**PEMBANGUNAN APLIKASI TUR INTERAKTIF**

**PERUMAHAN BALIMBINGAN PERMAI PT. KARYA PROPERTINDO UTAMA BERBASIS VIRTUAL REALITY**

**SKRIPSI**

Keminatan Multimedia, Game, dan Mobile

Disusun oleh  
REINHARD JONATHAN SLALAHI  
175150200111040



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

**MALANG  
2020**

**DAFTAR ISI**

[BAB 1 PENDAHULUAN 6](#_Toc74479376)

[1.1 Latar Belakang 6](#_Toc74479377)

[1.2 Identifikasi Masalah 10](#_Toc74479378)

[1.3 Rumusan Masalah 10](#_Toc74479379)

[1.4 Tujuan Penelitian 10](#_Toc74479380)

[1.5 Manfaat Penelitian 11](#_Toc74479381)

[1.6 Batasan Penelitian/Ruang Lingkup Penelitian 11](#_Toc74479382)

[1.7 Sistematika Pembahasan/Laporan 12](#_Toc74479383)

[BAB 2 LANDASAN KEPUSTAKAAN 14](#_Toc74479384)

[2.1 KAJIAN PUSTAKA 14](#_Toc74479385)

[2.2 Virtual Reality 16](#_Toc74479386)

[2.3 Unity 3D 16](#_Toc74479387)

[2.4 Android 17](#_Toc74479388)

[2.5 Pengujian Fungsional 17](#_Toc74479389)

[2.6 Pengujian *Usability* 17](#_Toc74479390)

[BAB 3 METODOLOGI PENELITIAN 18](#_Toc74479391)

[3.1 Studi Literatur 19](#_Toc74479392)

[3.2 Analisis Kebutuhan Sistem 20](#_Toc74479393)

[3.3 Perancangan Sistem 20](#_Toc74479394)

[3.4 Implementasi Sistem 20](#_Toc74479395)

[3.5 Pengujian Sistem 21](#_Toc74479396)

[3.6 Kesimpulan dan Saran 21](#_Toc74479397)

[BAB 4 REKAYASA KEBUTUHAN 22](#_Toc74479398)

[4.1. Gambaran Umum Aplikasi 22](#_Toc74479399)

[4.1.1. Deskripsi 22](#_Toc74479400)

[4.1.2. Cara Penggunaan 22](#_Toc74479401)

[4.2. Analisis Kebutuhan Sistem 22](#_Toc74479402)

[4.2.1 Identifikasi Aktor 23](#_Toc74479403)

[4.2.2 Kebutuhan Fungsional 23](#_Toc74479404)

[4.2.3 Kebutuhan Non Fungsional 24](#_Toc74479405)

[4.3. Pemodelan Kebutuhan 24](#_Toc74479406)

[*4.3.1.* *Use Case Diagram* 24](#_Toc74479407)

[*4.3.2.* *Use case Scenario* 25](#_Toc74479408)

[BAB 5 PERANCANGAN DAN IMPLEMENTASI 27](#_Toc74479409)

[5.1 Perancangan 27](#_Toc74479410)

[5.1.1 Perancangan Arsitektur Sistem 27](#_Toc74479411)

[5.1.2 Perancangan Komponen 28](#_Toc74479412)

[*5.1.3* *Sequence Diagram* 28](#_Toc74479413)

[5.1.4 Class Diagram 30](#_Toc74479414)

[5.1.5 Perancangan Basis Data 31](#_Toc74479415)

[5.1.6 Perancangan Algoritme 32](#_Toc74479416)

[5.1.7 Perancangan User Interface 33](#_Toc74479417)

[5.2 Implementasi Perangkat Lunak 34](#_Toc74479418)

[5.2.1 Spesifikasi Sistem 34](#_Toc74479419)

[5.2.2 Batasan Implementasi 37](#_Toc74479420)

[5.2.3 Implementasi Basis Data 37](#_Toc74479421)

[5.2.4 Implementasi Kode Program 37](#_Toc74479422)

[5.2.5 Implementasi Antarmuka 50](#_Toc74479423)

[BAB 6 PENGUJIAN 51](#_Toc74479424)

[DAFTAR PUSTAKA 52](#_Toc74479425)

[LAMPIRAN 54](#_Toc74479426)

**DAFTAR GAMBAR**

[Gambar 3.1 Metode Pengembangan 19](#_Toc74237570)

[Gambar 4.1 *Use Case Diagram* Pengguna 25](#_Toc74237571)

[Gambar 5.1 Rancangan Arsitektur Sistem 27](#_Toc74237572)

[Gambar 5.2 Rancangan *Activity Diagram* Menampilkan Daftar Rumah 29](#_Toc74237573)

[Gambar 5.3 Rancangan *Activity Diagram* Menampilkan Tur Interaktif 30](#_Toc74237574)

[Gambar 5.4 Sequence Diagram Menampilkan Daftar Rumah 31](#_Toc74237575)

[Gambar 5.5 Sequence Diagram Menampilkan Tur Interaktif 32](#_Toc74237576)

[Gambar 5.6 Sequence Diagram Menampilkan Daftar Rumah (API Service) 32](#_Toc74237577)

[Gambar 5.7 *Class Diagram* API Service 34](#_Toc74237578)

[Gambar 5.8 *Class Diagram* Aplikasi Android 34](#_Toc74237579)

[Gambar 5.9 Rancangan Basis Data dalam ERD 35](#_Toc74237580)

[Gambar 5.10 Hasil Perancangan Kerangka Antarmuka 37](#_Toc74237581)

**DAFTAR TABEL**

[Tabel 4.1 Identifikasi Aktor 23](#_Toc74389422)

[Tabel 4.2 Penjelasan Format Kode Kebutuhan 23](#_Toc74389423)

[Tabel 4.3 Kebutuhan Fungsional Sistem 23](#_Toc74389424)

[Tabel 4.4 Kebutuhan Non-Fungsional Sistem 24](#_Toc74389425)

[Tabel 4.5 Use Case Scenario Menampilkan Daftar Rumah 26](#_Toc74389426)

[Tabel 4.6 Use Case Scenario Menampilkan 3D Tour 26](#_Toc74389427)

[Tabel 5.1 Rancangan Tabel *estate* 33](#_Toc74389428)

[Tabel 5.7 Spesifikasi Perangkat Keras Lingkungan Pengembangan Sistem 36](#_Toc74389429)

[Tabel 5.8 Spesifikasi Perangkat Keras Lingkungan *Api Service* 36](#_Toc74389430)

[Tabel 5.9 Spesifikasi Perangkat Lunak Lingkungan Pengembangan Sistem 37](#_Toc74389431)

[Tabel 5.10 Spesifikasi Perangkat Lunak Lingkungan *Web Service* 37](#_Toc74389432)

[Tabel 5.11 DDL Basis Data Tur Interaktif 38](#_Toc74389433)

[Tabel 5.4 Kode Program HomeActivity (Android) 38](#_Toc74389434)

[Tabel 5.5 Kode Program HomePresenter (Android) 39](#_Toc74389435)

[Tabel 5.6 Kode Program EstateAdapter (Android) 40](#_Toc74389436)

