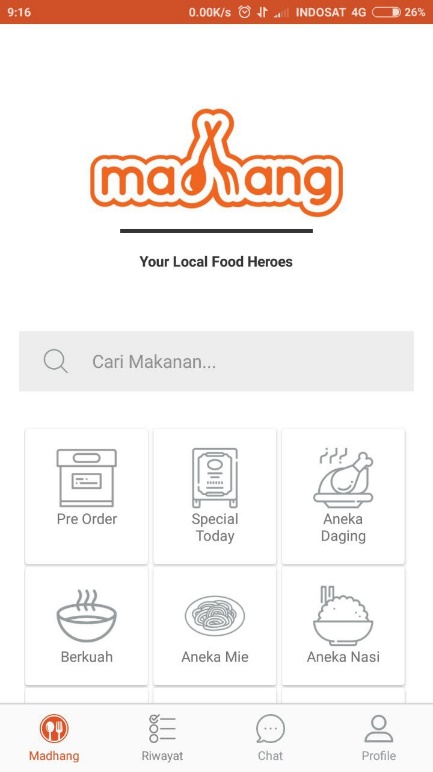
* + 1. **Layout Halaman Utama**

Berikut merupakan halaman utama yang ditampilkan setelah pengguna melakukan login dan registrasi untuk pertama kali. Untuk pengguna yang pertama kali menggunakan atau memasang aplikasi ini akan disajikan berupa tutorial singkat.

Tabel 4. 2 Tabel Tutorial Intro

|  |  |
| --- | --- |
| **Tutorial Intro Pada Mode User (Pembeli)** | |
| **E:\Kuliah\Laporan KKI\Layout\photo6080382830581491885.jpg** | **E:\Kuliah\Laporan KKI\Layout\photo6080382830581491884.jpg** |
| **E:\Kuliah\Laporan KKI\Layout\photo6080382830581491883.jpg** | **E:\Kuliah\Laporan KKI\Layout\photo6080382830581491882.jpg** |

Berikut merupakan tampilan halaman utama dengan beberapa fitur yang ada pada *bottom tab navigation* nya.



Gambar 4. 8 Tab Halaman Utama

Diatas merupakan layout yang terdapat pada tab Madhang. Pada tab ini user dapat memilih makanan berdasarkan beberapa kategori. Kategori yang disajikan dalam aplikasi madhang ini diantaranya yaitu :

1. PreOrder : merupakan kategori yang berisikan menu preorder
2. Special Today : merupakan kategori yang berisikan menu special

untuk hari saat user membuka aplikasi.

1. Aneka daging : merupakan kategori yang disediakan berisikan

tentang makanan olahan daging seperti ayam,

kambing dll.

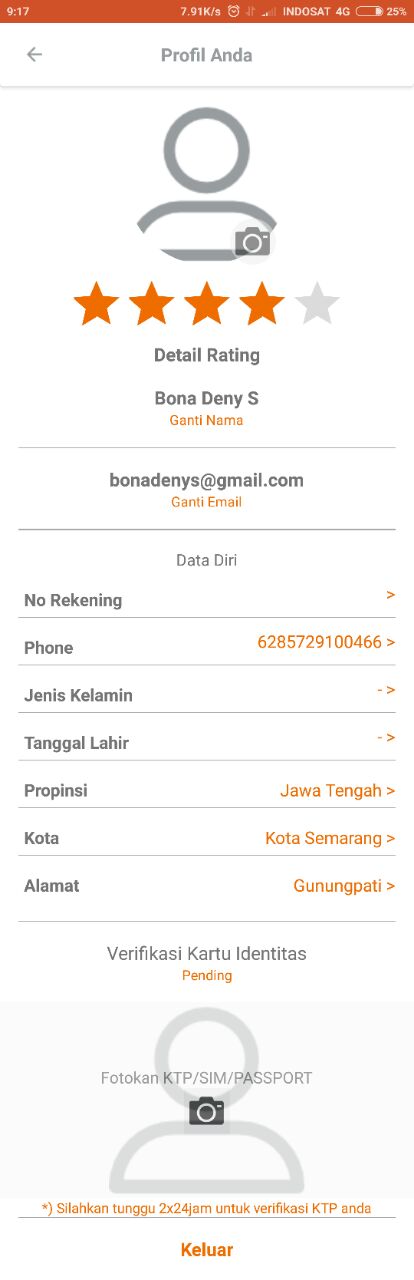
1. Berkuah : merupakan kategori yang berisi makanan berkuah.
2. Aneka Mie : merupakan kategori yang berisikan makanan

bertemakan Mie.

Selain kelima kategori diatas didalam aplikasi madhang juga terdapat beberapa kategori lainnya yang dapat dipilih oleh user sesuai selera.

