

**PEMBANGUNAN APLIKASI TUR INTERAKTIF
PERUMAHAN BALIMBINGAN PERMAI PT. KARYA PROPERTINDO
UTAMA BERBASIS VIRTUAL REALITY**

SKRIPSI

Keminatan Multimedia, Game, dan Mobile

Disusun oleh

REINHARD JONATHAN SLAHI

175150200111040



**TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2020

DAFTAR ISI

BAB 1	PENDAHULUAN	6
1.1	Latar Belakang	6
1.2	Identifikasi Masalah	9
1.3	Rumusan Masalah	10
1.4	Tujuan Penelitian.....	10
1.5	Manfaat Penelitian	11
1.6	Batasan Penelitian/Ruang Lingkup Penelitian	12
1.7	Sistematika Pembahasan/Laporan.....	12
BAB 2	LANDASAN KEPUSTAKAAN	14
2.1	KAJIAN PUSTAKA	14
2.2	Virtual Reality	16
2.3	Unity 3D.....	16
2.4	Android.....	17
2.5	Pengujian Fungsional	17
2.6	Pengujian <i>Usability</i>	17
BAB 3	METODOLOGI PENELITIAN	18
3.1	Studi Literatur.....	19
3.2	Analisis Kebutuhan Sistem	20
3.3	Perancangan Sistem	20
3.4	Implementasi Sistem	20
3.5	Pengujian Sistem	21
3.6	Kesimpulan dan Saran	21
BAB 4	REKAYASA KEBUTUHAN	22
4.1.	Gambaran Umum Aplikasi	22
4.1.1.	Deskripsi.....	22
4.1.2.	Cara Penggunaan	22
4.2.	Analisis Kebutuhan Sistem	22
4.2.1	Identifikasi Aktor	23

4.2.2	Kebutuhan Fungsional	23
4.2.3	Kebutuhan Non Fungsional.....	24
4.3.	Pemodelan Kebutuhan	24
4.3.1.	<i>Use Case Diagram</i>	25
4.3.2.	<i>Use case Scenario</i>	25
BAB 5	PERANCANGAN DAN IMPLEMENTASI	27
5.1	Perancangan.....	27
5.1.1	Perancangan Arsitektur Sistem.....	27
5.1.2	Activity Diagram.....	28
5.1.3	<i>Sequence Diagram</i>	31
5.1.4	Class Diagram.....	33
5.1.5	Perancangan Basis Data	33
5.1.6	Perancangan Algoritme	35
5.1.7	Perancangan User Interface	37
5.2	Implementasi Perangkat Lunak.....	37
DAFTAR PUSTAKA	51
LAMPIRAN	53

DAFTAR GAMBAR

Gambar 3.1 Metode Pengembangan.....	19
Gambar 4.1 <i>Use Case Diagram</i> Pengguna	25
Gambar 5.1 Rancangan Arsitektur Sistem	27
Gambar 5.2 Rancangan <i>Activity Diagram</i> Menampilkan Daftar Rumah	29
Gambar 5.3 Rancangan <i>Activity Diagram</i> Menampilkan Tur Interaktif	30
Gambar 5.4 Sequence Diagram Menampilkan Daftar Rumah	31
Gambar 5.5 Sequence Diagram Menampilkan Tur Interaktif.....	32
Gambar 5.6 Sequence Diagram Menampilkan Daftar Rumah (API Service)	32
Gambar 5.7 <i>Class Diagram</i> API Service.....	34
Gambar 5.8 <i>Class Diagram</i> Aplikasi Android	34
Gambar 5.9 Rancangan Basis Data dalam ERD	35
Gambar 5.10 Hasil Perancangan Kerangka Antarmuka	37

DAFTAR TABEL

Tabel 4.1 Identifikasi Aktor	23
Tabel 4.2 Penjelasan Format Kode Kebutuhan.....	23
Tabel 4.3 Kebutuhan Fungsional Sistem	23
Tabel 4.4 Kebutuhan Non-Fungsional Sistem	24
Tabel 4.5 Use Case Scenario Menampilkan Daftar Rumah	26
Tabel 4.6 Use Case Scenario Menampilkan 3D Tour	26
Tabel 5.1 Rancangan Tabel <i>estate</i>	35
Tabel 5.2 Rancangan Algoritme Mengunduh Model 3 Dimensi Rumah	35
Tabel 5.3 Rancangan Algoritme Mendeteksi Objek Perabotan.....	36
Tabel 5.4 Kode Program HomeActivity (Android)	37
Tabel 5.5 Kode Program HomePresenter (Android).....	39
Tabel 5.6 Kode Program EstateAdapter (Android)	39
Tabel 5.7 Kode Program EstateApiService (Android)	40
Tabel 5.8 Kode Program Main (Unity)	40
Tabel 5.9 Kode Program FirstPersonController (Unity)	41
Tabel 5.10 Kode Program API untuk mengunduh model 3 dimensi rumah (Unity)	43
Tabel 5.11 Kode Program Inventory Perabotan (Unity)	45
Tabel 5.12 Kode Program Main.go (API Service)	47
Tabel 5.13 Kode Program EstateController.go (API Service)	48
Tabel 5.14 Kode Program Estate.go (API Service)	49

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemasaran merupakan hal penting dalam penjualan properti dan harus benar-benar dipertimbangkan oleh perusahaan. Dalam hal pemasaran, tentunya pihak perusahaan pengembang perumahan harus menentukan strategi pemasaran yang baik untuk menggapai konsumen dan harus mampu mendorong konsumen sehingga tertarik untuk membeli rumah yang ditawarkan. Pemasaran bisnis properti atau perumahan saat ini umumnya masih menggunakan media brosur, seperti pada perumahan Balimbingan Permai, PT. Karya Propertindo Utama. Pemasaran menggunakan brosur memiliki keterbatasan di mana konsumen atau pembeli hanya bisa melihat bentuk rumah berupa 2 dimensi yang mana gambar rumah tidak bisa dilihat secara detail dari berbagai arah. Balimbingan Permai, merupakan perumahan yang dibangun oleh perusahaan PT. Karya Propertindo Utama yang berlokasi di Kabupaten Balimbingan Pematang Siantar, Sumatera Utara. Perumahan ini terdiri dari 80 rumah dari berbagai tipe. Mulai dari tipe 36, 45, 54 dan tipe 70. Untuk harga rumah dari perumahan Balimbingan Permai ini sendiri yaitu berkisar antara 130 sampai dengan 260 juta per unitnya. Berdasarkan data dari PT. Karya Propertindo Utama, jumlah pemilik rumah di perumahan Balimbingan Permai, Pematang Siantar saat ini masih mencapai 53 penghuni. Hal tersebut berbeda dengan capaian jumlah pembelian rumah yang diharapkan oleh perusahaan PT. Karya Propertindo yaitu dengan terjualnya semua rumah. Sampai sekarang masih tersisa 27 rumah atau lahan kosong yang belum dibeli oleh konsumen. Tersisanya rumah yang masih belum terjual berdampak akan tidak didapatkannya keuntungan terhadap rumah yang sudah dibangun oleh perusahaan, serta tidak berkembangannya dan tidak terjadinya pembangunan lanjut perumahan Balimbingan Permai. Berdasarkan data beberapa tahun dari perusahaan PT. Karya Propertindo Utama, terdapat 48% dari 25 orang yang batal membeli rumah memberikan informasi yang jelas terkait kendala atau alasan batal membeli kepada pihak perusahaan PT. Karya Propertindo Utama. Jika

dilakukan perhitungan terhadap potensial omset yang bisa didapatkan, maka perusahaan PT. Karya Propertindo Utama berpotensi untuk mendapatkan keuntungan total sekitar 300 juta atau lebih. Jumlah tersebut akan didapat apabila 48% orang konsumen tersebut tidak membatalkan untuk membeli rumah. Berdasarkan keterangan informasi yang diberikan oleh perusahaan PT. Karya Propertindo, 48% konsumen yang batal beli rumah memiliki alasan/kendala tidak dapat melakukan pengamatan dikarenakan beberapa dari mereka berdomisili di luar kota dan beberapa dari mereka memiliki waktu yang sangat terbatas karena berhalangan dengan pekerjaan di kantor atau tempat kerja. Beberapa dari konsumen yang berdomisili di luar kota juga mengajukan saran untuk diberikan akses pengamatan model 3 dimensi perumahan yang akan dibeli. Dikarenakan menampilkan objek model 3 dimensi yang dibangun oleh perusahaan PT. Karya Propertindo menggunakan perangkat lunak yang memerlukan komputer dengan kebutuhan spesifikasi khusus, maka sulit bagi perusahaan untuk mengarahkan konsumen melakukan pengamatan model 3 dimensi rumah yang ingin dibeli melalui komputer.

Dengan kemajuan teknologi yang semakin maju, media kini beralih ke media virtual reality (VR) (Moura, 2017). Realitas virtual adalah tampilan gambar 3D yang dihasilkan komputer yang dibuat secara realistis menggunakan perangkat tertentu dan membuat seolah-olah pengguna terlibat langsung dengan lingkungan (Puto, 2015). Realitas virtual telah menjadi konsep interaktif yang sangat mudah digunakan dengan perkembangan teknologi seluler yang dapat memainkan peran pendukung (Pius, 2017). Dengan pemanfaatan Teknologi Virtual Reality (VR) diharapkan konsumen akan mengetahui lebih detail produk rumah yang akan dibeli. Karena menurut survei yang telah terjadi, kebanyakan konsumen ketika memesan produk, belum mengetahui bentuk asli dan nyata seperti apa, itu dikarenakan media promosi yang digunakan masih menggunakan katalog gambar dua dimensi yang hanya bisa dilihat dari satu arah saja (Fitrana, 2019). Berdasarkan data survey yang dilakukan pada 56 responden dari penelitian yang berjudul "Pengembangan Aplikasi Katalog Rumah Berbasis Virtual Reality Menggunakan Algoritma FAST", oleh Eis Akmeliny Fitrana, didapatkan bahwa 96,2% responden

mengatakan mereka tertarik apabila ada sebuah aplikasi yang memuat tentang konsep 3 dimensi pada katalog perumahan yang berbasis android. Dan 100% dari konsumen/calon pembeli menyatakan bahwa dengan adanya konsep 3 dimensi pada aplikasi katalog perumahan mempermudah mereka dalam memvisualisasikan sebuah bangunan/rumah yang nantinya akan dipilih. Berdasarkan hasil survei tersebut, aplikasi AR katalog memberikan dampak yang signifikan dalam kegiatan promosi untuk meningkatkan minat para konsumen.

Adapun penelitian sebelumnya yang mengembangkan aplikasi serupa, seperti pada penelitian pertama yang membuat model arsitektur virtual dari objek perumahan menggunakan game engine Unity 3D dengan tujuan untuk menghemat uang dan saat proses jual beli rumah. Mereka menggunakan perangkat seperti Google Cardboard dan Oculus Rift (Deaky dan Parv, 2017). Meskipun demikian, tidak didapatkan informasi terkait software yang mereka gunakan apakah bisa didapat secara gratis atau tidak. Penelitian kedua yaitu pengembangan aplikasi katalog virtual reality penjualan rumah berbasis android yang mana pada aplikasi memiliki fitur untuk mengubah warna cat dinding, pintu, dan jendela pada model rumah 3D (Husniah, L dkk, 2016). Pada penelitian ketiga yang berjudul “A Low-Cost and Lightweight 3D Interactive Real Estate-Purposed Indoor Virtual Reality Application”, oleh Kasim Ozacar, terdapat data survey yang menyimpulkan bahwa setelah melakukan tur secara virtual, responden berkeinginan untuk membeli rumah. Beberapa dari responden juga menyarankan tur dengan penggunaan kontrol arah panah memberikan pengalaman yang lebih baik daripada harus melakukan *teleporting* pada saat melakukan perpindahan lokasi. Berdasarkan ketiga penelitian tersebut penulis ingin mengajukan pembangunan aplikasi yakni penggabungan dari ketiga penelitian dengan beberapa perbaikan/modifikasi, yaitu aplikasi berbasis android yang mampu melakukan tur secara virtual menggunakan kontrol joystick pada *3d walkthrough*-nya. Dalam proses pembangunan, penulis menggunakan teknologi Unity 3D Engine untuk membangun model 3D dan tur virtual yang lingkungannya berupa objek 3 dimensi.