[Tabel 5.7 Kode Program EstateApiService (Android) 41](#_Toc74389437)

[Tabel 5.8 Kode Program Main (Unity) 41](#_Toc74389438)

[Tabel 5.9 Kode Program FirstPersonController (Unity) 42](#_Toc74389439)

[Tabel 5.10 Kode Program API untuk mengunduh model 3 dimensi rumah (Unity) 44](#_Toc74389440)

[Tabel 5.11 Kode Program Inventory Perabotan (Unity) 45](#_Toc74389441)

[Tabel 5.12 Kode Program Main.go (API Service) 48](#_Toc74389442)

[Tabel 5.13 Kode Program EstateController.go (API Service) 49](#_Toc74389443)

[Tabel 5.14 Kode Program Estate.go (API Service) 49](#_Toc74389444)

# PENDAHULUAN

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 perusahan 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 berpotensial 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 menvisualisasikan 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 Blackbox Testing (pengujian fungsional), dan Pengujian non-fungsional (pengujian usability dan compatibility). Untuk pengujian usability peneliti melakukan pengumpulan data dengan memberikan kuesioner System Usability Scale (SUS) kepada responden calon pembeli / pengunjung perumahan.

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.

Rumusan Masalah

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

1. Bagaimana hasil analisis kebutuhan untuk sistem Aplikasi Tur Interaktif ?
2. Bagaimana hasil rancangan sistem Aplikasi Tur Interaktif dengan arsitektur MVP ?
3. Bagaimana hasil implementasi sistem Aplikasi Tur Interaktif dengan arsitektur MVP?
4. Bagaimana hasil pengujian *usability* dari Aplikasi Pameran dan Tur Virtual Perumahan Balimbingan Permai?
5. Bagaimana hasil pengujian *compatibility* dari Aplikasi Pameran dan Tur Virtual Perumahan Balimbingan Permai?

Tujuan Penelitian

1. Tujuan Umum

Dengan dibangunnya aplikasi berbasis Virtual Reality ini, maka diharapkan mampu mengatasi permasalahan pada konsumen yang memiliiki 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.

1. Tujuan Khusus
2. Dibangunnya Aplikasi berbasis Virtual Reality yang memiliki kemudahan dalam menampilkan model 3 dimensi perumahan dan melakukan tur secara virtual.
3. Mampu membantu konsumen untuk melakukan pengamatan jarak jauh dan membayangkan bentuk asli dari properti/rumah yang ingin dibeli.

Manfaat Penelitian

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

1. Konsumen
2. Mampu membantu konsumen untuk melakukan pengamatan jarak jauh melalui objek maya yang berbentuk 3 dimensi dan melakukan pengamatan lingkungan sekitar rumah melalui 3D tour.
3. Membantu konsumen untuk lebih mudah membandingkan tipe-tipe perumahan yang akan dibeli.
4. Peneliti
5. Mengetahui teknik membangun aplikasi secara umum.
6. Sebagai bahan referensi untuk penelitian dan pengembangan aplikasi Android selanjutnya.

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.

Sistematika Pembahasan/Laporan

* + 1. **BAB I Pendahuluan**

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

* + 1. **BAB II Landasan Kepustakaan**

Bab landasan kepustakaan 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. **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. **BAB IV Perancangan Sistem**

Bab ini membahas perancangan sistem menggunakan metode yang telah dipilih.

* + 1. **BAB V Implementasi Sistem**

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

* + 1. **BAB VI Pengujian**

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 ouput 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. **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.

# 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.

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 virtuall 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.

1. 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 imaginasi (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).

1. Unity 3D

Unity 3D merupakan sebuah tools yang terintegrasi untuk membuat bentuk obyek 3Dpada 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 dapatberjalan 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).

1. 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).

1. 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.

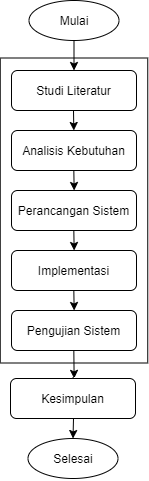
1. Pengujian *Usability*

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”.

# 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 hanyan 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 serat dapat dipastikan bahwa kebutuhan-kebutuhan dari aplikasi yang akan dikembangkan serta dapat dipastikan bahwa kebutuhan-kebutuhan yan 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

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.

1. 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 (Unfied 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.

1. Perancangan Sistem

Perancangan sistem digunakan untuk merancang tiap-tiap kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Tahap perancangan digunakan sebagai bahan perancangan/pembentukkan 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

1. 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.

1. 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 kesalahankesalahan 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).

1. 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.

# 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.

1. Gambaran Umum Aplikasi

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

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*.

* + 1. 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 interaktifsecara virtual.

1. 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.

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. |
| Admin | Aktor yang dapat melakukan penambahan data rumah yang akan tampil di halaman Daftar Rumah pada aplikasi di sisi pengguna. |

1. 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 *Software Requirement Specification*. |
| 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. |
| 3 | SRS-TIVR-F-3 | Menambah data rumah | Sistem dapat menambah data rumah |

1. 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

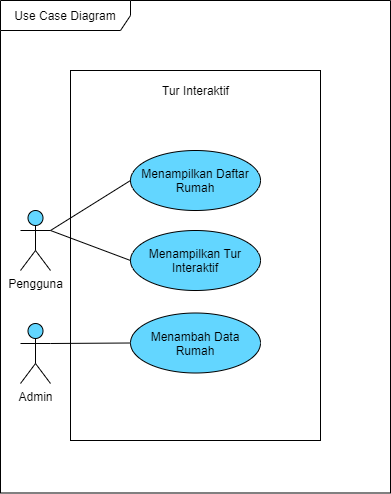
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kode Fungsi** | **Nama Fungsi** | **Deskripsi** |
| 1 | SRS-TIVR-NF-1 | *Usability* | *Usability* menjelaskan tentang kepuasan dan kemudahan pengguna terhadap pemakaian aplikasi. Usabilitas diangkat karena tujuan dari dikembangkannya aplikasi adalah untuk meningkatkan daya beli *home seeker*, maka diperlukan kebutuhan akan usabilitas yang menjamin kepuasan dan kemudahan pengguna dalam menggunakan aplikasi. |
| 2 | SRS-TIVR-NF-2 | *Compatibility* | *Compatibility* menjelaskan tentang banyaknya jenis perangkat atau versi teknologi yang bisa digunakan untuk menjalankan aplikasi. |

1. 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).

* + 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. Aktor dapat berupa pengguna maupun sistem eksternal yang dapat berinteraksi kepada sistem (Sukamto & Shalahuddin, 2016).



