**BAB 1**

**PENDAHULUAN**

1. **Latar Belakang**

Era globalisasi memberikan dampak pada seluruh aspek kehidupan, salah satunya yaitu aspek sosial budaya. Informasi tentang budaya luar bisa di konsumsi atau diadopsi dengan mudah. Hal ini bisa memberikan dampak negatif jika minat akan budaya lokal bergeser atau bahkan tergantikan. Menurut Malinowski, budaya yang lebih tinggi dan aktif akan mempengaruhi budaya yang lebih rendah dan pasif melalui kontak budaya (Mulyana, 2005). Dampak globalisasi terhadap budaya lokal memperlihatkan berbagai permasalahan, salah satunya yaitu berkurangnya rasa cinta atau minat terhadap budaya lokal, erosi nilai kebudayaan lokal dan terjadinya akulturasi (Dwijendra & Mahardika, 2018). Karena itu dibutuhkan upaya untuk mengatasi permasalahan-permasalahan tersebut. Salah satu upaya yang bisa dilakukan yaitu edukasi budaya lokal.

Pada tahun 2018, lebih dari setengah populasi di Indonesia atau 56,2% telah menggunakan ponsel pintar atau gawai (Pusparisa, 2020). Hal ini dapat dimanfaatkan dengan membuat sebuah aplikasi berbasis ponsel atau gawai dengan tema edukasi budaya lokal. Gawai yang menguasai pasar Indonesia menggunakan sistem operasi *android* (Yudhistira, 2021), sistem operasi tersebut dapat digunakan agar aplikasi dapat lebih tersebar luas kepada pengguna.

Teknologi yang sedang perkembangannya yaitu teknologi *augmented reality* (AR). Teknologi AR dapat menggabungkan dunia maya dengan realita. Tanpa menggantikan dunia sesungguhnya, AR menambahkan objek *virtual* ke dalam linkungan realita dengan interaksi yang dilakukan oleh pengguna. Kemampuan ini dapat dimanfaatkan sebagai media edukasi yang interaktif. Pendekatan ini memungkinkan untuk meningkatkan efektivitas dan daya tarik pembelajaran karena tidak menggunakan pengalaman yang statis (Kesim & Ozarslan, 2012).

Berdasarkan latar belakang diatas, akan dibuat sebuah aplikasi edukasi tentang pengenalan budaya Indonesia menggunakan teknologi *augmented reality* berbasis *android*.

1. **Rumusan Masalah**

Bagaimana merancang sebuah aplikasi edukasi tentang budaya Indonesia dengan teknologi *augmented reality* berbasis *android*.

1. **Tujuan dan Manfaat Penelitian**

Tujuan dari penelitian ini adalah untuk menghasilkan sebuah aplikasi edukasi pengenalan budaya Indonesia menggunakan teknologi *augmented reality* berbasis *android.*

Manfaat dari penelitian ini adalah membantu pengguna dalam menambah wawasan tentang budaya Indonesia.

1. **Batasan Masalah**

Agar pengerjaan tugas akhir ini menjadi lebih terarah dan spesifik, maka aplikasi yang dirancang dibatasi pada ruang lingkup pembahasan sebagai berikut:

1. Aplikasi hanya menyajikan informasi mengenai budaya Indonesia hanya berdasarkan 3 provinsi saja, yaitu provinsi Jawa Barat, DKI Jakarta dan Papua.
2. Konten dari aplikasi berupa infromasi tentang budaya yang disajikan dalam bentuk video dan objek 3D menggunakan teknologi *augmented Reality*.
3. Aplikasi yang dikembangkan memiliki spesifikasi minimum sistem operasi *android* versi 7.0 (API Level 24).
4. Teknologi *augmented reality* yang digunakan adalah ARCore.
5. Target pengguna aplikasi untuk pengguna umum yang berbahasa Indonesia.
6. **Metodologi**
7. **Metode Pengembangan Perangkat Lunak**

Pengembangan sistem dilakukan menggunakan *system development life cycle* (SDLC) dengan metode *waterfall* yang memiliki fase mulai dari pengumpulan kebutuhan, desain, implementasi, pengujian dan perawatan. Perpindahan fase dari sebelumnya ke tahap selanjutnya hanya dapat dilakukan jika fase sebelumnya sudah selesai (Kasser, 2002; Balaji & Murugaiyan, 2012; Mahalakshmi & Sundararajan, 2013).

1. **Metode Pengumpulan Data**

Pengumpulan data pada penelitian ini ditujukan untuk keperluan pengembangan sistem. Metode pengumpulan data yang digunakan yaitu:

1. Studi literatur

Studi literatur dilakukan untuk mendapatkan informasi-informasi terkait pengembangan sistem yang didapat dari buku dan jurnal terkait.

1. Observasi

Observasi dibutuhkan untuk mengamati kinerja dari teknologi *augmented reality* yang digunakan pada sistem.

1. ***State of The Art***

Tabel 1.1 *State of The Ar*t

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Peneliti** | **Judul Penelitian** | **Hasil Penelitian** | **Perbandingan** |
| 1. | Marianus Magno Seran Bria, Gede Putra Kusuma, Lous Khrisna Putera Suryapranata. (2018). | Promoting Timor Leste’s Tais Cloth Using Mobile Augmented Reality Application. | Menghasilkan sebuah aplikasi pengenalan pakaian Tais, menggunakan teknologi AR yang dibangun menggunakan aplikasi Unity3D dan Vuforia SDK. Metode AR yang digunakan yaitu *marker based tracking*. | * Informasi budaya yang diperkenalkan berupa video dan objek 3D. * Budaya yang diperkenalkan dari provinsi DKI Jakarta, Jawa Barat dan Papua. * Aplikasi yang dibangun menggunakan Android Studio dan ARCore. |
| 2. | Irsan Pueng, Virginia Tulenan, Xaverius B. N. Najoan. (2020). | Penerapan Teknologi Augmented Reality Untuk Pengenalan Rumah Adat Bolaang Mongondow. | Menghasilkan sebuah aplikasi pengenalan rumah adat Bolaang Mongondow dengan menerapkan teknologi AR. Metodologi penelitian yang digunakan adalah *multimedia development life cycle* (MDLC). | * Informasi budaya yang diperkenalkan meliputi pakaian, rumah adat, alat musik dan pahlawan. * Metodologi yang digunakan dalam pengembangan aplikasi adalah *waterfall*. |
| 3. | H. Lesmana, A. Anas, Suhardi. (2021). | Rancang Bangun Media Edukasi Pengenalan Hewan Menggunakan Teknologi Augmented Reality Berbasis Android. | Menghasilkan sebuah aplikasi pengenalan hewan dengan teknologi AR. Metode penelitian yang digunakan yaitu *research and development* (R&D). | * Metodologi yang digunakan adalah *waterfall*. * Informasi yang disajikan berupa budaya-budaya yang ada di provinsi Indonesia. |
| 4. | Inmaculada Remolar, Cristina Rebollo, Jon A. Fernádez-Moyano. (2021). | Learning History Using Virtual and Augmented Reality. | Menghasilkan sebuah aplikasi bertemakan permainan edukasi tentang sejarah, menggunakan teknologi AR dan VR. | * Tema aplikasi adalah edukasi yang disajikan dalam bentuk video dan objek 3D. * Hanya menggunakan teknologi *augmented reality* sebagai alat dalam menyajikan informasi. |
| 5. | [Melani Indriasari](http://repository.iti.ac.id/jspui/browse?type=author&value=Melani+Indriasari) [Rachman, Ainun Nagiah](http://repository.iti.ac.id/jspui/browse?type=author&value=Rachman,+Ainun+Nagiah). (2021). | Mobile Augmented Reality Pengenalan Sejarah Uang Di Mata Uang Dengan Metode Marker Based Augmented | Menghasilkan sebuah aplikasi pengenalan sejarah uang dengan teknologi AR yang diimplementasikan menggunakan metode *marker based augmented*. | * Informasi yang diperkenalkan berupa budaya Indonesia. * Aplikasi yang dibangun menggunakan Android Studio dan ARCore. |

1. **Sistematika Penulisan**

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bab yang tersusun sebagai berikut:

**BAB 1 PENDAHULUAN**

Bab ini membahas mengenai latar belakang penulisan penelitian, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, *state of the art* dan sistematika penulisan.

**BAB 2 LANDASAN TEORI**

Bab ini mejelaskan pengertian teori dasar yang digunakan dalam penelitian tugas akhir ini, diantaranya teori tentang budaya Indonesia, *augmented reality* dan metode pengenmbangan perangkat lunak.

**BAB 3 ANALISIS DAN PERANCANGAN**

Bab ini membahas mengenai analisis dan perancangan dari permasalahan yang ada dalam proses membangun aplikasi. Diantaranya penjelasan mengenai proses bisnis aplikasi, konsep aplikasi, desain antarmuka, cara kerja aplikasi dan diagram dari hasil analisa sistem.

**BAB 4 IMPLEMENTASI DAN PENGUJIAN**

Bab ini membahas tentang implementasi dari proses analisa dan perancangan aplikasi ke dalam bentuk kode program, yang kemudian membahas juga mengenai pengujian aplikasi agar kode program yang telah diimplementasikan sesuai dengan konsep yang telah dibuat.

**BAB 5 PENUTUP**

Bab ini berisi tentang kesimpulan dan saran dari pembahasan implementasi aplikasi.

**BAB 2**

**LANDASAN TEORI**

1. **Rancang Bangun Sistem**

Rancang atau perancangan merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk dideskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Buchari dkk, 2015).

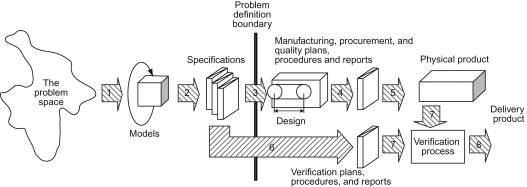
Bangun atau pembangunan adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada, baik secara keseluruhan maupun sebagian (Aslah dkk, 2017).

Sistem berasal dari bahasa Greek yang memiliki arti untuk menempatkan bersama-sama. Dalam pembahasan rekayasa perangkat lunak sistem memiliki arti sendiri yaitu satu paket elemen yang yang saling terintegrasi yang dapat dioperasikan, masing-masing dengan kemampuan yang ditentukan dan dibatasi secara eksplisit, bekerja secara sinergis untuk melakukan pemrosesan nilai tambah untuk memungkinkan Pengguna memenuhi kebutuhan operasionl dalam lingkungan operasi dengan hasil dan keberhasilan yang ditentukan (Wasson, 2006).

Sehingga dapat disimpulkan bahwa rancang bangun sistem merupakan kegiatan untuk menciptakan suatu sistem dengan mengikuti aturan atau prosedur yang sudah dianalisa sebelumnya.

Proses rancang bangun sistem secara umum dapat dilihat pada Gambar 2.1, dengan deskripsi tahapan sebagai berikut (Grady, 2014).

1. Mengubah ruang linkup permasalahan menjadi kumpulan model yang dapat diolah sesuai dengan hasil produk.
2. Model permasalahan kemudian diubah menjadi sekumpulan spesifikasi produk.
3. Spesifikasi produk kemudian diubah menjadi solusi desain.
4. Mengimplementasikan pengetahuan solusi desain menjadi sumber pengadaan, perencanaan manufaktur, dan perencanaan jaminan kualitas.
5. Mengimplementasikan rangkaian rencana, prosedur, dan bahan menjadi sebuah produk.
6. Mengubah hasil spesifikasi menjadi suatu prosedur untuk dijadikan sebagai panduan verifikasi produk.
7. Melakukan prosedur pengujian produk, proses ini menentukan hasil tes produk memenuhi persyaratan atau tidak yang telah didefinisikan sebelumnya pada tahap spesifikasi.
8. Melakukan penyebaran produk kepada pengguna/pelanggan.



Gambar 2.1 Transformasi rancang bangun sistem

1. **Aplikasi**

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan  
yang dibuat untuk melaksanankan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instructiom*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* (Jogiyanto, 2005).

1. **Edukasi**

Edukasi atau disebut juga dengan pendidikan merupakan segala upaya yang direncanakan untuk mempengaruhi orang lain baik individu, kelompok, atau masyarakat sehingga mereka melakukan apa yang diharapkan oleh pelaku pendidikan (Notoatmodjo, 2003).

Edukasi diartikan juga sebagai pendidikan yang diperoleh melalui belajar, dari yang tidak tahu menjadi tahu, dari yang tidak tahu mengatasinya sampai tahu solusinya (Fitriani, 2011).

1. **Budaya Indonesia**

Budaya atau kebudayaan berasal dari bahasa Sanskerta yaitu buddhayah, yang merupakan bentuk jamak dari buddi (budia atau akal) diartikan sebagai hal-hal yang berkaitan dengan budi, dan akal manusia. Bentuk lain dari kata budaya adalah kultur yang berasal dari bahasa inggris yaitu *culture* dan bahasa latin *cultura* (Sutirna, 2021).

Pengertian budaya adalah suatu cara hidup yang berkembang, dan dimiliki bersama oleh sebuah kelompok orang, dan diwariskan dari generasi ke generasi. Budaya terbentuk dari banyak unsur yang rumit, termasuk sistem agama dan politik, adat istiadat, bahasa, perkakas, pakaian, bangunan, dan karya seni (Tubbs dkk, 2000).

Budaya Indonesia adalah seluruh kebudayaan nasional, kebudayaan lokal, maupun kebudayaan asal asing yang telah ada di Indonesia sebelum [Indonesia](https://id.wikipedia.org/wiki/Indonesia" \o "Indonesia) merdeka pada tahun 1945. Budaya Indonesia dapat juga diartikan bahwa Indonesia memiliki beragam suku bangsa dan budaya yang beragam seperti tarian daerah, pakaian adat, dan rumah adat. Budaya Indonesia tidak hanya mencakup budaya asli [bumiputera](https://id.wikipedia.org/wiki/Pribumi-Nusantara" \o "Pribumi-Nusantara), tetapi juga mencakup budaya-budaya [pribumi](https://id.wikipedia.org/wiki/Pribumi-Nusantara" \o "Pribumi-Nusantara) yang mendapat pengaruh budaya Tionghoa, Arab, India, dan Eropa (Putri, 2020).

1. ***Augmented Reality***

*Augmented Reality* (AR) atau dalam Bahasa Indonesia diterjemahkan menjadi realitas tambahan adalah sebuah teknik yang menggabungkan benda maya dua dimensi maupun tiga dimensi ke dalam sebuah lingkup nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata (Pamoedji dkk, 2017). Metode *augemnted reality* secara umum dibagi menjadi 2 yaitu *marker* dan *markerless based tracking*.

1. *Marker Based Tracking*

*Marker based tracking* yaitu metode AR yang bekerja dengan cara membutuhkan penanda (*marker*) sebagai pemicu munculnya objek *virtual* ke dalam dunia nyata. Contoh dari metode AR diantaranya yaitu QR code dan *augmented faces*.

1. *Markerless Based Tracking*

*Markerless based tracking* yaitu metode AR yang tidak membutuhkan tanda sebagai pemicu munculnya objek *virtual*. Metode ini bekerja dengan mendeteksi kondisi dunia nyata sehingga objek *virtual* bisa langsung dimunculkan dari kondisi tersebut. Contoh aplikasi dari metode ini yaitu *Pokemon GO*, aplikasi mebel dan sebagainya.

Dalam membangun produk dengan teknologi AR sangat membutuhkan waktu dan tenaga jika kode program yang dibuat dari proses awal. Untuk mengatasi hal tersebut maka banyak kode program bersifat *open sources* yang siap digunakan untuk menjalankan fitur AR. Kode program yang bersifat *open sources* dan siap digunakan untuk khalayak umum biasanya disebut pustaka. Salah satu pustaka yang digunakan untuk menjalankan pengalaman/fitur AR yaitu ARCore.

1. **ARCore**

ARCore adalah *platform* Google yang digunakan untuk membuat pengalaman *augmented reality*. Dengan menggunakan API yang berbeda, ARCore memungkinkan ponsel Anda untuk memahami lingkungan sekitar, memahami dunia, dan berinteraksi dengan informasi. Beberapa API tersedia di Android dan iOS untuk memungkinkan pengalaman AR di berbagai sistem operasi.

ARCore menggunakan tiga kemampuan utama untuk mengintegrasikan konten *virtual* dengan dunia nyata seperti yang terlihat melalui kamera ponsel Anda:

1. [Pelacakan gerakan](https://developers.google.com/ar/discover/concepts" \l "motion_tracking) memungkinkan ponsel memahami dan melacak posisinya relatif terhadap dunia.
2. [Pemahaman lingkungan](https://developers.google.com/ar/discover/concepts" \l "environmental_understanding) memungkinkan ponsel mendeteksi ukuran dan lokasi semua jenis permukaan: permukaan horizontal, vertikal, dan miring seperti tanah, meja kopi, atau dinding.
3. [Estimasi cahaya](https://developers.google.com/ar/discover/concepts" \l "light_estimation) memungkinkan ponsel memperkirakan kondisi pencahayaan lingkungan saat ini.
4. **Cara Kerja ARCore**

Pada dasarnya, ARCore melakukan dua hal: melacak posisi perangkat seluler saat bergerak, dan membangun pemahamannya sendiri tentang dunia nyata.

Teknologi pelacakan gerakan ARCore menggunakan kamera ponsel untuk mengidentifikasi titik-titik utama (*point of interest*), yang disebut fitur, dan melacak bagaimana titik tersebut bergerak dari waktu ke waktu. Dengan kombinasi pergerakan titik dan pembacaan ini dari sensor inersia ponsel, ARCore menentukan posisi dan orientasi ponsel saat bergerak melalui ruang angkasa.

Selain mengidentifikasi titik-titik utama, ARCore dapat mendeteksi permukaan datar, seperti tabel atau lantai, dan juga dapat memperkirakan pencahayaan rata-rata di area di sekitarnya. Kemampuan ini digabungkan untuk memungkinkan ARCore membangun pemahamannya sendiri tentang lingkungan di sekitarnya.

Pemahaman ARCore tentang dunia nyata memungkinkan untuk menempatkan objek, anotasi, atau informasi lainnya dengan terintegrasi secara mulus dengan dunia nyata. ARCore dapat menempatkan anak kucing tidur di sudut meja kopi, atau menganotasi lukisan dengan informasi biografi tentang seniman tersebut. Pelacakan gerakan berarti bergerak dan melihat objek dari sudut mana pun, dan bahkan jika objek tersebut tidak disorot oleh kamera ponsel kemudian ponsel menyorot kembali, maka objek atau anotasi akan berada tepat di tempat semula saat kamera ponsel meninggalkannya.

ARCore menyediakan *software development kit* (SDK) untuk berbagai lingkungan pengembangan paling populer. SDK ini menyediakan API *native* untuk semua fitur AR penting seperti pelacakan gerakan, pemahaman lingkungan, dan perkiraan cahaya. Dengan kemampuan ini, memungkinkan untuk membuat aplikasi dengan pengalaman AR yang sepenuhnya baru atau menyempurnakan aplikasi yang ada dengan fitur AR.

1. **Pengembangan ARCore**

ARCore menyediakan beberapa modul pengembangan yang sudah siap digunakan dalam mengintegrasikan kemampuan AR pada suatu produk. Modul pengembangan yang disediakan ARCore yaitu:

1. Gambar *Augmented*

Gambar *augmented* API di ARCore memungkinkan untuk membuat aplikasi AR yang dapat mendeteksi gambar 2 dimensi pada lingkungan nyata, seperti gambar poster atau pengemasan produk.

ARCore menggunakan algoritma *computer vision* untuk mengekstrak fitur dari informasi hitam putih di setiap gambar, dan menyimpan representasi fitur ini di satu atau beberapa *database* Gambar *augmented*.

Saat *runtime*, ARCore menelusuri fitur ini pada permukaan datar di lingkungan nyata. Hal ini memungkinkan ARCore mendeteksi gambar di lingkungan realita dan memperkirakan posisi, orientasi, dan ukurannya jika tidak disediakan.

1. Wajah *Augmented*

Wajah *augmented* API memungkinkan untuk merender aset *vitual* pada wajah manusia tanpa menggunakan *hardware* khusus. Fitur ini menyediakan titik fitur yang memungkinkan aplikasi untuk otomatis mengidentifikasi wilayah berbeda pada wajah yang terdeteksi. Aplikasi kemudian bisa menggunakan *region* tersebut untuk menempatkan aset dengan cara yang benar-benar cocok dengan kontur tiap-tiap wajah.

1. **Diagram Alir Program (*Flowchart*)**

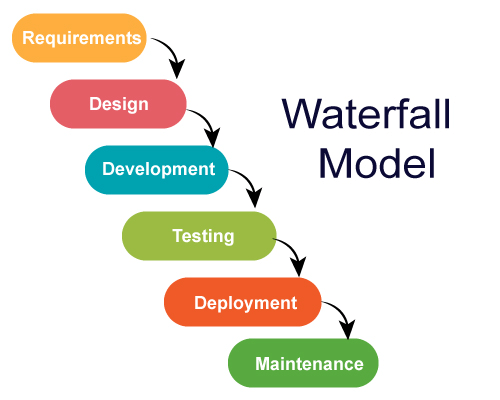
*Flowchart* merupakan suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan berhubungan antara suatu proses (intruksi) dengan proses lainnya dalam suatu program (Wibawanto, 2017).

