

BAB II

TINJAUAN PUSTAKA

2.1 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi yang dapat digunakan oleh bermacam peranti bergerak. [4] Aplikasi Android dapat ditulis menggunakan bahasa Kotlin, Java, dan C++. Fitur Android SDK mengompilasi coding file bersama data dan file resource menjadi sebuah APK. APK itu sendiri berupa suatu *paket Android*, berbentuk file arsip dengan akhiran .apk. Satu file APK berisi semua konten aplikasi Android dan merupakan file yang digunakan perangkat Android untuk menginstal aplikasi.[5] Pada Unity dengan menggunakan fitur Android SDK file resource seperti splash, scan, tentang app, tentang telekomunikasi akan menjadi sebuah APK dimana apk tersebut dapat langsung diinstall pada handphone android dan dapat digunakan tanpa adanya penggunaan internet. Hasil akhir APK tersebut akan di *publish* ke *play store* sehingga pengguna dapat mendownload aplikasi HOPE secara gratis dimana saja.

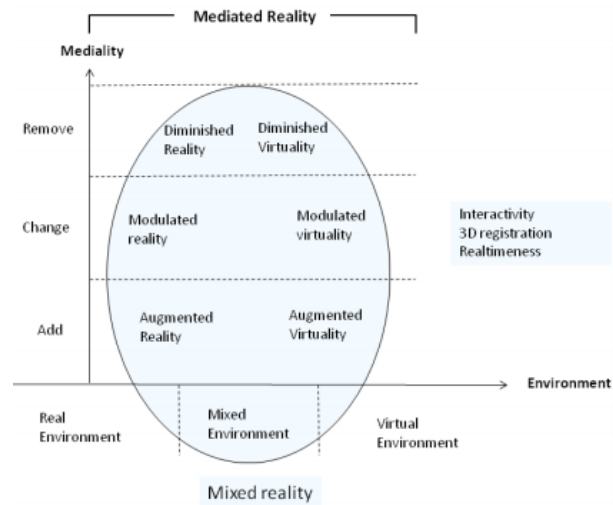
2.2 Augmented Reality (AR)

Augmented reality adalah teknologi yang menggabungkan benda maya dua dimensi ataupun tiga dimensi ke dalam sebuah dunia nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut ke dunia nyata. *Augmented Reality* dapat meningkatkan persepsi manusia, sehingga menjembatani kesenjangan antara ruang nyata dan virtual. Benda fisik dan buatan dicampur bersama tanpa mempengaruhi pergerakan pengguna, Augmented reality sering ditemukan dalam kehidupan sehari-hari.

Augmented Reality pada dasarnya adalah sebuah konsep yang mencitrakan gambar tiga dimensi yang seolah nyata. Proses ini bisa

dirincikan menjadi beberapa proses dan komponen. Untuk mencitrakan gambar tiga dimensi tersebut, sistem *augmented reality* terlebih dahulu harus melakukan penglihatan terhadap lingkungan yang akan dicitrakan objek virtual, kemudian dilakukanlah proses *tracking* terhadap objek spesifik yang menentukan letak citraan objek virtual tersebut. Kemudian, objek tersebut akan dikenali atau dianalisis. Setelah dikenali dan dianalisis maka komputer akan melakukan proses pencitraan objek tersebut, dan akan tampak pada perlengkapan display[6].