Gambar 4.1 *Use Case Diagram* Pengguna

* + 1. *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 *Use Case* | 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 | 1. Pengguna membuka halaman utama 2. 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

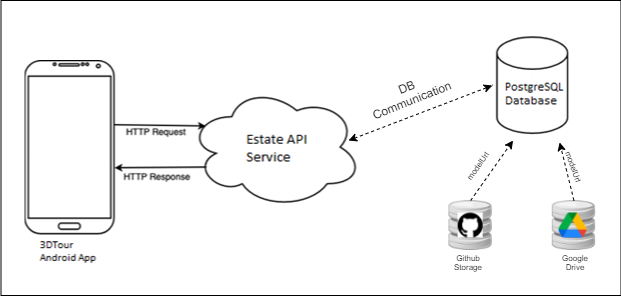
|  |  |
| --- | --- |
| Nama *Use Case* | Menampilkan 3D Tour |
| Kode Kebutuhan Terkait | SRS-TIVR-F-2 |
| Aktor | Pengguna |
| Tujuan | Menampilkan Tur Interaktif berbasis Virtual Reality dengan menampilkan objek 3 dimensi rumah. |
| Pre Condition | Aktor telah berada pada halaman utama |
| Main Flow | 1. Pengguna berada di halaman utama 2. Pengguna melakukan tap pada salah satu item rumah 3. Sistem menjalankan Tur Interaktif dari objek 3 dimensi rumah. |
| Post Condition | Sistem menampilkan Tur Interaktif *Unity* |
| Alternative Flow | - |

Tabel 4.7 *Use Case Scenario* Menambah Data Rumah

|  |  |
| --- | --- |
| Nama *Use Case* | Menambah Data Rumah |
| Kode Kebutuhan Terkait | SRS-TIVR-F-3 |
| Aktor | Admin |
| Tujuan | Menambah data rumah |
| Pre Condition | Aktor telah berada pada halaman utama |
| Main Flow | 1. Admin membuka halaman utama 2. Sistem menampilkan form tambah data rumah 3. Admin melakukan pengisian data 4. Admin melakukan proses submit dan sistem memproses *request* |
| Post Condition | Sistem menampilkan pesan berhasil menambah data |
| Alternative Flow | - |

# PERANCANGAN DAN IMPLEMENTASI

1. Perancangan
2. **Perancangan Arsitektur Sistem**



Gambar 5.1 Rancangan Arsitektur Sistem

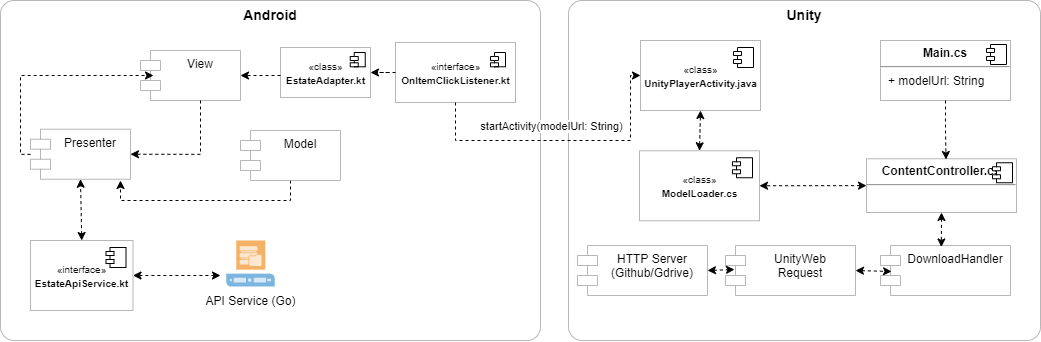
Rancangan arsitektur sistem yang digunakan pada penelitian ini digambarkan pada Gambar 5.1. Komunikasi yang digunakan untuk menghubungkan aplikasi 3DTour dengan web service adalah HTTP. Adapun proses pertukaran data menggunakan format JSON. Web service berfungsi untuk menyimpan data pada database serta 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.

1. Perancangan Komponen

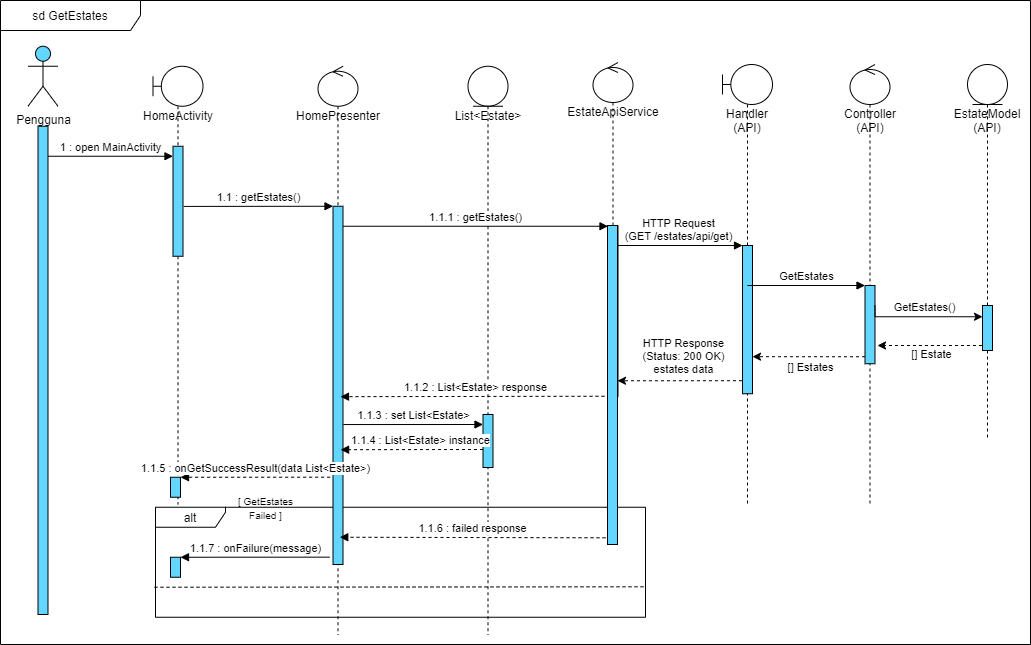
Rekayasa perangkat lunak berbasis komponen (CBSE), juga disebut pengembangan berbasis komponen (CBD), adalah cabang rekayasa perangkat lunak yang menekankan pemisahan perhatian sehubungan dengan fungsionalitas luas yang tersedia di seluruh sistem perangkat lunak yang diberikan. Ini adalah pendekatan berbasis penggunaan kembali untuk mendefinisikan, mengimplementasikan, dan menyusun komponen independen yang digabungkan secara longgar ke dalam sistem. Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Berikut diagram komponen dari sistem Aplikasi Tur Interaktif.



Gambar 5.2 Perancangan Komponen Android dan Unity

1. *Sequence Diagram*

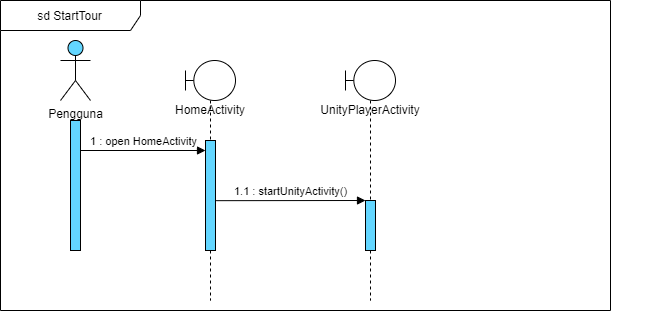
Sequence Diagram dibuat untuk menjelaskan interaksi kelas-kelas yang terlibat dalam satu alur kerja. Terdapat 2 buah sequence diagram yang dibuat untuk menjelaskan alur pada aplikasi Android dan 1 buah sequence diagram untuk menjelaskan alur pada api service.