1. ***System Development Life Cycle* (SDLC)**

*System Development Life Cycle* (SDLC) adalah proses yang menentukan bagaimana suatu sistem informasi dapat mendukung kebutuhan bisnis, merancang sistem, dan membangun. Dalam siklus SDLC, proses membangun dibagi menjadi beberapa langkah dan pada sistem yang besar, masing-masing langkah dikerjakan oleh tim yang berbeda. SDLC tidak hanya penting untuk proses produksi *software*, tapi juga sangat penting untuk proses *maintenance* itu sendiri. Salah satu contoh dari SDLC yaitu metode *waterfall*.

1. **Metode *Waterfall***

Model *waterfall* atau sering kali disebut sebagai *classic life cycle* adalah model pengembangan perangkat lunak yang menekankan fase-fase yang berurutan dan sistematis (Royce, 1970). Fase metode *waterfall* dimulai dari spesifikasi kebutuhan konsumen dan berkembang melalui proses perencanaan (*planning*), pemodelan (*modelling*), pembangunan (*construction*), dan penyebaran (*deployment*), yang berujung pada dukungan terus menerus untuk sebuah perangkat lunak yang utuh. Disebut model *waterfall* karena setiap tahapan harus menunggu tahap sebelumnya selesai dan harus berjalan berurutan (Pressman, 2015). Tahapan model *waterfall* dimulai dari Model ini dapat digunakan pada saat kebutuhan untuk sebuah masalah telah dipahami dengan baik, dan pekerjaan dapat mengalir secara linear dari proses komunikasi hingga penyebaran (*deployment*). Situasi ini ditemui saat adaptasi atau perpanjangan dari sistem yang ada sudah terdefinisi dengan baik. Adapun model ini dapat digunakan pada situasi di mana sumber daya manusia yang dimiliki terbatas tetapi kebutuhan perangkat lunak sudah terdefinisi dengan baik. Namun, dalam pengembangan perangkat lunak, model ini cenderung menjadi salah satu pendekatan yang kurang iteratif dan fleksibel, karena proses mengalir satu arah ke bawah seperti [air terjun](https://id.wikipedia.org/wiki/Air_terjun" \o "Air terjun) (Pressman, 2015). Model *waterfall* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Model *waterfall*

1. Tahap Analisis Kebutuhan

Mengumpulkan dan menganalisa semua data yang dibutuhkan dalam proses pengembangan aplikasi. Proses pengumpulan dan analisa data dilakukan dengan mencari referensi dari buku, artikel atau jurnal.

1. Tahap Desain Sistem

Data yang telah dikumpulkan dan dianalisa diolah sedemikian rupa kemudian menghasilkan rancangan yang bisa diterapkan pada aplikasi. Rancangan yang dihasilkan yaitu desain tampilan, fitur aplikasi dan arsitektur aplikasi.

1. Tahap Implementasi

Pada tahap ini, hasil dari tahap sistem desain diterapkan ke dalam kode program sehingga akan menghasilkan sebuah produk.

1. Tahap Pengujian

Pada tahap ini dilakukan proses pengujian kode program dengan tujuan untuk memastikan kode program sudah sesuai dengan desain rancangan yang telah dibangun.

1. Tahap Penyebaran

Setelah proses pengujian aplikasi selesai, dilakukan proses penyebaran (*deployment*) aplikasi agar aplikasi bisa dikonsumsi secara masal.

1. Tahap Perawatan

Saat aplikasi terpublikasi pada pengguna, dimungkinkan aplikasi terjadi *error* pada perangkat tertentu. Proses memperbaiki *error* tersebut dilakukan pada tahap ini.

1. ***Unified Modeling Language* (UML)**

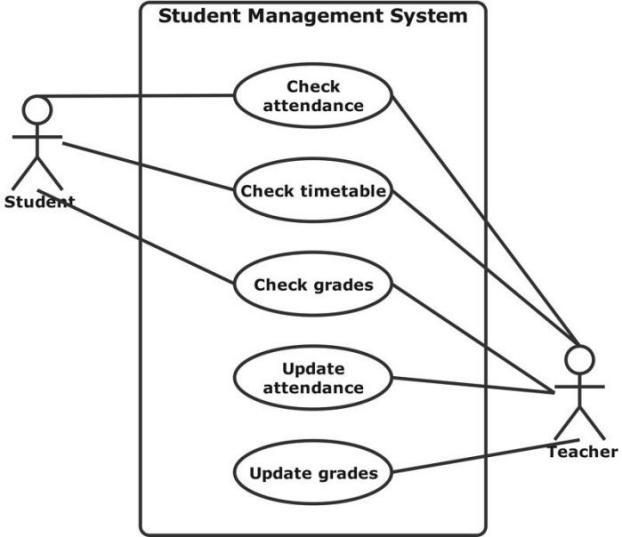
*Unified Modelling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek (Rosa & Shalahuddin, 2018). Pemanfaatan UML dalam pengembangan sebuah produk dapat membantu lebih jelas, komprehensif, ringkas dan dapat dikembangkan. Contoh dari diagram UML diantaranya *use case*, *activity, class*, s*equence* dan *deployment diagram*.

1. ***Use Case Diagram***

*Use case diagram* merupakan diagram yang menggambarkan interaksi antara sistem dengan *actor* (Hamilton & Miles, 2006). *Use case diagram* berfungsi untuk memodelkan proses bisnis yang dilihat dari kebutuhan pengguna. Penggunaan *use case diagram* sebagai pemodelan sistem dapat membantu komunikasi antara *domain expert* dengan *end user*. Komponen-komponen dalam membangun *use case diagram* diantaranya:

1. *Use case* adalah fungsi yang digunakan pada sistem untuk memenuhi kebutuhan pengguna*.* Komponen *uses case* di gambarkan dalam bentuk lonjong.
2. *Actor adalah* pengguna yang berinteraksi dengan sistem. Biasanya *actor* di gambarkan berbentuk gambar *sticky figure* atau berbentuk persegi panjang.
3. *Communication line* adalah komponen yang menghubungkan antara *actor* dengan sistem. Komponen ini digambarkan dalam bentuk garis.
4. *System boundaries* adalah Komponen yang memisahkan komponen yang ada didalam sistem dengan yang diluar sistem. Dengan adanya komponen ini pemisahan fungsi atau tanggung jawab komponen menjadi jelas. *System boundaries* digambarkan dalam bentuk persegi.

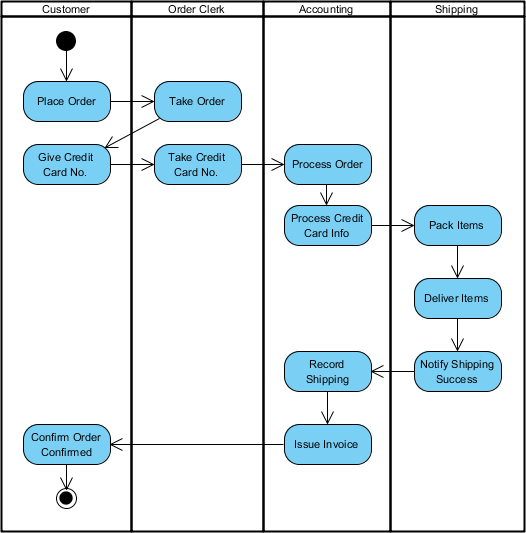
Contoh *use case diagram* dapat dilihat pada Gambar 2.3.



Gambar 2.3. Contoh *use case* *diagram*

1. ***Activity Diagram***

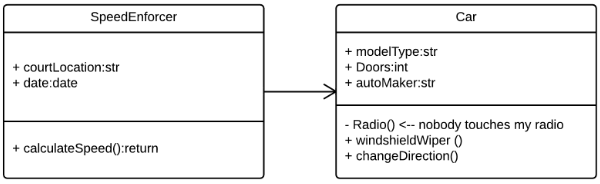
*Activity diagram* merupakan diagram yang menggambarkan proses bisnis pada sistem. Jika *use case diagram* menggambarkan apa yang bisa sistem lakukan, maka *activity diagram* menggambarkan bagaimana sistem mencapai/menjalankan fungsi tersebut (Hamilton & Miles, 2006). Proses bisnis yang ada pada sistem di gambarkan berurutan secara vertikal. Komponen pada *activity diagram* sama dengan komponen yang ada pada *flowchart* (Hamilton & Miles, 2006), seperti komponen *decision* (percabangan) dan proses. Contoh *activity diagram* dapat dilihat pada Gambar 2.4.



Gambar 2.4. Contoh *activity diagram*

1. ***Class Diagram***

*Class diagram* merupakan diagram yang menggambarkan relasi antar kelas yang terjadi pada sistem (Hamilton & Miles, 2006). Kelas itu sendiri merupakan tipe objek yang terdiri atas atribut yang merupakan representasi karakteristik dari suatu *class* dan metode yang menggambarkan perilaku dari suatu *class*. Contoh *class diagram* dapat dilihat pada Gambar 2.5.



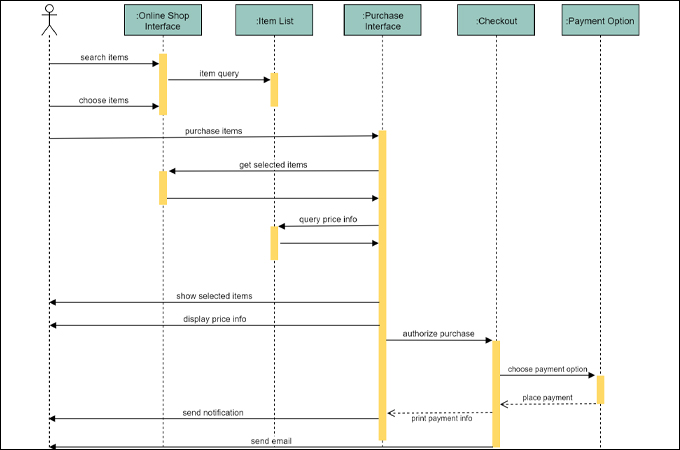
Gambar 2.5. Contoh *class diagram*

1. ***Sequence Diagram***

*Sequence diagram* merupakan diagram yang menggambarkan interaksi antara objek-objek yang ada pada sistem dengan memperhatikan urutan interaksinya (Hamilton & Miles, 2006). *Sequence diagram* juga menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Dengan menggunakan *sequence diagram* dalam pengembangan sistem, bisa digunakan untuk mendeskripsikan interaksi apa yang dipanggil saat fungsi di eksekusi dan urutan eksekusinya. Komponen dalam membangun *sequence diagram* diantaranya:

1. *Actor* merupakan komponen yang menggambarkan seorang pengguna yang berada di luar sistem dan sedang berinteraksi dengan sistem. Dalam sequence diagram, *actor* biasanya digambarkan dengan simbol *stick figure*.
2. Objek merupakan komponen yang digambarkan memiliki bentuk kotak, berisikan nama dari objek dengan garis bawah. Objek berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem.
3. *Lifeline* merupakan komponen yang digambarkan dengan bentuk garis putus-putus. Komponen ini biasanya memiliki kotak yang berisi objek yang memiliki fungsi untuk menggambarkan aktifitas dari objek.
4. *Activation box* merupakan komponen yang merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya. Semakin lama waktu yang diperlukan, maka secara otomatis *activation box* juga akan menjadi lebih panjang. Komponen ini digambarkan dengan bentuk persegi panjang.
5. *Messages* merupakan komponen untuk menggambarkan komunikasi antar objek. Komponen *messages* direpresentasikan dengan anak panah.

Contoh *sequence diagram* dapat dilihat pada Gambar 2.6.

****

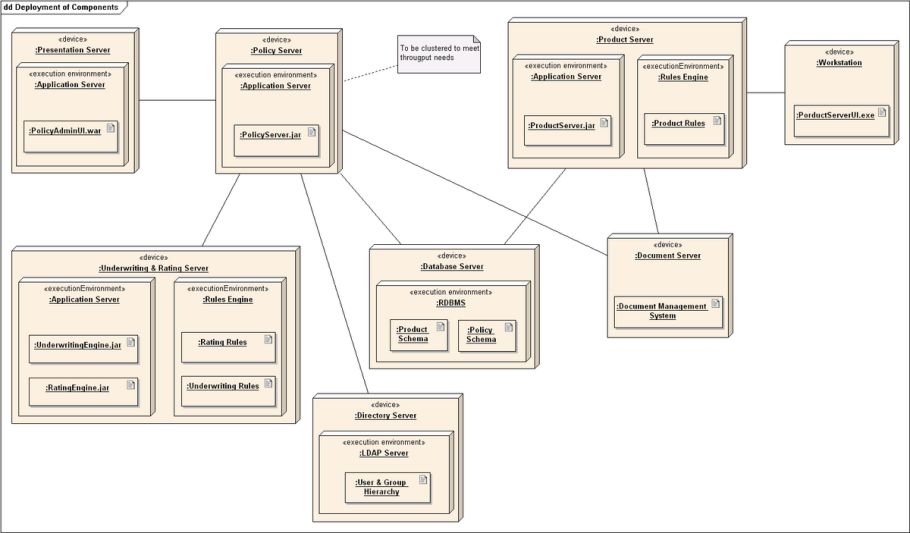
Gambar 2.6. Contoh *sequence diagram*

1. ***Deployment Diagram***

*Deployment diagram* merupakan diagram yang menggambarkan bagaimana perangkat lunak ditugaskan kepada perangkat keras dan bagaimana mereka saling berkomunikasi (Hamilton & Miles, 2006). Komponen yang ada pada *deployment diagram* diantaranya:

1. *Node* merupakan penggambaran perangkat keras yang ada pada sistem. *Node* di gambarkan dalam bentuk persegi 3 dimensi.
2. *Artifact* merupakan berkas yang dapat diekseusi/dijalankan pada perangkat keras. Komponen ini digambarkan berbentuk persegi panjang.

Contoh *deployment diagram* dapat dilihat pada Gambar 2.7.



Gambar 2.7. Contoh *deployment diagram*

1. **Pengujian Perangkat Lunak**

Pengujian perangkat lunak *(software testing*) merupakan tindakan investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (*under test*). Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara objektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya. Teknik-teknik pengujian mencakup, tetapi tidak terbatas pada, proses mengeksekusi suatu bagian program atau keseluruhan aplikasi dengan tujuan untuk menemukan *[bug](https://id.wikipedia.org/wiki/Bug" \o "Bug)* perangkat lunak (kesalahan atau cacat lainnya).

Pengujian perangkat lunak dapat dinyatakan sebagai proses validasi dan verifikasi bahwa sebuah produk:

* Memenuhi kebutuhan (*requirement*) yang mendasari perancangan dan pengembangan perangkat lunak tersebut.
* Berjalan sesuai dengan yang diharapkan.
* Dapat diterapkan menggunakan karakteristik yang sama.
* Memenuhi kebutuhan semua pihak yang berkepentingan.

Pengujian perangkat lunak dibagi ke dalam 2 jenis yaitu *black* dan *white box test*.

1. ***Black Box Testing***

Pengujian yang dilakukan berdasarkan fungisonalitas dari aplikasi seperti tampilan aplikasi, fitur-fitur yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan. Proses pengujian ini tidak melihat dan menguji kode sumber program. Menurut Rizki Soetam *black box testing* adalah tipe pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya (Soetam, 2011).

*Black box testing* cenderung untuk menemukan hal-hal berikut:

* Fungsi yang tidak benar atau tidak ada.
* Kesalahan antarmuka (*interface errors*).
* Kesalahan pada struktur data dan akses basis data.
* Kesalahan performansi (*performance error*s).
* Kesalahan inisialisasi dan terminasi.

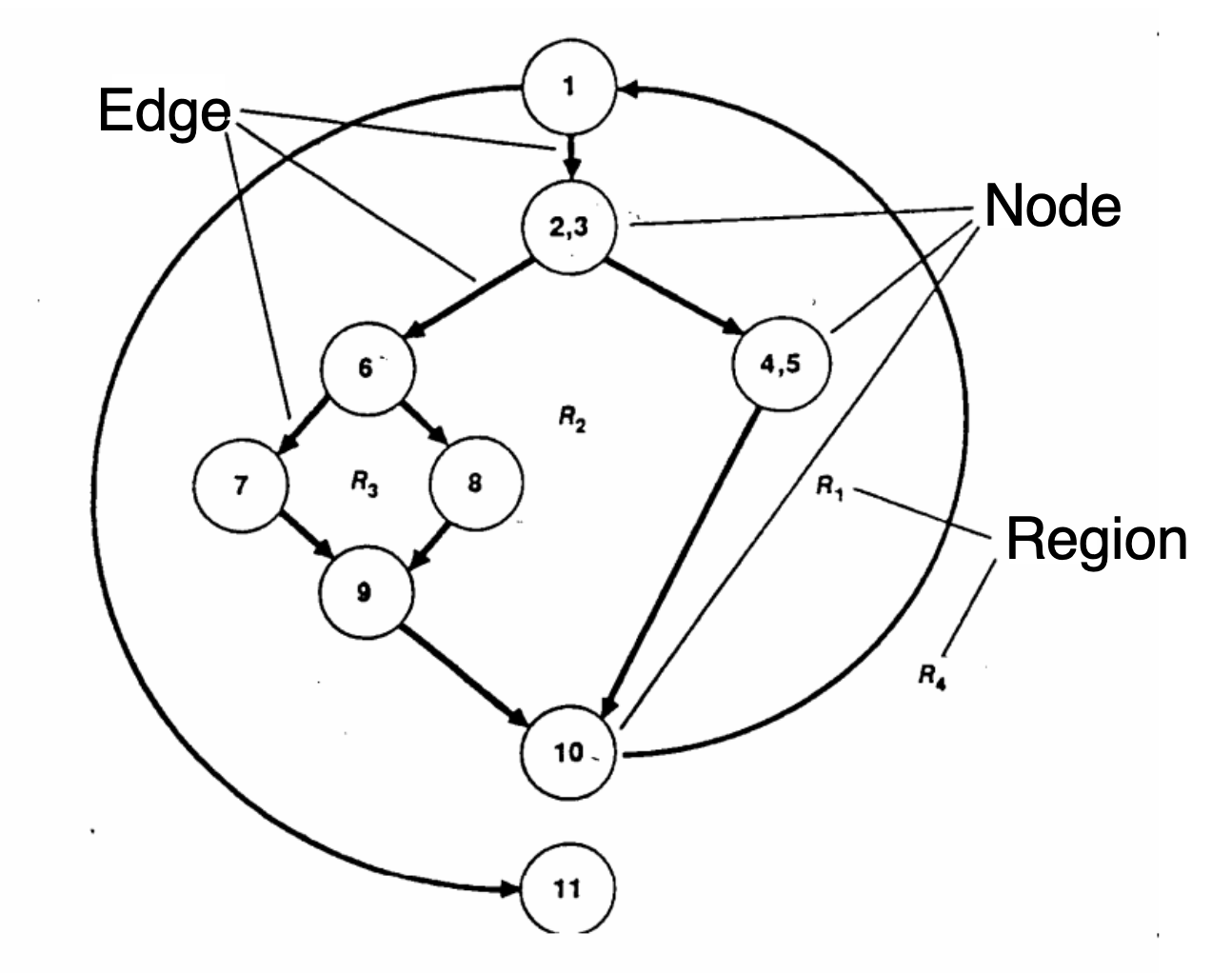
1. ***White Box Testing***

Pengujian yang dilakukan berdasarkan pada detail prosedur dan alur logika kode program. Pada kegiatan *white box testing*, penguji (*tester*) melihat kode sumber program dengan bertujuan untuk menemukan *bugs* dari kode program yang akan diuji. Proses pengujian ini dilakukan sampai kepada detail pengecekan kode sumber program. Salah satu teknik dari *white box testing* yaitu *basis path*.

1. **Teknik *Basis Path Testing***

Teknik *basis path* *testing* merupakan salah satu teknik pengujian *white box* yang pertama kali dikemukakan oleh Tom McCabe. Teknik ini memungkinkan penguji untuk mengukur kompleksitas logika dari rancangan prosedural dan mendefinisikan alur yang dieksekusi. Teknik *basis path* ini menjamin akan mengeksekusi setiap *statement* dari aplikasi yang diujikan setidaknya satu kali pada saat tahap pengujian (Handy & Susilo, 2014).

Teknik *basis path testing* bekerja dengan menggambarkan aliran control logika yang disebut dengan *flow graph*. *Flow graph* terdiri atas *edge*, *node*, dan *region*. Contoh *flow graph* dapat dilihat pada Gambar 2.8.



Gambar 2.8. Contoh *flow graph*

Penentuan jumlah alur *basis path* dapat dilakukan menggunakan perhitungan kompleksitas siklomatis. Kompleksitas siklomatis adalah sebuah pengukuran perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika pada sebuah program (Pressman, 2015). Dengan hasil dari pengukuran ini, dapat menentukan apakah program yang dibuat sederhana atau kompleks berdasarkan logika yang diterapkan pada program. Berikut ini merupakan rumus dari kompleksitas siklomatis.

*V(G) = E - N + 2*

*V(G) =* Kompleksitas Siklomatis

*E* = Jumlah *edges* pada *flowgraph*

*N* = Jumlah *nodes* pada *flowgraph*

*P* = jumlah predicates *nodes* pada *flowgraph*

**BAB 3**

**ANALISIS DAN PERANCANGAN**

1. **Analisis Permasalahan Sistem**

Dampak negatif dari globalisasi pada aspek sosial budaya dapat mengakibatkan tergantikannya budaya lokal dengan budaya lain. Hal tersebut dapat terjadi karena banyak hal, salah satunya karena kurangnya informasi tentang budaya lokal. Untuk mengantisipasi hal tersebut dibutuhkan upaya untuk meningkatkan pengetahuan budaya lokal dengan membuat sebuah aplikasi edukasi tentang pengenalan budaya Indonesia.