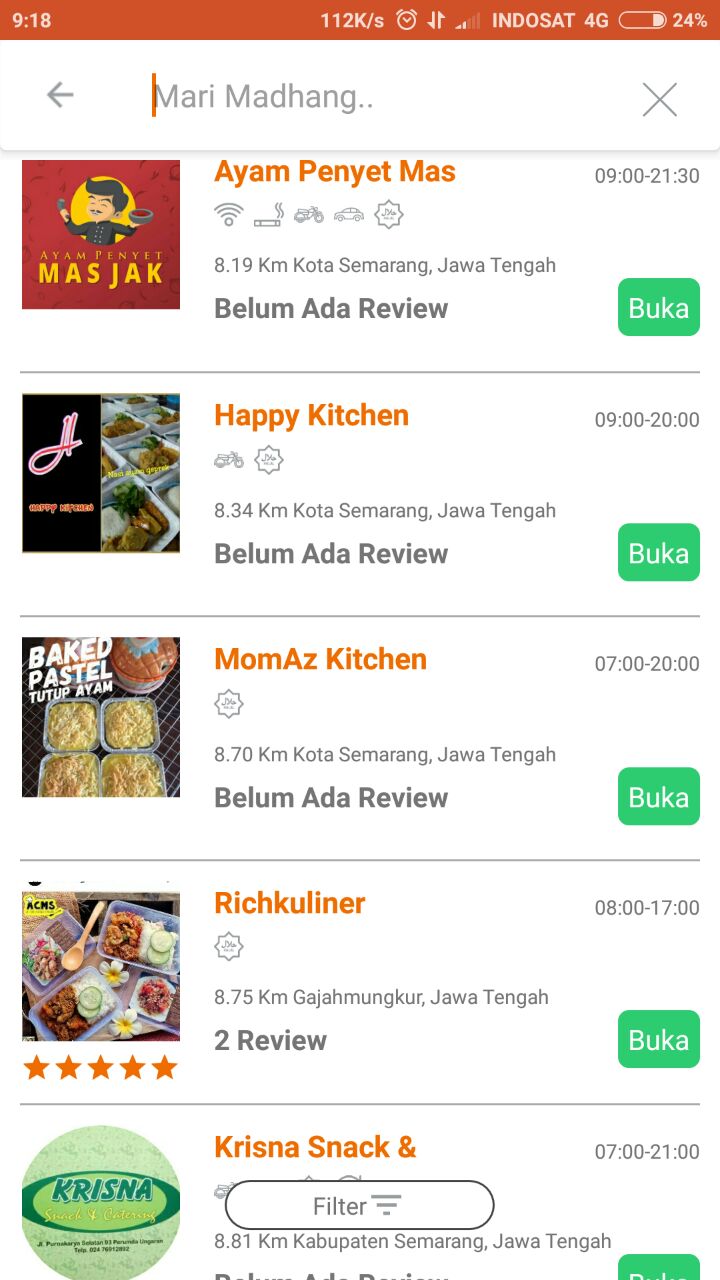
* + 1. **Layout Halaman Detail Profil**

Berikut merupakan halaman yang menyajikan mengenai informasi detai data diri mengenai pengguna.



Gambar 4. 9 Halaman Detail Profil

* + 1. **Layout Halaman List Menu**

****

Gambar 4. 10 Halaman List Menu

Pada tampilan ini pengguna atau dalam hal ini yaitu pembeli dapat memilih makanan baik memilih secara langsung maupun menggunakan fitur pencari makanan atau minuman yang dapat digunakan oleh pembeli dengan mengetikan kata atau nama makanan / minuman, nama dapur (nama dagang yang dipakai penjual) pada kolom pencarian.

