

BAB I

PENDAHULUAN

A. Latar Belakang

Bahasa isyarat menurut KBBI yaitu bahasa yang tidak menggunakan bunyi ucapan manusia atau tulisan dalam sistem perlambangannya, bahasa yang menggunakan isyarat (gerakan tangan, kepala, badan, dsb), khusus diciptakan untuk tunarungu, tunawicara, tunanetra, dsb (Kamus Besar Bahasa Indonesia Online, 2017).

Bahasa Isyarat merupakan bahasa yang menggunakan bahasa tubuh, gerak bibir dan ekspresi wajah sehingga tidak menggunakan suara atau bunyi untuk berkomunikasi dan mengungkapkan pikiran mereka terhadap lingkungan sekitar. Bahasa isyarat menjadi alat komunikasi yang efektif karena penyandang tunarungu dan tunawicara kekurangan dalam kemampuan berbicara.

Dalam Bahasa isyarat, setiap gerakan sudah menegaskan makna, dan aturan yang kuat dari konteks dan tata bahasa dapat diterapkan untuk membuat pengakuan penurut. Di Indonesia bahasa isyarat yang sudah dibakukan pemerintah adalah Sistem Isyarat Bahasa Indonesia (SIBI). Sedangkan bahasa isyarat alami merupakan bahasa isyarat alami yang digunakan oleh kaum tunarungu dengan kaum tunarungu, berkembang secara alami dan disepakati antar pemakai. Di Indonesia bahasa isyarat alami disebut Bahasa Isyarat Indonesia.

B. Rumusan Masalah

Berdasarkan latar belakang di atas maka didapatkan rumusan masalahnya adalah “Implementasi *Augmented Reality* pada pengenalan bahasa isyarat Bahasa Indonesia Berbasis Android?

C. Batasan Masalah

Adapun batasan masalah pada sistem ini adalah

1. Bagaimana hasil penerapan aplikasi *Augmented Reality* pada pengenalan bahasa Isyarat Indonesia yang menggunakan metode *markerless*?
2. Bagaimana sistem kerja *markerless* sehingga dapat menampilkan objek 3D di *smartphone* android?

D. Tujuan Penelitian

Dengan adanya aplikasi *Augmented Reality* (AR) pada pengenalan bahasa isyarat menggunakan metode Bahasa Isyarat Indonesia orang normal, penyandang tunarungu dan tunawicara dapat lebih cepat mengerti dan mudah memahami bahasa isyarat karena menggunakan objek 3 Dimensi (3D).

E. Manfaat Penelitian

Manfaat yang diharapkan dengan adanya media informasi Bahasa isyarat berbasis *Augmented Reality* yaitu:

1. Bagi Mahasiswa
 - a. Sebagai referensi pengenalan Bahasa isyarat selanjutnya berbasis *Augmented Reality*.

- b. Mengetahui cara menerapkan gambar 3 dimensi dalam sebuah marker seperti Bahasa isyarat berbasis *Augmented Reality*.

2. Bagi Kampus

- a. Sebagai referensi untuk meningkatkan informasi terkait Bahasa isyarat kepada masyarakat.
- b. Mendapatkan alternatif baru dalam memperkenalkan dan memperkenalkan Bahasa isyarat.

3. Bagi Masyarakat

- a. Meningkatkan minat masyarakat untuk mengenal Bahasa isyarat yang di lengkapi dengan teknologi.
- b. Memberikan informasi yang belum ada pada masyarakat terkait Bahasa isyarat.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Penelitian relevan berisikan penelitian yang terdahulu kemudian dicari bahan penelitian yang ada dan relevan dengan penelitian ini karena sangat berguna untuk mendukung kajian teori yang dikemukakan, sehingga dapat digunakan sebagai landasan pada penyusunan kerangka pikir. Beberapa hasil penelitian yang relevan dengan penelitian ini adalah penelitian yang dilakukan oleh:

1. Leonardi dan Hermawan (2019) dalam penelitiannya yang berjudul "Penerapan *Augmented Reality* Pada Brosur Interior Berbasis Multi *Markerless* Pada *Android*" manfaat dari penerapan *Augmented Reality* pada brosur interior adalah pelanggan dapat melihat dengan jelas barang-barang yang ingin di beli karena gambar yang ada dalam brosur bisa dilihat dalam bentuk 3 dimensi. Dalam penelitian ini, pengembangan aplikasi menggunakan metode *Rational Unified Process (RUP)*. Hasil yang didapat dari penelitian ini adalah dapat membuat aplikasi *Android* untuk brosur interior yang dapat menampilkan barang-barang interior seperti meja, kursi dan lain-lain. Pada pembuatan gambar 3 dimensi peneliti tidak membuat objek 3 dimensinya karena objek yang berupa barang-barang interior telah dibuat oleh CV Wijaya Lesmana Sejahtera, sehingga penulis tidak harus membuat model 3D lagi tetapi tinggal di *export* dari *archicad*.

2. Nadira, Sujaini dan Pratiwi (2016) dalam penelitiannya yang berjudul “Implementasi *Augmented Reality* Pada Brosur Teknik Informatika Universitas Tanjungpura Menggunakan Metode *Marker* ” saat ini brosur Teknik Informatika Universitas Tanjungpura masih berisi gambar-gambar dari kampus itu sendiri, tetapi gambar-gambar tersebut masih kurang memberikan informasi yang jelas sehingga calon mahasiswa sulit untuk membayangkan gambaran dari kampus Teknik Informatika Universitas Tanjungpura dan akhirnya calon mahasiswa tidak terlalu tertarik untuk memilih Teknik Informatika Universitas Tanjungpura. Manfaat dari penerapan *Augmented Reality* pada brosur Teknik Informatika Universitas Tanjungpura yaitu dapat memberikan gambaran yang jelas karena dilengkapi dengan gambar bangunan kampus dalam bentuk 3 dimensi. Dalam penelitian ini menggunakan metode *Marker Based Tracking*. Hasil yang didapat dari penelitian ini adalah peneliti menguji aplikasi yang dibuat dan hasilnya berjalan lancar pada *handphone android* dan pengguna menyatakan setuju bahwa aplikasi *Augmented Reality* brosur dapat memperjelas informasi pada brosur Teknik Informatika Universitas Tanjungpura.
3. Hermawan dan Hariadi (2015) dalam penelitiannya yang berjudul “Pemanfaatan *Augmented Reality* Sebagai Media Informasi Kampus Menggunakan Brosur” Manfaat dari penerapan *Augmented Reality* pada brosur kampus yaitu pembaca dapat melihat bagian brosur *Augmented Reality* dengan detail serta lingkungan pada brosur juga akan terasa lebih hidup dengan adanya animasi pendukung seperti gedung, pohon, dan lain

sebagainya. Metode penelitian yang akan digunakan dalam pembuatan brosur berbasis *Augmented Reality* yaitu terdiri dari empat proses yaitu perencanaan, pengumpulan data, analisis dan dokumentasi hasil yang telah dibuat. Hasil yang didapat dari penelitian ini adalah aplikasi berjalan lancar dan telah diuji menggunakan komputer dan dapat menampilkan objek 3 dimensi pada brosur yang dapat memberikan informasi yang belum ada pada brosur sebelumnya.

B. Bahasa Isyarat

Widyatmoko (2020) Bahasa Isyarat adalah bahasa yang mengutamakan komunikasi manual, bahasa tubuh, dan gerak bibir, bukannya suara, untuk berkomunikasi. Kaum tunarungu adalah kelompok utama yang menggunakan bahasa ini, biasanya dengan mengkombinasikan bentuk tangan, orientasi dan gerak tangan, lengan, dan tubuh, serta ekspresi wajah untuk mengungkapkan pikiran mereka.

Bertentangan dengan pendapat banyak orang, pada kenyataannya belum ada bahasa isyarat internasional yang sukses diterapkan. Bahasa isyarat unik dalam jenisnya di setiap negara. Bahasa isyarat bisa saja berbeda di negara-negara yang berbahasa sama.

Di Indonesia ada dua bahasa isyarat yang digunakan yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Dengan letak perbedaan yaitu SIBI merupakan bahasa isyarat yang diciptakan oleh Alm. Anton Widyatmoko mantan kepala sekolah SLB Widya Bakti Semarang bekerjasama dengan mantan kepala sekolah SLB di Jakarta dan Surabaya tanpa melalui

musyawarah dan persetujuan dari Gerakan Kesejahteraan Tunarungu Indonesai (GERKATIN) yang pada akhirnya mengeluarkan sebuah produk kamus bernama SIBI. SIBI tidak dapat digunakan dalam komunikasi sehari-hari penyandang tunarungu karena penerapan kosakata yang tidak sesuai dengan aspirasi dan nurani kaum tunarungu, terlebih penerapan bahasa yang terlalu baku dengan tata bahasa kalimat bahasa Indonesia yang membuat kesulitan kaum tunarungu untuk berkomunikasi. Kemudian dalam SIBI ditemukan banyak pengaruh alami, budaya, dan isyarat tunarungu dari luar negeri yang sulit dimengerti sehingga memang benar SIBI sulit dipergunakan oleh kaum tunarungu untuk berkomunikasi.

Berbeda dengan Bahasa Isyarat Indonesia (BISINDO) yang belakangan ini diperjuangkan oleh Gerakan Kesejahteraan Tunarungu Indonesia (GERKATIN). BISINDO merupakan bahasa isyarat alami budaya asli Indonesia yang mudah dapat digunakan dalam pergaulan isyarat kaum tunarungu sehari-hari.

Kalau perbedaan mendasar antara SIBI dan BISINDO adalah SIBI menggunakan abjad sebagai panduan bahasa isyarat tangan satu, sementara BISINDO menggunakan gerakan tangan dua tangan sebagai upaya komunikasi antar pengguna bahasa isyarat dapat dilihat gerak tangan untuk kalimat Bahasa Isyarat Indonesia (BISINDO).

C. Smartphone

Andi widi (2021) Ponsel cerdas (bahasa Inggris: *smartphone*) adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti

ponsel cerdas. Bagi beberapa orang, ponsel cerdas merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, ponsel cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat elektronik), internet dan kemampuan membaca buku *elektronik* (*e-book*) atau terdapat papan ketik (baik sebagaimana jadi maupun dihubungkan) dan penyambung VGA. Dengan kata lain, ponsel cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon.

Singkatnya, *smartphone* adalah perangkat yang memungkinkan anda melakukan panggilan telepon, sekaligus memiliki fitur yang dimasa lalu hanya bisa ditemukan pada personal *digital assistant* (PDA) atau komputer- seperti kemampuan untuk mengirim dan menerima *e-mail* dan editing dokumen.

Ponsel terutama digunakan untuk menelpon, sementara PDA digunakan sebagai semacam asistensi digital pribadi atau digital organizer. PDA dapat menyimpan info kontak dan agenda harian serta bisa disinkronisasi dengan komputer. Perkembangan ini lantas mendorong terciptanya apa yang kini dikenal sebagai *smartphone*.

D. Bahasa Pemrograman C#

C# (baca: *C sharp* atau *see sharp*) sering dianggap sebagai bahasa penerus C++ atau versi canggih dari C++, karena ada anggapan bahwa tanda # adalah perpaduan dari 4 buah tanda tambah yang disusun sedemikian rupa sehingga membentuk tanda pagar. Akan tetapi, terlepas dari benar tidaknya anggapan tersebut, C# adalah sebuah bahasa pemrograman yang sangat menjanjikan. C#

adalah sebuah bahasa pemrograman yang berorientasi pada objek yang dikembangkan oleh *Microsoft* dan menjadi salah satu bahasa pemrograman yang mendukung *.Net programming* melalui *Visual Studio*. Andi widi (2021)

C# didasarkan pada bahasa pemrograman C++, C# juga memiliki kemiripan dengan beberapa bahasa pemrograman seperti *Visual Basic*, *Java*, *Delphi*, dan tentu saja C++. C# memiliki kemudahan *syntax* (cara penulisan) seperti *Visual Basic*, dan tentu saja ketangguhan seperti *Java* dan C++. Kemiripan-kemiripan ini tentunya memudahkan programmer dari berbagai latar belakang bahasa pemrograman tidak perlu waktu yang lama untuk menguasainya, karena bagaimanapun juga C# lebih sederhana dibandingkan bahasa-bahasa pemrograman seperti C++ dan *Java*.

C# didesain oleh program *designer* dari *microsoft*, *Anders Hajlsberg*. Sebelum bekerja pada *Microsoft*, Anders bekerja di *Borland*, tempat dia menulis *Pascal compiler*. Sebelum mengembangkan C# Anders pernah mengembangkan J++ untuk *Microsoft*. Setelah itu Anders mengembangkan C# dan *Common Language Runtime* yang merupakan mesin *Virtual dan Runtime library* yang merupakan salah satu tiang utama teknologi. *.Net Anders* mengetahui berbagai macam kekurangan pada bahasa C++, *Delphi*, *Java*, dan *Smalltalk*, karena itu Anders menciptakan bahasa C# yang lebih tangguh. Hal ini juga menjelaskan mengapa C# memiliki kemiripan dengan beberapa bahasa tersebut.

