

**APLIKASI *MOBILE* PERIKLANAN DIGIPROM BERBASIS ANDROID**

**(Skripsi)**

**Oleh**

**MUHAMMAD HAFIZ TARULIGANI SIHITE  
NPM 1617051036**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2022**

**APLIKASI *MOBILE* PERIKLANAN DIGIPROM BERBASIS ANDROID**

Oleh

**MUHAMMAD HAFIZ TARULIGANI SIHITE**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2022**

## ABSTRAK

### APLIKASI *MOBILE* PERIKLANAN DIGIPROM BERBASIS ANDROID

Oleh

MUHAMMAD HAFIZ TARULIGANI SIHITE

Digiprom merupakan sebuah *startup* yang menyediakan layanan penayangan iklan. Iklan tersebut nantinya akan ditayangkan di tempat-tempat umum, seperti swalayan, kafe, mall, dan lainnya. Permasalahan yang dihadapi oleh Digiprom adalah belum adanya suatu teknologi informasi untuk membantu interaksi antara Digiprom dengan kliennya (pebisnis). Interaksi yang dimaksud adalah seperti pendaftaran menjadi klien, proses pembuatan kontrak antara klien dengan Digiprom, proses pemantauan kontrak yang sudah dibuat dan penyampaian informasi lainnya mengenai Digiprom. Untuk menyelesaikan masalah tersebut, maka akan dikembangkan sebuah aplikasi *mobile* yang berbasis Android untuk membantu interaksi Digiprom dengan kliennya.

Penelitian ini berfokus pada sisi pengembangan aplikasi Androidnya, sedangkan dari sisi server tidak termasuk dalam penelitian ini. Pengembangan aplikasi ini akan menggunakan metode *Waterfall* dan diuji menggunakan *black box testing* dan *user acceptance testing*. Desain sistem akan digambarkan menggunakan tiga diagram UML, yaitu *use case* diagram, *activity* diagram dan *class* diagram. Kode program yang ditulis menggunakan Bahasa Kotlin dengan arsitektur MVVM (*Model, View, ViewModel*). Terdapat tiga objek utama pada sistem ini, yaitu *user* (pebisnis/klien), *ads* (iklan) dan *location* (lokasi pemasangan iklan). Setiap objek akan memiliki repositorinya masing-masing. Repositori memiliki tiga sumber data, yaitu *network* (server), Room Database, dan Data Store. Penggunaan arsitektur MVVM memudahkan proses pengembangan dan maintenance, karena jika ada perubahan struktur data pada suatu objek, maka tidak akan mempengaruhi struktur data objek lainnya. Hasil dari penelitian ini (aplikasi Android) akan diunggah ke Google Play Store dengan nama Digiprom.

Kata Kunci: Iklan, Kotlin.

## **ABSTRACT**

### **DIGIPROM ADVERTISING MOBILE APPLICATION BASED ON ANDROID**

**By**

**MUHAMMAD HAFIZ TARULIGANI SIHITE**

Digiprom is a startup that provides ad serving services. These advertisements will later be shown in public places, such as supermarkets, cafes, malls, and others. The problem faced by Digiprom is that there is no information technology to help interact between Digiprom and its clients (businessman). The interactions in question are such as registration as a client, the process of making a contract between the client and Digiprom, the process of monitoring the contract that has been made and the delivery of other information about Digiprom. To solve this problem, an Android-based mobile application will be developed to help Digiprom interact with its clients.

This research focuses on the development side of the Android application, while the server side is not included in this study. The development of this application will use the Waterfall method and tested using black box testing and user acceptance testing. The system design is described using three UML diagrams, namely use case diagrams, activity diagrams and class diagrams. Program code written using Kotlin language with MVVM architecture (Model, View, ViewModel). There are three main objects in this system, namely users (businessman/clients), ads (advertisements) and location (advertising locations). Each object will have its own repository. The repository has three data sources, namely network (server), Room Database, and Data Store. The use of MVVM architecture facilitates the development and maintenance process, because if there is a change in the data structure of an object, it will not affect the data structure of other objects.

**Keywords:** Advertisement, Kotlin.

Judul Skripsi : **APLIKASI *MOBILE* PERIKLANAN  
DIGIPROM BERBASIS ANDROID**

Nama Mahasiswa : **Muhammad Hafiz Taruligani Sihite**

Nomor Pokok Mahasiswa : 1617051036

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

**MENYETUJUI**

1. Komisi Pembimbing



**Ardiansyah, S.Kom., M.Kom.**  
NIP 19870128 201803 1 001



**Rizky Prabowo, S.Kom., M.Kom.**  
NIP 19880807 201903 1 011

2. Ketua Jurusan Ilmu Komputer



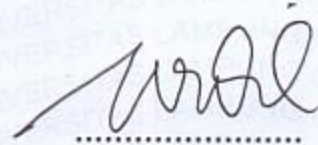
**Didik Kurniawan, S.Si., M.T.**  
NIP 19800419 200501 1 004



## MENGESAHKAN

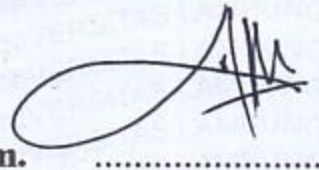
### 1. Tim Penguji

Ketua : **Ardiansyah, S.Kom., M.Kom.**



.....

Sekretaris : **Rizky Prabowo, S.Kom., M.Kom.**



.....

Anggota : **Febi Eka Febriansyah, S.T., M.T.**



.....

### 2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng. Supto Dwi Yuwono, S.Si., M.T.**

NIP. 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **18 Agustus 2022**

## PERNYATAAN

Saya yang bertanda tangan dibawah ini menyatakan bahwa skripsi saya yang berjudul “*APLIKASI MOBILE PERIKLANAN DIGIPROM BERBASIS ANDROID*” adalah benar hasil karya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan ilmiah Universitas Lampung. Jika dikemudian hari terbukti skripsi saya adalah hasil penjiplakan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 05 September 2022  
Penulis



Muhammad Hafiz Taruligani Sihite  
NPM. 1617051036

## RIWAYAT HIDUP



Penulis dilahirkan di Pesawaran, Lampung pada tanggal 17 Juni 1998 sebagai anak kedua dari empat bersaudara dari pasangan Bapak Erbin Sihite, S.T. dan Ibu Ir. Triningsih. Penulis menyelesaikan Pendidikan Sekolah Dasar (SD) di SDN 4 Bagelen pada tahun 2010, Pendidikan Sekolah Menengah Pertama (SMP) di SMPN 1 Gedong Tataan pada tahun 2013 dan Pendidikan Sekolah Menengah Atas (SMA) di SMAN 1 Gadingrejo pada tahun 2016.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2016 melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjadi mahasiswa, penulis aktif dalam berbagai kegiatan dan organisasi kemahasiswaan baik internal maupun eksternal kampus, antara lain:

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2016/2017.
2. Menjadi Koordinator Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Pemrograman Terstruktur pada periode semester genap 2017/2018.



3. Menjadi pengurus di Bidang Eksternal Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2017/2018 sampai periode 2018/2019.
4. Menjadi Ketua Pelaksana Himakom Turnamen (HT) pada tahun 2018.
5. Menjadi anggota Himpunan Mahasiswa Islam (HMI) Komisariat Teknik, Universitas Lampung pada tahun 2018.
6. Melaksanakan Kerja Praktik di DPD Partai GOLKAR Provinsi Lampung pada tahun ajaran genap 2019/2020.
7. Melaksanakan Kuliah Kerja Nyata (KKN) dan menjadi Koordinator Kecamatan pada periode I tahun 2020 di Kecamatan Kelumbayan Barat, Kabupaten Tanggamus, Lampung.

## **PERSEMBAHAN**

### *Alhamdulillahirabbil'alamin*

Puji syukur kepada Allah SWT atas segala berkah, rahmat, dan karunia-Nya, sehingga saya dapat menyelesaikan penulisan skripsi ini. Shalawat serta salam selalu disanjungkan kepada Nabi Muhammad SAW.

Saya persembahkan skripsi ini kepada:

Kedua orang tua yang sangat saya cintai, yakni **Ayahanda Erbin Sihite, S.T. dan Ibunda Ir. Triningsih** yang selalu mendoakan, membimbing dan mendukung sehingga saya dapat menyelesaikan Pendidikan Sarjana ini. Saya berharap kepada Allah SWT agar dapat membalas jasa kedua orang tua saya yang begitu besar dan tak terukur.

**Seluruh keluarga, sahabat dan teman-teman** yang senantiasa memberikan semangat dan dukungan.

**Almamaterku tercinta, Universitas Lampung**

## SANWACANA

Puji dan syukur ke hadirat Allah SWT yang telah memberikan rahmat, karunia serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi *Mobile* Periklanan Digiprom Berbasis Android”. Penyusunan skripsi ini tidak terlepas dari dukungan banyak pihak yang telah membantu, membimbing dan memberi dukungan, sehingga pada kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Kedua orang tua, yakni Bapak Erbin Sihite, S.T. dan Ibu Ir. Triningsih beserta keluarga yang selalu mendoakan, membimbing dan mendukung penulis.
2. Bapak Ardiansyah, S.Kom., M.Kom. selaku pembimbing utama yang telah memberikan bimbingan dan arahan sehingga skripsi ini dapat terselesaikan.
3. Bapak Rizky Prabowo, S.Kom., M.Kom. selaku pembimbing kedua yang telah memberikan topik skripsi, bimbingan dan arahan sehingga skripsi ini dapat terselesaikan.
4. Bapak Febi Eka Febriansyah, S.T., M.T. selaku pembahas yang telah memberikan masukan dalam proses penyusunan skripsi ini.
5. Bapak Didik Kurniawan, S.Si., M.T., selaku pembimbing akademik dan Ketua Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
6. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc selaku Sekretaris Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung dan juga pernah menjadi pembimbing kerja praktik penulis.
7. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan ilmu dan pengalaman yang bermanfaat selama penulis menjadi mahasiswa.

8. Seluruh staf, karyawan, dan laboran Jurusan Ilmu Komputer yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.
9. Saudari Kasandra Cahyani, S.Kom. yang telah menyemangati, memotivasi, menemani, membantu dan berjuang bersama selama masa perkuliahan.
10. Kader HMI Fakultas MIPA dan Fakultas Teknik Universitas Lampung yang telah berjuang bersama dan berbagi pengalaman dalam berbagai hal selama masa perkuliahan.
11. Rekan-rekan Ilmu Komputer yang telah banyak membantu selama masa perkuliahan.

Dalam proses penyusunan skripsi ini tentunya tidak terlepas dari kekurangan dan kesalahan yang disebabkan oleh kekurangan dan keterbatasan pengetahuan penulis. Besar harapan penulis agar pembaca ataupun pihak lain mendapat ilmu dan manfaat dari hasil penyusunan skripsi ini.

Bandar Lampung, 05 September 2022  
Penulis

Muhammad Hafiz Taruligani Sihite  
NPM. 1617051036

## DAFTAR ISI

Halaman

<b>DAFTAR TABEL .....</b>	<b>v</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>I. PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan .....	3
1.5. Manfaat .....	4
<b>II. LANDASAN TEORI .....</b>	<b>5</b>
2.1. Android .....	5
2.2. <i>Software Development Life Cycle (SDLC)</i> .....	5
2.3. <i>Unified Modeling Language (UML)</i> .....	5
2.3.1. <i>Use Case Diagram</i> .....	6
2.3.2. <i>Activity Diagram</i> .....	7
2.3.3. <i>Class Diagram</i> .....	8
2.4. Kotlin .....	8
2.5. Coroutine.....	9
2.6. Room.....	9
2.7. <i>Black Box Testing</i> .....	9
2.8. <i>User Acceptance Testing (UAT)</i> .....	10
<b>III. METODOLOGI PENELITIAN .....</b>	<b>11</b>
3.1. Sumber Data.....	11
3.2. Tempat dan Waktu Penelitian.....	11
3.3. Alat dan Bahan.....	11
3.4. Kerangka Penelitian .....	12
3.4.1. Analisa Kebutuhan .....	12
3.4.2. Desain Sistem.....	13
3.4.3. Desain <i>Interface</i> .....	29
3.4.4. Penulisan Kode Program.....	41
3.4.5. Pengujian Program .....	42



<b>IV. HASIL DAN PEMBAHASAN.....</b>	<b>47</b>
4.1. Hasil Implementasi .....	47
4.2. Hasil Pengujian .....	61
<b>V. KESIMPULAN DAN SARAN .....</b>	<b>69</b>
5.1. Kesimpulan .....	69
5.2. Saran .....	69
<b>DAFTAR PUSTAKA .....</b>	<b>71</b>

## DAFTAR TABEL

Tabel	Halaman
1. Simbol <i>Use case diagram</i> (Hendini, 2016).....	6
2. Simbol <i>Activity Diagram</i> (Hasanah & Untari, 2020).....	7
3. Simbol <i>Class diagram</i> (Mallick & Das, 2013) .....	8
4. Bobot Penilaian UAT (Abraham & Ismail, 2021) .....	10
5. Rancangan Pengujian Sistem .....	42
6. Kuesioner UAT .....	46
7. Hasil Pengujian Aplikasi.....	62
8. Hasil Kuesioner UAT.....	66

## DAFTAR GAMBAR

Gambar	Halaman
1. <i>Storyboard</i> Digiprom. ....	2
2. Metode <i>Waterfall</i> (Dima & Maassen, 2018).....	12
3. <i>Use case diagram</i> .....	13
4. <i>Activity diagram login</i> .....	14
5. <i>Activity diagram</i> daftar akun. ....	15
6. <i>Activity diagram</i> atur ulang kata sandi.....	15
7. <i>Activity diagram</i> lihat tempat iklan. ....	16
8. <i>Activity diagram</i> pasang iklan. ....	17
9. <i>Activity diagram</i> pantau iklan. ....	18
10. <i>Activity diagram</i> perpanjang iklan. ....	18
11. <i>Activity diagram</i> lengkapi data profil.....	19
12. <i>Activity diagram</i> ganti kata sandi.....	20
13. <i>User class diagram</i> .....	21
14. <i>Ads class diagram</i> . ....	22
15. <i>Location class diagram</i> .....	23
16. <i>Room class diagram</i> .....	24
17. <i>Network class diagram</i> .....	25
18. <i>Repository class diagram</i> .....	27

19. MVVM <i>class diagram</i> .....	28
20. Desain <i>interface</i> masuk akun. ....	29
21. Desain <i>interface</i> daftar akun. ....	30
22. Desain <i>interface</i> lupa kata sandi. ....	31
23. Desain <i>interface</i> melengkapi data profil. ....	32
24. Desain <i>interface</i> beranda. ....	33
25. Desain <i>interface list</i> lokasi iklan. ....	34
26. Desain <i>interface</i> detail tempat iklan. ....	35
27. Desain <i>interface list</i> iklan. ....	36
28. Desain <i>interface</i> detail iklan. ....	37
29. Desain <i>interface</i> buat iklan. ....	38
30. Desain <i>interface</i> pembayaran. ....	39
31. Desain <i>interface</i> profil. ....	40
32. Desain <i>interface</i> bantuan. ....	41
33. Halaman pendaftaran akun. ....	48
34. Halaman masuk. ....	49
35. Halaman lupa kata sandi. ....	50
36. Halaman beranda. ....	51
37. Halaman <i>list</i> lokasi iklan. ....	52
38. Halaman detail lokasi iklan. ....	53
39. Halaman pembuatan iklan. ....	54
40. Halaman menambah lokasi pemasangan iklan .....	55
41. Halaman pembayaran. ....	56
42. Halaman detail iklan. ....	57