Indonesia memiliki banyak sekali keberagaman kebudayaan, oleh karena itu aplikasi yang dibangun hanya akan membahas beberapa kebudayaan berdasarkan provinsi-provinsi yang ada di Indonesia, yaitu budaya dari provinsi DKI Jakarta, Jawa Barat dan Papua. Dan bentuk budaya yang di bahas berupa pakaian adat, rumah adat, alat musik dan pahlawana yang akan dijelaskan dalam bentuk video dan objek 3D. Karena budaya yang dibahas terbatas, maka dibutuhkan sumber/referensi lain yang diharapkan dapat melengkapi informasi terkait pengetahuan budaya Indonesia.

Teknologi *augmented reality* (AR) dimanfaatkan sebagai media pengenalan budaya Indonesia pada sistem. Metode AR yang digunakan adalah metode *marker based tracking* yang membutuhkan sebuah *marker* untuk di deteksi. AR pada aplikasi edukasi pengenalan budaya Indonesia diimplementasikan dengan menggunakan pustaka ARCore.

1. **Analisis Metode *Marker Based Tracking***

Metode *marker based tracking* merupakan salah satu metode teknologi AR yang bekerja dengan memindai sebuah *marker* (penanda) kemudian jika *marker* dikenali atau terdeteksi, maka objek *virtual* akan tampil pada lingkungan nyata. Bagian bab ini akan membahas bagaimana metode *marker based tracking bekerja* pada sistem dengan cara mengajukan pertanyaan dan menjawabnya. Jawaban dari pertanyaan didapat dari studi literatur terkait metode AR. Pada bagian analisa sistem sebelumnya, disebutkan bahwa teknologi AR yang digunakan adalah ARCore, sehingga pembahasan hanya mencakup teknologi tersebut. Pertanyaan-pertanyaan yang muncul terkait metode *marker based tracking* diantaranya:

1. Bagaimana *marker* bisa dikenali oleh sistem?

*Marker* dapat dikenali oleh sistem berkat algoritma *computer vision* yang sudah diimplementasikan oleh ARCore dan terdapat juga *database* untuk menyimpan informasi *marker* tersebut. *Computer vision* bekerja dengan mengekstrak fitur dari informasi hitam putih pada *marker*, kemudian menyimpan representasi fitur pada *database marker*.

Proses *marker* bisa dikenali ketika pengguna memindai *marker* dengan kamera, maka algoritma *computer vision* akan menangkap/mendeteksi *marker* tersebut sekaligus mencoba mengecek, apakah marker ada *di* database atau tidak. Jika marker ada di *database*, maka objek *virtual* akan ditampilkan pada layar kamera, jika tidak ada maka tidak ada proses yang terjadi.

1. Kriteria *marker* seperti apa yang bisa disimpan oleh *database*?

* *Marker* dengan minimal resolusi 300 x 300 pixel.
* *Marker* dengan format PNG atau JPEG.
* Sebaiknya *marker* tidak memiliki pola yang berulang atau banyak pola geometris, karena bisa menyebabkan masalah pada performa deteksi.
* Bisa menggunakan alat ARCore SDK untuk mendapatkan skor kualitas. Skor kualitas memiliki jangkauan nilai dari 0 sampai 100. Minimal skor kualitas yang direkomendasikan adalah 75.

1. Kriteria *marker* seperti apa yang bisa dikenali oleh kamera?

* *Marker* harus setidaknya 25% dari frame kamera.
* *Marker* berada pada bidang yang datar.
* *Marker* terlihat jelas di kamera, seperti *marker* tidak dikaburkan sebagian, dilihat pada sudut yang sangat miring, atau dilihat saat kamera bergerak terlalu cepat.

1. **Analisis Kebutuhan Fungsional**

Kebutuhan fungsional merupakan fungsionalitas atau fitur-fitur apa saja yang akan diterapkan pada pengembangan aplikasi edukasi pengenalan budaya Indonesia, Fitur-fitur tersebut antara lain sebagai berikut:

1. Pengguna dapat melihat video edukasi pengenalan budaya Indonesia pada layar kamera, setelah pengguna memindai sebuah logo provinsi Indonesia menggunakan teknologi AR.
2. Pengguna dapat melihat objek budaya Indonesia dalam bentuk 3D dan terdapat deskripsi atau penjelasan dari objek tersebut.
3. Pengguna dapat menambahkan efek pada wajah dalam bentuk atribut-atribut kebudayaan menggunakan teknologi AR.
4. Pengguna dapat melihat referensi lain tentang budaya.
5. **Analisis Kebutuhan Perangkat**

Tidak semua perangkat android mendukung pustaka ARCore, dibutuhkan sebuah sertifikat pada perangkat yang di tentukan oleh ARCore. Sertifikasi diperlukan supaya pengguna mendapatkan pengalaman yang baik pada aplikasi AR. Hal ini terutama terkait dengan pelacakan gerakan sensitif, yang dilakukan dengan menggabungkan gambar kamera dan input sensor gerakan untuk menentukan cara perangkat pengguna bergerak di dunia nyata (“ARCore *devices*”, 2022).

Sertifikasi ARCore pada perangkat dapat diketahui dengan melihat *website* ARCore atau dengan cek perangkat apakah perangkat sudah atau bisa *install* aplikasi ARCore. Kemudian juga ada batas minimum sistem operasi android yang mendukung pustaka ARCore, yaitu *android* versi 7.0 (Nougat, SDK 24). Sehingga spesifikasi perangkat yang mendukung dalam pembangunan aplikasi edukasi pengenalan budaya Indonesia sebagai berikut:

1. Minimum sistem operasi *android* versi 7.0 (Nougat, SDK 24).
2. Terdapat sertifikasi ARCore.
3. **Perancangan Sistem**

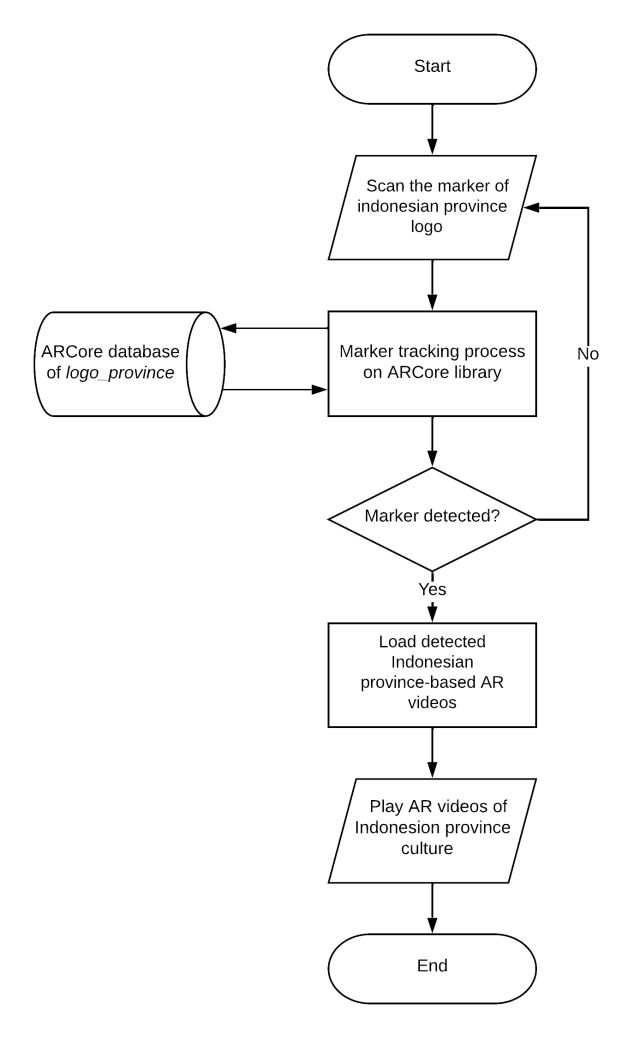
Perancangan sistem pada aplikasi edukasi pengenalan budaya Indonesia berupa perancangan alur kerja metode *marker based tracking* bekerja pada sistem, pemodelan sistem menggunakan UML, perancangan navigasi sistem dan perancangan antarmuka.

1. **Perancangan Alur Kerja Metode *Marker Based Tracking* Pada Sistem**

Alur kerja metode marker based tracking pada sistem aplikasi edukasi pengenalan budaya Indonesia sebagai berikut:

1. Pindai logo provinsi Indonesia dengan kamera ponsel.
2. Pustaka ARCore akan mendeteksi gambar yang disesuaikan dengan yang ada di *database system*.
3. Jika gambar terdeteksi, maka akan memuat video AR budaya sesuai dengan provinsi yang di pindai. Lalu gambar tidak terdeteksi, maka proses scan tetap berlanjut.
4. Video AR budaya provinsi Indonesia berhasil diputar.

Alur kerja metode *marker based tracking* pada sistem dalam bentuk *flowchart* dapat dilihat pada Gambar 3.1.



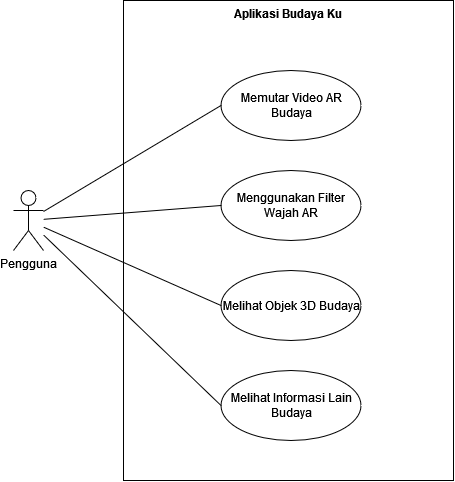
Gambar 3.1. *Flowchart marker based tracking* pada sistem

1. **UML Aplikasi Edukasi Pengenalan Budaya Indonesia**

UML (*Unified Modeling Language*) yang dibuat sebagai pemodelan sistem aplikasi edukasi pengenalan budaya Indonesia yaitu *use case diagram*, a*ctivity diagram*, *sequence diagram*, *class diagram* dan *deployment diagram*.

1. ***Use Case Diagram* Aplikasi Edukasi Pengenalan Budaya Indonesia**

*Use case diagram* yang dapat dilihat pada Gambar 3.2, menunjukkan hanya terdapat 1 *actor* yang ada pada sistem dan 4 fungsionalitas sistem. Yang dapat dilakukan *actor* pada sistem yaitu memutar video budaya, melihat objek 3D budaya, menggunakan efek wajah budaya dan melihat referensi lain tentang budaya.



**Aplikasi Edukasi Pengenalan Budaya Indonesia**

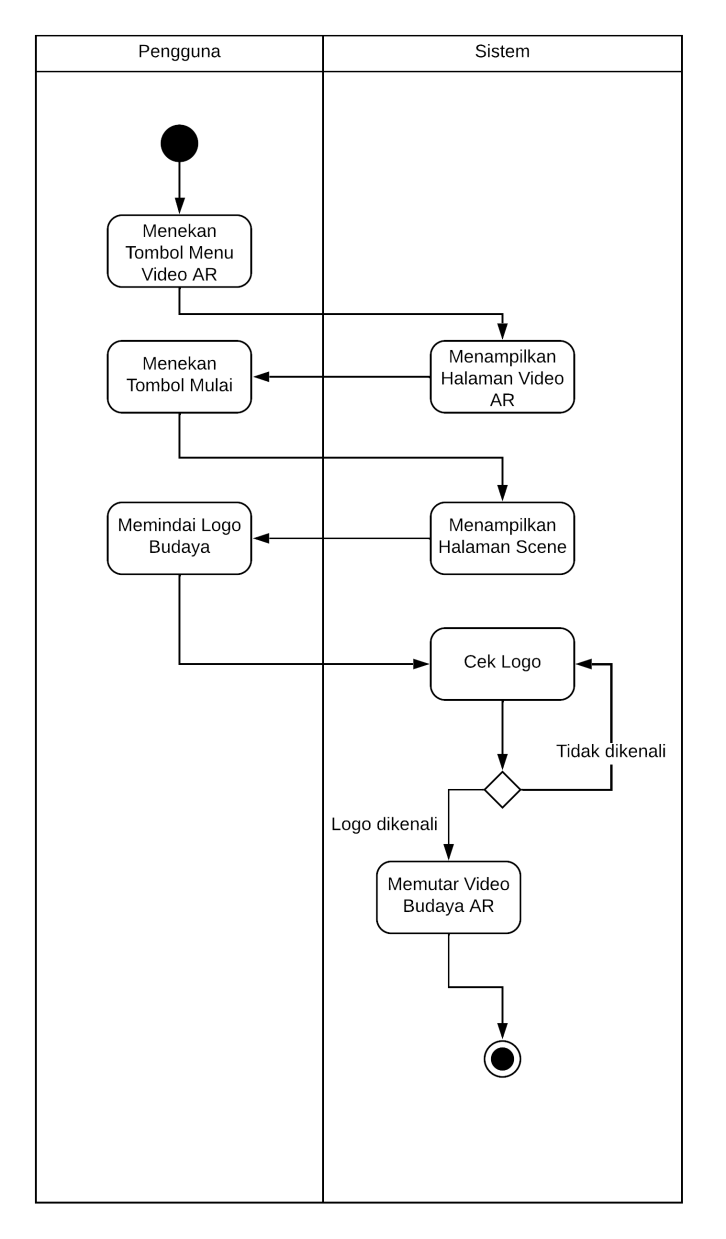
Gambar3.2. *Use case* *diagram* aplikasi edukasi pengenalan budaya Indonesia

1. ***Activity Diagram* Aplikasi Edukasi Pengenalan Budaya Indonesia**

*Activity diagram* yang dibuat mejelaskan aktivitas yang terjadi pada setiap *use case*. Berikut ini merupakan *activity diagram* yang ada pada aplikasi edukasi pengenalan budaya Indonesia.

1. *Activity Diagram* Memutar Video AR Budaya

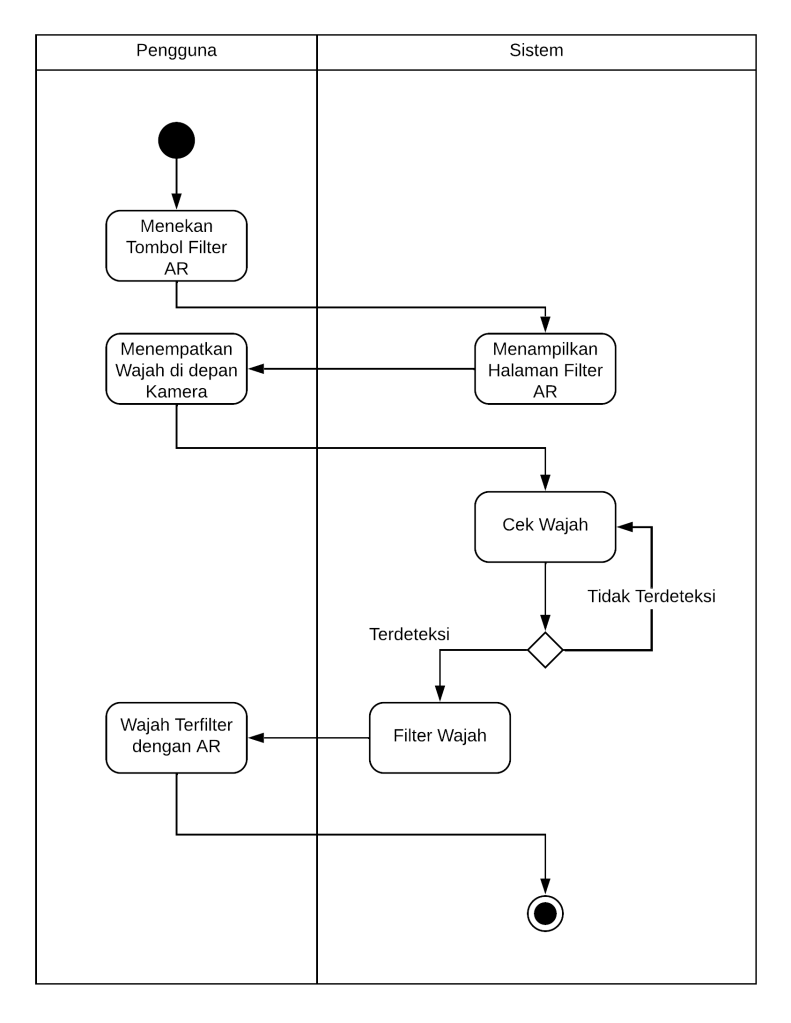
*Acitivity diagram* memutar video AR budaya ditunjukkan pada Gambar 3.3, menunjukkan terdapat proses pemindaian logo yang kemudian dilanjutkan dengan proses pengecekan logo. Jika logo terdeteksi maka sistem akan memutar video AR budaya.



Gambar 3.3. *Activity diagram* memutar video AR budaya

1. *Activity Diagram* Menggunakan *Filter* Wajah AR Budaya

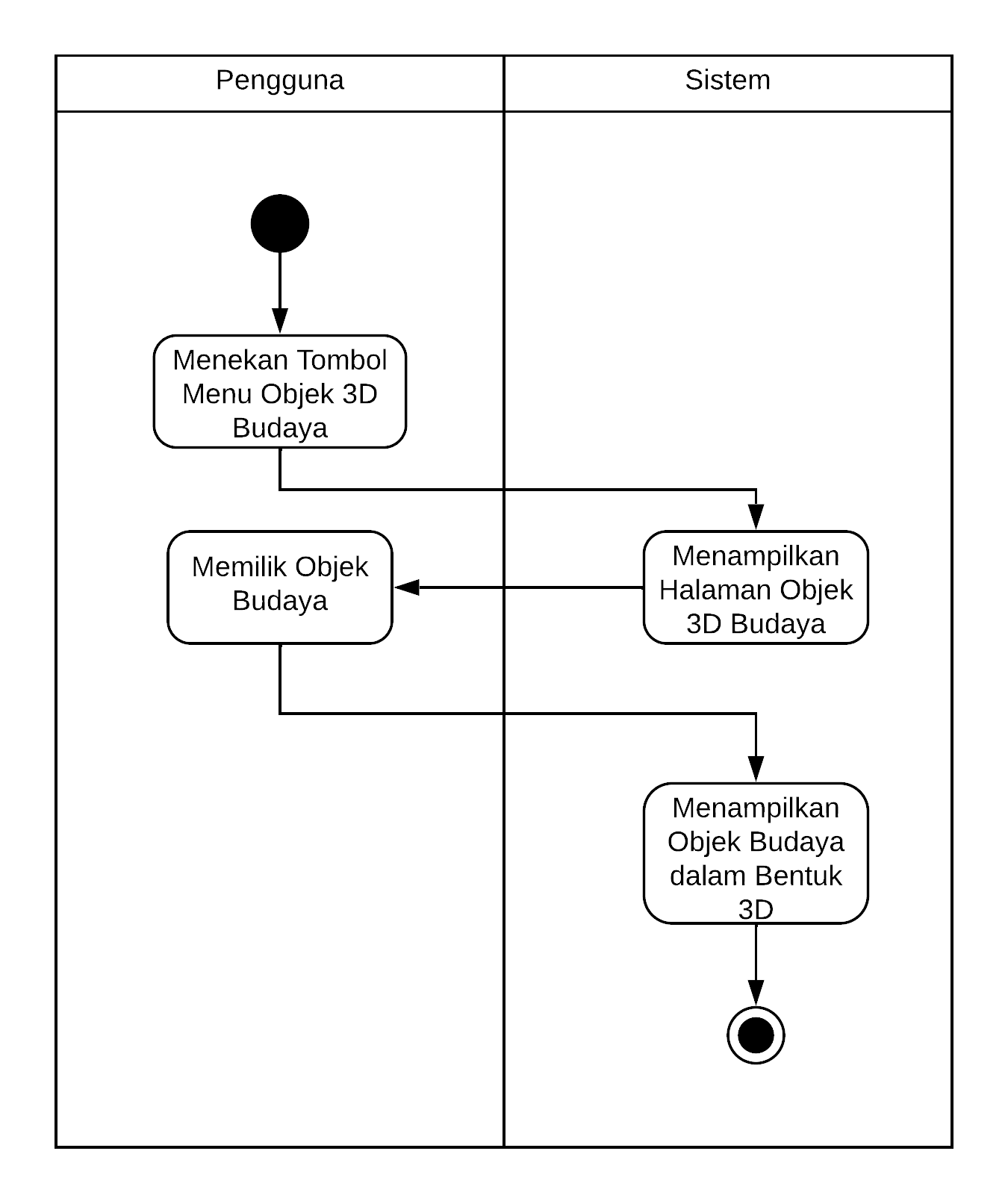
*Activity diagram* menggunakan *filter* wajah AR budaya ditunjukkan pada Gambar 3.4, gambar tersebut menujukkan bahwa terdapat proses pengguna menempatkan wajah di depan kamera kemudian akan diproses sampai wajah memiliki efek AR budaya.



Gambar 3.4. *Activity diagram* menggunakan *filter* AR budaya

1. *Activity Diagram* Melihat Objek 3D Budaya

*Activity diagram* melihat objek 3D budaya yang ditunjukkan pada Gambar 3.5, menujukkan bahwa terdapat proses memilih objek lalu sistem baru akan menampilkan objek budaya dalam bentuk 3D.