Teknologi Tur Interaktif berbasis Virtual Reality dari Unity 3D ini dapat menjadi solusi terhadap permasalahan konsumen yang berdomisili di luar kota dan memiliki waktu sangat terbatas untuk melakukan pengamatan jarak jauh. Dengan adanya aplikasi ini konsumen dapat melihat objek 3 dimensi dari rumah dan dapat melakukan tur secara virtual terhadap lingkungan rumah. Aplikasi ini juga membantu developer dalam memasarkan rumah yang akan ditawarkan ke konsumen. Oleh karena itu, berdasarkan kemampuan teknologi ar dan 3D tour dari Unity 3D yang mampu menyelesaikan permasalahan di atas, maka penulis mengajukan untuk melakukan pembangunan aplikasi tur interaktif perumahan balimbingan permai PT. Karya Propertindo Utama berbasis virtual reality, dengan harapan dapat mengatasi permasalahan konsumen tersebut dan juga dapat mendukung proses promosi perusahaan kepada konsumen sehingga dapat meningkatkan potensial pembelian rumah di perumahan Balimbingan Permai, Pematang Siantar. Dalam pembangunan aplikasi, peneliti menggunakan metode pengembangan dengan model Waterfall SDLC, dimana metode tersebut terdiri dari beberapa fase yaitu perencanaan sistem, analisis sistem, perancangan sistem, implementasi sistem, dan pemeliharaan sistem. Dalam pengujiannya, peneliti menggunakan 3 metode, yaitu pengujian dengan metode Whitebox Testing (pengujian unit), pengujian dengan Blackbox Testing (pengujian fungsional), dan Pengujian non-fungsional (pengujian compatibility dan usability). Untuk pengujian usability peneliti melakukan pengumpulan data dengan memberikan kuesioner System Usability Scale (SUS) kepada responden calon pembeli / pengunjung perumahan.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang di atas, maka ada beberapa permasalahan yang dapat diidentifikasi, di antaranya sebagai berikut :

- 1) Konsumen terkendala melakukan pengamatan rumah yang akan dibeli dikarenakan konsumen berlokasi di luar kota.

- 2) Konsumen terkendala masalah pengamatan rumah yang akan dibeli dikarenakan waktu yang terbatas.
- 3) Perumahan yang terletak di kabupaten berjarak jauh dari tempat asal konsumen sehingga membutuhkan waktu yang lama bagi konsumen datang ke lokasi perumahan untuk melakukan pengamatan dan pengunjungan kembali rumah yang akan dibeli.

1.3 Rumusan Masalah

Permasalahan yang akan diteliti pada penelitian ini dapat dirumuskan sebagai berikut:

- 1) Bagaimana rancangan Aplikasi Pameran dan Tur Virtual Perumahan Balimbingan Permai?
- 2) Bagaimana hasil pengujian dari Aplikasi Pameran dan Tur Virtual Perumahan Balimbingan Permai?
- 3) Bagaimana pengalaman pengguna akan pengamatan perumahan sesudah adanya aplikasi?

1.4 Tujuan Penelitian

- 1) Tujuan Umum

Dengan dibangunnya aplikasi berbasis Virtual Reality ini, maka diharapkan mampu mengatasi permasalahan pada konsumen yang memiliki keterbatasan untuk melakukan survey jarak jauh. Dengan adanya aplikasi ini diharapkan dapat menjadi solusi untuk permasalahan tersebut dan juga meningkatkan minat beli konsumen sehingga mampu menambah jumlah properti/rumah yang terjual dari perumahan Balimbingan Permai PT. Karya Propertindo Utama.

2) Tujuan Khusus

- a. Dibangunnya Aplikasi berbasis Virtual Reality yang memiliki performa cepat dalam menampilkan model 3 dimensi perumahan dan melakukan tur secara virtual.
- b. Mampu membantu konsumen untuk melakukan pengamatan jarak jauh dan membayangkan bentuk asli dari properti/rumah yang ingin dibeli.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi beberapa pihak, antara lain sebagai berikut:

1) Konsumen

- a. Mampu membantu konsumen untuk melakukan pengamatan jarak jauh melalui objek maya yang berbentuk 3 dimensi dan melakukan pengamatan lingkungan sekitar rumah melalui 3D tour.
- b. Membantu konsumen untuk lebih mudah membandingkan tipe-tipe perumahan yang akan dibeli.

2) Peneliti

- a. Mengetahui teknik membangun aplikasi secara umum.
- b. Sebagai bahan referensi untuk penelitian dan pengembangan aplikasi Android selanjutnya.

1.6 Batasan Penelitian/Ruang Lingkup Penelitian

Permasalahan dalam penelitian ini cukup luas, sehingga perlu dilakukan pembatasan masalah. Permasalahan yang dibahas antara lain sebagai berikut:

- 1) Pengembangan Aplikasi berbasis Virtual Reality dengan Unity 3D yang mampu menampilkan objek properti/rumah secara virtual berupa model 3 dimensi dan digunakan untuk melakukan tur lingkungan properti/rumah secara sehingga dapat melihat secara detail bagian dalam dan luar rumah.
- 2) Aplikasi digunakan hanya pada perumahan Balimbingan Permai PT. Karya Propertindo Utama.
- 3) Aplikasi diterapkan pada bidang properti/perumahan saja.

1.7 Sistematika Pembahasan/Laporan

1.1.1. BAB I Pendahuluan

Bab Pendahuluan menjelaskan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika pembahasan.

1.1.2. BAB II Landasan Kepustakaan

Bab landasan kepastakaan menjelaskan tentang kajian pustaka terkait penelitian yang telah ada seperti penelitian tentang penggunaan virtual reality di berbagai bidang seperti edukasi, pariwisata, maupun di bidang perumahan. Dengan macam-macam metode dan penggunaan metode. Penjelasan teori berisi teori-teori yang mendukung dalam pengembangan dan perancangan.

1.1.3. BAB III Metodologi Penelitian

Bab ini membahas tentang tahap-tahap penelitian baik itu dalam pengumpulan data yang akan digunakan dan bagaimana cara mendapatkan datanya, analisis teoritis terhadap suatu metode dan juga proses penelitian.

1.1.4. BAB IV Perancangan Sistem

Bab ini membahas perancangan sistem menggunakan metode yang telah dipilih.

1.1.5. BAB V Implementasi Sistem

Bab ini menjelaskan setelah merancang sistem maka dilakukanlah implementasi terhadap metode yang telah dipilih.

1.1.6. BAB VI Pengujian dan Analisis

Bab ini menjelaskan setelah di implementasikan maka terjadi pengujian dari tahap pengguna memulai aplikasi untuk memilih menu virtual reality atau 3D tour. Jika memilih virtual reality, aplikasi akan mulai mengenali pola yang ada pada brosur dan kemudian menghasilkan output berupa objek virtual. Kemudian jika memilih 3D Tour maka aplikasi akan menampilkan tampilan di mana pengguna bisa melakukan tur berlingkungan 3 dimensi secara virtual. Kemudian untuk hasil pengujiannya yaitu didapat dari apakah dalam menampilkan hasil, aplikasi sudah memenuhi kriteria, sesuai dengan apa yang diinginkan pengguna.

1.1.7. BAB VII Penutup

Bab ini menjelaskan hasil dari Kesimpulan yang di dapatkan dari penelitian ini dan memberikan saran agar sistem dapat lebih baik lagi untuk memenuhi kriteria yang telah ditentukan.

BAB 2

LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari pustaka ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Dalam landasan kepustakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Jika dibutuhkan sesuai dengan karakteristik penelitiannya dan syarat kecukupan khusus keminatan tertentu, bisa juga terdapat kajian pustaka yang menjelaskan secara umum penelitian-penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan dan perbedaan skripsi tersebut terhadap penelitian terdahulu yang dituliskan.

2.1 KAJIAN PUSTAKA

Berikut adalah beberapa penelitian terkait yang berhubungan dengan topik penelitian, diantaranya adalah penelitian yang berjudul “Aplikasi Virtual Reality untuk Pengenalan Pola Satwa Menggunakan Vuforia” oleh Uning dan Annafi, “Implementasi Virtual Reality untuk Android Sebagai Media Promosi Menggunakan Unity dan Vuforia” oleh Affix Endy Abidita, “Aplikasi Virtual tour Berbasis Multimedia Interaktif Menggunakan AutoDesk 3Ds Max” oleh Akip Suhendar, dan “Interaktif Virtual Reality untuk Katalog Penjualan Rumah Berbasis Android” oleh Husniah.

Penelitian yang berjudul “Aplikasi Virtual Reality untuk Pengenalan Pola Satwa Menggunakan Vuforia” adalah aplikasi Virtual Reality pada tempat wisata kebun binatang Gembira Loka Yogyakarta yang dibuat oleh Uning Lestari dan Annafi pada tahun 2015. Aplikasi ini dikembangkan untuk membantu pengunjung mendapatkan informasi yang lebih detail mengenai satwa-satwa yang ada pada kebun binatang Gembira Loka Yogyakarta. Adapun cara penggunaannya yaitu dengan mengarahkan kamera handphone ke gambar marker dan aplikasi melakukan proses pemindaan untuk

pencocokan pola yang kemudian akan ditampilkan gambar 2 dimensi dari hewan dan informasi detail tentang hewan tersebut.

Penelitian yang berjudul “Implementasi Virtual Reality untuk Android Sebagai Media Promosi Menggunakan Unity” yang dilakukan oleh Affix Endy Abidita pada tahun 2015. Aplikasi ini memiliki fitur yang menggunakan teknologi AR untuk menampilkan informasi kampus bangunan STMIK AKAKOM berbentuk 3 dimensi, serta memiliki tambahan elemen multimedia berupa audio. Aplikasi ini ditujukan untuk calon mahasiswa yang ingin mengetahui informasi STMIK AKAKOM meliputi S1 Teknik Informatika, S1 Sistem Informasi, D3 Komputerisasi Akuntansi, D3 Manajemen Informatika, D3 Teknik Komputer, Syarat Daftar dan Fasilitas.

Penelitian yang berjudul “Aplikasi Virtual tour Berbasis Multimedia Interaktif Menggunakan AutoDesk 3Ds Max” oleh Akip Suhendar pada tahun 2016, merupakan aplikasi yang dapat melakukan tur secara virtual yang berjenis virtual model 3 dimensi pada Gedung 1 Universitas Serang Raya. Yang mana dalam pengembangannya, peneliti menggunakan tool Game Engine Unity 3D dan 3D Studio Max. Aplikasi tour ini dikembangkan menggunakan first person controller yang mana membuat pengguna mampu untuk melakukan tur berjalan di dalam ruangan.

Penelitian yang berjudul “Interaktif Virtual Reality untuk Katalog Penjualan Rumah Berbasis Android” oleh Husniah pada tahun 2016. Aplikasi berbasis virtual reality dapat digunakan untuk pemasaran dalam penjualan rumah berbasis android yang mana aplikasi ini merupakan pengembangan dari penelitian serupa sebelumnya, dengan penambahan pada fitur yaitu aplikasi mampu mengubah warna cat dinding, pintu, dan jendela pada model rumah 3D.

2.2 Virtual Reality

Virtual Reality (VR) atau realitas maya adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan oleh computer (computer-simulated environment), suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi (Sihite, 2013). Konsep VR mengacu pada system prinsip-prinsip, metode dan teknik yang digunakan untuk merancang dan menciptakan produk-produk perangkat lunak untuk digunakan oleh bantuan dari beberapa sistem computer multimedia dengan sistem perangkat khusus (Lacrama, 2007).

Lingkungan realitas maya terkini umumnya menyajikan pengalaman visual, yang ditampilkan pada sebuah layar komputer atau melalui sebuah penampil stereoskopik, tapi beberapa simulasi mengikut sertakan tambahan informasi hasil pengindraan, seperti suara melalui speaker atau headphone. Beberapa sistem haptic canggih sekarang meliputi informasi sentuh, biasanya dikenal sebagai umpan balik kekuatan pada aplikasi berjudi dan medis (Sihite, 2013).

2.3 Unity 3D

Unity 3D merupakan sebuah tools yang terintegrasi untuk membuat bentuk obyek 3D pada video games atau untuk konteks interaktif lain seperti Visualisasi Arsitektur atau animasi 3D real-time. Lingkungan dari pengembangan Unity 3D berjalan pada Microsoft Windows dan Mac Os X, serta aplikasi yang dibuat oleh Unity 3D dapat berjalan pada Windows, Mac, Xbox 360, Playstation 3, Wii, iPad, iPhone dan tidak ketinggalan pada platform Android. Unity juga dapat membuat game berbasis browser yang menggunakan Unity web player plugin, yang dapat bekerja pada Mac dan Windows, tapi tidak pada Linux. (Mutia dan Djuniadi ,2015).

2.4 Android

Android merupakan sebuah sistem operasi yang digunakan pada telepon seluler yang dikembangkan berbasis linux (Nazaruddin, 2012). Android pertama kali dibuat oleh Android, Inc. dan pada tahun 2005 Google pun resmi membeli Android. Android menjadi salah satu sistem operasi yang sangat populer saat ini dan sudah banyak perangkat seluler yang menggunakan sistem operasi Android. Dalam membuat aplikasi yang berjalan pada sistem operasi Android, pengembang aplikasi saat ini dapat menuliskan kode dengan menggunakan bahasa Kotlin, Java, dan C++ (Developers, 2019).