Gambar 5.3 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. Aplikasi Tur 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.

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.



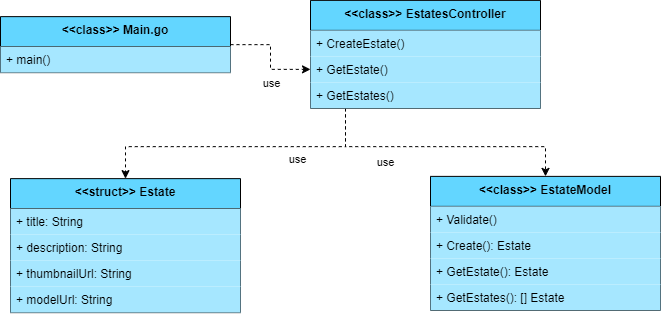
Gambar 5.4 Sequence Diagram Menampilkan Tur Interaktif

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.

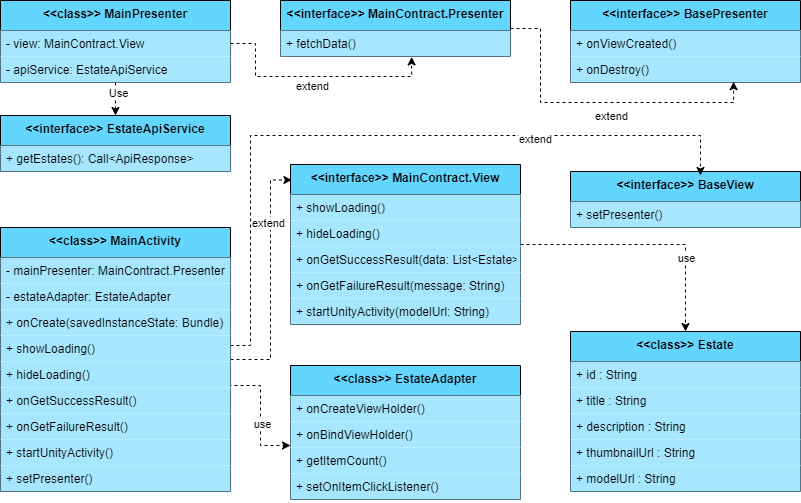
1. Class Diagram

*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*.



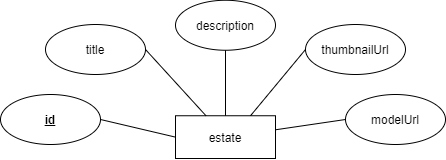
Gambar 5.5 *Class Diagram* API Service



Gambar 5.6 *Class Diagram* Aplikasi Android

1. 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 Rancangan Basis Data dalam ERD

Tabel 5.1 Rancangan Tabel *estate*

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Kolom | Tipe Data | Deskripsi |
| 1 | id | Int | ID rumah |
| 2 | title | String | Judul rumah |
| 3 | description | String | Deskripsi rumah |
| 4 | thumbnailUrl | String | Alamat situs gambar yang akan ditampilkan pada *item* daftar rumah. |
| 5 | modelUrl | String | Alamat situs model 3 dimensi yang akan ditampilkan pada aplikasi unity yang di-*embed­* di aplikasi Android. |

1. Perancangan Algoritme

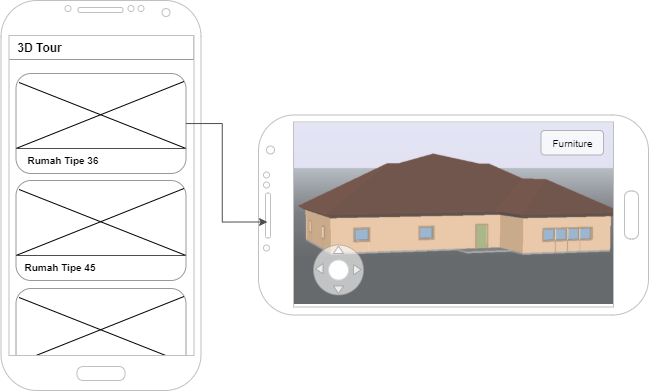
Perancangan algoritme dilakukan untuk memberikan gambaran pada tahap implementasi terhadap alur kerja sistem. Algoritme utama yang dijelaskan pada Tabel 5.2 adalah algoritme untuk mengunduh model 3 dimensi rumah. Kemudian algoritme selanjutnya dijelaskan pada Tabel 5.3 untuk mendeteksi objek perabotan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

Tabel 5.2 Rancangan Algoritme Mendeteksi Objek Perabotan

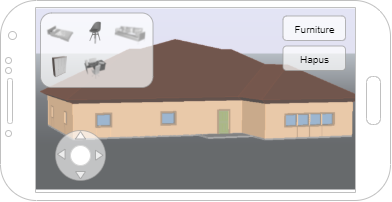
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | RaycastHit hit;  Ray ray = playerController.playerCamera.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));  if (Physics.Raycast(ray, out hit, 5f))  {  Transform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

1. 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 5.8 Hasil Perancangan Kerangka Antarmuka Daftar Rumah dan Tur Interaktif



Gambar 5.9 Hasil Perancangan Antarmuka Inventory

1. Implementasi Perangkat Lunak

Implementasi aplikasi Android Tur Interaktif menggunakan pola arsitektur MVP yang mana Activity berperan sebagai View dan terhubung dengan *class* Presenter yang berisi logika-logika alur. Kemudian, terdapat kelas Model yang berfungsi sebagai representasi entitas-entitas yang digunakan dalam sistem. Kemudian, program *api service* diimplementasikan dengan bahasa Go dan program yang dikembangkan menggunakan Unity 3D Engine menggunakan C#.

1. Spesifikasi Sistem

Bagian ini menjabarkan tentang lingkungan pengembangan sistem dan lingkungan di mana *api service* berjalan.

1. Spesifikasi Perangkat Keras

Sistem yang berupa aplikasi berbasis Android dan *web service* dikembangkan di lingkungan yang dijelaskan pada **Error! Reference source not found.**. Adapun **Error! Reference source not found.** menjabarkan spesifikasi perangkat keras di mana *web service* berjalan.

Tabel 5.3 Spesifikasi Perangkat Keras Lingkungan Pengembangan Sistem

|  |  |
| --- | --- |
| Perangkat | Asus Vivobook S14 S410 UN |
| Prosesor | 2,7 GHz Dual-Core Intel Core i5 |
| RAM | 8 GB 1867 MHz DDR3 |
| GPU | Nvidia MX 150 |
| Ruang Penyimpanan | 256 GB |
| OS | Windows 10 Home Single language |

Tabel 5.4 Spesifikasi Perangkat Keras Lingkungan *Api Service*

|  |  |
| --- | --- |
| Prosesor | 1 vCPU |
| RAM | 1 GB |
| Ruang Penyimpanan | 20 GB SSD |
| Lokasi | Singapura |

1. Spesifikasi Perangkat Lunak

Daftar perangkat lunak yang digunakan pada pengembangan sistem RestoCrowd dijabarkan pada

. Adapun lingkungan di mana *web service* berjalan dijabarkan pada **Error! Reference source not found.**.