43. Halaman profil. ....	58
44. Halaman melengkapi data profil. ....	59
45. Halaman mengubah kata sandi. ....	60
46. Halaman bantuan.....	61



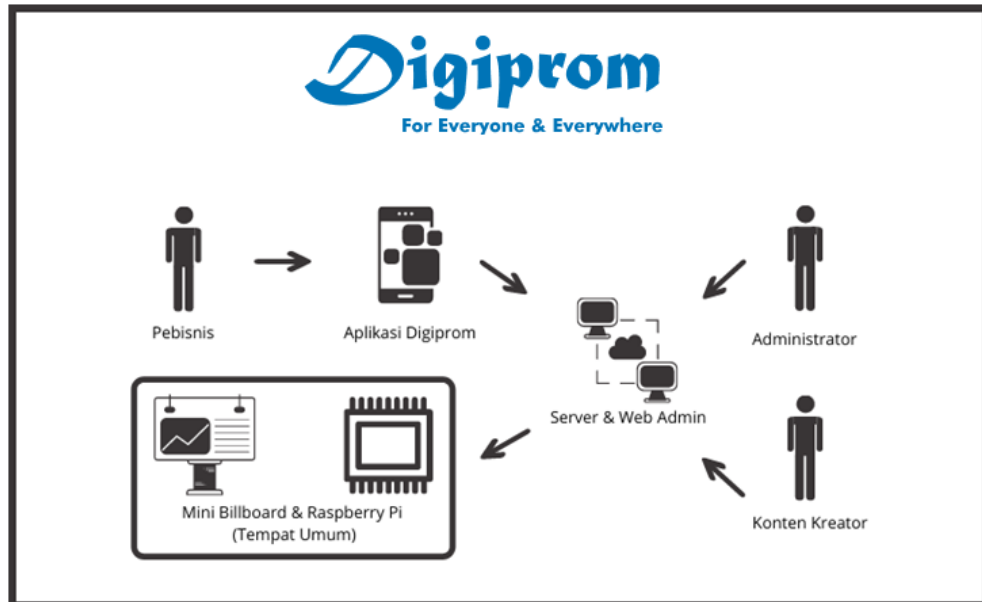
## I. PENDAHULUAN

### 1.1. Latar Belakang

Teknologi informasi merupakan gabungan antara teknologi komputer dengan teknologi telekomunikasi (Kadir & Triwahyuni, 2014). Teknologi informasi dapat diartikan sebagai sebuah bentuk komunikasi yang dibangun menggunakan teknologi komputer, sehingga terbentuk suatu otomatisasi dalam komunikasi tersebut yang dikontrol program-program tertentu. Dengan adanya otomatisasi tersebut maka akan memudahkan suatu pekerjaan yang berulang-ulang dan kompleks. Android merupakan salah satu contoh dari teknologi informasi yang berkembang pesat saat ini. Android merupakan sistem operasi *open source* yang dikembangkan oleh Google (Google, 2016). Sistem operasi Android beroperasi di berbagai *device*, termasuk pada perangkat *smartphone*. Perkembangan teknologi informasi ini membantu berbagai bidang kehidupan, seperti bidang ekonomi, sosial, budaya, politik, bisnis dan lain-lain.

Bisnis merupakan suatu kegiatan individu atau kelompok yang melakukan aktivitas produksi, pembelian atau penjualan dalam bentuk barang maupun jasa untuk meraih suatu tujuan tertentu (Sembiring, 2014). Dalam bidang bisnis, para pebisnis membutuhkan suatu bentuk strategi untuk mengembangkan bisnisnya. Salah satu bentuk strategi itu adalah menggunakan iklan untuk mempromosikan produk-produk bisnis yang mereka jalani. Iklan merupakan suatu bentuk komunikasi yang bersifat komersial dan non-personal tentang ide-ide, barang atau jasa melalui suatu media tertentu (Kriyantono, 2013). Para pelaku bisnis mengiklankan produknya pada suatu perusahaan khusus yang bergerak di bidang periklanan. Di Kota Bandar Lampung sendiri

sudah terdapat beberapa perusahaan ataupun bentuk organisasi lainnya yang bergerak di bidang jasa periklanan ini, salah satunya adalah Digiprom.



Gambar 1. *Storyboard* Digiprom.

Digiprom merupakan sebuah *startup* baru yang bergerak dalam bidang periklanan yang beroperasi di wilayah Bandar Lampung. Digiprom menampung iklan dari para klien dalam bentuk video berdurasi 10 detik yang nantinya akan ditampilkan melalui beberapa perangkat media layar dengan ukuran 40 *inch*. Media layar ini terpasang di tempat-tempat umum seperti rumah makan, bengkel, kafe, swalayan dan lain-lain. Iklan para klien nantinya akan muncul di tempat-tempat sesuai dengan kontrak yang sudah ditetapkan antara Digiprom dengan kliennya.

Permasalahan yang dihadapi oleh Digiprom adalah belum adanya suatu aplikasi *mobile* untuk membantu interaksi antara Digiprom dengan kliennya. Interaksi yang dimaksud adalah seperti pendaftaran menjadi klien, proses pembuatan kontrak antara klien dengan Digiprom, proses pemantauan kontrak yang sudah dibuat dan penyampaian informasi lainnya mengenai Digiprom. Untuk menyelesaikan masalah ini, maka akan dibangun sebuah aplikasi *mobile*

yang berbasis Android. Diharapkan nantinya aplikasi tersebut akan memudahkan interaksi antara Digiprom dengan kliennya, sehingga membantu bisnis yang dijalankan oleh Digiprom.

## **1.2. Rumusan Masalah**

Berdasarkan penjelasan pada latar belakang, dibutuhkan suatu aplikasi *mobile* berbasis Android untuk membantu proses bisnis pada Digiprom. Proses bisnis yang dimaksud adalah proses bisnis dalam hal interaksi antara Digiprom dengan kliennya. Dengan adanya permasalahan tersebut, rumusan masalah yang terdapat pada penelitian ini adalah sebagai berikut:

1. Bagaimana membangun aplikasi Android untuk membantu interaksi pebisnis dengan Digiprom.
2. Bagaimana hasil uji dari segi fungsionalitas aplikasi.
3. Bagaimana hasil uji kebermanfaatan aplikasi terhadap Digiprom.

## **1.3. Batasan Masalah**

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Aplikasi dibangun dengan basis Android dan menggunakan *database* SQL server yang dikelola oleh rekan tim lainnya.
2. Klien dapat melakukan pendaftaran dan pengolahan akun pada aplikasi.
3. Klien dapat membuat kontrak dengan pihak Digiprom. Proses-proses yang ada pada kontrak ini yaitu pengunggahan konten iklan, masa tayang iklan, tempat penayangan iklan dan tarif yang harus dibayar klien.
4. Klien dapat memantau status kontrak termasuk dalam hal sisa waktu penayangan iklan yang dimiliki klien.
5. Pengguna bisa mendapatkan informasi tentang Digiprom, termasuk tempat yang disediakan Digiprom untuk pemasangan iklan.

## **1.4. Tujuan**

Tujuan dari penelitian ini adalah:

1. Membangun suatu aplikasi *mobile* berbasis Android yang dapat mendukung dan mengembangkan proses bisnis pada Digiprom. Proses

bisnis yang dimaksud adalah interaksi antara Digiprom dengan kliennya, mulai dari proses pendaftaran menjadi klien, mengadakan kontrak, hingga pemantauan status kontrak yang telah dibuat.

2. Memberikan saran kepada pihak Digiprom untuk menggunakan dan mengembangkan aplikasi ini guna mendukung proses bisnis Digiprom.

### **1.5. Manfaat**

Penelitian ini memberikan manfaat kepada dua pihak, yaitu pihak penulis dan pihak Digiprom. Beberapa poin manfaat dari penelitian ini sebagai berikut:

1. Topik ini menjadi skripsi sebagai syarat kelulusan di Jurusan S1 Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.
2. Pihak Digiprom dapat menggunakan aplikasi ini dalam proses bisnisnya.
3. Berkembangnya cakupan bisnis dari Digiprom dengan terunggahnya aplikasi ini di Google Play Store.
4. Memudahkan para pebisnis untuk mendaftarkan diri sebagai klien.
5. Memudahkan klien untuk membuat suatu kontrak pengiklanan produknya.
6. Memudahkan klien untuk memantau kontrak pengiklanan yang dibuat.
7. Terbentuknya suatu komunikasi bisnis yang baik antara Digiprom dengan klien.

## II. LANDASAN TEORI

### 2.1. Android

Android merupakan sistem operasi *open source* yang dikembangkan oleh Google (Google, 2016). Sistem operasi Android beroperasi di berbagai *device*, termasuk pada perangkat *smartphone*. Android sendiri menjadi sistem operasi yang banyak digunakan oleh para pengembang karena sifatnya yang *open source* dibanding saingannya seperti IOS. Karakteristik dari Android yang *open source* menyebabkan banyak pengguna *smartphone* yang memilih Android dibandingkan sistem operasi lain.

### 2.2. *Software Development Life Cycle (SDLC)*

SDLC adalah alur pengembangan sistem yang terdapat di dalam Rekayasa Perangkat Lunak (Bhuvanewari & Prabakaran, 2013). SDLC itu sendiri merupakan sebuah pola yang berfungsi untuk melakukan proses pembuatan maupun perubahan sistem dari sebuah aplikasi. Dapat disimpulkan bahwa dalam proses membangun suatu sistem informasi harus menggunakan SDLC yang tepat sesuai dengan *requirement* sistem dan pembuat sistem itu sendiri. Hal ini dilakukan agar proses pembangunan dan pengembangan sistem informasi menjadi terarah sehingga dapat mencapai hasil yang maksimal.

### 2.3. *Unified Modeling Language (UML)*

UML adalah suatu bahasa yang digunakan untuk mendeskripsikan atau menggambarkan suatu ide (sistem) yang akan dibangun (Wazlawick, 2014). UML dikembangkan oleh James Rumbaugh, Grady Booch dan Ivar Jacobson dan masih terus direvisi sampai sekarang dan memiliki 3 keluarga diagram sebagai berikut:




1. *Structure diagrams*: digunakan untuk mendefinisikan komponen-komponen apa saja yang diperlukan pada sebuah sistem. Anggota dari *structure diagrams* antara lain *package*, *class*, *objects*, *composite structure*, *component*, *profile*, dan *deployment diagrams*.
2. *Behavior diagrams*: digunakan untuk menggambarkan proses berjalannya suatu sistem dan fungsi - fungsi yang ada di dalamnya. Diagram - diagram yang termasuk dalam *behavior diagrams* adalah *use case*, *activity*, dan *state machine diagrams*.
3. *Interaction diagrams*: digunakan untuk memperjelas aliran kontrol antar komponen-komponen yang ada pada sistem. Anggota dari *interaction diagrams* antara lain *communication*, *sequence*, *timing*, dan *interaction overview diagrams*.

Dalam pengembangan suatu sistem tidak semua diagram-diagram tersebut harus digunakan, tetapi hanya diagram-diagram yang merepresentasikan gambaran penting suatu sistem. Penelitian ini akan menggunakan *use case* diagram, *activity diagram* dan *class diagram* dengan penjelasan sebagai berikut:

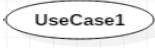




### 2.3.1. Use Case Diagram

*Use case diagram* merupakan pemodelan untuk aktivitas apa saja yang ada pada suatu sistem. *Use case* juga memperjelas aktor atau siapa saja yang terlibat dalam sistem tersebut. Aktor disini tidak berarti hanya manusia, tetapi juga sistem eksternal yang menunjang sistem tersebut. Simbol-simbol yang digunakan dalam *Use case* diagram dijelaskan pada Tabel 1 sebagai berikut:

Tabel 1. Simbol *Use case diagram* (Hendini, 2016)

No.	Gambar	Nama	Keterangan
1		<i>Actor</i>	Merupakan bentuk abstraksi dari orang/pengguna/subjek yang terlibat dalam sistem








Tabel 1 (lanjutan)

No.	Gambar	Nama	Keterangan
2		<i>Use Case</i>	Menunjukkan fitur-fitur/fungsionalitas dari suatu sistem
3		<i>Association</i>	Digunakan untuk merepresentasikan hubungan/interaksi antara <i>actor</i> dengan <i>use case</i>
4		<i>Include</i>	Menspesifikasikan suatu <i>use case</i> membutuhkan <i>use case</i> yang lainnya
5		<i>Extend</i>	Menunjukkan suatu <i>use case</i> dapat dijalankan apabila <i>use case</i> tersebut dijalankan
6		<i>Generalization</i>	Menunjukkan suatu <i>child object</i> memiliki perilaku seperti <i>parent object</i> .

### 2.3.2. Activity Diagram

*Activity diagram* menggambarkan proses-proses atau aliran yang terjadi pada suatu aktivitas atau fungsi sistem. Simbol-simbol yang digunakan dalam *activity diagram* disajikan dalam Tabel 2 sebagai berikut:

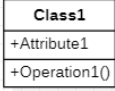




Tabel 2. Simbol *Activity Diagram* (Hasanah & Untari, 2020)

No	Gambar	Nama	Keterangan
1		<i>Initial</i>	Menunjukkan titik awal <i>activity</i> .
2		<i>Final</i>	Menunjukkan titik akhir <i>activity</i> .
3		<i>Action</i>	Menggambarkan suatu kegiatan atau proses.
4		<i>Decision</i>	Digunakan untuk menggambarkan suatu ketentuan atau <i>statement</i> .
5		<i>Control Flow</i>	Menunjukkan aliran dari proses-proses pada <i>activity diagram</i> .
6		<i>Fork</i>	Menunjukkan suatu kegiatan atau proses dilakukan secara paralel.
7		<i>Join</i>	Menggabungkan dua kegiatan atau proses menjadi satu.

### 2.3.3. Class Diagram

*Class diagram* adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sebuah sistem. Kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan nama kelas. Setiap kelas memiliki fungsi-fungsi yang berkaitan dengan *attribute* kelas tersebut. Komponen simbol yang ada di *Class diagram* dijelaskan pada Tabel 3 sebagai berikut:

Tabel 3. Simbol *Class diagram* (Mallick & Das, 2013)

No	Gambar	Nama	Keterangan
1		<i>Class</i>	Terdiri dari <i>class name</i> , atribut dan operasi yang dilakukan pada <i>class</i> tersebut
2		<i>Association</i>	Menghubungkan satu <i>class</i> dengan <i>class</i> yang lain secara umum
3		<i>Generalization</i>	Menghubungkan <i>parent class</i> dengan <i>child class</i>
4		<i>Dependencies</i>	Menunjukkan hubungan dimana satu <i>class</i> bergantung dengan <i>class</i> yang lainnya
5		<i>Aggregation</i>	Menunjukkan bahwa <i>class</i> tersebut merupakan bagian yang lebih kecil dari <i>class</i> lainnya

## 2.4. Kotlin

Kotlin adalah bahasa pemrograman yang berbasis *Java Virtual Machine* (Sibarani dkk., 2018). Karena bahasa ini berbasis JVM, maka proyek Kotlin dapat digabung dengan bahasa Java. Hal ini membuat proyek dengan menggunakan bahasa Kotlin dapat menjadi lebih fleksibel untuk dikembangkan. Kotlin dikembangkan oleh JetBrains pada tahun 2011 dan diimplementasikan ke pengembangan aplikasi Android oleh Google pada tahun 2017.