2.5 Pengujian Fungsional

Pengujian fungsional adalah pengujian yang bertujuan untuk memberikan hasil evaluasi berupa kepastian bahwa aplikasi atau sistem yang dikembangkan telah memenuhi seluruh kebutuhan aplikasi atau sistem. Pengujian ini bukan berfokus pada keberhasilan proses sistem, melainkan berfokus pada hasil dari proses sistem. Salah satu metode dalam pengujian ini yaitu seperti black box.

2.6 Pengujian *Usability*

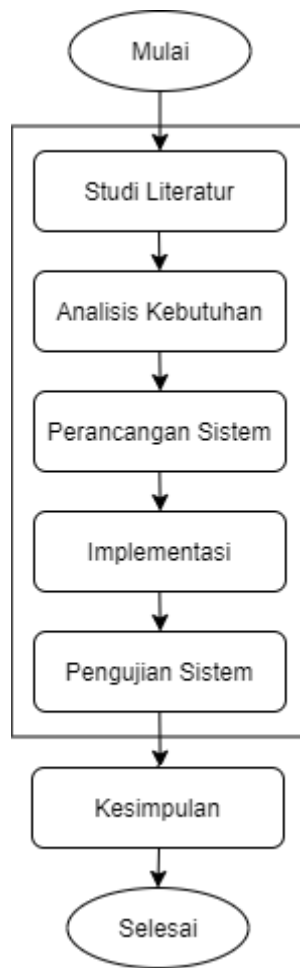
Usability merupakan suatu arti yang berasal dari kata usable, artinya dapat digunakan dengan baik. Suatu yang dihasilkan dapat dikatakan berguna atau baik digunakan apabila kemungkinan terjadinya kegagalan saat penggunaan kecil atau minimal sehingga harus mampu memberi kepuasan dan manfaat kepada pengguna (Jeff Rubin, 2008). Sedangkan menurut ISO 9241:11 (1998), usability yaitu “sejauh mana suatu produk dapat digunakan oleh pengguna tertentu untuk mencapai target yang ditetapkan dengan efektifitas, efisiensi, dan mencapai kepuasan penggunaan dalam konteks tertentu”.

BAB 3

METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan mengenai metode pelaksanaan penelitian yang digunakan dalam merancang dan mengembangkan aplikasi tur interaktif pada perumahan Balimbingan Permai berbasis *Virtual Reality*. Pada bab ini akan dijelaskan terkait tahapan-tahapan yang dilakukan dalam pengimplementasian metode pengembangan aplikasi. Dalam penelitian ini, peneliti menggunakan metode dengan model pengembangan sistem yang mengadaptasi model Waterfall SDLC.

Pola pengembangan waterfall merupakan pola pengembangan perangkat lunak yang bersifat searah, di mana proses pengembangan hanya akan berjalan maju dan tidak dapat kembali ke tahapan sebelumnya. Sehingga pola pengembangan ini cocok diterapkan apabila pengembang telah mampu memahami permasalahan serta kebutuhan dari aplikasi yang akan dikembangkan serta dapat dipastikan bahwa kebutuhan-kebutuhan dari aplikasi yang akan dikembangkan serta dapat dipastikan bahwa kebutuhan-kebutuhan yang didapatkan dari proses elisitasi tidak akan mengalami perubahan secara signifikan baik dalam masa pengembangan perangkat lunak maupun di masa yang akan datang (Sommerville, 2011). Pola pengembangan waterfall diterapkan karena aplikasi yang hendak dikembangkan dalam penelitian ini memiliki daftar kebutuhan yang konsisten atau jarang berubah. Metode pengembangan waterfall memiliki 7 fase yang terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, dan kesimpulan seperti pada diagram di bawah ini.



Gambar 3.1 Metode Pengembangan

3.1 Studi Literatur

Studi literatur menjelaskan dasar-dasar teori yang disusun berdasarkan referensi yang diperoleh dari artikel, buku, jurnal serta penelitian-penelitian terkait yang sejenis. Studi literatur digunakan untuk penunjang dan pendukung dalam penulisan skripsi, serta sebagai pengetahuan tambahan dalam melakukan analisis kebutuhan, perancangan, implementasi serta pengujian terhadap suatu perangkat lunak sebagai penelitian. Selain itu, studi literatur dilakukan untuk mencegah plagiasi dari karya/penelitian yang telah ada sebelumnya.

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem berguna untuk mendefinisikan kebutuhan yang diperlukan dalam pengembangan perangkat lunak. Metode yang digunakan dalam analisis kebutuhan perangkat lunak pada penelitian ini adalah Object Oriented Analysis (OOA). Setelah melakukan analisis kebutuhan, kemudian kebutuhan dimodelkan dengan menggunakan bahasa pemodelan UML (Unified Modeling Language) yang meliputi Use Case Diagram dan Use Case Scenario. Use Case Diagram digunakan untuk mendeskripsikan kebutuhan fungsional sistem/perangkat lunak dari sisi pengguna. Dan Use Case Scenario digunakan untuk menjelaskan lebih detail dari tiap-tiap case pada use case diagram.

3.3 Perancangan Sistem

Perancangan sistem digunakan untuk merancang tiap-tiap kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Tahap perancangan digunakan sebagai bahan perancangan/pembentukan langkah kerja yang meliputi seluruh isi sistem yang akan dikembangkan. Pada tahap ini terdapat perancangan arsitektur, perancangan komponen, perancangan data dan perancangan antarmuka. Tiap-tiap kebutuhan yang telah dijabarkan secara spesifik, dirancang pada tahap perancangan dengan menggunakan diagram Sequence. Sequence diagram menjelaskan alur dari sistem berdasarkan fungsi yang digambarkan dengan garis waktu. Selain Sequence diagram, adapun class diagram yang berguna menggambarkan hubungan antara tiap-tiap kelas yang saling terkait satu sama lain pada sebuah perangkat lunak yang dikembangkan

3.4 Implementasi Sistem

Implementasi sistem merupakan tahapan yang dilakukan ketika semua kebutuhan yang telah dijelaskan selesai dirancang. Implementasi yang dilakukan meliputi implementasi basis data menggunakan DBMS MySQL, implementasi logika program dan implementasi antarmuka. Implementasi basis data merupakan perancangan basis data yang dilakukan dengan mendefinisikan tabel yang digunakan dengan DDL (Data

Definition Language). Implementasi logika program menggunakan framework Android SDK dengan bahasa pemrograman Java.

3.5 Pengujian Sistem

Pengujian sistem dilakukan untuk menguji sistem yang dikembangkan, apakah semua kebutuhan yang telah dijabarkan telah berhasil diimplementasikan pada sistem atau tidak. Pengujian sistem ini juga digunakan untuk mencari kesalahan/kesalahan yang terdapat pada perangkat lunak yang sedang dikembangkan. Pada pengujian sistem, menggunakan pengujian whitebox untuk menguji struktur perangkat lunak bagian dalam (source code) dan menggunakan pengujian blackbox untuk menguji struktur perangkat lunak bagian luar (interface).

3.6 Kesimpulan dan Saran

Kesimpulan didapat setelah semua tahapan metodologi mulai dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian sistem, dan analisis hasil berhasil dilaksanakan dengan teratur. Kesimpulan yang valid diperoleh ketika perangkat lunak yang dikembangkan telah lolos tahap pengujian. Selain kesimpulan, adapun saran yang berguna untuk memperbaiki apa-apa saja yang kurang dalam pengembangan aplikasi tur interaktif berbasis virtual reality. Saran berguna sebagai pengembangan lebih lanjut untuk sistem kedepannya.

BAB 4

REKAYASA KEBUTUHAN

Bagian ini menjelaskan mengenai fase analisis kebutuhan dari pembangunan aplikasi tur interaktif Perumahan Balimbingan Permai PT. Karya Propertindo Utama berbasis *virtual reality*. Pada bab ini akan dijelaskan gambaran umum sistem yang akan dibangun, aktor yang akan menggunakan aplikasi, storyboard, kebutuhan dalam membangun sistem, use case diagram dan use case scenario.

4.1. Gambaran Umum Aplikasi

Gambaran umum aplikasi menjelaskan tentang aplikasi Pameran dan Tur Virtual Perumahan Balimbingan Permai dan cara penggunaannya.

4.1.1. Deskripsi

Aplikasi yang dikembangkan pada penelitian ini adalah aplikasi untuk pengguna *home seeker* yang mana pengguna dapat menggunakan aplikasi untuk mengamati bentuk 3 dimensi rumah dan melakukan *3d tour* secara virtual. Aplikasi ini dapat menampilkan objek 3 dimensi secara virtual melalui *virtual reality* yang dikembangkan menggunakan *Unity 3D Engine*.

4.1.2. Cara Penggunaan

Untuk menggunakan aplikasi Pameran dan Tur Virtual ini, dibutuhkan aplikasi yang terinstal pada *smartphone* Android. Kemudian pengguna melakukan pencarian rumah yang diinginkan dan memilih salah satu dari hasil pencarian. Setelahnya akan terdapat opsi untuk pengguna dapat menampilkan tur interaktif secara virtual.

4.2. Analisis Kebutuhan Sistem

Analisis kebutuhan dilakukan dengan cara studi pustaka terkait sistem yang sudah ada. Sistem yang dijadikan acuan ialah aplikasi Google Maps Popular Times dan sistem yang dikembangkan oleh Lesani dan Miranda-Moreno (2018). Analisis kebutuhan sistem merupakan analisis kebutuhan dasar sebuah sistem yang harus tersedia maupun kebutuhan pendukung sistem. Analisis kebutuhan sistem dapat dibagi menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Pada penelitian ini, kode fungsi digunakan untuk merepresentasikan kebutuhan fungsional dan non-fungsional tertentu dengan kode SRS-TIVR-F-N untuk kode kebutuhan fungsional dan SRS-TIVR-NF-

N untuk kebutuhan non-fungsional, dimana N adalah angka dari 1 sampai jumlah kebutuhan masing-masing.

4.2.1 Identifikasi Aktor

Aktor adalah seseorang atau sistem yang dapat melakukan interaksi dengan sistem. Pada penelitian ini, aktor yang ada deskripsikan pada Tabel 4.1.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Pengguna	Aktor yang dapat melakukan pengamatan bentuk 3 dimensi dari rumah dan melakukan tur secara interaktif. Pengguna berupa pengguna umum yang dapat menggunakan aplikasi secara langsung.

4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang terkait dengan fungsi produk (Sukamto & Shalahuddin, 2016). Kebutuhan fungsional pada pengembangan lanjut ini dapat dilihat pada Tabel 4.3

Tabel 4.2 Penjelasan Format Kode Kebutuhan

Kode Format	Deskripsi
SRS	SRS merupakan inisial dari <i>Software Requirement Specification</i> .
TIVR	TIVR merupakan inisial untuk aplikasi Tur Interaktif berbasis Virtual Reality
F/NF	F merupakan kode penunjuk untuk kebutuhan fungsional sedangkan NF untuk kebutuhan non-fungsional
XXX	XXX menyatakan nomor kebutuhan

Tabel 4.3 Kebutuhan Fungsional Sistem

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SRS-TIVR-F-1	Menampilkan daftar rumah	Sistem dapat menampilkan daftar rumah.
2	SRS-TIVR-F-2	Menampilkan Tur Interaktif	Sistem dapat menampilkan Tur Interaktif berbasis Virtual Reality dengan menampilkan objek 3 dimensi rumah.

