

## ***Augmented Reality* Kekarangan Bali dengan *Natural Feature Tracking***

I Gede Bagus Anom<sup>a1</sup>, I Wayan Santiyasa<sup>a2</sup>, I Ketut Gede Suhartana<sup>a3</sup>, I Dewa Made Bayu Atmaja Darmawan<sup>a4</sup>, Luh Gede Astuti<sup>a5</sup>, Made Agung Raharja<sup>a6</sup>

<sup>a</sup>Informatics Engineering, Faculty of Math and Science, University of Udayana  
South Kuta, Badung, Bali, Indonesia

<sup>1</sup>gusanom1503@yahoo.co.id

<sup>2</sup>santiyasa@unud.ac.id

<sup>3</sup>ikg.suhartana@unud.ac.id

<sup>4</sup>dewabayu@unud.ac.id

<sup>5</sup>lg.astuti@unud.ac.id

<sup>6</sup>made.agung@unud.ac.id

### **Abstract**

*The art of carving that developed in Bali has very distinctive characteristics and can make Balinese carvings easily recognizable, among others, from several combinations of leaf, fruit, and flower motifs in very beautiful convex and concave shapes. Balinese carvings are generally divided into three parts, namely Keketusan, Papatran and Kekarangan. Kekarangan is a Balinese carving that is placed in the corner of the building and has grown and developed in the life of the people in Bali. The lack of information about Kekarangan Bali makes people less aware of the forms and meanings contained in Kekarangan Bali due to limited distance and time to be able to study Kekarangan Bali directly. Therefore, Augmented Reality (AR) is needed as a solution to be used as a teaching aid in helping to introduce Kekarangan Bali more specifically by using the Natural Feature Tracking (NFT) method because it can directly implement feature detection and then provide descriptors at every corner detected in the target image that is visible in the real environment so that the application of Augmented Reality (AR) is not fixated on the target image that has been determined to be able to display 3D objects.*

**Keywords:** *Kekarangan Bali, Augmented Reality, Natural Feature Tracking.*

### **1. Pendahuluan**

Seni ukir yang berkembang di Bali mempunyai ciri-ciri yang sangat khas serta dapat membuat seni ukir Bali mudah dikenali antara lain dari beberapa perpaduan motif daun, buah, serta bunga dalam bentuk cembung dan cekung yang sangat indah. Seni ukir Bali secara umum terbagi menjadi tiga bagian yaitu Keketusan, Papatran serta Kekarangan [6]. Ketiga bagian seni ukir Bali tersebut memiliki pengembangan secara luas namun dalam penelitian ini difokuskan pada Kekarangan Bali. Kekarangan merupakan Seni ukir Bali yang telah tumbuh serta berkembang pada kehidupan masyarakat di Bali. Hal ini dapat dibuktikan dengan adanya seni ukiran yang melekat pada bagian tertentu arsitektur pura, alat musik serta bangunan tradisional di Bali dengan pakem yang terletak pada tatanan penempatan yang sesuai dengan bentuk-bentuk dari ukiran tersebut yang diambil dari penggabungan corak binatang dan tumbuh-tumbuhan serta tetap menonjolkan estetikanya. Informasi mengenai Kekarangan Bali yang minim membuat masyarakat khususnya yang ada di luar pulau Bali kurang mengetahui tentang bentuk dan makna yang terkandung dalam Kekarangan Bali tersebut yang dikarenakan nilai kesakralan yang tinggi pada bangunan pura yang dihiasi dengan Kekarangan ini serta terbatas dengan jarak dan waktu yang diperlukan untuk dapat mempelajari Kekarangan Bali secara langsung. Karena hal ini kita memerlukan Augmented Reality (AR) sebagai solusi untuk dijadikan alat peraga dalam membantu mengenalkan Kekarangan Bali ini secara lebih spesifik melalui objek 3D yang ditampilkan pada lingkungan nyata.

Secara umum *Augmented Reality (AR)* adalah sebuah penggabungan benda-benda virtual (maya) dua dimensi ataupun tiga dimensi dan lingkungan nyata dengan interaktif secara langsung. Dengan adanya teknologi *Augmented Reality (AR)*, proses belajar dapat dilaksanakan dengan lebih interaktif, kreatif, serta mudah karena melalui objek visual tiga dimensi yang langsung diterapkan ke dunia nyata secara virtual. Untuk mengaplikasikan *Augmented Reality (AR)* dibutuhkan image target

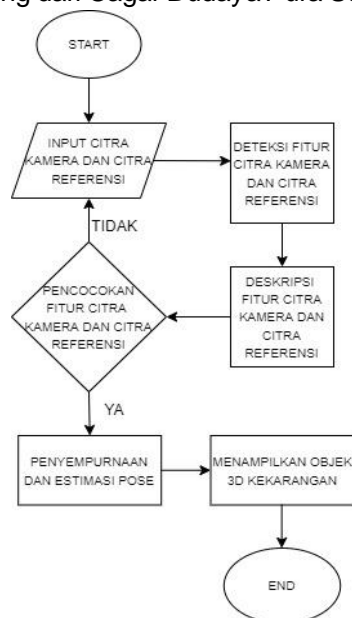
atau marker yang akan menjadi patokan dalam menampilkan objek 3D dengan menggunakan metode *Natural Feature Tracking (NFT)* yang merupakan teknik pelacakan dengan cara mendeteksi serta melacak fitur secara normal atau alami yang berupa tepi, gumpalan, sudut dan lain-lain [13].