## 2.5. Coroutine

Coroutine merupakan suatu *library* dari Kotlin yang dapat digunakan untuk mengelola kode yang dijalankan secara asinkron (Google, 2022a). Terdapat beberapa kasus dimana kode harus memuat data dengan waktu yang cukup lama. Jika kode tersebut dijalankan di *main thread* maka hal tersebut dapat memblokir kode *User Interface* sehingga menyebabkan aplikasi tidak responsif. Disinilah kegunaan Coroutine, dimana kode yang memuat data tersebut dijalankan di *thread* lain (tidak di *main thread*) sehingga kode *User Interface* tidak terblokir oleh kode tersebut.

Beberapa fitur unggulan dari Coroutine meliputi:

1. Menghindari kebocoran memori.
2. Mendukung *automatic code cancelation* jika *parent thread* dibatalkan.
3. Terintegrasi dengan beberapa *library* Jetpack lainnya seperti Room, Data Store, dan lain – lain.

## 2.6. Room

Room merupakan salah satu *library* dari Android Jetpack yang berguna untuk membuat *database* lokal pada perangkat Android (Google, 2022b). Room merupakan pengembangan dari *library* SQLite sehingga mendukung kompilasi *Structured Query Language* (SQL). Beberapa keunggulan dari Room adalah sebagai berikut:

1. Penggunaan anotasi yang praktis sehingga menghindari kode *boilerplate* yang rentan akan *error*.
2. Jalur migrasi *database* sederhana.
3. Mendukung *Structured Query Language*.

## 2.7. Black Box Testing

Metode *Black box testing* adalah suatu metode pengujian perangkat lunak dari segi fungsionalitas (Snadhika, 2018). *Black box testing* langsung berfokus pada domain terakhir dari suatu sistem (*activity* dari setiap fitur). Pengujian tersebut dikelompokkan dengan sebuah kelas uji, yang dimana setiap kelas uji

terdapat skenario-skenario pengujian. Keberhasilan pengujian diukur dari setiap skenario pengujian yang dibuat.

## 2.8. *User Acceptance Testing* (UAT)

*User Acceptance Testing* merupakan sebuah pengujian sistem dari perspektif pengguna, pemilik atau orang yang berkepentingan dengan sistem yang dibangun (Hambling & Goethem, 2013). *User Acceptance Testing* berfokus pada poin apakah sistem yang dibangun memberikan manfaat pada proses bisnis atau tidak ketika sistem tersebut dioperasikan oleh pengguna. *User Acceptance Testing* dilakukan dengan pertanyaan langsung kepada pengguna atau pemilik sistem dan penilaian dari setiap pertanyaan dapat menggunakan keterangan seperti pada Tabel 4.

Tabel 4. Bobot Penilaian UAT (Abraham & Ismail, 2021)

Bobot Nilai	Keterangan
1	Sangat tidak setuju
2	Tidak setuju
3	Cukup
4	Setuju
5	Sangat Setuju

Semua bobot penilaian dari setiap pertanyaan akan dihitung rata-ratanya dalam bentuk persentase. Berdasarkan nilai tersebut nantinya akan diambil kesimpulan apakah sistem diterima atau tidak dengan indikator sebagai berikut:

Sangat Diterima	= 81% - 100%
Diterima	= 61% - 80%
Cukup Diterima	= 41% - 60%
Tidak Diterima	= 21% - 40%
Sangat Tidak Diterima	= 0% - 20%

### **III. METODOLOGI PENELITIAN**

#### **3.1. Sumber Data**

Sumber data dalam penelitian yaitu segala sesuatu yang dapat memberikan informasi mengenai data. Data yang dibutuhkan dalam penelitian ini diperoleh dari hasil wawancara (*interview*) mendalam dengan pihak Digiprom. Peneliti mendapatkan gambaran dan informasi dari narasumber untuk dapat menjawab permasalahan yang ada dalam penelitian ini. Data ini juga diperoleh dengan cara diskusi dan tinjauan langsung ke lapangan (*observasi*).

#### **3.2. Tempat dan Waktu Penelitian**

Penelitian Pembuatan Aplikasi *Mobile* Periklanan “Digiprom” Berbasis Android ini dilaksanakan pada Semester Genap tahun ajaran 2020/2021. Penelitian ini dilaksanakan di Jurusan S1 Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung.

#### **3.3. Alat dan Bahan**

Perangkat keras yang digunakan dalam penelitian adalah laptop dengan spesifikasi ini sebagai berikut:

1. Prosesor: Intel(R) Core(TM) i5-8250U
2. Memori: 12GB RAM
3. Penyimpanan: SSD 250gb (Samsung SSD 970)
4. Kartu Grafis: NVIDIA GeForce MX150

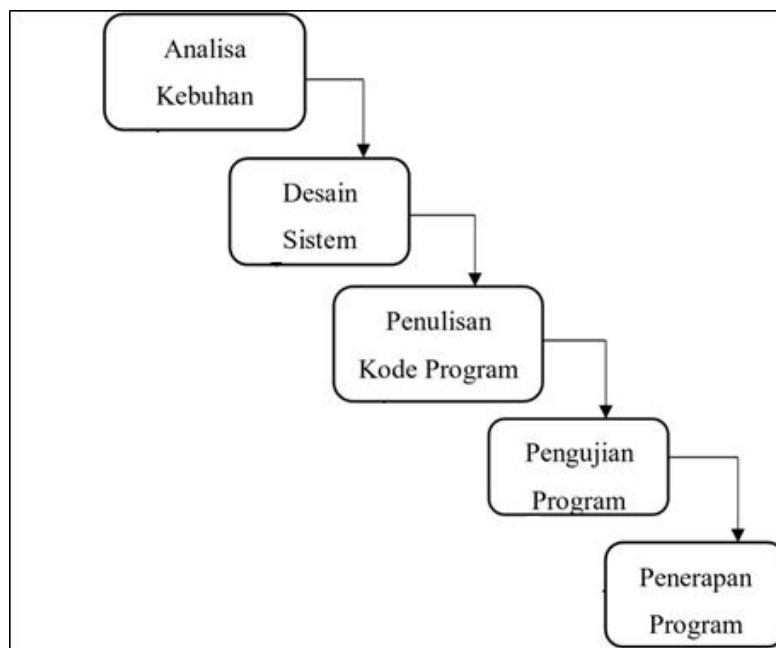
Penelitian ini juga didukung dengan beberapa perangkat lunak, di antaranya:

1. Windows 10 Pro
2. Android Studio

3. *Android Virtual Device*
4. Figma
5. Star UML

### 3.4. Kerangka Penelitian

Metode pengembangan sistem yang digunakan pada penelitian ini mengikuti kerangka kerja SDLC dengan menggunakan metode *Waterfall*.



Gambar 2. Metode *Waterfall* (Dima & Maassen, 2018)

#### 3.4.1. Analisa Kebutuhan

Setelah sumber data dikumpulkan, didapatkan beberapa poin mengenai kebutuhan sistem, di antaranya:

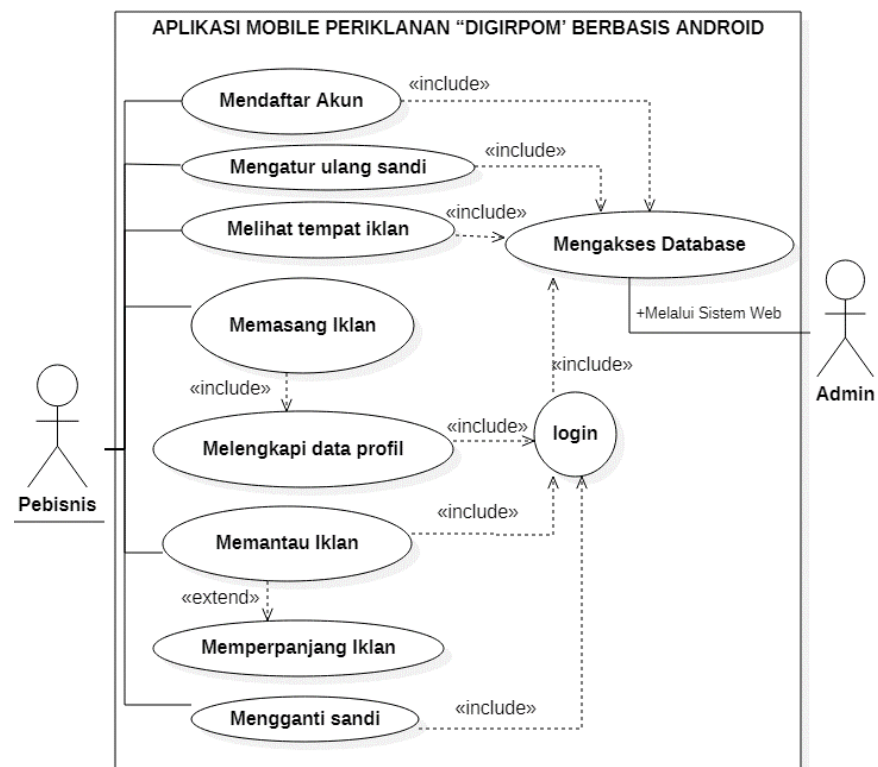
- a. Melakukan pendaftaran, *login* dan pengaturan akun *user*.
- b. Menambahkan iklan.
- c. Memantau iklan.
- d. Melakukan perpanjangan iklan.
- e. Aplikasi dapat berjalan dengan versi minimum Android Lollipop.
- f. Tidak terjadi *bottleneck* pada saat aplikasi berjalan.

### 3.4.2. Desain Sistem

Dalam tahap mendesain sistem akan menggunakan *Unified Modeling Language* dengan 3 diagram, yaitu *use case*, *activity*, dan *class diagram*. Berikut penjelasan dari desain sistem yang dibuat :

#### a. Use case diagram

Sistem ini memiliki dua aktor atau pengguna, yaitu *user* dan *admin*. Akan tetapi admin hanya bisa mengakses *database* melalui website, sehingga desain sistem ini akan lebih berfokus pada aktor *user*. Terlihat pada Gambar 3 bahwa pada beberapa *activity*, *user* harus melakukan *login* terlebih dahulu sebelum melakukan *activity* tersebut. Hal tersebut dilakukan agar *user* dapat melihat aktivitas lainnya secara sekilas terlebih dahulu sebelum melakukan pendaftaran akun.



Gambar 3. Use case diagram.

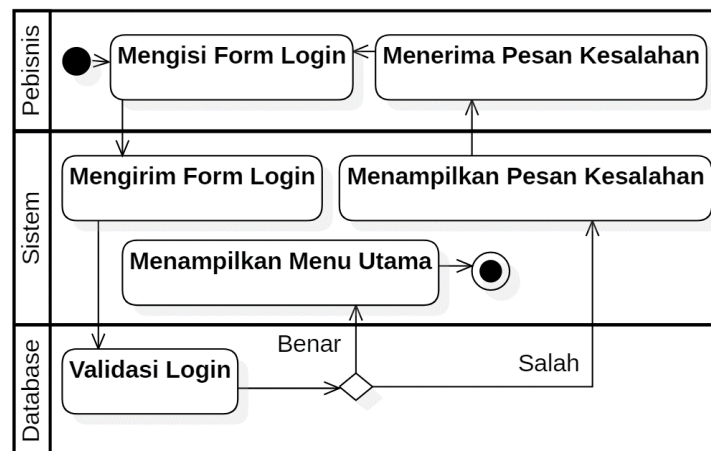


## b. Activity diagram

*Activity diagram* didesain dengan mengarahkan alur kerja aplikasi supaya mengantisipasi kesalahan – kesalahan *user* dalam melakukan *input data*. Hal tersebut menjadi perhatian karena mengingat aplikasi ini merupakan versi pertama sehingga *user* belum mengenal jalannya aplikasi. Berikut enam *activity diagram* dan penjelasannya:

### 1. Activity diagram Login

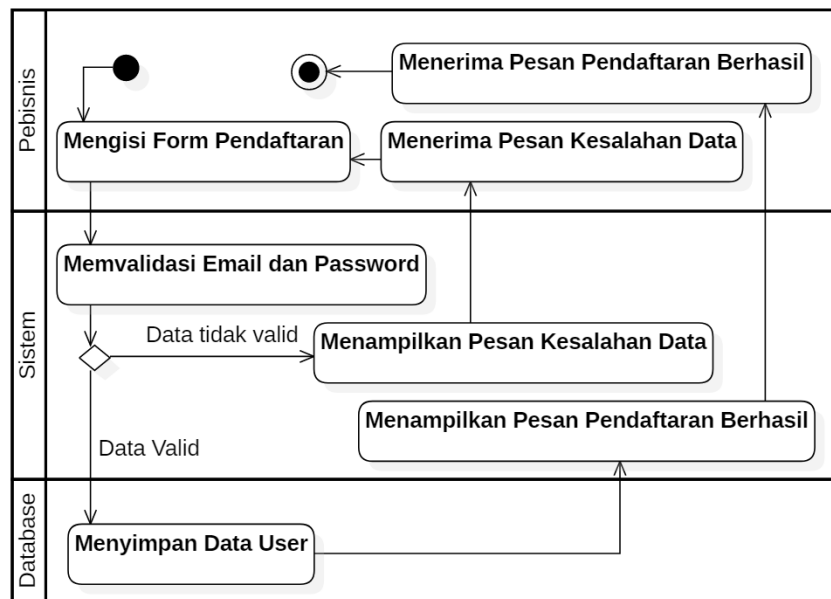
Aktivitas dimulai dengan *user* mengisi *email* dan kata sandi akun yang akan divalidasi dengan *database*. Hasil validasi yang sukses akan membawa *user* ke menu utama, tetapi apabila validasi gagal maka sistem akan menampilkan pesan kesalahan dan *user* diminta mengisi kembali.



Gambar 4. Activity diagram login.

### 2. Activity diagram daftar akun

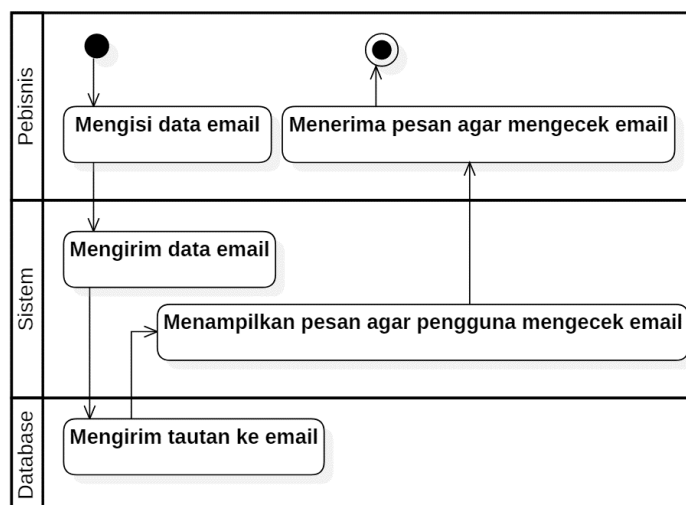
*User* mengisi formulir pendaftaran yang kemudian datanya akan divalidasi oleh sistem. Jika data valid maka data akan dikirim ke *database* dan sistem menampilkan pesan bahwa pendaftaran akun telah berhasil, tapi jika data tidak valid maka *user* akan diminta untuk mengecek data.



Gambar 5. Activity diagram daftar akun.