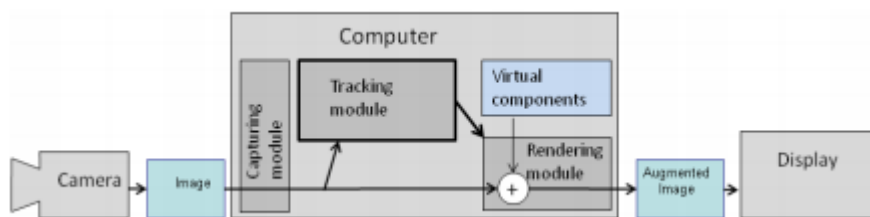
Saat ini sebagian besar definisi *augmented reality* dan *mixed reality* didasarkan pada definisi yang disajikan oleh Milgram, Azuma dan Mann. Namun, kategorisasi tidak tepat dan membatasi area yang berbeda seringkali sulit atau tidak stabil, dan terkadang bahkan kontradiktif. Misalnya, Mann mendefinisikan realitas virtual sebagai sub area realitas campuran, sedangkan Azuma benar-benar memisahkan total virtualitas dari realitas campuran. Mendefinisikan *virtual reality* (VR) sebagai lingkungan imersif yang disimulasikan oleh komputer. Bentuk *virtual reality* yang paling sederhana adalah gambar 3D yang dapat dijelajahi pengguna secara interaktif dari komputer pribadi, biasanya dengan memanipulasi tombol atau mouse. Sistem VR yang canggih terdiri dari layar tampilan bungkus, aktual Ruang VR, komputer yang dapat dikenakan, perangkat haptic, joystick, dll. Dengan memperluas *virtual reality* menjadi *augmented virtuality*, misalnya dengan menambahkan elemen nyata seperti umpan video langsung ke dunia virtual. Aplikasi *augmented reality* sebagian besar berkonsentrasi pada visual *augmented reality* dan sampai batas tertentu pada sensasi sentuhan dalam bentuk umpan balik haptik.



Gambar 2.1 Mediated Reality Taxonomy

Sumber : [Sanni Siltanen, 2012]

2.2.1 Cara Kerja Augmented Reality



Gambar 2.2 Blok Diagram Simple AR

Sumber : [Sanni Siltanen, 2012]

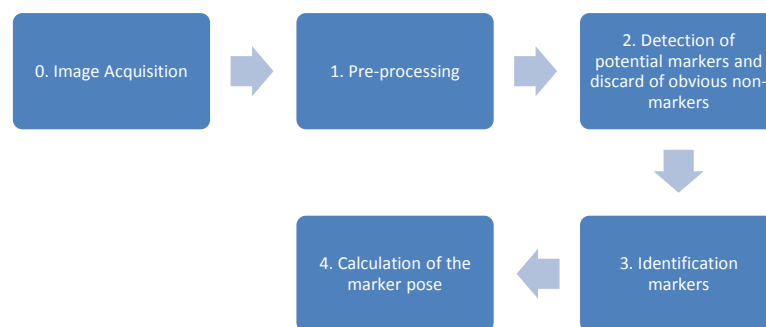
1. Kamera menangkap data dari *marker* dalam dunia nyata dan mengirimkan informasinya ke komputer.
2. *Software* pada komputer akan melacak bentuk kotak dari *marker* dan mendeteksi beberapa video pada *framanya*.
3. Jika kotak telah ditemukan, maka *software* menggunakan perhitungan matematis untuk menghitung posisi dari kamera *relative* terhadap kotak hitam pada *marker*.
4. Setelah dikalkulasi, model grafis akan muncul pada posisi yang sama dan berada didalam lingkup kotak hitam. Lalu ditampilkan ke layar untuk melihat grafis dalam dunia nyata.

2.2.2 Metode Augmented Reality

Ada beberapa metode yang dapat digunakan untuk mendapatkan *Augmented Reality*, namun yang digunakan dalam pembuatan AR pada aplikasi HOPE menggunakan metode *Marker Based Tracking*. Metode ini menggunakan ilustrasi berbentuk persegi panjang berwarna hitam dan juga putih serta terdapat batasan hitam tebal berlatar belakang putih. Pada metode marker based tracking akan digunakan tiga sumbu utama yakni X, Y, dan Z dengan bantuan titik koordinat (0, 0, 0) dan juga virtual secara 3 dimensi. Metode jenis ini mulai dikembangkan dari tahun 1980 an, namun secara khusus dikenal oleh masyarakat sekitar awal tahun 1990 an[7].