*Natural Feature Tracking (NFT)* dipilih karena metode ini dapat langsung menerapkan deteksi fitur lalu memberi deskriptor pada setiap sudut yang terdeteksi pada image target yang terlihat di lingkungan nyata sehingga penerapan *Augmented Reality (AR)* ini tidak terpacu pada satu *image target* yang telah ditentukan untuk dapat menampilkan objek 3D Kekarangan tersebut. Ada beberapa algoritma yang bisa diterapkan dengan metode NFT yaitu ORB, SURF, BRISK, SIFT dan lainnya tetapi dalam penelitian ini menggunakan akan algoritma *ORB (Oriented Fast And Rotated Brief)* yang menurut Hamidia, Zenati-Henda, Belqhit, & Bellarbi (2015) [10] merupakan algoritma untuk melakukan deskriptor yang bersifat biner pada fitur dengan cepat serta memiliki tingkat akurasi serta ketahanan yang tinggi dalam melakukan deteksi fitur sehingga cocok untuk diterapkan pada aplikasi *Augmented Reality (AR)*. Algoritma *ORB (Oriented Fast And Rotated Brief)* digunakan untuk mendeteksi titik sudut dan menentukan deskriptor pada citra. Algoritma ini merupakan kombinasi dari *FAST (Features for Accelerated Segment Test)* serta *BRIEF (Binary Robust Independent Elementary Features)* sehingga menghasilkan efisiensi yang lebih baik jika dibandingkan dengan SIFT dalam berbagai situasi [5]. Pada penelitian ini juga menerapkan tahap pencocokan fitur menggunakan algoritma *Brute-Force Hamming* yang disebutkan Abdeen (2011) [12] merupakan algoritma pendeteksi kemiripan yang diterapkan dengan sederhana serta dapat digunakan untuk proses pencocokan pola ataupun bentuk. Algoritma *Brute-Force Hamming* digunakan untuk mencocokkan citra deskriptor lalu menambahkan variabel *hamming count* pada citra deskriptor yang terdeteksi berbeda. *Brute-Force Hamming* dipilih karena algoritma ini menghasilkan bobot persentase yang tinggi dalam mendeteksi kemiripan [1].

Jadi dalam penelitian ini akan mengimplementasikan teknologi *Augmented Reality (AR)* dengan menerapkan metode *Natural Feature Tracking (NFT)* menggunakan algoritma *ORB (Oriented Fast And Rotated Brief)* untuk melakukan deteksi serta deskripsi fitur dan algoritma *Brute-Force Hamming* untuk melakukan pencocokan fitur dalam pengenalan Kekarangan Bali.

## 2. Metode Penelitian

Penelitian ini akan mengaplikasikan beberapa algoritma untuk menjalankan metode *Natural Feature Tracking (NFT)* antara lain algoritma *ORB (Oriented Fast And Rotated Brief)* dengan penerapan berbasis algoritma *FAST (Features for Accelerated Segment Test)* untuk proses deteksi fitur, algoritma *ORB (Oriented Fast And Rotated Brief)* yang berdasar pada algoritma *BRIEF (Binary Robust Independent Elementary Features)* untuk proses deskripsi fitur, algoritma *Brute-Force Hamming* untuk proses pencocokan fitur serta fungsi *RANSAC (Random Sample Consensus)* untuk proses penyempurnaan dan estimasi pose. Data yang digunakan untuk membuat objek 3D dalam penelitian ini diambil secara langsung dari Cagar Budaya Pura Saih. Berikut merupakan alur program dari penelitian ini.



Gambar 1. Flowchart Aplikasi

Langkah pertama yang dilakukan adalah melakukan *input* citra pada kamera *realtime* dan citra referensi (marker). Lalu dilakukan proses deteksi fitur algoritma *ORB (Oriented Fast And Rotated Brief)* untuk kedua citra tersebut. Setelah fitur pada kedua citra terdeteksi dilanjutkan dengan proses deskripsi fitur algoritma *ORB (Oriented Fast And Rotated Brief)*. Kedua fitur deskriptor tersebut akan dicocokkan dengan menggunakan algoritma *Brute Force Hamming*, apabila kedua fitur tidak cocok maka proses akan diulang dari proses *input* citra kamera *realtime* lalu ketika kedua fitur deskriptor tersebut cocok maka proses akan dilanjutkan dengan proses estimasi pose citra dan terakhir menampilkan *output* berupa objek virtual 3D kedalam lingkungan nyata.