Tabel 5.5 Spesifikasi Perangkat Lunak Lingkungan Pengembangan Sistem

|  |  |
| --- | --- |
| Pengolah Kata | Microsoft Word for Mac 2016 Ver 16.16.16. |
| *Editor* Pemrograman | 1. Android Studio 3.5 2. Visual Studio Code 3. Unity 3D Engine |
| Bahasa Pemrograman | 1. Kotlin 1.3.61 2. Go 1.12.5 3. C# |

Tabel 5.6 Spesifikasi Perangkat Lunak Lingkungan *Web Service*

|  |  |
| --- | --- |
| *Host* OS | Debian GNU/Linux 9.11 (Stretch) |
| DBMS | MariaDB 10.2.27 |
| *Container Engine* | Docker 19.03.4 Community Edition |
| Docker Image OS | Alpine Linux 3.10.3 |

1. Batasan Implementasi

Terdapat hal yang menjadi batasan implementasi pada pengembangan sistem TurInteraktif ini, yaitu aplikasi TurInteraktif ini hanya dapat berjalan pada *smartphone* berbasis Android dengan versi 8.0 ke atas.

1. Implementasi Basis Data

Tabel 5.7 DDL Basis Data Tur Interaktif

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | CREATE TABLE `estates` (  `id` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,  `title` varchar(255) NOT NULL,  `description` varchar(255) NOT NULL,  `thumbnailUrl` varchar(255) NOT NULL,  `modelUrl` varchar(255) NOT NULL,  PRIMARY KEY (`id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; |

1. Implementasi Kode Program

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 dikembangan 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 5.8 Kode Program HomeActivity (Android)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65 | class HomeActivity : AppCompatActivity(), MainContract.View {  private lateinit var materialDialog: MaterialDialog  private lateinit var mainPresenter: MainContract.Presenter  private lateinit var estateAdapter: EstateAdapter  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.activity\_main)  setPresenter(  MainPresenter(  this,  EstateApplication.getDependencyInjection(this).getApiService()  )  )  val layoutManager = LinearLayoutManager(this)  layoutManager.orientation = LinearLayoutManager.VERTICAL  recyclerView.layoutManager = layoutManager  estateAdapter = EstateAdapter(baseContext)  estateAdapter.setOnItemClickListener(object : EstateAdapter.OnItemClickListener {  override fun onItemClicked(data: Estate) {  startUnityActivity(data.modelUrl)  }  })  recyclerView.adapter = estateAdapter  mainPresenter.fetchData()  }  override fun showLoading() {  materialDialog = MaterialDialog.Builder(this)  .title("Perumahan Balimbingan Permai")  .content("loading...")  .progress(true, 0)  .show()  }  override fun hideLoading() {  materialDialog.dismiss()  }  override fun onGetSuccessResult(data: List<Estate>?) {  estateAdapter.data = data  estateAdapter.notifyDataSetChanged()  }  override fun onGetFailureResult(message: String) {  Toast.makeText(this, message, Toast.LENGTH\_SHORT).show()  }  override fun startUnityActivity(modelUrl: String) {  val intent = Intent(this, UnityPlayerActivity::class.java)  intent.putExtra("modelUrl", modelUrl)  startActivity(intent)  }  override fun setPresenter(presenter: MainContract.Presenter) {  mainPresenter = presenter  }  } |

Tabel 5.9 Kode Program HomePresenter (Android)

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

Tabel 5.10 Kode Program EstateAdapter (Android)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | class EstateAdapter(var context: Context) : RecyclerView.Adapter<EstateAdapter.ViewHolder>() {  var data: List<Estate>? = ArrayList()  var itemClickListener: OnItemClickListener? = null  override fun onCreateViewHolder(parent: ViewGroup, p1: Int): ViewHolder {  val view = LayoutInflater.from(context).inflate(R.layout.item\_layout, parent, false);  return ViewHolder(view)  }  override fun onBindViewHolder(holder: ViewHolder, position: Int) {  holder.itemView.title.text = data?.get(position)?.title ?: ""  val options: RequestOptions = RequestOptions()  .centerCrop()  .placeholder(R.mipmap.ic\_launcher\_round)  .error(R.mipmap.ic\_launcher\_round)  Glide.with(context).load(data?.get(position)?.thumbnailUrl).apply(options)  .into(holder.itemView.image)  }  override fun getItemCount(): Int {  return data?.size!!  }  fun setOnItemClickListener(itemClickListener: OnItemClickListener) {  this.itemClickListener = itemClickListener  }  inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  init {  itemView.setOnClickListener {  data?.get(adapterPosition)?.let { it2 -> itemClickListener?.onItemClicked(it2) }  }  }  }  interface OnItemClickListener {  fun onItemClicked(data: Estate)  }  } |

Tabel 5.11 Kode Program EstateApiService (Android)

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

Tabel 5.12 Kode Program Main (Unity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | public class Main : MonoBehaviour  {  private bool hasExtra = false;  public string modelUrl = "";  public AndroidJavaObject extras;  public AndroidJavaObject intent;  void Start()  {  Application.targetFrameRate = 60;    AndroidJavaClass UnityPlayer = new AndroidJavaClass("com.unity3d.player.UnityPlayer");  AndroidJavaObject currentActivity = UnityPlayer.GetStatic<AndroidJavaObject>("currentActivity");  intent = currentActivity.Call<AndroidJavaObject>("getIntent");  hasExtra = intent.Call<bool>("hasExtra", "modelUrl");  Debug.Log("start");  if (hasExtra)  {  Debug.Log("has extra");  extras = intent.Call<AndroidJavaObject>("getExtras");  modelUrl = extras.Call<string>("getString", "modelUrl");  Debug.Log("Model URL : " + modelUrl);  }  else  {  Debug.Log("no extra");  }  }  void FixedUpdate()  {  if (Application.platform == RuntimePlatform.Android)  {  if (Input.GetKey(KeyCode.Escape))  {  Application.Quit();  }  }  }  } |