### 3. Activity diagram atur ulang kata sandi

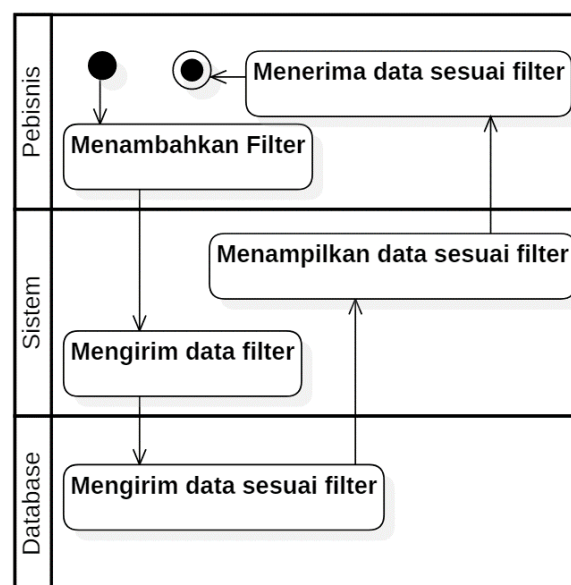
Jika *user* lupa terhadap kata sandinya saat akan melakukan *login*, maka *user* dapat mengatur ulang kata sandinya dengan mengisi data *email* yang digunakan. Setelah data *email* dikirim oleh sistem, maka *user* akan menerima tautan pada *emailnya* yang akan diarahkan ke halaman untuk mengatur ulang kata sandi *user*.



Gambar 6. Activity diagram atur ulang kata sandi.

#### 4. Activity diagram lihat tempat iklan

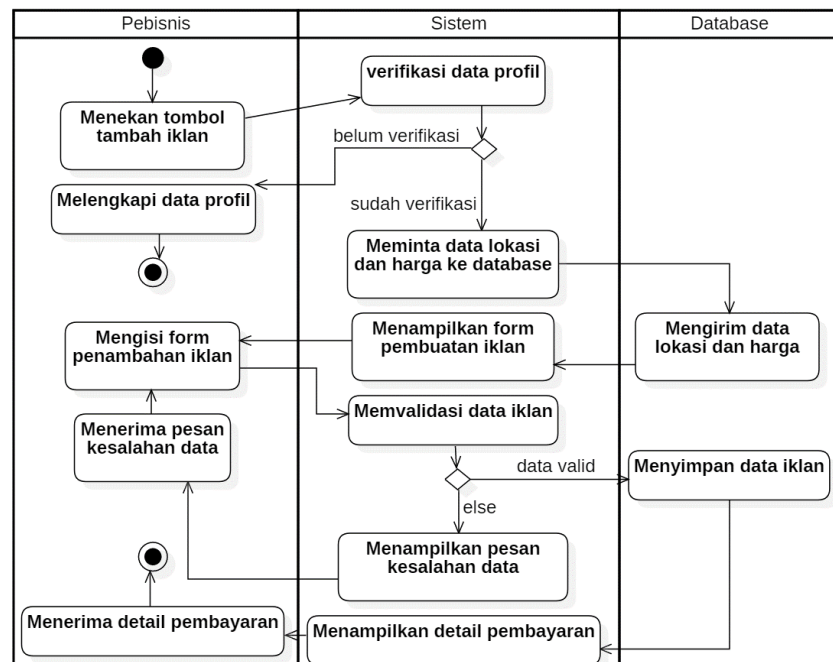
Sistem akan menampilkan tempat-tempat iklan yang disediakan. *User* juga dapat menambahkan beberapa filter, misalnya berdasarkan kategori tempat (tempat umum, rumah makan, toko, swalayan) atau berdasarkan rentang harga. Jika *user* ingin melihat informasi detail tempat iklan, maka *user* dapat menekannya dan sistem akan menampilkan informasi detail dari tempat iklan.



Gambar 7. Activity diagram lihat tempat iklan.

#### 5. Activity diagram pasang iklan

Pada *activity diagram* ini, sistem akan mengecek apakah *user* sudah melengkapi data profil atau belum, jika belum maka sistem akan mengarahkan ke *activity* lengkapi data profil. Sistem akan *request* data pilihan lokasi dan harga terlebih dahulu ke *database*. Hal ini dikarenakan kedua data tersebut bisa saja berubah – ubah sesuai dengan kebijakan lokasi dan harga yang diberlakukan oleh Digiprom.

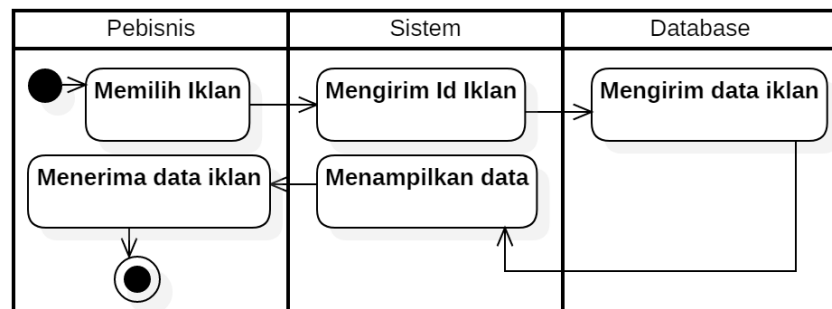


Gambar 8. Activity diagram pasang iklan.

Setelah data diterima, sistem akan menampilkan formulir yang akan diisi oleh *user*. *User* dapat menambahkan konten iklan dengan tipe video, foto, ataupun bentuk lainnya. *User* juga memilih sendiri lokasi dimana iklannya akan ditayangkan dengan paket harga yang ditawarkan. Data kemudian divalidasi dan jika sukses akan dikirimkan ke server, apabila gagal maka sistem akan menampilkan pesan kesalahannya. Setelah data sukses terkirim, sistem akan menampilkan detail pembayaran yang harus dilakukan oleh *user* agar iklan dapat ditayangkan.

#### 6. Activity diagram pantau iklan

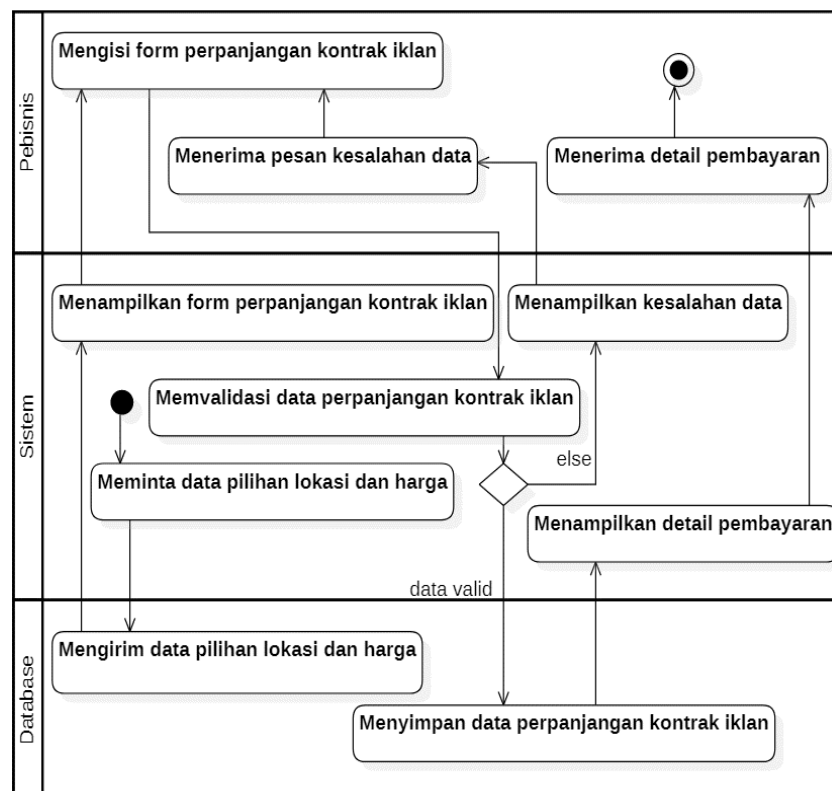
Semua nama iklan dan statusnya yang dimiliki oleh *user* akan tersusun pada halaman utama. *User* dapat memilih dengan cara menekan iklan yang ingin dilihat secara detail. *User* juga dapat melakukan perpanjangan masa tayang iklan pada halaman ini dengan menekan tombol yang ada.



Gambar 9. *Activity diagram* pantau iklan.

### 7. *Activity diagram* perpanjang iklan

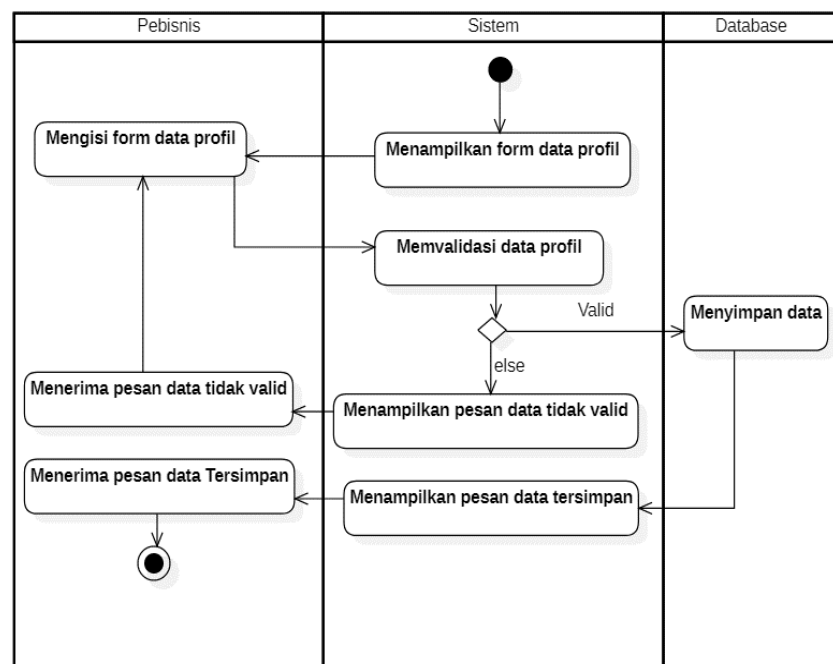
Sistem akan meminta data pilihan lokasi dan harga terbaru kepada server, kemudian ditampilkan dalam bentuk formulir yang akan diisi oleh *user*. Data akan divalidasi terlebih dahulu sebelum dikirimkan ke server.



Gambar 10. *Activity diagram* perpanjang iklan.

### 8. Activity diagram lengkapi data profil

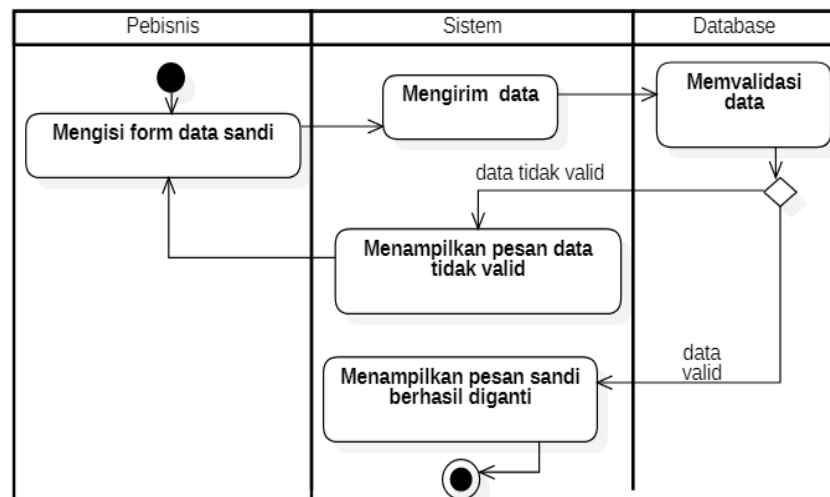
*User* akan diminta untuk melengkapi data profil terlebih dahulu sebelum membuat iklan. Data yang sudah diisi oleh *user* akan divalidasi terlebih dahulu oleh sistem jika ada kesalahan pengisian data. Jika data sudah valid, maka sistem akan mengirim data tersebut ke *database* dan menampilkan pesan jika data sudah tersimpan.



Gambar 11. Activity diagram lengkapi data profil.

### 9. Activity diagram ganti kata sandi

*User* dapat mengganti kata sandi dengan mengisi terlebih dahulu kata sandi lama dan kata sandi baru. Sistem akan memvalidasi terlebih dahulu apakah kata sandi lama yang dimasukkan benar. Jika kata sandi lama yang dimasukkan oleh *user* benar, maka sistem akan menampilkan pesan bahwa kata sandi berhasil diganti.



Gambar 12. Activity diagram ganti kata sandi.

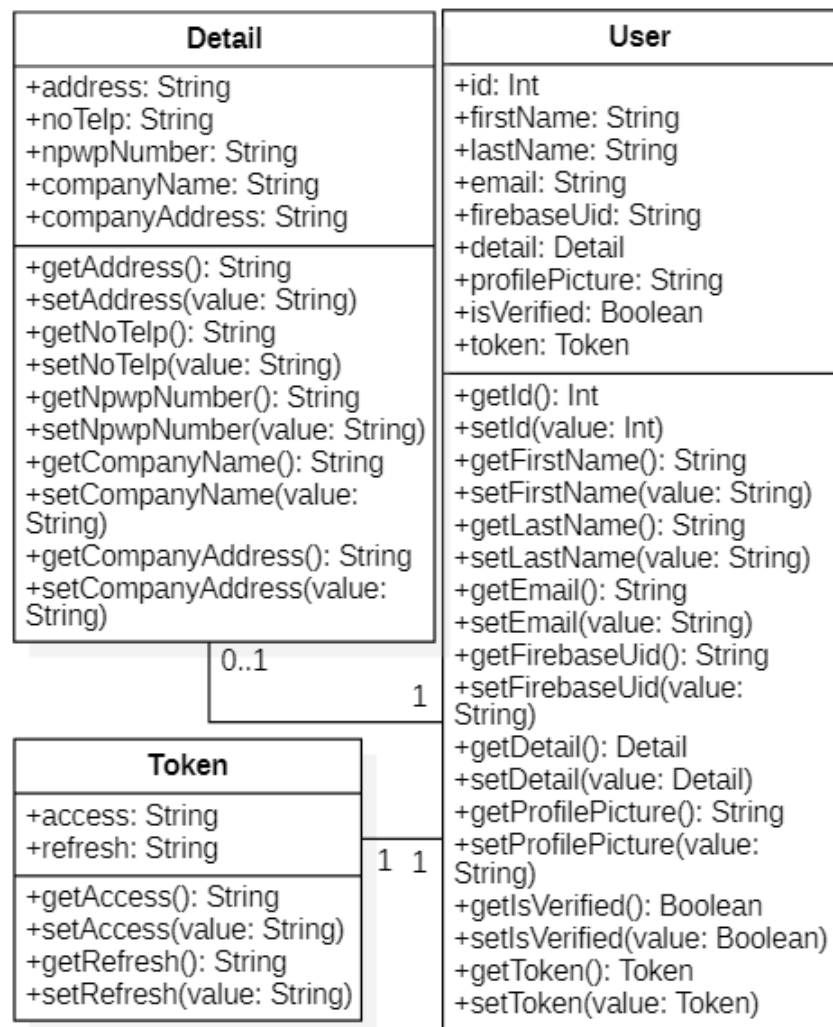
### c. Class Diagram

Sistem ini dikembangkan menggunakan paradigma arsitektur MVVM (*Model, View, ViewModel*) dan memiliki tiga objek utama, yaitu *user*, *location*, dan *ads* (Iklan). Objek-objek tersebut akan dikelola dengan *repository class* nya masing-masing, akan tetapi tidak menutup kemungkinan satu *repository* membutuhkan *repository* lainnya. *Repository* memiliki data dari tiga sumber, yaitu *network*, Room Database, dan Data Store. *Repository* nantinya akan terhubung dengan *ViewModel*, dan *ViewModel* terhubung dengan *View*. Penjelasan lebih lanjut mengenai Digiprom *Class Diagram* akan dijabarkan sebagai berikut:

#### 1. User Class Diagram

*User class* memiliki beberapa atribut di antaranya *id*, *detail*, *token* dan lain-lain. Setiap atribut memiliki *set and get function* nya masing-masing. Atribut *detail* sendiri merupakan sebuah objek *class* bernama *Detail*, yang berisi informasi detail dari *user*. Sebuah *User Class* memiliki satu *Detail Class* ataupun tidak sama sekali, tetapi sebuah *Detail Class* harus dimiliki oleh satu *User*

*Class*. Atribut token pada *User Class* merupakan representasi dari objek *Token Class*, dimana *User Class* harus memiliki satu *Token Class*. Atribut pada *Token Class* terdiri dari *access* dan *refresh*. *Access* merupakan sebuah nilai token yang digunakan untuk mengakses data dari server melalui API. *Access token* memiliki masa *expired* dengan durasi tertentu, sehingga untuk mendapatkan token baru harus mengakses sebuah API untuk mendapatkan data token yang baru. Untuk mendapatkan token baru harus menggunakan *refresh token*. Seluruh atribut dari objek *User* ini disimpan ke data lokal menggunakan *Data Store*.

