

BAB I

PENDAHULUAN

A. Latar Belakang

Di era digital saat ini, informasi dan pengetahuan semakin mudah diakses melalui berbagai platform, termasuk buku fisik dan digital. Dengan semakin banyaknya buku yang diterbitkan, baik di toko buku maupun perpustakaan, menjadi tantangan tersendiri bagi pembaca untuk mencari buku yang dengan proses pencarian informasi pada umumnya dilakukan oleh orang berdasarkan kata kunci berupa teks. Setelah kata kunci dalam bentuk teks diinputkan pengguna, maka hasil informasi yang dimaksud akan tampil. Namun proses pencarian informasi dengan menggunakan teks dapat terjadi kesalahan dalam penulisan, sehingga informasi yang dimaksud bisa saja tidak sesuai dengan yang diinginkan oleh pengguna. Proses pencarian informasi lainnya dapat juga dilakukan dengan pencarian berdasarkan gambar, sehingga tingkat validasi informasi yang diperoleh lebih besar karena tidak perlu menuliskan hanya memasukkan gambar (Muh Sya'bani, 2018).

Kebutuhan untuk sistem yang efisien dalam mengidentifikasi dan mengelola buku menjadi semakin penting. Deteksi buku berdasarkan sampul adalah salah satu metode inovatif yang dapat mempermudah pengelolaan dan pencarian buku dengan memanfaatkan teknologi komputer.

Augmented reality adalah teknologi yang menggabungkan benda maya dua dimensi ataupun tiga dimensi ke dalam sebuah lingkungan nyata lalu memproyeksikan benda-benda maya tersebut secara realitas dalam waktu nyata. Realitas tertambah dapat diaplikasikan untuk semua indra, termasuk pendengaran, sentuhan, dan penciuman. Selain digunakan dalam bidang-bidang seperti kesehatan, militer, industry manufaktur maupun dunia pendidikan. Terknologi *AR* ini dapat menyisipkan suatu informasi tentu ke dalam dunia maya dan menampilkannya di dunia nyata dengan bantuan perlengkapan seperti *unity*, *smartphone Android* (Abd.Rasman, 2022).

Oleh karena itu, pendeteksian buku berdasarkan sampul menjadi penting, karena dapat membantu pembaca untuk dengan cepat memilih buku-buku yang sesuai dengan minat dan selera mereka.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka didapatkan rumusan masalah yaitu bagaimana merancang aplikasi deteksi buku berdasarkan sampul berbasis *augmented reality*?

C. Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan dengan harapan penelitian terfokus dengan tujuan yang dibuat. Adapun tujuan dalam penelitian sebagai berikut:

1. Untuk mengidentifikasi dan mengenali buku secara otomatis melalui analisis gambar sampul.
2. Mengklasifikasikan buku berdasarkan genre atau kategori menggunakan fitur visual pada sampul

3. Menciptakan aplikasi yang memudahkan pengguna dalam menemukan informasi buku hanya dengan memindai sampulnya.

D. Batasan Penelitian

Penelitian ini memiliki beberapa batasan dengan harapan penelitian terfokus dengan batasan-batasan yang dibuat. Adapun batasan batasan dalam penelitian sebagai berikut:

1. Efektivitas deteksi sangat bergantung pada resolusi dan kualitas gambar sampul yang digunakan
2. Penelitian mungkin terbatas pada sejumlah sampul buku tertentu, sehingga tidak mencakup variasi yang luas.
3. Pencahayaan latar belakang saat pengujian dapat mempengaruhi akurasi deteksi.
4. Hanya diuji pada perangkat tertentu, seperti smartphone dengan ARKit atau ARCore.
5. Mungkin ada batasan pada jumlah objek yang dapat ditampilkan bersamaan tanpa menurunkan performa.

E. Manfaat Penelitian

Dengan adanya aplikasi ini, besar harapan untuk dapat memberikan manfaat bagi semua kalangan :

1. Bagi pengguna

Memudahkan pengguna dalam mengenal buku untuk memberikan informasi mengenai detail buku.

2. Bagi penulis

Memberi kesempatan pada penulis untuk mengaplikasikan ilmu dan teori yang dipelajari selama ini. Selain itu, diharapkan dapat menambah wawasan tentang cara kerja sistem deteksi buku .

3. Bagi pembaca

Peneliti ini diharapkan mampu memberikan pengetahuan kepada pembaca mengenai informasi buku.

F. Sistem Penulisan

Adapun system penulisan pada proposal ini sebagai berikut:

1. BAB I : PENDAHULUAN

Bab pendahuluan berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II : TINJAUAN PUSTAKA

Bab ini membahas teori yang berhubungan dengan referensi dalam pemesanan undangan digital.

3. BAB III : METODE PENELITIAN

Dalam bab ini penulis menjelaskan tentang metode penelitian pengumpulan data yang dilakukan dalam membuat aplikasi.

4. BAB IV : HASIL DAN PERANCANGAN SISTEM

Dalam bab ini menjelaskan tentang hasil dari penelitian serta menjelaskan sistem berjalannya aplikasi.

5. BAB V : PENUTUP

Dalam bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

A. Kajian Teori

1. Deteksi

Menurut Febriyan M.Fajar,ST (2014) dalam Fajar Syawal (2020) Deteksi merupakan proses dimana algoritma bekerja untuk mengenali informasi - informasi berdasarkan gambar yang di ambil dari frame kamera video, kemudian membandingkan informasi tersebut dengan informasi yang berada pada database.

2. Sampul

Menurut penerbitdeepublis.com (2021) cover atau sampul buku mempunyai peran penting dalam pembuatan buku karena tidak hanya untuk menjadi pembungkus naskah, namun sampul buku bisa mendongkrak penjualan dari sebuah buku dan menjadi penentu untuk menarik minat dari sang pembaca. Rustan (2009) juga mengatakan bahwa cover buku mempunyai penangan khusus dikarenakan mempunyai porsi emphasis yang besar. Berikut merupakan elemen-elemen yang ada pada cover buku:

- a. Yang pertama ada teks judul sebagai penjabaran tentang apa yang dibahas pada buku.
- b. Yang kedua ada sub judul sebagai penjelasan dari judul.
- c. Yang ketiga gambar latar belakang untuk dijadikan pendukung utama atau background.

- d. Yang keempat ada ornamen sebagai tambahan untuk menambah emphasis pada desain
- e. Yang kelima logo sebagai identitas dari perusahaan penerbit maupun perorangan.

Rustan (2009) menyebutkan bahwa sampul buku dibagi menjadi 3 bagian yaitu elemen visual, teks & paragraf, dan teknik fotografi:

- a. Dalam pembuatan cover elemen visual kita membuat dengan gaya komik maupun kartun seperti karya buku dari Jest Ninney yang berjudul *Journal of a Sneaky Twerp*
- b. Untuk teks dan paragraf mempunyai variasi emphasis dengan judul, sub judul dan juga elemen visual lainnya yang akhirnya menjadi sequence.
- c. Menggunakan teknik fotografi dan keseimbangan asimetris untuk membuat cover yang menarik dengan menggabungkan elemen visual dan juga teks.

3. Buku

Buku bisa diartikan sebagai kumpulan kertas yang dijilid dan biasanya berisi dengan tulisan maupun kosong. Untuk pengertian dari ahli, buku merupakan hasil dari pemikiran dan juga analisa yang kemudian disusun dengan kata-kata agar semua orang bisa memahaminya. (Kurniasih,2014)

informasi tentang buku secara umum, biasanya ada di perpustakaan atau bisa Anda temui di berbagai sumber:

- a. Judul : Nama lengkap dari buku tersebut. Ini adalah elemen utama yang digunakan untuk mengidentifikasi buku.
- b. Penulis : Nama penulis atau pengarang buku. Buku sering kali dikaitkan dengan reputasi atau karya-karya terkenal dari penulis tersebut.
- c. Penerbit : Nama penerbit yang menerbitkan buku tersebut. Penerbit bisa memberikan gambaran tentang kredibilitas buku, terutama jika penerbit tersebut sudah terkenal di bidangnya.
- d. Tahun terbit : Tahun pertama kali buku tersebut diterbitkan. Ini dapat membantu mengidentifikasi relevansi buku tersebut dengan perkembangan pengetahuan atau budaya pada saat itu.
- e. Kota terbit : Kota terbit merujuk pada kota tempat penerbit menerbitkan atau mencetak buku. Ini adalah bagian dari informasi bibliografis yang sering dicantumkan pada halaman judul buku. Kota terbit ini membantu memberikan konteks terkait asal buku tersebut, karena terkadang buku yang diterbitkan di kota tertentu memiliki ciri khas atau konteks budaya tertentu.
- f. Halaman : Jumlah halaman buku. Ini memberi gambaran tentang panjang buku tersebut.

4. Perpustakaan

Perpustakaan adalah sebuah institusi atau lokasi yang menyediakan akses kepada berbagai jenis bahan bacaan dan sumber informasi untuk digunakan oleh masyarakat umum atau anggota tertentu. Ini merupakan pusat pengetahuan

yang penting dalam mendukung proses pembelajaran, riset, dan kebutuhan informasi bagi individu maupun kelompok

Klasifikasi buku di perpustakaan untuk mengorganisir dan mengelompokkan buku-buku berdasarkan topik, subjek, atau genre tertentu. Sistem klasifikasi ini memudahkan pengunjung perpustakaan untuk mencari dan menemukan buku sesuai dengan tema atau bidang yang mereka inginkan. Klasifikasi yang paling umum digunakan adalah Sistem Klasifikasi Dewey (*Dewey Decimal Classification*) atau disingkat DDC dan Rak buku ini biasanya diatur sesuai dengan sistem klasifikasi buku yang digunakan oleh perpustakaan.

Sistem ini paling sering digunakan di perpustakaan umum dan sekolah. Sistem Dewey mengelompokkan buku ke dalam 10 kelas utama berdasarkan subjek atau tema. Setiap kelas utama dibagi lagi menjadi subkelas lebih spesifik. Kelas-kelas utama dalam Dewey Decimal Classification:

- a. Kelas 000 - Ilmu Pengetahuan Umum
Meliputi: Ensiklopedia, buku referensi, jurnal, dan komputer.
- b. Kelas 100 - Filsafat dan Psikologi
Meliputi: Filsafat, etika, logika, psikologi, dan sosiologi.
- c. Kelas 200 - Agama
Meliputi: Buku tentang agama, teologi, mitologi.
- d. Kelas 300 - Ilmu Sosial
Meliputi: Ilmu politik, hukum, ekonomi, pendidikan, dan etika sosial.
- e. Kelas 400 - Bahasa

Meliputi: Pembelajaran bahasa, linguistik, dan penerjemahan.

f. Kelas 500 - Ilmu Alam

Meliputi: Matematika, astronomi, biologi, dan ilmu fisika.

g. Kelas 600 - Teknologi Terapan

Meliputi: Kedokteran, pertanian, teknik, dan teknologi lainnya.

h. Kelas 700 - Seni dan Hiburan

Meliputi: Seni rupa, musik, teater, dan olahraga.

i. Kelas 800 - Sastra

Meliputi: Sastra umum, sastra dunia, puisi, drama, dan cerita pendek.

j. Kelas 900 - Sejarah dan Geografi

Meliputi: Sejarah dunia, geografi, biografi, dan buku perjalanan.

Kategori buku di perpustakaan merujuk pada pengelompokan atau pembagian buku berdasarkan tema, subjek, atau jenisnya. Kategori ini bertujuan untuk mempermudah pengunjung dalam mencari buku sesuai dengan minat atau kebutuhan informasi mereka. Dengan adanya kategori, buku-buku di perpustakaan dapat diorganisir dengan sistematis dan memudahkan pengunjung menemukan bahan bacaan yang relevan.

Koleksi buku di perpustakaan merujuk pada semua jenis buku yang dimiliki dan dikelola oleh perpustakaan untuk disediakan kepada pengunjung atau anggota perpustakaan. Buku-buku ini merupakan sumber utama informasi dan pengetahuan di perpustakaan. Koleksi buku dapat bervariasi dalam hal topik,

jenis, dan format, tergantung pada tujuan perpustakaan tersebut. Jenis Koleksi Buku di Perpustakaan umum yaitu:

- a. Buku fiksi adalah karya sastra yang bersifat imajinatif, seperti novel, cerita pendek, puisi, dan drama. Buku fiksi ini bertujuan untuk menghibur dan memberikan pengalaman emosional kepada pembaca.
- b. Buku non-fiksi menyajikan informasi yang faktual dan dapat dipertanggungjawabkan. Buku ini dapat mencakup berbagai topik, mulai dari ilmu pengetahuan, sejarah, hingga biografi.
- c. Buku referensi adalah buku yang digunakan untuk mencari informasi atau referensi spesifik.
- d. Koleksi buku untuk anak-anak disusun dengan bahasa yang mudah dimengerti dan sering kali disertai ilustrasi. Buku-buku ini bertujuan untuk mendidik sekaligus menghibur.