## 2.1 Data Penelitian

Data yang dipergunakan dalam merancang penelitian ini terbagi menjadi dua jenis yaitu data citra gambar serta data objek virtual. Data citra gambar ini dapat dikelompokkan lagi menjadi dua macam lagi yaitu citra referensi (marker) yang di *input* kan oleh *user* melalui kamera perangkat yang digunakan dan citra kamera *realtime* yang didapatkan dari *frame* kamera secara *realtime* saat menampilkan 3D. Kedua citra tersebut akan dicocokkan menggunakan algoritma pencocokan *ORB (Oriented Fast And Rotated Brief)* lalu jika kedua citra tersebut tidak cocok maka akan dikembalikan pada proses *input* citra kamera *realtime*. Apabila pada proses pencocokan fitur ditemukan kemiripan antara dua citra tersebut, maka objek 3D akan ditampilkan melalui kamera perangkat yang digunakan secara *realtime*. Data objek 3D adalah data-data dari beberapa komponen berbentuk kekarangan yang diambil dari Cagar Budaya Pura Saih yang merupakan situs Cagar Budaya yang berada dikabupaten Badung. Data ini diperoleh dengan cara melakukan peninjauan langsung yang selanjutnya akan dipergunakan untuk merancang model objek 3D. Adapun rincian dari komponen-komponen yang akan dipergunakan adalah sebagai berikut:

### a) Karang Boma

Karang Boma merupakan kekarangan berbentuk kepala dari bhutakala yang merupakan *symbol* dari ruang dan waktu. Karang boma biasanya ditempatkan pada candi bentar (bangunan pintu masuk) yang ada di bagian tengah suatu bangunan (*bwah loka*). Karang boma mengambil ide/konsep dari cerita Bomantara yang dilahirkan dari adanya pertemuan Waraha Awatara yang digambarkan dalam bentuk babi sebagai jelmaan dari Dewa Wisnu dengan Dewi Pertiwi, atau dapat dikatakan sebagai pertemuan antara air dengan tanah yang menyebabkan terjadinya kehidupan. Karang boma ini diwujudkan dalam bentuk muka raksasa yang digabungkan dengan ornamen papatran dan keketusan dengan makna setiap diri kita menatap karang boma ini diharapkan dapat menyadari bahwa raganya dibatasi oleh ruang dan waktu [11].

### b) Karang Goak/Manuk

Karang guak merupakan sebuah stilisasi dari kepala burung yang memiliki kaitan erat dengan mitos-mitos mengenai tugas serta sifat dari burung sebagai penguasa alam atas (*swah loka*). Mitologi yang dianut oleh masyarakat Bali disampaikan dalam ceritera lahirnya Garuda, Tantri Kamandaka serta Bhomantaka secara verbal atau oral mengenai bagaimana ketangguhan dan keperkasaan dari burung dalam menjalankan tugasnya untuk menjaga kedamaian dunia [7].

### c) Karang Tapel

Karang tapel atau topeng ini diambil dari bentuk wajah manusia raksasa dalam motologi masyarakat bali yang digunakan sebagai penutup muka atau topeng. Bentuk wayah manusia raksasa ini lalu digubah dalam bentuk karang tapel yang juga dikombinasikan dengan beberapa keketusan serta papatran [6]. Karang tapel ditempatkan pada bagian tengah suatu bangunan (*bwah loka*). Wajah manusia raksasa dalam karang tapel diwujudkan dengan mata melotot, mempunyai gigi taring, bentuk bibir tebal, pipi bulat serta memiliki bentuk hidung layaknya manusia.