C# bergantung pada CLR yang juga merupakan sumber library bagi program. Semua program C# memerlukan CLR (berarti juga memerlukan *.Net Framework*) untuk dapat dijalankan. Sama halnya dengan *Visual Basic 6* yang

memerlukan *runtime library* tertentu untuk dapat dijalankan. Bahasa C# dapat digunakan untuk menciptakan aplikasi *windows*, *console*, *class* yang dapat digunakan kembali, dan aplikasi *web*. Andi widi (2021)

Ada beberapa alasan kuat yang mendasari pemilihan bahasa C# untuk mengembangkan aplikasi-aplikasi, yaitu:

a. C# benar-benar berorientasi objek

C# adalah bahasa yang benar-benar objek *oriented*. Ini dapat dilihat dari kemampuan C# dalam pembentukan objek, *class*, melakukan *encapsulation*, *inheritance*, dan *polymorphism* dengan mudah.

b. C# sangat sederhana

Bahasa C# bersifat sederhana karena didasarkan pada bahasa C dan C++ bahkan bahasa Java. Tetapi C# lebih sederhana dari bahasa-bahasa tersebut karena C# dibuat dengan menghilangkan kelemahan-kelemahan dari bahasa-bahasa yang mendasarinya.

c. Mampu membuat berbagai aplikasi

Dengan C# kita dapat membuat berbagai macam aplikasi, mulai dari aplikasi *console*, pengolahan kata, *form web*, dan lain-lain.

d. Efisien

Bahasa C# merupakan bahasa yang mengandalkan library yang sangat lengkap, karena itu bahasa C# hanya memiliki sedikit *keywords*. Jadi para pengembang dapat mengingatnya dan memahami kegunaanya dengan baik.

e. C# bersifat modular

Bahasa C# sangat modular, tiap *class* disimpan dalam *namespace* yang dapat dimanfaatkan kembali oleh program lain yang membutuhkannya.

E. Android (SDK)

Android SDK (*Software Development Kit*) adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan suatu aplikasi pada *platform android* menggunakan bahasa pemrograman java. Andi widi (2021)

Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang direlease oleh *google*. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* aplikasi netral, Android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan aplikasi bawaan *Handphone/smartphone*. Beberapa fitur Android adalah :

- a. *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*
- b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile*
- c. *Integrated browser* berdasarkan *engine open source Webkit*
- d. *Grafis* yang dioptimalkan dan didukung oleh *libraries grafis 2D, grafis 3D* berdasarkan spesifikasi *opengl ES 1,0*
- e. SQLite untuk penyimpanan data/*database*
- f. *Media suport* yang mendukung audio, video, dan gambar
- g. *Bluetooth, EDGE,3G,WiFi (tergantung hardware)*
- h. Kamera, GPS,kompas dan *accelerometer* (tergantung hardware)

Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori dan *plugin* untuk IDE *eclipse*.

F. JDK (Java Development Kit)

Java Development Kit merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Environment*.

Java Development Kit wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java. Namun *Java Development Kit* tidak wajib terinstall dikomputer yang akan menjalankan aplikasi yang dibangun menggunakan Java.

Seperti namanya, JDK adalah semacam kotak peralatan (kit) yang digunakan untuk *Development*. JDK ini berguna saat anda menulis *code* program. Seperti halnya JRE, JDK juga memiliki JVM di dalamnya.

- a. Java yaitu program untuk meng-*compile* kode sumber
- b. Java, yaitu program untuk meluncurkan aplikasi.

G. Unity

Unity ini adalah sebuah aplikasi berbasis *Multi Platform*, *Multi Platform* adalah aplikasi yang dapat mempublish ke banyak format tipe *file*, contohnya : *exe*, *apk*, dan lain-lain. *Unity* perlu lisensi untuk dapat dipublish ke *platform* tertentu. Tetapi *Unity* menyediakan untuk *free user* dan bisa di *publish* dalam bentuk *Standalone (.exe)* dan *web*. Untuk saat ini *Unity* sedang mengembangkan

berbasis AR (*Augmented Reality*). *Unity* ini tidak hanya untuk membuat game 3D saja, namun 2D juga.

Unity Technologies dibangun pada tahun (2019) oleh *David Helgason* (CEO), *Nicholas Francis* (CCO), dan *Joachim Ante* (CTO) di *Copenhagen*, Denmark setelah game pertama mereka *GooBall*, gagal lagi dalam meraih sukses. Ketiganya menyadari nilai sebuah *engine* dan *tool* dalam sebuah pengembangan game dan berencana untuk membuat sebuah *engine* yang dapat digunakan oleh semua dengan harga terjangkau. *Unity Technologies* mendapat bantuan dana dari *Sequoia Capital*, *WestSummit Capital*, and *iGlobe Partners*.

Kesuksesan *Unity* terletak pada fokus mereka untuk memenuhi kebutuhan *indie developer* yang tidak dapat membangun game engine mereka sendiri atau membeli lisensi *game engine* yang terlalu mahal. Fokus perusahaan ini adalah “*Democratize game development*” atau diterjemahkan sebagai “Demokrasi Pembangunan Game” dan membuat sebuah pembangunan game baik 2D maupun 3D bisa dicapai oleh banyak orang. Oleh siapa pun. Pada tahun 2008, *Unity* melihat kebangkitan *iPhone* dan menjadi *game engine* pertama yang melakukan dukungan penuh pada *platform* tersebut. *Unity* sekarang di gunakan oleh 53.1% *developers* (termasuk *mobile game developer*) dengan ratusan game yang dirilis baik untuk *iOS* maupun *Android*. Pada tahun 2009, *Unity* mulai meluncurkan produk mereka secara gratis. Jumlah *developer* yang mendaftar melonjak drastis sejak pengumuman tersebut. Pada April 2012, *Unity* mencapai popularitas yang sangat tinggi dengan lebih dari 1 juta *developer*.

Unity sangat mampu melihat berbagai peluang dan perubahan. Hal inilah yang menjadikannya sebagai *game engine* “termurah” yang paling banyak digunakan oleh seluruh orang di dunia. Ingat, *Unity* bisa digunakan untuk perorangan dan tidak selalu harus digunakan oleh sebuah *studio game* yang berjumlah ratusan orang. Tampaknya, demokrasi yang diusung sebagai slogan *Unity* memang benar adanya. Apalagi semenjak dirilisnya *Unity* dengan lisensi free.

H. Marker

Marker adalah object dunia nyata yang sudah di inisialisasi sebelumnya, yang nantinya akan di kenali oleh *system* sebagai wadah untuk memunculkan *object* atau area interaksi. Dalam pembuatan marker dalam hal ini markerless diperlukan sebuah file gambar.JPG yang nantinya akan di *upload* ke *Vuforia*, marker yang telah di-*upload* akan dinilai kualitasnya oleh *system*. Berikut adalah metode dan alur pengenalan pola gambar (Mario Fernando, 2013).

1. Metode Pengenalan Pola Gambar

Qualcomm sebagai salah satu pengembang *Augmented Reality* melakukan proses pendeteksian marker menggunakan pengenalan pola gambar. Metode yang digunakan dalam QCAR adalah *Natural Features Tracking* dengan metode *FAST Corner Detection* yaitu pendeteksian dengan mencari titik-titik (*interest point*) atau sudut-sudut (*corner*) pada suatu gambar. Istilah *corner* dan *interest point* sering digunakan secara bergantian. Pertama-tama dilakukan pendeteksian tepi (*edge*), keudian dilakukan analisa tepi untuk mendapatkan pendeteksian sudut (*corner*) secara tepat. *Algoritma* ini kemudian dikembangkan, sehingga deteksi

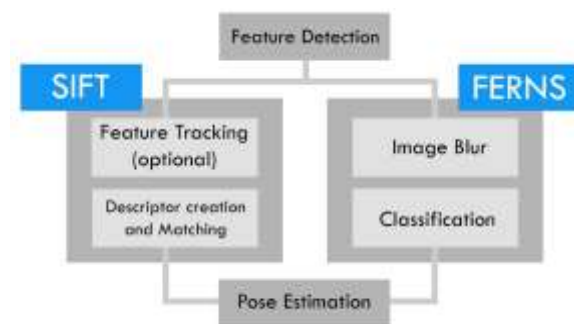
tepi secara *eksplisit* tidak lagi diperlukan. Misalnya mendeteksi kelengkungan dalam gradient gambar. Pada saat itu juga ternyata bagian-bagian yang tidak berbentuk sudut (*corner*) terdeteksi juga sebagai bagian dari gambar, misalnya titik-titik kecil pada latar belakang gelap mungkin terdeteksi. Titik – titik ini yang disebut interest point namun istilah corner tetap digunakan.

2. *Natural Feature Tracking and Detection*

Ponsel adalah *platform* kinerja rendah dengan sumber daya yang sangat terbatas dibandingkn dengan komputer. Akibatnya, keterbatasan ponsel dalam setiap aspek harus diperhitungkan ketika mengembangkan sebuah teknologi AR. Banyak teknologi AR yang dirancang untuk ponsel, menggunakan penanda berbasis pelacakan untuk menimbulkan estimasi dan ekstraksi. Pendekatan ini bekerja dengan cukup baik dan menyederhanakan proses pelacakan, sehingga bagi pengguna akhir, lingkungan harus dipersiapkan terlebih dahulu. Hal ini membuat prosedur pengembang menjadi rumit tetapi membawa fleksibilitas sehingga setiap benda dengan tekstur yang cukup rinci dapat dengan mudah dilacak.

Dalam metode ini informasi yang diperlukan untuk tujuan pelacakan dapat diperoleh dengan cara *optical-flow* berbasis pencocokan template atau korespondensi fitur. *Optical flow* atau aliran *optic* adalah pola gerakan jelas suatu benda, permukaan, dan tepi dalam adegan visual yang disebabkan oleh gerakan relatif antara pengamat (mata atau kamera) dan adegan. Korespondensi fitur bekerja lebih baik dan lebih efektif daripada pencocokan template karena mereka bergantung pada pencocokan fitur local. Mengingat korespondensi tersebut, pose secara kasar dapat dihitung dengan estimasi yang kuat yang membuatnya cukup

sensitive terhadap oklusi persial, blur, refleksi, perubahan skala, kemiringan, perubahan iluminasi atau kesalahan pencocokan. Salah satu unsur diterapkan pendekatan pelacakan fitur alami didasarkan pada versi modifikasi dari fitur *SIFT* dan *FERN descriptor*. *SIFT* sangat baik dalam mengekstrak tetapi prosesor intensif bekerja karena komputasi, sementara *FERN* menggunakan klasifikasi fitur yang cepat tetapi membutuhkan kapasitas memori yang besar. Dalam hal ini pelaksanaan *SIFT* dan *FERN* telah terintegrasi, tetapi dengan signifikan modifikasi untuk membuat sebuah system pelacakan cocok untuk ponsel.



Gambar 2. 1 Alur *SIFT* dan *FERNs*

I. Vuforia SDK (Software Development Kit)

Vuforia SDK adalah *software development Kit* berbasis AR yang menggunakan layar perangkat *mobile* sebagai “Lensa ajaib” atau kaca untuk melihat kedalam dunia *augmented Reality* dimana dunia nyata dan virtual muncul berdampingan. Aplikasi ini membuat *preview* kamera secara langsung pada layar *smartphone* untuk mewakili pandangan dari dunia fisik. *Object* 3 dimensi akan nampak secara langsung di layar *smartphone*, sehingga akan terlihat *object* 3 dimensi berada didalam dunia nyata.(Febriyan M.Fajar,2014)

Vuforia SDK mendukung berbagai 2D dan 3D jenis sasaran termasuk *markerless Image Target*, konfigurasi *3D Multi-Target*, dan bentuk beralamat *Fidusia Marker* dikenal sebagai *Bingkai Marker* . Fitur tambahan dari SDK termasuk lokal *Occlusion Deteksi* menggunakan *Virtual Buttons*, pemilihan target gambar runtime, dan kemampuan untuk membuat dan mengkonfigurasi ulang sasaran set pemrograman saat *runtime*. *Vuforia* menyediakan Antarmuka *Application Programming (API)* di *C++*, *Java*, *Objective-C* , dan bahasa *Net* . Melalui perpanjangan untuk mesin permainan *Unity*. Dengan cara ini , SDK mendukung pengembangan asli untuk *iOS* dan *Android* sementara juga memungkinkan pengembangan aplikasi AR di *Unity* yang mudah dibawa ke kedua *platform* . Aplikasi AR yang dikembangkan menggunakan *Vuforia* karena itu kompatibel dengan berbagai perangkat *mobile* termasuk *iPhone (4/4S)*, *iPad*, dan ponsel *Android* dan *tablet* yang menjalankan *OS Android* versi 2.2 atau yang lebih besar dan *ARMv6* atau 7 prosesor dengan *FPU (Floating Point Unit)* kemampuan pemrosesan.