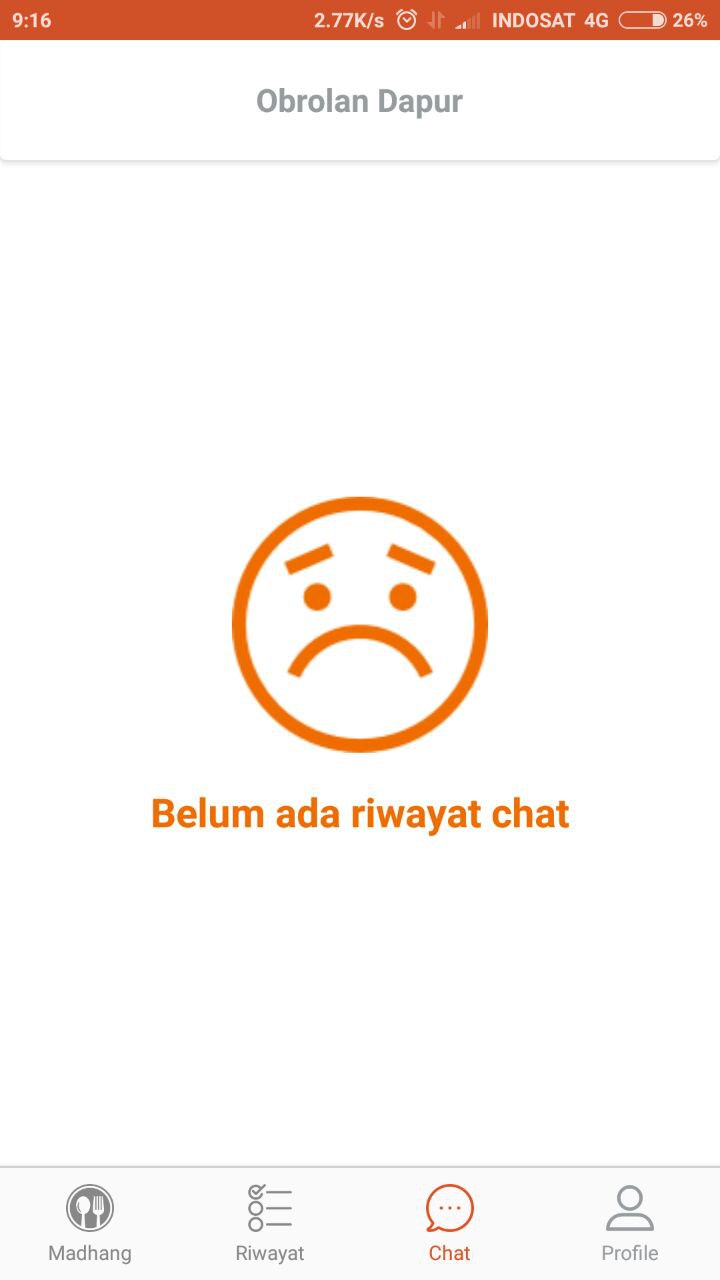
* + 1. **Layout Tab Histori User**

Berikut merupakan tampilan yang berfungsi untuk memberikan informasi kepada pengguna untuk dapat melihat informasi mengenai riwayat pembelian makanan yang telah dilakukan. Tampilan ini bias dikunjungi dengan menekan tab Histori yang ada pada bagian bawah *Bottom Navigation*. Pada fitur ini digunakan *fragment* sebagai pengelola tampilan yang dimasukkan kedalam sebuah *activity main* dan juga dikelola menggunakan *view pager*.

****

Gambar 4. 11 Halaman Tab History

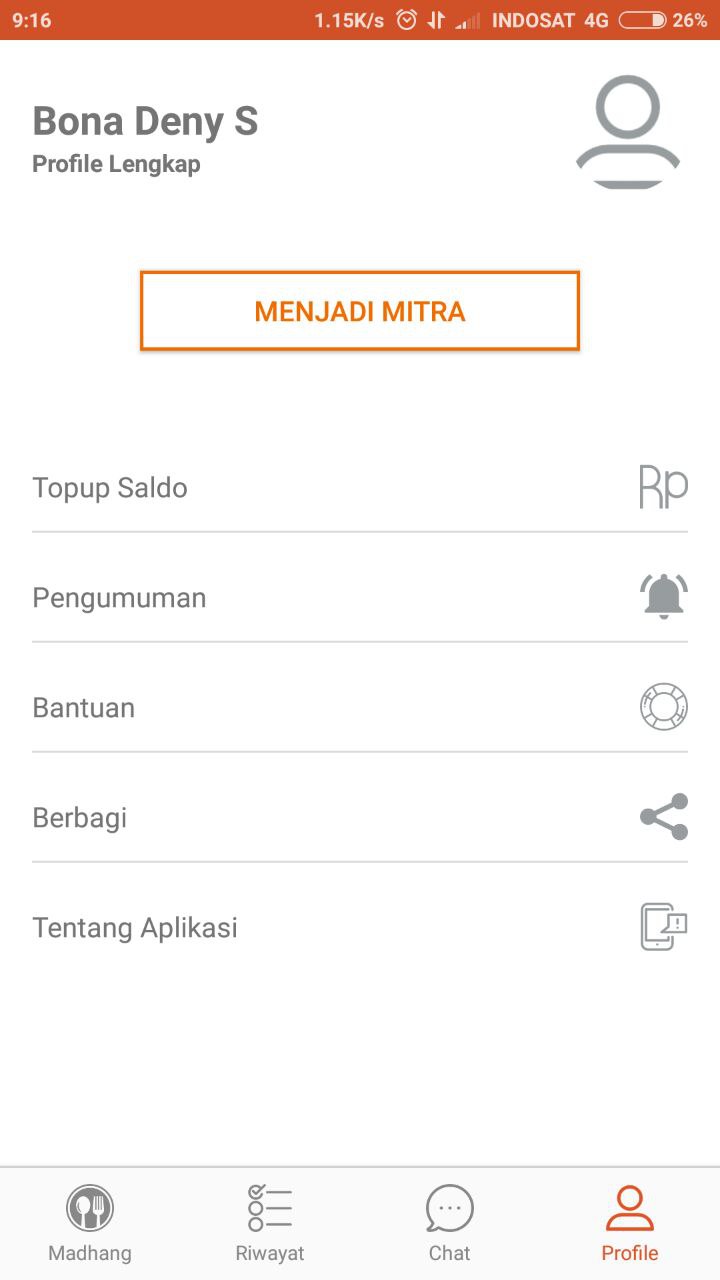
* + 1. **Layout Tab Chat User**

****

Gambar 4. 12 Tab Chat User

Pada tampilan ini berfungsi untuk memberikan informasi kepada pengguna untuk dapat melihat informasi mengenai riwayat pesan / obrolan antara pengguna dan mitra / penjual yang telah dilakukan.