a. Prosedur Deteksi Marker

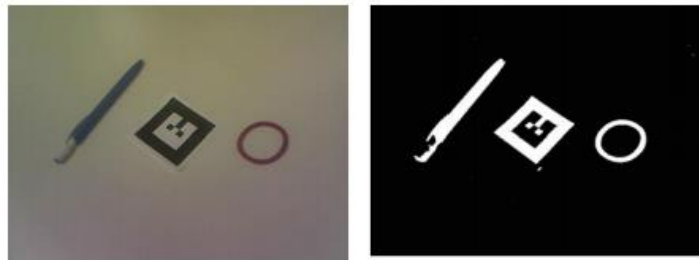
Tujuan pertama dari proses deteksi *marker* adalah menemukan garis besar potensi penanda, lalu menyimpulkan lokasi sudut penanda pada gambar. Selain itu, sistem deteksi perlu memastikan bahwa itu benar-benar sebuah penanda dan menguraikannya identitas. Terakhir, sistem menghitung *pose* menggunakan informasi dari lokasi penanda terdeteksi.



Gambar 2.3 Prosedur deteksi *marker*

0. Image Acquisition : Langkah akuisisi citra sebenarnya merupakan proses terpisah, hanya menyediakan gambar untuk proses deteksi penanda.

1. **Pre-processing** : Sebelum mendeteksi penanda yang sebenarnya, sistem perlu mendapatkan intensitas gambar (gambar abu-abu). Jika sistem mendeteksi penanda dari gambar ambang batas, gambar abu-abu digunakan untuk menemukan tepi atau sudut dengan akurasi lebih tinggi. Sehingga sistem deteksi marker beroperasi dengan gambar abu-abu.



Gambar 2.4 *Pre-processing*

Sumber : [Sanni Siltanen, 2012]

2. **Detection of potential markers and discard of obvious non-markers** : Suatu sistem dapat menolak area yang hanya terdiri dari beberapa piksel. Sering kali terdeteksi selain penanda, bahkan jika itu penanda ukurannya yang kecil akan sulit terdeteksi. Artinya penanda tersebut sangat jauh dari kamera. Dalam hal ini pose penanda akan sangat tidak pasti dan karena itu tidak berguna. Selanjutnya jika file penanda menyusut ke ukuran beberapa piksel, sistem tidak dapat mengidentifikasi penanda, kecuali itu melacak sejarah setiap penampilan penanda.



Gambar 2.5 Deteksi potensi marker

Sumber : [Sanni Siltanen, 2012]

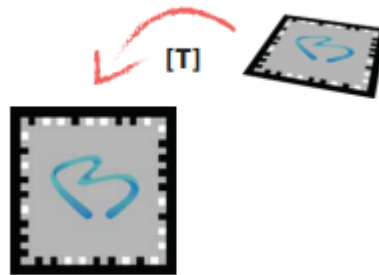
Gambar perspektif persegi selalu berbentuk segi empat. Segiempat memiliki empat garis lurus dan empat sudut yang tepat. Jumlah garis dan sudut mudah dihitung. Oleh karena itu, penerapan deteksi marker sering kali menggunakan salah satunya sebagai kriteria persetujuan atau penolakan cepat. Beberapa kamera menghasilkan gambar yang lebih gelap ke arah tepi luar gambar, efek ini disebut *vignetting*. Dengan menggunakan asumsi vignetting untuk menolak area yang melintasi batas gambar (lihat Gambar 2.4).[8]



Gambar 2.6 Membuang *non-marker*

Sumber : [Sanni Siltanen, 2012]

3. **Identification markers** : Setelah deteksi potensial marker pada gambar masih belum jelas, terlepas dari penerapan sistem pelacakan penanda. Maka marker membutuhkan ciri-ciri tertentu untuk membedakannya dari marker disekitarnya. Ciri-ciri tersebut berfungsi sebagai fitur tambahan identifikasi dapat berupa korelasi, kode digital atau topologi karakteristik. Metode akses seragam untuk isi penanda diperlukan untuk korelasi dan berbasis kode identifikasi. Penanda persegi biasanya dibagi menjadi beberapa yang lebih kecil kotak misalnya 7×7 dimana 5×5 berisi kode dan sisanya mendefinisikan eksternal berbatasan. Untuk dapat mengekstrak kode, proyeksi perspektif penanda perlu untuk dihapus (Gambar 2.5). Transformasi perspektif adalah transformasi yang mengambil sudut penanda di ruang gambar dan menemukan pemetaan yang sebenarnya sudut penanda. [9]