4.2.3 Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak langsung berhubungan dengan sistem namun menjadi penunjang bagi kebutuhan fungsional (Summerville, 2011). Kebutuhan non-fungsional dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kebutuhan Non-Fungsional Sistem

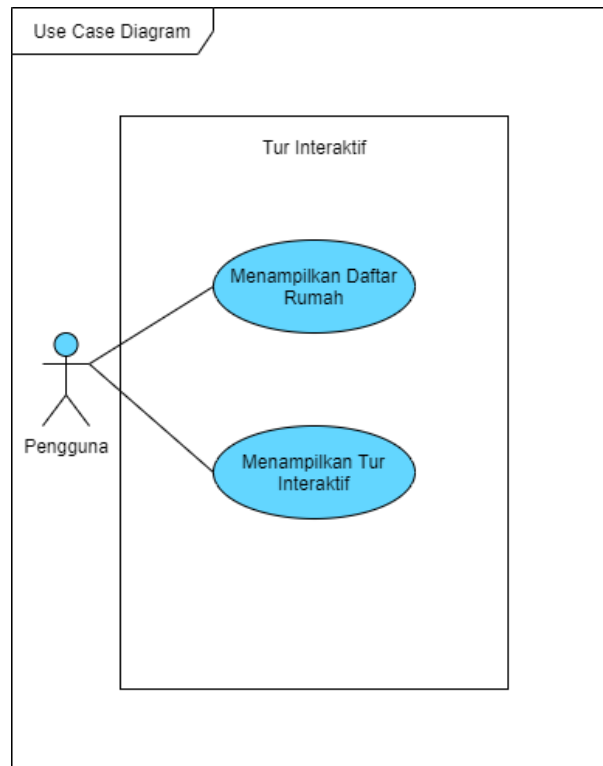
No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SRS-TIVR-NF-1	<i>Usability</i>	<i>Usability</i> menjelaskan tentang kepuasan dan kemudahan pengguna terhadap pemakaian aplikasi. Usabilitas diangkat karena tujuan dari dikembangkanya aplikasi adalah untuk meningkatkan daya beli <i>home seeker</i> , maka diperlukan kebutuhan akan usabilitas yang menjamin kepuasan dan kemudahan pengguna dalam menggunakan aplikasi.
2	SRS-TIVR-NF-2	<i>Compatibility</i>	<i>Compatibility</i> menjelaskan tentang banyaknya jenis perangkat atau versi teknologi yang bisa digunakan untuk menjalankan aplikasi.
3	SRS-TIVR-NF-3	<i>Performance</i>	Performance menjelaskan tentang batasan mengenai kecepatan operasional, respon, <i>throughput</i> , dan <i>timing</i> . Dalam pembangunan aplikasi ini, penulis lebih memfokuskan analisa pengujian pada bagian kecepatan operasional berupa <i>frame rate per second (fps)</i> saja.

4.3. Pemodelan Kebutuhan

Use case mendefinisikan tentang bagaimana aktor berinteraksi dengan sistem pada kondisi tertentu. Use case menggambarkan perangkat lunak dari sudut pandang aktor (Pressman, 2010).

4.3.1. Use Case Diagram

. Use case diagram adalah diagram yang dapat menunjukkan actor dan hal yang dapat dilakukan oleh actor kepada sistem untuk melakukan sesuatu pada sistem. Actor dapat berupa pengguna maupun sistem eksternal yang dapat berinteraksi kepada sistem (Sukamto & Shalahuddin, 2016).



Gambar 4.1 Use Case Diagram Pengguna

4.3.2. Use case Scenario

Use Case Scenario adalah penjelasan yang detail terhadap setiap use case yang sudah dipetakan pada *use case diagram*. *Use Case Scenario* digambarkan dalam bentuk table yang terdiri dari nama use case, kode kebutuhan, actor, tujuan, pre condition, main flow, post condition, dan alternative flow. Bagian pre condition mendeskripsikan tentang kondisi sistem yang harus dipenuhi sebelum masuk ke bagian main flow.

Tabel 4.5 Use Case Scenario Menampilkan Daftar Rumah

Nama <i>Use Case</i>	Menampilkan Daftar Rumah
Kode Kebutuhan Terkait	SRS-TIVR-F-1
Aktor	Pengguna
Tujuan	Menampilkan daftar rumah berdasarkan filter yang diberikan oleh pengguna
Pre Condition	Aktor telah berada pada halaman utama
Main Flow	<ol style="list-style-type: none">1. Pengguna membuka halaman utama2. Sistem menampilkan daftar rumah berdasarkan filter yang diberikan oleh pengguna
Post Condition	Sistem menampilkan daftar rumah
Alternative Flow	-

Tabel 4.6 Use Case Scenario Menampilkan 3D Tour

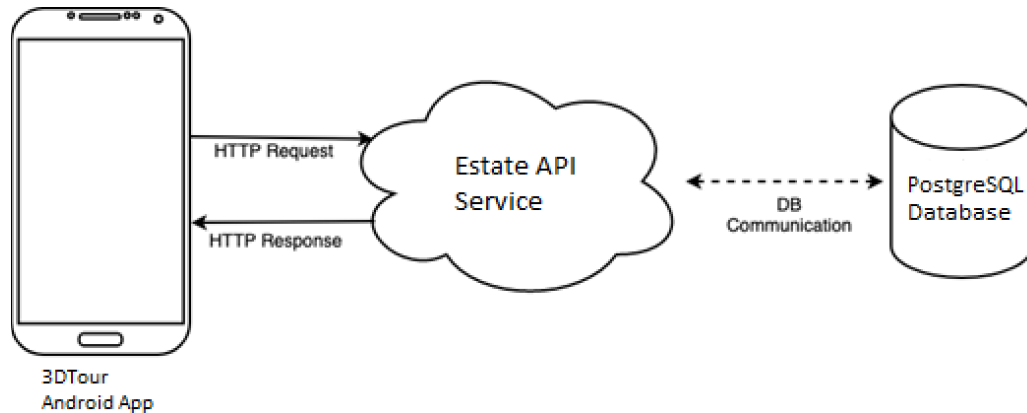
Nama <i>Use Case</i>	Menampilkan 3D Tour
Kode Kebutuhan Terkait	SRS-TIVR-F-2
Aktor	Pengguna
Tujuan	Menampilkan Tur Interaktif berbasis Virtual Reality dengan menampilkan objek 3 dimensi rumah.
Pre Condition	Aktor telah berada pada halaman utama
Main Flow	<ol style="list-style-type: none">1. Pengguna berada di halaman utama2. Pengguna melakukan tap pada salah satu item rumah3. Sistem menjalankan Tur Interaktif dari objek 3 dimensi rumah.
Post Condition	Sistem menampilkan Tur Interaktif <i>Unity</i>
Alternative Flow	-

BAB 5

PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

5.1.1 Perancangan Arsitektur Sistem



Gambar 5.1 Rancangan Arsitektur Sistem

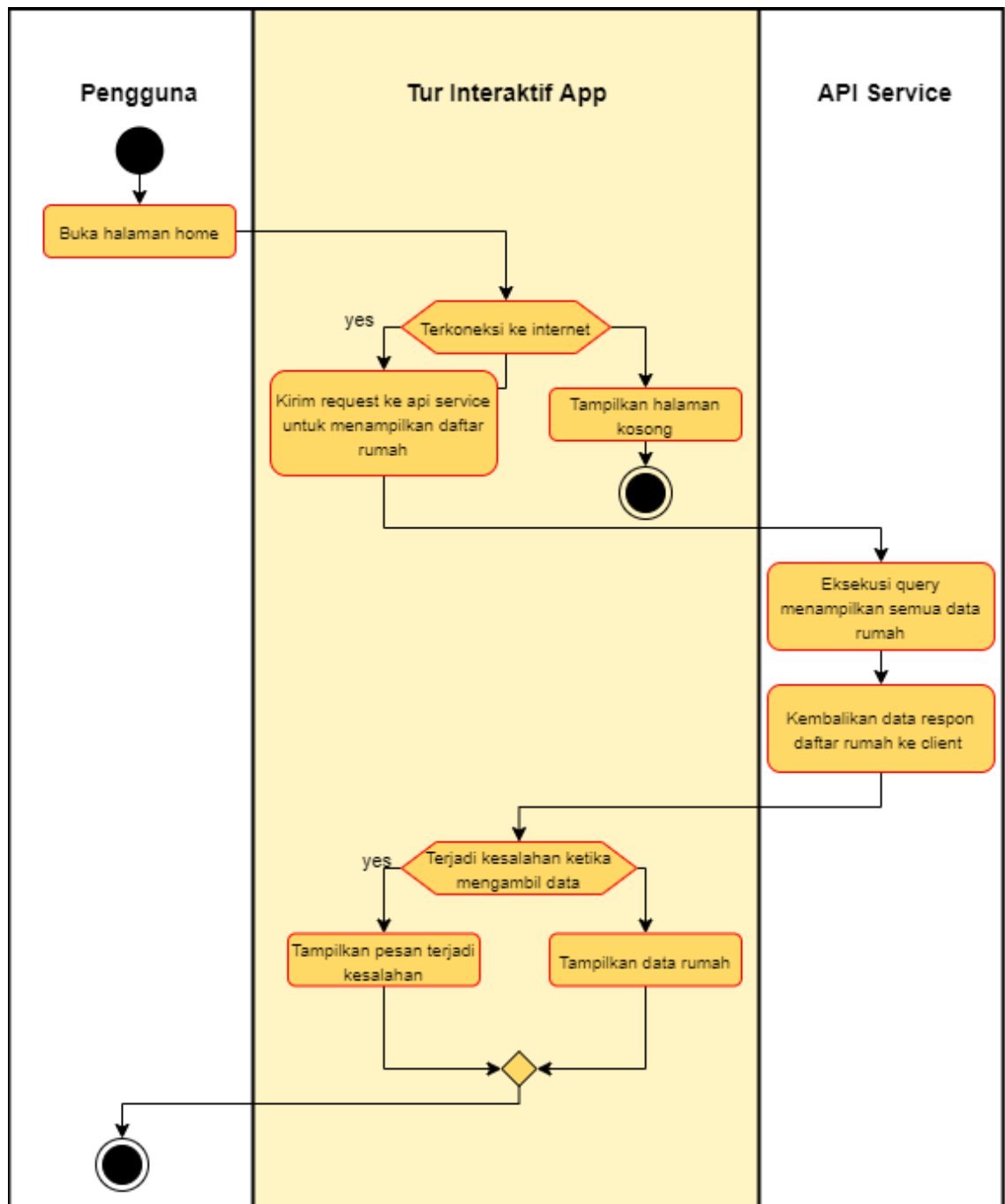
Rancangan arsitektur sistem yang digunakan pada penelitian ini digambarkan pada **Error! Reference source not found..** Komunikasi yang digunakan untuk menghubungkan aplikasi 3DTour dengan web service adalah HTTP. Adapun proses pertukaran data menggunakan format JSON. Web service berfungsi untuk menyimpan data pada database serta mengagregasi data yang telah dikirim oleh aplikasi 3DTour.

Aplikasi 3DTour dirancang dengan menggunakan pola arsitektur MVP. Berdasarkan pola arsitektur tersebut, setiap halaman memiliki dua buah class utama, yaitu View dan Presenter, yang saling mereferensi satu sama lain melalui interface. Contohnya adalah halaman main yang memiliki interface `MainViewContract` yang diimplementasikan oleh class `MainActivity` dan `MainPresenterContract` yang diimplementasikan oleh class `MainPresenter`. Adapun masing-masing class (`MainActivity` dan `MainPresenter`) saling mereferensi satu sama lain dengan bersandar pada interface yang diimplementasi. Sebagai contoh, `MainActivity` yang memiliki referensi kepada `MainPresenter` dengan memiliki sebuah instance variable bertipe `HomePresenterContract` dan sebaliknya.

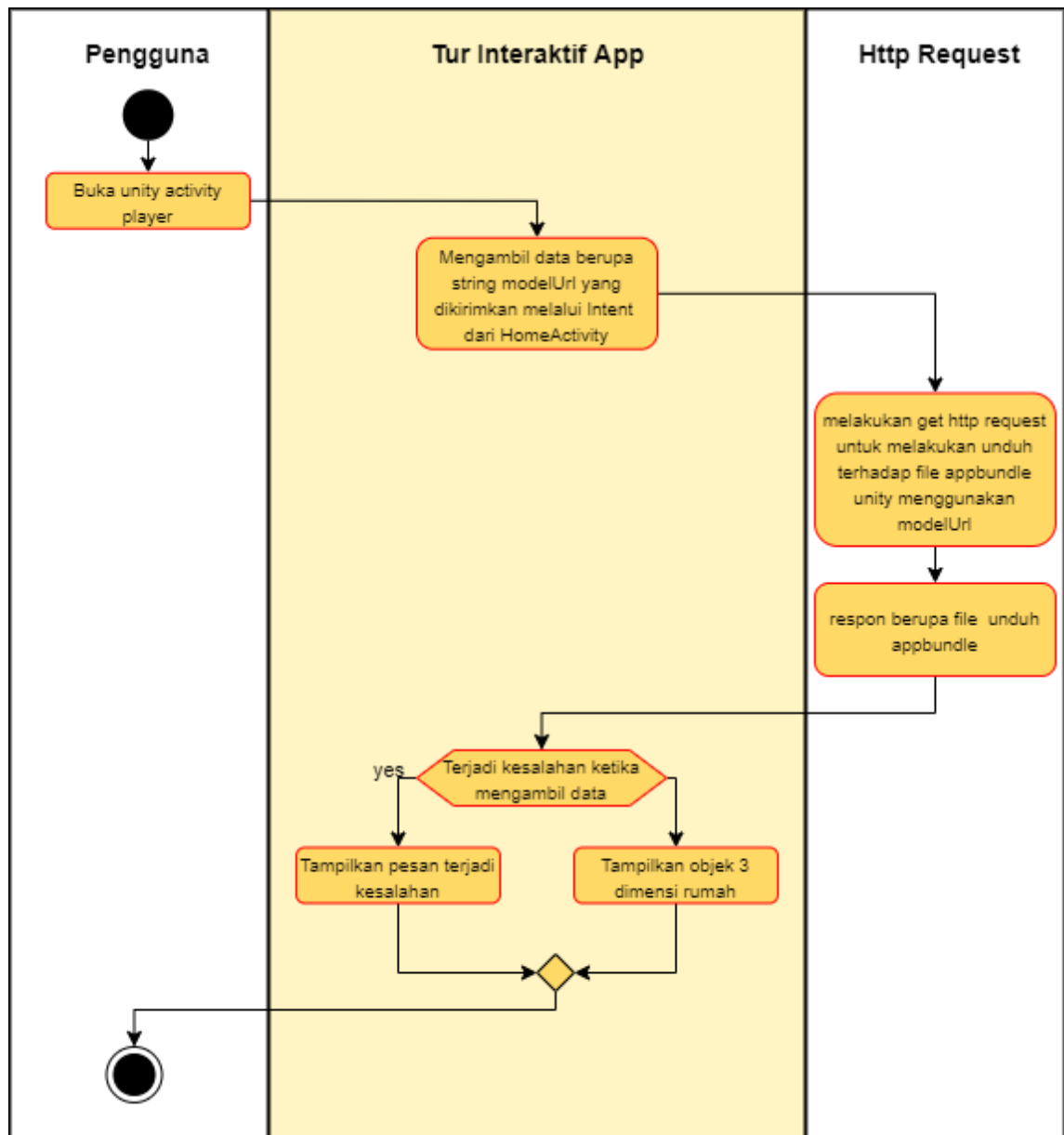
Selanjutnya, sistem Estate API Service dirancang tanpa menggunakan arsitektur khusus tetapi tetap terdapat pemisahan konsentrasi antara controller, model, HTTP route handler, dan konektivitas ke database. Controller berfungsi untuk menjalankan pekerjaan spesifik berdasarkan request yang dikirim dari user. Model merepresentasikan entitas yang terlibat dalam sistem. HTTP route handler berperan sebagai pemroses rute HTTP request yang datang dan menghubungkannya ke controller yang sesuai. Selanjutnya, objek database yang mengabstraksikan operasi-operasi terhadap database.