* + 1. **Layout Tab Profile User**

****

Gambar 4. 13 Tab Profil User

Tampilan ini merupakan tampilan yang dapat dikelola oleh pengguna, yang berisikan pilihan menu – menu atau fitur seperti Topup Saldo, Pengumuman, Bantuan, dan menu Tentang Aplikasi.

* + 1. **Layout Halaman List Pencarian Menu**

****

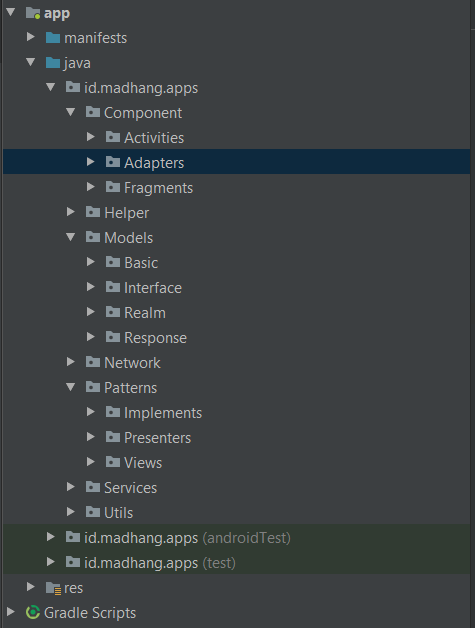
Gambar 4. 14 Tab List Pencarian Menu

Tampilan diatas merupakan tampilan hasil dari pencarian yang dilakukan oleh sistem. Pencarian tersebut didapat dari *trigger* atau inputan yang diberikan oleh pembeli atau pengguna pada kolom *box* pencarian.

* 1. **Implementasi MVP**

Penerapan struktur atau pola penulisan (*pattern*) program MVP (*Model View Presenter*) pada aplikasi madhang dilakukan di seluruh *source code* yang ada pada aplikasi ini. Namun dikarenakan aplikasi ini merupakan aplikasi komersil yang bersifat close source, penulis hanya dapat menampilkan sebagian potongan *source code* dan struktur file dari keseluruhan yang ada.

* + 1. **Struktur File MVP**



Gambar 4. 15 Struktur File MVP (Model View Presenter)

Gambar diatas merupakan struktur MVP (*Model View Presenter*) yang digunakan pada aplikasi Madhang. Model diatas memiliki struktur atau pola yang terdiri dari *implements package, presenters package, views*, *package* dimana masing – masing diantaranya terdapat file – file yang menunjang fitur – fitur aplikasi Madhang.

Tiap *pattern package* memiliki peran dan fungsi yang berbeda, berikut merupakan peran dari tiap *package* :

1. *Implements package*

*Implements package* memiliki peran untuk mengelola *logic* dan alur data dari aplikasi yang di kembangkan. Seluruh komunikasi baik antara *view* dengan *model* (komponen penunjang kelompok data) ataupun sebaliknya haruslah melalui file *implement* yang ada didalam *implements package*. Dengan kata lain *implement package* merupakan pusat seluruh kegiatan “dapur” aplikasi.

1. *Presenter package*

*Presenters package* disini memiliki peran sebagai tempat pembuatan  *method* yang nantinya akan digunakan dan diisi di dalam file *implement* yang terdapat di dalam *implements package* sebagai *method*  “@Override”.

1. *Views package*

*Views package* memiliki peran sebagai *method abstrak* penampil data yang dihasilkan dari pengelolaan di proses file *implement* yang ada di dalam *implements package* ke sebuah *activity*, maupun *fragment*.

* + 1. **MVP Halaman Login**

Pada halaman login telah diimplementasikan model MVP (*Model View Presenter*). MVP pada halaman login meliputi beberapa file pendukung diantaranya seperti :

1. *activity login* : merupakan tampilan aktivitas login
2. *fragment login* : merupakan fragment dari aktivitas login
3. *ImplementLogin* : berisikan logika – logika yang dibutuhkan saat

login.

1. *PresenterLogin* : berisikan abstrak yang dibutuhkan di implement

login.

1. *ViewFragments* : berisikan *abstract of view* dari fragment login.
2. *Helper Facebook* : merupakan file yang berisikan logika bantuan

untuk menunjang pengelolaan login via SDK *facebook*.