Gambar 2.7 Transformasi perspektif

Sumber : [Hannah Reuterdaahl, 2014]

- 4. Calculation of Marker Pose :** Untuk dapat memvisualisasikan objek 3D dengan benar di dunia nyata, dibutuhkan penaksir pose kamera yang dapat melacak penanda yang ditempatkan di dunia nyata. Saat mengetahui pose kamera tidak bisa hanya posisikan objek 3D dengan benar tetapi pose kamera juga mempertahankan posisinya terlepas dari gerakan kamera. Estimasi pose setara dengan menemukan enam derajat kebebasan, DOF, dari kamera saat ini. Masalah menemukan 6DOF dari n titik korespondensi disebut masalah Perspective-n-Point (PnP) [10]

Ada beberapa pendekatan untuk memecahkan masalah PnP, salah satunya menggunakan Direct Linear Transform (DLT). Konsep dari algoritma DLT adalah memberikan empat simpul yang bisa menyelesaikan hubungan antara koordinat gambar dan koordinat dunia. Umumnya ini berarti menghitung homografi antara dua bingkai. Homografi mendeskripsikan hubungan geometris antara bingkai kamera dan titik minat dalam gambar, misalkan sebuah penanda. sebagai berikut: Diberikan titik (x, y, z) dalam satu gambar dan yang sesuai titik (x', y', z') pada gambar kedua, maka homografi diberikan oleh H:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = H \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \dots\dots\dots(1)$$

Algoritma ini telah dinyatakan sebagai salah satu pengoptimalan metode yang paling andal dalam hal akurasi dan ketahanan terhadap *outlier* dan *noise*. Kalibrasi kamera yang tepat sangat penting untuk inisialisasi kamera pelacak pose. Kalibrasi kamera sering kali dilakukan secara offline sebelum pendaftaran dunia nyata dan akan memberi kita transformasi perspektif dan distorsi lensa dari kamera tertentu yang digunakan.

Prinsip di balik kalibrasi kamera adalah menemukan titik yang diproyeksikan dalam gambar yang diberi titik 3D di dunia nyata. Salah satu metodenya adalah dengan menggunakan gambar papan catur diambil pada beberapa sudut berbeda. Karena dimensi papan catur diketahui kita dapat memperkirakan proyeksi dengan menghitung homografinya di antara bingkai yang berbeda. Hasil kalibrasi kamera disebut matriks intrinsik, K yang menggambarkan parameter kamera seperti panjang fokus, kemiringan dan titik utama:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(2)$$

Parameter ini memberi pemetaan antara gambar ideal dan sensor kamera mengkoordinasikan dan menentukan bagaimana gambar akan terbentuk pada sensor. Mengingat fakta bahwa kamera saat ini memiliki piksel kuadrat dan kolom dan baris sering kali didapatkan lurus;

$$\begin{aligned} s &= 0 \\ \alpha_x &= \alpha_y \end{aligned} \dots\dots\dots(3)$$

Matriks intrinsik tidak cukup untuk menggambarkan transformasi kaku, yang kita butuhkan rotasi dan translasi. Rotasi dan translasi diberikan oleh matriks ekstrinsik yang disebut, E .

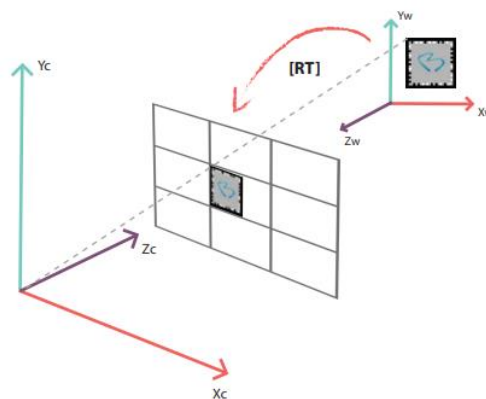
$$E = [R \quad T] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_y \end{bmatrix} \dots\dots\dots(4)$$

Dengan menggabungkan matriks intrinsik, 3x3, dan ekstrinsik, 4x4, kita akan mendapatkan ekspresi akhir menerjemahkan koordinat dunia 3D ke koordinat layar 2D. Matriks ini disebut matriks kalibrasi kamera, C 3x4. Kamera intrinsik parameter akan tetap konstan untuk kamera tertentu sementara parameter ekstrinsik akan berubah seiring dengan pergerakan kamera dan perlu dihitung (menggunakan pendekatan yang disebutkan di atas) untuk setiap frame.