5. Augmented Reality

Menurut Bimber dan Raskar (2017), dalam Abd.Rasman (2022), *Augmented Reality* berarti mengintegrasikan informasi sintesis kedalam lingkungan nyata. Menurut penjelasan Haller Dkk (2017), riset *Augmented Reality* bertujuan untuk mengembangkan teknologi yang mengiakan penggabungan secara real-time terhadap digital *content* yang dibuat oleh komputer dengan dunia nyata. Tidak seperti teknologi Virtual reality (VR) yang membawa user sepenuhnya ke dalam lingkungan sintesis, *Augmented Reality* memperbolehkan penggunaan melihat objek maya tiga dimensi maupun video yang diproyeksikan terhadap dunia nyata. *Augmented Reality* pada dasarnya adalah

sebuah konsep yang mencitrakan gambar tiga dimensi yang seolah nyata. Proses ini bisa diperinci menjadi beberapa proses dan komponen.

Untuk mencitrakan gambar tiga dimensi tersebut, sistem *Augmented Reality* terlebih dahulu harus melakukan penglihatan atau vision terhadap lingkungan yang padanya 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 posisi dan orientasinya, maka komputer akan melakukan proses pencitraan objek tersebut, dan akan tampak pada perangkat *display* komponen penting yang harus ada adalah:

- a) Perangkat tampilan (*Display*)
- b) Alat *tracking* (pencarian)
- c) Peralatan input
- d) Perangkat komputer

Perlengkapan tampilan digunakan untuk menampilkan informasi gambar atau objek tiga dimensi yang dicitrakan terhadap dunia nyata tempat user melihat. Perlengkapan tampilan terbagi menjadi tiga jenis, yakni *Head Mounted Display*, *Handheld Display*, dan *Spatial Display*. *Head Mounted Display* adalah perlengkapan tampilan yang dikenakan dikepala *user* dan digunakan sebagai ‘kecamata’ untuk melihat dunia nyata, yang telah digabungkan dengan objek virtual yang telah diregistrasikan dalam sistem, *Handheld Display* adalah perlengkapan ringkas yang dibawa-bawa ke mana saja dan dapat dimuat di tangan, contohnya adalah *smartphone* dan *android phone*. *Spatial Display* adalah

sistem pencitraan yang menggunakan proyektor digital untuk mempetakan informasi grafis pada objek fisik. Hal yang paling membedakan *Spatial Display* adalah bahwa pencitraannya tidak terasosiasi dengan setiap individu *user*, namun secara berkelompok.

Tracking biasanya dilakukan dengan teknologi-teknologi menangkap gambar, misalnya dengan kamera digital, sensor optis lainnya, GPS, kompas dan lain sebagainya. Selain itu, alat *tracking* yang sekarang meningkat popularitanya adalah *webcam*, karena praktis, kecil, mudah dibawa dan diatur untuk dijalankan. Peralatan input hingga sekarang ini masih banyak menjadi objek penelitian. Hingga saat ini, alat yang digunakan mencakup alat '*pinch glove*' tongkat bertombol atau peralatan handheld seperti *smartphone*. Perangkat komputer, terutama dengan CPU yang kuat dan jumlah RAM yang cukup besar untuk memproses gambar yang ditangkap. Sistem yang digunakan untuk mobilitas biasanya menggunakan laptop yang dilengkapi dengan *webcam*, sementara untuk yang bersifat diam menggunakan *workstation* dengan kartu grafis yang kuat.

Implementasi perhitungan pada teknologi *Augmented Reality* proses ini dilakukan perhitungan untuk mendapatkan koordinat wisata sehingga bisa ditampilkan dalam layar *handphone*.

a. Perhitungan sudut azimuth

Dalam sistem navigasi terdapat istilah azimuth yaitu sudut antara utara manetis (nol derajat) dengan titik/sasaran yang kita tuju relatif terhadap sudut utara. Sudut *azimuth* pada masing-masing titik dihitung berdasarkan sudut arctangent antara perbedaan latitude dengan lokasi pengguna dibagi dengan

perbedaan longitude. Setelah mendapatkan sudut *azimuth* pada setiap titik menampilkan titik-titik lokasi wisata. Sudut *azimuth* di-update terus menerus sehingga bisa mewakili posisi imajiner di depan pengguna saat pengguna melakukan rotsi atau memutar layar *handphone* kekiri atau kekanan.

b. Perhitungan sudut inklinasi

Sudut inklinasi merupakan sudut vertical yang dibentuk antara titik tempat pengguna menggunakan aplikasi dengan titik ketinggian suatu lokasi dapat diketahui secara akurat. Setelah didapatkan ketinggian masing-masing titik langkah berikutnya adalah menentukan sudut vertical antara 2 titik tersebut dengan rumus trigonometri. Sensor akselerometer melakukan pembacaan terhadap orientasi *handphone* yang dipegang pengguna. Jika pengguna memegang dengan posisi layar menghadap depan dan rata dengan tanah maka ikon wisata akan ditampilkan sejajar terhadap horizon. Namun jika pengguna memegang device pada posisi 45 derajat kebawah maka ikon wisata muncul pada layar bagian atas. Sumbu – X dalam representasi dunia nyata adalah garis horizontal, sumbu – Y merupakan garis vertical dan sumbu – z mengarah kearah depan layar sedangkan area dibelakang layar memiliki nilai Z negatif.

6. Smartphome

Menurut Bimber dan Raskar (2017), dalam Abd.Rasman (2022), Ponsel cerdas (Bahasa Inggris : smartphine) adalah telepon geggam 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 berkerja menggunakan seluruh perangkat

lunak system operasi yang menyediakan hubungan standar 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 electronic (e-book) atau terdapat papan ketik (baik sebagaimana jadi maupun dihubung keluar) 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. Pengembangan ini lantas mendorong terciptanya apa yang kini dikenal sebagai smartphone.

7. Bahasa Pemrograman C#

Menurut Bimber dan Raskar (2017), dalam Abd.Rasman (2022), C# (*baca: C sharp atau see sharp*) sering dianggap sebagai bahasa penerus C++ atau 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 berorientasi pada objek yang dikembangkan oleh *Microsoft* dan menjadi salah satu bahasa pemrograman yang mendukung *INet programming* melalui *Visual Studio*.

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 lama untuk menguasainya, karena bagaimana juga C# lebih sederhana dibandingkan bahasa-bahasa pemrograman seperti C++ dan *Java*.

C# didesain oleh program *designer* dari *Microsoft*, *Anders Hajlsberg*. Sebelum berkerja pada *Microsoft*. 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 C++, *Delphi*, *Java*, dan *Smaltalk*, 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*.

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 pembuatan 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, *from 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 kegunaannya dengan baik.

e. C# bersifat modular

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

8. Android (SDK)

(Kasmir 2018) Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler, mencakup sistem operasi, *middleware*, dan aplikasi. Sistem ini awalnya dikembangkan oleh sebuah perusahaan kecil di Silicon Valley bernama Android Inc. Pada tahun 2005, *Google* mengadopsi Android dan mengumumkannya sebagai sistem operasi "*Open Source*". Akibatnya, siapa pun dapat menggunakannya secara gratis, termasuk kode sumber yang diperlukan untuk mengkompilasi sistem operasi tersebut. Android kini menjadi sistem operasi dengan jumlah pengguna terbanyak di dunia, karena sifatnya yang terbuka memungkinkan para *developer* untuk membuatnya lebih fungsional.

Android menyediakan lingkungan yang unik untuk pengembangan aplikasi. Semua aplikasi beroperasi pada level yang sama, tanpa perbedaan antara aplikasi inti dan aplikasi pihak ketiga. *Application Programming Interface (API)* yang disertakan memberikan akses ke perangkat keras serta data ponsel cerdas atau data sistem itu sendiri. Pengguna bahkan memiliki kemampuan untuk menghapus aplikasi inti dan menggantinya dengan aplikasi pihak ketiga. Adapun Kelebihan dari android yaitu:

- a) *Platform* yang komprehensif: Menawarkan berbagai alat yang bermanfaat untuk menciptakan aplikasi, yang dapat terus dikembangkan oleh pengembang.
- b) *Platform Open Source*: Karena bersifat terbuka, pengembang memiliki kemudahan dalam mengembangkan Android.

- c) *Platform* gratis: Pengembang bebas untuk mengembangkan, mendistribusikan, dan menjual sistem operasi Android tanpa perlu membayar royalti atau memperoleh lisensi.

9. Unity

Menurut Bimber dan Raskar (2017), dalam Abd.Rasman (2022), 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. Tentu Unity menyediakan untuk free user dan bisa di publish dalam bentuk Standalone (.exe) dan web. Untuk saat ini Unity sedang mengembangkan berbasis AR (Augment Reality). Unity ini tidak hanya untuk membuat game 3D saja, namun 2D juga.

Unity Teknologi dibangun pada tahun 2004 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 bantau 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 “Demokrasi Pembangunan Game” dan membuat sebuah pengembangan game baik 2D maupun 3D bisa dicapai oleh banyak orang. Oleh siapa pun. Pada tahun

2008, 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.

10. Marker

Menurut Bimber dan Raskar (2017), dalam Abd.Rasman (2022), Marker adalah object dunia nyata yang sudah di inisialisai sebelumnya, yang nantinya akan si kenali oleh system sebagai wahana 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, 2023).

1. Metode Pengenalan Pola Gambar

Gualcomm sebagai salah satu pengembang *Augmented Reality* melakukan proses pendeteksian marker menggunakan pengenalan pola gambar. Metode yang digunakan dalam GCAR 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*), kemudian 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 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 dibandingkan dengan komputer. Akibatnya, keterbatasan ponsel dalam setiap aspek harus diperhitungkan. Ketika mengembangkan sebuah teknologi AR. Banyak teknologi AR yang dirancang untuk ponsel, menggunakan penana berbasis pelacakan untuk menimbulkan estimasi dan ekstraksi. Pendekatan ini berkerja 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 inci dapat dengan mudah dilacak.

Dengan 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 antara pengamat (mata atau kamera) dan adegan. Korespondensi fitur bekerja lebih banyak dan efektif daripada pencocokan template karena mereka bergantung pada pencocokan template karena mereka bergantung pada pencocokan fitur local. Mengingat korespondensi tersebut, fase secara kasar dapat dihitung dengan estimasi yang kuat yang membuatnya cukup sensitive terhadap oklusi parsial, blur, refleksi, perubahan skala, kemiringan, pembaruan iluminasi atau kesalahan kecocokan. 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 berkerja 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 signifikasi modifikasi untuk membuat sebuah system pelacakan cocok untuk ponsel.

11. Firebase

Firebase adalah platform pengembangan aplikasi yang menyediakan berbagai layanan backend sebagai layanan (BaaS) yang dapat digunakan oleh pengembang untuk membangun aplikasi web dan mobile dengan lebih cepat dan efisien. Platform ini awalnya dikembangkan oleh Firebase, Inc. dan kemudian diakuisisi oleh Google pada tahun 2014. Firebase menawarkan berbagai fitur yang mencakup autentikasi pengguna, database real-time, hosting, pengiriman notifikasi, dan lainnya, yang memungkinkan pengembang untuk fokus pada pengembangan aplikasi tanpa harus mengelola infrastruktur backend secara langsung.

Layanan Utama dalam Firebase yang disediakan oleh Firebase:

a. Firebase Authentication

Firebase Authentication adalah layanan yang memungkinkan pengembang untuk mengelola proses autentikasi pengguna dengan mudah. Layanan ini mendukung berbagai metode autentikasi seperti email dan kata sandi, login sosial (Google, Facebook, Twitter), serta autentikasi menggunakan nomor telepon atau penyedia lain. Firebase Authentication menyediakan SDK untuk Android, iOS, dan web yang memungkinkan integrasi autentikasi dalam aplikasi dengan sedikit kode.

b. Firebase Realtime Database dan Cloud Firestore

Firebase menyediakan dua jenis database utama untuk penyimpanan data aplikasi: Realtime Database dan Cloud Firestore. Keduanya adalah database NoSQL yang memungkinkan pengembang menyimpan data dalam format

JSON dan mendukung pembaruan data secara real-time. Realtime Database lebih sederhana dan terstruktur sebagai pohon JSON, sementara Cloud Firestore memberikan fleksibilitas lebih dengan menyimpan data dalam bentuk dokumen dan koleksi yang lebih terstruktur dan dapat diskalakan.

c. Firebase Cloud Storage

Firebase Cloud Storage memungkinkan pengembang untuk menyimpan dan melayani konten besar seperti gambar, video, dan file lainnya. Layanan ini terintegrasi dengan Firebase Authentication dan menyediakan kontrol akses berbasis aturan, sehingga hanya pengguna yang berwenang yang dapat mengakses atau mengubah file.

d. Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging adalah layanan untuk mengirimkan notifikasi push kepada pengguna Android, iOS, atau aplikasi web. FCM memungkinkan pengembang untuk mengirimkan pesan secara langsung kepada pengguna, yang berguna untuk pemberitahuan, pembaruan, atau pesan yang disesuaikan.

e. Firebase Hosting

Firebase Hosting adalah layanan untuk hosting aplikasi web statis, termasuk HTML, CSS, JavaScript, dan file lainnya. Layanan ini memberikan pengiriman konten dengan HTTPS secara default dan memungkinkan pengembang untuk meng-host aplikasi mereka di server global Google yang memiliki kecepatan tinggi.

f. Firebase Cloud Functions

Firebase Cloud Functions memungkinkan pengembang untuk menulis dan menjalankan kode serverless di cloud, yang bisa merespons event-event tertentu seperti perubahan data di database, pengiriman pesan, atau permintaan HTTP. Fungsi ini tidak memerlukan pengelolaan server, dan hanya dijalankan berdasarkan kebutuhan.

g. Firebase Analytics

Firebase Analytics adalah alat untuk melacak dan menganalisis interaksi pengguna dengan aplikasi. Layanan ini menyediakan laporan mendalam mengenai perilaku pengguna, termasuk aktivitas di dalam aplikasi, demografi pengguna, dan efektivitas kampanye pemasaran. Firebase Analytics terintegrasi dengan layanan Firebase lainnya, seperti Cloud Messaging dan Remote Config.