1. *Helper Google* : merupakan file yang berisikan logika bantuan

untuk menunjang pengelolaan login via SDK *Google*.

1. *Helper Phone* : merupakan file yang berisikan logika bantuan

untuk menunjang pengelolaan login via SDK *account kit facebook*.

* + 1. **MVP Halaman Form**

Pada halaman form, penerapan MVP (*Model View Presenter*) dilakukan hanya pada *activity form*, hal tersebut dilakukan karena pada halaman ini layout form langsung tertuju pada file *activity* dan tidak memerlukan fragment. MVP pada halaman form meliputi beberapa file pendukung diantaranya seperti :

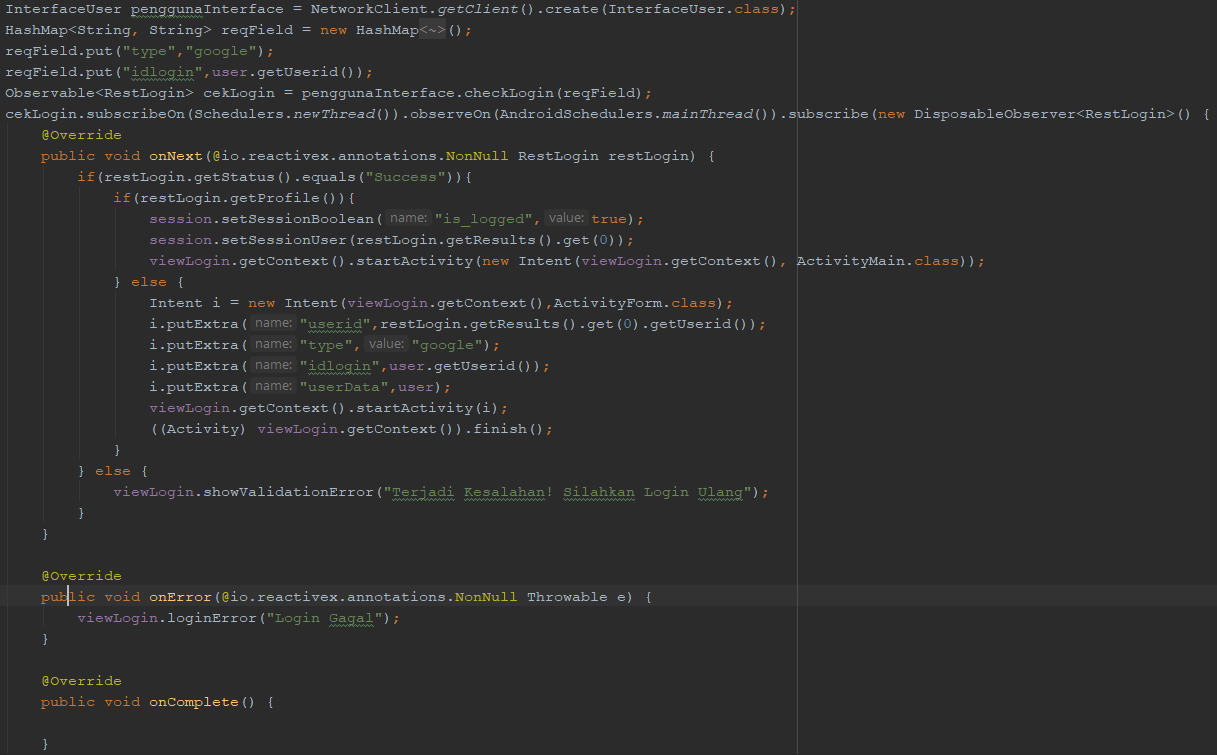
* + - 1. *activity form* : merupakan tampilan aktivitas form.
      2. *ImplementForm* : berisikan logika – logika yang dibutuhkan saat

Pengisian form.

* + - 1. *PresenterForm* : berisikan abstrak yang dibutuhkan di implement

form.

* + - 1. *ViewActivities* : berisikan *abstract of view* dari activity form.
  1. **Implementasi Rx Java**
     1. **Rx Java Check Login**



Gambar 4. 16 Potongan Source Code Rx Java Check Login

Diatas Merupakan potongan *source code* atau kode sumber yang digunakan pada aktivitas login guna mengecek apakah user yang melakukan login sudah terdaftar dalam sistem atau belum. Jika user belum terdaftar maka akan dialihkan ke ativitas form, agar user dapat mendaftarkan dirinya. Pada request ini digunakan Digunakan method observable dari Rx Java untuk mengambil data. Pada method observable, akan dihasilkan 3 kondisi kemungkinan diantaranya yaitu :

1. On Next

Kondisi ketika *asynctask* request asyncronus berjalan dan masih diproses. Ada kondisi ini, request akan menghasilkan data berupa model serialize yang dapat di ketahui isinya dengan menggunakan getter yang dibuat pada model basic.