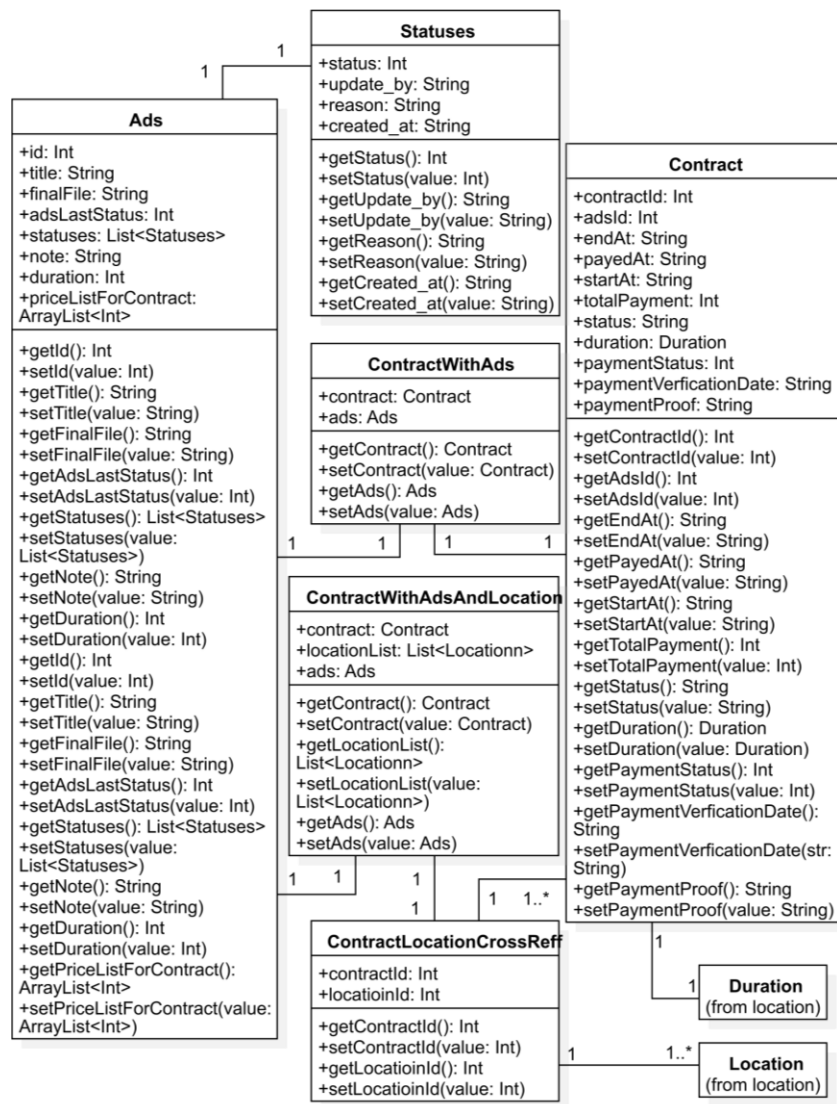


Gambar 13. *User class diagram*



## 2. Ads Class Diagram

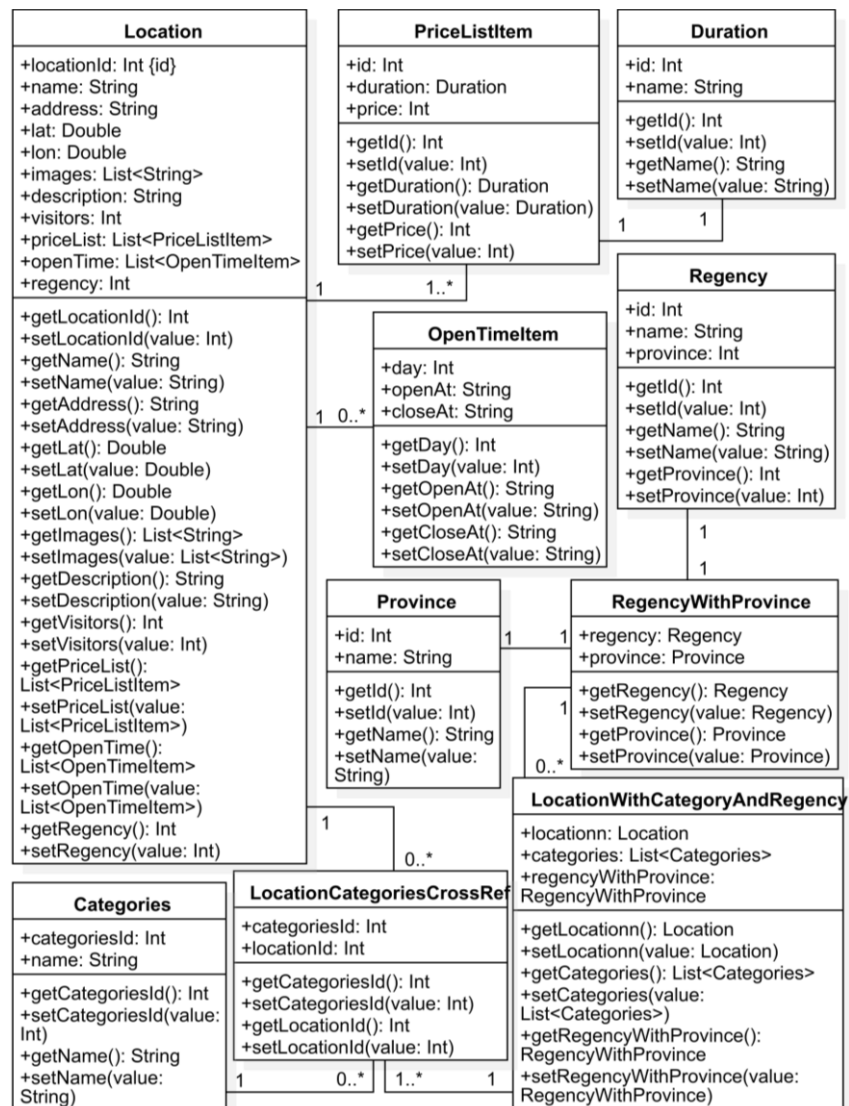
Setiap data iklan akan dimodelkan menggunakan *Ads Class* yang dapat memiliki lebih dari satu *Contract Class*. Setiap *Contract Class* dapat memiliki lebih dari satu *Location Class*, yang menggambarkan iklan dapat dipasang lebih dari di satu tempat (dicatat di kontrak iklan). Beberapa *class* tersebut dihubungkan dengan *Contract With Ads And Location Class* atau *Contract With Ads Class*. Atribut durasi pada kontrak (*Duration Class*) menggambarkan berapa lama durasi kontrak tersebut.



Gambar 14. Ads class diagram.

### 3. Location Class Diagram

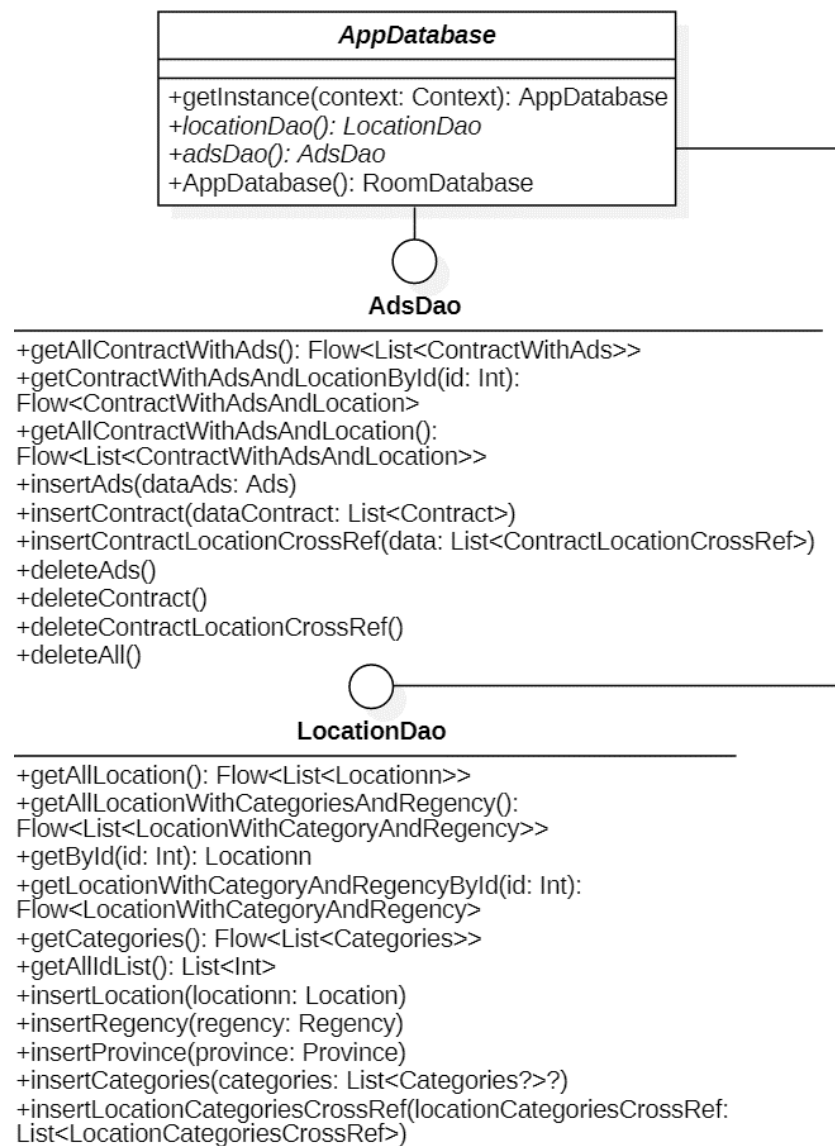
*Location Class* merupakan sebuah *class* untuk menggambarkan data dari tempat yang dapat digunakan untuk menayangkan iklan. *Categories Class* adalah kategori dari tempat tersebut (swalayan, mall, dll). Kedua *class* tersebut dihubungkan dengan *Location With Category And Regency*. *Price List Class* memodelkan data biaya pemasangan iklan dengan durasi kontrak tertentu dari sebuah tempat. Setiap *Price List* memiliki durasinya masing-masing yang digambarkan melalui *Duration Class*.



Gambar 15. Location class diagram

#### 4. Room Class Diagram

*App Database Class* digunakan untuk membuat Room Database yang akan menyimpan data lokasi dan iklan sehingga data dapat dilihat walaupun dalam keadaan *offline*. Terdapat dua *interface* yang diimplementasikan dalam *class* tersebut, yaitu *Ads Dao Interface* dan *Location Dao Interface* dimana kedua interface tersebut menyediakan berbagai fungsi yang berguna untuk menuliskan berbagai *queri* ke dalam *database*.



Gambar 16. Room class diagram

## 5. Network Class Diagram

*Api Config Class* (Gambar 17) merupakan sebuah *class* yang menangani transaksi data dengan server Digiprom. *Class* ini mengimplementasikan *Api Service Interface* yang berisi sekumpulan fungsi dan protokol untuk bertransaksi data dengan server. *Api Config* juga terhubung dengan *Data Store Helper* guna mendapatkan *access token* yang terdapat pada Data Store. Setiap fungsi pada *Api Service Class* memiliki *constructor* dan *destructor* masing-masing, yang berisi sebuah *class* untuk mengelola data JSON yang dikirim ataupun diterima. Akan tetapi kumpulan *data transfer object class* tersebut tidak akan digambarkan pada penulisan ini, dikarenakan beberapa *class* tersebut hanya digunakan untuk pengolahan data JSON saja.

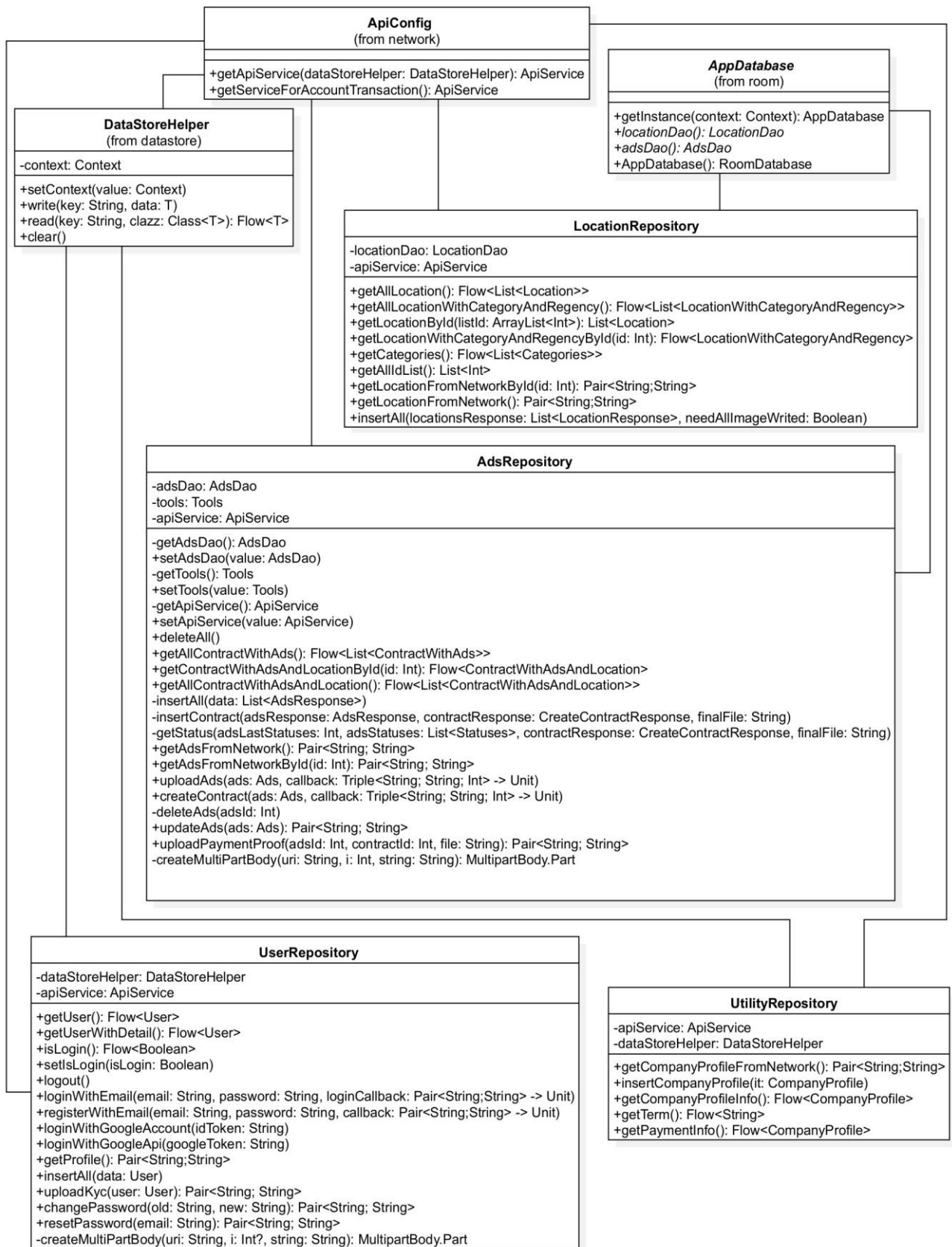


Gambar 17. Network class diagram

## 6. *Repository Class Diagram*

Terdapat tiga buah sumber data pada sistem ini, yaitu *network* yang terhubung dengan server Digiprom melalui jaringan dan dua penyimpanan data lokal (Gambar 18). Room Database yang merupakan pengembangan dari SQLite sehingga berguna untuk mengelola penyimpanan lokal dengan basis SQL, sedangkan Data Store yang merupakan pengembangan dari *library* Shared Preference sehingga berguna untuk mengelola penyimpanan lokal dengan basis *key-value pair*. Segala sesuatu pada sistem ini yang ingin terhubung dengan server maka akan menggunakan *Api Config Class*, sehingga *class* tersebut terhubung dengan setiap *repository*, dikarenakan setiap *repository* akan mendapatkan data dari *network* dalam bentuk JSON. Setelah data didapatkan dari *network*, maka nantinya setiap *repository* akan menyimpan datanya pada penyimpanan lokal.