Firebase memiliki sejumlah keuntungan yang menjadikannya pilihan populer bagi pengembang aplikasi, antara lain:

- 1) Mudah digunakan: Firebase menyediakan SDK yang sederhana dan dokumentasi lengkap yang memudahkan integrasi fitur ke dalam aplikasi.
- 2) Pengelolaan serverless: Pengembang tidak perlu mengelola server atau infrastruktur backend, yang memungkinkan fokus pada pengembangan fitur aplikasi.
- 3) Real-time sync: Dengan Realtime Database dan Cloud Firestore, Firebase memungkinkan sinkronisasi data secara langsung antara server dan perangkat pengguna dalam waktu nyata.

- 4) Skalabilitas: Firebase dirancang untuk menangani skala aplikasi besar, dengan kemampuan untuk menangani lonjakan trafik yang besar tanpa memerlukan perubahan besar pada arsitektur aplikasi.
- 5) Keamanan: Firebase menyediakan kontrol akses yang kuat melalui Firebase Authentication dan aturan keamanan berbasis peran di database dan storage.

12. Vuforia SDK (Software Development Kit)

Menurut Febriyan M.Fajar (2014) dalam Muhammad Basri (2020). 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. Objek 3 dimensi akan nampak secara langsung dilayar smartphone, sehingga akan terlihat objek 3 dimensi berada didalam dunia nyata.

Vuforia SDK mendukung berbagai 2D dan 3D jenis jenis sasaran termasuk markeless image target, Konfigurasi 3D Multi-Target, dan bentuk beralamat Fidusia Marker dikenal sebagai Bingkai Marker. Fitur tambahan dari SDK termasuk local Occlusion deteksi menggunakan Virtual Buttons, pemilihan taget 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 beberapa Batasan tentang kaligrafi, tetapi jika kata tersebut dicetak atau digital dilayar komputer software tidak menunjukkan masalah.

a. Cloud penyimpanan

Perangkat lunak ini juga memungkinkan pengguna data awan disimpan. Fitur ini bisa sangat berguna bagi orang-orang yang tidak memiliki masalah kinerja pada ponsel mereka. Muhammad Basri (2020).

b. 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. Muhammad Basri (2020).

c. Tombol virtual

Perangkat lunak memiliki build-in maya manajemen tombol. Fitur ini bisa sangat menarik untuk menambah lebih banyak interaksi antara pemain dan spidol. Muhammad Basri (2020).

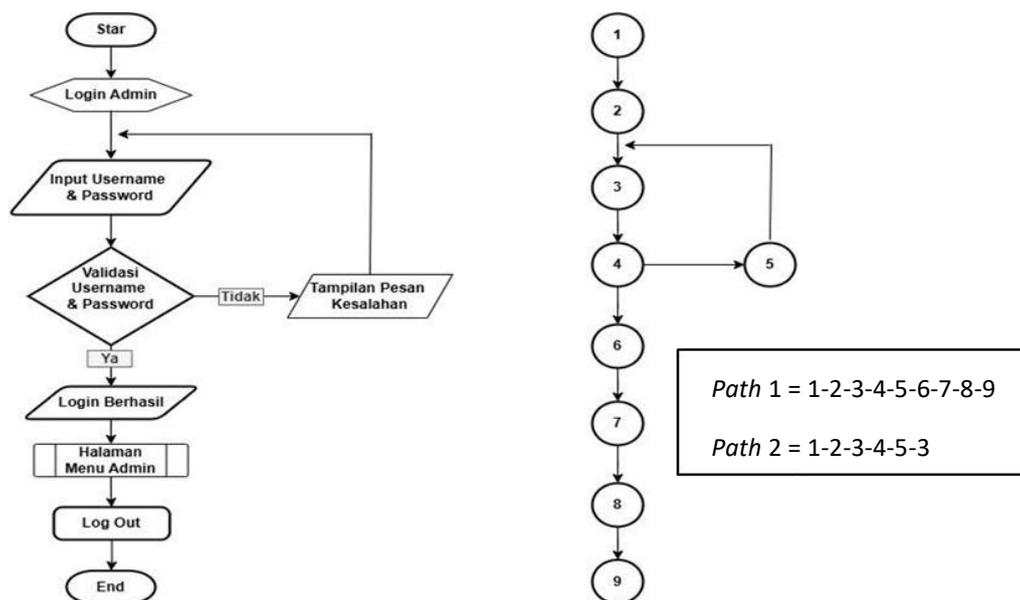
13. *Whitebox testing*

Whitebox testing bertujuan untuk mengetahui apakah struktur pada aplikasi yang dibuat sudah sesuai dengan ketentuan. Whitebox testing menitik beratkan pada pengujian dengan mengecek detail perancangan perangkat lunak. Whitebox

testing dinilai dengan mendefinisikan semua alur dari perangkat lunak, kemudian membangun kasus yang akan digunakan dalam proses pengujian, kemudian menguji kasus tersebut untuk memperoleh hasilnya.

Berikut ini adalah contoh pengujian white box pada sistem login yang terdapat pada Sistem Informasi laporan Data Statistik Desa Pananrang berbasis

Web. (Asmil. (2020)), yaitu :



Gambar 2. 1 Contoh *Whitebox Testing*

Perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki region = 2.

menghitung *cyclomatic complexity* dari edge dna node, dengan rumus :

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

$$E(\text{edge}) = 9, N(\text{node}) = 9$$

$$\text{Penyelesaian : } V(G) = 9 - 9 + 2$$

$$V(G) = 2$$

Jadi jumlah path pada *flowgraph* diatas adalah 2. Independent path

pada *flowgraph* diatas adalah :

Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9

Path 2 = 1 - 2 - 3 - 4 - 5 - 3

Tabel 2. 1 *Whitebox testing*

	1	2	3	4	5	6	7	8	9	E-1
1	1									1-1=0
2		1								1-1=0
3			1		1					2-1=1
4				1						1-1=0
5					1					1-1=0
6						1				1-1=0
7							1			1-1=0
8								1		1-1=0
9									1	1-1=0
SUM (E+ 1)										1+1=2

14. *Blackbox testing*

Blackbox testing terfokus pada fungsional dari program yang ada. Pada Blackbox testing diuji dengan cara menjalankan program kemudian diamati apakah program tersebut berhasil atau tidak. Blackbox testing menggunakan teknik equivalence partitions yang merupakan pengujian berdasarkan masukan setiap menu yang terdapat pada program, setiap menu masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasar fungsinya.

Berikut ini adalah pengujian black box pada sistem Monitoring Inventory Control pada Asuransi Jiwa Bersama (AJB) Bumipuetra untuk fungsi menu Laporan (2016), yaitu sebagai berikut:

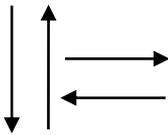
Tabel 2. 2 Contoh *Blackbox testing*

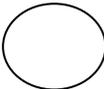
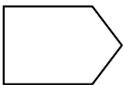
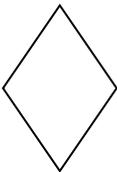
No	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
1	<p>Tambah Laporan (Data yang di <i>input</i> tidak lengkap), lalu klik "Simpan"</p> <p><i>Test Case :</i></p> 	<p>Sistem tidak akan menyimpan ketika kolom tidak terisi semua, maka akan menampilkan pesan "Data belum lengkap"</p> <p>Hasil Pengujian :</p> 	Valid
2	<p>Tambah Data laporan <i>input</i> data Dari Tanggal dan S/D Tanggal lalu klik <i>Print</i></p> <p><i>Test Case :</i></p> 	<p>Sistem sukses, berhasil mencetak Laporan maka akan menampilkan Laporan Perperiode yang diinginkan</p> <p>Hasil Pengujian :</p> 	Valid

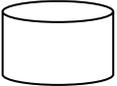
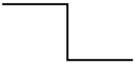
15. Flowchart

Flowchart (diagram alir) merupakan diagram yang memuat simbol - simbol grafis yang menyatakan aliran algoritma atau proses dari langkah - langkah instruksi dalam bentuk kotak persegi dan bulat dan pernyataan instruksi, dimana hubungan dan urutan proses tiap instruksi ditunjukkan dengan simbol tanda panah. (Anggrawan, 2018)

Tabel 2. 3 *Flowchart*

Simbol Flowchart	Nama Simbol	Penjelasan
	Terminator (<i>Start & End</i>)	Simbol untuk tanda mulai (<i>start</i>) dan tanda selesai (<i>stop/end</i>) dari kegiatan proses.
	Arah Aliran	Simbol yang menghubungkan antara simbol yang satu dengan simbol lainnya (atau antara kegiatan proses) dan sekaligus menyatakan arah proses.
	Persiapan (<i>Preparation</i>)	Simbol pemberian harga awal.

Simbol Flowchart	Nama Simbol	Penjelasan
	Keluaran / Masukan (<i>Input Output</i>)	Simbol yang menyatakan proses input dan output (berlaku untuk semua media <i>input</i> dan <i>output</i>)
	Proses (<i>Procces</i>)	Simbol yang melambangkan kegiatan pemrosesan / pengelolaan input.
	Operasi Secara Manual	Simbol menyatakan proses / pengelolaan yang tidak dilakukan oleh komputer.
	Konektor / Tanda Sambung	Simbol untuk tanda penyambungan proses pada lembar atau halaman yang sama.
	Konektor / Tanda Sambung	Simbol untuk tanda penyambungan proses pada lembar atau halaman berbeda.
	Percabangan atau Pilihan Keputusan	Simbol proses pemilihan keputusan tergantung kondisi, jika pemeriksaan kondisi terpenuhi benar, maka jalur pilihan yang diproses adalah jalur ya atau <i>yes</i> dan sebaliknya jika pemeriksaan kondisi tidak terpenuhi tidak benar, maka jalur yang harus dipilih adalah jalur tidak atau <i>no</i> .
	Prosedur	Simbol ini memiliki peran sebagai blok proses flowchart yang berdiri sendiri diluar flowchart utama (berupa subprogram prosedur atau fungsi). Pada simbol ini di flowchart cukup tuliskan nama prosedurnya atau fungsinya saja, atau pada saat memanggil prosedur atau fungsi dari program utama (main program), cukup panggil dengan nama prosedur/ fungsi.

Simbol Flowchart	Nama Simbol	Penjelasan
	Dokumen	Simbol ini menyatakan output dicetak pada kertas atau berbentuk dokumen.
	Drum Magnetik	Simbol untuk <i>input</i> / <i>output</i> yang gunakan media drum magnetik.
	Disk Magnetik	Simbol untuk <i>input</i> / <i>output</i> yang gunakan media disk magnetik.
	Penyimpan Daring (<i>Online Storege</i>)	Simbol <i>input</i> / <i>output</i> yang menggunakan media penyimpanan online.
	Kartu Berlubang (<i>Punch Card</i>)	Simbol yang mewakili bahwa input berasal dari data kartu dan output dicetak ke kartu.
	Display	Simbol <i>input</i> / <i>output</i> yang menggunakan pita magnetik.
	Magnetik Tape	Simbol tranmisi data lewat kanal komunikasi.

16. Unified Modelling Language (UML)

UML (*Unified Modeling Language*) adalah sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasi dan melakukan spesifikasi pada sistem. (Pitman, 2005).

Tabel 2. 4 Diagram UML

No	Diagram	Tujuan
1	<i>Class</i>	Untuk menangkap detail tentang entitas yang membentuk sistem dan hubungan statis di antara mereka.
2	<i>Component</i>	Menunjukkan organisasi dan dependensi yang terlibat dalam implementasi suatu sistem.
3	<i>Deployment</i>	Menunjukkan bagaimana sistem benar-benar dieksekusi dan ditugaskan ke berbagai perangkat keras.