Gambar 3.5. *Activity diagram* melihat objek 3D budaya

1. *Activity Diagram* Melihat Informasi Lain Budaya

*Activity diagram* melihat informasi lain budaya yang ditunjukkan pada Gambar 3.6, menujukkan bahwa proses menuju referensi lain budaya akan dijalankan setelah pengguna memilih referensi budaya.



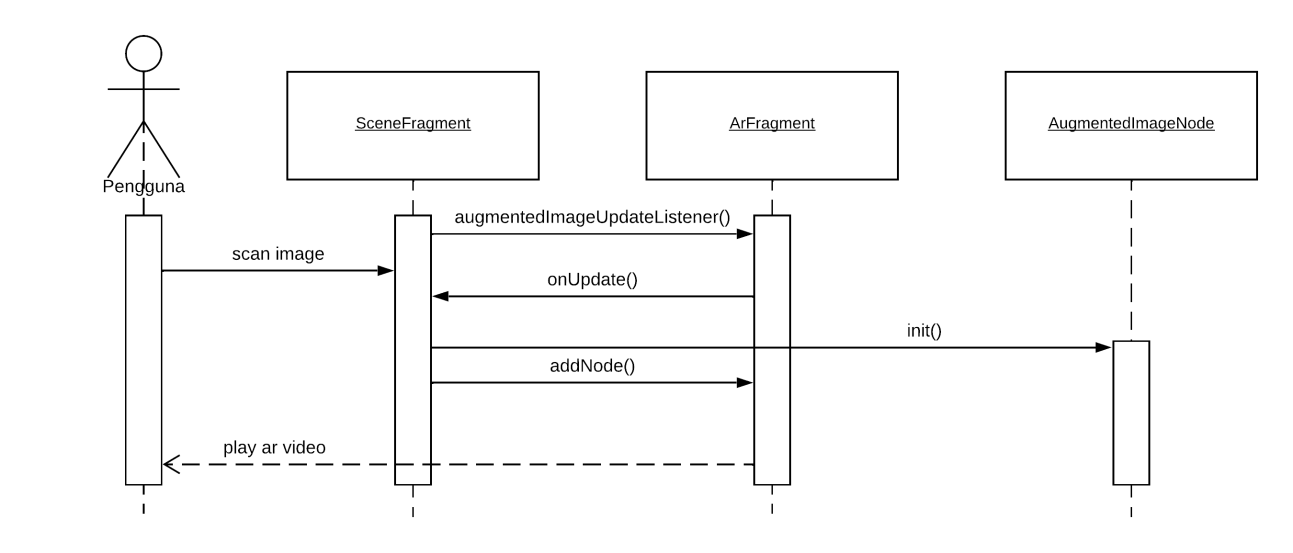
Gambar 3.6. *Activity diagram* melihat informasi lain budaya

1. ***Sequence Diagram* Aplikasi Edukasi Pengenalan Budaya Indonesia**

*Sequence Diagram* yang dibuat mejelaskan interaksi antar objek yang terjadi di *use case* pada sistem aplikasi. Berikut ini merupakan *Sequence Diagram* yang ada pada aplikasi edukasi pengenalan budaya Indonesia.

1. *Sequence Diagram* Memutar Video AR Budaya

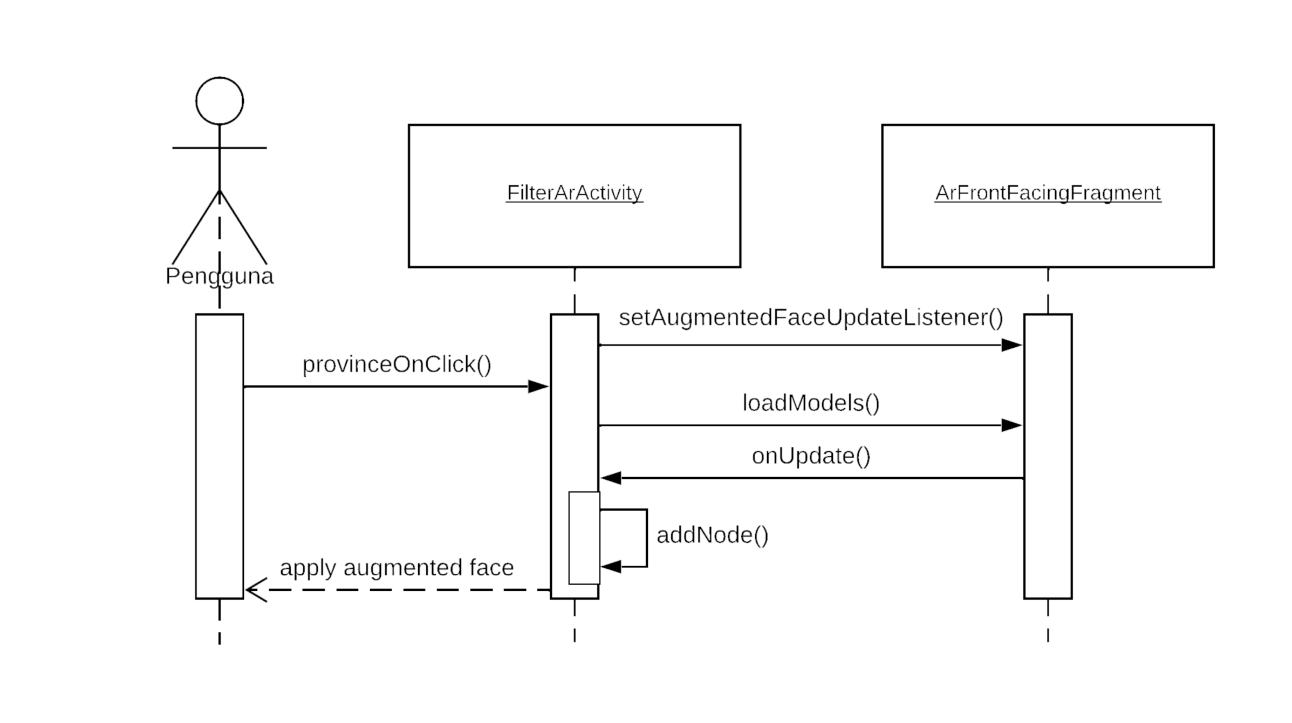
*Sequence diagram* memutar video AR budaya terdiri atas 1 pengguna dan 3 objek yang ada pada sistem. Diagram ini menggambarkan bagaimana objek pada sistem menjalankan fitur memutar video AR budaya yang dimulai dari aksi memindai gambar yang dilakukan oleh pengguna. Gambar *sequence diagram* memutar video AR budaya dapat dilihat pada Gambar 3.7.



Gambar 3.7. *Sequence diagram* memutar video AR budaya

1. *Sequence Diagram* Menggungakan *Filter* Wajah AR Budaya

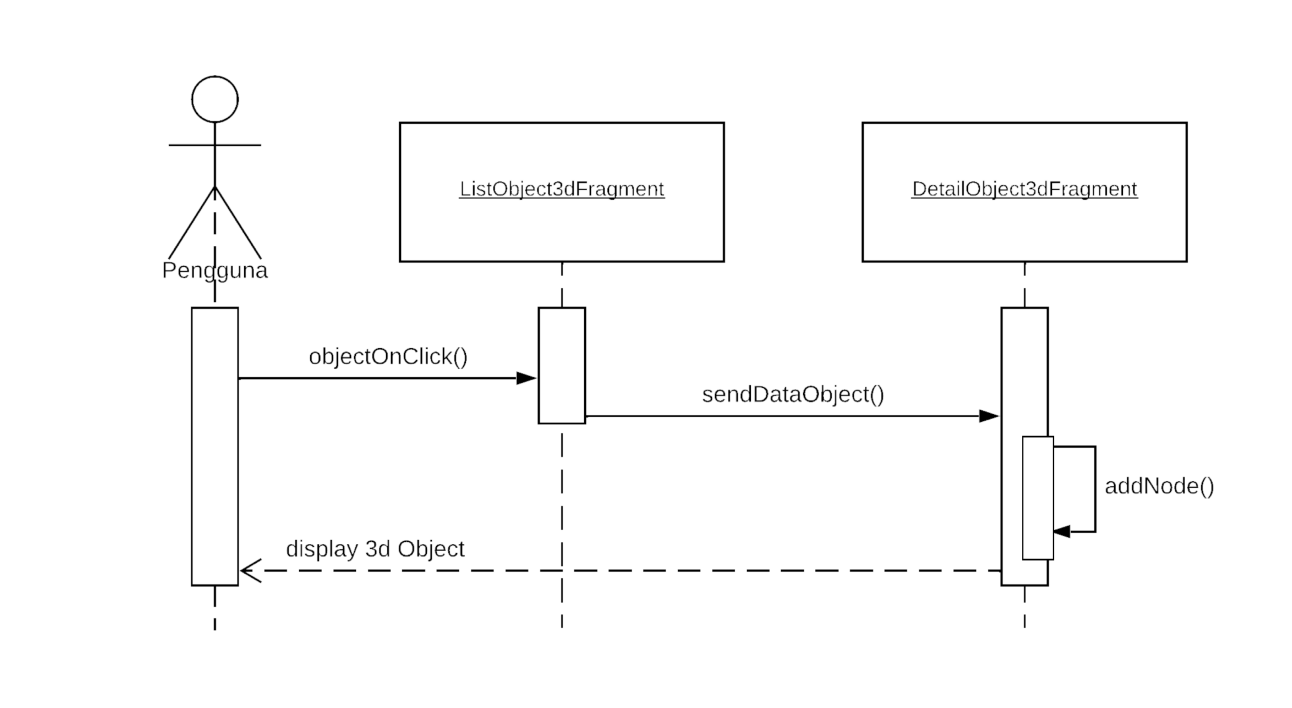
*Sequence diagram* menggunakan *filter* wajah AR budaya terdiri atas 1 pengguna dan 2 objek yang ada pada sistem. Diagram ini menggambarkan bagaimana pengguna dapat menjalankan fitur *filter* wajah. Gambar *sequence diagram* menggunakan *filter* wajah AR budaya dapat dilihat pada Gambar 3.8.



Gambar 3.8. *Sequence diagram* menggunakan *filter* wajah AR budaya

1. *Sequence Diagram* Melihat Objek 3D Budaya

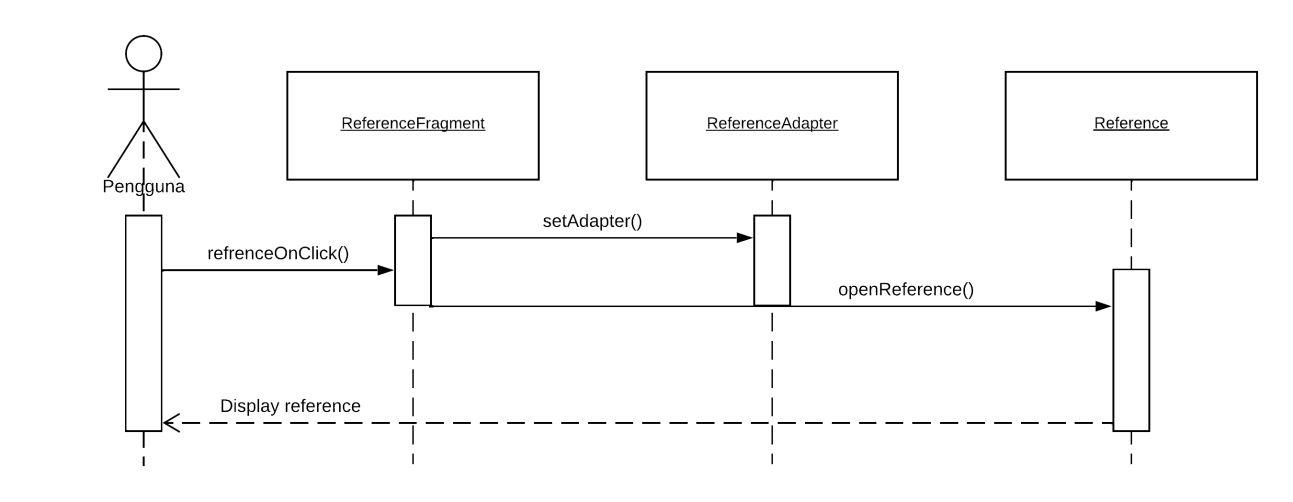
*Sequence diagram* melihat objek 3D budaya terdiri dari 1 pengguna dan 2 objek yang ada pada sistem. Diagram ini menggambarkan bagaimana sistem menjalankan fitur menampilkan objek 3D budaya yang dilihat oleh pengguna. Gambar *sequence diagram* melihat objek 3D budaya dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Sequence diagram* melihat objek 3D budaya

1. *Sequence Diagram* Melihat Referensi Lain Budaya

*Sequence diagram* melihat referensi lain budaya terdiri atas 1 pengguna dan 3 objek yang ada pada sistem. Diagram ini menggambarkan bagaimana pengguna bisa mengakses referensi lain dari budaya pada sistem. Gambar *sequence diagram* melihat referensi lain budaya dapat dilihat pada Gambar 3.10.



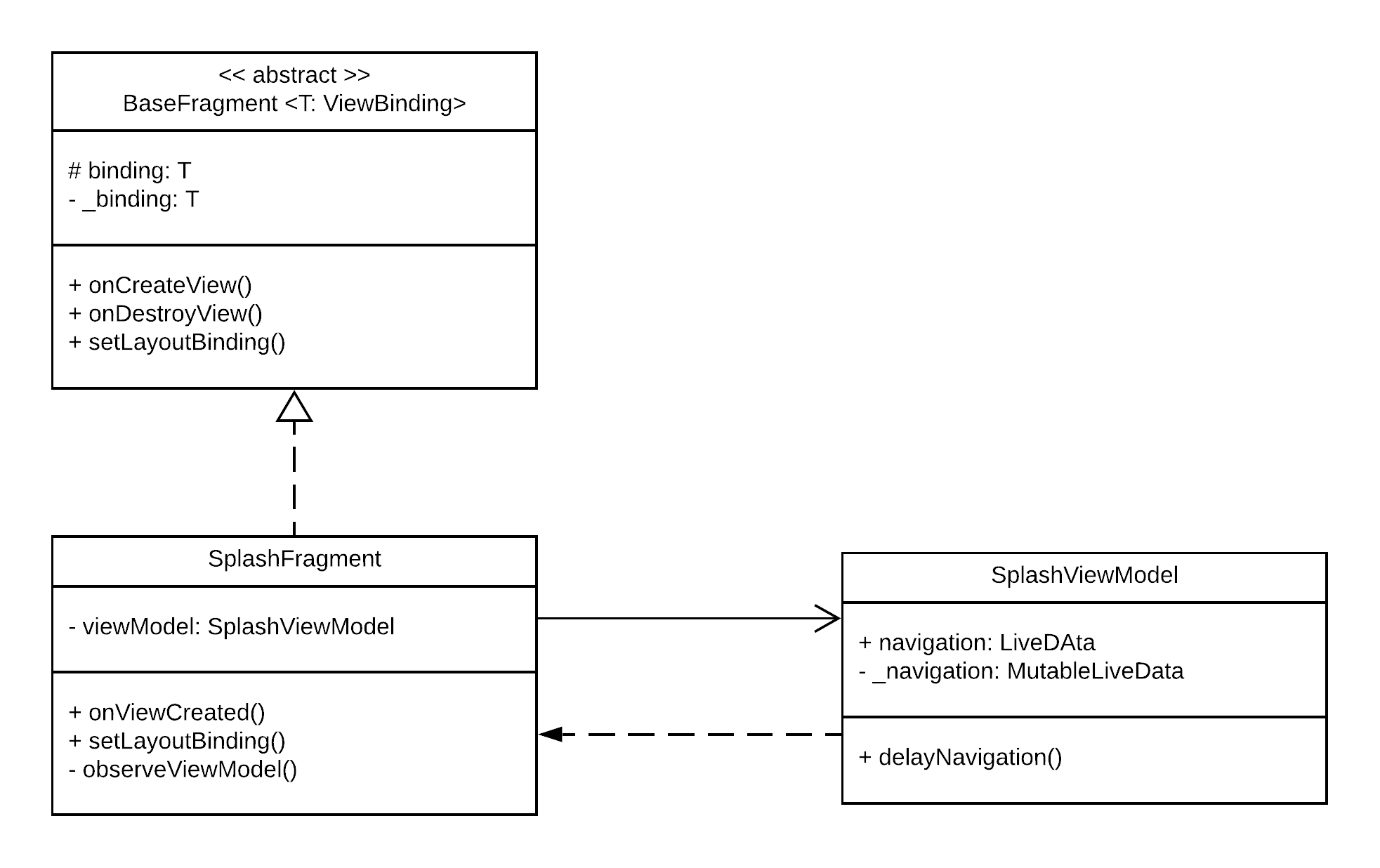
Gambar 3.10. *Sequence diagram* melihat referensi lain budaya

1. ***Class Diagram* Aplikasi Edukasi Pengenalan Budaya Indonesia**

*Class diagram* yang dibuat berupa *class*, atribut dan fungsiyang ada pada sistem aplikasi. Berikut ini merupakan *class diagram* yang terdapat pada aplikasi edukasi pengenalan budaya Indonesia.

1. *Class Diagram Splash*

*Class diagram splash* yang dibuat terdiri atas *class SplashFragment, SplashViewModel* dan *BaseFragment. Class SplashFragment* memiliki tanggung jawab untuk menangani semua perilaku terkait kebutuhan tampilan pada halaman *Splash*. *SplashFragment* berelasi dengan *class SplashViewModel* dengan tujuan agar semua proses logika pada aplikasi dapat dilakukan secara terpisah. *Class ViewMode*l juga memiliki tanggung jawab untuk menangani navigasi yang ada pada halaman *Splash*. Relasi terakhir, *class SplashFragment* mewarisi kemampuan *class BaseFragment*, karena *class BaseFragment* merupakan *class* yang mengatur konfigurasi bawaan suatu tampilan halaman *fragment*. *Class diagram splash* dapat dilihat pada Gambar 3.11.



Gambar 3.11. *Class diagram splash*

1. *Class Diagram Home*

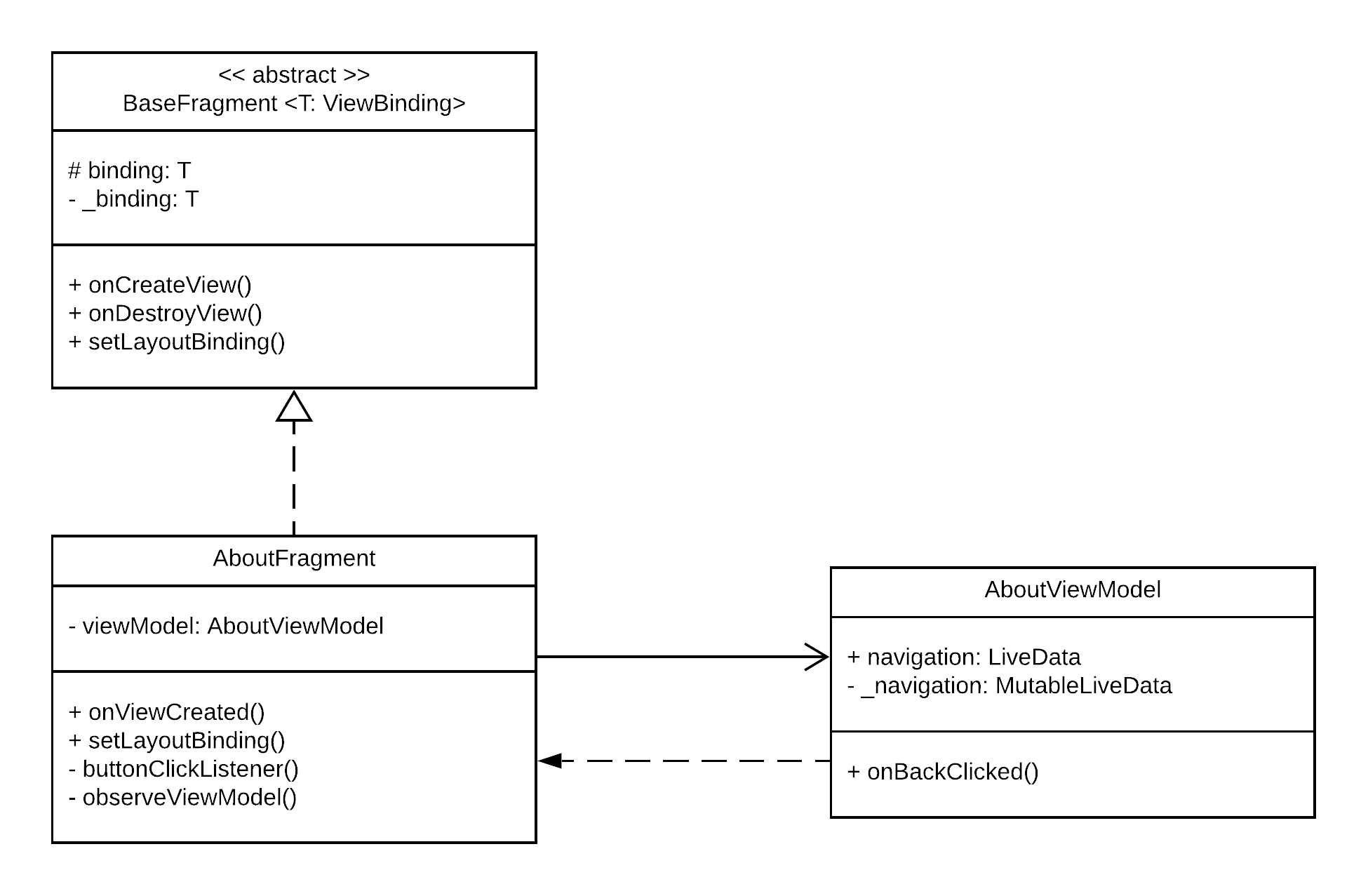
*Class diagram home* yang dibuat terdiri atas *class HomeFragment, HomeViewModel* dan *BaseFragment. Class HomeFragment* memiliki tanggung jawab untuk menangani semua perilaku terkait kebutuhan tampilan pada halaman *Home*. *HomeFragment* berelasi dengan *class HomeViewModel* dengan tujuan agar semua proses logika pada aplikasi dapat dilakukan secara terpisah. *Class ViewMode*l juga memiliki tanggung jawab untuk menangani navigasi, menampilkan *dialog* dan aksi dari komponen halaman. Relasi terakhir, *Class HomeFragment* mewarisi kemampuan *class BaseFragment*, karena *class BaseFragment* merupakan *class* yang mengatur konfigurasi bawaan suatu tampilan halaman *fragment*. *Class diagram home* dapat dilihat pada Gambar 3.12.