Di bawah ini adalah seluruh ekspresi yang mengubah koordinat dunia 3D menjadi 2D bidang gambar:

$$\begin{bmatrix} x_u \\ y_v \\ 1 \end{bmatrix} = [K] [R \ T] \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} [R \ T] \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \dots\dots\dots(5)$$

Memiliki semua hal di atas, untuk mengetahui efek gabungan dari transformasi objek-ke-dunia nyata, dunia nyata-ke-kamera dan kamera-ke-gambar dan kita dapat memperkirakan posenya dengan benar, (Gambar A.1).



Gambar 2.8 Transformasi dunia nyata ke gambar

Sumber : [Hannah Reuterdaahl, 2014]

Transformasi lebih lanjut mungkin diperlukan dari koordinat gambar ke layar koordinat karena resolusi perangkat dan gambar mungkin tidak sama.

2.3 Unity 3D



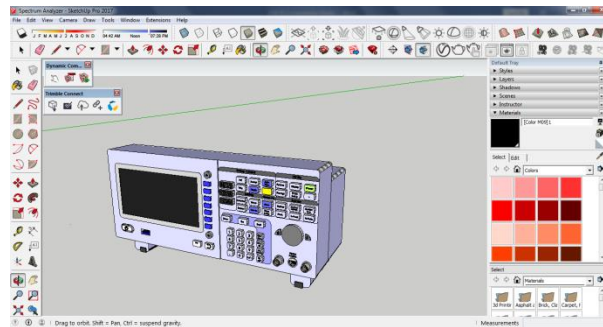
Gambar 2.9 Tampilan Workspace Unity

Sumber : dokumentasi pribadi

Unity adalah suatu aplikasi yang digunakan sebagai *workspace*. Unity *engine* memiliki kerangka kerja (*framework*) lengkap untuk pengembangan profesional. Sistem inti *engine* ini menggunakan beberapa pilihan bahasa pemrograman, diantaranya *C#*, *javascript*, dan *boo*. Unity 3D *editor* menyediakan beberapa alat untuk mempermudah pengembangan seperti *unity tree* dan *terrain creator* yang dapat mempermudah pembuatan vegetasi dan terrain serta MonoDevelop dalam proses pemrograman. Jenis Unity yang digunakan Unity version 2018.4.23f1 Personal. Dimana system requirements untuk operating system minimal windows 7 SP1+ 64-bit version only.

QR Code, *Object 3D* dan *audio* akan di bentuk menjadi aplikasi augmented reality berbasis android pada *Unity*. Di *Unity* pula dapat membentuk halaman splash screen, menu, scan, panduan, tentang telekomunikasi dan tentang app. Setelah dilakukan *coding* di *Unity* maka selanjutnya akan dilakukan proses *compile*, hasil proses *compile* tersebut akan menghasilkan *file format* berkas yang dapat di *install* pada sistem operasi *android* berbentuk file APK.