No	Diagram	Tujuan
4	<i>Package</i>	Tipe diagram kelas yang sangat spesial.
5	<i>Object</i>	Menunjukkan bagaimana sintaksis yang sama dengan diagram kelas dan tunjukkan bagaimana contoh aktual dari kelas terkait pada waktu tertentu.
6	<i>Activity</i>	Menunjukkan aliran dari satu perilaku aktivitas, ke perilaku berikutnya.
7	<i>Communication</i>	Menunjukkan interaksi yang berfokus pada elemen - elemen yang terlibat dalam perilaku tertentu dan pesan apa yang mereka bolak – balik.
8	<i>Sequence</i>	Menunjukkan interaksi yang menekankan jenis dan urutan pesan yang dikirimkan antar elemen selama eksekusi.
9	<i>State machine</i>	Untuk menangkap transisi keadaan internal suatu elemen.
10	<i>Use Case</i>	Untuk menangkap persyaratan fungsional untuk suatu sistem

Jenis - jenis diagram UML :

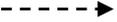
a. *Use Case Diagram*

Use case adalah salah satu konsep inti dari pemodelan berorientasi objek.

Use case digunakan untuk mewakili fungsionalitas tingkat tinggi dan bagaimana pengguna akan menangani sistem. Simbol - simbol dari *Use case* diagram ditampilkan seperti pada tabel dibawah ini :

Tabel 2. 5 Simbol - Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	suatu <i>use case</i> harus diprakarsai oleh seseorang atau sesuatu ruang lingkup <i>use case</i> .
2		<i>Dependency</i>	Dalam hubungan semacam ini, elemen sumber tergantung pada elemen bawahannya.

No	Gambar	Nama	Keterangan
3		<i>Generalization</i>	Elemen anak bergantung atau berbagi perilaku dan struktur data terhadap elemen yang ada di atasnya.
4		<i>Include</i>	Untuk sumber elemen secara <i>eksplisit</i> .
5		<i>Extend</i>	Perilaku dari elemen target dapat diperluas melalui titik pada elemen sumber.
6		<i>Association</i>	Sekumpulan tautan yang menghubungkan elemen - elemen dari model UML.
7		<i>System</i>	Untuk menampilkan sistem secara terbatas.
8		<i>Use case</i>	Untuk mewakili fungsionalitas tingkat tinggi dan bagaimana pengguna akan menangani sistem
9		<i>Collaboration</i>	Elemen yang bertitik dengan nama yang tertulis di dalamnya
10		<i>Note</i>	Untuk menggambarkan bagian fisik dari suatu sistem.

b. Activity Diagram

Activity diagram adalah diagram yang berfungsi untuk berbagai kegiatan yang dilakukan oleh berbagai komponen sistem. itu dilambangkan sama dengan diagram mesin negara. *Activity* diagram berisikan keadaan awal, keadaan akhir, kotak keputusan, dan notasi tindakan. Simbol - simbol dari *activity* diagram ditampilkan seperti pada tabel dibawah ini :

Tabel 2. 6 Simbol - Simbol *Activity* Diagram

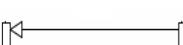
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Suatu perilaku yang diperhitungkan dalam satu tindakan atau lebih.

No	Gambar	Nama	Keterangan
2		<i>Action</i>	Suatu tindakan mewakili satu langkah dalam suatu kegiatan di mana manipulasi atau pemrosesan data terjadi dalam sistem yang dimodelkan.
3		<i>Initial Node</i>	Menunjukkan kontrol tempat aliran dimulai saat aktivitas dipanggil
4		<i>Activity Final Node</i>	Menjelaskan kontrol yang menghentikan semua aliran dalam suatu kegiatan.
5		<i>Fork Node</i>	Untuk membagi aliran masuk menjadi beberapa aliran keluar secara bersamaan.

c. *Sequence Diagram*

Sequence diagram menggambarkan interaksi yang menekankan jenis dan urutan pesan yang dikirimkan antar elemen selama eksekusi. Simbol - simbol dari *sequence diagram* ditampilkan seperti pada tabel dibawah ini :

Tabel 2. 7 Simbol - Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		Life Line	Suatu objek yang bisa saling berinteraksi.
2		Message	Pesan informasi suatu objek terhadap objek yang lainnya yang telah terjadi.
3		Message	Pesan informasi suatu objek terhadap objek yang lainnya yang telah terjadi.

B. Kajian Hasil Penelitian Terdahulu

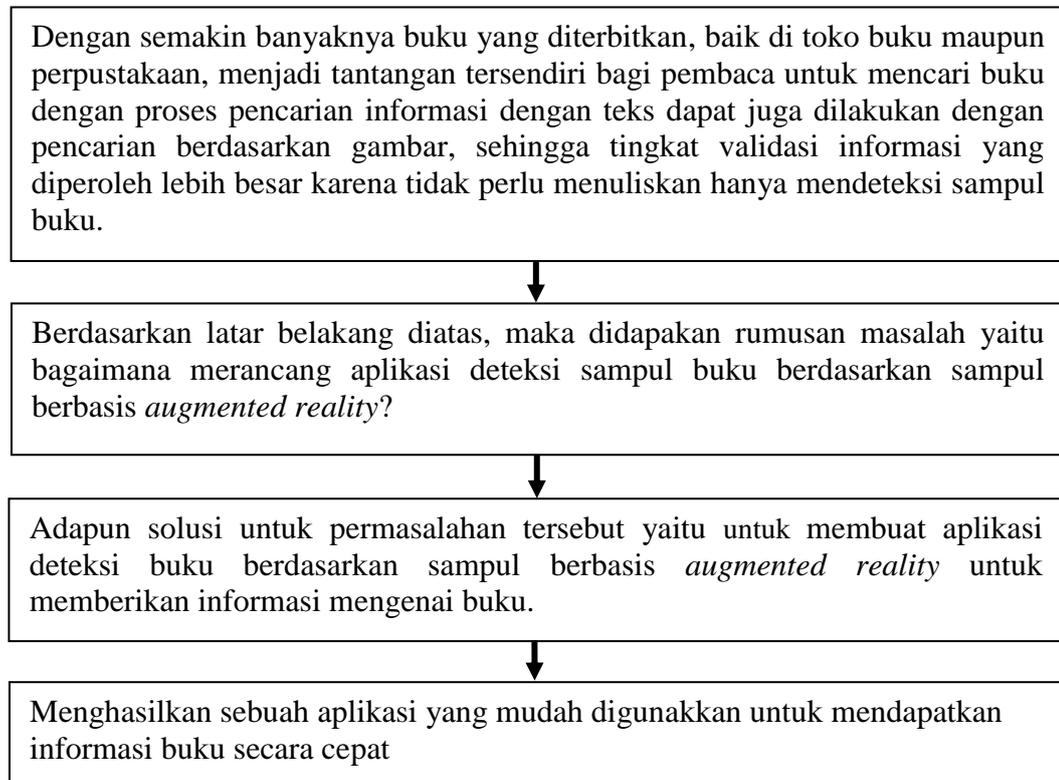
“APLIKASI DETEKSI TIPE RUMAH SUBSIDI BERBASIS AUGMENTED REALITY ” oleh Masyhuddin. S (2023). Pada penelitian ini berfokus menampilkan informasi rumah subsidi berbasis *augmented reality* yang memudahkan masyarakat meneliti desain rumah subsidi tersebut.

“DETEKSI IDENTITAS BERDASRKAN WAJAH” oleh Abd. Rasman (2022). Pada penelitian ini memberikan terobosan pengenalan wajah dengan menggunakan metode *centroid* dan *geometri* mean untuk kategori wajah *frontall*.

“APLIKASI PENDETEKSI KHASIAT DAUN BERBASIS AUGMENTED REALITY” oleh Muhammad Basri (2020). Pada penelitian aplikasi ini dengan memanfaatkan fitur Augmented reality yang di sediakan oleh Vuforia dalam proses pendeteksian objek guna untuk menampilkan informasi

C. Kerangka Berpikir

Untuk memahami kerangka pikir penelitian ini dapat dilihat pada gambar :



BAB III

METODE PENELITIAN

A. Jenis Penelitian

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

B. Lokasi Dan Waktu

Kegiatan penelitian ini dilakukan di Perpustakaan Umum Kota Parepare dan waktu penelitian dilakukan selama 3 (tiga) bulan.

C. Alat Dan Bahan

Dalam pelaksanaan penelitian ini digunakan beberapa alat dan bahan yang terdiri dari Perangkat Keras (Hardware) dan Perangkat Lunak (Software).

Perangkat Keras yang digunakan terdiri Personal Computer (PC/Laptop) ASUS dengan spesifikasi yaitu Processor dengan Intel Celeron N4000(4M Cache, up to 2.6 GHz, Ram dengan 4 GB RAM, LCD Monitor dengan 14 inch, dan Hardisk dengan 1 TB.

Sedangkan Perangkat Lunak yang digunakan yaitu Sistem operasi Windows, Unity, Android, Vuforia, dan Firebase

D. Teknik 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.

E. Metode pengujian

Metode pengujian data pada aplikasi ini ada 2 (dua) yaitu:

1. ***Whitebox testing***

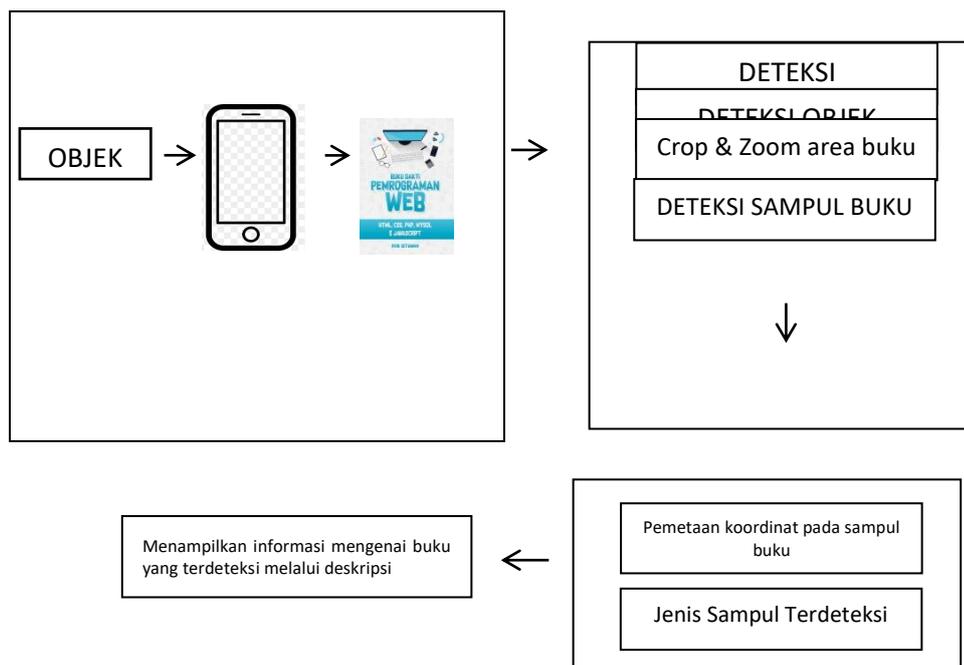
White box Testing disebut sebagai pemeriksaan struktur. Penguji dapat melihat dengan tepat program mana yang sedang diuji. Karena struktur internal kode diketahui dengan menguji setiap bagian yang dapat dievaluasi untuk mengidentifikasi masalah logis dalam kode sumber perangkat lunak, pengujian

dibuat dari sudut pandang pengembang. Jovanovic dalam jurnal (Praniffa et al., 2023).

2. *Blackbox testing*

Tujuan *Black box Testing* adalah untuk mengevaluasi setiap fitur infrastruktur program aplikasi. *Black box Testing* bertujuan untuk menentukan apakah perangkat lunak beroperasi sebagaimana mestinya tanpa memiliki akses ke kode sumber yang sebenarnya (Ismail & Efendi, 2020).

F. Digram Alir



Gambar 3. 1 Diagram Sistem

Pada aplikasi deteksi buku berdasarkan sampul ini merupakan desain sistem dari sistem yang akan dibuat nantinya. Dalam pembuatan aplikasi ini, dilakukan deteksi gerak dan objek untuk menentukan ada atau tidaknya objek yang diharapkan pada tiap frame yang diproses. Apabila sampul buku sudah

terdeteksi, maka langkah berikutnya adalah mencari posisi atau bentuk sampul buku untuk kemudian dipetakan pada koordinat layar handphone. Dengan aplikasi ini dapat mendeteksi sampul buku dan menampilkan informasi buku dalam bentuk deskripsi.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Sistem Yang Diusulkan

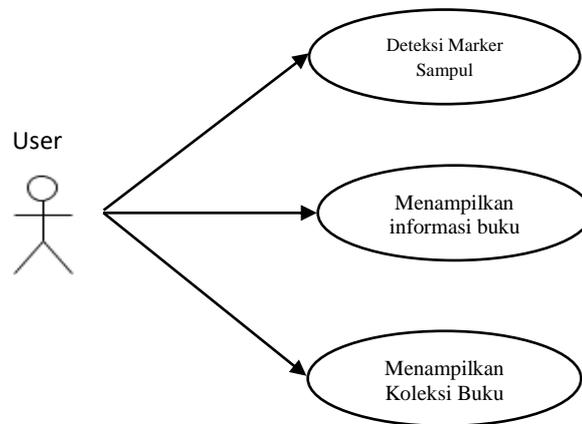
Analisis sistem merupakan suatu kegiatan penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.