Vuforia SDK berisi banyak fitur menarik, berhubungan dengan AR dunia:

1. **QR code mengakui:** Perangkat lunak ini dapat memindai dan mengenali QR code.
2. **Gambar mengenali:** *Vuforia* SDK dapat memindai dan mengenali *Images*, ada beberapa keterbatasan kemampuan memindai nya, misalnya, sudut atau bentuk melengkung dapat lebih sulit untuk mengenali, dengan cara yang sama warna yang dibuat menggunakan nuansa yang sama dari warna yang sama. Namun *software* ini benar-benar kuat.

3. **Quadrilaterals Kecil objek** : Perangkat lunak ini juga mampu mengenali benda kecil, menggunakan wajah mereka sebagai pembuat.
4. **Kecil objek silinder**: Salah satu rilis terbaru dari perangkat lunak memperkenalkan kemungkinan menggunakan benda *silinder* kecil sebagai penanda. Saat ini aku tidak bereksperimen namun dengan fitur ini dan aku tidak yakin bagaimana kuat dan dapat diandalkan itu, tapi saya akan mencoba untuk yakin.
5. **Kata pengakuan** : *Vuforia* SDK juga mampu mengenali kata *Inggris* tertentu. Saya menguji fitur ini banyak, dan itu sangat menarik, ada beberapa batasan tentang kaligrafi, tetapi jika kata tersebut dicetak atau digital di layar komputer *software* tidak menunjukkan masalah.
6. **Cloud penyimpanan** : Perangkat lunak ini juga memungkinkan penggunaan data awan disimpan. Fitur ini bisa sangat berguna bagi orang-orang yang tidak memiliki masalah kinerja pada ponsel mereka.
7. **Multi-target** : *Vuforia* dapat memindai dan mengenali lebih dari satu target sekaligus, biasanya ada batasan sampai 5 target sekaligus, tapi *software* ini sangat lincah dan itu tergantung pada jenis perangkat keras yang digunakan.
8. **Tombol virtual** : perangkat lunak memiliki *build-in* maya manajemen tombol. Fitur ini bisa sangat menarik untuk menambahkan lebih banyak interaksi antara pemain dan spidol.

J. Augmented Reality dan videoplayback

Augmented reality adalah sebuah teknologi yang menggabungkan benda maya berbentuk dua dimensi dan bisa juga tiga dimensi ke dalam sebuah

lingkungan nyata lalu memproyeksikannya sebagai realitas dalam waktu nyata. Nantinya realitas ini dapat diaplikasikan untuk semua indera, termasuk pendengaran, penciuman dan sentuhan.

Teknologi ini banyak digunakan dalam bidang-bidang kehidupan manusia, selain seperti kesehatan, militer, industri manufaktur maupun dunia pendidikan. Teknologi AR juga dipakai untuk menyisipkan suatu informasi tertentu ke dalam dunia maya, lalu menampilkannya ke dunia nyata melalui bantuan alat perlengkapan tambahan, seperti webcam, komputer, HP, android hingga kacamata.

Pengertian lain yang menjelaskan apa itu augmented reality adalah teknologi yang memperoleh penggabungan secara real-time terhadap digital konten yang dibuat oleh komputer bersama dunia nyata. Teknologi ini memperbolehkan penggunanya melihat objek maya 2D atau 3D yang diproyeksikan dari dunia nyata.

Teknologi ini semakin berkembang, yang awalnya diterapkan pada PC dan pada tahun 1990-an bahkan sudah diterapkan pada perangkat mobile. Perangkat mobile dinilai mendukung dalam pengembangan teknologi ini, hal itu tak lepas dari adanya komponen hardware dan software yang dibutuhkan untuk mengembangkan teknologinya.

1. *Cara Kerja Augmented Reality*

Sesuai dengan pengertiannya di atas bahwa teknologi AR bisa ditampilkan dalam berbagai perangkat mobile. Dalam prosesnya, perangkat mobile akan bertindak sebagai output device, yang bisa menampilkan informasi berupa video,

gambar, animasi dan model 3D. Nantinya, para pengguna akan melihat hasilnya berupa cahaya buatan dan alami.

Cara kerja teknologi ini menggunakan SLAM atau Simultaneous Localization and Mapping, sensor dan pengukur kedalaman. Seperti untuk pengukuran data sensor yang berfungsi sebagai alat untuk menentukan suatu lokasi dan menghitung jarak dari lokasi sebelumnya ke lokasi tujuan. Dan proses ini tak lepas dari komponen pendukung lainnya, seperti berikut ini.

2. Kamera dan Sensor

Kamera dan sensor digunakan untuk mengumpulkan data informasi yang berkolaborasi dengan pengguna, lalu dikirimkan untuk kemudian diproses. Kamera pada perangkat seperti ponsel memiliki kemampuan untuk memeriksa lingkungan dan data yang diperoleh, selain itu fungsi kamera lainnya adalah untuk menemukan barang fisik dan menghasilkan objek 3D.

3. Proyeksi

Mengacu pada proyektor kecil, seperti headset AR misalnya karena alat ini bisa dipakai untuk mengambil informasi dari sensor. Setelah itu memproyeksikan konten yang terkomputerisasi ke permukaan untuk kemudian dilihat. Meskipun sistem yang memanfaatkan komponen ini sepenuhnya belum dirancang ke dalam alat.

4. Refleksi

Bagi para pengguna gadget, jika teliti maka dapat dilihat pada beberapa gadget terdapat cermin yang fungsinya membantu mata manusia dalam melihat gambar secara virtual. Beberapa memiliki variasi cermin yang ditekuk dan yang

lain memiliki cermin sisi ganda, fungsinya memantulkan cahaya ke kamera dan ke mata pengguna.

K. UML (Unified Modelling Language)

UML (*Unified Modelling Language*) menurut Adi nugroho (2011:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”,

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

a. Sesuatu (*things*)

Ada empat tings dalam *Unified Modelling Language* (UML):

- 2) *Structural things*, bagian yang relatif statis dapat berupa elemen yang bersifat fisik maupun konseptual.
- 3) *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
- 4) *Grouping things*, bagian yang pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem.
- 5) *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model UML.

b. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

- 1) Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen independent akan mempengaruhi elemen yang bergantung padanya.
- 2) Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
- 3) Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi sedangkan arah sebaliknya dinamakan generalisasi.
- 4) Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

Diagram *Use Case* menyajikan intraksi antara *Use Case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun.

Adapun simbol-simbol *Use Case* diagram antara lain :

Table 2. 1 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

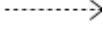
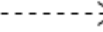
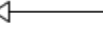
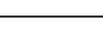
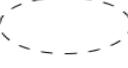
NO	GAMBAR	NAMA	KETERANGAN
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>Use Case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Table 2. 2 Simbol Class Diagram

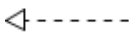
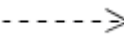
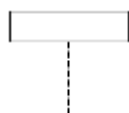
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Table 2. 3 Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Table 2. 4 Simbol StateChart Diagram

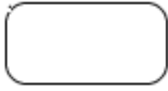
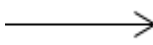
NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Table 2. 5 Simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

NO	GAMBAR	NAMA	KETERANGAN
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian deskriptif dimana memberikan gambaran mengenai apa yang sesungguhnya terjadi. Dalam pembuatan Skripsi ini digunakan metode deskripsif yang menggambarkan fakta-fakta dan informasi secara sistematis, faktual dan akurat.

Penelitian ini dilakukan melalui internet yang dapat memberikan sumber data dan pengetahuan mengenai sistem yang diteliti, kemudian mencocokkan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

B. Waktu dan Tempat Penelitian

Adapun waktu dan tempat penelitian yang dilaksanakan pada bulan Juni tahun 2023, dimana proses pengambilan data Bahasa isyarat yang akan dimasukkan dalam aplikasi nantinya

Table 3. 1 Kegiatan Penelitian

NO	Uraian Kegiatan	Bulan 2023				
		Mei	Juni	Juli	Agustus	September
1	Studi Literatur					
2	Perancangan aplikasi					
3	Pembuatan aplikasi					
4	Pengujian					

C. Metode Pengumpulan Data

Untuk memperoleh data-data yang dibutuhkan dalam rangka melakukan penelitian, maka penulis mengumpulkan data melalui beberapa cara yaitu :

1. Analisis Data

Menganalisa data-data yang sebelumnya telah dikumpulkan.

2. Perancangan Program

Sebagai pedoman dalam penulisan program atau kode-kode agar berjalan sesuai rencana.

3. Uji Coba Program

Pengujian program dilakukan untuk memastikan bahwa program yang dibuat dapat berjalan dengan baik.

4. Evaluasi

Sistem yang telah selesai dibangun perlu adanya evaluasi untuk menemukan kelemahan yang terdapat pada program yang telah dibangun tadi, yang nantinya bisa digunakan sebagai acuan untuk memperbaiki program sehingga lebih sempurna.

D. Jenis Data

Data yang digunakan dalam penelitian ini berupa data-data yang telah dikumpulkan melalui Penelitian Pustaka (*Library Research*). Adapun jenis data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

a. Data primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber data dan pengetahuan.

b. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Peneliti mendapatkan data yang sudah jadi dari internet, *website* dan jurnal.

E. Tahapan Penelitian

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data, analisis perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

2. Pengumpulan Data

Pada tahap ini peneliti melakukan observasi dengan peninjauan, pencatatan dan pengamatan langsung di tempat penelitian.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan oleh user.

F. Metode Pengujian Aplikasi

Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, query, atau jalur navigasi yang mewakili penggunaan sistem yang *tipical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail. Pengujian harus mencakup *unit testing*, yang mengecek validasi dari prosedur dan fungsi-fungsi secara independen dari komponen sistem yang lain. Kemudian modul testing harus menyusul dilakukan untuk mengetahui apakah penggabungan beberapa unit dalam satu modul sudah berjalan dengan baik, termasuk eksekusi dari beberapa modul yang saling berelasi, apakah sudah berjalan sesuai karakteristik sistem yang diinginkan.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

1. *White Box Testing*

Uji coba *white box testing* merupakan metode perancangan *testcase* yang menggunakan struktural untuk mendapatkan *testcase*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

Contoh pengujian *white box testing* ini merupakan peringatan ketika user menginputkan password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada *user* mengenai kesalahan yang dilakukan.

2. *Black Box*

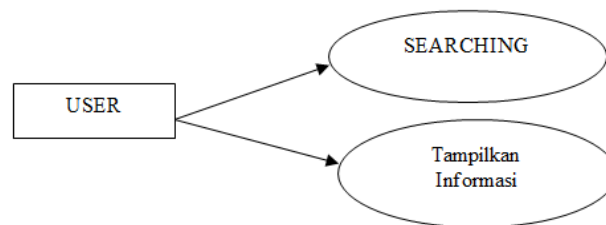
Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang dijalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah ditetapkan dan harus diikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika user belum menginput data yang harusnya di input sesuai ketentuan sistem yang dijalankan dan sistem memberikan informasi kepada user karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

G. Metode Pengujian

Metode pengujian yang dilakukan pada penelitian ini menggunakan *blackbox* yang dipakai untuk mengetahui kinerja aplikasi pada saat melakukan proses *Tour Guide* dan mengetahui berfungsinya semua aplikasi yang telah dibuat, *blackbox* dalam metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja.