Tabel 5.13 Kode Program FirstPersonController (Unity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122 | namespace Scripts  {  public class FirstPersonController : MonoBehaviour  {  public Camera playerCamera;  public Transform cameraTransform;  public CharacterController characterController;  public float cameraSensitivity;  public float moveSpeed;  public float moveInputDeadZone;  int leftFingerId, rightFingerId;  float halfScreenWidth;  Vector2 lookInput;  float cameraPitch;  Vector2 moveTouchStartPosition;  Vector2 moveInput;  protected Joystick Joystick;  void Start()  {  leftFingerId = -1;  rightFingerId = -1;  halfScreenWidth = Screen.width / 2;  moveInputDeadZone = Mathf.Pow(Screen.height / moveInputDeadZone, 2);  }  void Awake()  {  Joystick = FindObjectOfType<Joystick>();  }  void Update()  {  GetTouchInput();  if (rightFingerId != -1)  {  LookAround();  }  if (leftFingerId != -1)  {  Vector2 movementDirection = Joystick.AxisNormalized \* moveSpeed \* Time.deltaTime;  characterController.Move(transform.right \* Joystick.AxisNormalized.x \* 0.15f + transform.forward \* movementDirection.y);  }  }  void GetTouchInput()  {  for (int i = 0; i < Input.touchCount; i++)  {  Touch t = Input.GetTouch(i);  switch (t.phase)  {  case TouchPhase.Began:  if (t.position.x < halfScreenWidth && leftFingerId == -1)  {  leftFingerId = t.fingerId;  moveTouchStartPosition = t.position;  }  else if (t.position.x > halfScreenWidth && rightFingerId == -1)  {  rightFingerId = t.fingerId;  }  break;  case TouchPhase.Ended:  case TouchPhase.Canceled:  if (t.fingerId == leftFingerId)  {  leftFingerId = -1;  }  else if (t.fingerId == rightFingerId)  {  rightFingerId = -1;  }  break;  case TouchPhase.Moved:  if (t.fingerId == rightFingerId)  {  lookInput = t.deltaPosition \* cameraSensitivity \* Time.deltaTime;  }  else if (t.fingerId == leftFingerId)  {  moveInput = t.position - moveTouchStartPosition;  }  break;  case TouchPhase.Stationary:  if (t.fingerId == rightFingerId)  {  lookInput = Vector2.zero;  }  break;  }  }  }  void LookAround()  {  cameraPitch = Mathf.Clamp(cameraPitch - lookInput.y, -90f, 90f);  playerCamera.transform.localRotation = Quaternion.Euler(cameraPitch, 0, 0);  transform.Rotate(transform.up, lookInput.x);  }  void Move()  {  if (moveInput.sqrMagnitude <= moveInputDeadZone) return;  Vector2 movementDirection = moveInput.normalized \* moveSpeed \* Time.deltaTime;  characterController.Move(transform.right \* movementDirection.x + transform.forward \* movementDirection.y);  }  }  } |

Tabel 5.14 Kode Program API untuk mengunduh model 3 dimensi rumah (Unity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61 | public class API : MonoBehaviour  {  public Text ProgressIndicator;  public Image LoadingBar;    public void GetBundleObject(string assetName, UnityAction<GameObject> callback, Transform bundleParent)  {  StartCoroutine(GetDisplayBundleRoutine(assetName, callback, bundleParent));  }  string GetFilePath(string url)  {  string[] pieces = url.Split('/');  string filename = pieces[pieces.Length - 1];  return Path.Combine(Application.persistentDataPath, filename);  }  IEnumerator GetDisplayBundleRoutine(string bundleURL, UnityAction<GameObject> callback, Transform bundleParent)  {  Debug.Log("Requesting bundle at " + bundleURL);  UnityWebRequest www = UnityWebRequestAssetBundle.GetAssetBundle(bundleURL);  yield return www.SendWebRequest();  StartCoroutine(ShowDownloadProgress(www));  if (www.isNetworkError)  {  Debug.Log("Network error");  }  else  {  AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(www);  if (bundle != null)  {  string rootAssetPath = bundle.GetAllAssetNames()[0];  GameObject arObject = Instantiate(bundle.LoadAsset(rootAssetPath) as GameObject, bundleParent);  bundle.Unload(false);  callback(arObject);  }  else  {  Debug.Log("Not a valid asset bundle");  }  }  }  IEnumerator ShowDownloadProgress(UnityWebRequest req)  {  float downloadDataProgress;  while (!req.isDone)  {  downloadDataProgress = req.downloadProgress \* 100;  Debug.Log("Download: " + downloadDataProgress);  ProgressIndicator.text = ((int)downloadDataProgress).ToString() + "%";  LoadingBar.fillAmount = downloadDataProgress / 100;  yield return null;  }  }  } |

Tabel 5.15 Kode Program Inventory Perabotan (Unity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134 | namespace Scripts  {  public class SC\_InventorySystem : MonoBehaviour  {  public Texture crosshairTexture;  public FirstPersonController playerController;  public SC\_PickItem[] availableItems; //List with Prefabs of all the available items  int[] itemSlots = new int[6];  bool showInventory = false;  SC\_PickItem detectedItem;  int detectedItemIndex;  Vector2 firstTouchPosition;  List<TouchField> touchedFields = new List<TouchField>();  GameObject inventory;  void Start()  {  for (int i = 0; i < itemSlots.Length; i++)  {  itemSlots[i] = -1;  }  for (int i = 0; i < availableItems.Length; i++)  {  itemSlots[i] = i;  }  inventory = GameObject.Find("Inventory");  inventory.SetActive(false);    }  void Update()  {  if (showInventory && touchedFields.Count > 0)  {  TouchField touchField = touchedFields[0];  for (int j = 0; j < availableItems.Length; j++)  {  if (touchField.itemName == availableItems[j].itemName)  {  RaycastHit hit;  Ray ray = playerController.playerCamera.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));  if (Physics.Raycast(ray, out hit, Mathf.Infinity))  {  Transform objectHit = hit.transform;  Vector3 hitPoint = hit.point;  hitPoint += objectHit.up;  Instantiate(availableItems[j], hitPoint, transform.rotation);  // Instantiate(availableItems[j], playerController.playerCamera.transform.position + (playerController.playerCamera.transform.forward), Quaternion.identity);  }  break;  }  }  touchedFields.RemoveAt(0);  }  }  public void addTouchedField(TouchField touchField)  {  touchedFields.Add(touchField);  }  public void removeItemClicked()  {  if (detectedItem && detectedItemIndex > -1)  {  detectedItem.PickItem();  }  }  public void toggleShowInventory()  {  showInventory = !showInventory;  inventory.SetActive(showInventory);  }  void FixedUpdate()  {  RaycastHit hit;  Ray ray = playerController.playerCamera.ViewportPointToRay(new Vector3(0.5F, 0.5F, 0));  if (Physics.Raycast(ray, out hit, 5f))  {  Transform objectHit = hit.transform;  if (objectHit.CompareTag("Respawn"))  {  if ((detectedItem == null || detectedItem.transform != objectHit) && objectHit.GetComponent<SC\_PickItem>() != null)  {  SC\_PickItem itemTmp = objectHit.GetComponent<SC\_PickItem>();  for (int i = 0; i < availableItems.Length; i++)  {  if (availableItems[i].itemName == itemTmp.itemName)  {  detectedItem = itemTmp;  detectedItemIndex = i;  }  }  }  }  else  {  detectedItem = null;  }  }  else  {  detectedItem = null;  }  }  void OnGUI()  {  GUI.color = detectedItem ? Color.green : Color.white;  GUI.DrawTexture(new Rect(Screen.width / 2 - 4, Screen.height / 2 - 4, 8, 8), crosshairTexture);  GUI.color = Color.white;    if (detectedItem)  {  GUI.color = new Color(0, 0, 0, 0.84f);  GUI.Label(new Rect(Screen.width / 2 - 75 + 1, Screen.height / 2 - 50 + 1, 150, 20), detectedItem.objectName);  GUI.color = Color.green;  GUI.Label(new Rect(Screen.width / 2 - 75, Screen.height / 2 - 50, 150, 20), detectedItem.objectName);  }  }  }  } |