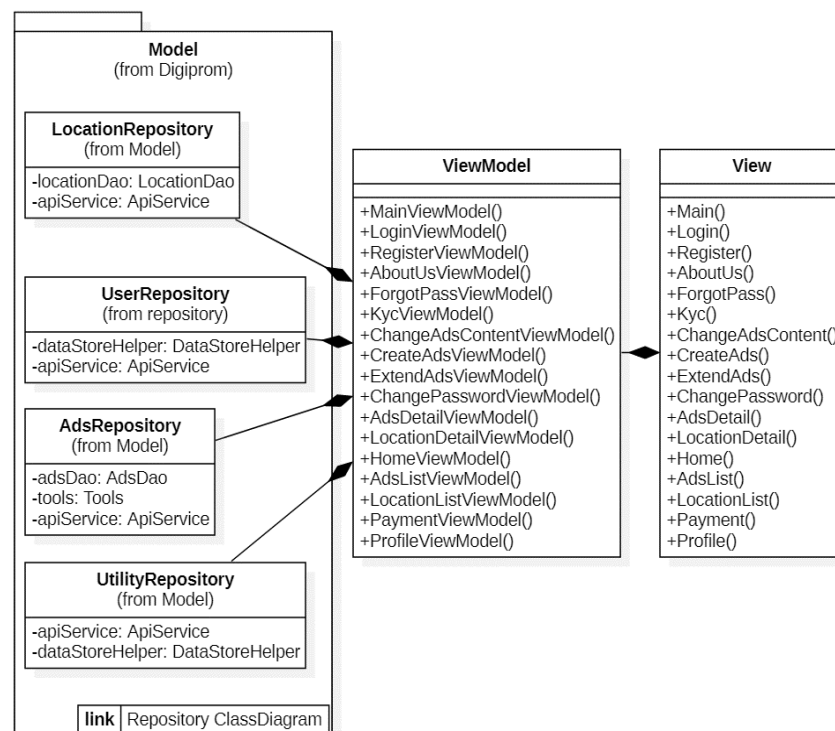
Penyimpanan data lokal dari *Location Repository Class* dan *Ads Repository Class* akan menggunakan Room Database yang diimplementasikan menggunakan *App Database Class*. *User Repository Class* dan *Utility Repository Class* menyimpan datanya secara lokal menggunakan Data Store (*key-value pair*) yang diimplementasikan menggunakan *Data Store Helper Class*. Setiap repositori akan membuat, membaca, dan memanipulasi data objeknya masing-masing. Data dari setiap objek nantinya akan disediakan oleh *repositori* masing-masing, sehingga lapisan *View Model* nantinya tidak akan pernah terhubung langsung dengan sumber-sumber data yang ada. Hal ini mempermudah pengembangan dikarenakan jika ada perubahan struktur data pada suatu objek, maka tidak akan banyak mempengaruhi struktur data dari objek yang lainnya.



Gambar 18. Repository class diagram

## 7. MVVM Class Diagram

Seperti yang sudah dijelaskan bahwa sistem ini dikembangkan dengan paradigma arsitektur MVVM. Lapisan *Model* sendiri diimplementasikan dengan empat buah *repository class* yang terhubung dengan lapisan *View Model*. Setiap repository merupakan komposisi dari *View Model*, sehingga *repository* tidak akan dibuat juga *View Model* tidak dibuat. Setiap *class* pada lapisan *View* memiliki dan mengkomposisi *View Model* nya masing-masing. Lapisan *View* tidak terhubung langsung dengan lapisan *Model*, melainkan melalui lapisan *View Model*. Lapisan *View* akan memberikan respon yang terjadi pada UI, dan lapisan *View Model* akan memberikan data yang didapat dari repository baik secara langsung maupun tidak langsung sesuai dengan kebutuhan.



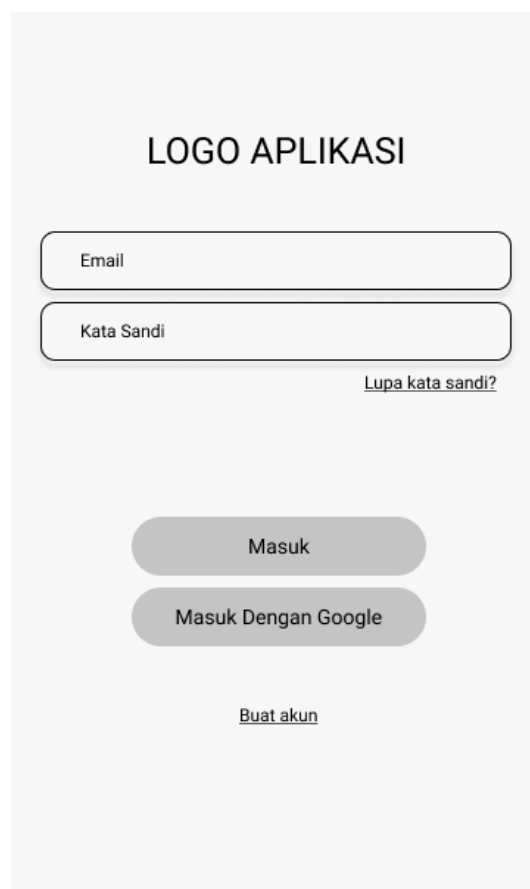
Gambar 19. MVVM class diagram

### 3.4.3. Desain *Interface*

Desain *Interface* dibuat menggunakan Figma dengan hasil yang akan disajikan pada gambar-gambar di bawah:

#### a. Halaman Masuk

*User* dapat memasukkan akun yang dimiliki melalui halaman ini (Gambar 20). Terdapat dua metode untuk masuk akun, yaitu menggunakan akun Google yang terintegrasi dengan *device* milik *user* atau menggunakan email serta kata sandi. Terdapat kolom email beserta kata sandi yang dapat diisi oleh *user*. Setelah mengisi kedua kolom tersebut, terdapat tombol “Masuk” untuk melanjutkan proses masuk akun dengan data email dan kata sandi yang sudah dimasukkan.



Gambar 20. Desain *interface* masuk akun.



## b. Halaman Daftar Akun

Halaman ini (Gambar 21) dapat digunakan oleh *user* apabila ia belum memiliki akun. *User* dapat melakukan pendaftaran dengan cara memasukkan email dan kata sandi yang akan dibuat. Kata sandi dimasukkan dua kali untuk mengkonfirmasi bahwa *user* memasukkan kata sandi yang diinginkan. Jika proses pendaftaran ini berhasil, maka *user* akan diarahkan ke halaman untuk melengkapi data profil. Apabila *user* sudah memiliki akun, *user* dapat menekan tombol “punya akun” dan akan diarahkan ke halaman masuk. *User* juga dapat langsung menekan tombol “Masuk dengan Google” apabila ingin masuk menggunakan akun Google yang terintegrasi dengan *device* tersebut.

LOGO APLIKASI

Email

Kata Sandi

Ulangi Kata Sandi

S&K

Daftar

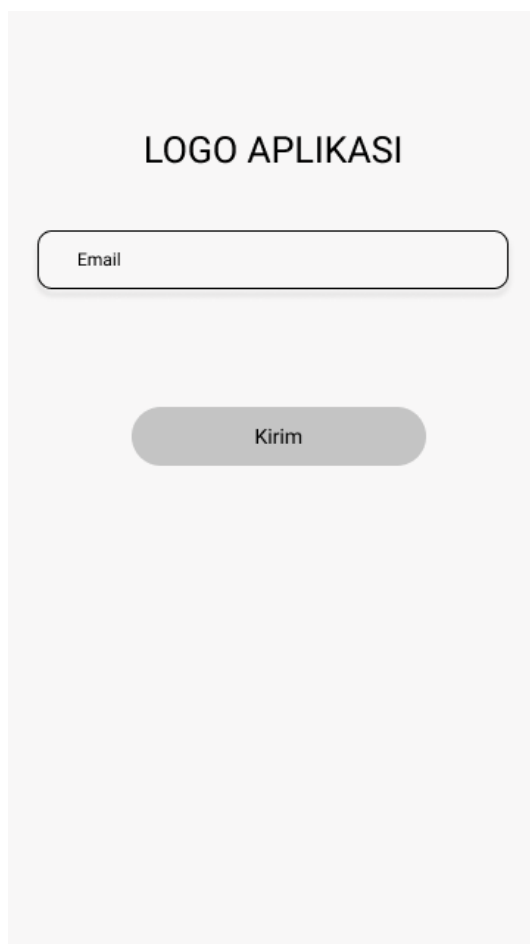
Masuk Dengan Google

[Punya akun](#)

Gambar 21. Desain *interface* daftar akun.

### c. Halaman Lupa Kata Sandi

Jika *user* lupa terhadap kata sandi dari akunnya, maka *user* dapat menyetel ulang kata sandi dengan cara mengakses halaman lupa kata sandi (Gambar 22). Halaman ini dapat diakses melalui halaman masuk. *User* dapat mengisi email dari akun yang dimilikinya dan kemudian menekan tombol “Kirim”. Apabila email yang dimasukkan oleh *user* ditemukan, maka akan memunculkan pesan bahwa *link* untuk menyetel ulang kata sandi sudah dikirimkan ke alamat email yang dimiliki oleh *user* dan aplikasi mengarahkan kembali ke halaman masuk. Proses penggantian kata sandi tersebut dapat dilakukan *user* dengan cara mengakses *link* tersebut.

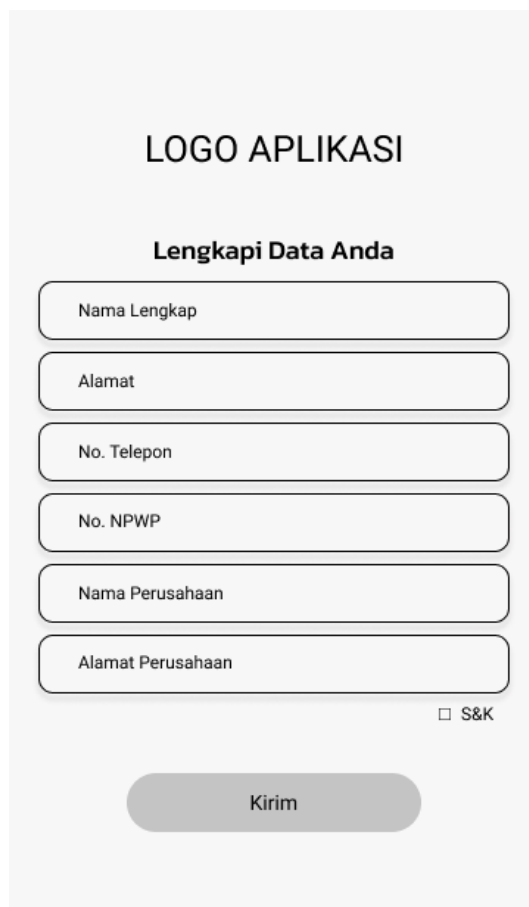


The image shows a mobile application interface for password recovery. At the top center, the text "LOGO APLIKASI" is displayed. Below this is a rounded rectangular input field with the placeholder text "Email". Further down is a rounded rectangular button with the text "Kirim". The entire interface is set against a light gray background.

Gambar 22. Desain *interface* lupa kata sandi.

d. Halaman Melengkapi Data Profil

*User* dapat melengkapi data profil melalui halaman ini (Gambar 23). Halaman ini dapat diakses setelah *user* melakukan pendaftaran. Selain dari halaman tersebut, *user* juga dapat mengakses halaman ini melalui halaman profil. Data profil yang dimaksud adalah data pribadi dari *user* seperti nama, alamat, nomor telepon, nomor NPWP dan data perusahaan yang produk atau jasanya akan diiklankan. Terdapat juga *checkbox* yang menyatakan persetujuan *user* terhadap syarat dan ketentuan yang berlaku. Apabila *user* ingin membaca terlebih dahulu syarat dan ketentuannya, maka *user* dapat menekan tombol “S&K” yang ada di sebelah *checkbox*. Semua data tersebut wajib diisi oleh *user*.



LOGO APLIKASI

**Lengkapi Data Anda**

Nama Lengkap

Alamat

No. Telepon

No. NPWP

Nama Perusahaan

Alamat Perusahaan

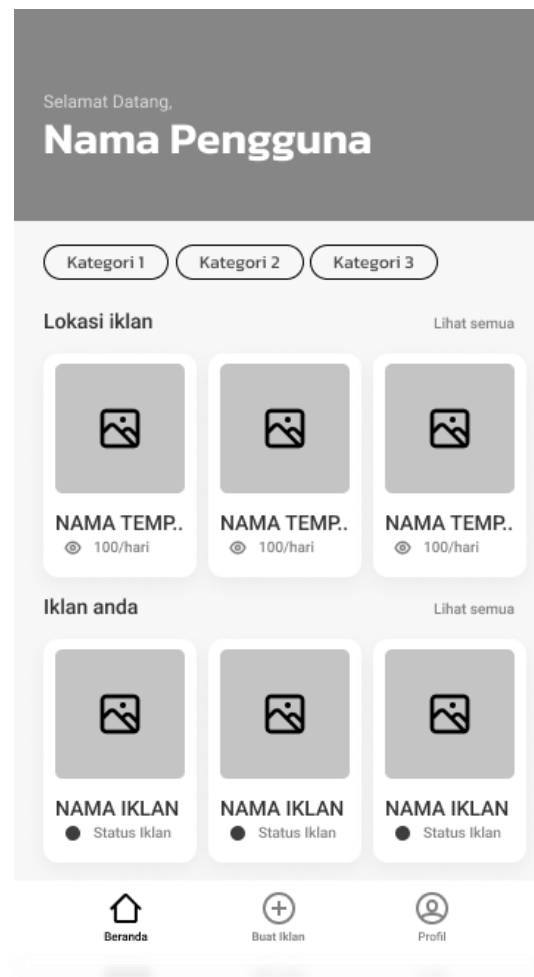
S&K

Kirim

Gambar 23. Desain *interface* melengkapi data profil.