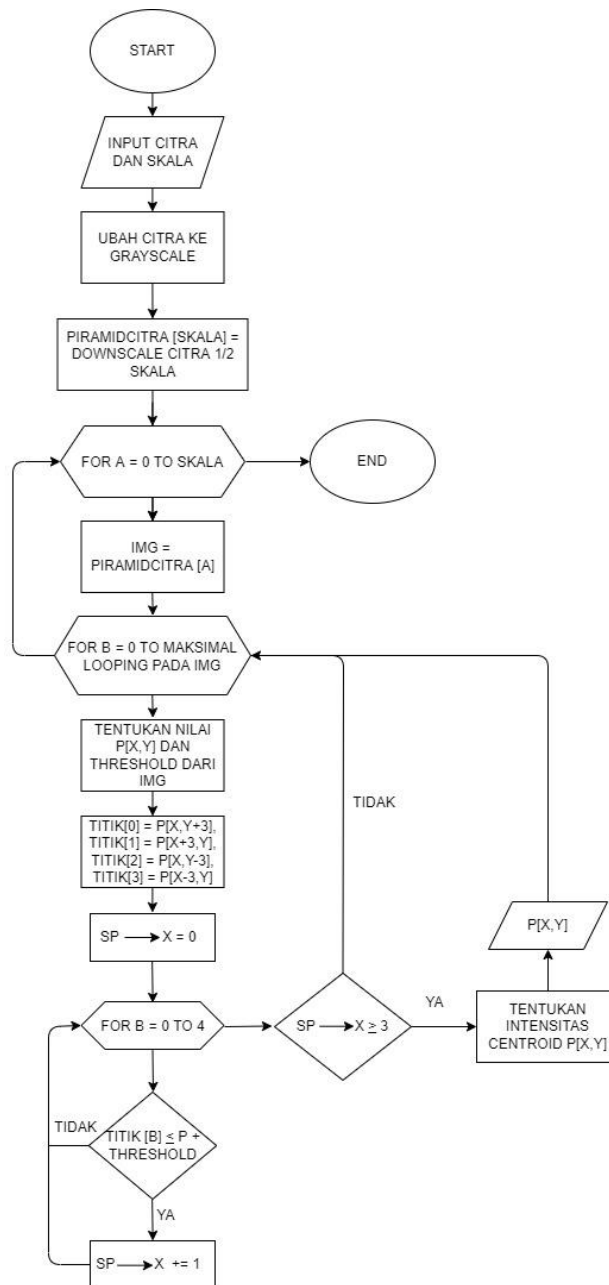
### d) Karang Gajah

Karang Gajah merupakan ukiran yang betuk serta konsepnya diambil dari binatang gajah yang dipercaya mempunyai kekuatan magis yang besar dan diletakan di dasar suatu bangunan (*bhur loka*) pada bagian pojok bawah sebagai lambing atau simbol kekuatan atau kekokohan dari bangunan tersebut [8].

## 2.2 Deteksi Fitur (*Feature Detection*)

Deteksi fitur digunakan untuk mencari dan mengidentifikasi objek dalam citra. Algoritma untuk deteksi fitur harus efisien, yaitu dapat memperoleh hasil dalam persyaratan waktu nyata dan berulang, yaitu memastikan bahwa fitur yang sama akan ada terdeteksi di gambar yang berbeda [2]. Deteksi fitur pada penelitian ini menggunakan algoritma *ORB (Oriented Fast And Rotated Brief)* dengan

penerapan berbasis algoritma *FAST Corner Detection* dengan inputan citra awal akan proses menjadi *grayscale*. Berikut merupakan alur proses pendeteksian fitur pada algoritma *FAST Corner Detection*.