2.4 SketchUp



Gambar 2.10 Tampilan Workspace SketchUp

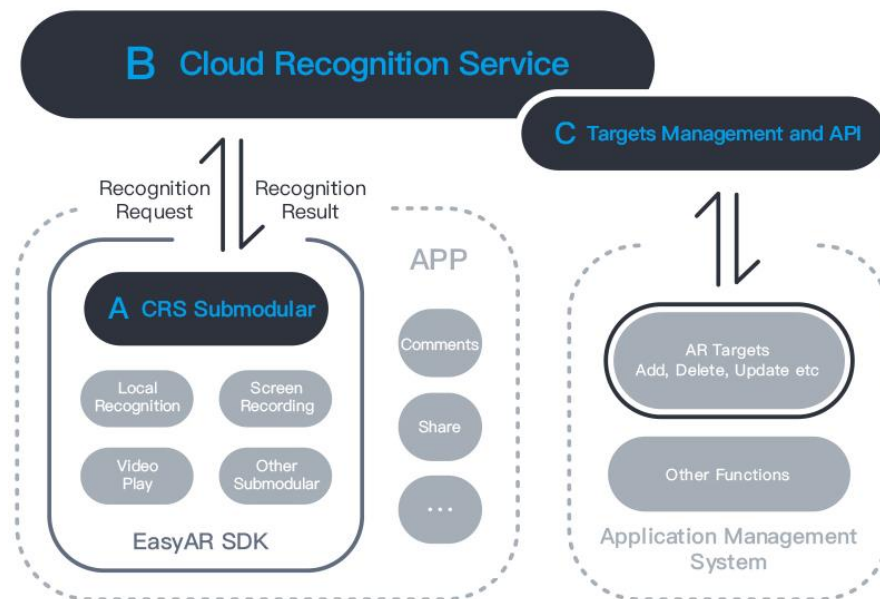
Sumber : dokumentasi pribadi

SketchUp merupakan salah satu software yang mempunyai fungsi dalam desain grafis model 3 dimensi yang digunakan dan dirancang untuk para profesional di bidang teknik sipil, arsitektur, dalam pembuatan game, film, dan rancangan yang terkait didalamnya. Dalam perancangan object 3 dimensi pada aplikasi HOPE digunakan SketchUp 2017 yang mendukung windows 7 64-bit. Object 3D dibentuk satu demi satu kemudian di export kedalam bentuk .fbx yang dapat dibuka Unity nantinya.

2.5 Easy AR

EasyAR Cloud Recognition Service (CRS) adalah layanan web yang memungkinkan keunggulan komputasi cloud dan penyimpanan, untuk memberikan pengenalan gambar berbasis cloud untuk basis data skala besar. EasyAR CRS menyediakan pengambilan gambar dan manajemen target sebagai layanan web. Pada aplikasi HOPE menggunakan versi EasyARUnitySensePlugin 4.0.1. dimana dengan fitur-fitur yang dimiliki EasyAR kita dapat memberikan image target pada QR Code dan Image tracker pada object 3D dan audio. Sehingga ketika kamera dihadapkan pada QR Code akan mendeteksi dan menampilkan object 3D beserta audio.

2.5.1 Block Diagram of EasyAR CRS



Gambar 2.11 Block Diagram of EasyAr CRS

Sumber : easycr.com/

2.5.2 Fitur pada EasyAR CRS

a. Offline Cache for Cloud Recognition

Target yang Dikembalikan dari Cloud Dapat Disimpan Sebagai Lokal, Target untuk Penggunaan Offline

b. Cloud Database with Large Capacity

Target 100 ribu per basis data untuk biaya-efektif lebih tinggi
Dapat diperpanjang per permintaan pengguna

c. Fast, Accurate Recognition

Pengenalan cepat dalam 0,1s
Akurasi pengenalan tinggi hingga 98%

d. Efficient API

API yang dirancang sederhana dan efektif membuat pengembangan lebih nyaman

e. Smooth Management

Antarmuka manajemen grafis yang mudah digunakan memberikan manajemen yang mudah untuk basis data cloud.

f. Online Target Rating

Menganalisis kualitas untuk pengakuan dan kesamaan terhadap seluruh database dengan laporan detail untuk target gambar.

2.6 QR Code



Gambar 2.12 *QR Code*

Sumber : en.wikipedia.org/wiki/QR_code

Barcode 2D diciptakan pada awal 1990-an, tidak hanya memegang encoding yang lebih luas dan kapasitas yang lebih besar dari barcode 1D, tetapi juga rahasia, anti-pemalsuan dan koreksi kesalahan dalam kasus-kasus tertentu. Oleh karena itu, barcode 2D telah diterapkan secara luas ke berbagai bidang sejak pembuatannya. Dari lebih dari 30 jenis barcode 2D yang saat ini tersedia di pasaran, QR Code memiliki kelebihan dari banyak barcode 2D yang berbeda, dan telah banyak digunakan selama beberapa tahun di negara seperti Jepang.[11]

QR Code ditemukan oleh Denso Wave yang didirikan pada tahun 1994, di mana "QR" mengacu pada "Respon Cepat." menurut situs resmi Denso Wave. QR Code tidak hanya memiliki kapasitas tinggi, tetapi juga dapat menyimpan berbagai jenis informasi. Bahkan memiliki tingkat koreksi kesalahan yang tinggi. yang berarti bahwa bahkan 30% simbol sudah tercemar atau sobek. Kode QR masih dapat dibaca dengan benar. berikut ini, fitur dan aplikasi QR Code dijelaskan.