**

Gambar 3.12. *Class diagram home*

1. *Class Diagram About*

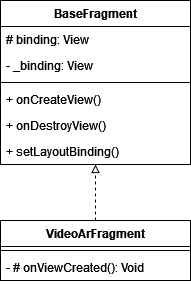
*Class diagram about* yang dibuat terdiri atas *class AboutFragment, AboutViewModel* dan *BaseFragment. Class AboutFragment* memiliki tanggung jawab untuk menangani semua perilaku terkait kebutuhan tampilan pada halaman *About*. *AboutFragment* juga berelasi dengan *class ViewModel* dengan tujuan agar semua proses logika pada aplikasi dapat ditangani secara terpisah. *Class ViewMode*l juga memiliki tanggung jawab untuk menangani navigasi yang ada pada halaman *About*. Relasi terakhir, *class AboutFragment* mewarisi kemampuan *class BaseFragment*, karena *class BaseFragment* merupakan *class* yang mengatur konfigurasi bawaan suatu tampilan halaman *fragment*. *class diagram about* dapat dilihat pada Gambar 3.13.



Gambar 3.13. *Class diagram about*

1. *Class Diagram Video AR*

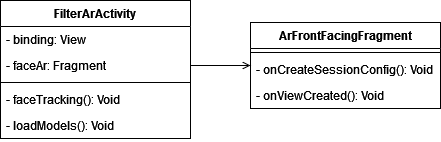
*Class diagram video AR* terdiri atas *class BaseFragment* dan *VideoArFragment*. *Class BaseFragment* bertanggung jawab untuk menampung semua kode-kode umum terkait pengaturan tampilan halaman *fragment*. Lalu *class VideoArFragment* mewarisi *class BaseFragment* dan *class* ini berfungsi untuk mengani aksi dari pengguna. *Class diagram video AR* dapat dilihat pada Gambar 3.14.

**

Gambar 3.14. *Class diagram* video *AR*

1. *Class Diagram Filter AR*

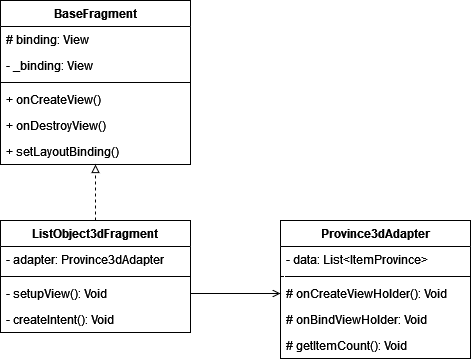
*Class diagram filter AR* terdiri atas *class FilterActivity* dan *ArFrontFacingFragment*. *Class FilterActivity* merupakan *class* yang menampung *class ArFrontFacingFragment* kemudian juga terdapat fungsi untuk membuat model objek wajah dan fungsi untuk menambahkan objek model tersebut ke *class ArFrontFacingFragment*. *Class diagram filter AR* dapat dilihat pada Gambar 3.15.

**

Gambar 3.15. *Class diagram filter* AR

1. *Class Diagram List 3D Object*

*Class diagram list 3D object* terdiri atas *class BaseFragment, ListObject3dFragment* dan *Province3dAdapter. Class BaseFragment* merupakan *class* yang bertanggung jawab mengatur konfigurasi halaman tampilan *fragment*. Lalu *class ListObject3dFragment* merupakan *class* yang menampilkan daftar objek dari budaya provinsi. Dan *class Province3dAdapter* sebagai *class* yang mengatur daftar objek dari budaya provinsi tersebut. *Class diagram list 3D object* dapat dilihat pada Gambar 3.16.

**

Gambar 3.16. *Class diagram* *list* 3D *object*

1. *Class Diagram References*

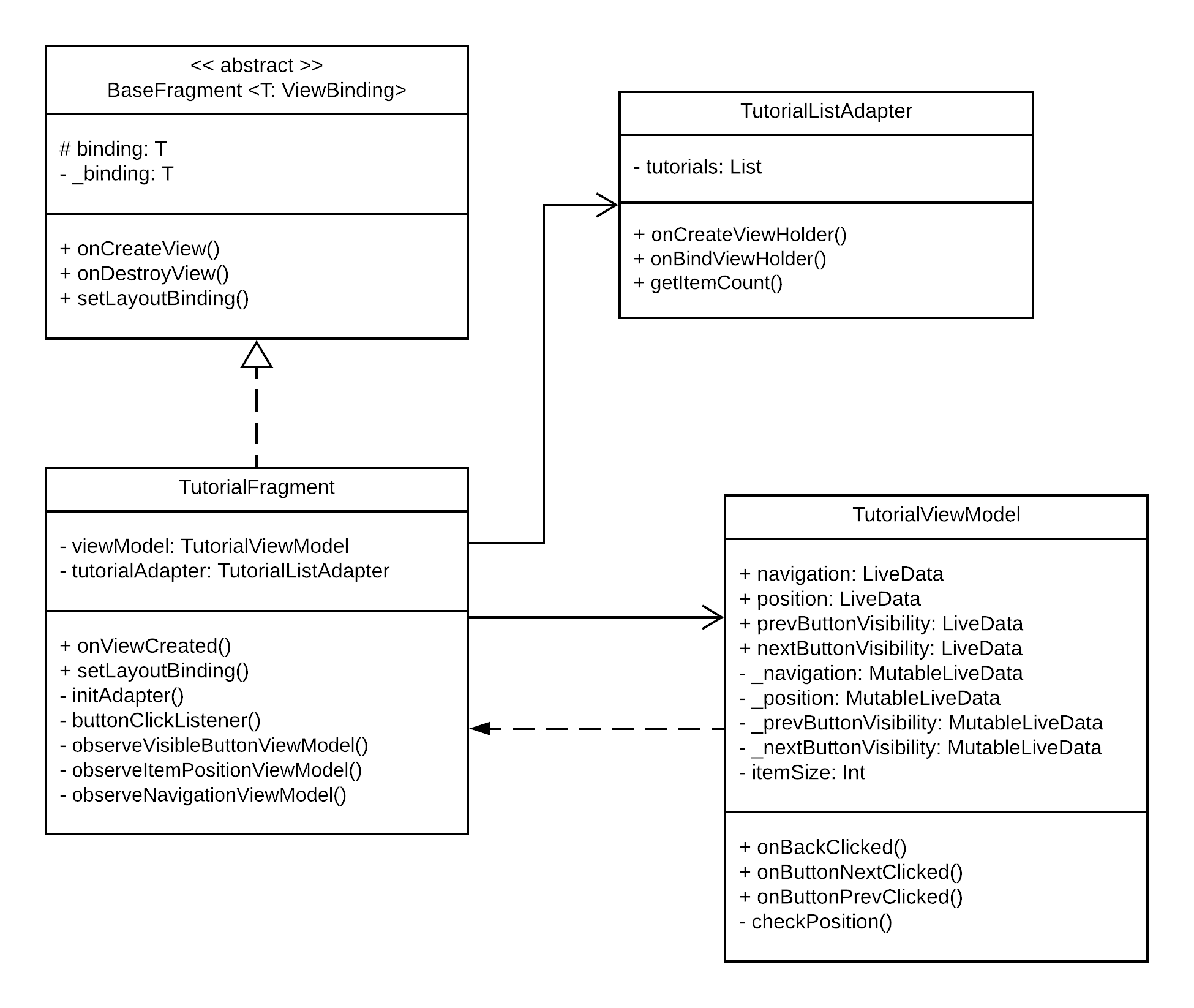
*Class diagram references* terdiri atas class *BaseFragment*, *RefFragment* dan *RefAdapter*. *Class BaseFragment* merupakan *class* yang menampung kode konfigurasi untuk tampilan halaman *fragment*. Lalu *class RefFragment* memiliki tanggung jawab sebagai *class* yang menampilkan daftar referensi dari budaya provinsi. Dan *class RefAdapter* sebagai *class* yang mengatur daftar referensi tersebut. *Class diagram references* dapat dilihat pada Gambar 3.17.



Gambar 3.17. *Class diagram references*

1. *Class Diagram Tutorial*

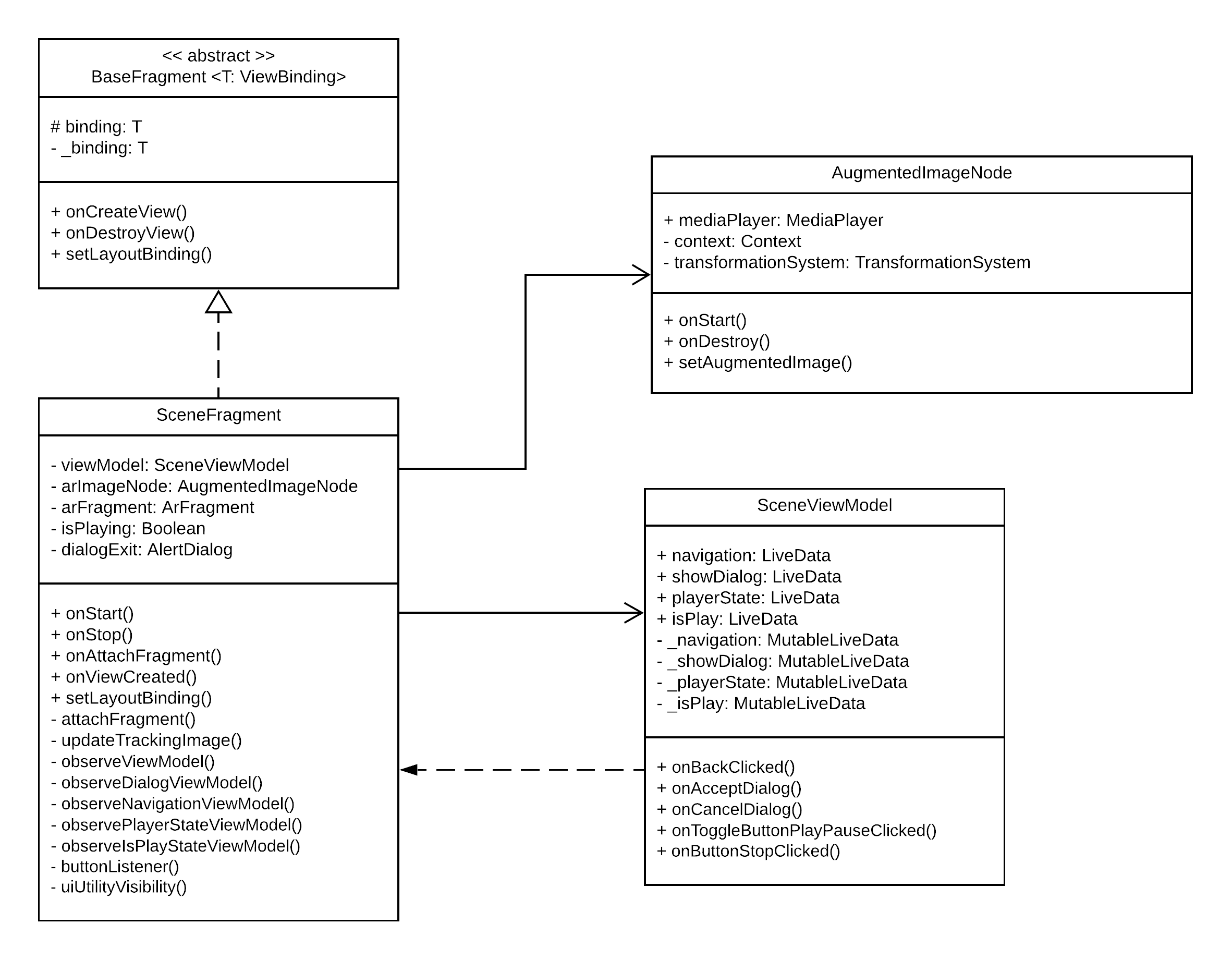
*Class diagram tutorial* yang dibuat terdiri atas *class TutorialFragment, TutorialViewModel, TutorialListAdapter* dan *BaseFragment. Class TutorialFragment* memiliki tanggung jawab untuk menangani semua perilaku terkait kebutuhan tampilan pada halaman *Tutorial*. *TutorialFragment* berelasi dengan *class ViewModel* dengan tujuan agar semua proses logika pada aplikasi dapat ditangani secara terpisah. *TutorialFragment* juga berelasi dengan *class TutorialListAdapter* karena halaman tutorial memiliki informasi berupa daftar cara penggunaan fitur dari aplikasi. Relasi terakhir, *class TutorialFragment* mewarisi kemampuan *class BaseFragment*, karena *class BaseFragment* merupakan *class* yang mengatur konfigurasi bawaan suatu tampilan halaman. *Class diagram tutorial* dapat dilihat pada Gambar 3.18.



Gambar 3.18. *Class diagram* tutorial

1. *Class Diagram Scene*

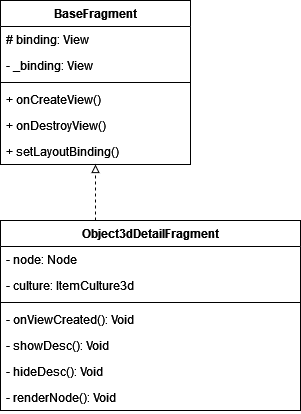
*Class diagram scene* yang dibuat terdiri atas *class SceneFragment, SceneViewModel, AugmentedImageNode* dan *BaseFragment. Class SceneFragment* memiliki tanggung jawab untuk menangani semua perilaku terkait kebutuhan tampilan pada halaman *scene*. *SceneFragment* berelasi dengan *class ViewModel* dengan tujuan agar semua proses logika pada aplikasi dapat ditangani secara terpisah. *SceneFragment* juga berelasi dengan *class AugmentedNode*, *class* tersebut bertugas untuk membuat objek yang ditampilkan pada dunia realita. Relasi terakhir, *class SceneFragment* mewarisi kemampuan *class BaseFragment*, karena *class BaseFragment* merupakan *class* yang mengatur konfigurasi bawaan suatu tampilan halaman. *Class diagram scene* dapat dilihat pada Gambar 3.19.

**

Gambar 3.19. *Class diagram scene*

1. *Class Diagram Detail 3D Object*

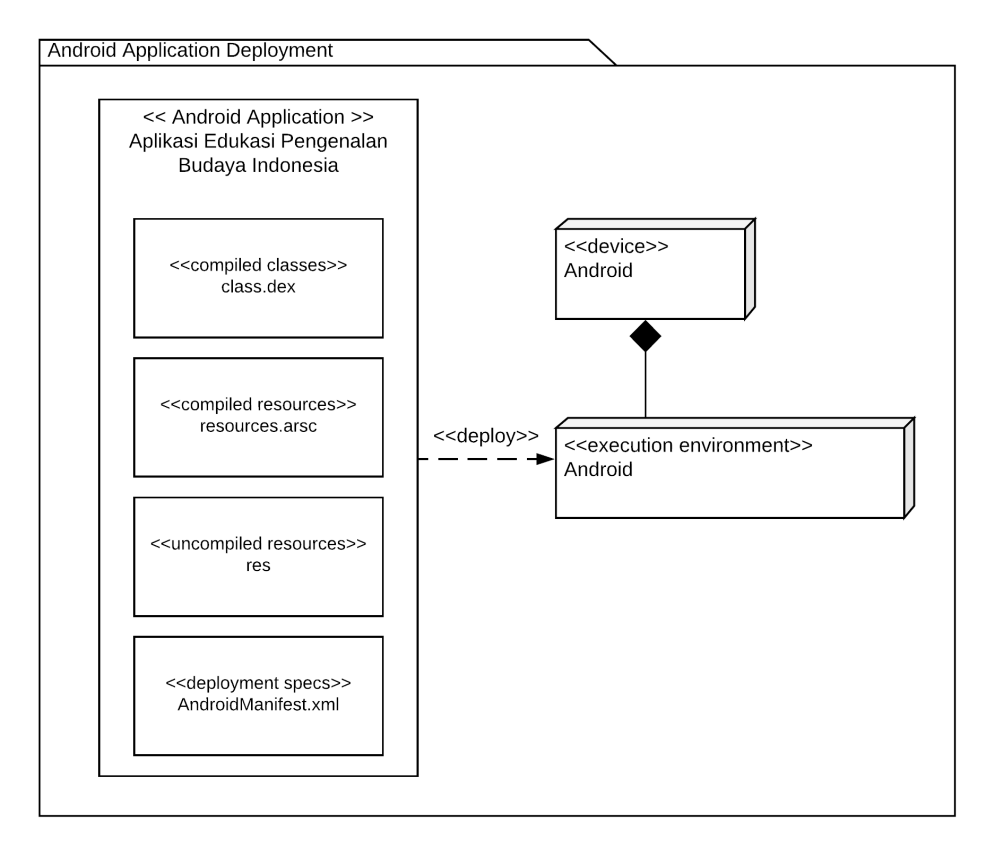
*Class diagram detail 3D object* terdiri atas *class BaseFragment* dan *Object3dDetailFragment*. *Class BaseFragment* berfungsi sebagai basis *class* dari *fragment*. Basis *class* tersebut menampung kode-kode umum yang digunakan oleh *class fragment* dan juga mengatur konfigurasi bawaan tampilan halaman. Kemudian *class Object3dDetailFragment* merupakan *class* yang digunakan untuk memuat informasi objek budaya dalam bentuk 3D. Gambar *Class diagram detail 3D object* dapat dilihat pada Gambar 3.20.



Gambar 3.20. *Class diagram detail 3D object*

1. ***Deployment Diagram* Aplikasi Edukasi Pengenalan Budaya Indonesia**

*Deployment diagram* aplikasi edukasi pengenalan budaya menunjukkan bagaimana aplikasi di *compil*e yang kemudian di *deploy* ke perangkat *mobile* android. *Deployment diagram* dari aplikasi pengenalan budaya Indonesia dapat dilihat pada Gambar 3.21.



Gambar 3.21. *Deployment diagram* aplikasi edukasi pengenalan budaya Indonesia

1. **Perancangan Struktur Navigasi Aplikasi Edukasi Pengenalan Budaya Indonesia**

Perancangan struktur navigasi menjelaskan struktur navigasi yang ada pada sistem aplikasi berdasarkan halaman-halaman yang ada. Halaman aplikasi pada aplikasi pengenalan budaya Indonesia diantaranya halaman *splash*, *home*, *about*, video AR, *filter* AR, *list* 3D *object*, *references*, tutorial, *scene* dan *detail* 3D *object*.

1. Halaman *Splash*

Halaman *splash* merupakan halaman yang akan tampil pada saat pertama kali aplikasi dibuka. Pada halaman ini terdapat tampilan logo dan nama aplikasi. Tujuan dari halaman ini sebagai tampilan untuk memberitahukan kepada pengguna bahwa pengguna telah memasuki aplikasi.

1. Halaman *Home*

Halaman *home* bertujuan sebagai halaman utama dari aplikasi. Halaman *home* akan tampil setelah halaman *splash*. Pada halaman ini terdapat banyak tombol yang memiliki tujuan sebagai navigasi ke halaman lainnya.

1. Halaman *About*

Halaman *about* merupakan halaman yang menampilkan informasi tentang aplikasi, seperti nama aplikasi, nama pembuat aplikasi, sumber informasi budaya dan sebagainya. Halaman *about* hanya dapat diakses dari halaman *Home* dengan cara menekan tombol tentang.

1. Halaman Video AR

Halaman video AR merupakan halaman yang akan tampil setelah menekan tombol menu video AR pada halaman *home*. Halaman ini berisi tombol 3 tombol, 2 diantaranya untuk menampilkan halaman tutorial dan *scene*.

1. Halaman *Filter* AR

Halaman *Filter* AR merupakan halaman yang berfungsi untuk menjalankan *filter* AR yang diterapkan pada wajah pengguna. Halaman ini menampilkan hasil dari kamera depan ponsel yang digunakan untuk menangkap wajah pengguna. Halaman *filter* AR bisa di akses dengan menekan tombol menu *filter* AR yang berada di halaman *home*.

1. Halaman *List* 3D *Object*

Halaman *list* 3D *object* menampilkan daftar budaya berdasarkan provinsi yang jika ditekan akan pindah ke halaman *detail*. Daftar budaya tersebut adalah budaya-budaya yang bisa tampil dalam bentuk objek 3D. Halaman ini bisa di akses dengan menekan tombol menu 3D *object* yang ada di halaman *home*.