5.1.2 Activity Diagram

Activity Diagram berfungsi untuk menggambarkan alur kerja yang disusun berdasarkan urutan aktivitas atau aksi yang dikerjakan oleh aktor atau sistem tertentu. Diagram tersebut juga mendukung percabangan, perulangan, dan paralelisme. Adapun hasil perancangan menghasilkan dua buah activity diagram yang dituangkan pada Gambar 5.2 dan Gambar 5.3. Dua buah alur yang tergambarkan adalah alur untuk melihat daftar rumah dan alur untuk menampilkan tur interaktif rumah.



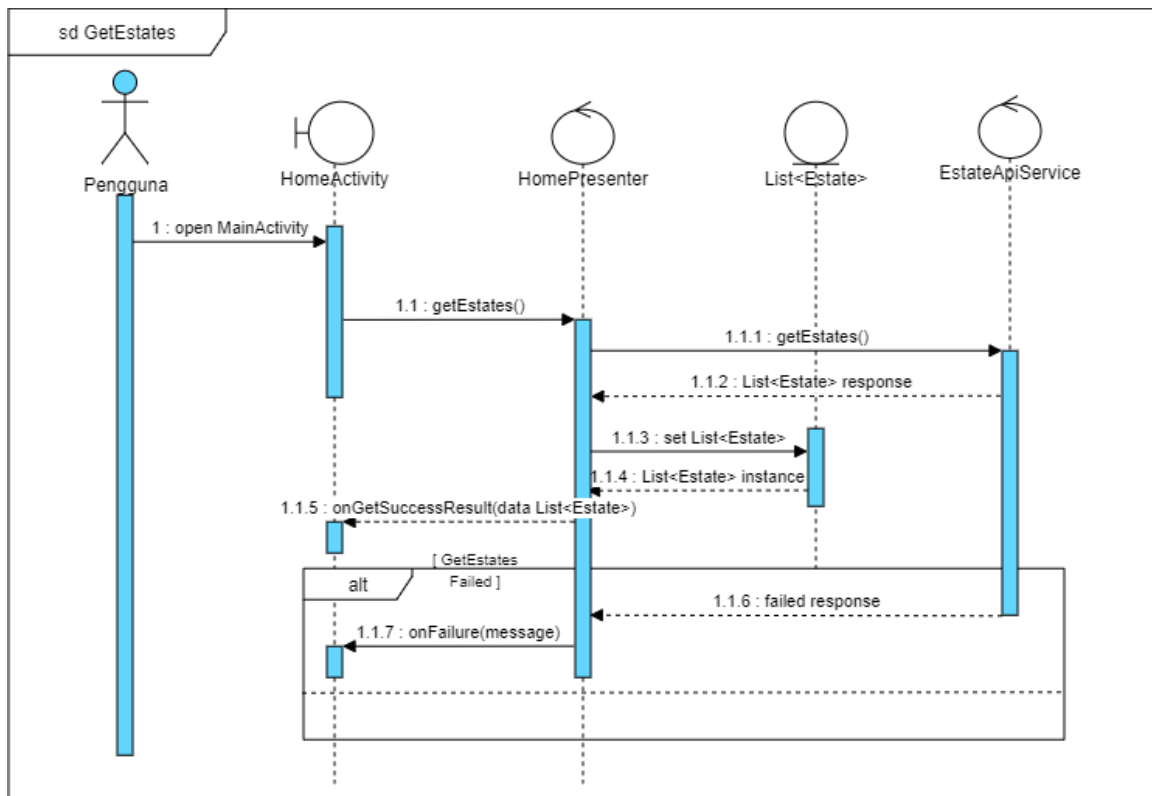
Gambar 5.2 Rancangan *Activity Diagram* Menampilkan Daftar Rumah



Gambar 5.3 Rancangan *Activity Diagram* Menampilkan Tur Interaktif

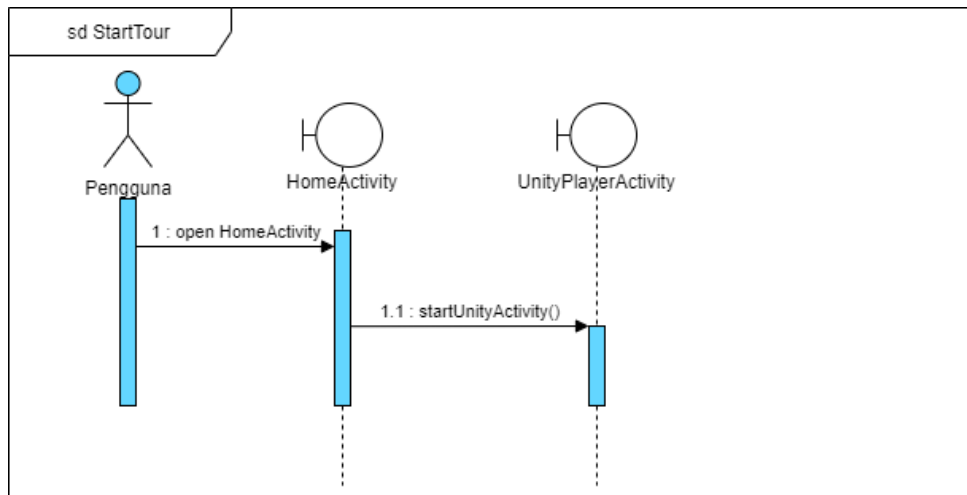
5.1.3 Sequence Diagram

Sequence Diagram dibuat untuk menjelaskan interaksi kelas-kelas yang terlibat dalam satu alur kerja. Terdapat 2 buah sequence diagram yang dibuat untuk menjelaskan alur pada aplikasi Android dan 1 buah sequence diagram untuk menjelaskan alur pada api service.



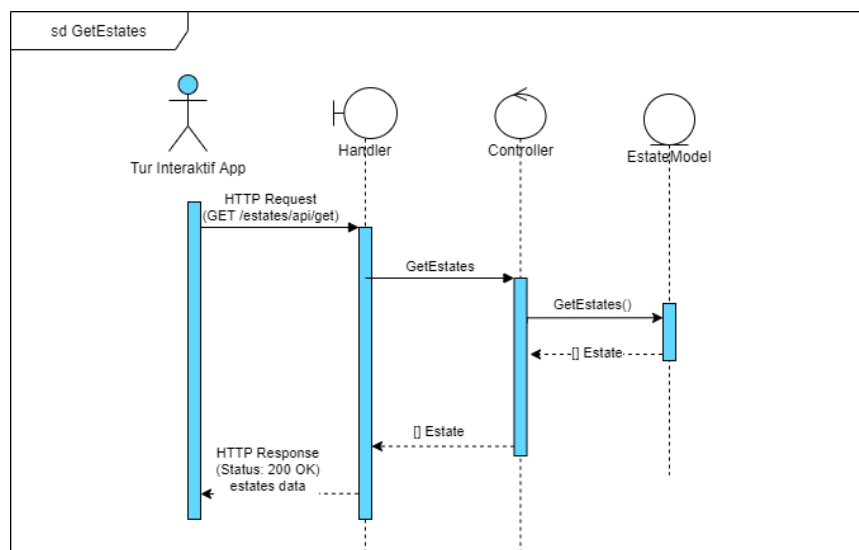
Gambar 5.4 Sequence Diagram Menampilkan Daftar Rumah

Gambar 5.4 menggambarkan alur untuk melihat daftar rumah. Saat pengguna membuka aplikasi, akan dipanggil fungsi `getEstates()` pada `HomePresenter` diikuti oleh pemanggilan fungsi `getEstates()` pada `ApiService`. Selanjutnya `API Service` mengembalikan data respon berupa daftar rumah berbentuk json. Yang mana bentuk json diubah secara otomatis menjadi objek dengan tipe `List<Estate>`. Kemudian setelah berhasil dari `HomePresenter` memanggil callback `onGetSuccessResult()` dengan mengembalikan objek data bertipe `List<Estate>` yang diinisialisasi di kelas `HomeActivity`. Pada method `onGetSuccessResult()` ini dilakukan populasi data ke dalam `recyclerview`.



Gambar 5.5 Sequence Diagram Menampilkan Tur Interaktif

Gambar 5.5 menggambarkan alur menampilkan tur interaktif. Pengguna berada HomeActivity dengan pre-condition data daftar rumah telah ditampilkan pada halaman HomeActivity. Kemudian user melakukan tap ke salah satu item data daftar rumah. Kemudian method startUnityActivity dipanggil dari HomeActivity dengan menyisipkan data berupa modelUrl dengan objek Intent saat memanggil method startActivity(). Kemudian sistem menampilkan unity activity yang dikembangkan menggunakan unity 3D Engine dan di-embed di projek Android Studio.



Gambar 5.6 Sequence Diagram Menampilkan Daftar Rumah (API Service)

Gambar 5.6 menggambarkan alur pengambilan data semua estate. TurInteraktifApp yang merupakan aplikasi Android Tur Interaktif memanggil endpoint

“/estates/api/get”. Selanjutnya, kelas Handler yang menangani *request* yang masuk akan memanggil fungsi *GetEstates* dari kelas Controller. Fungsi tersebut akan memanggil fungsi *GetEstates()* dari kelas *EstateModel*. Fungsi *GetEstates()* dari kelas Model ini akan mengembalikan data berupa list Estate.

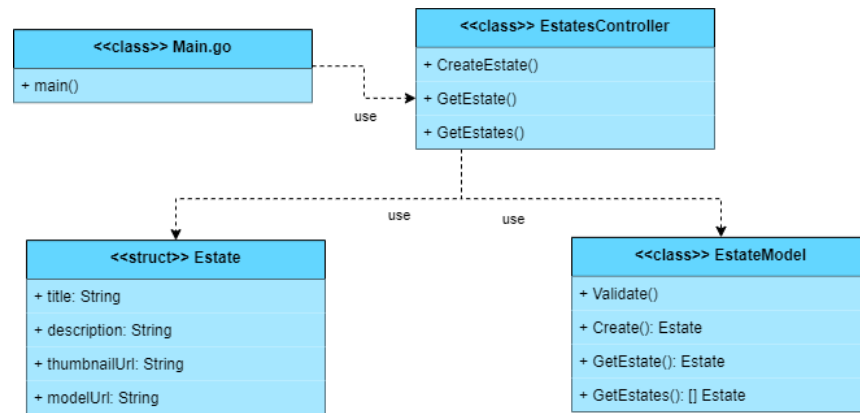
5.1.4 Class Diagram

Class diagram bertujuan untuk memodelkan struktur dengan mendeskripsikan kelas-kelas beserta atribut, operasi, dan hubungan antar kelasnya terhadap sistem yang sedang dirancang. Pada bagian ini, terdapat dua *class diagram* yaitu *class diagram* untuk *web service* yang ada pada Gambar 5.7 dan *class diagram* untuk aplikasi berbasis Android seperti pada Gambar 5.8.