Analisis Sistem yang diusulkan adalah sebuah sistem yang dapat memudahkan *user* dalam mengetahui informasi buku hanya dengan mendeteksi sampul buku tersebut. Dengan adanya aplikasi ini diharapkan dapat lebih menarik minat masyarakat untuk mengetahui informasi buku.

Rancangan sistem tersebut penulis gambarkan menggunakan diagram UML yaitu *use case* diagram, *activity* digram, *sequence* diagram.

1. *Use Case* Diagram

Use Case diagram adalah diagram yang menggambarkan hubungan antar actor dengan sistem *Use Case* diagram mendeksripsikan interaksi antara satu atau lebih actor dengan sistem yang dibuat (Purnasari et al., 2022).



Gambar 4. 1 *Use Case Diagram*

Gambar desain sistem yang diusulkan diatas menjelaskan pada aplikasi ini ketika user melakukan pencarian buku user mengarahkan smartphone diatas marker, ketika valid aplikasi akan menampilkan informasi pada buku. User juga bisa melihat koleksi buku.

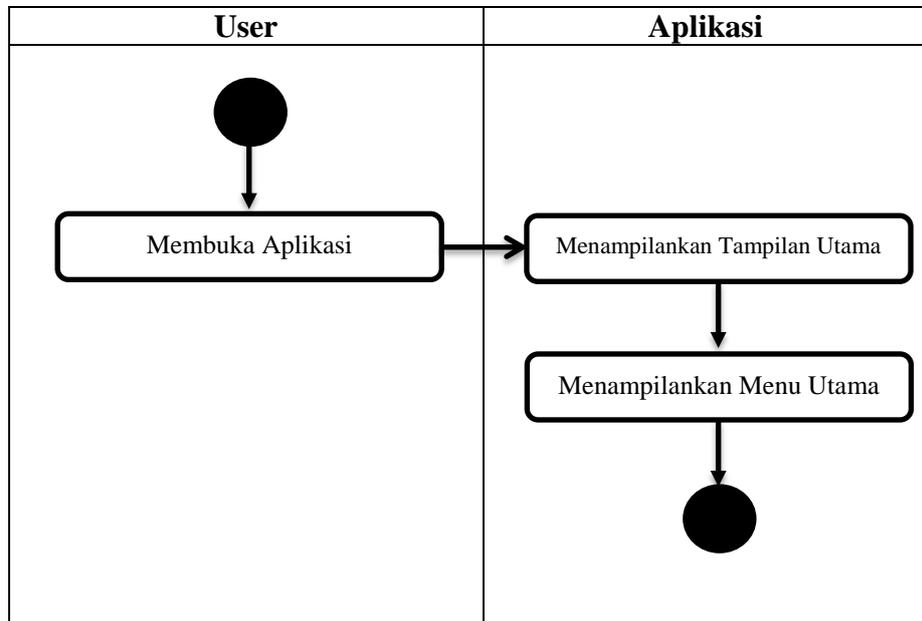
Tabel 4. 1 Penjelasan *Use Case Diagram*

Nama Use Case	Deskripsi Use Case
Deteksi Marker Sampul	Use case ini menjelaskan bahwa user melakukan pendeteksi marker sampul
Menampilkan informasi buku	Use case ini menjelaskan bahwa setelah melakukan mendeteksi marker akan menampilkan informasi buku
Menampilkan Koleksi Buku	Use case ini menjelaskan bahwa ketika user memilih koleksi buku akan menampilkan koleksi-koleksi buku yang ada di aplikasi tersebut.

2. Activity Diagram

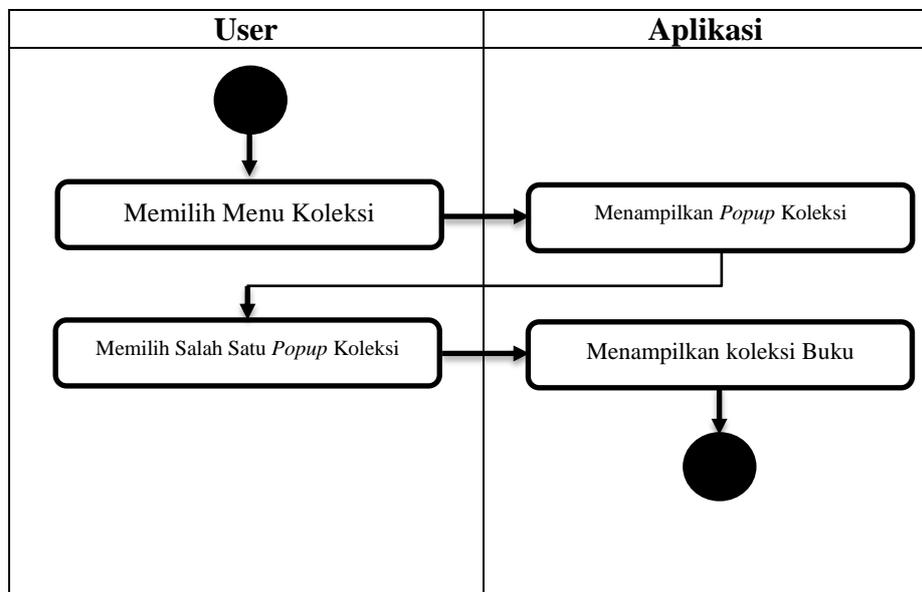
Activity diagram berarti diagram yang dapat mensimulasikan proses apa yang terjadi dengan sistem. Gambar vertikal menunjukkan rangkaian proses suatu sistem (Syach Putra et al., 2024).

a. *Activity Diagram Tampilan Utama*



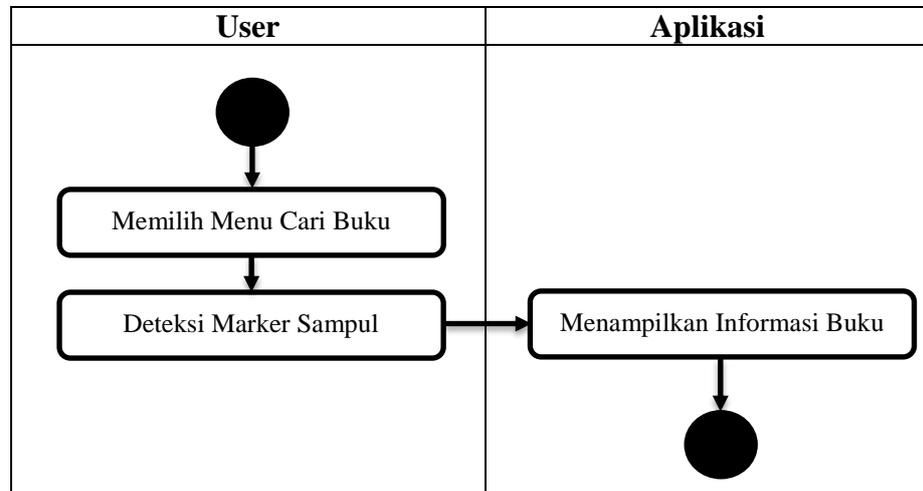
Gambar 4. 2 *Activity Diagram Tampilan Utama*

b. *Activity Diagram Koleksi*



Gambar 4. 3 *Activity Diagram Koleksi Buku*

c. *Activiy Diagram Cari Buku*

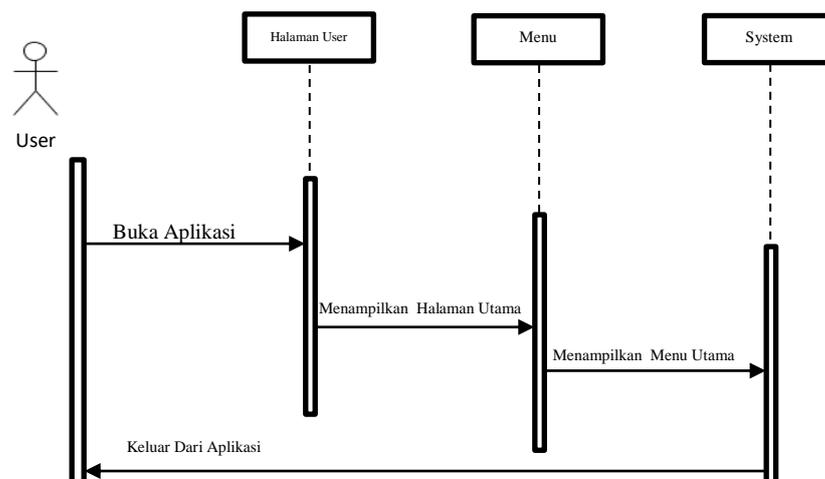


Gambar 4. 4 *Activiy Diagram Cari Buku*

3. *Sequence Diagram*

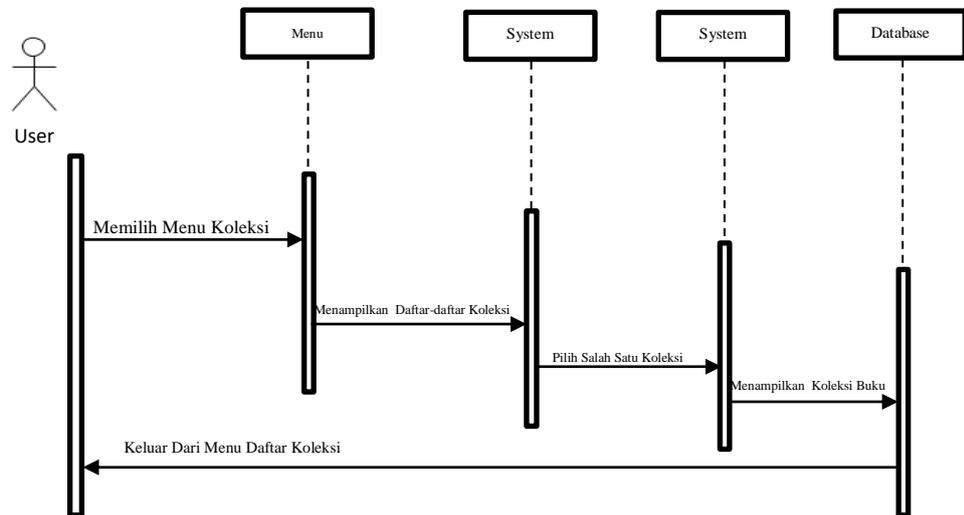
Sequence diagram membantu interaksi berkomunikasi diagram yang menunjukkan bagaimana objek berinteraksi dengan bagian interaksi (interaksi khusus) dan komunikasi yang dilakukan secara bertahap. (Syach Putra et al., 2024).