## e. Halaman Beranda

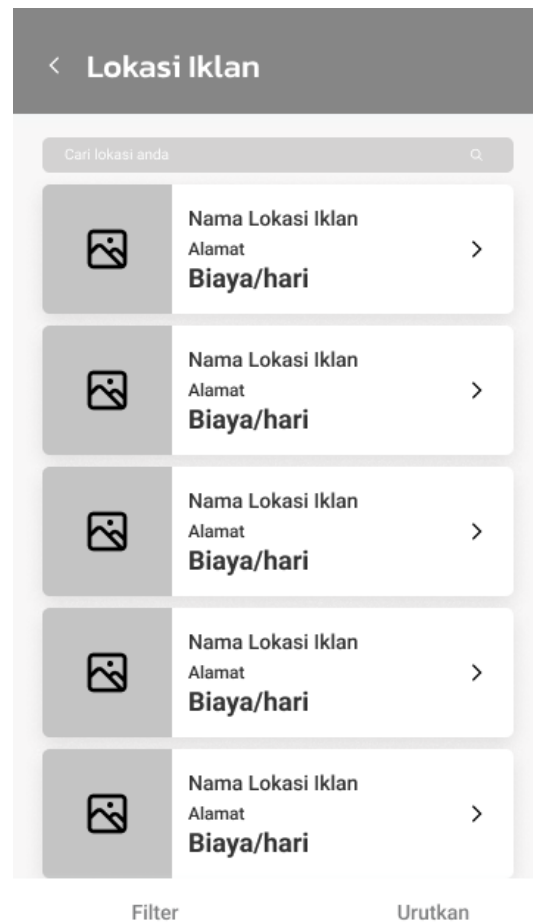
Pada bagian atas halaman ini (Gambar 24) terdapat kalimat “Selamat Datang” yang diikuti oleh nama dari *user*. Kemudian terdapat *list* kategori lokasi iklan yang apabila ditekan, maka akan menampilkan halaman *list* lokasi iklan dengan *filter* kategori tersebut. Tampilan selanjutnya terdapat informasi berupa tempat-tempat yang dapat dipasangkan iklan oleh *user* dan juga iklan-iklan yang dimiliki oleh *user*. Pada bagian bawah terdapat *bottom navigation* yang berguna untuk menavigasikan halaman beranda, halaman buat iklan, dan halaman profil.



Gambar 24. Desain *interface* beranda.

f. Halaman *List* Lokasi Iklan

Halaman ini berisi informasi mengenai tempat-tempat yang dapat digunakan oleh *user* untuk memasang iklannya (Gambar 25). Terdapat juga pencarian, *filter* dan *sorting* yang dapat diterapkan ke dalam *list* lokasi iklan.

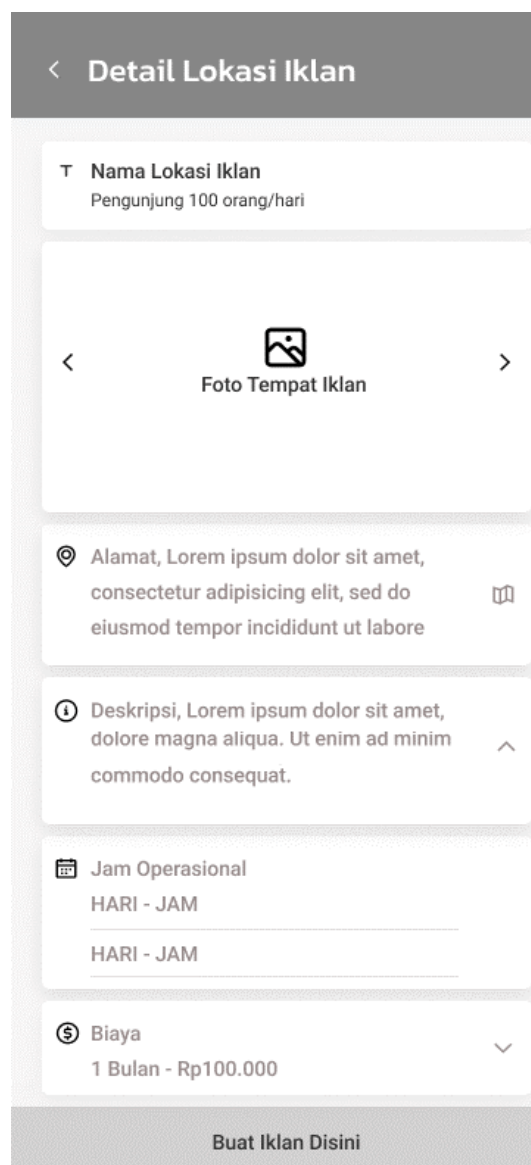


Gambar 25. Desain *interface list* lokasi iklan.

g. Halaman Detail Lokasi Iklan

Informasi detail dari lokasi iklan ditampilkan pada halaman ini (Gambar 26) dan dapat diakses dengan cara menekan salah satu *item* pada halaman *list* lokasi iklan. Pada bagian atas terdapat nama dari tempat tersebut beserta estimasi jumlah pengunjung per harinya.

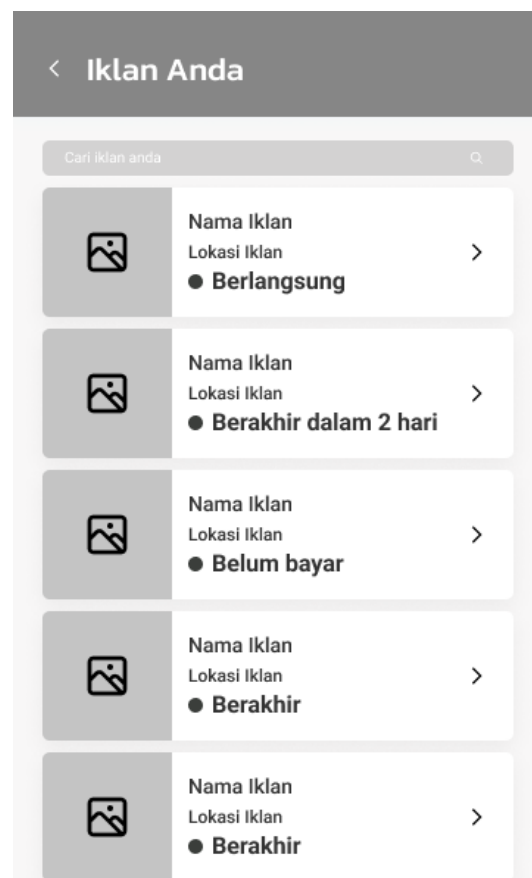
Foto-foto dari tempat iklan ditampilkan menggunakan *horizontal scroll viewpager*. Terdapat informasi alamat dari lokasi iklan, beserta tombol dengan gambar peta yang apabila ditekan maka aplikasi akan mengarahkan ke aplikasi Google Maps dan menunjukkan tempat tersebut pada map. Kolom deskripsi berisi informasi singkat mengenai tempat tersebut, diikuti oleh informasi jam operasional (per hari) beserta biaya dari pemasangan iklan di tempat tersebut.



Gambar 26. Desain *interface* detail tempat iklan.

#### h. Halaman *List* Iklan

Halaman ini berisi *list* dari iklan yang dimiliki oleh *user* (Gambar 27). Terdapat informasi berupa nama iklan, lokasi iklan, dan status dari iklan tersebut (apakah sedang tayang atau lainnya). Apabila *user* menekan salah satu *item* dari *list* tersebut, maka aplikasi akan menampilkan halaman detail dari iklan tersebut.

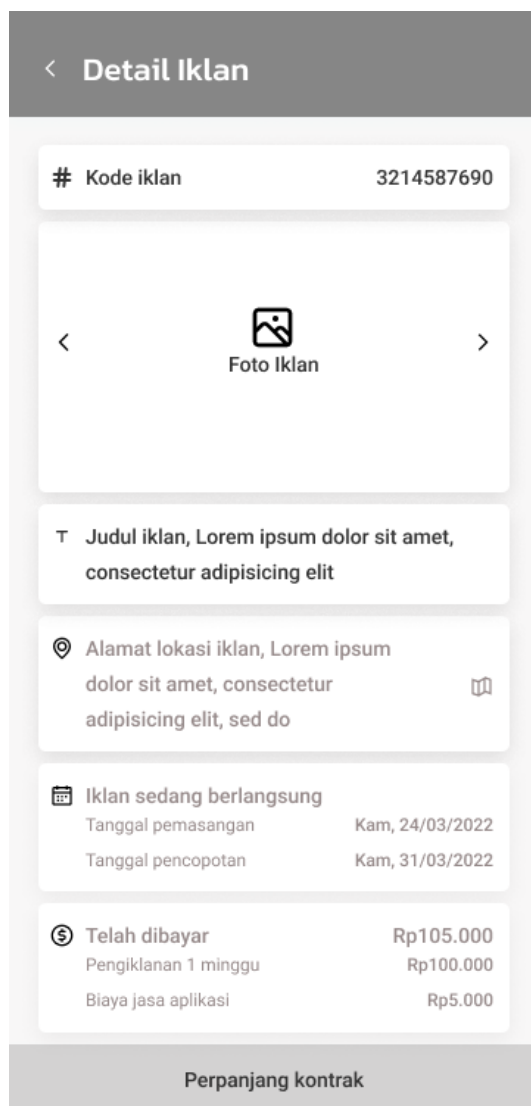


Gambar 27. Desain *interface list* iklan.

#### i. Halaman Detail Iklan

Halaman ini berisi detail dari iklan dan kontrak iklan tersebut (Gambar 28). Di bagian atas berisi nama, kode kontrak iklan dan diikuti oleh konten dari iklan tersebut. Konten tersebut dapat berupa foto atau video. Terdapat juga informasi tempat-tempat penayangan

iklan. Bagian status iklan berisi informasi dari status iklan tersebut beserta tanggal penayangan dan tanggal pencopotan konten iklan. Fungsi tombol pada bagian bawah menyesuaikan dengan status iklan.



Gambar 28. Desain *interface* detail iklan.

#### j. Halaman Buat Iklan

Untuk membuat iklan, *user* dapat mengakses halaman ini (Gambar 29) melalui halaman beranda atau halaman detail lokasi iklan. *User* dapat memasukkan informasi-informasi yang dibutuhkan untuk



membuat iklan. Pada bagian atas terdapat *view* untuk memilih konten iklan menggunakan *device file explorer* atau semacamnya. *User* dapat memasukkan judul dari iklan dan memilih tempat-tempat dimana iklan tersebut akan ditayangkan. Jika *user* sudah memilih tempat-tempat untuk penayangan iklan, *view* durasi akan menyesuaikan dengan menampilkan pilihan durasi kontrak penayangan iklan beserta biaya yang dibutuhkan.

**< Buat Iklan**

**< Foto/video dipilih >**

**T Judul iklan**  
Tulis judul iklan anda 100

**📍 Lokasi iklan**

**Nama lokasi iklan 1** ⓘ

1 minggu	Rp 100.000
1 bulan	Rp 1.000.000
3 bulan	Rp 3.000.000
6 bulan	Rp 6.000.000
1 tahun	Rp 10.000.000

**Nama lokasi iklan 2** ⓘ

1 minggu	Rp 100.000
1 bulan	Rp 1.000.000
3 bulan	Rp 3.000.000
6 bulan	Rp 6.000.000
1 tahun	Rp 10.000.000

Pilih lokasi iklan >

**🕒 Durasi**  
Pilih durasi >

**Berikutnya**


Gambar 29. Desain *interface* buat iklan.

### k. Halaman Pembayaran

Terdapat ringkasan informasi dari iklan yang telah dibuat dan detail pembayaran (Gambar 30). Informasi detail pembayaran berisi data tentang jumlah pembayaran, rekening tujuan untuk pembayaran dan tombol *copy* pada masing-masing data tersebut. Pada bagian bawah halaman terdapat tombol “Kirim bukti pembayaran” yang apabila ditekan maka *user* dapat memilih bukti pembayaran untuk diunggah dan diperiksa oleh admin.

**< Ringkasan Pembayaran**


# Kode iklan 3214587690

Judul iklan anda 

Nama lokasi iklan  
Alamat

Tanggal pemasangan  
Pemasangan Kam, 24/03/2022  
Pencopotan Kam, 31/03/2022

Biaya  
Pengiklanan 1 minggu Rp100.000  
Biaya jasa aplikasi Rp5.000  
Total Rp105.000

 Nama Bank  
No. Rekening  
1234 5678 9087 1254 SALIN

Silakan lakukan pembayaran dengan transfer sejumlah total biaya ke no. rekening diatas sebelum hari Rabu, 23/03/2022.

Pembayaran anda akan dikonfirmasi setelah anda mengirim bukti pembayaran sebelum hari Kamis, 24/03/2022.

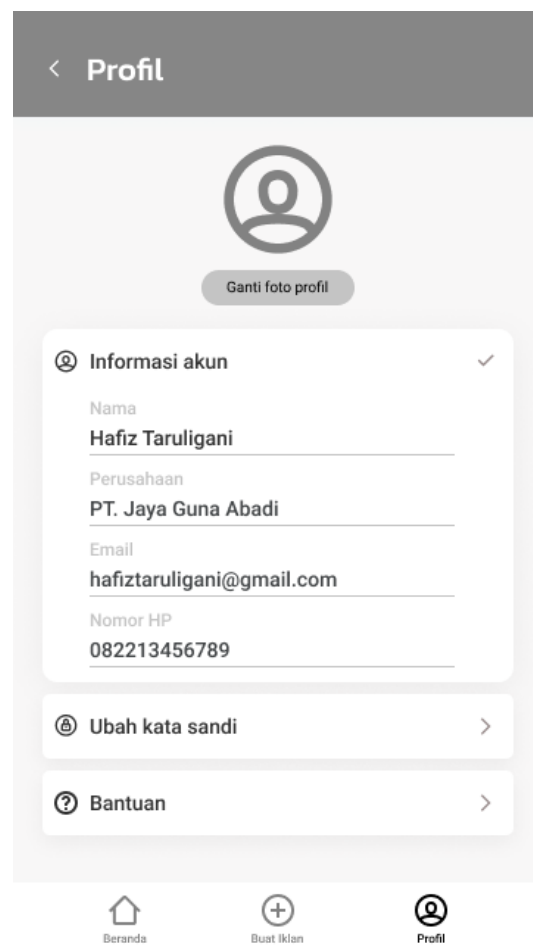
Pastikan anda tidak terlambat melakukan pembayaran dan mengirim bukti pembayaran.

**Kirim Bukti Pembayaran**

Gambar 30. Desain *interface* pembayaran.

## 1. Halaman Profil

Pada halaman profil (Gambar 31) berisi informasi mengenai data akun. Apabila *user* ingin mengubah kata sandi akun, maka hal tersebut dapat dilakukan dengan menekan tombol “Ubah sandi” yang terdapat pada halaman ini. Informasi bantuan dan tentang Digiprom dapat diakses dengan menekan tombol “Bantuan”.