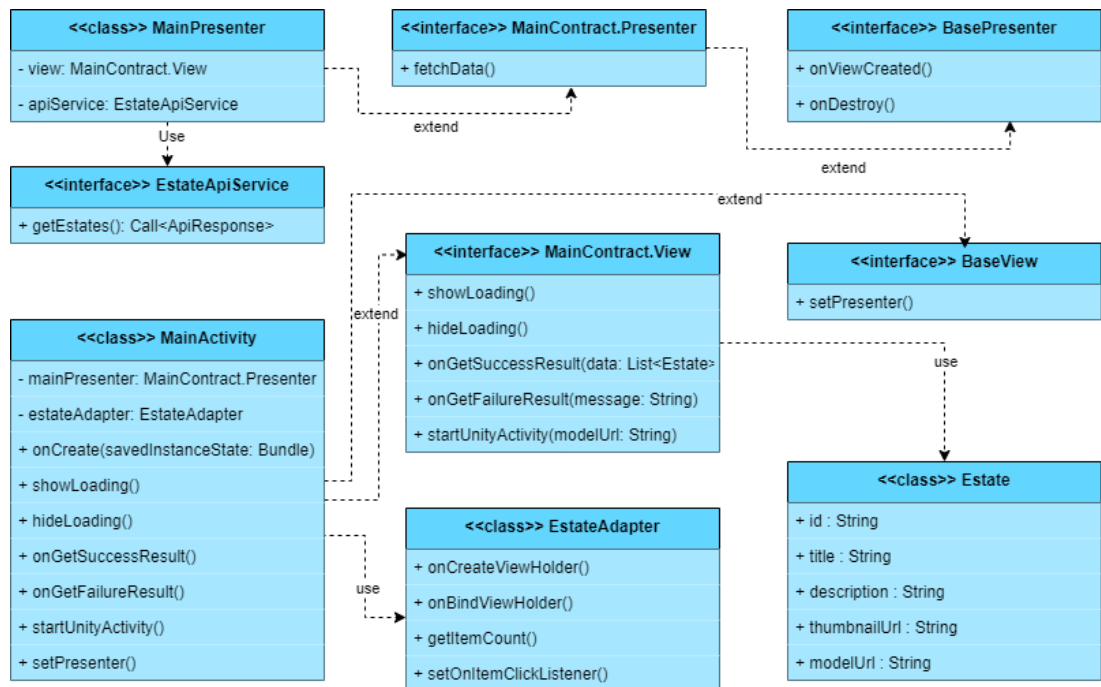
Pada *class diagram* untuk aplikasi berbasis Android, terdapat pemisahan antara View dan Presenter. Hal ini merupakan dampak dari penggunaan pola arsitektur MVP. Kelas Activity yang berperan sebagai View dihubungkan dengan kelas Presenter yang berisi mengenai *business logic*.

5.1.5 Perancangan Basis Data

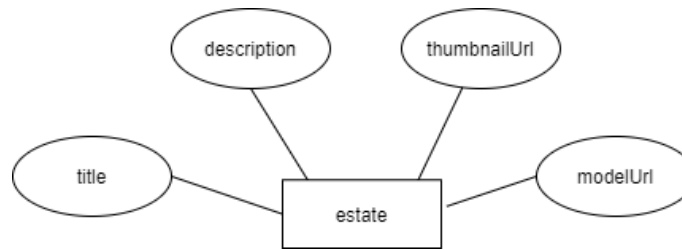
Perancangan basis data dilakukan untuk menentukan bagaimana data diorganisasi dalam *database*. Basis data dirancang dalam bentuk *entity-relationship diagram* (ERD) seperti yang tergambar pada Gambar 5.9. Kemudian, hasil ERD akan dinormalisasi satu table yang didefinisikan pada Tabel 5.1.



Gambar 5.7 Class Diagram API Service



Gambar 5.8 Class Diagram Aplikasi Android



Gambar 5.9 Rancangan Basis Data dalam ERD

Tabel 5.1 Rancangan Tabel *estate*

No.	Nama Kolom	Tipe Data	Deskripsi
1	id	Int	ID rumah
2	description	String	Deskripsi rumah
3	thumbnailUrl	String	Alamat situs gambar yang akan ditampilkan pada <i>item</i> daftar rumah.
4	modelUrl	String	Alamat situs model 3 dimensi yang akan ditampilkan pada aplikasi unity.

5.1.6 Perancangan Algoritme

Perancangan algoritme dilakukan untuk memberikan gambaran pada tahap implementasi terhadap alur kerja sistem. Algoritme utama yang dijelaskan pada Tabel 5.2 adalah algoritme untuk mengunduh model 3 dimensi rumah. Kemudian algoritme selanjutnya dijelaskan pada Tabel 5.3 untuk mendeteksi objek perabotan yang berada di dekat *player* pada aplikasi unity yang dikembangkan.

Tabel 5.2 Rancangan Algoritme Mengunduh Model 3 Dimensi Rumah

1	UnityWebRequest www =
2	UnityWebRequestAssetBundle.GetAssetBundle(bundleURL);
3	yield return www.SendWebRequest();
4	
5	StartCoroutine(ShowDownloadProgress(www));
6	
7	if (www.isNetworkError)
8	{
9	Debug.Log("Network error");
10	}
11	else
12	{
13	AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(www);
14	if (bundle != null)

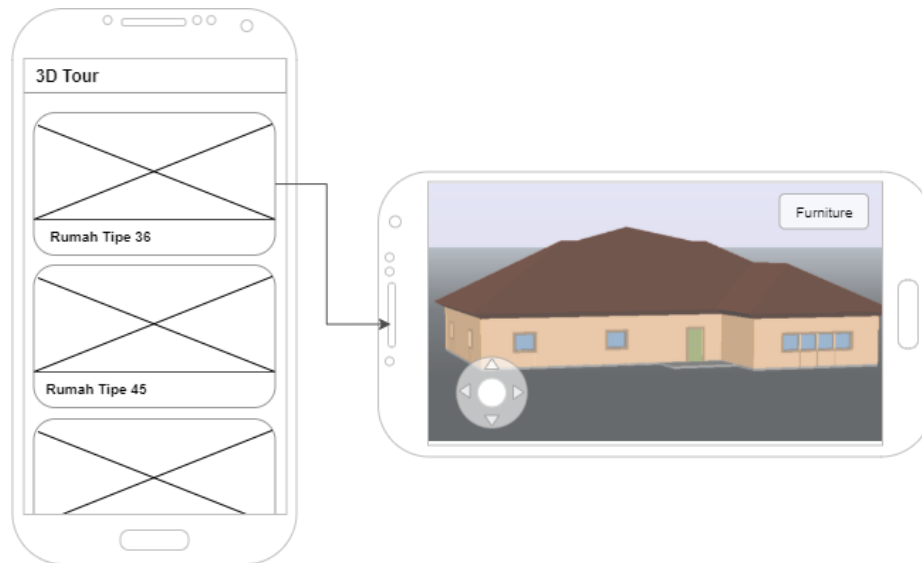
15	{
16	string rootAssetPath = bundle.GetAllAssetNames()[0];
17	GameObject arObject = Instantiate(bundle.LoadAsset(rootAssetPath)
18	as GameObject, bundleParent);
19	bundle.Unload(false);
20	callback(arObject);
21	}
22	else
23	{
24	Debug.Log("Not a valid asset bundle");
25	}
26	return results
27	}

Tabel 5.3 Rancangan Algoritme Mendeteksi Objek Perabotan

1	RaycastHit hit;
2	Ray ray = playerController.playerCamera.ViewportPointToRay(new
	Vector3(0.5F, 0.5F, 0));
3	
4	if (Physics.Raycast(ray, out hit, 5f))
5	{
6	Transform objectHit = hit.transform;
7	
8	if (objectHit.CompareTag("Respawn"))
9	{
10	if ((detectedItem == null detectedItem.transform != objectHit)
	&& objectHit.GetComponent<SC_PickItem>() != null)
11	{
12	SC_PickItem itemTmp = objectHit.GetComponent<SC_PickItem>();
13	
14	//Check if item is in availableItemsList
15	for (int i = 0; i < availableItems.Length; i++)
16	{
17	if (availableItems[i].itemName == itemTmp.itemName)
	{
18	detectedItem = itemTmp;
19	detectedItemIndex = i;
20	}
21	}
22	}
23	}
24	else
25	{
26	detectedItem = null;
27	}
28	}
29	else
30	{
31	detectedItem = null;
32	}

5.1.7 Perancangan User Interface

Perancangan antarmuka bertujuan untuk memberi gambaran dasar terkait sistem yang akan digunakan oleh penggunanya. Hasil perancangan antarmuka dituangkan pada Gambar 5.10. Rancangan tersebut terdiri dari 2 halaman, yaitu halaman *home* sebagai halaman utama dan tur virtual.



Gambar 5.10 Hasil Perancangan Kerangka Antarmuka

5.2 Implementasi Perangkat Lunak

Aplikasi berbasis Android Tur Interaktif diprogram dengan menggunakan bahasa Kotlin, C# pada pengembangan dengan *Unity 3D Engine*, dan Go pada pengembangan *API Service*-nya. Adapun kode program yang di implementasikan untuk pengembangan aplikasi Androidnya dijelaskan pada Tabel 5.4 hingga Tabel 5.7.

Program tur interaktif yang dikembangkan menggunakan Unity 3D Engine diprogram dengan menggunakan bahasa C#. Implementasi kode C# dijelaskan pada tabel 5.8 hingga Tabel 5.11. Program *web service* diprogram dengan menggunakan bahasa Go. Implementasi kode program Go dijelaskan pada Tabel 5.12 hingga Tabel 5.14.

Tabel 5.4 Kode Program HomeActivity (Android)

1	<code>class HomeActivity : AppCompatActivity(), MainContract.View {</code>
2	<code> private lateinit var materialDialog: MaterialDialog</code>
3	<code> private lateinit var mainPresenter: MainContract.Presenter</code>

4	private lateinit var estateAdapter: EstateAdapter	
5		
6	override fun onCreate(savedInstanceState: Bundle?) {	
7	super.onCreate(savedInstanceState)	
8	setContentView(R.layout.activity_main)	
9		
10	setPresenter(
11	MainPresenter(
12	this,	
13	EstateApplication.getDependencyInjection(this).getApiService()	
14)	
15)	
16		
17	val layoutManager = LinearLayoutManager(this)	
18	layoutManager.orientation = LinearLayoutManager.VERTICAL	
19	recyclerView.layoutManager = layoutManager	
20		
21	estateAdapter = EstateAdapter(baseContext)	
22	estateAdapter.setOnItemClickListener(object	:
23	EstateAdapter.OnItemClickListener {	
24	override fun onItemClick(data: Estate) {	
25	startUnityActivity(data.modelUrl)	
26	}	
27	})	
28	recyclerView.adapter = estateAdapter	
29	mainPresenter.fetchData()	
30	}	
31		
32	override fun showLoading() {	
33	materialDialog = MaterialDialog.Builder(this)	
34	.title("Perumahan Balimbingan Permai")	
35	.content("loading...")	
36	.progress(true, 0)	
37	.show()	
38	}	
39		
40		
41	override fun hideLoading() {	
42	materialDialog.dismiss()	
43	}	
44		
45	override fun onGetSuccessResult(data: List<Estate>?) {	
46	estateAdapter.data = data	
47	estateAdapter.notifyDataSetChanged()	
48	}	
49		
50		
51	override fun onGetFailureResult(message: String) {	
52	Toast.makeText(this, message, Toast.LENGTH_SHORT).show()	
53	}	
54		
55	override fun startUnityActivity(modelUrl: String) {	
56	val intent = Intent(this, UnityPlayerActivity::class.java)	
57	intent.putExtra("modelUrl", modelUrl)	
58	startActivity(intent)	
59	}	
60		
61	override fun setPresenter(presenter: MainContract.Presenter) {	
62	mainPresenter = presenter	
63	}	
64		

65	}
----	---

Tabel 5.5 Kode Program HomePresenter (Android)