2.6.1 QR Code Generator

Adalah alat yang sederhana dan nyaman yang membantu dalam pembuatan citra QR Code ditampilkan pada layar. QR code generator ini dapat di temukan di aplikasi play store maupun di website secara gratis. QR Code dapat diisi dengan text, Url, Email, Nomor telepon, kontak Geolocation dan SMS.

2.7 Object 3D



Gambar 2.13 Kendaraan Bajaj Sebagai Object 3D

Sumber : freelancer.co.id/

Objek 3 dimensi (3D) adalah sebuah benda/ruang yang memiliki panjang, lebar dan tinggi yang memiliki bentuk. 3D tidak hanya digunakan dalam matematika dan fisika saja melainkan dibidang grafis, seni, animasi, komputer dan lain - lain. Konsep tiga dimensi atau 3D menunjukkan sebuah objek atau ruang memiliki tiga dimensi geometris yang terdiri dari: kedalaman, lebar dan tinggi. Contoh tiga dimensi suatu objek / benda adalah bola, piramida atau benda spasial seperti kotak sepatu. Istilah "3D" juga digunakan untuk menunjukkan representasi dalam grafis komputer (digital), dengan cara menghilangkan gambar stereoscopic atau gambar lain dalam pemberian bantuan, dan bahkan efek stereo sederhana, yang secara konstruksi membuat efek 2D (dalam perhitungan proyeksi perspektif, shading). Saat ini 3D digambarkan untuk mensimulasikan perhitungan berdasarkan layar proyeksi dua dimensi dan efek tiga-dimensi seperti monitor komputer atau televisi.[12]

2.8 Peralatan Telekomunikasi yang di jadikan Object 3D

Pada perancangan objek 3 dimensi aplikasi HOPE digunakan 9 objek utama yang akan dijadikan model dalam pembuatan objek 3D yaitu :

Tabel 2.1 Perangkat Objek 3D pada Aplikasi HOPE

No	Nama Objek
1.	Spectrum Analyzer
2.	Radio Detection and Ranging
3.	Base Transceiver Station (BTS)
4.	Satellite
5.	Access Point
6.	Fiber Optic
7.	Internet
8.	Router
9.	Switch

2.9 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (Integrated Development Environment/IDE) resmi untuk pengembangan aplikasi Android. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi Android, seperti:

- Sistem build berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android
- Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- Framework dan fitur pengujian yang lengkap

- Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine

2.10 Photoshop CC

Adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan, bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi oleh Adobe Systems. Pada pembuatan design halaman hingga icon Aplikasi HOPE menggunakan Photoshop CC.

2.11 Audio

Audio memiliki dua cara penggunaan berbeda dalam augmented reality: sebagai bagian dari antarmuka pengguna atau untuk augmentasi aural. Pada pembuatan aplikasi HOPE audio yang ditambah dan antarmuka audio hanya sebagian dari fungsinya. Sistem menggunakan pengenalan karakter optik (OCR) dan teknik computer vision untuk mendeteksi objek dan membaca teks. Kamera mendeteksi penanda dan menampilkan objek kemudian aplikasi membaca informasi dengan nyaring.