H. Desain Sistem

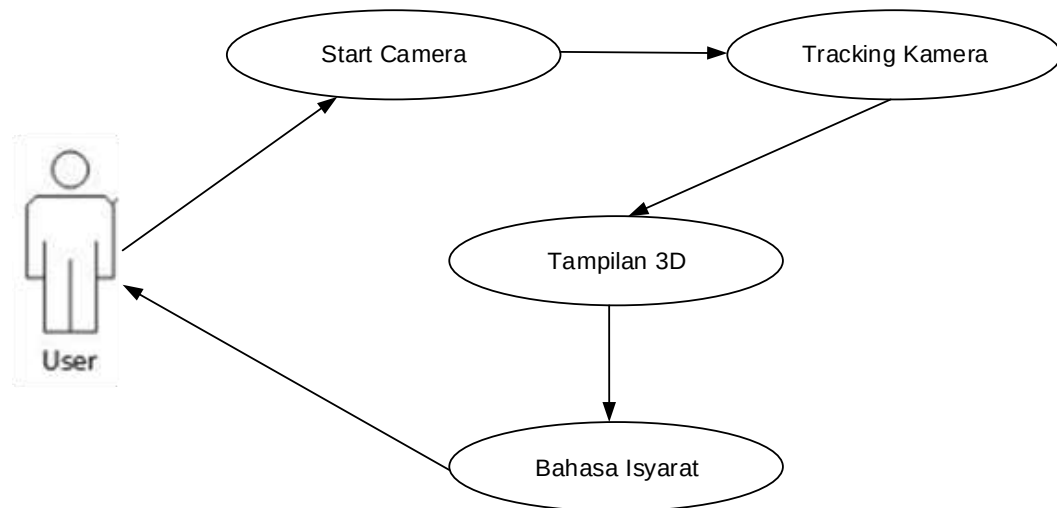
1. Sistem yang berjalan



Gambar 3. 1 Sistem Yang Berjalan

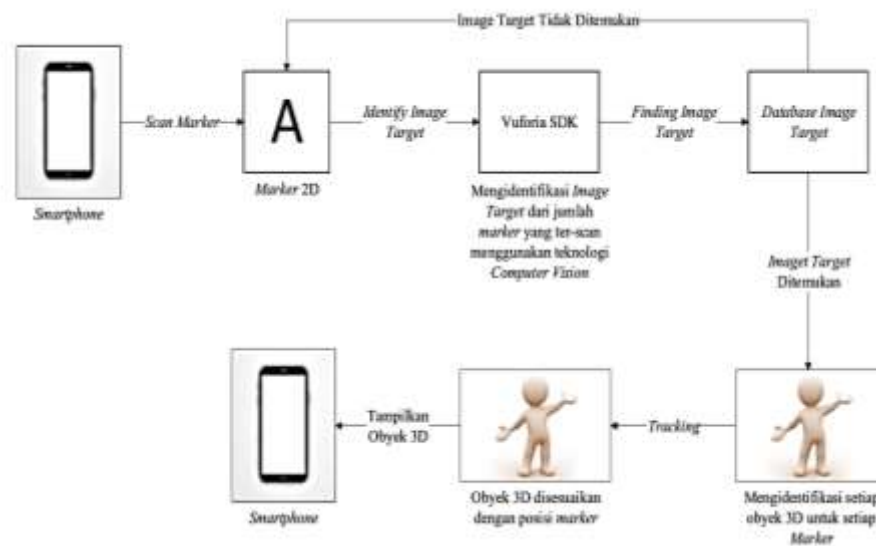
Pada gambar diatas menjelaskan mengenai sistem yang berjalan saat ini yang mana saat pengguna ingin mencari tahu mengenai Bahasa isyarat, pengguna terlebih dahulu *searching* di *google* atau *web* untuk mendapatkan informasi atau pemahaman mengenai Bahasa isyarat.

2. Sistem Yang Diusulkan



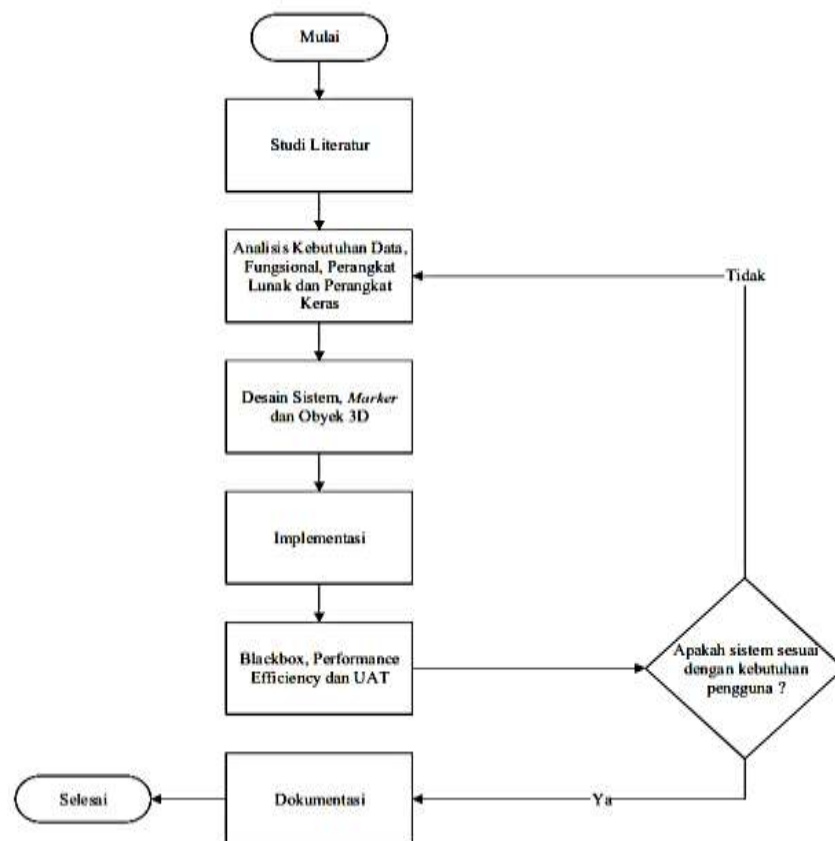
Gambar 3. 2 Sistem Yang Diusulkan

Pada gambar diatas menjelaskan mengenai sistem yang akan diusulkan, cara kerja aplikasi yang mana saat pengguna menjalankan aplikasi kemudian melakukan tracking kamera, pada saat itu akan muncul tampilan 3D tangan yang menampilkan bentuk Bahasa isyarat dan juga disertai dengan huruf agar lebih mudah untuk dipahami.



Gambar 3. 3 Lanjutan_Sistem Yang Diusulkan2

Pada gambar diatas menjelaskan mengenai alur aplikasi, 1) menjalankan aplikasi, 2) pengguna menyediakan marker 2D, 3) Vuforia sdk melakukan identifikasi pada *image* target yang di *scan* apakah marker sesuai dengan *image* target yang ada pada *database* atau tidak, 4) saat *image* target tidak ditemukan maka pengguna harus mengganti marker yang sesuai dengan *image* target yang ada pada *database*, 5) saat *image* target ditemukan maka akan muncul objek 3D untuk setiap marker yang ada, 6) saat objek 3D muncul akan disesuaikan dengan posisi marker dan akan ditampilkan pada smartphone.



Gambar 3. 4 Lanjutan_Sistem Yang Diusulkan3

Pada gambar *flowchart* menjelaskan mengenai sistem pembuatan aplikasi yaitu 1) mulai 2) Studi literatur melakukan analisis terhadap informasi-informasi yang relevan, 3) menyediakan kebutuhan analisis kebutuhan data, fungsional dll, 4) kemudian pembuatan desain sistem, marker dan pembuatan 3D objek yang dibutuhkan, 5) Melakukan implementasi pada aplikasi, 6) melakukan pengujian aplikasi pada *BlackBox* dan *WhiteBox* yang bertujuan untuk sesuai dengan kebutuhan dan desain sistem, 7) Pembuatan dokumentasi, 8) Selesai

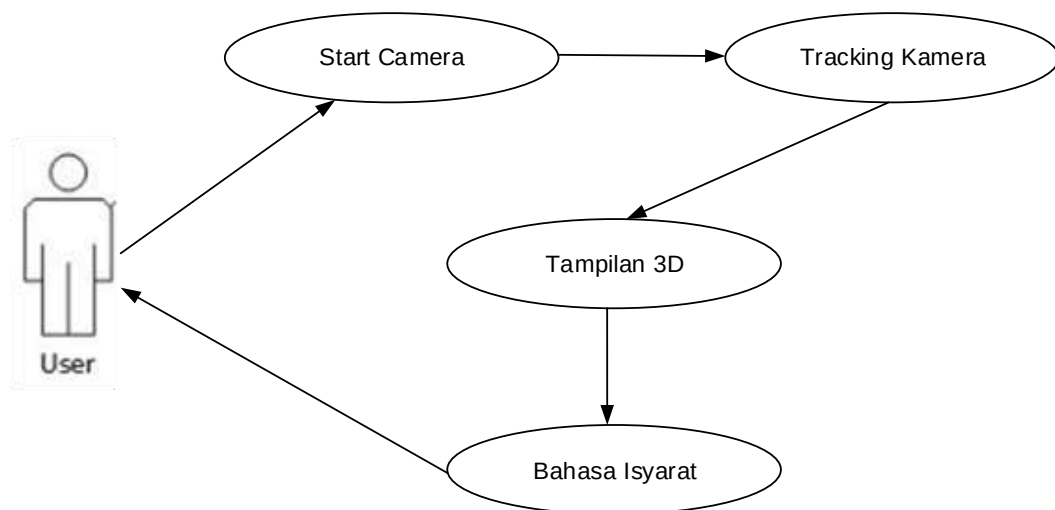
BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan Sistem yang di Usulkan

Sistem yang penulis buat adalah dengan metode pengunjung mengambil alat berupa smartphone berbasis android untuk selanjutnya menggunakannya sebagai Alat untuk deteksi nama Huruf dengan cara menyorotkan camera ke marker atau objek tersebut guna mendapatkan informasi.

1. Use Case Diagram



Gambar 4. 1 Diagram *User Case system* yang diajukan

Penjelasan *Use Case* Diagram di atas :

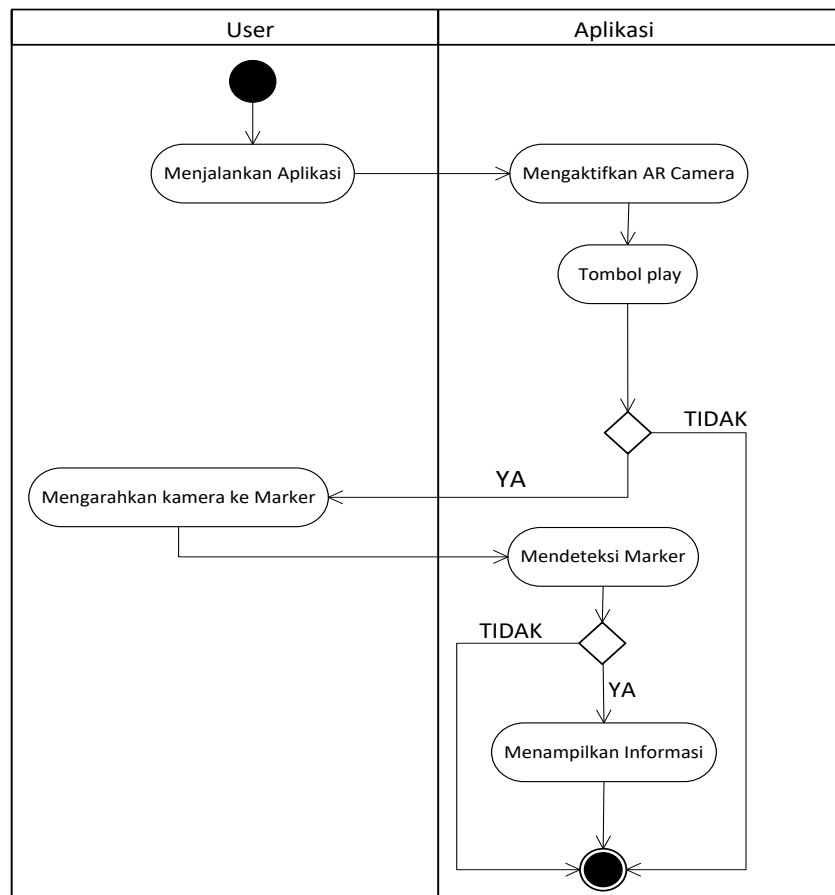
a. *Actor* Pengguna

Table 4. 1 *Actor* Pengguna

Nama Use Case	Deskripsi Use Case
AR Camera	<i>Use Case</i> ini menjelaskan bahwa setelah user mengaktifkan aplikasi, akan terbuka form AR kamera yang di dalamnya juga terdapat form pengaturan dan menu <i>User</i> sudah dapat menggunakannya untuk mendapatkan informasi dengan mengarahkannya ke marker atau objek.
Deteksi Marker	<i>Use Case</i> ini menjelaskan tentang proses Pendeteksian marker
Munculkan Informasi dari benda yang di sorot	<i>Use Case</i> ini menjelaskan mengenai informasi yang tampil setelah Proses Deteksi Marker selesai.
<i>Menu play</i>	<i>Use Case ini</i> menjelaskan bahwa jika kita menekan tombol <i>play</i> maka akan tampil sebuah proses deteksi marker