a. *Sequence Diagram Tampilan Utama*



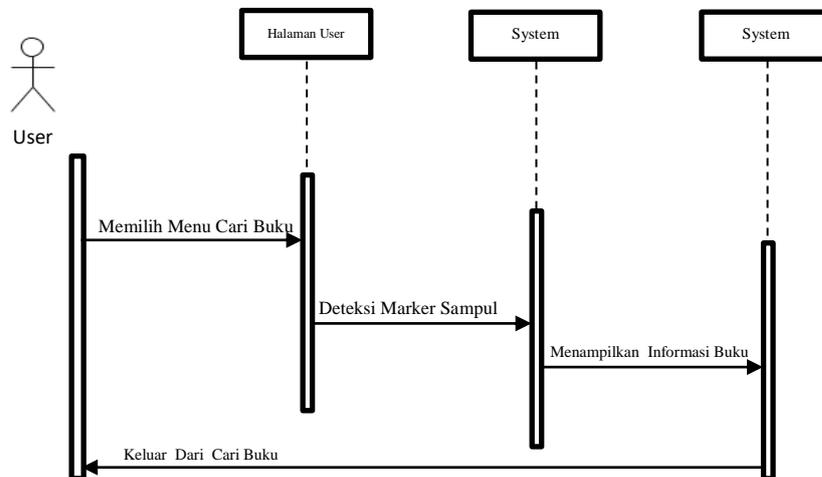
Gambar 4. 5 *Sequence Diagram Tampilan Utama*

b. *Sequence Diagram Koleksi*



Gambar 4. 6 *Sequence Diagram Koleksi*

c. *Sequence Diagram Cari Buku*



Gambar 4. 7 *Sequence Diagram Cari Buku*

B. Detail Aplikasi

1. Tampilan Utama

Tampilan utama aplikasi terdapat menu koleksi dan cari buku

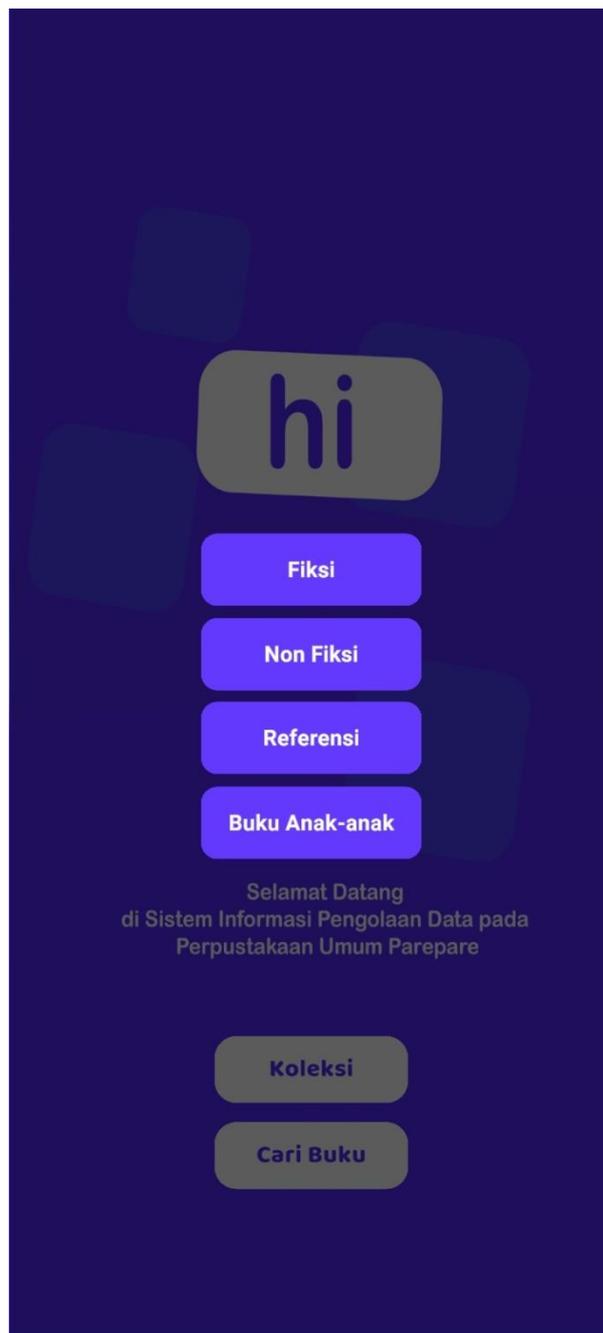


Gambar 4. 8 Tampilan Utama

Cara membuat tampilan utama, Masuk aplikasi unity → Pilih Scene Home → Tambahkan Komponen (Canvas Panel dan Button) → Atur ukuran Canvas dan Panel → Tambahkan Gambar 2D di Inspector Image pada Panel → Tambahkan juga Gambar 2D Pada Inspector Image Pada ke 2 Button.

2. Tampilan *Pop up* Koleksi

Merupakan halaman apabila kita memilih menu koleksi maka menampilkan pop up koleksi buku.



Gambar 4.9 *Pop up* Koleksi

Tabel 4. 2 *Script* Popup Koleksi

```

using UnityEngine;

public class PopupController : MonoBehaviour
{
    private Animator animator;
    private bool canShowPopup = true;

    void Start()
    {
        animator = GetComponent<Animator>();
    }

    public void ShowPopup()
    {
        if (canShowPopup)
        {
            animator.SetTrigger("Show");
            canShowPopup = false;
        }
    }

    public void HidePopup()
    {
        animator.SetTrigger("Hide");
    }

    // Panggil fungsi ini di akhir animasi "Hide"
    public void EnableShow()
    {
        canShowPopup = true;
    }
}

```

Tabel 4. 3 *Script* Panel Button

```

using System.Collections;
using UnityEngine;
using UnityEngine.UI;

public class PanelClick : MonoBehaviour
{
    public GameObject targetObject; // Objek yang ingin di nonaktifkan
    private Button panelButton; // Panel yang akan diklik (pastikan memiliki
    komponen Button)

    void Start()
    {
        // Mendapatkan komponen Button dari panel
        panelButton = GetComponent<Button>();
    }
}

```

```

// Menambahkan listener pada event klik button
if (panelButton != null)
{
    panelButton.onClick.AddListener(OnPanelClick);
}
else
{
    Debug.LogError("Button component not found on this panel.");
}
}
void OnPanelClick()
{
    // Menonaktifkan objek target saat panel diklik
    if (targetObject != null)
    {
        targetObject.SetActive(false);
    }
    else
    {
        Debug.LogError("Target object not assigned.");
    }
}
}
}

```

Tabel 4. 4 *Script Class Menu*

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class menu : MonoBehaviour
{
    public void pindahScene(string scenename)
    {
        SceneManager.LoadScene(scenename);
    }
    public void BackButton(string scenename)
    {
        SceneManager.LoadScene(scenename);
    }
    public void QuitButton()
    {
        Application.Quit();
        Debug.Log("ANJAY");
    }
}
}

```

1. Tampilan Koleksi-Koleksi Buku

Merupakan halaman apabila kita memilih salah satu daftar-daftar koleksi maka menampilkan koleksi-koleksi buku.



Gambar 4. 10 Tampilan Koleksi-Koleksi Buku

Tabel 4. 5 Script Menampilakn Koleksi-Koleksi Buku

```

using System;
using Firebase;
using Firebase.Database;
using Firebase.Extensions;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class BookManager : MonoBehaviour{
    private DatabaseReference dbReference;
    public GameObject bookPrefab;
    public Transform contentPanel;
    void Start()
    {
        Debug.Log("Start method called.");
        InitializeFirebase();
    }
    private void InitializeFirebase()
    {
        Debug.Log("Initializing Firebase...");
        FirebaseApp
            .CheckAndFixDependenciesAsync()
            .ContinueWithOnMainThread(task =>
            {
                if (task.Result == DependencyStatus.Available)
                {
                    dbReference = FirebaseDatabase
                        .GetInstance("https://perpustakaan-apk-default-rtdb.firebaseio.com/")
                        .RootReference;
                    Debug.Log("Firebase connected successfully.");
                    GetBookDetails();
                }
                else
                {
                    Debug.LogError("Firebase connection failed: " + task.Result);
                }
            });
    }

    private void GetBookDetails()
    {
        // Membaca koleksi yang dipilih dari PlayerPrefs
        string selectedKoleksi = PlayerPrefs.GetString("SelectedKoleksi", "Fiksi"); //
        Default ke "Fiksi" jika tidak ada
        Debug.Log($"Fetching books from collection: {selectedKoleksi}");

        // Query Firebase untuk mengambil buku berdasarkan koleksi yang dipilih
        dbReference
            .Child("books")

```

```

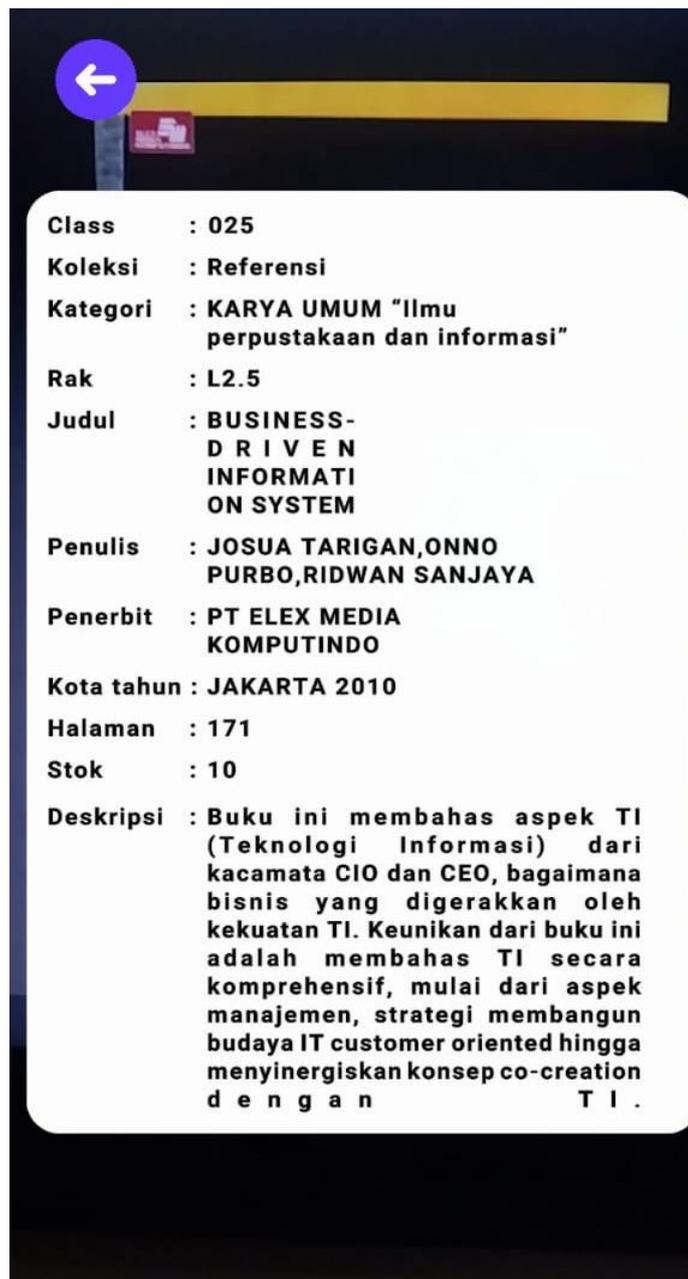
.OrderByChild("koleksi")
.EqualTo(selectedKoleksi) // Filter berdasarkan koleksi
.GetValueAsync()
.ContinueWithOnMainThread(task =>
{
    if (task.IsCompleted)
    {
        if (task.Exception != null)
        {
            Debug.LogError("Exception occurred while fetching books: " +
task.Exception);
            return;
        }
        DataSnapshot snapshot = task.Result;

        if (snapshot.HasChildren)
        {
            Debug.Log($"Total books in {selectedKoleksi} collection:
{snapshot.ChildrenCount}");
            foreach (DataSnapshot bookSnapshot in snapshot.Children)
            {
                string bookId = bookSnapshot.Key;
                string bookClass = bookSnapshot.Child("class").Value?.ToString();
                string judul = bookSnapshot.Child("judul").Value?.ToString();
                string penulis = bookSnapshot.Child("penulis").Value?.ToString();
                string kategori = bookSnapshot.Child("kategori").Value?.ToString();
                string rak = bookSnapshot.Child("rak").Value?.ToString();
                string penerbit = bookSnapshot.Child("penerbit").Value?.ToString();
                string kotaTahun =
bookSnapshot.Child("kota_tahun").Value?.ToString();
                string halaman = bookSnapshot.Child("halaman").Value?.ToString();
                string koleksi = bookSnapshot.Child("koleksi").Value?.ToString();
                int stok = Convert.ToInt32(bookSnapshot.Child("stok").Value ?? 0);
                Debug.Log($"Processing book ID: {bookId}");
                // Instansiasi prefab dan mengisi data buku
                GameObject newBookItem = Instantiate(bookPrefab, contentPanel);
                TMP_Text[] texts =
newBookItem.GetComponentInChildren<TMP_Text>();
                foreach (TMP_Text text in texts)
                {
                    switch (text.name)
                    {
                        case "class":
                            text.text = bookClass;
                            break;
                        case "judul":
                            text.text = judul;
                            break;
                        case "penulis":
                            text.text = penulis;

```


3. Cari buku

Merupakan halaman apabila kita memilih menu cari buku dan user mengarahkan smartphone ke marker/target maka menampilkan informasi buku setelah terdeteksi.



Gambar 4. 11 Tampilan Cari Buku

Tabel 4. 6 *Script Cari Buku*

```

using System;
using System.Collections;
using System.Collections.Generic;
using Firebase;
using Firebase.Database;
using Firebase.Extensions;
using TMPro;
using UnityEngine;
using Vuforia;

public class ScanBook : MonoBehaviour
{
    private DatabaseReference dbReference;
    public GameObject bookPrefab;
    public Transform contentPanel;

    // Tambahkan list ObserverBehaviour untuk mendukung banyak target
    public List<ObserverBehaviour> targetObservers;

    // Flag untuk memastikan Firebase sudah diinisialisasi
    private bool isFirebaseInitialized = false;
    // Coroutine reference
    private Dictionary<string, Coroutine> fetchCoroutines = new Dictionary<string,
Coroutine>();
    // Fetch interval in seconds
    [SerializeField]
    private float fetchInterval = 5f; // Sesuaikan sesuai kebutuhan
    void Start()
    {
        Debug.Log("[Start] Method called.");
        InitializeFirebase();
    }
    private void InitializeFirebase()
    {
        Debug.Log("[InitializeFirebase] Initializing Firebase...");