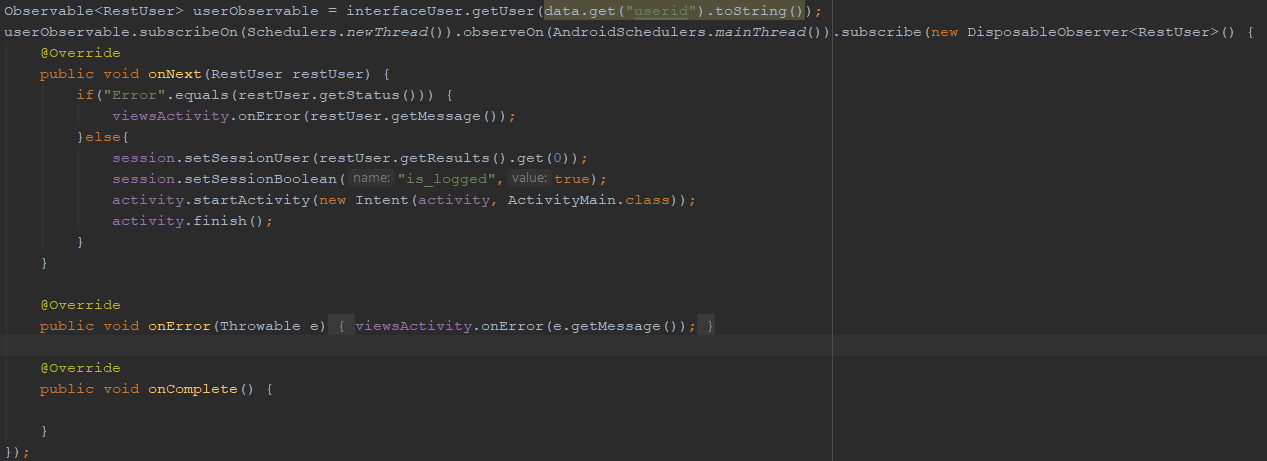
1. On Error

Kondisi ketika request yang di buat terjadi kesalahan atau error pada saat request berjalan.

1. On Complete

Merupakan kondisi akhir atau kondisi disaat *asyncronus* telah selesai dan berhasil melakukn request.

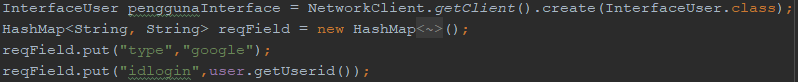
* + 1. **Rx Java Data User**



Gambar 4. 17 Potongan Source Code Rx Java Data User

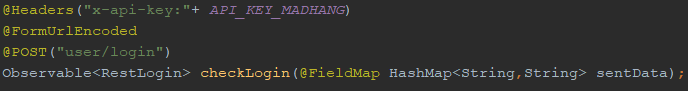
Diatas merupakan potongan *source code* untuk mendapatkan data atau pengguna, yang diberikan secara default oleh platform login yang digunakan (facebook, google, phone number). Untuk method yang digunakan sama seperti Rx Java Pada Check Login, yaitu Obeservable dengan respon yang dihasilkan onComplete, onNext dan onError.

* 1. **Implementasi Retrofit**
     1. **Retrofit Check Login**



Gambar 4. 18 Potongan Source Code Retrofit Check Login Pada File Java

Untuk dapat mengambil data dari api, aplikasi ini menggunakan *library* retrofit dan OkHttp untuk metode pengambilan datanya, gambar diatas merupakan gambar pemanggilan dari settingan endpoint yang dikemas dalam bentuk file Interface Java sehingga dapat dipanggil sebagai method class baru di tempat atau file yang dibutuhkan.



Gambar 4. 19 Potongan Source Code Abstraksi Retrofit Check Login

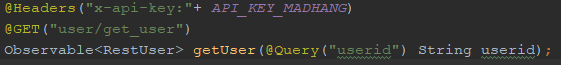
Gambar diatas merupakan potongan *source code* yang terdapat pada file interface java, yang berisikan endpoint dari api yang digunakan pada aplikasi madhang.

* + 1. **Retrofit Data User**



Gambar 4. 20 Potongan Source Code Retrofit Data User Pada File Java

Diatas merupakan pemanggilan interface retrofit untuk mengambil data user. Pemanggilan dilakukan dengan mengirimkan userid sebagai parameter.