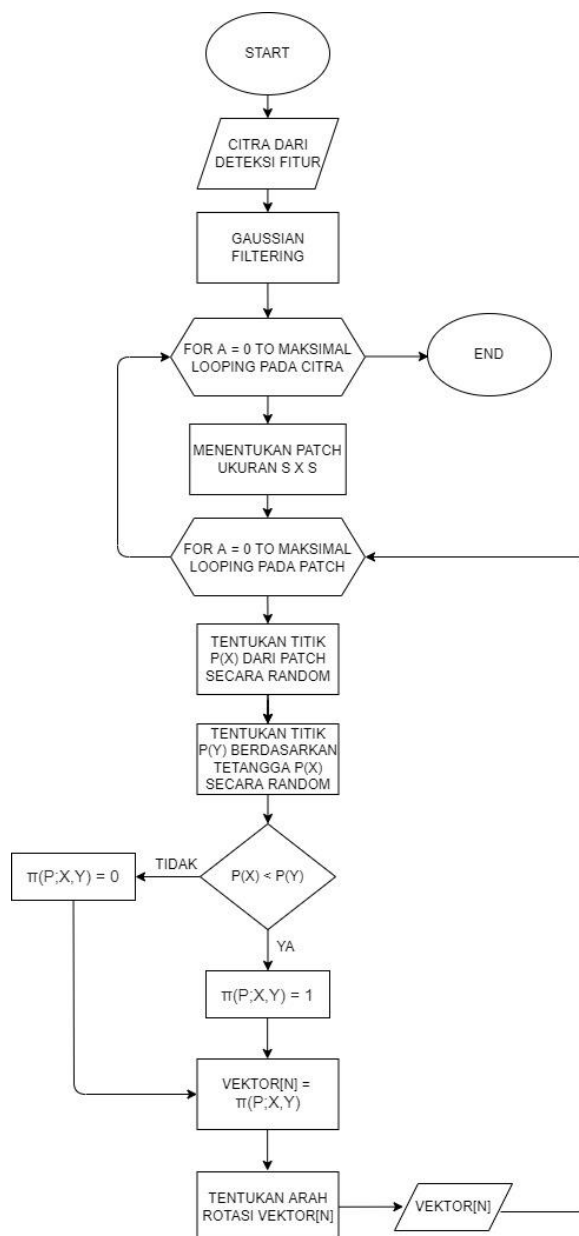


**Gambar 2.** Flowchart Deteksi Fitur

Langkah pertama pada algoritma ini adalah mengubah citra referensi (marker) dan citra kamera *realtime* menjadi *grayscale* agar menjadi 1 kanal warna untuk mempercepat pemrosesan. Selanjutnya dilakukan proses *downscale* sebanyak  $\frac{1}{2}$  skala untuk membuat *Pyramid* Citra dengan perbedaan ukuran  $\frac{1}{2}$  dari ukuran asli dari kedua citra sebelumnya untuk mendapatkan skala yang beragam agar pada proses pencocokan kedua citra tersebut menjadi invarian dengan perubahan rotasi dan skala. Lalu tentukan sebuah *pixel* untuk dijadikan acuan titik deteksi (p) serta tentukan *threshold* untuk dijadikan acuan titik agar dapat melakukan pemisahan sesuai dengan kriteria. Lakukan deteksi pada kedua citra yang terindikasi pada variabel *Pyramid* Citra lalu tentukan empat *pixel* sebagai titik uji berdasarkan radius jarak tiga *pixel* dari titik acuan (p) yang telah ditentukan sebelumnya. Periksa kesesuaian dari setiap nilai uji dengan kriteria algoritma yang digunakan. Jika terdapat paling sedikit tiga dari empat titik uji yang berwarna signifikan dari batas warna yang telah ditentukan, maka dapat ditarik kesimpulan bahwa titik p tersebut merupakan sebuah titik sudut atau *corner*. Selanjutnya tentukan intensitas *centroid* pada setiap titik uji serta skala citra untuk dapat melakukan pemrosesan yang invarian terhadap perubahan rotasi. Proses ini akan menghasilkan titik sudut yang dideteksi dari kedua citra.

### 2.3 Deskripsi Fitur (*Feature Descriptor*)

Tahap deskriptor fitur ini bertujuan untuk mendeskripsikan sebuah wilayah pada daerah sekitar titik kunci agar dapat menghasilkan deskripsi dari vektor sebelumnya [10]. Menurut Carmigniani & Furht (2011) [9] deskriptor fitur harus memiliki skala, dapat melakukan orientasi secara invarian, harus memungkinkan untuk melakukan perhitungan dengan cepat serta harus bisa memberikan gambaran mengenai titik dengan seunik mungkin menggunakan informasi seminimal mungkin. Deskriptor fitur pada penelitian ini menggunakan algoritma *ORB (Oriented Fast And Rotated Brief)* berdasarkan pada algoritma *BRIEF (Binary Robust Independent Elementary Features)* yang selanjutnya akan dikembangkan untuk dapat menghitung jumlah deskriptor pada setiap titik poin untuk menentukan vektor dari deskriptor biner yang terdiri atas 1 dan 0. Berikut adalah alur dari proses deskriptor fitur pada algoritma *BRIEF (Binary Robust Independent Elementary Features)* [3].