2. *Activity Diagram*

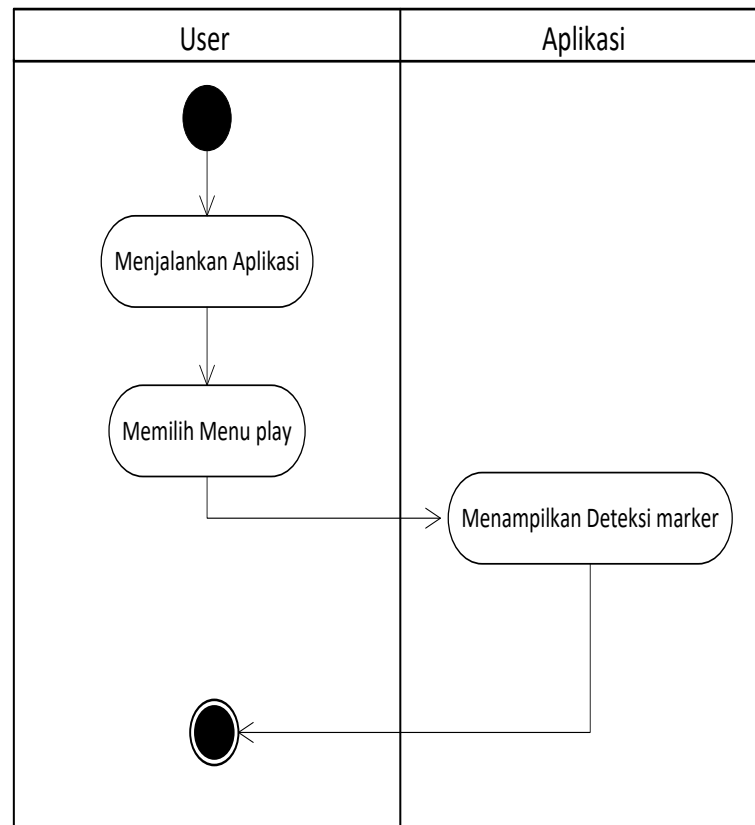
a. *Activity Diagram* Aplikasi



Gambar 4. 2 Diagram Activity Aplikasi Sistem yang diajukan

Activity Diagram di atas menjelaskan bahwa ketika user menjalankan Aplikasi ini akan langsung di arahkan ke AR kamera yang di dalamnya juga sudah tersedia *menu play*, apabila *user* mengarahkannya ke marker akan menampilkan informasi, jika terkendala koneksi Aplikasi akan keluar, dan jika marker tidak terdeteksi dengan baik, informasi yang di inginkan tidak akan tampil.

b. Actifity Diagram Menu play

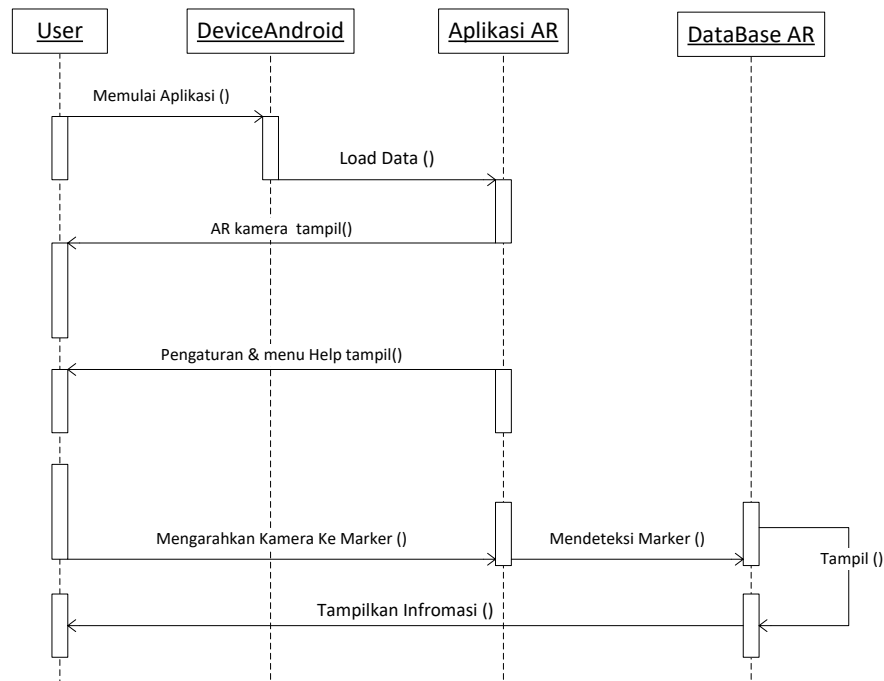


Gambar 4. 3 Diagram Activity Menu play Sistem yang diajukan

Activity Diagram di atas menjelaskan bahwa jika user menjalankan Aplikasi ini di bagian sudut atas aplikasi terdapat tombol play yang apabila di tekan akan tampil sebuah *PopUp* yang berisikan proses deteksi marker.

3. Sequence Diagram

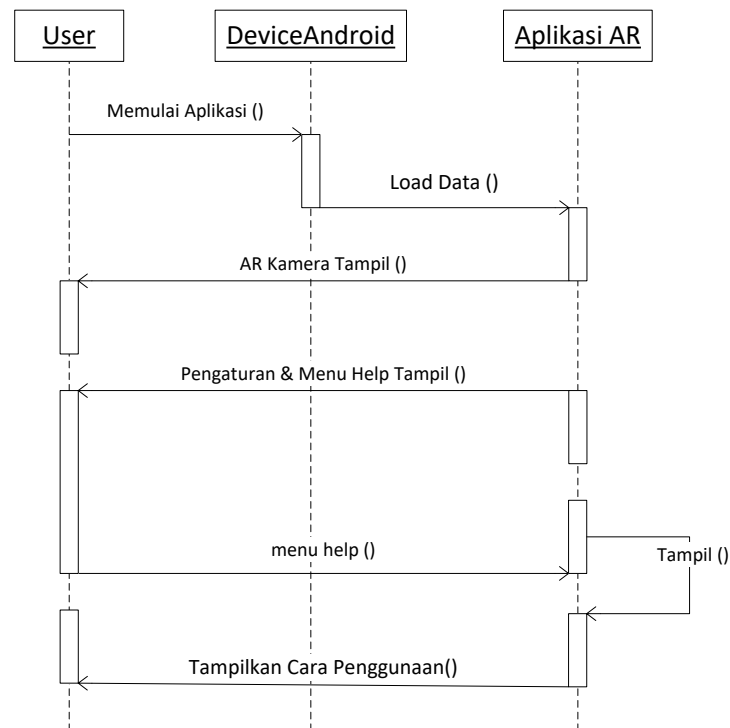
a. Sequence Diagram Aplikasi



Gambar 4. 4 Diagram Sequence Aplikasi Sistem yang diajukan

Diagram Sequence di atas menjelaskan bahwa ketika *user* memulai Aplikasi maka *device Android* yang di gunakan akan meload data untuk selanjutnya memulai aplikasi ini, setelah itu akan tampil AR *camera* dan *Menu* pengaturan serta tombol, setelah AR kamera tampil *user* akan mengarahkan *device android* tersebut ke marker yang telah disediakan yang kemudian aplikasi melakukan proses pendeteksian Marker untuk selanjutnya melakukan pencocokan dengan database jika sesuai dengan data base informasi dari objek atau marker yang di sorot akan tampil.

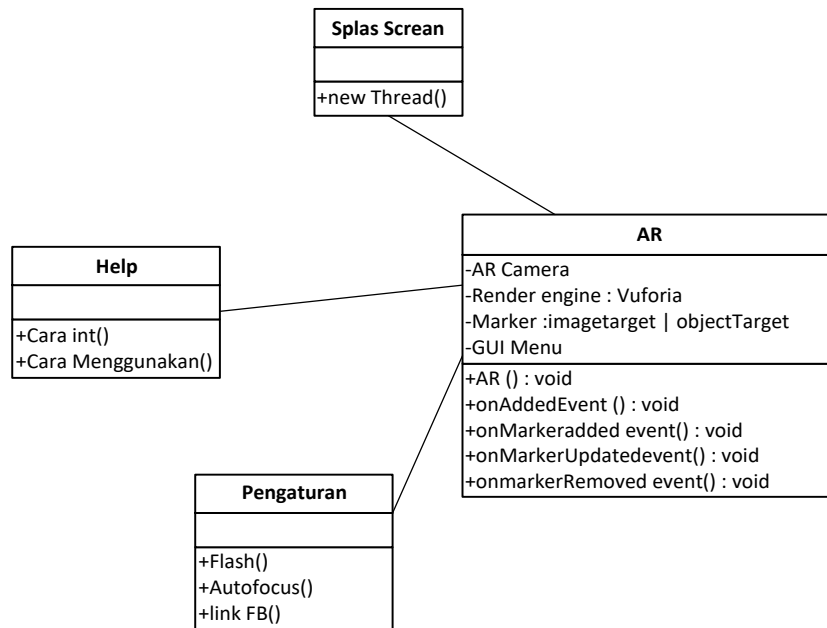
b. *Sequence Diagram Menu Play*



Gambar 4. 5 Diagram *Menu play* Sistem yang diajukan

Diagram Sequence di atas menjelaskan bahwa ketika *user* menjalankan Aplikasi akan tampil AR kamera di sertai dengan menu play dan tombol *play*, ketika *user* menekan tombol *play* Aplikasi Akan menampilkan *PopUp* Aplikasi ini.

4. Class Diagram



Gambar 4. 6 Diagram Class Sistem yang diajukan

Class Diagram di atas menjelaskan keterkaitan antara setiap *form* dan fungsi-fungsi yang ada di dalamnya, seperti ketika Aplikasi di jalankan pertama kali maka yang akan tampil pertama kali adalah munculnya sebuah *splashscreen* sebelum masuk ke form *AR* yang merupakan inti dari aplikasi ini di dalamnya proses pendeteksian marker terjadi, adapun jika kita menekan tombol *Play* maka akan di arahkan ke form *PopUp* yang berisi cara penggunaan aplikasi ini secara umum.

B. Detail Aplikasi

1. Membuat Akun *Vuforia Engine*.

- a. Buka Situs Web *Vuforia*: Akses situs web resmi *Vuforia* di

<https://developer.vuforia.com/>.

- b. Pilih "**Log In**" atau "**Register**": Temukan tombol atau opsi yang memungkinkan Anda membuat akun baru. Biasanya, ini berada di sudut kanan atas atau tengah halaman beranda.
- c. Isi Informasi Akun: Anda akan diminta untuk mengisi formulir pendaftaran. Informasi yang diminta biasanya mencakup nama, alamat *email*, dan pembuatan kata sandi.
- d. Verifikasi *Email* Anda: Setelah mengisi formulir, cek *email* Anda untuk mengonfirmasi dan melakukan verifikasi akun.
- e. Login ke Akun Anda: Setelah verifikasi berhasil, *login* ke akun *Vuforia* Anda menggunakan *email* dan kata sandi yang telah Anda buat.
- f. Mulai Penggunaan *Vuforia*: Setelah *login*, Anda akan memiliki akses ke portal pengembang *Vuforia*. Di sini, Anda dapat mulai membuat proyek, mengakses dokumentasi, dan mengelola kunci API serta *lisensi*.
- g. Membuat *Lisensi* pilih **License Manager**, beri nama "Umpar".



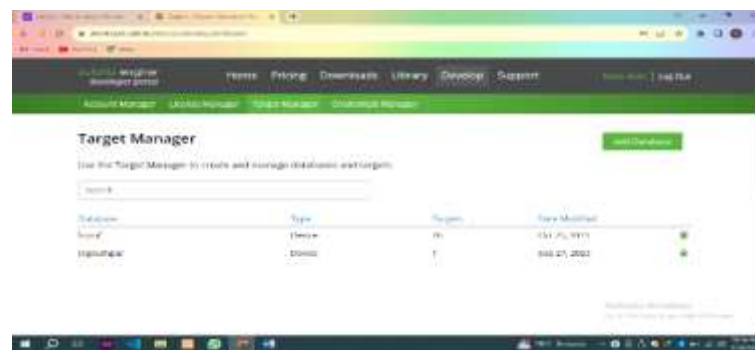
Gambar 4. 7 Tampilan *Licency Manager*

- h. Salin kunci lisensi yang diberikan dan terapkan ke dalam kode proyek Anda.



Gambar 4. 8 Tampilan *Licency*

- i. Selanjutnya membuat *Database* pada menu **Target Manager**, lalu pilih **Add Database** isi nama database dan pilih *type* yang diinginkan.



Gambar 4.9 Tampilan *Licency*

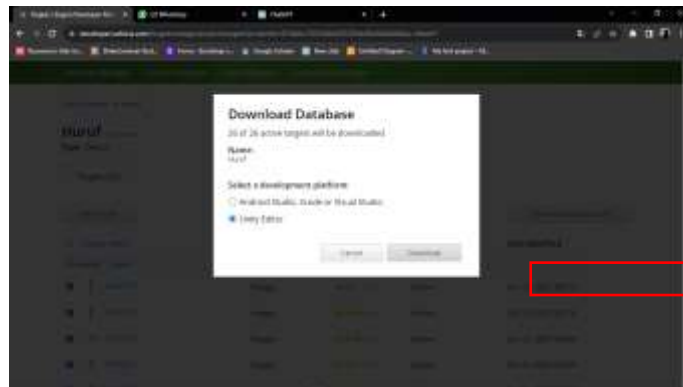


Gambar 4. 10 Tampilan *Add Target*

- j. Untuk menambahkan target pilih *database* yang telah dibuat, lalu klik **Add Target** , pilih dan isi **Type**, **File** yang akan dijadikan target, **Width**

(200), dan Nama **Target** lalu klik **Add**.