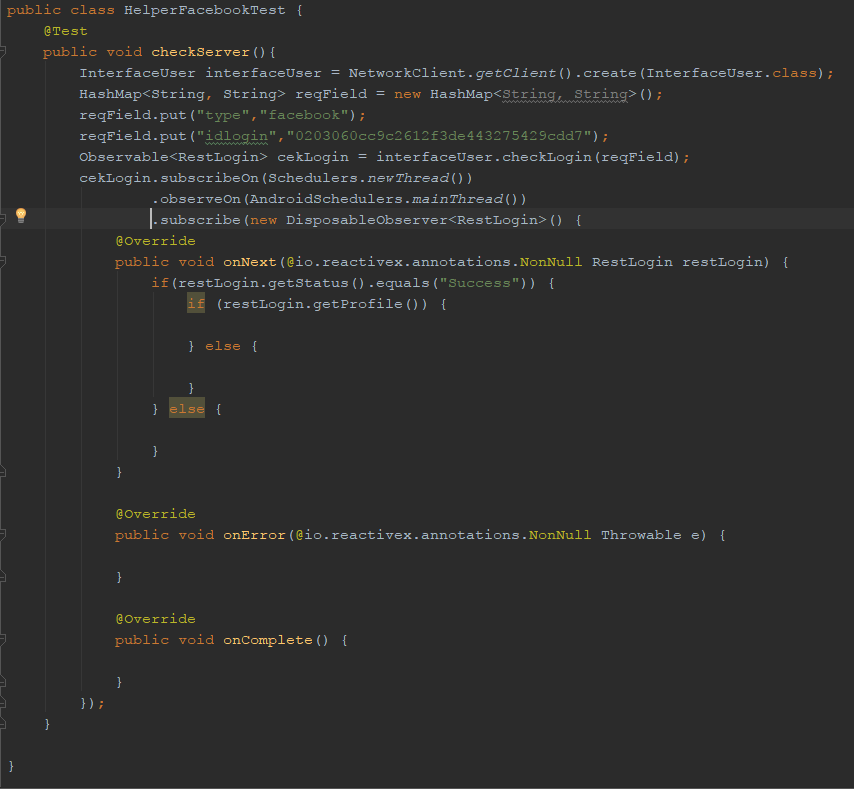
Gambar 4. 21 Potongan Source Code Abstraksi Retrofit Data User

Gambar diatas merupakan potongan *source code* yang terdapat pada file interface java, yang berisikan endpoint dari api yang digunakan pada aplikasi madhang.

* 1. **Pengujian Aplikasi**

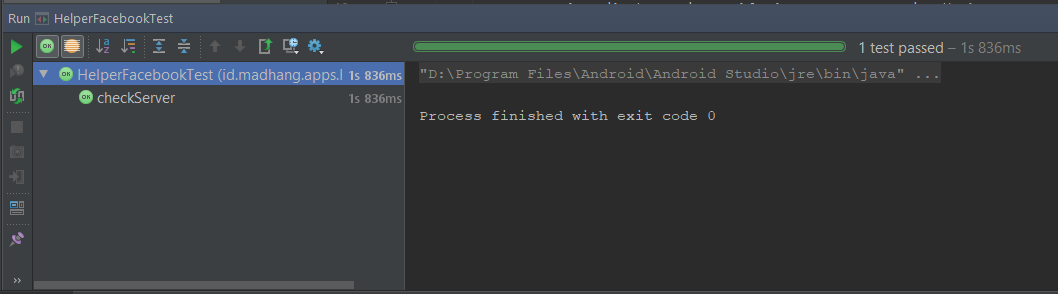
Pada penenlitian ini, peneliti menggunakan metode pengembangan Test Driven Development (TDD) sehingga pruses pengujian dilakukan secara berulang – ulang secara terus menerus dan beriterasu selama proses pengembangan hingga hasil test sesuai dengan yang diharapkan.

Berikut merupakan salah satu contoh *test case* yang dibuat sebelum proses pembuatan atau penulisan *code* aplikasi, *test case* yang dilakukan meliputi pengujian fitur dan *unit test* dengan menggunakan *library* Junit sebagai *test runner* yang tersedia pada IDE Android Studio.



Gambar 4. 22 Test Case Helper Facebook

Dari *test case*  tersebut ketika dijalankan akan menghasilkan output sukses sebagai berikut :



Gambar 4. 23 Hasil Test Case Helper Facebook

Hari hasil tersebut didapatkan bahwa *source code* yang ditulis dalam *test case* dapat berjalan dengan baik tanpa *error*. Dan runtime yang didapatkan untuk mengeksekusi test case tersebut yaitu sekitar 836ms

**BAB V**

**KESIMPULAN DAN SARAN**

* 1. **Kesimpulan**

Berdasarkan penelitian tang telah dilakukan ini , maka dapat diambil kesimpulan bahwa metode MVP (*Model View Presenter*), *library* RxJava, dan Retrofit serta testing dengan *library Junit* dapat diterapkan dengan lancar pada aplikasi Madhang berbasis android. Sehingga proses *debugging* atau *maintenance* aplikasi ketika ada kendala atau *bugs* dapat teratasi dengan mudah. Selain itu dampak lainnya karena penggunaan *pattern* MVP adalah secara tidak langsung proses jual – beli makanan dapat dilakukan dengan lancar kapanpun dan dimanapun secara realtime.