1	class HomePresenter(private val view: MainContract.View, private val
2	apiService: EstateApiService) :
3	MainContract.Presenter {
4	override fun fetchData() {
5	view.showLoading()
6	val call: Call<ApiResponse> = apiService.getEstates()
7	call.enqueue(object : Callback<ApiResponse> {
8	override fun onResponse(call: Call<ApiResponse>?, response:
9	Response<ApiResponse>?) {
10	try {
11	Log.d("SKRIPSHIT", response?.toString())
12	if (response!!.isSuccessful) {
13	val data: ApiResponse? = response.body()
14	view.onGetSuccessResult(data?.result)
15	view.hideLoading()
16	} else {
17	view.hideLoading()
18	view.onGetFailureResult(response.message())
19	}
20	} catch (e: Exception) {
21	view.hideLoading()
22	view.onGetFailureResult(e.message!!)
23	}
24	}
25	override fun onFailure(call: Call<ApiResponse>?, t:
26	Throwable?) {
27	view.hideLoading()
28	view.onGetFailureResult(t!!.message!!)
29	})
30	}
31	override fun onViewCreated() {
32	TODO("Not yet implemented")
33	}
34	override fun onDestroy() {
35	TODO("Not yet implemented")
36	}
37	}
38	
39	
40	
41	
42	
43	
44	
45	
46	}

Tabel 5.6 Kode Program EstateAdapter (Android)

1	class EstateAdapter(var context: Context) :
2	RecyclerView.Adapter<EstateAdapter.ViewHolder>() {

3	var data: List<Estate>? = ArrayList()
4	var itemClickListener: OnItemClickListener? = null
5	
6	override fun onCreateViewHolder(parent: ViewGroup, p1: Int):
7	ViewHolder {
8	val view =
9	LayoutInflater.from(context).inflate(R.layout.item_layout, parent, false);
10	return ViewHolder(view)
11	}
12	override fun onBindViewHolder(holder: ViewHolder, position: Int) {
13	holder.itemView.title.text = data?.get(position)?.title ?: ""
14	val options: RequestOptions = RequestOptions()
15	.centerCrop()
16	.placeholder(R.mipmap.ic_launcher_round)
17	.error(R.mipmap.ic_launcher_round)
18	
19	Glide.with(context).load(data?.get(position)?.thumbnailUrl).apply(options)
20	.into(holder.itemView.image)
21	}
22	
23	override fun getItemCount(): Int {
24	return data?.size!!
25	}
26	
27	fun setOnItemClickListener(itemClickListener: OnItemClickListener) {
28	this.itemClickListener = itemClickListener
29	}
30	
31	inner class ViewHolder(itemView: View) :
32	RecyclerView.ViewHolder(itemView) {
33	
34	init {
35	itemView.setOnClickListener {
36	data?.get(adapterPosition)?.let { it2 ->
37	itemClickListener?.onItemClicked(it2) }
38	}
39	}
40	
41	}
42	
43	interface OnItemClickListener {
44	fun onItemClicked(data: Estate)
45	}
46	
47	}

Tabel 5.7 Kode Program EstateApiService (Android)

1	interface EstateApiService {
2	@GET("api/estates/get")
3	fun getEstates(): Call<ApiResponse>
4	}

Tabel 5.8 Kode Program Main (Unity)

1	public class Main : MonoBehaviour
2	{

3	
4	private bool hasExtra = false;
5	public string modelUrl = "";
6	public AndroidJavaObject extras;
7	public AndroidJavaObject intent;
8	
9	void Start()
10	{
11	Application.targetFrameRate = 60;
12	
13	AndroidJavaClass UnityPlayer = new
14	AndroidJavaClass("com.unity3d.player.UnityPlayer");
15	AndroidJavaObject currentActivity =
16	UnityPlayer.GetStatic<AndroidJavaObject>("currentActivity");
17	intent = currentActivity.Call<AndroidJavaObject>("getIntent");
18	hasExtra = intent.Call<bool>("hasExtra", "modelUrl");
19	Debug.Log("start");
20	if (hasExtra)
21	{
22	Debug.Log("has extra");
23	extras = intent.Call<AndroidJavaObject>("getExtras");
24	modelUrl = extras.Call<string>("getString", "modelUrl");
25	Debug.Log("Model URL : " + modelUrl);
26	}
27	else
28	{
29	Debug.Log("no extra");
30	}
31	
32	void FixedUpdate()
33	{
34	if (Application.platform == RuntimePlatform.Android)
35	{
36	if (Input.GetKey(KeyCode.Escape))
37	{
38	Application.Quit();
39	}
40	}
41	}
42	}

Tabel 5.9 Kode Program FirstPersonController (Unity)

1	namespace Scripts
2	{
3	public class FirstPersonController : MonoBehaviour
4	{
5	public Camera playerCamera;
6	public Transform cameraTransform;
7	public CharacterController characterController;
8	public float cameraSensitivity;
9	public float moveSpeed;
10	public float moveInputDeadZone;
11	int leftFingerId, rightFingerId;
12	float halfScreenWidth;
13	Vector2 lookInput;
14	float cameraPitch;
15	Vector2 moveTouchStartPosition;
16	Vector2 moveInput;

```

17         protected Joystick Joystick;
18
19         void Start()
20         {
21             leftFingerId = -1;
22             rightFingerId = -1;
23             halfScreenWidth = Screen.width / 2;
24             moveInputDeadZone = Mathf.Pow(Screen.height /
moveInputDeadZone, 2);
25         }
26
27         void Awake()
28         {
29             Joystick = FindObjectOfType<Joystick>();
30         }
31
32         void Update()
33         {
34             GetTouchInput();
35
36             if (rightFingerId != -1)
37             {
38                 LookAround();
39             }
40
41             if (leftFingerId != -1)
42             {
43                 Vector2 movementDirection = Joystick.AxisNormalized *
moveSpeed * Time.deltaTime;
44                 characterController.Move(transform.right *
Joystick.AxisNormalized.x * 0.15f + transform.forward *
movementDirection.y);
45             }
46         }
47
48         void GetTouchInput()
49         {
50             for (int i = 0; i < Input.touchCount; i++)
51             {
52                 Touch t = Input.GetTouch(i);
53
54                 switch (t.phase)
55                 {
56                     case TouchPhase.Began:
57                         if (t.position.x < halfScreenWidth && leftFingerId
== -1)
58                         {
59                             leftFingerId = t.fingerId;
60                             moveTouchStartPosition = t.position;
61                         }
62                         else if (t.position.x > halfScreenWidth &&
rightFingerId == -1)
63                         {
64                             rightFingerId = t.fingerId;
65                         }
66                         break;
67                     case TouchPhase.Ended:
68                     case TouchPhase.Canceled:
69                         if (t.fingerId == leftFingerId)
70
71
72
73

```

74	{
75	leftFingerId = -1;
76	}
77	else if (t.fingerId == rightFingerId)
78	{
79	rightFingerId = -1;
80	}
81	
82	break;
83	case TouchPhase.Moved:
84	if (t.fingerId == rightFingerId)
85	{
86	lookInput = t.deltaPosition * cameraSensitivity * Time.deltaTime;
87	}
88	else if (t.fingerId == leftFingerId)
89	{
90	moveInput = t.position - moveTouchStartPosition;
91	}
92	
93	break;
94	case TouchPhase.Stationary:
95	if (t.fingerId == rightFingerId)
96	{
97	lookInput = Vector2.zero;
98	}
99	break;
100	}
101	}
102	}
103	
104	void LookAround()
105	{
106	cameraPitch = Mathf.Clamp(cameraPitch - lookInput.y, -90f, 90f);
107	playerCamera.transform.localRotation = Quaternion.Euler(cameraPitch, 0, 0);
108	
109	transform.Rotate(transform.up, lookInput.x);
110	}
111	
112	void Move()
113	{
114	if (moveInput.sqrMagnitude <= moveInputDeadZone) return;
115	
116	Vector2 movementDirection = moveInput.normalized * moveSpeed * Time.deltaTime;
117	characterController.Move(transform.right * movementDirection.x + transform.forward * movementDirection.y);
118	
119	}
120	
121	}
122	}

Tabel 5.10 Kode Program API untuk mengunduh model 3 dimensi rumah (Unity)

1	public class API : MonoBehaviour
2	{
3	public Text ProgressIndicator;

```

4      public Image LoadingBar;
5
6      public void GetBundleObject(string assetName, UnityAction<GameObject>
callback, Transform bundleParent)
7      {
8          StartCoroutine(GetDisplayBundleRoutine(assetName,          callback,
bundleParent));
9      }
10
11     string GetFilePath(string url)
12     {
13         string[] pieces = url.Split('/');
14         string filename = pieces[pieces.Length - 1];
15         return Path.Combine(Application.persistentDataPath, filename);
16     }
17
18     IEnumerator          GetDisplayBundleRoutine(string          bundleURL,
UnityAction<GameObject> callback, Transform bundleParent)
19     {
20         Debug.Log("Requesting bundle at " + bundleURL);
21         UnityWebRequest          www          =
UnityWebRequestAssetBundle.GetAssetBundle(bundleURL);
22         yield return www.SendWebRequest();
23
24         StartCoroutine(ShowDownloadProgress(www));
25
26         if (www.isNetworkError)
27         {
28             Debug.Log("Network error");
29         }
30         else
31         {
32             AssetBundle          bundle          =
DownloadHandlerAssetBundle.GetContent(www);
33             if (bundle != null)
34             {
35                 string rootAssetPath = bundle.GetAllAssetNames()[0];
36                 GameObject          arObject          =
Instantiate(bundle.LoadAsset(rootAssetPath) as GameObject, bundleParent);
37                 bundle.Unload(false);
38                 callback(arObject);
39             }
40             else
41             {
42                 Debug.Log("Not a valid asset bundle");
43             }
44         }
45     }
46
47     IEnumerator ShowDownloadProgress(UnityWebRequest req)
48     {
49         float downloadDataProgress;
50         while (!req.isDone)
51         {
52             downloadDataProgress = req.downloadProgress * 100;
53             Debug.Log("Download: " + downloadDataProgress);
54
55             ProgressIndicator.text          =
((int)downloadDataProgress).ToString() + "%";
56             LoadingBar.fillAmount = downloadDataProgress / 100;
57
58             yield return null;
59         }

```

60	}
61	}

Tabel 5.11 Kode Program Inventory Perabotan (Unity)

1	namespace Scripts
2	{
3	public class SC_InventorySystem : MonoBehaviour
4	{
5	public Texture crosshairTexture;
6	public FirstPersonController playerController;
7	public SC_PickItem[] availableItems; //List with Prefabs of all
8	the available items
9	int[] itemSlots = new int[6];
10	bool showInventory = false;
11	SC_PickItem detectedItem;
12	int detectedItemIndex;
13	Vector2 firstTouchPosition;
14	List<TouchField> touchedFields = new List<TouchField>();
15	GameObject inventory;
16	void Start()
17	{
18	for (int i = 0; i < itemSlots.Length; i++)
19	{
20	itemSlots[i] = -1;
21	}
22	for (int i = 0; i < availableItems.Length; i++)
23	{
24	itemSlots[i] = i;
25	}
26	inventory = GameObject.Find("Inventory");
27	inventory.SetActive(false);
28	}
29	void Update()
30	{
31	if (showInventory && touchedFields.Count > 0)
32	{
33	TouchField touchField = touchedFields[0];
34	for (int j = 0; j < availableItems.Length; j++)
35	{
36	if (touchField.itemName == availableItems[j].itemName)
37	{
38	RaycastHit hit;
39	Ray ray
40	playerController.playerCamera.ViewportPointToRay(new Vector3(0.5F, 0.5F,
41	0));
42	if (Physics.Raycast(ray, out hit, Mathf.Infinity))
43	{
44	Transform objectHit = hit.transform;
45	Vector3 hitPoint = hit.point;
46	hitPoint += objectHit.up;
47	Instantiate(availableItems[j], hitPoint,
48	transform.rotation);
49	}
50	}
51	}
52	}
53	}

```

54 // Instantiate(availableItems[j],
playerController.playerCamera.transform.position +
(playerController.playerCamera.transform.forward, Quaternion.identity);
55 }
56
57 break;
58 }
59 }
60 touchedFields.RemoveAt(0);
61 }
62 }
63
64 public void addTouchedField(TouchField touchField)
65 {
66     touchedFields.Add(touchField);
67 }
68
69 public void removeItemClicked()
70 {
71     if (detectedItem && detectedItemIndex > -1)
72     {
73         detectedItem.PickItem();
74     }
75 }
76
77 public void toggleShowInventory()
78 {
79     showInventory = !showInventory;
80     inventory.SetActive(showInventory);
81 }
82
83 void FixedUpdate()
84 {
85     RaycastHit hit;
86     Ray ray = playerController.playerCamera.ViewportPointToRay(new
Vector3(0.5F, 0.5F, 0));
87
88     if (Physics.Raycast(ray, out hit, 5f))
89     {
90         Transform objectHit = hit.transform;
91
92         if (objectHit.CompareTag("Respawn"))
93         {
94             if ((detectedItem == null || detectedItem.transform !=
objectHit) && objectHit.GetComponent<SC_PickItem>() != null)
95             {
96                 SC_PickItem itemTmp =
objectHit.GetComponent<SC_PickItem>();
97
98                 for (int i = 0; i < availableItems.Length; i++)
99                 {
100                     if (availableItems[i].itemName ==
itemTmp.itemName)
101                     {
102                         detectedItem = itemTmp;
103                         detectedItemIndex = i;
104                     }
105                 }
106             }
107         }
108         else
109         {
110             detectedItem = null;