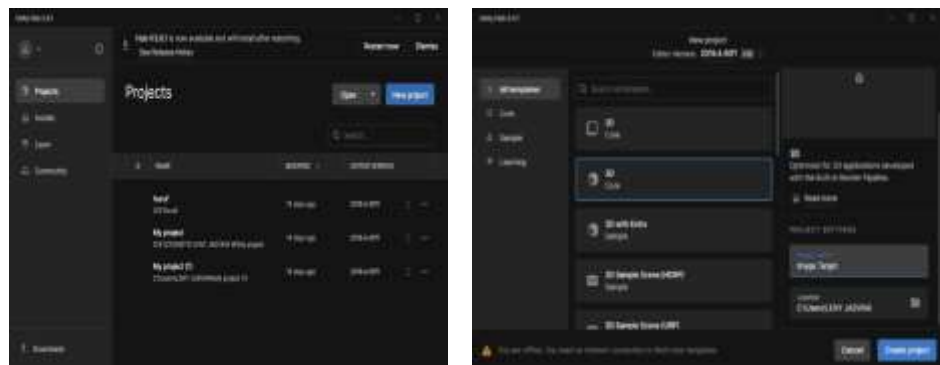
- k. Pilih target yang telah ditambahkan, lalu pilih **Download Database** (pilih *Unity Editor*), lalu **Download**.



Gambar 4. 11 Tampilan *Add Database*

2. Membuat Projek pada aplikasi *Unity*

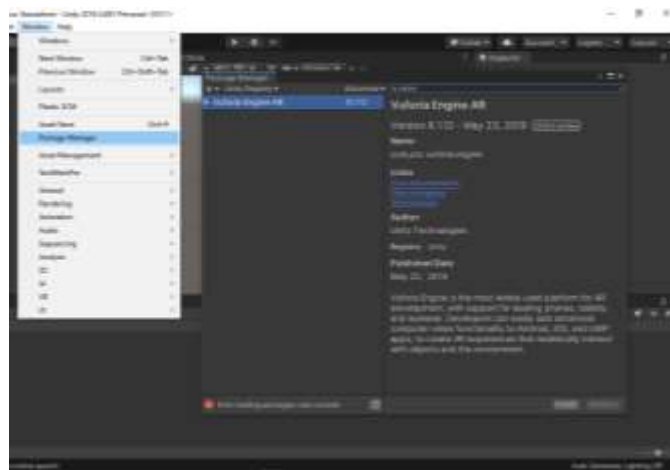
- a. Membuka Aplikasi *Unity*, berikut tampilan awal aplikasi *unity*



Gambar 4. 12 Tampilan *Project Unity*

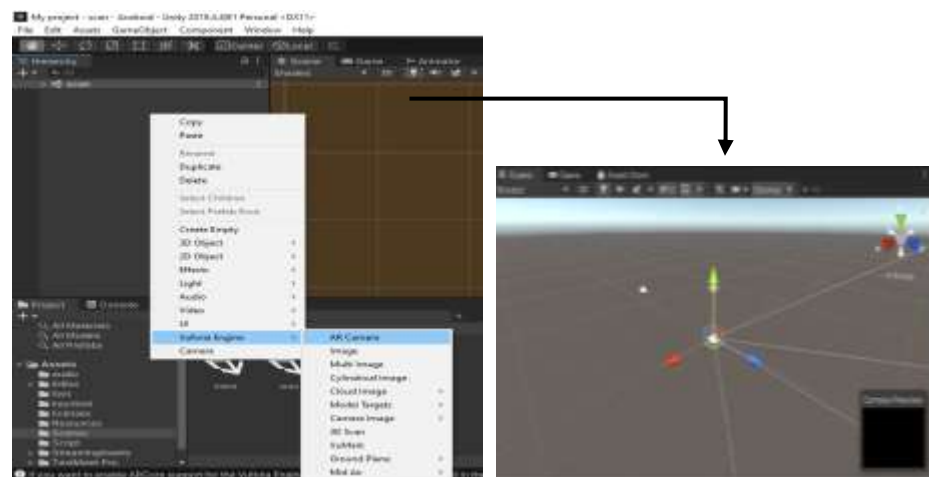
- b. Buat projek baru, klik **New Project** pilih **3D** isi nama projek lalu **Create**.
- c. Setelah muncul tampilan awal projek, selanjutnya menginstal **Vuforia Engine** dengan cara ke menu **Window > Package Manager**. Pada

kotak pencarian ketikkan **Vuforia** lalu **Install**.



Gambar 4. 13 Tampilan *Package Manager*

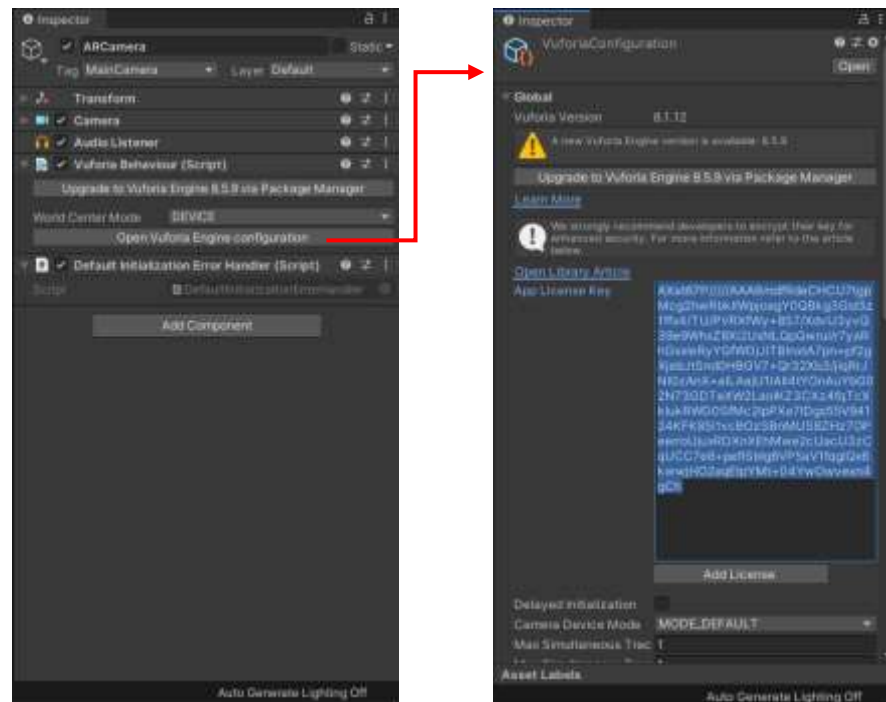
- d. Selanjutnya ke menu **Hierarchy**, klik kanan pilih **Vuforia Engine** > **AR Camera**.



Gambar 4. 14 Tampilan *AR Camera*

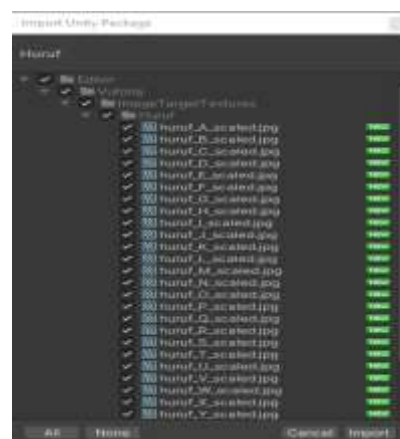
- e. Menambahkan **Lisensi Key** agar dapat terhubung ke **Vuforia Engine** yang telah dibuat sebelumnya kedalam AR Camera. Klik **AR Camera** dibagian **Inspector** > **Open Vuforia Engine Configuration** > **paste**

lisensi yang telah dicopy sebelumnya.



Gambar 4. 15 Tampilan Mengisi Licency

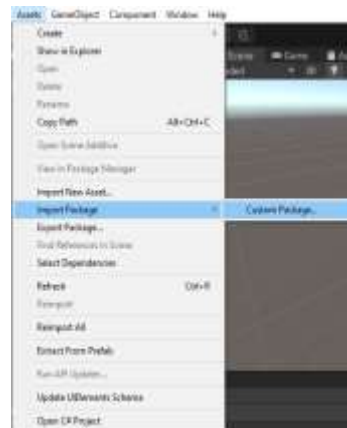
1. Menambahkan Image



Gambar 4. 16 Tampilan Image Target

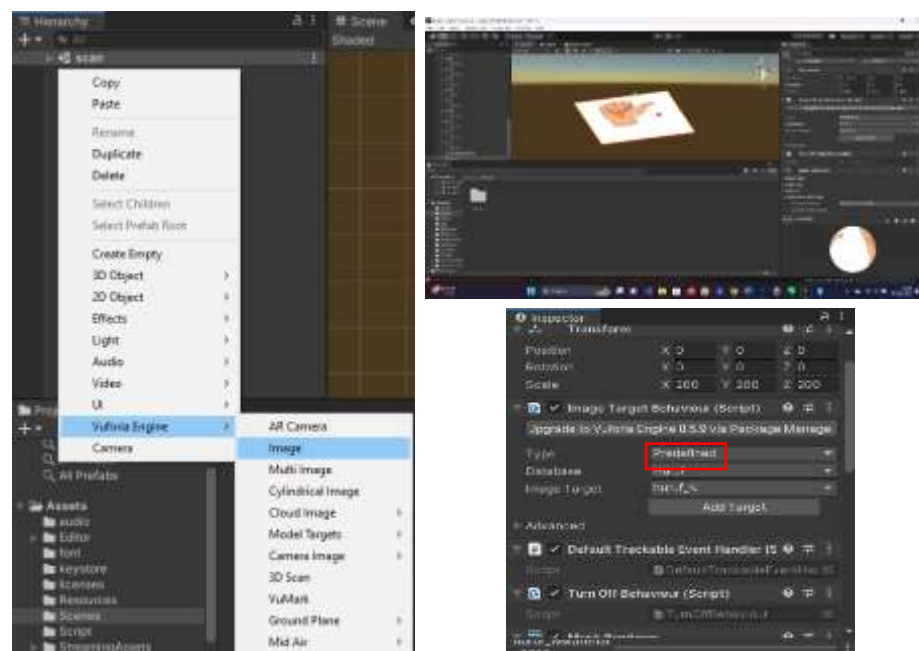
a. Mengimport *database* yang telah didownload sebelumnya dengan cara

ke Menu bar **Assets > Import Package > Costum Package**, pilih file yang telah didownload tadi lalu **Import**.



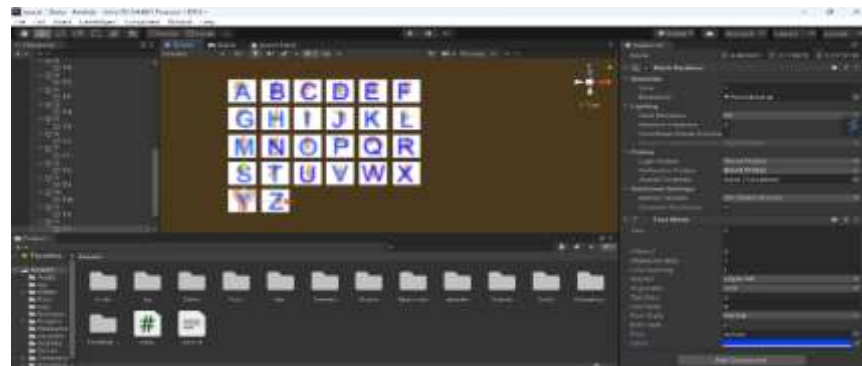
Gambar 4. 17 Tampilan *Import Database*

- b. Menambahkan *image*, caranya pada Menu **Hierarchy** klik kanan
- c. **Vuforia Engine > Image**.



Gambar 4. 18 Tampilan *Vuforia Enggine*

Ulangi langkah yang sama untuk menambahkan semua image huruf yang telah di import (A-Z) atur setiap gambar berdasarkan huruf pada **Inspector > Image Target**.

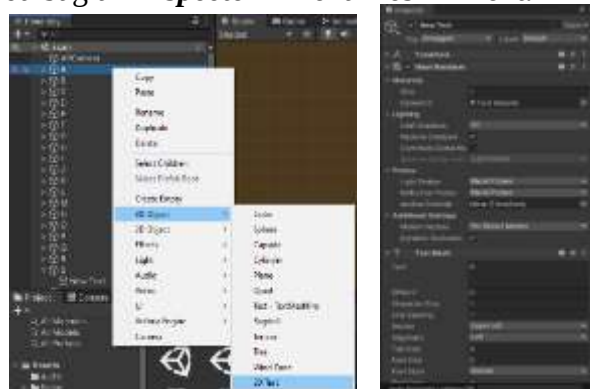


Gambar 4. 19 Tampilan data Huruf

2. Menambahkan 3D text

Dengan klik kanan pada bagian image yang akan ditambahkan 3D text pilih **3D Object > 3D Text**, lakukan hal yang sama pada seluruh gambar.

Ubah text pada bagian **Inspector > Text Mesh > Text**.