        FirebaseApp.CheckAndFixDependenciesAsync().ContinueWithOnMainThread(task
=>
        {
            var dependencyStatus = task.Result;
            Debug.Log($"[InitializeFirebase] Dependency Status: {dependencyStatus}");
            if (dependencyStatus == DependencyStatus.Available)
            {
                FirebaseApp app = FirebaseApp.DefaultInstance;
                dbReference = FirebaseDatabase
                    .GetInstance(app)
                    .GetReference("books"); // Mengarahkan langsung ke node "books"
                Debug.Log("[InitializeFirebase] Firebase connected successfully.");
                isFirebaseInitialized = true;
            }
        });
    }
}

```

```

        InitializeVuforia(); // Inisialisasi Vuforia setelah Firebase siap
    }
    else
    {
        Debug.LogError($"[InitializeFirebase] Firebase connection failed:
{dependencyStatus}");
        // Opsional: Beritahu pengguna melalui UI
    }
});
}

private void InitializeVuforia()
{
    Debug.Log("[InitializeVuforia] Initializing Vuforia...");
    if (targetObservers == null || targetObservers.Count == 0)
    {
        Debug.LogError("[InitializeVuforia] No targetObservers assigned in the
Inspector.");
        return;
    }
    foreach (var observer in targetObservers)
    {
        if (observer != null)
        {
            observer.OnTargetStatusChanged += OnTargetStatusChanged;
            Debug.Log($"[InitializeVuforia] Subscribed to target
'{observer.TargetName}'.");
        }
        else
        {
            Debug.LogError("[InitializeVuforia] One of the targetObservers is null.");
        }
    }
}

private void OnTargetStatusChanged(ObserverBehaviour behaviour, TargetStatus
status)
{
    Debug.Log($"[OnTargetStatusChanged] Target '{behaviour.TargetName}' status
changed to {status.Status}");

    if (status.Status == Status.TRACKED)
    {
        Debug.Log($"[OnTargetStatusChanged] Target '{behaviour.TargetName}' is
TRACKED.");
        if (isFirebaseInitialized)
        {
            if (!fetchCoroutines.ContainsKey(behaviour.TargetName))
            {
                Debug.Log($"[OnTargetStatusChanged] Starting coroutine for target

```

```

'behaviour.TargetName}");
        fetchCoroutines[behaviour.TargetName] =
StartCoroutine(FetchBookDetailsRepeatedly(behaviour.TargetName));
    }
    else
    {
        Debug.Log($"[OnTargetStatusChanged] Coroutine already running for
target '{behaviour.TargetName}");
    }
}
else
{
    Debug.LogWarning("[OnTargetStatusChanged] Firebase not initialized.
Cannot fetch book data.");
}
}
else
{
    Debug.Log($"[OnTargetStatusChanged] Target '{behaviour.TargetName}' is
not TRACKED.");
    if (fetchCoroutines.ContainsKey(behaviour.TargetName))
    {
        Debug.Log($"[OnTargetStatusChanged] Stopping coroutine for target
'behaviour.TargetName}");
        StopCoroutine(fetchCoroutines[behaviour.TargetName]);
        fetchCoroutines.Remove(behaviour.TargetName);
    }
    // Anda mungkin perlu memodifikasi ClearBookDetails untuk menangani
multiple target
    ClearBookDetails();
}
}
private IEnumerator FetchBookDetailsRepeatedly(string targetName)
{
    Debug.Log($"[Coroutine] Starting FetchBookDetailsRepeatedly for targetName:
'{targetName}");
    while (true)
    {
        FetchBookDetailsByTarget(targetName);
        yield return new WaitForSeconds(fetchInterval);
    }
}
private void FetchBookDetailsByTarget(string targetName)
{
    Debug.Log($"[FetchBookDetailsByTarget] Called with targetName:
'{targetName}");

    if (string.IsNullOrEmpty(targetName))
    {
        Debug.LogError("[FetchBookDetailsByTarget] Target name is null or

```

```

empty.");
    return;
}
dbReference.Child(targetName).GetValueAsync().ContinueWithOnMainThread(task
=>
{
    if (task.IsCompleted)
    {
        if (task.Exception != null)
        {
            Debug.LogError($"[FetchBookDetailsByTarget] Exception:
{task.Exception}");
            return;
        }
        DataSnapshot bookSnapshot = task.Result;

        if (bookSnapshot.Exists)
        {
            Debug.Log($"[FetchBookDetailsByTarget] Found book data for
'{targetName}': {bookSnapshot.ChildrenCount} entries");

            // Ekstrak data buku dengan pengecekan null
            string bookClass = bookSnapshot.Child("class").Value?.ToString() ??
"N/A";
            string judul = bookSnapshot.Child("judul").Value?.ToString() ?? "N/A";
            string penulis = bookSnapshot.Child("penulis").Value?.ToString() ??
"N/A";
            string kategori = bookSnapshot.Child("kategori").Value?.ToString() ??
"N/A";
            string rak = bookSnapshot.Child("rak").Value?.ToString() ?? "N/A";
            string penerbit = bookSnapshot.Child("penerbit").Value?.ToString() ??
"N/A";
            string kotaTahun = bookSnapshot.Child("kota_tahun").Value?.ToString()
?? "N/A";
            string halaman = bookSnapshot.Child("halaman").Value?.ToString() ??
"N/A";
            string koleksi = bookSnapshot.Child("koleksi").Value?.ToString() ??
"N/A";

            int stok = 0;
            if (bookSnapshot.Child("stok").Exists)
            {
                if (int.TryParse(bookSnapshot.Child("stok").Value.ToString(), out int
parsedStok))
                {
                    stok = parsedStok;
                }
                else
                {
                    Debug.LogWarning($"[FetchBookDetailsByTarget] Stok untuk
'{targetName}' tidak dapat diubah ke integer. Default ke 0.");
                }
            }
        }
    }
}

```

```

    }
}

Debug.Log($"[FetchBookDetailsByTarget] Book Data: {judul},
{penulis}, {kategori}, {rak}, {penerbit}, {kotaTahun}, {halaman}, {koleksi},
{stok}");

// Tampilkan detail buku
DisplayBookDetails(
    bookClass,
    judul,
    penulis,
    kategori,
    rak,
    penerbit,
    kotaTahun,
    halaman,
    koleksi,
    stok
);
}
else
{
    Debug.LogWarning($"[FetchBookDetailsByTarget] No book data found
for '{targetName}'. Check database.");
    ClearBookDetails();
}
}
else
{
    Debug.LogError($"[FetchBookDetailsByTarget] Task not completed.
Exception: {task.Exception}");
}
});
}
private void DisplayBookDetails(
    string bookClass,
    string judul,
    string penulis,
    string kategori,
    string rak,
    string penerbit,
    string kotaTahun,
    string halaman,
    string koleksi,
    int stok
)
{
    Debug.Log("[DisplayBookDetails] Displaying book details in UI..");

```

```
if (contentPanel == null)
{
    Debug.LogError("[DisplayBookDetails] Content Panel is not assigned.");
    return;
}
if (bookPrefab == null)
{
    Debug.LogError("[DisplayBookDetails] Book Prefab is not assigned.");
    return;
}
// Clear previous entries
foreach (Transform child in contentPanel)
{
    Destroy(child.gameObject);
}
// Instantiate a new book item
GameObject newBookItem = Instantiate(bookPrefab, contentPanel);
TMP_Text[] texts = newBookItem.GetComponentsInChildren<TMP_Text>();

foreach (TMP_Text text in texts)
{
    switch (text.name.ToLower())
    {
        case "class":
            text.text = bookClass;
            break;
        case "judul":
            text.text = judul;
            break;
        case "penulis":
            text.text = penulis;
            break;
        case "kategori":
            text.text = kategori;
            break;
        case "rak":
            text.text = rak;
            break;
        case "penerbit":
            text.text = penerbit;
            break;
        case "kota_tahun":
            text.text = kotaTahun;
            break;
        case "hal":
            text.text = halaman;
            break;
        case "koleksi":
            text.text = koleksi;
            break;
    }
}
```