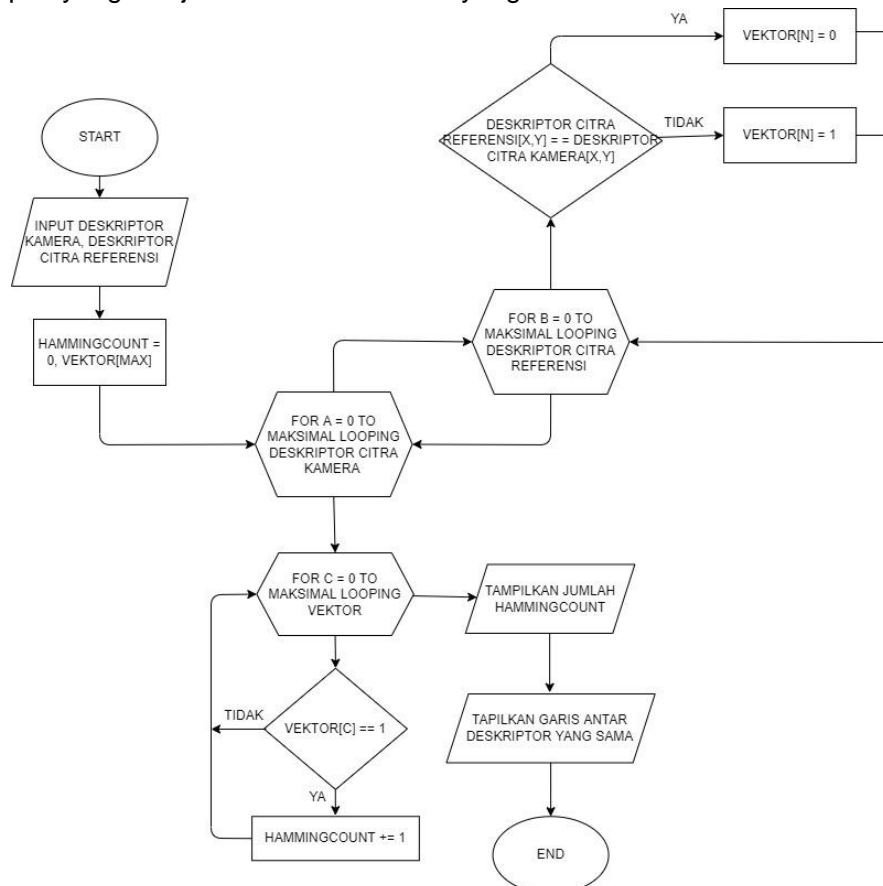
Gambar 3. Flowchart Deskriptor Fitur

Untuk meningkatkan ketahanan terhadap *digital noise* pertama-tama akan diterapkan *gaussian filter* terhadap kedua citra hasil dari deteksi fitur. Proses selanjutnya adalah menentukan ukuran *patch* berdasarkan titik hasil dari deteksi fitur untuk mendapatkan potongan dari kedua citra lalu ditentukan nilai  $p(x)$  dari titik yang berdekatan secara acak dan ditentukan nilai  $p(y)$  berdasarkan ketetanggaan dari  $p(x)$ . Bandingkan kedua titik tersebut apabila titik  $p(x)$  lebih gelap daripada  $p(y)$  maka nilai binernya 1 dan jika titik  $p(x)$  lebih terang dari  $p(y)$  maka nilai binernya 0. Nilai tersebut akan di simpan

pada variabel vektor dan intensitas *centroid* pada proses deteksi fitur akan menjadi acuan vektor untuk menyesuaikan arah dan rotasinya sehingga dapat menghasilkan arah dari setiap garis yang tergabung. Proses ini akan dilakukan secara berulang pada setiap titik hasil dari deteksi fitur pada *patch* lalu selanjutnya akan dibuatkan *patch* baru pada titik fitur lain saat *patch* sebelumnya selesai dibandingkan. Proses ini dilakukan berulang pada kedua citra dan skala citra yang lain. Proses ini akan menghasilkan vektor yang memiliki nilai biner untuk selanjutnya akan dicocokkan pada proses pencocokan fitur.

## 2.4 Pencocokan Fitur (*Feature Matching*)

Pencocokan fitur dilakukan untuk menentukan kesamaan deskriptor titik antara kedua citra. Pencocokan fitur yang dilakukan pada penelitian ini menggunakan algoritma *Brute Force Hamming* yang merupakan gabungan dari dua metode pencocokan yaitu *Brute Force* dan *Hamming Distance*. Metode pencocokan dengan basis titik ini dijadikan metode utama dalam melakukan pencocokan citra karena metode ini menggunakan perhitungan sederhana dengan akurasi pencocokan tinggi, dapat invarian dengan distorsi grafis, *grayscale* ataupun pencahayaan dengan waktu yang singkat [3]. Metode *Brute-Force Matcher* akan melakukan pengukuran pada bentangan jarak titik fitur lalu akan dipilih jarak terdekat berdasarkan semua hasil pengukuran deskriptor sebagai titik pencocokan. Metode *Hamming Distance* bekerja dengan cara melakukan pengukuran terhadap jumlah karakter tidak cocok atau berbeda pada dua deskriptor *ORB (Oriented Fast And Rotated Brief)* yang memiliki panjang yang sama. Nilai *Hamming Distance* menggambarkan banyaknya jumlah perbedaan dari kedua deskriptor yang menjadi tolak ukur dari citra yang berbeda.



**Gambar 4.** Flowchart Pencocokan Fitur

Langkah pertama pada algoritma ini adalah menginput deskriptor dari kedua citra lalu buat variabel untuk menyimpan nilai total *Hamming* yang terjadi dan buat vektor untuk menyimpan hasil proses. Lakukan proses *Brute-Force* yaitu mencocokkan satu persatu dari kedua citra deskriptor tersebut secara berulang. Lalu terapkan algoritma *Hamming Distance* yaitu memberikan nilai 1 untuk setiap perbedaan deskriptor dan memberikan nilai 0 untuk setiap kesamaan deskriptor sesuai dengan aturan XOR secara berulang sampai semua deskriptor dari kedua citra tersebut habis dibandingkan lalu simpan nilai-nilai tersebut pada variabel vektor. Tentukan nilai kesalahan pada deskriptor dengan cara melakukan pengujian secara berulang pada vektor yang ada. Selanjutnya tambahkan variabel

*Hamming Count* pada setiap vektor yang bernilai 1 sebagai penanda jumlah kesalahan yang didapat pada setiap proses perbandingan deskriptor. Jika deskriptor kedua citra tersebut cocok maka akan dilanjutkan dengan proses selanjutnya dan jika tidak maka akan dikebalikan ke proses input citra kamera secara *realtime* sebelum melakukan deteksi fitur.