Metode pengujian dan pengembangan *Test Driven Development* juga berhasil menjadikan aplikasi yang dibuat lebih teruji atau *testable*, sehingga dapat meminimalisir kemungkinan munculnya *bug* kesalahan sistem saat pengembangan ataupun saat sudah di *publish / deploy*.

* 1. **Saran**

Sistem ini tentu saja tidak lepas dari kelemahan dan kekurangan. Maka dari itu, agar aplikasi Madhang dapat digunakan dengan maksimal penulis menyarankan beberapa hal, sebagai berikut :

1. Perlu di buat aplikasi Madhang menggunakan platform lain seperti iOS, sehingga pengguna iOS juga dapat menikmati dan menggunakan seluruh fasilitas yang disediakan oleh aplikasi Madhang.
2. Kedepannya dapat ditambah fitur untuk sistem rekomendasi makanan berdasarkan keadaan dengan parameter seperti cuaca, kondisi tempat atau daerah dan umur pengguna sebagai acuan pokok penentuan rekomendasi makanan yang cocok.

**DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | M. J. J. S. Peter Brunn, "e-Marketplaces : Crafting A Winning Strategy," *European Management,* p. 286–298, 2002. |
| [2] | Databoks, Katadata Indonesia, "Berapa Jumlah Wanita Usia Produktif di Indonesia?," 21 April 2017. [Online]. Available: https://databoks.katadata.co.id/datapublish/2017/04/21/berapa-jumlah-wanita-usia-produktif-di-indonesia. |
| [3] | L. R. Fajriah, "Ini Penyebab Jumlah Pengangguran di RI Meningkat," 06 11 2017. [Online]. Available: https://ekbis.sindonews.com/read/1254956/34/ini-penyebab-jumlah-pengangguran-di-ri-meningkat-1509959826. |
| [4] | P. Graham, Paul Graham: The Art of Funding a Startup (A Mixergy Interview), Hyperink and Mixergy, 2011. |
| [5] | E. Ries, The Lean Startup, Amerika Serikat: Crown Publishing Group, 2011. |
| [6] | Y. X. Wu Bo, "Application of MVP Architect in Reengineering of Legacy Financial System," in *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on*, Wuhan, 2009. |
| [7] | D. W. Artem Syromiatnikov, "A Journey Through the Land of," in *Software Architecture (WICSA), 2014 IEEE/IFIP Conference on*, Sydney, 2014. |
| [8] | R. D., "RANCANG BANGUN REST WEB SERVICE UNTUK PERBANDINGAN HARGA PENGIRIMAN DENGAN METODE WEB SCRAPPING DAN PEMANFAATAN API," *J. Chem. Inf. Model,* vol. 53, no. 9, pp. 1689-1699, 2013. |
| [9] | A. H. B. A. P. K. Ibrahim Yunus Muhammad Fiqhan, "Pengembangan Aplikasi Pengingat Salat Dengan Konsep," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 2, no. 10, pp. 3802-3809, 2018. |
| [10] | Wikipedia, "Wikipedia," Wiki, October 2016. [Online]. Available: https://en.wikipedia.org/wiki/Reactive\_programming. [Accessed 19 February 2018]. |
| [11] | ReactiveX, "ReactiveX," ReactiveX, [Online]. Available: http://reactivex.io/intro.html. [Accessed 19 February 2018]. |
| [12] | A. Rosid, "Medium," Medium, 31 July 2016. [Online]. Available: https://medium.com/@ocittwo/belajar-reactivex-programming-ab1b55c1369b. [Accessed 19 February 2018]. |
| [13] | E. R. R., Pemanfaatan JSON (JavaScript Object Notation) Sebagai Data Interchange Pada Sistem Automatic Testing Dan Web Based Learning D3 Teknik Informatika UNS Tugas, Solo, 2012. |
| [14] | K. D. A., Kolaborasi Dahsyat Android dengan PHP & MySql, Yogyakarta: Lokomedia, 2013. |
| [15] | J. Tamplin, "Firebase Expands To Become A Unified App Platform," May 2016. [Online]. Available: https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html. [Accessed 8 January 2018]. |
| [16] | Google, "Firebase," 2016. [Online]. Available: http://firebase.google.com. [Accessed 8 January 2018]. |
| [17] | Google, "Firebase Realtime Database," [Online]. Available: https://firebase.google.com/products/database. [Accessed 9 January 2018]. |

**LAPORAN KKI**

**IMPLEMENTASI MVP (*MODEL VIEW PRESENTER*) MENGGUNAKAN RETROFIT & RX JAVA PADA APLIKASI MADHANG BERBASIS ANDROID**

Diajukan sebagai syarat untuk menyelesaikan program studi Teknik Informatika S1 di Universitas Dian Nuswantoro



Disususn oleh :

Nama : Bona Deny Suryana

NIM : A11.2014.08381

Program Studi : Teknik Informatika

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

SEMARANG

2018