1. Halaman *References*

Halaman *references* merupakan halaman yang menampilkan daftar referensi dari budaya berdasarkan provinsi. Jika referensi di tekan, maka akan membuka aplikasi lain berdasarkan referensi yang di klik. Referensi yang disediakan berupa referensi teks dan video.

1. Halaman Tutorial

Halaman tutorial memiliki tujuan untuk memeberikan informasi terkait cara atau langkah-langkah bagimana mengakses fiture utama aplikasi. Halaman tutorial hanya bisa diakses dari halaman *home* dengan cara menekan tombol tutorial.

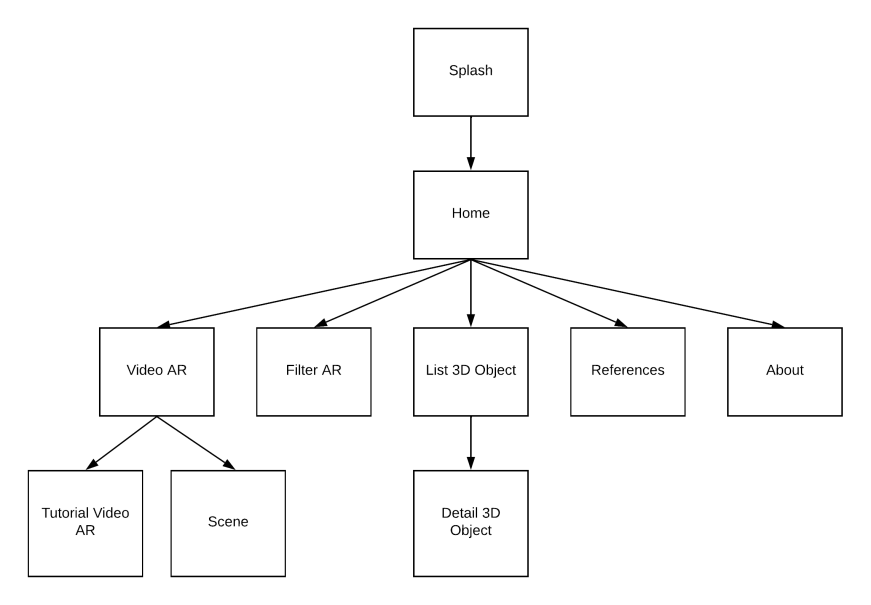
1. Halaman *Scene*

Halaman *scene* adalah halaman yang akan menampilkan fitur utama aplikasi, yaitu video AR. Halaman *scene* berupa tampilan kamera dari perangkat ponsel dan terdapat bingkai yang digunakan untuk memindai logo dari suatu provinsi. Halaman *scene* hanya dapat diakses dari halaman *home* dengan cara menekan tombol *scene*.

1. *Detail* 3D *Object*

Halaman *detail* 3D *object* merupakan halaman yang akan tampil setelah daftar budaya yang ada di halaman *list* 3D *object* di tekan. Halaman ini menampilkan budaya dari suatu provinsi dalam bentuk objek 3D.

Gambar perancangan struktur navigasi aplikasi edukasi pengenalan budaya Indonesia dapat dilihat pada Gambar 3.22.



Gambar 3.22. Perancangan struktur navigasi aplikasi edukasi pengenalan budaya Indonesia

1. **Perancangan Antarmuka Aplikasi Edukasi Pengenalan Budaya Indonesia**

Perancangan antarmuka menggambarkan rancangan antarmuka halaman yang ada pada aplikasi pengenalan budaya Indonesia. Rancangan antarmuka yang dibuat diantaranya perancangan antarmuka halaman *splash*, *home*, *about*, video AR, *filter* AR, *list* 3D *object*, *references*, tutorial, *scene* dan detail 3D *object*.

1. **Perancangan Antarmuka Halaman *Splash***

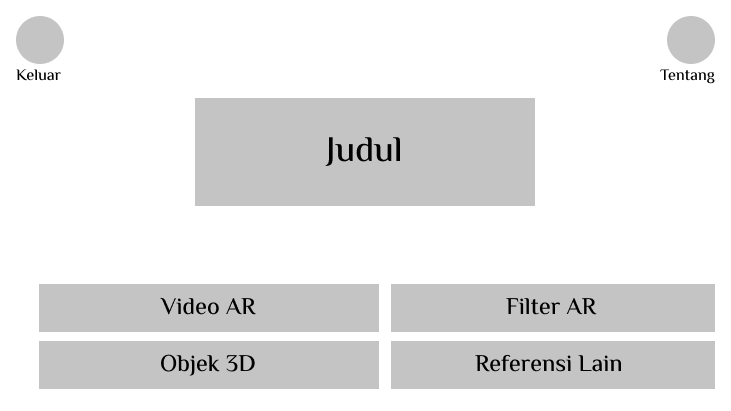
Rancangan antarmuka halaman *splash* hanya terdiri dari logo aplikasi dan nama aplikasi di posisi tengah layar. Perancangan antarmuka halaman *splash*, dapat dilihat pada Gambar 3.23.



Gambar 3.23. Perancangan antarmuka halaman *Splash*

1. **Perancangan Antarmuka Halaman *Home***

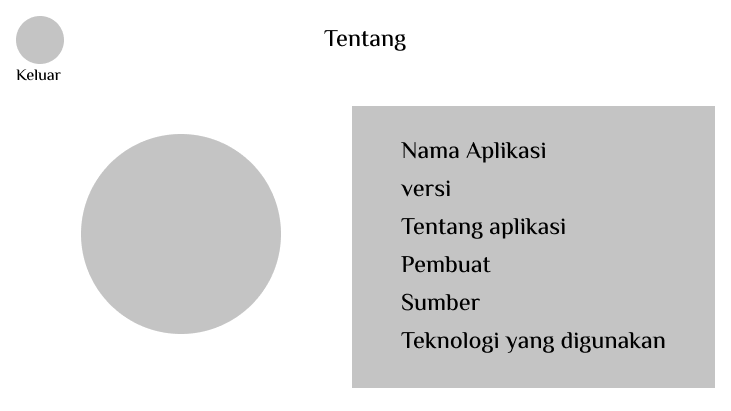
Rancangan antarmuka halaman *home* terdiri atas 1 teks judul dan 6 tombol. Perancangan antarmuka halaman *home*, dapat dilihat pada Gambar 3.24.



Gambar 3.24. Perancangan antarmuka halaman *home*

1. **Perancangan Antarmuka Halaman *About***

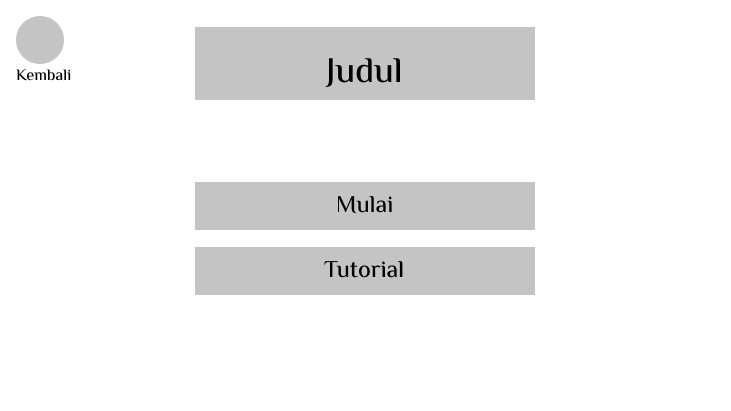
Rancangan antarmuka halaman *about* terdiri atas 1 teks judul, 1 tombol keluar dan informasi-informasi terkait aplikasi, seperti logo aplikasi, nama aplikasi, versi aplikasi dan sebagainya. Perancangan antarmuka halaman *about*, dapat dilihat pada Gambar 3.25.



Gambar 3.25. Perancangan antarmuka halaman *about*

1. **Perancangan Antarmuka Halaman Video AR**

Rancangan antarmuka halaman video AR terdiri atas 1 teks judul dan 3 tombol. Perancangan antarmuka halaman video AR, dapat dilihat pada Gambar 3.26.



Gambar 3.26. Perancangan antarmuka halaman video AR

1. **Perancangan Antarmuka Halaman *Filter* AR**

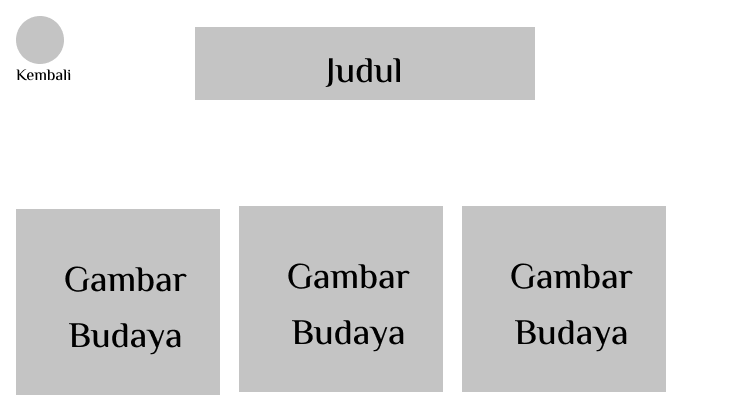
Rancangan antarmuka halaman video AR terdiri atas 1 teks judul, 1 tombol dan 1 gambar logo provinsi budaya. Perancangan antarmuka halaman *filter* AR, dapat dilihat pada Gambar 3.27.



Gambar 3.27. Perancangan antarmuka halaman *filter* AR

1. **Perancangan Antarmuka Halaman *List* 3D *Object***

Rancangan antarmuka halaman *list* 3D *object* terdiri atas 1 teks judul, 1 tombol dan daftar berupa gambar-gambar budaya. Perancangan antarmuka halaman *list* 3D *object*, dapat dilihat pada Gambar 3.28.

**

Gambar 3.28. Perancangan antarmuka halaman *list* 3D *object*

1. **Perancangan Antarmuka Halaman *References***

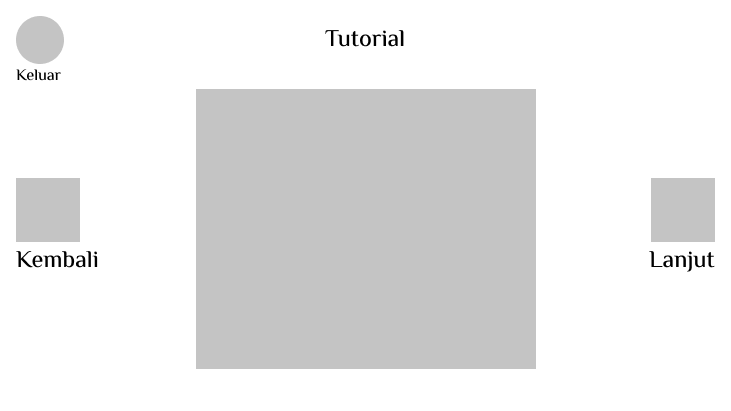
Rancangan antarmuka halaman *references* terdiri atas 1 teks judul, 1 tombol dan daftar berupa logo provinsi beserta teks referensi. Perancangan antarmuka halaman *references*, dapat dilihat pada Gambar 3.29.

**

Gambar 3.29. Perancangan antarmuka halaman *references*

1. **Perancangan Antarmuka Halaman Tutorial**

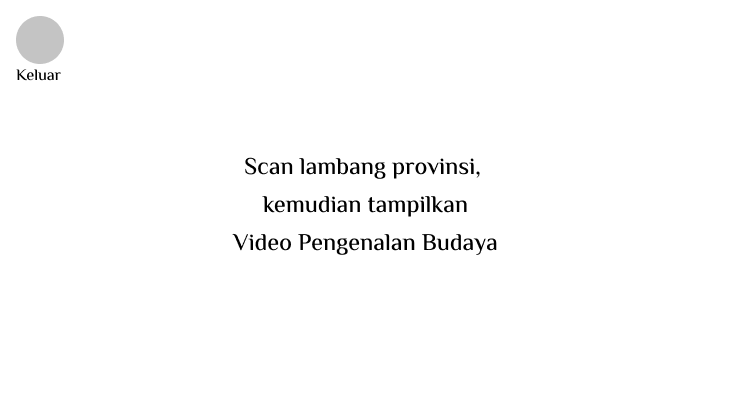
Rancangan antarmuka halaman tutorial terdiri atas 1 teks judul, 1 tombol keluar, 2 tombol navigasi untuk tutorial dan 1 gambar informasi terkait tutorial tepat berada di posisi tengah layar. Perancangan antarmuka halaman tutorial, dapat dilihat pada Gambar 3.30.



Gambar 3.30. Perancangan antarmuka halaman tutorial

1. **Perancangan Antarmuka Halaman *Scene***

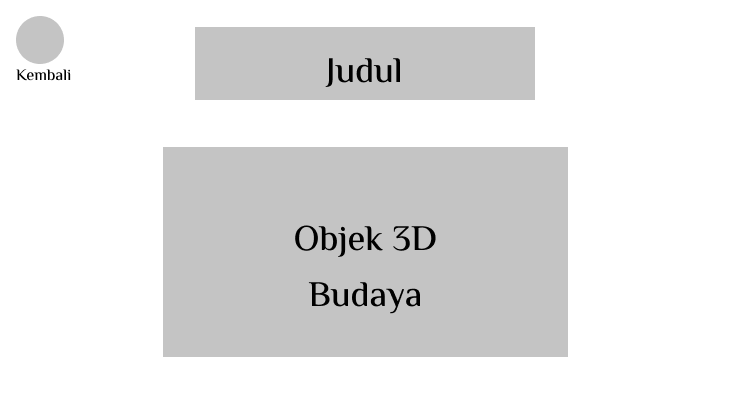
Rancangan antarmuka halaman *scene* terdiri atas 1 tombol berda di pojok kiri atas dan informasi untuk pengguna berada di posisi tengah. Perancangan antarmuka halaman *scene*, dapat dilihat pada Gambar 3.31.



Gambar 3.31. Perancangan antarmuka halaman *scene*

1. **Perancangan Antarmuka Halaman *Detail* 3D *Object***

Rancangan antarmuka halaman *detail* 3D *object* terdiri atas 1 teks judul, 1 tombol dan objek 3D dari budaya berada di posisi tengah. Perancangan antarmuka halaman *detail* 3D *object*, dapat dilihat pada Gambar 3.32.

****

Gambar 3.32. Perancangan antarmuka halaman *detail* 3D *object*

**BAB 4**

**IMPLEMENTASI DAN PENGUJIAN**

1. **Implementasi Sistem**

Implementasi sistem aplikasi edukasi pengenalan budaya Indonesia disesuaikan dengan analisis dan rancangan sistem yang telah dibuat pada bab sebelumnya. Implementasi sistem yang akan dijelaskan yaitu implementasi modul dan antarmuka sistem.

1. **Implementasi Modul Aplikasi Edukasi Pengenalan Budaya Indonesia**

Implementasi modul adalah implementasi dari rancangan UML *class diagram* dalam bentuk kode program. Implementasi modul yang ada pada sistem yaitu modul *base*, *splash*, *home*, *about*, video AR, *filter* AR, *list 3d object*, *references*, detail 3D object, tutorial dan *scene*.

1. **Modul *Base***

Modul *base* merupakan modul yang digunakan sebagai basis kode yang menangani semua fungsi yang terkait kode antarmuka. Implementasi modul base dapat dilihat pada Gambar 4.1.



Gambar 4.1. Modul *base*

1. **Modul *Splash***

Implementasi modul *splash* terdiri atas 2 bagian, yaitu *SplashFragment* dan *SplashViewModel*. Implentasi *SplashFragment* dapat dilihat pada Gambar 4.2 dan *SplashViewModel* pada Gambar 4.3.

****

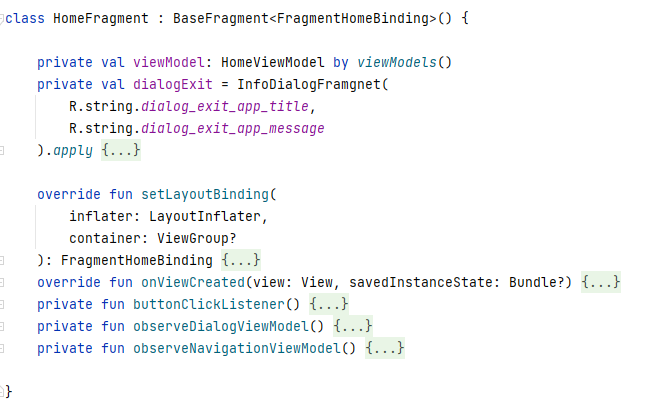
Gambar 4.2. Modul *splash* (*SplashFragment*)

****

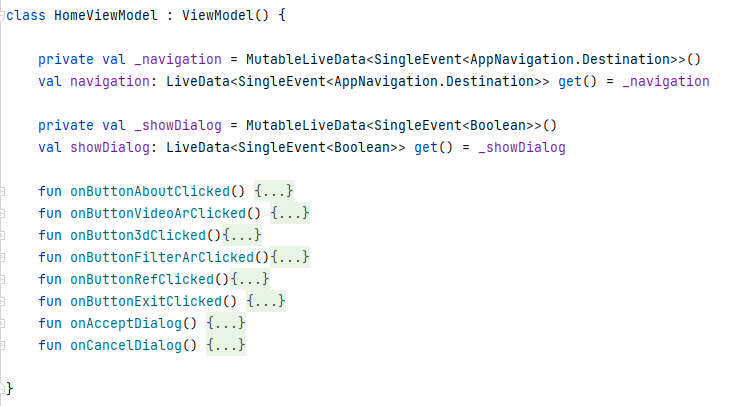
Gambar 4.3. Modul *splash* (*SplashViewModel*)

1. **Modul *Home***

Implementasi modul *home* terdiri atas 2 bagian, yaitu *HomeFragment* dan *HomeViewModel*. Implentasi *HomeFragment* dapat dilihat pada Gambar 4.4 dan *HomeViewModel* pada Gambar 4.5.



Gambar 4.4. Modul *home* (*HomeFragment*)

****

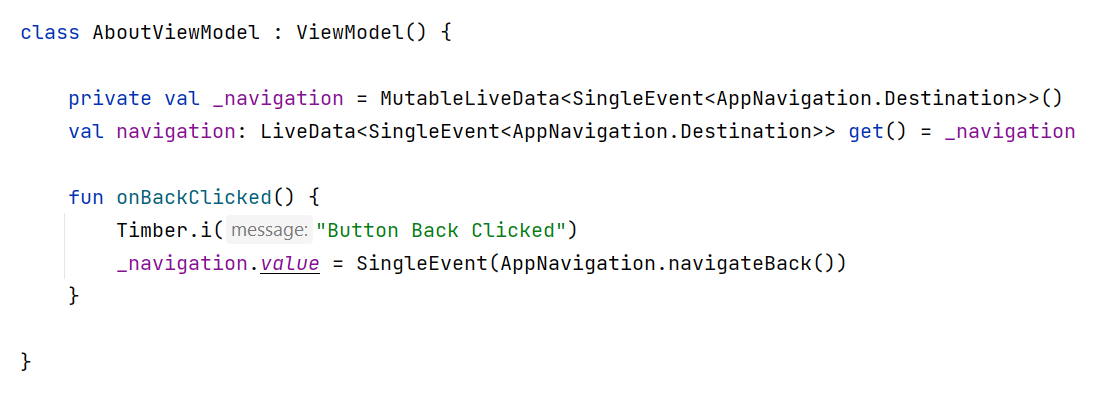
Gambar 4.5. Modul *home* (*HomeViewModel*)

1. **Modul *About***

Implementasi modul *about* terdiri atas 2 bagian, yaitu *AboutFragment* dan *AboutViewModel*. Implentasi *AboutFragment* dapat dilihat pada Gambar 4.6 dan *AboutViewModel* pada Gambar 4.7.



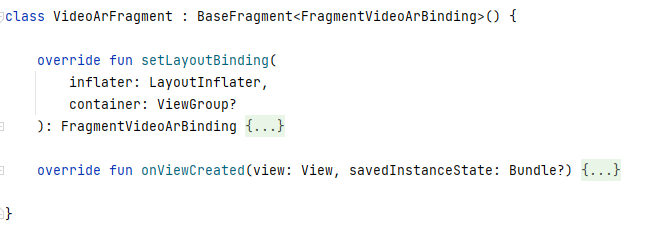
Gambar 4.6. Modul *about* (*AboutFragment*)

****

Gambar 4.7. Modul *about* (*AboutViewModel*)

1. **Modul Video AR**

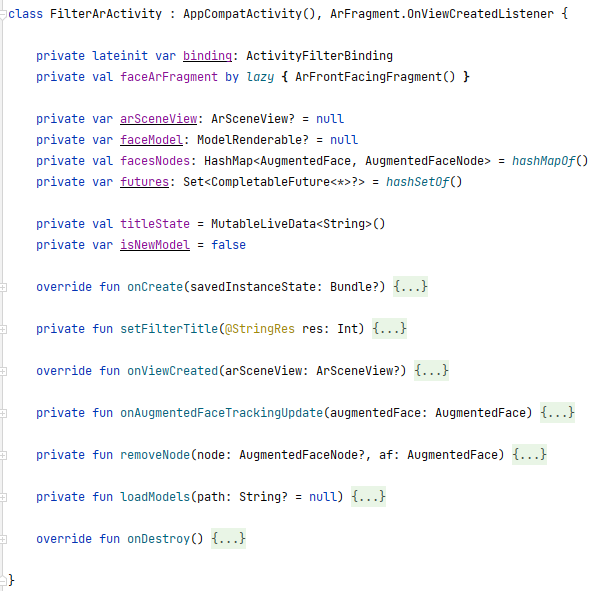
Implementasi modul video AR dapat dilihat pada Gambar 4.8.