2.12 Perbandingan Penelitian Sebelumnya

Sebagai bahan pertimbangan pembuatan tugas akhir, diperlukan untuk mengetahui penelitian-penelitian yang telah dilakukan sebelumnya, antara lain sebagai berikut:

Tabel 2.2 Perbandingan Penelitian Sebelumnya

No	Penulis	Tahun Jurnal	Platform	Kelebihan	Kekurangan
1.	Dias Aziz Pramudita [13]	2015	Dekstop	Melakukan pengujian yang lebih rinci dari pada penelitian yang lain. Menggunakan teknologi facetracking.	Aplikasi sangat bergantung dengan lingkungan, intensitas cahaya, kecepatan gerak wajah, dan jarak wajah ke camera.
2.	Gede Yudha Prema Pangestu [18]	2017	Web	Memiliki object 3D dan design brosur yang sangat interaktif.	Kamera tidak leluasa dalam mendeteksi marker karena kamera yang digunakan kamera webcam.
3.	Alfi Syahrin [19]	2016	Android	Aplikasi menggunakan metode marker based tracker, pengujian jarak dilakukan pada pixel dan warna.	Perlunya pengujian tambahan, sulitnya aplikasi ditemukan secara umum.
4.	Ika Devi Perwitasari [20]	2018	Adroid	Aplikasi menggunakan metode marker based tracker, memiliki object 3D yang banyak	Perlunya pengujian tambahan. Sulitnya aplikasi ditemukan secara umum
5.	Ardiana Ratnasari [7]	2017	Android	Menggunakan model prototype, algoritma yang digunakan FAST <i>corner detection</i> ,	Tampilan aplikasi kurang interaktif.

2.13 Pengujian

2.13.1 Pengujian Performa Marker

Pengujian ini dilakukan untuk mengetahui efektivitas marker pada aplikasi HOPE yang telah dibuat. Pada pengujian performa marker akan diuji beberapa faktor yang mempengaruhi marker yaitu sudut, pencahayaan dan jarak.

2.13.2 Pengujian Software

Ada beberapa macam metode yang digunakan untuk menguji tingkat kelayakan suatu aplikasi, yaitu:

2.13.2.1 Black Box Testing (Fungsional)

Black box testing adalah teknik pengujian tanpa harus mengetahui cara kerja aplikasi secara interior. Tester mengabaikan arsitektur sistem dan tidak memiliki akses hingga ke dalam source code-nya. Biasanya, saat melakukan pengujian menggunakan metode black box, tester hanya akan berinteraksi dengan UI sistem dengan memberikan input dan menganalisis outputnya tanpa mengetahui bagaimana dan dimana input dikerjakan oleh sistem. Pada pengujian black box testing terdapat salah satu metode yang bernama pengujian alpha dimana pengujian alpha adalah metode black box yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun[13].

2.13.2.2 Pengujian Usability (Non-Fungsional)

Usability sebagai ukuran kualitas pengalaman pengguna ketika berinteraksi dengan produk atau sistem apakah situs web, aplikasi perangkat lunak, teknologi bergerak, maupun peralatan-peralatan lain yang dioperasikan oleh pengguna. Usability sebagai tingkat dimana produk bisa digunakan oleh pengguna tertentu untuk mencapai tujuannya dengan lebih efektif, efisien, dan memuaskan dalam ruang lingkup penggunaannya. Menurut Jacob Nielsen dalam bukunya *Usability*

Engineering, Ada 5 syarat yang harus dipenuhi agar suatu aplikasi mencapai tingkat *usability* yang ideal[14], yaitu:

1. *Learnability* (Mudah dipelajari)
2. *Efficiency* (Efisien)
3. *Memorability* (Kemudahan dalam mengingat)
4. *Errors*(Pencagahan kesalahan)
5. *Satisfaction* (Kepuasan pengguna)

Sebuah sistem yang memiliki *usability* yang rendah dapat mengakibatkan pemborosan waktu, penurunan produktifitas, meningkatkan frustrasi dan kecenderungan tidak ingin kembali mengunjungi atau menggunakan sistem yang ada. Untuk mengetahui tingkat *usability* sebuah sistem perlu dilakukan evaluasi terhadap sistem tersebut.

Aturan perhitungan skor untuk berlaku pada 1 responden. Untuk perhitungan selanjutnya, skor SUS dari masing-masing responden dicari skor rata-ratanya dengan menjumlahkan semua skor dan dibagi dengan jumlah responden. Berikut rumus menghitung skor SUS:

$$\bar{X} = \frac{\sum x}{n} \dots\dots\dots(6)$$

Dimana :

\bar{X} = Skor rata-rata

$\sum x$ = jumlah skor SUS

n = jumlah responden