Tabel 5.16 Kode Program Main.go (API Service)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | package main  import (  "fmt"  "net/http"  "os"  "simple-api/app"  "simple-api/controllers"  "github.com/gorilla/mux"  )  func main() {  router := mux.NewRouter()  router.HandleFunc("/api/user/new", controllers.CreateAccount).Methods("POST")  router.HandleFunc("/api/user/login", controllers.Authenticate).Methods("POST")  router.HandleFunc("/api/estates/new", controllers.CreateEstate).Methods("POST")  router.HandleFunc("/api/estate/{id}", controllers.GetEstate).Methods("GET")  router.HandleFunc("/api/estates/get", controllers.GetEstates).Methods("GET") // e.g : user/2/contacts  //attach JWT auth middleware  router.Use(app.JwtAuthentication)  port := os.Getenv("PORT")  if port == "" {  port = "8000" //localhost  }  fmt.Println(port)  err := http.ListenAndServe(":"+port, router) //Launch the app, visit localhost:8000/api  if err != nil {  fmt.Print(err)  }  } |

Tabel 5.17 Kode Program EstateController.go (API Service)

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

Tabel 5.18 Kode Program Estate.go (API Service)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  60  61 | package models  import (  "fmt"  u "simple-api/utils"  "github.com/jinzhu/gorm"  )  type Estate struct {  gorm.Model  Title string `json:"title"`  Description string `json:"description"`  ThumbnailUrl string `json:"thumbnailUrl"`  ModelUrl string `json:"modelUrl"`  }  func (estate \*Estate) Validate() (map[string]interface{}, bool) {  if estate.Title == "" {  return u.Message(false, "Estate title should be on the payload"), false  }  if estate.Description == "" {  return u.Message(false, "Description should be on the payload"), false  }  return u.Message(true, "success"), true  }  func (estate \*Estate) Create() map[string]interface{} {  if resp, ok := estate.Validate(); !ok {  return resp  }  GetDB().Create(estate)  resp := u.Message(true, "success")  resp["estate"] = estate  return resp  }  func GetEstate(id uint) \*Estate {  estate := &Estate{}  err := GetDB().Table("estates").Where("id = ?", id).First(estate).Error  if err != nil {  fmt.Println(err)  return nil  }  return estate  }  func GetEstates() []\*Estate {  estates := make([]\*Estate, 0)  err := GetDB().Table("estates").Find(&estates).Error  if err != nil {  fmt.Println(err)  return nil  }  return estates  } |

1. Implementasi Antarmuka

Antarmuka aplikasi berbasis Android TurInteraktif diimplementasikan dengan menggunakan format XML untuk mengatur layout dari masing-masing tampilan. Alur perpindahan halaman digambarkan pada **Error! Reference source not found.**. Terdapat 2 halaman hasil implementasi, yaitu halaman home, dan halaman Tur Interaktif yang digambarkan pada **Error! Reference source not found.**, **Error! Reference source not found.**, dan **Error! Reference source not found.** berturut-turut.

# PENGUJIAN

1. Pengujian *Black-Box*

Pengujian *Black-box* dilakukan untuk memastikan bahwa fitur yang dikembangkan sudah memenuhi kebutuhan fungsional yang sudah ditentukan. Pengujian dilakukan dengan mengikuti alur yang ada pada *use case scenario*. Hasil pengujian diringkas pada Tabel 6.1 hingga Tabel 6.3.

Tabel 6.1 Hasil Pengujian Black-box Menampilkan Daftar Rumah

|  |  |  |
| --- | --- | --- |
| Kode Kebutuhan | | SRS-TIVR-F-1 |
| Kasus uji 1 | Nama Kasus Uji | Menampilkan daftar rumah saat terhubung ke internet |
| Prosedur | 1. Pengguna memiliki koneksi ke jaringan internet. 2. Pengguna berada pada halaman *home*. |
| Hasil yang diharapkan | Sistem menampilkan data daftar rumah yang di dapat dari respon *request* *API Service*. |
| Hasil yang didapat | Sistem menampilkan data daftar rumah yang di dapat dari respon *request* *API Service*. |
| Status | Valid |
| Kasus uji 2 | Nama kasus uji | Menampilkan daftar rumah saat tidak terhubung ke internet |
| Prosedur | 1. Pengguna tidak memiliki koneksi ke jaringan internet. 2. Pengguna berada pada halaman *home*. |
| Hasil yang diharapkan | Sistem tidak menampilkan data daftar rumah yang di dapat dari respon *request* *API Service* dan memberikan *toast* pesan error tidak ada koneksi internet. |
| Hasil yang didapat | Sistem tidak menampilkan data daftar rumah yang di dapat dari respon *request* *API Service* dan memberikan *toast* pesan error tidak ada koneksi internet. |
| Status | Valid |

Tabel 6.2 Hasil Pengujian Menampilkan Tur Interaktif

|  |  |  |
| --- | --- | --- |
| Kode Kebutuhan | | SRS-TIVR-F-2 |
| Kasus uji 1 | Nama Kasus Uji | Menampilkan Tur Interaktif dengan *url* model berisi alamat yang valid |
| Prosedur | 1. Pengguna berada pada halaman home. 2. Pengguna melakukan *tap* pada salah satu *item* data rumah. 3. Sistem memanggil method start*Activity* untuk menjalankan program Unity dengan mengirimkan *string* yang berisi *url* dari model 3 dimensi rumah. |
| Hasil yang diharapkan | Sistem menampilkan tur interaktif dengan menjalankan *activity* Unity. |
| Hasil yang didapat | Sistem menampilkan tur interaktif dengan menjalankan *activity* Unity. |
| Status | Valid |
| Kasus uji 2 | Nama kasus uji | Menampilkan Tur Interaktif dengan *url* model berisi alamat yang tidak valid |
| Prosedur | 1. Pengguna berada pada halaman home. 2. Pengguna melakukan *tap* pada salah satu *item* data rumah. 3. Sistem memanggil method start*Activity* untuk menjalankan program Unity dengan mengirimkan *string* yang berisi nilai kosong atau nilai yang tidak valid. |
| Hasil yang diharapkan | Sistem tidak menampilkan tur interaktif. |
| Hasil yang didapat | Sistem tidak menampilkan tur interaktif. |
| Status | Valid |

Tabel 6.3 Hasil Pengujian Black-box Menambah Daftar Rumah

|  |  |  |
| --- | --- | --- |
| Kode Kebutuhan | | SRS-TIVR-F-3 |
| Kasus uji 1 | Nama Kasus Uji | Menambah data rumah saat terhubung ke internet |
| Prosedur | 1. Pengguna memiliki koneksi ke jaringan internet. 2. Pengguna berada pada halaman *home* dari aplikasi Admin. 3. Pengguna mengisi form data 4. Pengguna melakukan submit untuk mengirim data ke *API Service.* |
| Hasil yang diharapkan | Sistem berhasil mengirimkan *request* untuk menambah data rumah ke *API Service.* |
| Hasil yang didapat | Sistem berhasil mengirimkan *request* untuk menambah data rumah ke *API Service.* |
| Status | Valid |
| Kasus uji 2 | Nama Kasus Uji | Menambah data rumah saat tidak terhubung ke internet |
| Prosedur | 1. Pengguna tidak memiliki koneksi ke jaringan internet. 2. Pengguna berada pada halaman *home* dari aplikasi Admin. 3. Pengguna mengisi form data 4. Pengguna melakukan submit untuk mengirim data ke *API Service.* |
| Hasil yang diharapkan | Sistem menampilkan pesan error tidak ada internet melalui *toast android.* |
| Hasil yang didapat | Sistem menampilkan pesan error tidak ada internet melalui *toast android.* |
| Status | Valid |

1. Analisis Hasil Pengujian Black-box

Pengujian *Black-box* dilakukan untuk menguji apakah sistem yang dibuat sudah memenuhi kebutuhan yang didefinisikan. Pengujian dilakukan dengan menentukan kasus uji dan hasil yang diharapkan, kemudian aplikasi diuji sesuai dengan kasus yang ditentukan. Hasil pengujian dari 6 kasus uji bernilai valid.