Gambar 4.8. Modul video AR

1. **Modul *Filter* AR**

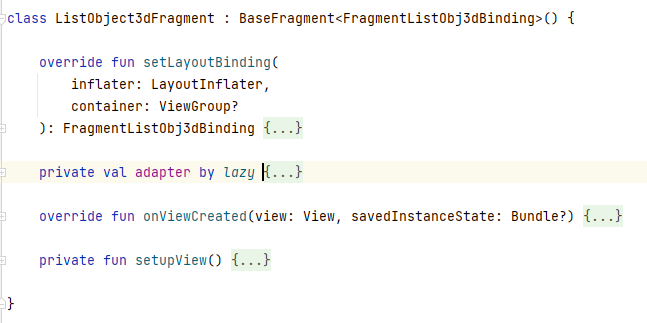
Implementasi modul *filter* AR dapat dilihat pada Gambar 4.9.



Gambar 4.9. Modul *filter* AR

1. **Modul *List* 3D *Object***

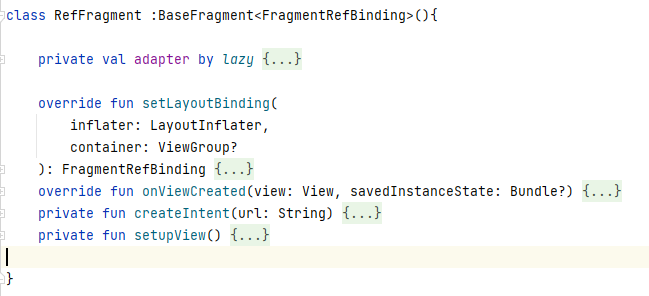
Implementasi modul *list* 3D *object* dapat dilihat pada Gambar 4.10.

****

Gambar 4.10. Modul *list* 3D *object*

1. **Modul *References***

Implementasi modul *references* dapat dilihat pada Gambar 4.11.



Gambar 4.11. Modul *references*

1. **Modul *Detail* 3D *Object***

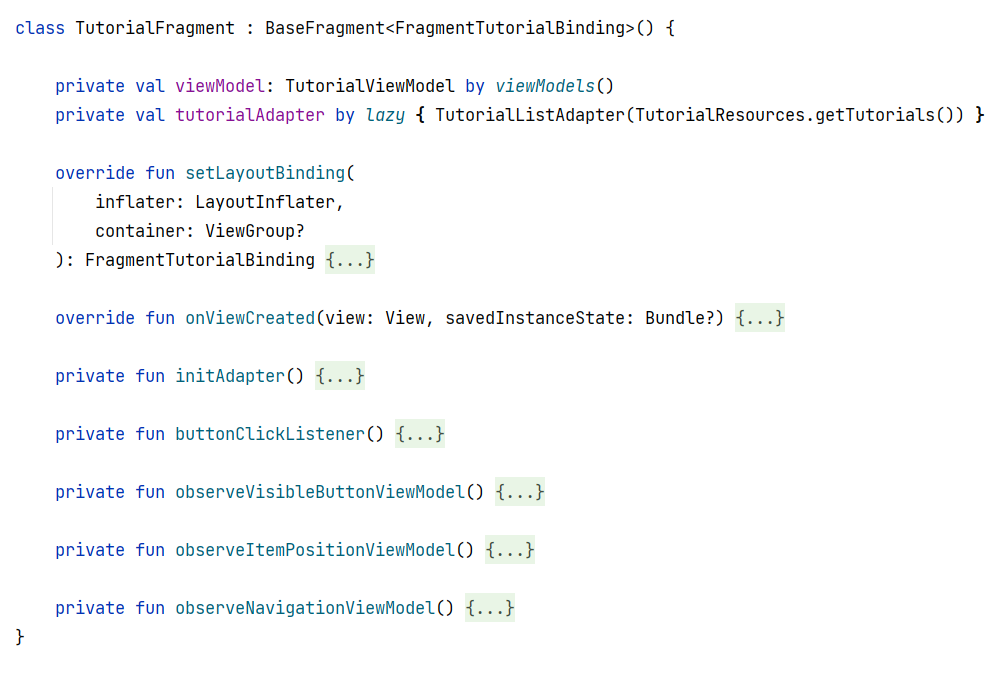
Implementasi modul *detail* 3D *object* dapat dilihat pada Gambar 4.12.



Gambar 4.12. Modul *detail* 3D *object*

1. **Modul Tutorial**

Implementasi modul *tutorial* terdiri atas 2 bagian, yaitu *TutorialFragment* dan *TutorialViewModel*. Implentasi *TutorialFragment* dapat dilihat pada Gambar 4.13 dan *TutorialViewModel* pada Gambar 4.14.



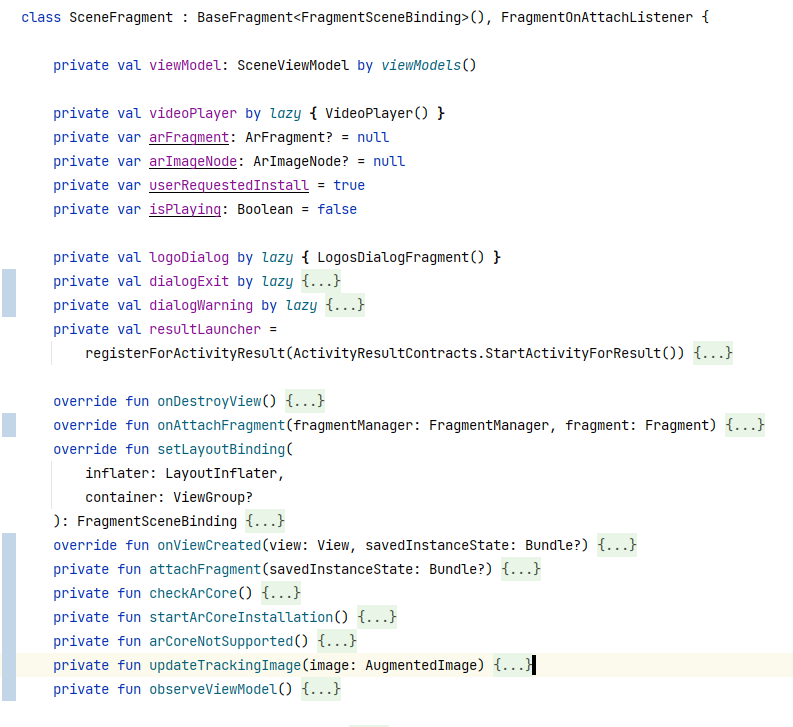
Gambar 4.13. Modul tutorial (*TutorialFragment*)

****

Gambar 4.14. Modul tutorial (*TutorialViewModel*)

1. **Modul *Scene***

Implementasi modul *scene* terdiri atas beberapa bagian, yaitu *SceneFragment*, *SceneViewModel* dan *ArImageNode*. Implentasi *SceneFragment*, *SceneViewModel* dan *ArImageNode* dapat dilihat pada Gambar 4.15, 4.16 dan 4.17.



Gambar 4.15. Modul *scene* (*SceneFragment*)

****

Gambar 4.16. Modul *scene* (*SceneViewModel*)

****

Gambar 4.17. Modul *scene* (*ArImageNode*)

1. **Impelemntasi Antarmuka Aplikasi Edukasi Pengenalan Budaya Indonesia**

Impelementasi antarmuka dibuat berdasarkan hasil perancangan antarmuka. Imeplementasi antarmuka terdiri atas halaman *splash*, *home*, *about,* video AR, *filter* AR, *list* 3D *object*, *references*, *detail* 3D *object*, tutorial dan *scene*.

1. **Halaman *Splash***

Impelementasi halaman *splash* disesuaikan dengan rancangan yang sudah dibuat, pada halaman ini terdiri dari logo dan nama aplikasi yang berada pada tengah layar perangkat. Implementasi halaman *splash* dapat dilihat pada Gambar 4.18.



Gambar 4.18. Implementasi halaman *splash*

1. **Halaman *Home***

Impelementasi halaman *home* disesuaikan dengan rancangan yang sudah dibuat, pada halaman ini terdiri 6 tombol yang digunakan untuk navigasi antar halaman dan logo dari aplikasi berada di tengah layar. Implementasi halaman *home* dapat dilihat pada Gambar 4.19.



Gambar 4.19. Implementasi halaman *home*

1. **Halaman About**

Implementasi halaman *about* disesuaikan dengan rancangan yang sudah dibuat, pada halaman ini terdapat 2 bagian informasi, bagian sebelah kiri merupakan informasi berupa logo aplikasi, sedangkan bagian kanan berupa informasi-informasi terkait halaman ini. Kemudian, terdapat judul untuk menandakan bahwa halaman ini merupakan halaman *about*. Implementasi halaman *about* dapat dilihat pada Gambar 4.20.



Gambar 4.20. Implementasi halaman *about*

1. **Halaman Video AR**

Halaman video AR terdiri atas judul dan 3 tombol untuk navigasi ke halaman lain. Gambar implementasi halaman video AR dapat dilihat pada Gambar 4.21.

****

Gambar 4.21. Implementasi halaman video AR

1. **Halaman *Filter* AR**

Halaman *filter* AR menampilkan layar kamera depan dengan tujuan untuk menangkap wajah pengguna yang kemudian akan di proses dengan efek AR. Kemudian terdapat gambar logo dari provinsi Indonesia yang berfungsi untuk mengganti efek wajah AR sesuai dengan budaya provinsi tersebut. Dan ada juga 1 tombol yang jika ditekan maka akan melakukan navigasi kembali ke halaman sebelumnya. Gambar implementasi halaman *filter* AR dapat dilihat pada Gambar 4.22.

****

Gambar 4.22. Implementasi halaman filter AR

1. **Halaman *List* 3D *Object***

Implementasi halaman *list* 3D *object* berupa daftar budaya-budaya yang bisa ditekan yang nantinya akan ditampilkan dalam bentuk objek 3D pada halaman *detail* 3D *object*. Terdapat juga judul halaman dan 1 tombol untuk navigasi kembali ke halaman sebelumnya. Gambar implementasi halaman *list* 3D *object* dapat dilihat pada Gambar 4.23.

****

Gambar 4.23. Implementasi halaman *list* 3D *object*

1. **Halaman *References***

Implementasi halaman *references* berupa daftar yang terdapat gambar logo provinsi dan 2 teks referensi yang bisa di tekan. Terdapat juga judul halaman dan 1 tombol untuk navigasi kembali ke halaman sebelumnya. Gambar implementasi halaman *references* dapat dilihat pada Gambar 4.24.

****

Gambar 4.24. Implementasi halaman *references*

1. **Halaman *Detail* 3D *Object***

Implementasi halaman *detail* 3D *object* dapat dilihat pada Gambar 4.25, menunujukkan bahwa objek 3D dari budaya berada di posisi tengah layar. Halaman memiliki 2 tombol, tombol disebelah kiri digunakan untuk navigasi kembali ke halaman sebelumnya. Kemudian tombol disebelah kanan berfungsi untuk menampilkan deskripsi dari budaya yang tampil. Terdapat juga nama budaya berada disebelah kanan tombol kembali.



Gambar 4.25. Implementasi halaman detail 3D *object*

1. **Halaman Tutorial**

Impelementasi halaman tutorial disesuaikan dengan rancangan yang sudah dibuat, pada halaman ini terdiri dari 3 tombol, 1 untuk navigasi dan 2 untuk mengubah tutorial yang tampil. Kemudian, terdapat judul untuk menandakan bahwa halaman ini merupakan halaman tutorial. Implementasi halaman tutorial dapat dilihat pada Gambar 4.26.



Gambar 4.26. Implementasi halaman tutorial

1. **Halaman *Scene***

Implementasi halaman *scene* disesuaikan dengan rancangan yang sudah dibuat, pada halaman ini terdapat 1 tombol untuk navigasi halaman, kemudian terdapat bidang persegi yang digunakan untuk *scanning* logo provinsi. Implementasi halaman *scene* dapat dilihat pada Gambar 4.27.

****

Gambar 4.27. Implementasi halaman *scene*

1. **Pengujian Sistem**

Pengujian sistem yang dilakukan yaitu pengujian metode *marker based tracking*, *black box test* dan *white box test*. Pengujian metode *marker based tracking* dilakukan dengan tujuan untuk mengetahui kualitas dari metode AR yang sudah diimplementasikan pada sistem. Sedangkan *black box test* dan *white box test* dilakukan untuk mengetahui apakah implementasi sistem terkait tampilan dan alur logika program sudah berjalan dengan baik sesuai dengan fungsinya.

1. **Pengujian Metode *Marker Based Tracking* Aplikasi Edukasi Pengenalan Budaya Indonesia**

Pengujian metode *marker based tracking* pada sistem dilakukan pada fitur video AR yang bekerja dengan melakukan pemindaian terhadap logo provinsi Indonesia kemudian akan menampilkan video budaya. Pengujian ini dilakukan untuk mengetahui kualitas dari deteksi *marker* yang ada pada sistem. Bagian bab ini juga menjelaskan indikator pengujian, alat pengujian, skenario pengujian dan hasil pengujian.

1. **Indikator Pengujian**

Indikator-indikator pengujian metode *marker based tracking* sebagai berikut:

1. Pemindaian *marker* dengan jarak pemindaian yang berbeda.
2. Pemindaian *marker* dengan ukuran *marker* yang berbeda.
3. Pemindaian *marker* dengan pencahayaan *marker* yang berbeda.
4. Waktu yang dibutuhkan untuk dapat menampilkan video AR terhadap indikator-indikator diatas.

Indikator pengujian diatas bertujuan sebagai tolak ukur atau batasan pengujian *marker* agar proses pengujian menjadi lebih terukur dan spesifik. Rincian indikator pengujian dapat dilihat pada Tabel 4.1.

Tabel 4.1. Spesifikasi indikator pengujian metode *marker based tracking*

|  |  |  |
| --- | --- | --- |
| **No.** | **Indikator** | **Spesifikasi** |
| 1. | Jarak pendeteksian | * 1-10 cm * 11-20 cm * 21-30 cm |
| 2. | Ukuran *marker* | * Kecil (± 3cm x 3cm) * Sedang (± 10cm x 10cm) * Besar (± 25cm x 25cm) |
| 3. | Pencahayaan/Intensitas Cahaya | * Gelap (0-200 Lx) * Terang (300-500 Lx) |

Tabel 4.2 menunjukkan sampel *marker* yang digunakan dalam pengujian yang sesuai dengan indikator diatas.

Tabel 4.2. Sampel *marker*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Nama** | **Gambar** | **Spesifikasi** |
| 1. | *logo\_dki* | logo_prov_dki | * Resolusi: 256 x 256 *px* * Kualitas Gambar 100% |
| 2. | *logo\_jabar* | logo_prov_jabar | * Resolusi: 800 x 933 *px* * Kualitas Gambar 100% |
| 3. | *logo\_papua* | logo_prov_papua | * Resolusi: 1146 x 1323 *px* * Kualitas Gambar 75% |

1. **Alat Pengujian**

Pengujian metode *marker based tracking* pada sistem menggunakan perangkat *smartphone* dan media pengujiandengan spesifikasi tertentu. Adapun spesifikasi alat pengujian yang digunakan diantaranya:

1. *Smartphone*

Spesifikasi *smartphone* yang digunakan sebagai berikut:

|  |  |  |
| --- | --- | --- |
| * OS | : | Android 11 |
| * RAM | : | 4GB |
| * CPU | : | Octa-core |
| * Main Camera | : | 12MP |

1. Media Pengujian

Media pengujian yang digunakan disesuaikan dengan indikator pengujian yang telah ditentukan sebelumnya, yaitu ukuran *marker* dan pencahayaan. Media *marker* dan pencahayaan yang digunakan sebagai berikut:

|  |  |  |
| --- | --- | --- |
| * Media *marker* | : | Kertas putih dengan ketebalan |
| * Pencahayaan | : | Lampu dengan 7 *watt* (700 lm) |

1. **Skenario Pengujian**

Skenario pengujian yang dilakukan sebagai berikut:

* Pengujian *marker* dilakukan sesuai dengan indikator yang ada, yaitu jarak, ukuran dan pencahayaan.
* Tempat uji coba dilakukan didalam diruangan tertutup dengan tingkat pencahayaan tertentu.
* Hanya menguji pada 1 perangkat dengan spesifikasi yang dijelaskan sebelumnya.
* Jumlah *marker* yang diuji berjumlah 9 gambar (3 gambar dengan ukuran dan gambar yang berbeda).
* Batas waktu maksimum untuk mendeteksi *marker* selama 1 menit.
* Tidak ada batasan posisi *marker* dengan kamera.
* Pengujian dilakukan sebanyak 10x setiap indikator *marker*.

1. **Hasil Pengujian**

Hasil pengujian metode *marker based tracking* dibagi kedalam 6 bagian berdasarkan gambar *marker* dan indikator pencahayaan. Tabel 4.3 - 4.8, menunjukkan hasil pengujian yang telah dilakukan.

Keterangan Tabel:

J = Jarak

W = Waktu (10 x 10 cm)

Lx = Lux

Cm = Sentimeter

Y = Ya/Berhasil

T = Tidak/Gagal

Tabel 4.3. Hasil pengujian metode *marker based tracking* terhadap logo DKI Jakarta (Terang)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logo DKI Jakarta (Terang)** | **Jarak Pengujian (cm)** | | | **Ket. Jarak (cm)** | | **Ket. Pencahayaan (lx)** | **Ket Waktu (s)** |
| **Ukuran *Marker* (cm)** | **1-10** | **11-20** | **21-30 +** | **Min** | **Maks** | **Rata-rata** | |
| **Kecil (3x3)** | T | T | T | - | - | ± 350 | - |
| **Sedang (10x10)** | T | Y | Y | 15 | 40 | ± 350 | ± 3 |
| **Besar (25x25)** | T | T | Y | 29 | 37 | ± 350 | ± 2 |

Tabel 4.4. Hasil pengujian metode *marker based tracking* terhadap logo DKI Jakarta (Gelap)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logo DKI Jakarta (Gelap)** | **Jarak Pengujian (cm)** | | | **Ket. Jarak (cm)** | | **Ket. Pencahayaan (lx)** | **Ket Waktu (s)** |
| **Ukuran *Marker* (cm)** | **1-10** | **11-20** | **21-30 +** | **Min** | **Maks** | **Rata-rata** | |
| **Kecil (3x3)** | T | T | T | - | - | ± 110 | - |
| **Sedang (10x10)** | T | Y | Y | 15 | 32 | ± 110 | ± 2 |
| **Besar (25x25)** | T | T | Y | 32 | 40 | ± 110 | ± 2 |

Tabel 4.5. Hasil pengujian metode *marker based tracking* terhadap logo Jawa Barat (Terang)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logo Jawa Barat (Terang)** | **Jarak Pengujian (cm)** | | | **Ket. Jarak (cm)** | | **Ket. Pencahayaan (lx)** | **Ket Waktu (s)** |
| **Ukuran *Marker* (cm)** | **1-10** | **11-20** | **21-30 +** | **Min** | **Maks** | **Rata-rata** | |
| **Kecil (3x3)** | T | T | T | - | - | ± 110 | - |
| **Sedang (10x10)** | T | Y | Y | 15 | 47 | ± 110 | ± 2 |
| **Besar (25x25)** | T | T | Y | 32 | 41 | ± 110 | ± 2 |

Tabel 4.6. Hasil pengujian metode *marker based tracking* terhadap logo Jawa Barat (Gelap)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logo Jawa Barat (Gelap)** | **Jarak Pengujian (cm)** | | | **Ket. Jarak (cm)** | | **Ket. Pencahayaan (lx)** | **Ket Waktu (s)** |
| **Ukuran *Marker* (cm)** | **1-10** | **11-20** | **21-30 +** | **Min** | **Maks** | **Rata-rata** | |
| **Kecil (3x3)** | T | T | T | - | - | ± 110 | - |
| **Sedang (10x10)** | T | Y | Y | 15 | 32 | ± 110 | ± 2 |
| **Besar (25x25)** | T | T | Y | 39 | 48 | ± 110 | ± 2 |

Tabel 4.7. Hasil pengujian metode *marker based tracking* terhadap logo Papua (Terang)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logo Papua (Terang)** | **Jarak Pengujian (cm)** | | | **Ket. Jarak (cm)** | | **Ket. Pencahayaan (lx)** | **Ket Waktu (s)** |
| **Ukuran *Marker* (cm)** | **1-10** | **11-20** | **21-30 +** | **Min** | **Maks** | **Rata-rata** | |
| **Kecil (3x3)** | T | T | T | - | - | ± 110 | - |
| **Sedang (10x10)** | T | Y | Y | 15 | 32 | ± 110 | ± 3 |
| **Besar (25x25)** | T | T | Y | 32 | 40 | ± 110 | ± 3 |

Tabel 4.8. Hasil pengujian metode *marker based tracking* terhadap logo Papua (Gelap)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logo Papua (Gelap)** | **Jarak Pengujian (cm)** | | | **Ket. Jarak (cm)** | | **Ket. Pencahayaan (lx)** | **Ket Waktu (s)** |
| **Ukuran *Marker* (cm)** | **1-10** | **11-20** | **21-30 +** | **Min** | **Maks** | **Rata-rata** | |
| **Kecil (3x3)** | T | T | T | - | - | ± 110 | - |
| **Sedang (10x10)** | T | Y | Y | 17 | 21 | ± 110 | ± 5 |
| **Besar (25x25)** | T | T | Y | 35 | 41 | ± 110 | ± 4 |

Kesimpulan dari hasil pengujian yang dilakukan adalah dapat diketahui bahwa *marker* berukuran kecil tidak bisa dideteksi/dikenali, sehingga objek *virtual* tidak pernah berhasil untuk ditampilkan.