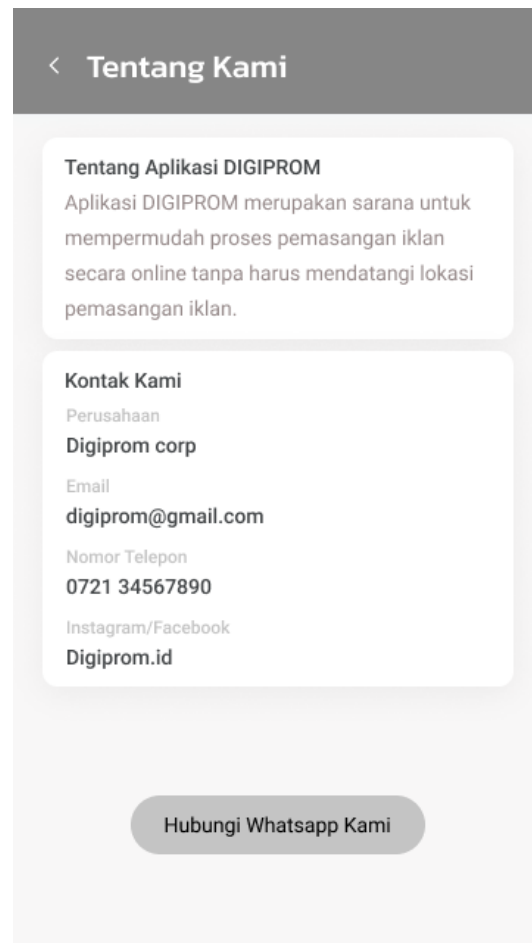


Gambar 31. Desain *interface* profil.

## m. Halaman Bantuan

Halaman ini berisi informasi singkat mengenai Digiprom beserta kontak yang dapat dihubungi oleh *user* apabila *user* membutuhkan bantuan. Terdapat tombol “Hubungi Whatsapp Kami” yang apabila

ditekan maka akan mengarahkan ke aplikasi Whatsapp dengan pengiriman pesan ke kontak Digiprom.



Gambar 32. Desain *interface* bantuan.

#### 3.4.4. Penulisan Kode Program

Bahasa yang akan digunakan dalam penulisan kode program adalah Kotlin dengan *database* lokal menggunakan *library* Room dan *database Structured Query Language (SQL)* server yang dikelola rekan tim lainnya. Penulisan program akan dilakukan dengan menggunakan arsitektur *model view viewmodel (MVVM)* sehingga bentuk data akan lebih terstruktur dan kode program tersusun rapi. Hal ini akan memudahkan *developer* dalam melakukan *maintenance* pada aplikasi. Aplikasi akan berjalan baik apabila tidak terjadinya *bottleneck*,

sehingga akan digunakan Coroutine pada beberapa bagian untuk mengontrol jalannya kode program.

### 3.4.5. Pengujian Program

Pengujian aplikasi dilakukan untuk memastikan bahwa aplikasi yang dibuat telah sesuai dengan kebutuhan dan dapat berjalan dengan baik. Pada pengujian sistem ini dilakukan dengan dua metode, yaitu metode *black box testing* dan metode *user acceptance testing* (UAT). Metode pengujian *black box* dilakukan dengan rancangan pengujian yang disajikan pada Tabel 5.

Tabel 5. Rancangan Pengujian Sistem

Kelas Uji	Skenario Pengujian	Hasil Pengujian yang Diharapkan
Daftar	Mengosongkan data	Menampilkan pesan <i>error</i> data tersebut harus diisi
	Memasukkan kata sandi kurang dari 8 karakter	Menampilkan pesan <i>error</i> kata sandi harus lebih dari 7 karakter
	Memasukkan kata sandi konfirmasi yang berbeda dengan kata sandi	Menampilkan pesan <i>error</i> konfirmasi kata sandi tidak sama
	Mengisi data valid	Menampilkan pesan berhasil Mendaftar, kemudian ke halaman beranda
Masuk	Mengosongkan data	Menampilkan pesan <i>error</i> data tersebut harus diisi
	Memasukkan <i>email</i> atau kata sandi salah	Menampilkan pesan <i>email</i> atau kata sandi salah
	Memasukkan <i>email</i> atau kata sandi benar	Menampilkan pesan berhasil masuk, kemudian ke halaman beranda
Lupa Kata sandi	Mengosongkan data	Menampilkan pesan <i>error</i> data tersebut harus diisi
	Mengisi <i>email</i> yang tidak terdaftar	Menampilkan pesan <i>email</i> tidak ditemukan
	Mengisi <i>email</i> yang terdaftar	Menampilkan pesan <i>link</i> untuk reset kata sandi sudah dikirimkan ke <i>email</i>

Tabel 5 (lanjutan)

Kelas Uji	Skenario Pengujian	Hasil Pengujian yang Diharapkan
Beranda	Menekan salah satu Menu kategori	Diarahkan ke <i>list</i> lokasi iklan dengan filter kategori yang di klik
	Menekan tombol "lihat semua" pada tampilan lokasi iklan	Diarahkan ke <i>list</i> lokasi iklan
	Menekan salah satu item <i>list</i> lokasi iklan	Diarahkan ke halaman detail lokasi iklan tersebut
	Menekan tombol "lihat semua" pada tampilan iklan <i>user</i>	Diarahkan ke <i>list</i> iklan <i>user</i>
	Menekan salah satu item iklan <i>user</i>	Diarahkan ke halaman detail iklan <i>user</i> tersebut
<i>List</i> Lokasi Iklan	Memasukkan <i>string</i> valid pada kolom pencarian	Menampilkan <i>list</i> lokasi sesuai dengan <i>string</i> yang dimasukkan
	Menambahkan filter kategori	Menampilkan <i>list</i> lokasi sesuai dengan kategori yang ditambahkan
	Menambahkan filter rentang harga	Menampilkan <i>list</i> lokasi sesuai dengan rentang harga yang ditambahkan
	Mengurutkan <i>list</i> berdasarkan kondisi tertentu	Menampilkan <i>list</i> lokasi diurutkan berdasarkan kondisi tersebut
	Menekan salah satu item <i>list</i> lokasi iklan	Diarahkan ke halaman detail lokasi iklan tersebut, jika kembali maka <i>list</i> iklan masih sesuai dengan filter dan <i>sorting</i> sebelumnya
Detail Lokasi Iklan	Menggeser foto <i>preview</i> lokasi iklan	Foto ditampilkan dengan baik
	Menekan tombol map pada tampilan alamat	Diarahkan ke aplikasi map dengan fokus koordinat lokasi iklan tersebut
	Menekan tombol "Buat Iklan Disini"	Diarahkan ke halaman pembuatan iklan dengan kondisi lokasi tersebut sudah tercantumkan pada formulir pembuatan iklan

Tabel 5 (lanjutan)

Kelas Uji	Skenario Pengujian	Hasil Pengujian yang Diharapkan
Buat Iklan	Mengosongkan data	Menampilkan pesan <i>error</i> data tersebut harus diisi
	Menekan tombol "Tambah Lokasi Iklan"	Diarahkan ke halaman <i>list</i> lokasi iklan dengan <i>checkbox</i> sebagai tanda lokasi yang dipilih
	Menampilkan durasi	Menampilkan durasi dan total biaya sesuai dengan lokasi - lokasi yang telah dipilih
	Menekan tombol berikutnya	Mengunggah data formulir iklan dan Menampilkan pesan data berhasil diunggah
Pembayaran	Menekan tombol salin pada tampilan nomor rekening bank	nomor rekening bank tersalin ke <i>clipboard user</i>
	Menekan tombol salin pada tampilan total pembayaran	total pembayaran tersalin ke <i>clipboard user</i>
	Menekan tombol "kirim bukti pembayaran"	Diarahkan untuk memilih bukti pembayaran
	Menekan tombol edit pada tampilan konfirmasi	Diarahkan untuk memilih kembali bukti pembayaran
	Menekan tombol konfirmasi pada tampilan konfirmasi	<i>file</i> bukti pembayaran diunggah. Setelah berhasil dan selesai, Menampilkan pesan bukti pembayaran berhasil dikirim
Detail Iklan	Menggeser <i>file preview</i> konten iklan	<i>file</i> konten iklan ditampilkan dengan baik
	Menampilkan halaman sesuai dengan status iklan	tampilan sesuai dengan status iklan
	Menekan tombol <i>map</i> pada tampilan alamat	Diarahkan ke aplikasi <i>map</i> dengan fokus koordinat lokasi iklan tersebut
Profil	Menampilkan halaman sesuai dengan kondisi apakah data profil sudah dilengkapi	Tampilan sesuai dengan kondisi apakah data profil sudah dilengkapi
	Menekan tombol lengkapi data profil	Diarahkan ke halaman untuk melengkapi data profil

Tabel 5 (lanjutan)

Kelas Uji	Skenario Pengujian	Hasil Pengujian yang Diharapkan
	Menekan tombol ubah kata sandi	Diarahkan ke halaman ubah kata sandi
	Menekan tombol bantuan	Diarahkan ke halaman bantuan
	Menekan tombol keluar	Menampilkan pesan berhasil keluar dan Diarahkan ke halaman masuk
Lengkapi Data Profil	Mengosongkan data	Menampilkan pesan <i>error</i> data tersebut harus diisi
	Menekan tombol "kirim"	Menampilkan pesan data berhasil dikirim
Ubah Kata Sandi	Mengosongkan data	Menampilkan pesan <i>error</i> data tersebut harus diisi
	Mengisi kolom kata sandi lama yang salah	Menampilkan pesan kata sandi lama salah
	Memasukkan kata sandi baru kurang dari 8 karakter	Menampilkan pesan <i>error</i> kata sandi harus lebih dari 7 karakter
	Memasukkan kata sandi konfirmasi yang berbeda dengan kata sandi baru	Menampilkan pesan <i>error</i> konfirmasi kata sandi tidak sama
	Mengisi data valid	Menampilkan pesan kata sandi berhasil diganti
Bantuan	Menekan tombol "Hubungi Whatsapp kami"	Diarahkan ke aplikasi Whatsapp dengan pengiriman pesan ke nomor Digiprom

Pengujian UAT dilakukan untuk mengetahui kebermanfaatan Aplikasi Digiprom yang sudah dibangun terhadap bisnis yang dijalankan. Pengujian ini akan diimplementasikan dengan memberikan pertanyaan-pertanyaan atau kuesioner (Tabel 6) kepada pemilik Digiprom itu sendiri yang nantinya akan dihitung persentase rata-rata dari hasil pengisian kuesioner tersebut.



Tabel 6. Kuesioner UAT

<b>No</b>	<b>Pertanyaan</b>
1	Apakah tampilan dari aplikasi menarik?
2	Apakah alur penggunaan aplikasi ini mudah dipahami?
3	Apakah informasi data pada aplikasi ini ditampilkan dengan baik? (Seperti ukuran tulisan, warna tulisan, dll)
4	Apakah informasi data pada aplikasi ini bersifat informatif?
5	Apakah aplikasi ini membantu proses pemasangan iklan?
6	Apakah fitur yang tersedia sudah memadai?
7	Apakah fitur aplikasi ini sesuai dengan kebutuhan bisnis Digiprom?
8	Apakah aplikasi ini nyaman digunakan?

## V. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Setelah hasil penelitian selesai, terdapat beberapa poin kesimpulan yang didapat, yaitu:

1. Aplikasi *Mobile* Periklanan Digiprom Berbasis Android telah berhasil diselesaikan, dibangun menggunakan bahasa Kotlin dengan fitur-fitur yang membantu interaksi pebisnis dengan Digiprom.
2. Hasil *Black Box Testing* menunjukkan fungsionalitas aplikasi berjalan sepenuhnya dengan tingkat keberhasilan dari semua skenario adalah 100%.
3. Hasil *User Acceptance Testing* menunjukkan bahwa aplikasi cukup diterima atau cukup bermanfaat bagi Digiprom.
4. Belum adanya *dark mode user interface*, sehingga pengguna hanya bisa menggunakan satu *user interface mode*.
5. Jika terjadi kesalahan jaringan (*network exception*), maka pesan kesalahan yang ditampilkan oleh aplikasi masih belum spesifik mengenai penyebab dari kesalahan tersebut.

### 5.2. Saran

Berdasarkan kesimpulan yang didapat, masih terdapat beberapa kekurangan pada penelitian ini, sehingga diperlukan beberapa perbaikan pada aplikasi, di antaranya:

1. Penambahan *dark mode user interface* agar pengguna dapat menggunakan aplikasi dengan mode tampilan gelap.
2. *Result and exception handling* pada beberapa fungsi *networking* di dalam kode program perlu disempurnakan lagi sehingga apabila terjadi kesalahan, maka pesan kesalahan yang ditampilkan akan lebih spesifik.

## **DAFTAR PUSTAKA**

## DAFTAR PUSTAKA

- Abraham, J., & Ismail, I. E. (2021). *Unit Testing dan User Acceptance Testing pada Sistem Informasi Pelayan Kategorial Pelayanan Anak*.
- Bhuvaneswari, T., & Prabakaran, S. (2013). A Survey on Software Development Life Cycle Models. *International Journal of Computer Science and Mobile Computing*, 2(May), 262–267.
- Dima, A. M., & Maassen, M. A. (2018). From waterfall to agile software: Development models in the IT sector, 2006 to 2018. impacts on company management. *Journal of International Studies*, 11(2), 315–326. <https://doi.org/10.14254/2071-8330.2018/11-2/21>
- Google, C. (2016). *Android Developer Fundamental Course: Practical Workbook*.
- Google, C. (2022a, Februari 2). *Kotlin coroutines on Android | Android Developers*. <https://developer.android.com/Kotlin/coroutines>
- Google, C. (2022b, Juni 7). *Room | Android Developers*. <https://developer.android.com/training/data-storage/room>
- Hambling, B., & Goethem, P. van. (2013). *User Acceptance Testing: A step-by-step guide*. BCS Learning and Development Ltd.
- Hasanah, F. N., & Untari, R. S. (2020). Buku Ajar Rekayasa Perangkat Lunak. Dalam M. Suryawinata (Ed.), *Buku Ajar Rekayasa Perangkat Lunak*. UMSIDA PRESS. <https://doi.org/10.21070/2018/978-602-5914-09-6>
- Hendini, A. (2016). *Pemodelan Uml Sistem Informasi Monitoring Penjualan dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak): Vol. IV (Issue DESEMBER)*.
- Kadir, A., & Triwahyuni, T. Ch. (2014). *Pengantar Teknologi Informasi Edisi Revisi (Issue August)*. Andi Yogyakarta. <https://doi.org/10.13140/2.1.4734.7840>
- Kriyantono, R. (2013). *Manajemen Periklanan: Teori dan Praktik*. Universitas Brawijaya Press.

- Mallick, B., & Das, N. (2013). An Approach to Extended Class Diagram Model of UML for Object Oriented Software Design. *International Journal of Innovative Technology & Adaptive Management (IJITAM)*, 1(2).
- Sembiring, R. (2014). *Pengantar Bisnis*. La Goods Publishing.
- Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Native Android Menggunakan Bahasa Pemrograman Java dan Kotlin. *Researchget*, December.  
[https://www.researchgate.net/publication/329525878\\_Analisis\\_Performa\\_Aplikasi\\_Android\\_Pada\\_Bahasa\\_Pemrograman\\_Java\\_dan\\_Kotlin](https://www.researchgate.net/publication/329525878_Analisis_Performa_Aplikasi_Android_Pada_Bahasa_Pemrograman_Java_dan_Kotlin)
- Snadhika, T. J. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 03(02), 45–48. <https://doi.org/10.30591/jpit.v3i1.647>
- Wazlawick, R. S. (2014). *Object-Oriented Analysis and Design for Information Systems*. Elsevier.