1. Pengujian Usability

Pengujian *usability* dilakukan untuk menguji kebergunaan aplikasi bagi pengguna. Pengujian dilakukan dengan cara memberikan *task scenario* kepada pengguna, kemudian pengguna akan melakukan aktivitas untuk menjalankan *task scenario* yang diberikan. *Task scenario* yang diberikan kepada pengguna merupakan langkah-langkah dalam menjalankan aplikasi. Langkah-langkah tersebut mengacu pada fungsionalitas aplikasi.

Pada pengujian ini, *task scenario* diberikan kepada dua jenis pengguna, yaitu pengguna dan *admin*. *Task scenario* untuk pengguna dapat dilihat pada Tabel 6.4 dan *task scenario* untuk admin dapat dilihat pada Tabel 6.5.

Tabel 6.4 Task Scenario Untuk Pengguna

|  |  |  |
| --- | --- | --- |
| *Task* | Nama *Task* | Langkah-langkah |
| 1 | Menampilkan daftar rumah | 1. Berada di halaman utama 2. Sistem menampilkan daftar rumah |
| 2 | Menampilkan tur interaktif | Pengguna menunggu proses menampilkan daftar rumah selesai  Pengguna melakukan *tap* pada salah satu *item* data rumah  Sistem menampilkan tur interaktif dengan membuka *activity* yang dikembangkan menggunakan Unity 3D Engine. |

Tabel 6.5 Task Scenario untuk Admin

|  |  |  |
| --- | --- | --- |
| *Task* | Nama *Task* | Langkah-langkah |
| 1 | Menampilkan daftar rumah | 1. Berada di halaman utama 2. Admin mengisi form 3. Admin menekan tombol *submit* |

Pada Tabel 6.4 merupakan 2 *task scenario* yang diberikan kapada pengguna aplikasi Tur Interaktif, sedangkan pada Tabel 6.5 merupakan 1 *task scenario* yang diberikan kepada admin aplikasi Tur Interaktif. Dalam melaksanakan *task scenario* yang diberikan, masing-masing pengguna harus menjalankan *task scenario* tersebut secara berurutan.

Kemudian responden diminta untuk mengisi kuesioner sebagai penilaian tentang usabilitas pada aplikasi ini. Pengujian pada penelitian ini menggunakan kuesioner SUPER-Qm yang terdiri dari 10 pertanyaan untuk masing-masing jenis penggna yang dapat dilihat pada Tabel 6.7 untuk pengguna aplikasi interaktif dan pada Tabel 6.8 untuk admin aplikas Tur Interaktif. Setiap pertanyaan yang diberikan memiliki skor dengan menggunakan skala likert dalam rentang 1 sampai 5 yang dijelaskan pada Tabel 6.6.

Tabel 6.6 Skor Skala Likert dari Setiap Pertanyaan

|  |  |
| --- | --- |
| Skor | Keterangan |
| 1 | Sangat Tidak Setuju (STS) |
| 2 | Tidak Setuju (TS) |
| 3 | Netral (N) |
| 4 | Setuju (S) |
| 5 | Sangat Setuju (SS) |

Tabel 6.7 Kuesioner SUPR-Qm untuk Pengguna

|  |  |
| --- | --- |
| No | Pertanyaan |
| 1 | Aplikasi ini penting untuk saya |
| 2 | Aplikasi ini merupakan aplikasi tur interaktif terbaik yang pernah saya gunakan |
| 3 | Saya tidak tahu apakah ada aplikasi tur interaktif yang lebih baik dari aplikasi ini |
| 4 | Saya tidak akan menghapus aplikasi ini dari smartphone saya |
| 5 | Saya akan menyarankan aplikasi ini ke teman saya |
| 6 | Saya suka mengeksplorasi fitur-fitur yang terdapat dalam aplikasi ini |
| 7 | Aplikasi ini memiliki seluruh fitur dan fungsi yang saya inginkan pada aplikasi tur interaktif |
| 8 | Desain dari aplikasi ini memudahkan saya dalam menemukan informasi yang saya inginkan |
| 9 | Menurut saya aplikasi ini menarik |
| 10 | Aplikasi ini sesuai dengan kebutuhan saya |
| 11 | Melakukan navigasi dalam aplikasi ini mudah bagi saya |
| 12 | Aplikasi ini mudah digunakan |

Tabel 6.8 Kuesioner SUPR-Qm untuk Admin

|  |  |
| --- | --- |
| No | Pertanyaan |
| 1 | Aplikasi ini penting untuk saya |
| 2 | Aplikasi ini merupakan aplikasi tur interaktif terbaik yang pernah saya gunakan |
| 3 | Saya tidak tahu apakah ada aplikasi tur interaktif yang lebih baik dari aplikasi ini |
| 4 | Saya tidak akan menghapus aplikasi ini dari smartphone saya |
| 5 | Saya akan menyarankan aplikasi ini ke teman saya |
| 6 | Saya suka mengeksplorasi fitur-fitur yang terdapat dalam aplikasi ini |
| 7 | Aplikasi ini memiliki seluruh fitur dan fungsi yang saya inginkan pada aplikasi tur interaktif |
| 8 | Desain dari aplikasi ini memudahkan saya dalam menemukan informasi yang saya inginkan |
| 9 | Menurut saya aplikasi ini menarik |
| 10 | Aplikasi ini sesuai dengan kebutuhan saya |
| 11 | Melakukan navigasi dalam aplikasi ini mudah bagi saya |
| 12 | Aplikasi ini mudah digunakan |

Pada Tabel 6.7 dan Tabel 6.8 merupakan 12 buah pertanyaan dengan menggunakan kuesioner SUPR-Qm yang nantinya akan didapatkan hasil secara kuantitatif terhadap kebergunaan dan kemudahan aplikasi yang dikembangkan.

Perhitungan SUPR-Q yang dilakukan untuk mendapatkan hasil kuantitatif dari skor pada pertanyaan kuesioner SUPR-Qm dilakukan dengan menggunakan rumus pada Persamaan 6.1.

**(6.1)**

Persamaan SUPR-Q pada Persamaan 6.1 digunakan untuk perhitungan nilai SUPR-Qm. Kemudian nilai hasil perhitungan tersebut akan dikonversikan ke dalam kategori penilaian *usability*.

# DAFTAR PUSTAKA

Sommerville, I. (2016). Software Engineering (10th ed.). Harlow, England: Pearson Education

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