Jarak antara *marker* dengan perangkat harus disesuaikan dengan ukuran *marker* juga. Karena jika jarak terlalu dekat tetapi ukuran *marker* besar, maka kamera tidak bisa menangkap ukuran keseluruhan *marker* akibatnya objek *virtual* tidak berhasil ditampilkan.

Intensitas cahanya tidak berpengaruh terhadap kualitas *marker* dengan catatan *marker* harus tetap terlihat pada bingkai kamera, tampilan tidak boleh kabur.

Semua hasil pengujian hanya dilakukan dengan 1 perangkat dengan spesifikasi tertentu, jika perangkat yang digunakan lain, memungkinkan untuk mendapatkan hasil yang berbeda.

1. ***Black Box Test* Aplikasi Edukasi Pengenanalan Budaya Indonesia**

Lingkungan pengujian *black box* dilakukan pada perangkat 3 perangkat *android*, 2 perangkat mendukung teknologi ARCore dan yang 1 tidak. Spesifikasi dari perangkat pengujian, dapat dilihat pada Tabel 4.9.

Tabel 4.9. Spesifikasi lingkungan pengujian

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | ***Brand*** | ***Type*** | **OS** | **RAM** | **CPU** | ***Main Camera*** | ***Selfie Camera*** | ***Support ARCore*** |
| 1. | Google | Pixel 2 | Android 11 | 4GB | Octa-core | 12MP | 8MP | Ya |
| 2. | Samsung | Galaxy A21S | Android 11 | 4GB | Octa-core | 48MP | 13MP | Tidak |
| 3. | Samsung | S9+ | Android 11 | 4GB | Octa-core | 12MP | 8MP | Ya |

Berikut ini pengujian *black box* yang dilakukan pada aplikasi pengenalan budaya Indonesia yang diimplementasikan pada setiap halaman yang ada pada sistem, yaitu halaman *splash*, *home*, *about*, video AR, *filter* AR, *list* 3D *object*, *references*, *detail* 3D *object*, tutorial dan *scene*.

1. **Pengujian Halaman *Splash***

Hasil pengujian halaman *splash* dapat dilihat pada Tabel 4.10.

Tabel 4.10. Hasil pengujian halaman *splash*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Aplikasi menampilkan halaman *splash* setelah pengguna membuka aplikasi. | Ketika aplikasi dibuka oleh pengguna, sistem menampilkan halaman *splash*. | [✓]  Sesuai  [ ]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Setelah kurang lebih 2 detik, halaman otomatis akan berpindah ke halaman *home*. | Ketika halaman splash tampil selama kurang lebih 2 detik, sistem melakukan navigasi ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman *Home***

Hasil pengujian halaman *home* dapat dilihat pada Tabel 4.11.

Tabel 4.11. Hasil pengujian halaman *home*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Halaman *home* menampilkan 6 buah tombol. | Pada halaman *home*, terdapat 6 buah tombol, yaitu tombol video AR, *filter* AR, objek 3D , referensi lain, tentang dan keluar. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Halaman video AR tampil setelah tombol video AR diklik. | Ketika tombol video AR pada halaman *home* diklik, maka halaman *home* akan berpindah ke halaman video AR. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai |
| 3. | Tidak ada | Halaman *filter* AR tampil setelah tombol *filter* AR diklik. | Ketika tombol *filter* AR pada halaman *home* diklik *,* maka halaman *home* akan berpindah ke halaman *filter* AR. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 4. | Tidak ada | Halaman *list 3D object* tampil setelah tombol objek 3D diklik. | Ketika tombol objek 3D pada halaman *home* diklik*,* maka halaman *home* akan berpindah ke halaman *list 3D object*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 5. | Tidak ada | Halaman *references* tampil setelah tombol referensi lain diklik. | Ketika tombol referensi lain pada halaman *home* diklik*,* maka halaman *home* akan berpindah ke halaman *references*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 6. | Tidak ada | Halaman *about* tampil setelah tombol tentang diklik. | Ketika tombol tentang pada halaman *home* diklik*,* maka halaman *home* akan berpindah ke halaman *about*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 7. | Tidak ada | Infromasi konfirmasi keluar dari aplikasi tampil setelah tombol keluar diklik. | Ketika tombol keluar pada halaman *home* diklik*,* maka infromasi konfirmasi untuk keluar akan tampil. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 8. | Tidak ada | Infromasi konfirmasi keluar tertutup setelah tombol tidak diklik. | Ketika tombol tidak pada informasi konfirmasi keluar diklik, maka informasi konfirmasi keluar akan tertutup atau tidak tampil. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 9. | Tidak ada | Aplikasi tertutup setelah tombol ya pada infromasi konfirmasi keluar diklik. | Ketika tombol ya pada informasi konfirmasi keluar diklik, maka aplikasi akan tertutup. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman *About***

Hasil pengujian halaman *about* dapat dilihat pada Tabel 4.12.

Tabel 4.12. Hasil pengujian halaman *about*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Informasi tentang aplikasi tampil pada halaman *about*. | Halaman *about* menampilkan informasi tentang aplikasi. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Halaman *home* tampil setelah tombol kembali diklik. | Ketika tombol kembali diklik, maka halaman *about* akan berpindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman Video AR**

Hasil pengujian halaman video AR dapat dilihat pada Tabel 4.13.

Tabel 4.13. Hasil pengujian halaman video *AR*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Halaman video AR menampilkan teks judul dan 3 buah tombol. | Pada halaman video AR terdapat teks judul bertuliskan Video AR dan juga terdapat 3 tombol, yaitu tombol mulai, tutorial dan kembali. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Halaman *home* tampil setelah menekan tombol kembali diklik. | Ketika tombol kembali diklik, maka halaman video AR akan berpindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 3. | Tidak ada | Halaman *scene* tampil setelah tombol mulai diklik. | Ketika tombol mulai pada halaman video AR ditekan, maka halaman video AR akan berpindah ke halaman *scene*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai |
| 4. | Tidak ada | Halaman tutorial tampil setelah tombol mulai diklik. | Ketika tombol tutorial diklik, maka halaman video AR akan berpindah ke halaman tutorial. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman *Filter* AR**

Hasil pengujian halaman *filter* AR dapat dilihat pada Tabel 4.14.

Tabel 4.14. Hasil pengujian halaman *filter* AR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Halaman *filter* AR menampilkan 4 tombol. | Pada halaman *filter* AR terdapat 4 tombol, 1 tombol kembali dan 3 tombol gambar provinsi Indonesia. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Tampil teks disebelah tombol kembali setelah tombol gambar provinsi Indonesia diklik. | Ketika salah satu tombol bergambar provinsi Indonesia diklik, maka akan tampil teks disebelah tombol kembali. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 3. | Wajah manusi | Efek AR diterapkan ke wajah manusia setelah tombol bergambar provinsi Indonesia diklik dan wajah tersebut berada didepan kamera. | Ketika salah satu tombol bergambar provinsi Indonesia diklik, dan menempatkan wajah di depan kamera, efek AR diterapkan ke wajah pengguna jika perangkat mendukung. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ]  Sesuai  [✓]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai |
| 4. | Tidak ada | Halaman *home* tampil setelah pengguna menekan tombol kembali. | Ketika tombol kembali diklik, maka halaman akan pindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman *List* 3D *Object***

Hasil pengujian halaman *list 3D object* dapat dilihat pada Tabel 4.15.

Tabel 4.15. Hasil pengujian halaman *list* 3D *object*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Daftar budaya tampil pada halaman *list* 3D *object*. | Halaman *list* 3D *object* menampilkan daftar budaya. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Halaman *detail* 3D *object* tampil setelah daftar budaya di klik. | Ketika daftar budaya di klik, maka halaman *list* 3D *object* akan berpindah ke halaman *detail* 3D *object*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai |
| 3. | Tidak ada | Halaman *home* tampil setelah tombol kembali diklik. | Ketika tombol kembali diklik, maka halaman *list 3D object* akan berpindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman References**

Hasil pengujian halaman *references* dapat dilihat pada Tabel 4.16.

Tabel 4.16. Hasil pengujian halaman *references*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Tampil daftar referensi budaya provinsi pada halaman *references*. | Halaman *references* menampilkan daftar referensi budaya provinsi. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Halaman referensi tampil setelah daftar referensi di klik. | Ketika daftar referensi di klik, maka halaman referensi akan tampil. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 3. | Tidak ada | Halaman *home* tampil setelah tombol kembali diklik. | Ketika tombol kembali diklik, maka halaman *references* akan berpindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman *Detail* 3D *Object***

Hasil pengujian halaman *detail* 3D *object* dapat dilihat pada Tabel 4.17.

Tabel 4.17. Hasil pengujian halaman *detail* 3D *object*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Tampil objek budaya dalam bentuk 3D dan 2 tombol pada halaman *detail 3D object*. | Pada halaman *detail 3D object* terdapat 2 tombol, tombol kembali dan tombol deskripsi. Tetapi untuk objek budaya 3D akan tampil jika perangkat mendukung. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ]  Sesuai  [✓]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Deskripsi dari budaya tampil setelah tombol deskripsi diklik. | Ketika tombol deskripsi diklik, maka akan menampilkan deskripsi dari budaya. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 3. | Tidak ada | Halaman *list 3D object* tampil setelah tombol kembali diklik. | Ketika tombol kembali diklik, maka halaman akan pindah ke halaman *list 3D object*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman Tutorial**

Hasil pengujian halaman tutorial dapat dilihat pada Tabel 4.18.

Tabel 4.18. Hasil pengujian halaman tutorial

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Terdapat informasi cara menggunakan fitur *augmented reality* pada halaman tutorial. | Halaman tutorial menampilkan informasi terkait cara menggunakan fitur *augmented reality*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai |
| 2. | Tidak ada | Terdapat tombol keluar, kembali dan selanjutnya pada halaman tutorial. | Halaman tutorial menampilkan tombol keluar, kembali dan selanjutnya. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [ ]  Tidak Sesuai | [✓] Sesuai  [] Tidak Sesuai |
| 3. | Tidak ada | Informasi tutorial berubah setelah tombol selanjutnya diklik. | Ketika tombol selanjutnya diklik, maka informasi tutorial berubah. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 4. | Tidak ada | Informasi tutorial berubah setelah tombol kembali diklik. | Ketika tombol kembali diklik, maka informasi tutorial berubah. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |
| 5. | Tidak ada | Halaman *home* tampil setelah tombol kembali diklik. | Ketika tombol keluar diklik, maka halaman tutorial akan berpindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai | [ ✓] Sesuai  [ ]  Tidak Sesuai |

1. **Pengujian Halaman *Scene***

Hasil pengujian halaman *scene* dapat dilihat pada Tabel 4.19.

Tabel 4.19. Hasil pengujian halaman *scene*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Data Masukan** | **Harapan** | **Pengamatan** | **Hasil** | | |
| **Pixel 2** | **Galaxy A21S** | **S9+** |
| 1. | Tidak ada | Tampil tombol kembali dan sebuah bingkai untuk *scanning* pada halaman *scene*. | Halaman *scene* menampilkan tombol kembali dan sebuah bingkai untuk *scanning*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ✓]  Sesuai  [ ]  Tidak Sesuai |
| 2. | Gambar logo provinsi Indonesia. | Video AR tampil setelah *scanning* logo provinsi dilakukan. | Ketika *scanning* logo provinsi Indonesia dilakukan. Maka sebuah video *augmented reality* akan tampil jika perangkat mendukung. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ]  Sesuai  [✓]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai |
| 3. | Tidak ada | Tombol *pause* dan tombol *stop* tampil saat video AR tampil. | Ketika video AR tampil, terdapat tombol *pause/play* dan *stop*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ]  Sesuai  [✓]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai |
| 4. | Tidak ada | *Icon* tombol berubah menjadi *play* dan video AR berhenti. | Ketika tombol *pause* diklik, maka *icon* tombol berubah menjadi *play* dan video berhenti. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ]  Sesuai  [✓]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai |
| 5. | Tidak ada | Video AR tidak tampil dan bingkai *scanning* kembali muncul setelah tombol *stop* diklik. | Ketika tombol *stop* diklik, maka video AR menghilang dan kemudian bingkai *scanning* kembali muncul. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [ ]  Sesuai  [✓]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai |
| 6. | Tidak ada | Halaman *home* tampil setelah tombol kembali diklik. | Ketika tombol keluar diklik, maka halaman *scene* akan berpindah ke halaman *home*. | [ ✓]  Sesuai  [ ]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai | [✓]  Sesuai  [ ]  Tidak Sesuai |

Kesimpulan dari pengujian *black box* di atas, bahwa perangkat yang mendukung teknologi AR sudah berjalan dengan baik, sedangkan untuk yang tidak mendukung teknologi AR semua fungsionalitas berjalan tetapi fitur AR tidak berfungsi.

1. ***White Box Test* Aplikasi Edukasi Pengenalan Budaya Indonesia**

*White box test* dilakukan pada setiap halaman yang ada pada sistem aplikasi. Teknik *white box test* yang dilakukan adalah *basis path*. Kemudian terdapat juga langkah-langkah yang harus dilakukan dalam teknik pengujian *basis path*, yaitu menghitung kompleksitas siklomatis dan penentuan jalur *independent path*.

1. ***Splash White Box Test***

*Splash white box test* dapat dilihat pada gambar Gambar 4.28 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| fc-splash | path-splash |

Gambar 4.28. *Splash white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 2

Jumlah busur (E) = 1

Jumlah region (R) = 1

Jumlah predikat (P) = 0

Jumlah region = 1

*V(G) = E - N + 2* = 1 - 2 + 2 = 1

*V(G) = P + 1* = 0 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* = 1, 2

Sehingga banyak jalur adalah 1

Kesimpulan dari *splash white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***Home White Box Test***

*Home white box test* dapat dilihat pada gambar Gambar 4.29 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_home_video_ar | path_home_video_ar |

Gambar 4.29. *Home white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 3

Jumlah busur (E) = 2

Jumlah region (R) = 1

Jumlah predikat (P) = 0

Jumlah region = 1

*V(G) = E - N + 2* = 2 - 3 + 2 = 1

*V(G) = P + 1* = 0 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* = 1, 2, 3

Sehingga banyak jalur adalah 1

Kesimpulan dari *home white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***About White Box Test***

*About white box test* dapat dilihat pada gambar Gambar 4.30 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_about | path_about |

Gambar 4.30. *About white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 3

Jumlah busur (E) = 2

Jumlah region (R) = 1

Jumlah predikat (P) = 0

Jumlah region = 1

*V(G) = E - N + 2* = 2 - 3 + 2 = 1

*V(G) = P + 1* = 0 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* = 1, 2, 3

Sehingga banyak jalur berjumlah 1

Kesimpulan dari *about white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. **Video AR *White Box Test***

Video AR *white box test* dapat dilihat pada gambar Gambar 4.31 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_home_video_ar | path_home_video_ar |

Gambar 4.31. Video AR *white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 3

Jumlah busur (E) = 2

Jumlah region (R) = 1

Jumlah predikat (P) = 0

Jumlah region = 1

*V(G) = E - N + 2* = 2 - 3 + 2 = 1

*V(G) = P + 1* = 0 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* = 1, 2, 3

Sehingga banyak jalur berjumlah 1

Kesimpulan dari *video* AR *white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***Filter* AR *Box Test***

*Filter* AR *white box test* dapat dilihat pada gambar Gambar 4.32 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_filter | path_filter |

Gambar 4.32. *Filter* AR *white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 5

Jumlah busur (E) = 5

Jumlah region (R) = 2

Jumlah predikat (P) = 1

Jumlah region = 2

*V(G) = E - N + 2* = 5 - 5 + 2 = 2

*V(G) = P + 1* = 1 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* 1= 1, 2, 3, 4, 5

*Path* 2= 1, 2, 3, 1, 2, 3, 4, 5

Sehingga banyak jalur berjumlah 2

Kesimpulan dari *filter* AR *white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***List* 3D *Object White Box Test***

*List 3D object white box test* dapat dilihat pada gambar Gambar 4.33 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_list | path_home_video_ar |

Gambar 4.33. *List 3D object white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 3

Jumlah busur (E) = 2

Jumlah region (R) = 1

Jumlah predikat (P) = 0

Jumlah region = 1

*V(G) = E - N + 2* = 2 - 3 + 2 = 1

V(G) = P + 1 = 0 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* = 1, 2, 3

Sehingga banyak jalur berjumlah 1

Kesimpulan dari *list 3D object white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***References White Box Test***

*References white box test* dapat dilihat pada gambar Gambar 4.34 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_list | path_home_video_ar |

Gambar 4.34. *References white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 3

Jumlah busur (E) = 2

Jumlah region (R) = 1

Jumlah predikat (P) = 0

Jumlah region = 1

*V(G) = E - N + 2* = 2 - 3 + 2 = 1

*V(G) = P + 1* = 0 + 1 = 1

1. Menentukan Jalur *Independent Path*

*Path* = 1, 2, 3

Sehingga banyak jalur berjumlah 1.

Kesimpulan dari *references white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***Detail* 3D *Object White Box Test***

*Detail 3D object white box test* dapat dilihat pada gambar Gambar 4.35 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_detail_3d | path-tutorial |

Gambar 4.35. *Detail 3D object white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) = 5

Jumlah busur (E) = 5

Jumlah region (R) = 2

Jumlah predikat (P) = 1

Jumlah region = 2

*V(G) = E - N + 2* = 5 - 5 + 2 = 2

*V(G) = P + 1* = 1 + 1 = 2

1. Menentukan Jalur *Independent Path*

*Path* 1= 1, 2, 3, 5

*Path* 2= 1, 2, 4, 5

Sehingga banyak jalur berjumlah 2

Kesimpulan dari *detail 3D object white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. **Tutorial *White Box Test***

*Tutorial white box test* dapat dilihat pada gambar Gambar 4.36 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_tutorial | path-tutorial |

Gambar 4.36. Tutorial *white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) =5

Jumlah busur (E) = 5

Jumlah region (R) = 2

Jumlah predikat (P) = 1

Jumlah region =2

*V(G) = E - N + 2* = 5 - 5 + 2 = 2

*V(G) = P + 1* = 1 + 1 = 2

1. Menentukan Jalur *Independent Path*

*Path* 1 = 1, 2, 3, 5

*Path* 2 = 1, 2, 4, 5

Sehingga banyak jalur berjumlah 2

Kesimpulan dari *tutorial white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

1. ***Scene White Box Test***

*Scene white box test* dapat dilihat pada gambar Gambar 4.37 yang terdiri atas *flowchart* berada di sisi sebelah kiri dan *flow graph* berada di sisi sebelah kanan.

|  |  |
| --- | --- |
| flow_scene | path_scene |

Gambar 4.37. *Scene white box test*

1. Menghitung Kompleksitas Siklomatis

Jumlah simpul (N) =5

Jumlah busur (E) = 5

Jumlah region (R) = 2

Jumlah predikat (P) = 1

Jumlah region = 2

*V(G) = E - N + 2* = 5 - 5 + 2 = 2

*V(G) = P + 1* = 1 + 1 = 2

1. Menentukan Jalur *Independent Path*

*Path* 1 = 1, 2, 3, 4, 5

*Path* 2 = 1, 2, 3, 1, 2, 3, 4, 5

Sehingga banyak jalur berjumlah 2

Kesimpulan dari *scene white box test* adalah jalur yang dilalui untuk mengerjakan seluruh *statement* program sebanyak 1 jalur.

**BAB 5**

**KESIMPULAN**

1. **Kesimpulan**

Sistem aplikasi edukasi pengenalan budaya Indonesia berhasil diimpelemtasikan dengan baik pada perangkat *android* yang mendukung teknologi ARCore.

1. **Saran**

Saran yang dapat diberikan untuk pengembangan lebih lanjut terkait sistem rancang bangun aplikasi kebudayaan menggunakan teknologi augmented reality diantaranya:

1. Sebaiknya menggunakan kualitas gambar dan resolusi yang sudah dianjurkan untuk *marker*, yaitu kualitas gambar 100% dan resolusi minimal 300x300 *px*, agar proses pemindaian *marker* lebih cepat dan efisien.
2. Perhatikan spesifikasi perangkat, karena penggunaan pustaka ARCore hanya berjalan pada perangkat tertentu. Solusinya, bisa menonaktifkan fitur ARCore jika ingin mencakup pengguna yang lebih banyak, atau menggunakan pustaka AR lain.
3. Memperbanyak jumlah provinsi Indonesia yang diperkenalkan dan fitur untuk menambah pengetahuan lainnya, seperti *quiz*, pembelajaran bahasa dan lainnya.
4. Perhatikan kebutuhan pengguna aplikasi, apakah fitur AR memang diperlukan atau tidak. Karena bagaimanapun teknologi AR membutuhkan sumber daya yang tidak sedikit.