```

```

111         }
112     }
113     else
114     {
115         detectedItem = null;
116     }
117 }
118
119 void OnGUI()
120 {
121     GUI.color = detectedItem ? Color.green : Color.white;
122     GUI.DrawTexture(new Rect(Screen.width / 2 - 4, Screen.height /
123 2 - 4, 8, 8), crosshairTexture);
124     GUI.color = Color.white;
125
126     if (detectedItem)
127     {
128         GUI.color = new Color(0, 0, 0, 0.84f);
129         GUI.Label(new Rect(Screen.width / 2 - 75 + 1,
130 Screen.height / 2 - 50 + 1, 150, 20), detectedItem.objectName);
131         GUI.color = Color.green;
132         GUI.Label(new Rect(Screen.width / 2 - 75, Screen.height /
133 2 - 50, 150, 20), detectedItem.objectName);
134     }
135 }

```

Tabel 5.12 Kode Program Main.go (API Service)

```

1 package main
2 import (
3     "fmt"
4     "net/http"
5     "os"
6     "simple-api/app"
7     "simple-api/controllers"
8     "github.com/gorilla/mux"
9 )
10
11 func main() {
12     router := mux.NewRouter()
13     router.HandleFunc("/api/user/new",
14 controllers.CreateAccount).Methods("POST")
15     router.HandleFunc("/api/user/login",
16 controllers.Authenticate).Methods("POST")
17     router.HandleFunc("/api/estates/new",
18 controllers.CreateEstate).Methods("POST")
19     router.HandleFunc("/api/estate/{id}",
20 controllers.GetEstate).Methods("GET")
21     router.HandleFunc("/api/estates/get",
22 controllers.GetEstates).Methods("GET") // e.g : user/2/contacts
23
24     //attach JWT auth middleware
25     router.Use(app.JwtAuthentication)
26
27     port := os.Getenv("PORT")
28     if port == "" {
29         port = "8000" //localhost
30     }
31 }

```

27	fmt.Println(port)
28	err := http.ListenAndServe(":"+port, router) //Launch the app,
	visit localhost:8000/api
29	if err != nil {
30	fmt.Print(err)
31	}
32	}

Tabel 5.13 Kode Program EstateController.go (API Service)

1	package controllers
2	
3	import (
4	"encoding/json"
5	"net/http"
6	"simple-api/models"
7	u "simple-api/utils"
8	"strconv"
9	"github.com/gorilla/mux"
10)
11	
12	var CreateEstate = func(w http.ResponseWriter, r *http.Request) {
13	estate := &models.Estate{}
14	
15	err := json.NewDecoder(r.Body).Decode(estate)
16	if err != nil {
17	u.Respond(w, u.Message(false, "Error while decoding request
	body"))
18	return
19	}
20	
21	resp := estate.Create()
22	u.Respond(w, resp)
23	}
24	
25	var GetEstate = func(w http.ResponseWriter, r *http.Request) {
26	params := mux.Vars(r)
27	id, err := strconv.Atoi(params["id"])
28	
29	if err != nil {
30	u.Respond(w, u.Message(false, "There was an error in your
	request"))
31	return
32	}
33	
34	data := models.GetEstate(uint(id))
35	resp := u.Message(true, "success")
36	resp["result"] = data
37	u.Respond(w, resp)
38	}
39	
40	var GetEstates = func(w http.ResponseWriter, r *http.Request) {
41	data := models.GetEstates()
42	resp := u.Message(true, "success")
43	resp["result"] = data
44	u.Respond(w, resp)
45	}

Tabel 5.14 Kode Program Estate.go (API Service)

```

1 package models
2
3 import (
4     "fmt"
5     u "simple-api/utils"
6     "github.com/jinzhu/gorm"
7 )
8
9 type Estate struct {
10     gorm.Model
11     Title      string `json:"title"`
12     Description string `json:"description"`
13     ThumbnailUrl string `json:"thumbnailUrl"`
14     ModelUrl    string `json:"modelUrl"`
15 }
16
17 func (estate *Estate) Validate() (map[string]interface{}, bool) {
18     if estate.Title == "" {
19         return u.Message(false, "Estate title should be on the
20 payload"), false
21     }
22     if estate.Description == "" {
23         return u.Message(false, "Description should be on the
24 payload"), false
25     }
26     return u.Message(true, "success"), true
27 }
28
29 func (estate *Estate) Create() map[string]interface{} {
30     if resp, ok := estate.Validate(); !ok {
31         return resp
32     }
33
34     GetDB().Create(estate)
35
36     resp := u.Message(true, "success")
37     resp["estate"] = estate
38     return resp
39 }
40
41 func GetEstate(id uint) *Estate {
42     estate := &Estate{}
43     err := GetDB().Table("estates").Where("id = ?",
44 id).First(estate).Error
45     if err != nil {
46         fmt.Println(err)
47         return nil
48     }
49     return estate
50 }
51
52 func GetEstates() []*Estate {
53     estates := make([]*Estate, 0)
54     err := GetDB().Table("estates").Find(&estates).Error
55     if err != nil {
56         fmt.Println(err)
57         return nil
58     }
59 }

```

58	
60	return estates
61	}

DAFTAR PUSTAKA

- Hendro Trieddiantoro Puto (2015). Kajian Virtual Reality. Makalah Studi Mandiri: Universitas Teknologi Yogyakarta, Januari 2015.
- Sherman, W. R., Craig, A. B. (2003). Understanding Virtual Reality. Morgan Kaufmann Publishing: Interface Application and Design. San Fransisco, CA.
- Amin, D. & Golvikar, R. (2015). Comparative Study of Augmented Reality SKD's . International Journal on Computational Sciences & Application (IJCSA), Hal 2-7.
- Ozacar, K., Ortakci, Y., dkk (2017). A Low-Cost and Lightweight 3D Interactive Real Estate-Purposed Indoor Virtual Reality Application. International Journal on Photogrammetry, Remote Sensing and Spatial Information Sciences. Hal 308-309.
- Fitrana, E. A., dkk (2019). Pengembangan Aplikasi Katalog Rumah Berbasis Augmented Reality Menggunakan Algoritma FAST. JISKa. Vol. 4, No. 1. Hal 10. ISSN : 2527-5836.
- Husniah, L. dkk. (2016). Interaktif Augmented Reality untuk Katalog Penjualan Rumah Berbasis Android. KINETIK. Vol. 1, No. 1. Hal 33. ISSN : 2503-2259.
- Sutrisno, Adam dkk. (2015). Implementasi Teknologi Augmented Reality pada Agen Penjualan Rumah. E-Journal Teknik Elektro dan Komputer. Hal 19-20. ISSN : 2301-8402.
- Rumajar, R. (2015). Perancangan Brosur Interaktif Berbasis Augmented Reality. E-Journal Teknik Elektro dan Komputer. Vol 4, No. 6. Hal 1-3. ISSN : 2301-8402.
- Wirawan, Raden dkk. (2016). Aplikasi Augmented Reality pada Sistem Informasi Smart Building. JNTETI. Vol. 5, No. 3. Hal 20. ISSN : 2301 – 4156
- Suhendar, Akip dkk. (2016). Aplikasi Virtual tour Berbasis Multimedia Interaktif Menggunakan AutoDesk 3Ds Max. Jurnal ProTekInfo. Vol. 3, No. 1. Hal 30-35. ISSN : 2406-7741.

Pramono, Basworo Ardi, 2012, Desain dan Implementasi Augmented Reality Berbasis Web Pada Aplikasi Furniture Shopping Manager Sebagai Alat Bantu Belanja Online. Jurnal Transformatika. Vol.10, No.1, Hal 28.

Gede Wahya Dhiyatmika, I., Putra, I., & Mandenni, N. (2015). Aplikasi Augmented Reality Magic Book Pengenalan Binatang Untuk Siswa TK. Lontar Komputer: Jurnal Ilmiah Teknologi Informasi, 120-127.

LAMPIRAN

DATA CATATAN PENGUNJUNG DAN HARGA JUAL PERUMAHAN BALIMBINGAN PERMAI PT. KARYA PROPERTINDO UTAMA

Berikut data pengunjung (home seeker) perumahan balimbingan, yang ingin membeli rumah namun batal. Terdapat data nama, tahun, tipe rumah yang ingin dibeli, serta kendala / alasan batal membeli rumah, seperti tertera pada tabel di bawah.

No	NAMA	ALAMAT / DOMISILI	TAHUN	TIPE RUMAH	KENDALA
1	Ronald Heron Siahahan	Palembang	2016	45	Batal berkunjung / survey dikarenakan jarak jauh, meminta model rumah berbentuk 3-dimensi.
2	Handoko	Kebun PTP IV Balimbingan	2016	45	Batal Berkunjung karena kerja kantor, tidak ada waktu kosong untuk survey lapangan
3	Riko Silalahi	Balige - Kab. Toba Samosir	2016	36	Batal Berkunjung karena kerja kantor, tidak ada waktu kosong untuk survey lapangan, meminta model rumah berbentuk 3-dimensi
4	Nihon Halawa	Bah Kisat - Tanah Jawa	2016	54	Tanpa kabar kembali
5	Jhon Harman Sinaga	Jakarta	2017	54	Batal berkunjung dikarenakan masalah jarak yang jauh untuk berkunjung, meminta model rumah berbentuk 3-dimensi
6	Jagusman Siregar	P. Siantar	2017	70	Batal berkunjung karena tidak ada waktu libur kerja kantor
7	Effendi Sitanggang	Tanah Jawa	2017	36	Tanpa kabar kembali
8	Muchtar Simanjuntak	P. Siantar	2017	54	Tidak jadi berkunjung
9	Silvi	P. Siantar	2017	45	Tidak jadi berkunjung
10	Sriantis Butar-Butar	Balige	2017	45	Tidak jadi berkunjung
11	Sariati Silitonga	Balimbingan	2017	45	Tanpa Alasan
12	Helmiati Tanjung	Rumah Sakit Balimbingan	2017	54	Tanpa kabar kembali
13	Renata Siagian	P. Siantar	2017	54	Tanpa kabar kembali
14	Judir Nadapdap	Tanjung Pasir - Tanah Jawa	2017	45	Tanpa kabar kembali
15	Desi Silalahi	Jakarta	2018	45	Batal berkunjung / survey dikarenakan jarak jauh, meminta model rumah berbentuk 3-dimensi.
16	Azis	Tonduhan - Kebun	2018	54	Tidak jadi berkunjung karena kerja kantor kebun
17	Arifin Sipayung	Tanah Jawa	2018	54	Tanpa kabar kembali
18	Lusi Simanjuntak	Tanah Jawa	2018	45	Tanpa kabar kembali

19	Robin Sinabariba	Balimbingan	2018	45	Tanpa kabar kembali
20	Darwis	TebingTinggi	16 Maret 2019	36	Batal survey dikarenakan masalah jarak yang jauh untuk berkunjung, minta 3D
21	Erick Tampubolon	Perdagangan	2019	36	Batal Berkunjung karena kerja kantor, tidak ada waktu kosong untuk survey lapangan
22	Pungua Sidabutar	Hutabayu	2019	54	Tanpa kabar kembali
23	Heuda P. Silitonga	Medan	2019	36	Batal Survey kendala jarak jauh, Minta 3D
24	Maslon Sirait	Kebun PTP IV Mandoge	2020	70	Batal berkunjung karena membatasi diri keluar rumah akibat covid-19
25	Riski Silalahi	P. Siantar	2020	54	Perlu model 3D rumah, membatasi diri untuk berkunjung karena pandedmi Covid-19
26	Bangun Sihombing, S.STP.M.Si	P. Siantar	2020	54	Perlu Model 3D
27	Betterpen. S	Huta Bayu	2020	36	Perlu Model 3D

Berikut data harga jual perumahan balimbingan permai, seperti pada tabel di bawah yang ditampilkan berdasarkan tipe, harga, beserta estimasi keuntungan.

No	Tipe Rumah	Harga Jual	Estimasi Keuntungan
1	36	Rp 130.000.000	Rp 18.000.000
2	45	Rp 160.000.000	Rp 30.000.000
3	54	Rp. 210.000.000	Rp. 45.000.000
4	70	Rp. 260.000.000	Rp. 52.000.000

Demikian data yang tertera di atas adalah benar-benar data perumahan Balimbingan Permai yang sah diberikan oleh perusahaan PT. Karya Propertindo Utama dan akan digunakan sebagai data pendukung untuk penelitian skripsi.

Balimbingan, 21 September 2020

PT. Karya Propertindo Utama,


 Ir. Mangantar Silalahi, MT
 Pemilik Proyek