Gambar 4. 20 Tampilan 3D Teks

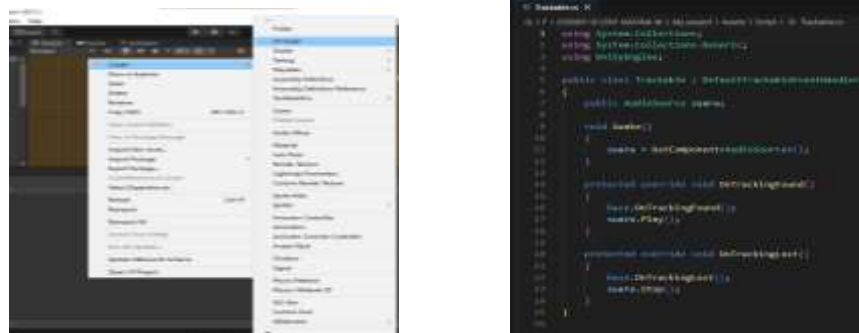
3. Menambahkan Audio

- a. Pertama yang harus di lakukan adalah persiapan file audio dengan menggunakan format **WAV, MP3, OGG, AIFF, FLAC**, dan **M4A**.



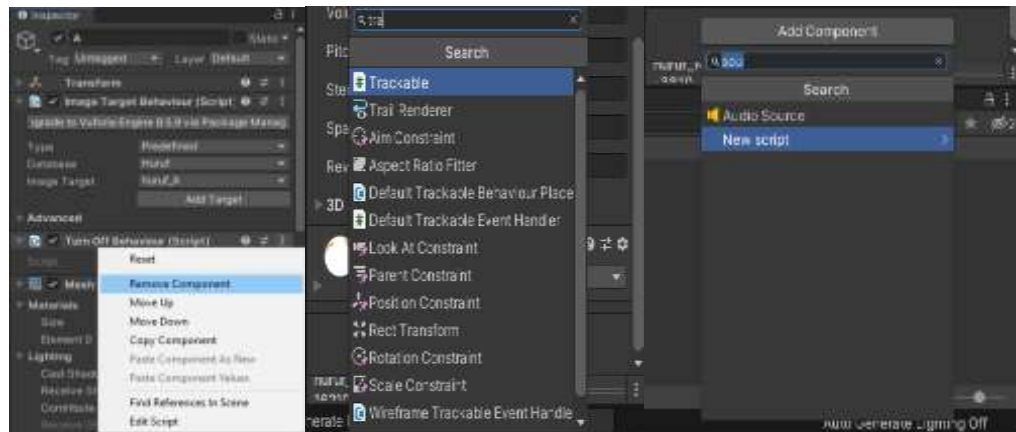
Gambar 4. 21 Tampilan *Audio*

- b. Import file audio ke unity dengan membuka jendela **Assets** > buat folder dengan nama **Audio** dan gunakan opsi **Import New Asset**.
- c. Selanjutnya buat script dengan menggunakan bahasa pemrograman **C#** di *unity*, untuk mengatur pemutaran audio. Dengan cara ke assets buat folder **Script**, pada folder script klik kanan > **Create** > **C#** lalu ganti nama filenya menjadi **Tackable**.



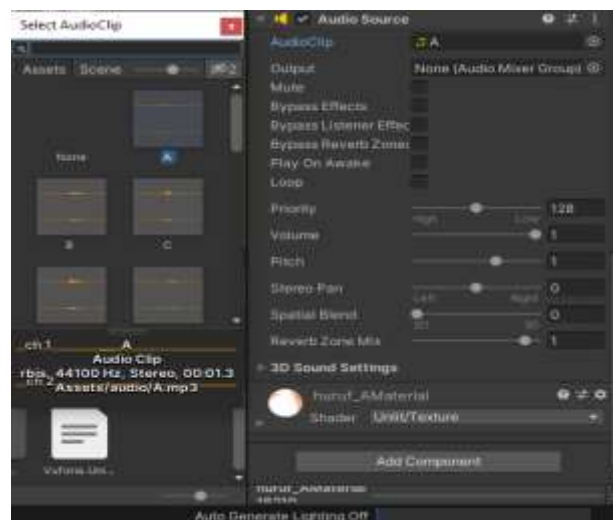
Gambar 4. 22 Tampilan *Skript*

- d. Atur properti *audio source*, blok semua huruf di **Hierarchy** lalu pada bagian inspector remove '**Trun Off Behaviour (Script)**' dan masih di Inspector **Add Component > Audio Source** dan **Trackable**.



Gambar 4. 23 Tampilan atur audio source

- e. Di bagian inspector **audio source** masukan audio di setiap huruf sesuai dengan gambar huruf tersebut. Pada audioClip dan hilangkan centang pada bagian **Loop** dan **Play On Awake**.

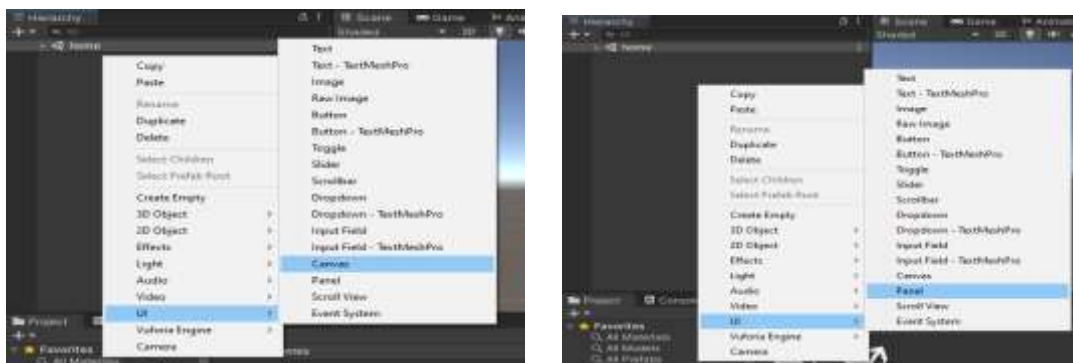


Gambar 4. 24 Tampilan Play On Aweken

- f. Maka saat project di *run*, gambar akan di scan dan muncul suara serta 3D *text* sesuai dengan huruf yang discan.

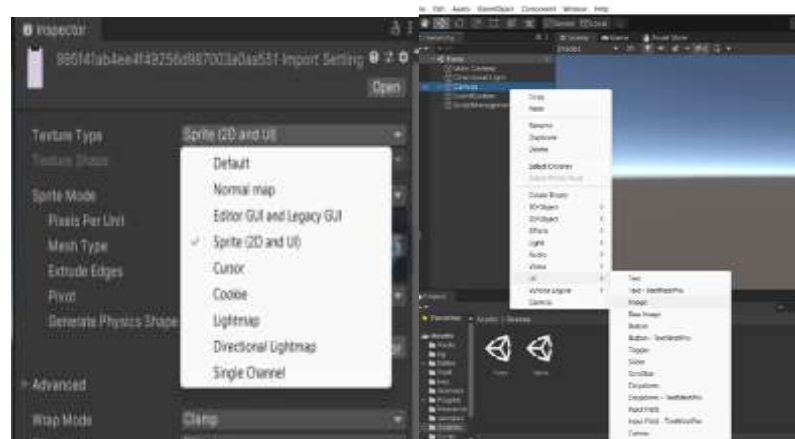
4. Membuat tampilan awal mulai aplikasi.

- a. Pertama buat scene baru dalam folder **Asset > Scene**. Buat Scene “Home” dengan cara klik kanan **Create > Scene**. Lalu double klik untuk masuk ke scene **Home**.
- b. Klik kanan pada bagian **Hierarchy > UI > Canvas** dan **Panel**



Gambar 4. 25 Tampilan Canva

- c. Mengatur tampilan panel ke halaman game dan atur sesuai dengan tampilan yang diinginkan. Di sini saya menggunakan tampilan horizontal dengan ukuran **1280 X 720 portrait**.
- d. Untuk menambahkan image atau background pada tampilan awal pilih **Canvas > Image**. Tapi sebelum itu siapkan image yang ingin di jadikan background dan masukkan di dalam asset lalu ganti formatnya dengan cara texture **Type > Sprite 2D Dan UI**



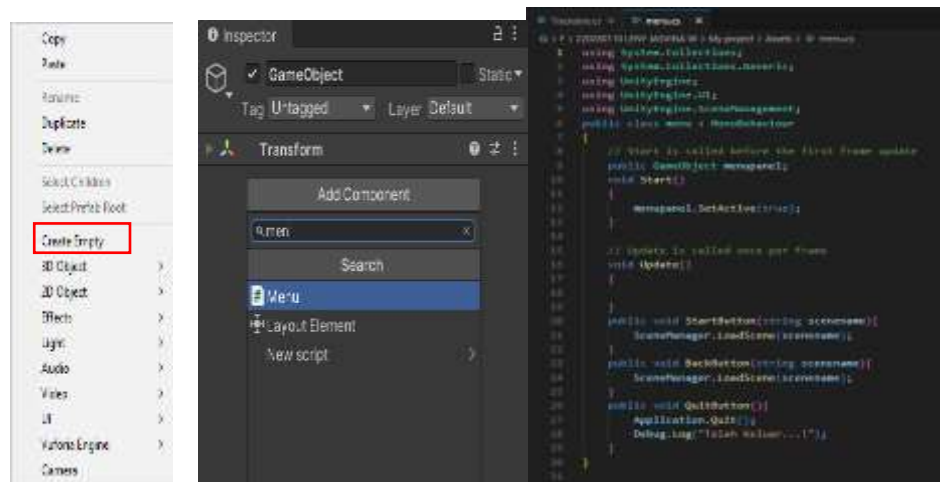
Gambar 4. 26 Tampilan *Sprite UI*

- e. Untuk membuat button Start dan Exit klik kanan kanvas **UI > button – textMeshPro** dan atur ukurannya menggunakan **rect tool**. Lalu masukkan image button di **source image** lalu ganti textnya di **Text (TMP)** menjadi **Start**. Lakukan hal yang sama untuk membuat tombol **Exit**.



Gambar 4. 27 Tampilan *Button*

- f. Setelah itu untuk menjalankan aksi button kita perlu membuat **Script** dengan cara *create empty* di **Hierarchy Home > Add Component (Tuliskan Menu) > New Script > Create**, maka script akan otomatis ada di menu asset, klik kanan untuk masuk ke script menu dan isikan \scriptnya.



Gambar 4. 28 Tampilan *Create Empty*

- g. Pilih menu button mulai lalu di bagian inspector di bagian *button on click()* pilih *add to list* dan *drack Game Object* yang isinya script tadi ke dalam *runtime only*, lalu pada bagian *no function* pilih menu *> startbutton (string)* dan pada bagian bawahnya masukan scene yang akan di tempati berpindah yaitu *scene scan*.



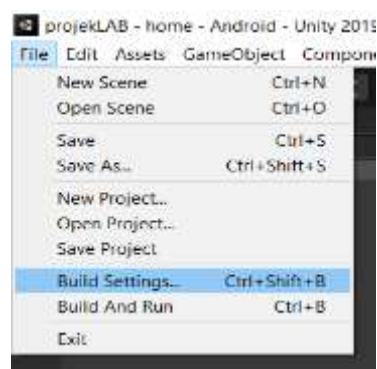
Gambar 4. 29 Tampilan *Start Button*

- h. Lalu pada *button* keluar buat hal yang sama dengan button mulai namun bedanya ada pada *functionnya* pilih *QuitButton()*.



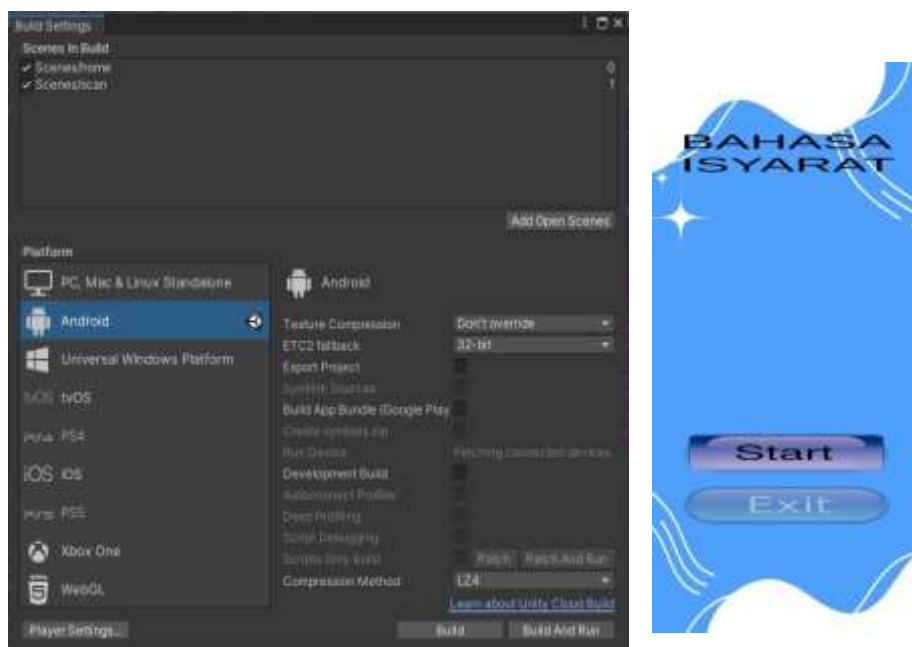
Gambar 4. 30 Tampilan *Menu*

5. Mengubah projek *unity* menjadi aplikasi *android*



Gambar 4. 31 Tampilan *Build Setting*

- Setelah semua proyek selesai dibuat, langkah berikutnya adalah melakukan *build* proyek. Pilih **file > build settings**
- Pada menu *build* (build setting), masukkan semua adegan secara berurutan, mulai dari adegan beranda hingga adegan pemindaian huruf, dan pastikan untuk mengubah platformnya menjadi Android. Jika Anda ingin melakukan penyesuaian seperti tampilan ikon aplikasi dan hal lainnya, Anda dapat masuk ke pengaturan pemain (pengaturan pemain). Setelah selesai, pilih pembangunan (*build*), berikan nama file aplikasi, dan simpan. Kemudian tunggu hingga proses build selesai. Setelah selesai pindahkan file aplikasi ke android anda dan install.