```

        case "stok":
            text.text = stok.ToString();
            break;
        default:
            Debug.LogWarning($"[DisplayBookDetails] Unrecognized TMP_Text
name: '{text.name}");
            break;
    }
}
Debug.Log("[DisplayBookDetails] Book details displayed in UI.");
}
private void ClearBookDetails()
{
    Debug.Log("[ClearBookDetails] Clearing book details from UI...");
    if (contentPanel == null)
    {
        Debug.LogError("[ClearBookDetails] Content Panel is not assigned.");
        return;
    }
    foreach (Transform child in contentPanel)
    {
        Destroy(child.gameObject);
    }
}
void OnDestroy()
{
    Debug.Log("[OnDestroy] Called. Unsubscribing from Vuforia events.");
    if (targetObservers != null)
    {
        foreach (var observer in targetObservers)
        {
            if (observer != null)
            {
                observer.OnTargetStatusChanged -= OnTargetStatusChanged;
                Debug.Log($"[OnDestroy] Unsubscribed from target
'{observer.TargetName}'.");
            }
        }
    }
    // Pastikan semua coroutine dihentikan saat objek dihancurkan
    foreach (var coroutine in fetchCoroutines.Values)
    {
        if (coroutine != null)
        {
            StopCoroutine(coroutine);
        }
    }
    fetchCoroutines.Clear();
}
}

```

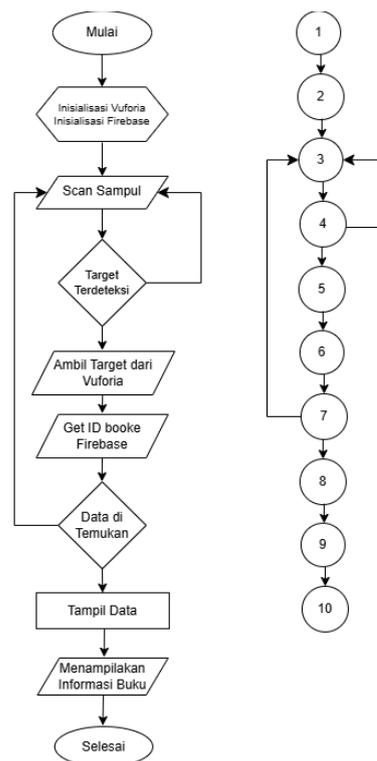
C. Pengujian Aplikasi

Pengujian pada sistem ini menggunakan pengujian *Black box* dan *White box*. Adapun rincian pengujian sistem dapat dilihat sebagai berikut.

1. *White box Testing*

Pengujian sistem menggunakan *White box Testing* yang menggunakan *flowchart* dan *flowgraph* dari sistem. Adapun di bawah adalah hasil pengujian sistem menggunakan *White box*.

a. Pengujian *Flowchart* dan *Flowgraph* Cari Buku



Gambar 4. 12 Pengujian *Flowchart* dan *Flowgraph* Cari Buku

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut :

- (1) Menghitung cyclomatic complexity $V(G)$ pada edge dan node

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

$$E \text{ (edge)} = 11$$

$$N \text{ (node)} = 10$$

$$P \text{ (Predikat node)} = 2$$

Penyelesaian :

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 11 - 10 + 2 \\ &= 3 \end{aligned}$$

$$\begin{aligned} \text{Predikat (P)} &= P + 1 \\ &= 2 + 1 \\ &= 3 \end{aligned}$$

(2) Berdasarkan perhitungan cyclomatic complexity dari flowgraph diatas memiliki region = 3

(3) Independent path pada flowgraph tersebut yakni :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10$$

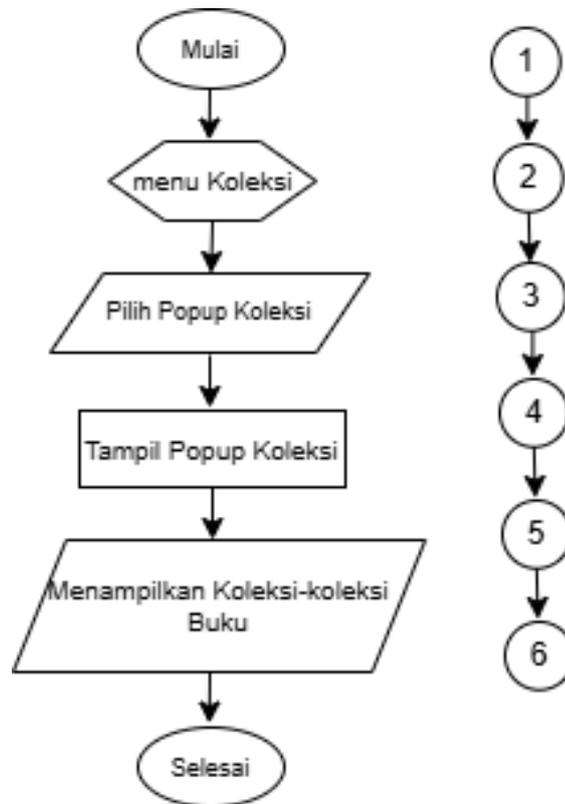
$$\text{Path 3} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10$$

(4) Grafik Matriks Cari Buku

Tabel 4. 7 Grafik Matriks Cari Buku

	1	2	3	4	5	6	7	8	9	10	E - 1
1		1									1-1=0
2			1								1-1=0
3				1							1-1=0
4			1		1						2-1=1
5						1					1-1=0
6							1				1-1=1
7			1					1			2-1=0
8									1		1-1=0
9										1	1-1=0
10											0
SUM (E + 1)											2+ 1 = 3

b. Pengujian *Flowchart* dan *Flowgraph* Koleksi Buku



Gambar 4. 13 Pengujian *Flowchart* dan *Flowgraph* Koleksi Buku
Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut :

(1) Menghitung cyclomatic complexity $V(G)$ pada edge dan node

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

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

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 5 - 6 + 2$$

$$= 1$$

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

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan cyclomatic complexity dari flowgraph diatas memiliki region = 1

(3) Independent path pada flowgraph tersebut yakni :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6$$

(4) Grafik Matriks Koleksi Buku

Tabel 4. 8 Grafik Matriks Koleksi Buku

	1	2	3	4	5	6	E - 1	
1		1					1-1=0	
2			1				1-1=0	
3				1			1-1=0	
4					1		1-1=0	
5						1	1-1=0	
6							0	
SUM (E + 1)								0+ 1 = 1

2. *Black box Testing*

Pengujian yang dilakukan pada sistem yaitu pengujian *Black box* dimana peneliti menguji tampilan aplikasi apakah berhasil tampil atau tidak, adapun hasil dan penjelasan detailnya akan dijelaskan selanjutnya.

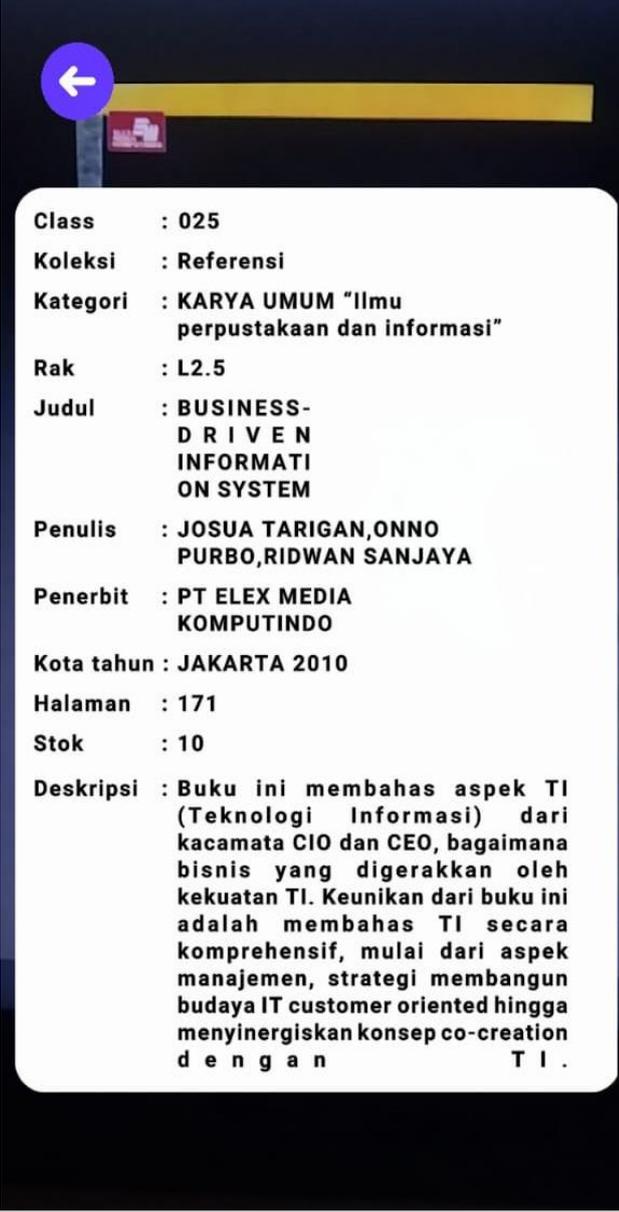
a. *Black box Testing* tampilan utama**Tabel 4. 9** *Black box Testing* tampilan utama

Tes Faktor	Hasil	Keterangan
<i>User</i> Membuka Aplikasi	✓	Berhasil Tampil Halaman utama
<i>Screenshot</i>		
		

b. *Black box Testing* Popup Koleksi**Tabel 4. 10** *Black box Testing* Popup Koleksi

Tes Faktor	Hasil	Keterangan
User Menekan Tombol koleksi	✓	Berhasil Tampil Popup Koleksi Dan Koleksi-Koleksi Buku
<i>Screenshot</i>		
		

c. *Black box Testing* Cari Buku**Tabel 4. 11** *Black box Testing* Cari Buku

Tes Faktor	Hasil	Keterangan
User Menekan Tombol Cari Buku	✓	Berhasil Mendeteksi Marker Buku Dan Menampilkan Informasi Buku
<i>Screenshot</i>		
 <p> Class : 025 Koleksi : Referensi Kategori : KARYA UMUM "Ilmu perpustakaan dan informasi" Rak : L2.5 Judul : BUSINESS-DRIVEN INFORMATI ON SYSTEM Penulis : JOSUA TARIGAN, ONNO PURBO, RIDWAN SANJAYA Penerbit : PT ELEX MEDIA KOMPUTINDO Kota tahun : JAKARTA 2010 Halaman : 171 Stok : 10 Deskripsi : Buku ini membahas aspek TI (Teknologi Informasi) dari kacamata CIO dan CEO, bagaimana bisnis yang digerakkan oleh kekuatan TI. Keunikan dari buku ini adalah membahas TI secara komprehensif, mulai dari aspek manajemen, strategi membangun budaya IT customer oriented hingga menyinergiskan konsep co-creation dengan TI. </p>		

3. Uji Tambahan (Khusus Kamera AR)

Adapun uji tambahan yang dilakukan yaitu dengan cara menguji jarak kamera ke marker yang di sorot dengan pencahayaan lampu yang digunakan mempunyai intensitas cahaya 45 Lux dan dilakukan di ruangan seluas 12 m², berikut adalah tabel hasil uji kamera AR.

Tabel 4. 12 Hasil uji tambahan dengan *marker* gambar

Percobaan	Jarak							Persen tase
	10 cm	20 cm	30 cm	40 cm	50 cm	60 cm	80 cm	
1	✓	✓	✓	✓	✓	✓	✗	85,71%
2	✓	✓	✓	✓	✓	✓	✗	85,71%
3	✓	✓	✓	✓	✓	✓	✗	85,71%
4	✓	✓	✓	✓	✓	✓	✗	85,71%
5	✓	✓	✓	✓	✓	✓	✗	85,71%
6	✓	✓	✓	✓	✓	✓	✗	85,71%
7	✓	✓	✓	✓	✓	✓	✗	85,71%
8	✓	✓	✓	✓	✓	✓	✗	85,71%
9	✓	✓	✓	✓	✓	✓	✗	85,71%
10	✓	✓	✓	✓	✓	✓	✗	85,71%

Berdasarkan tabel hasil uji tambahan di atas dapat disimpulkan bahwa nilai persentase rata-rata dengan 10 (sepuluh) kali percobaan secara berturut-turut adalah 87,5%. Adapun penyebab tidak terdeteksinya marker gambar dengan jarak 80 cm ketika kamera sudah tidak dapat membaca marker gambar dengan jelas akibat jarak.

4. Database

Tabel 4. 13 *Books*

No	Nama	Jenis
1	ID	Varchar
2	class	Varchar
3	kategori	Varchar
4	rak	Varchar
5	judul	Varchar
6	penulis	Varchar
7	penerbit	Varchar
8	kota_tahun	Varchar
9	halaman	Int
10	stok	Int
11	deskripsi	Varchar

BAB V

PENUTUP

A. Kesimpulan

Penelitian ini berhasil mengembangkan sistem deteksi buku berdasarkan sampul menggunakan teknologi Vuforia dan Firebase. Vuforia digunakan untuk mengenali dan melacak sampul buku dalam aplikasi berbasis Augmented Reality (AR), sementara Firebase berfungsi sebagai backend untuk penyimpanan data buku dan autentikasi pengguna. Sistem ini mampu memberikan informasi buku secara real-time dengan akurasi yang baik, meskipun terdapat beberapa tantangan terkait kualitas gambar sampul.

Hasil uji yang telah dilakukan diantaranya uji *Black box* dan *White box* semuanya berjalan baik yang mana tampilan dan tombol berfungsi dengan baik.

B. Saran

Penulis menyadari bahwa masih ada kekurangan yang ada pada penelitian ini dan masih perlu dikembangkan agar dapat lebih sempurna kedepannya. Daripada itu penulis memiliki beberapa saran untuk pengembang selanjutnya, sebagai berikut:

1. Mempercantik desain interface aplikasi sehingga dapat terlihat lebih menarik.
2. Menambah fitur-fitur aplikasi yang dapat diperlukan nantinya.

DAFTAR ISI

- Abd.Rasman. (2022). *Deteksi Identitas Berdasarkan Wajah*. Universitas Muhammadiyah Parepare.
- Basri, M. (2020). *Aplikasi Pendeteksi Khasiat Daun Berbasis Augmented Reality*.
- Ismail, I., & Efendi, J. (2020). Black-Box Testing: Analisis Kualitas Aplikasi Source Code Bank Programming. *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, 4(2), 1. <https://doi.org/10.35870/jtik.v5i1.148>
- Li, B A B. 2022. “5 Gambar 2.1. Foto Di Museum 2.2. Buku.” : 4–37.
- JASMINE, KHANZA. 2014. “APLIKASI PENDETEKSI KHASIAT DAUN BERBASIS AUGMENTED REALITY.” *Penambahan Natrium Benzoat Dan Kalium Sorbat (Antiinversi) Dan Kecepatan Pengadukan Sebagai Upaya Penghambatan Reaksi Inversi Pada Nira Tebu (0421)*.
- Kasmir. 2018. “Bab 1 pendahuluan.” *Pelayanan Kesehatan (2015)*: 3–13.
- Maedjaja, F., & Efraim. (2021). Sistem deteksi teks pada cover buku dengan pendekatan karakter teks. *Infact Ukrim*, 6(2).
- Masyhuddin, S, A Irmayani Pawelloi, and Marlina. 2022. “Aplikasi Deteksi Tipe Rumah Subsidi Berbasis Augmented Reality.” : 1–6. <https://jurnal.umpar.ac.id/index.php/sylog>.
- Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., Giansyah, Q. A., & Hamzah, M. L. (2023). Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web. 4.
- Ramadhan, R. (2023). Pengelolaan Perpustakaan Digital Di Badan Perpustakaan Dan Kearsipan Daerah Provinsi Jawa Barat. *Jurnal Pustaka Budaya*, 10(1), 21-31.
- S, M. (2023). *Aplikasi Deteksi Tipe Rumah Berbasis Augmented Reality*. Universitas Muhammadiyah Parepare.
- Sya'bani, M. M. (2018). Penerapan metode cosine similarity dan pembobotan TF/IDF pada sistem klasifikasi sinopsis buku di Perpustakaan Kejaksaan Negeri Jember. *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, 3(1), 31-42.
- Syach Putra, R., Apriliani, F., & Hartono, M. I. (2024). Perancangan Dan

Implementasi Sistem Informasi Sekolah Berbasis Web Pada SD Negeri Pamulang Barat. *Buletin Ilmiah Ilmu Komputer Dan Multimedia*, 2(2), 220– 227. <https://jurnalmahasiswa.com/index.php/biikma>

Wafiah, A., Suwardoyo, U., & Syawal, F. (2021). Aplikasi Mendeteksi Jenis Harga Assesorieshandphone Menggunakan Augmented Reality. *Jurnal Sintaks Logika*, 1(1), 26-32.

Winiarti, S., Widayanti, D., Ahdiani, U., & Ismail, T. (2022). Klasifikasi Jenis Buku Berdasarkan Cover dan Judul Buku Menggunakan Metode Support Vector Machine dan Cosine Similarity. *Sainteks*, 19(1), 53. <https://doi.org/10.30595/sainteks.v19i1.134>

Wahyuddin, Irmayani Pawelloi. 2022. “Aplikasi Pemesanan Menu Pada Warko Shearlock Berbasis Abdroid.” *Jurnal Sintaks Logika* 2(3):11–16. doi: 10.31850/jsilog.v2i3.1846.

Yusuf, Dani, and Freddy Nur Afandi. 2020. “Aplikasi Absensi Berbasis Android Menggunakan Validasi Kordinat Lokasi Dan Nomor Handpone Guna Menghindari Penularan Virus Covid 19.” *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi* 10(1):16–22. doi: 10.36448/jmsit.v10i1.1492.