Gambar 4. 32 Tampilan *Output*

C. Pengujian sistem

1. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Pada tahap ini design sistem yang telah dirancang diterapkan ke dalam bahasa pemrograman yang kemudian di lakukan pengujian sistem.

Adapun bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah bahasa pemrograman *C# (Sharp)*.

a. Kebutuhan *Hardware*

Spesifikasi minimum *hardware* pengguna aplikasi ini adalah sebagai berikut :

Table 4. 2 Spesifikasi *Hardware Smartphone*

Jenis	Spesifikasi
Smartphone	Semua Merk
Processor	Intel® Atom™ 1.60 Ghz
RAM	4 GB
Memori	6 GB
Camera	5 MP (Autofokus : Ya Flash : Ya)
Display	Touchscreen 4.0 inches

b. Kebutuhan *Software*

Spesifikasi minimum *software* pengguna aplikasi ini yaitu sebagai berikut :

Table 4. 3 Spesifikasi *Software*

Jenis	Spesifikasi
Sistem Operasi	Android Ver. 11

2. Pengujian aplikasi

a. Metode Pengujian

Pengujian aplikasi dilakukan dengan menggunakan dua metode pengujian yaitu Pengujian *Black Box (White Box Testing)* dan Pengujian *White Box (Black Box Testing)*.

b. Teknik Pengujian

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada interface perangkat lunak.

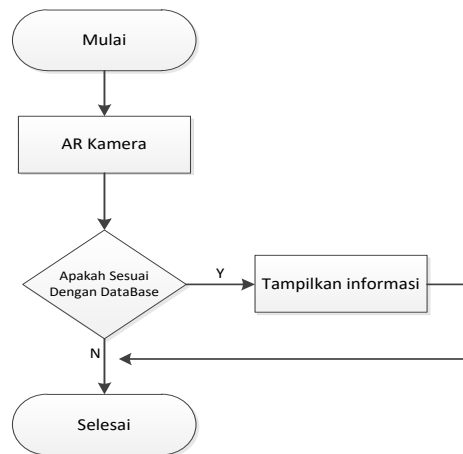
Meskipun didesain untuk mengungkapkan kesalahan, pengujian *black-box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional, bahwa input diterima dengan baik dan output dihasilkan dengan tepat, dan integritas informasi *external* (seperti file data) dipelihara. Pengujian *blackbox* menguji beberapa aspek dasar suatu aplikasi dengan sedikit memperhatikan struktur logika internal perangkat lunak tersebut.

sedangkan Pengujian *WhiteBox* berfokus pada efektifitas aplikasi yang dirancang.

c. Pengujian *White Box*

1. *White Box* Aplikasi

a. *FlowChart* Aplikasi

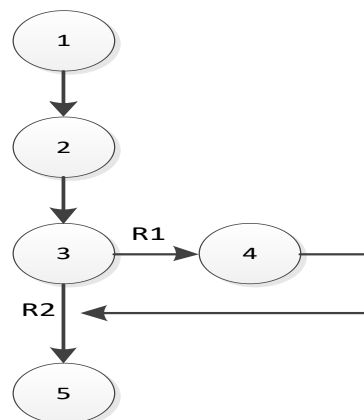


Gambar 4. 33 *FlowChart* Aplikasi

d. *FlowGraph* Aplikasi

Dari *FlowChart* Aplikasi yang di gunakan untuk pengujian perangkat lunak, maka di tentukan *FlowGraph* sebagai berikut :

a. *FlowGraph* Aplikasi



Gambar 4. 34 *FlowGraph* Aplikasi

1) Proses Perhitungan

Dari Gambar FlowGraph di atas dapat di lakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexcity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$$N(\text{node}) = 5$$

$$E(\text{edge}) = 5$$

$$P(\text{predikat node}) = 1$$

$$\text{Penyelesaian : } V(G) = E - N + 2$$

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat Node (N)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

b) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

$$\text{Path 2} = 1 - 2 - 3 - 5$$

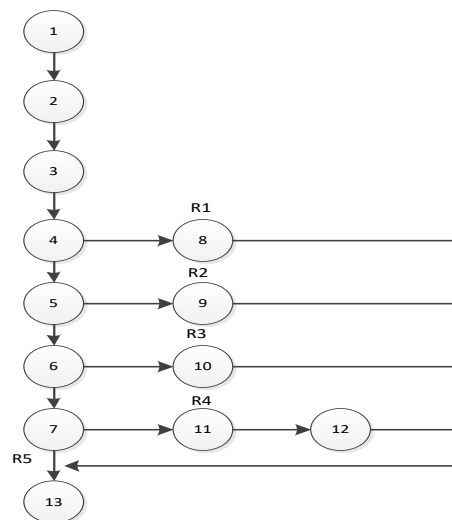
d) Grafik Matriks

Table 4. 4 Grafik Matriks aplikasi

	1	2	3	4	5	E-1
1		1				1-1=0
2			1			1-1=0

3				1	1	$2-1=1$
4					1	$1-1=0$
5						
Zum (E+1)						$1+1=2$

b. *FlowGraph* Pengaturan



Gambar 4. 35 *FlowGraph* Aplikasi

1) Proses Perhitungan

Dari Gambar *FlowGraph* di atas dapat dilakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$N(\text{node}) = 13$

$E(\text{edge}) = 16$

$P(\text{predikat node}) = 1$

Penyelesaian : $V(G) = E - N + 2$

$$= 16 - 13 + 2$$

e. Pengujian

Table 4. 6 Hasil Pengukuran Marker

Jarak	A	B	C	KET
20 cm	✕	✕	✓	✓=terlihat ✕= tidak terlihat
25 cm	✕	✓	✓	
30 cm	✓	✓	✓	
35 cm	✓	✓	✓	
40 cm	✓	✓	✓	
45 cm	✓	✓	✓	
50 cm	✓	✓	✓	
Derajat Kemiringan	A	B	C	
30 ⁰	✕	✕	✕	
35 ⁰	✕	✕	✕	
40 ⁰	✕	✕	✕	
45 ⁰	✕	✕	✕	
50 ⁰	✕	✕	✕	
55 ⁰	✕	✕	✕	
60 ⁰	✕	✕	✕	
65 ⁰	✕	✕	✕	
70 ⁰	✓	✓	✕	
75 ⁰	✓	✓	✓	
80 ⁰	✓	✓	✓	
85 ⁰	✓	✓	✓	
90 ⁰	✓	✓	✓	

Dari table di atas dapat di ambil kesimpulan sebagai berikut :

1. Huruf A

- Jarak Minimum : 30 cm
- Jarak Maximum : 50 cm
- Jarak Optimal : 40 cm
- Derajat Kemiringan : 70⁰

- Derajat Optimal : 90^0

2. Huruf B

- Jarak Minimum : 25 cm
- Jarak Maximum : 50 cm
- Jarak Optimal : 30 cm
- Derajat Kemiringan : 70^0
- Derajat Optimal : 90^0

3. Huruf C

- Jarak Minimum : 20 cm
- Jarak Maximum : 35 cm
- Jarak Optimal : 25 cm
- Derajat Kemiringan : 75^0
- Derajat Optimal : 90^0

Selain dari Marker di atas Penulis juga melakukan percobaan terhadap marker Huruf tertentu dengan tujuan untuk menguji sistem dari Aplikasi ini, khususnya dalam hal pendeteksian guna mengetahui apakah dapat di terapkan pada Huruf nyata (bukan dalam bentuk Marker / gambar).

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan oleh penulis maka dapat ditarik kesimpulan sebagai berikut :

Penerapan teknologi *Augmented Reality* dalam hal ini diharapkan dapat membantu Anak didik untuk mengenal huruf menggunakan kamera, yang dalam penelitian ini. metode dalam penyampaian informasi diaplikasi ini dengan memanfaatkan fitur *Augmented reality* yang dsediakan oleh *Vuforia* dalam proses pendeteksian objek untuk menampilkan infomasi. Secara umum proses pendeteksian yang dilakukan yaitu dengan melakukan proses image prosesing terhadap marker Huruf, kemudian dilakukan pengimputan data-data berupa informasi dari tiap-tiap marker huruf yang telah dilakukan proses *Image Prosessing* sebelumnya untuk dibuat dalam sebuah file APK yang dapat dikenali oleh Sistem Operasi Android. Penggunaan sistem operasi Android tidak lepas dari sifat dari OS ini yang bersifat *free*, dengan memanfaatkan Smartphone yang berOS Android. Dilakukan pengujian dengan menggunakan 26 sampel dengan jarak minimum adalah 20 cm dan jarak maximum yang bervariasi antara 35 cm – 50 cm, hal ini di sebabkan oleh beberapa factor di antaranya kualitasbenda yang dideteksi dan factor pencahayaan yang kurang memadai. Selain itu juga didapatkan hasil bahwa derajat yang paling baik atau sudut terbaik untuk

menampilkan hasil yang optimal berada pada sudut 90^0 atau posisi kamera berhadapan langsung dengan objek deteksi.

B. Saran

Setelah melakukan penelitian, maka penulis memberikan saran sebagai berikut :

1. Penulis menyarankan agar pada perkembangan aplikasi kedepannya dapat memperbaharui aplikasi sesuai dengan kebutuhan masyarakat sehingga dapat meminimalisir hardware yang digunakan.
2. Aplikasi ini masih kurang dalam segi fitur, interface dan kualitas aplikasi maka dari itu disarankan untuk meneliti selanjutnya dapat meningkatkan fitur aplikasi salah satunya dapat mendeteksi kata atau kalimat agar lebih mudah untuk di pelajari.
3. Dalam aplikasi ini masih memiliki berbagai kekurangan salah satunya. Meskipun telah dilakukan pengujian dengan jarak minimum dan maksimum tertentu, masih terdapat variasi jarak yang dapat memengaruhi kualitas deteksi huruf. Hal ini menunjukkan bahwa aplikasi mungkin memiliki keterbatasan dalam menangani jarak yang lebih jauh atau lebih dekat dari yang diujikan. Disarankan untuk Penelitian selanjutnya dapat melakukan meningkatkan kinerja aplikasi dalam mengenali huruf pada jarak yang lebih jauh atau lebih dekat.

DAFTAR PUSTAKA

- Adi nugroho. (2011). *Perancangan dan implementasi sistem basis data*. yogyakarta: Andi.
- Andi widi (2021) Ponsel cerdas (bahasa inggris: *smartphone*) adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer
- Cawood, Stephen dan Mark Fiala. 2017. *Augmented Reality: A Practical Guide*.
- Febriyan M.Fajar,2014. *Augmented Reality with ARToolKit*. Nulis Buku. Bandar Lampung.
- Hermawan dan Hariadi (2021) dalam penelitiannya yang berjudul "Pemanfaatan *Augmented Reality* Sebagai Media Informasi Kampus Menggunakan Brosur”
- <http://developer.vuforia.com/library/articles/training> [diakses pada 01-03-2024].
- Leonardi dan Hermawan (2019) dalam penelitiannya yang berjudul "Penerapan *Augmented Reality* Pada Brosur Interior Berbasis Multi *Markerless* Pada *Android*"
- Nadira, Sujaini dan Pratiwi (2020) dalam penelitiannya yang berjudul “Implementasi *Augmented Reality* Pada Brosur Teknik Informatika Universitas Tanjungpura Menggunakan Metode *Marker* ”
- Roedavan, Rickman. 2019. *Unity Tutorial Game Engine*. Informatika. Bandung.
- Ronald, T. Azuma. 1997. *A Survey of Augmented Reality*. Jurnal Teleoperators and Virtual Environments.
- Unity Technologies* dibangun pada tahun (2019) oleh *David Helgason (CEO)*, *Nicholas Francis (CCO)*, dan *Joachim Ante (CTO)* di *Copenhagen, Denmark* setelah game pertama mereka *GooBall*
- Widyatmoko (2020) Bahasa Isyarat bahasa yang mengutamakan komunikasi manual, bahasa tubuh, dan gerak bibir, bukannya suara, untuk berkomunikasi