## 2.5 Penyempurnaan dan Estimasi Pose

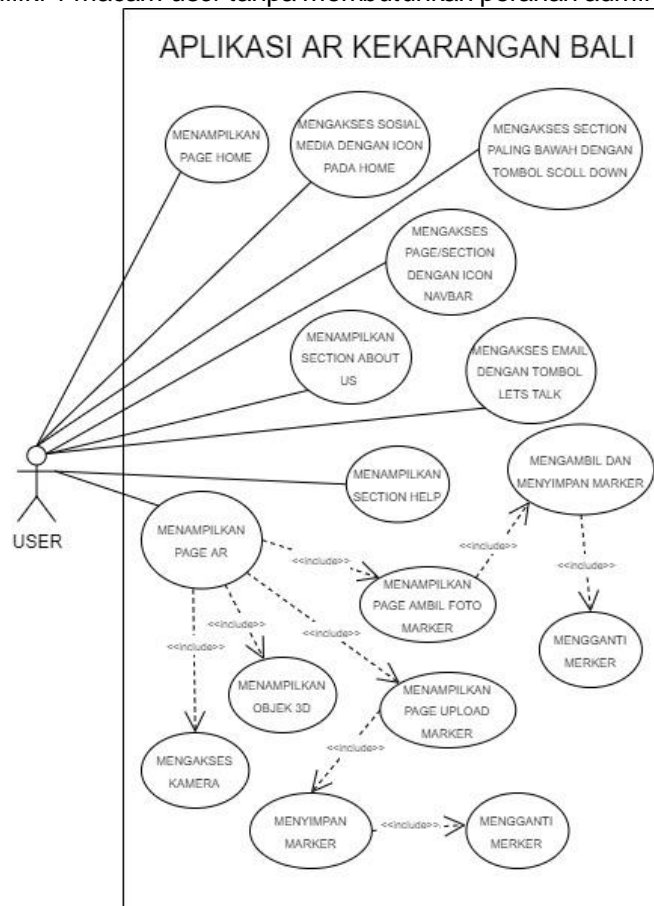
Estimasi dan penyempurnaan pose akhir ini berkaitan dengan penentuan letak posisi dari bingkai yang akan dilakukan setelah semua fitur dicocokkan. Biasanya kesalahan pencocokan (*outlier*) terjadi apabila sebagian fitur yang terdeteksi memberikan respon pasangan fitur yang tidak valid karena banyak pose yang diekstraksi. Untuk meminimalisir resiko tersebut akan diterapkan fungsi *RANSAC* (*Random Sample Consensus*) untuk menyaring *outlier* yang ada. Menurut Khoharja, Liliana, & Anita Nathania Purbowo (2017) [4] *RANSAC* (*Random Sample Consensus*) adalah suatu prosedur untuk melakukan estimasi dari transformasi yang terjadi pada suatu gambar/objek. *RANSAC* (*Random Sample Consensus*) dipergunakan sebagai alat penyaring vektor yang tidak cocok atau *false match* yang masih banyak dihasilkan pada proses pencocokan fitur. Cara kerja dari *RANSAC* (*Random Sample Consensus*) yang pertama yaitu memilih empat pasang fitur secara acak lalu dibentuk garis lurus secara vertikal ataupun horizontal yang digunakan sebagai penanda titik yang akan diseleksi. Lakukan penghitungan pada setiap titik yang telah terdeteksi diantara garis tersebut secara berulang sebanyak *n* untuk mendapatkan garis dengan nilai titik terbanyak. Titik yang berada diluar garis akan dieliminasi dan didapatkanlah garis optimasinya.

## 2.6 Desain Perancangan Aplikasi

Desain perangkat lunak merupakan tahapan *developer* memulai mendesain sebuah aplikasi untuk memenuhi setiap persyaratan dari analisis sebelumnya. Desain dari aplikasi ini adalah sebagai berikut:

### a) Usecase Diagram

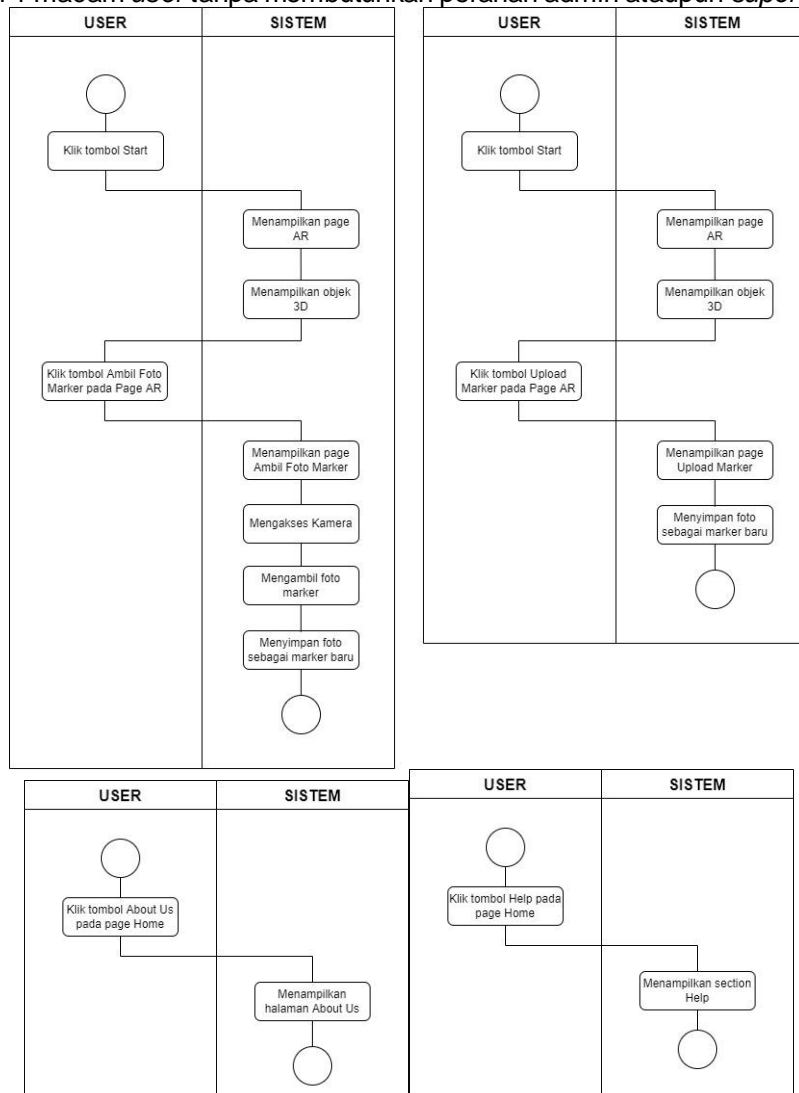
Berikut merupakan gambaran *use case diagram* aplikasi yang dipergunakan sebagai acuan dalam merencanakan tingkah laku *user* saat mengakses aplikasi ini. Pengguna aplikasi ini hanya memiliki 1 macam *user* tanpa membutuhkan peranan admin ataupun *super user*.



Gambar 5. Use Case Diagram

b) Activity Diagram

Berikut merupakan gambaran *activity diagram* aplikasi yang dipergunakan untuk memperlihatkan bagaimana siklus kerja mengakses aplikasi ini. Pengguna aplikasi ini hanya memiliki 1 macam *user* tanpa membutuhkan peranan admin ataupun *super user*.



Gambar 6. Activity Diagram

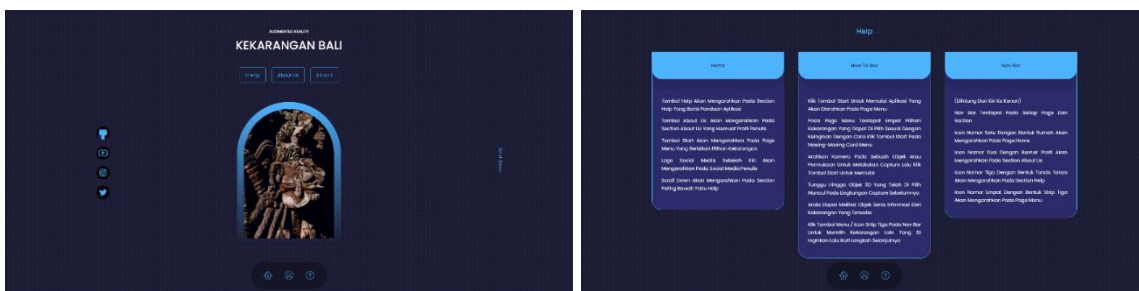
3. Hasil dan Pembahasan

Pada bagian hasil dan pembahasan ini, akan membahas hasil implementasi antarmuka serta pengujian dari aplikasi ini sebagai berikut:

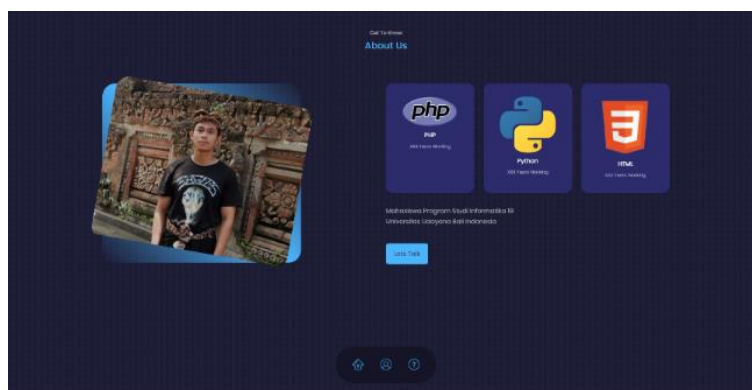
3.1 Implementasi Antarmuka

Implementasi antarmuka aplikasi ini terbagi menjadi 3 page yaitu *Home*, *Ambil Foto Marker*, *Upload Marker* dan *AR* serta terdapat 2 section pada page *Home* yaitu *Help* dan *About Us* dengan penjelasan sebagai berikut:

a) Tampilan Antarmuka page *Home*, *Help* dan *About Us*.

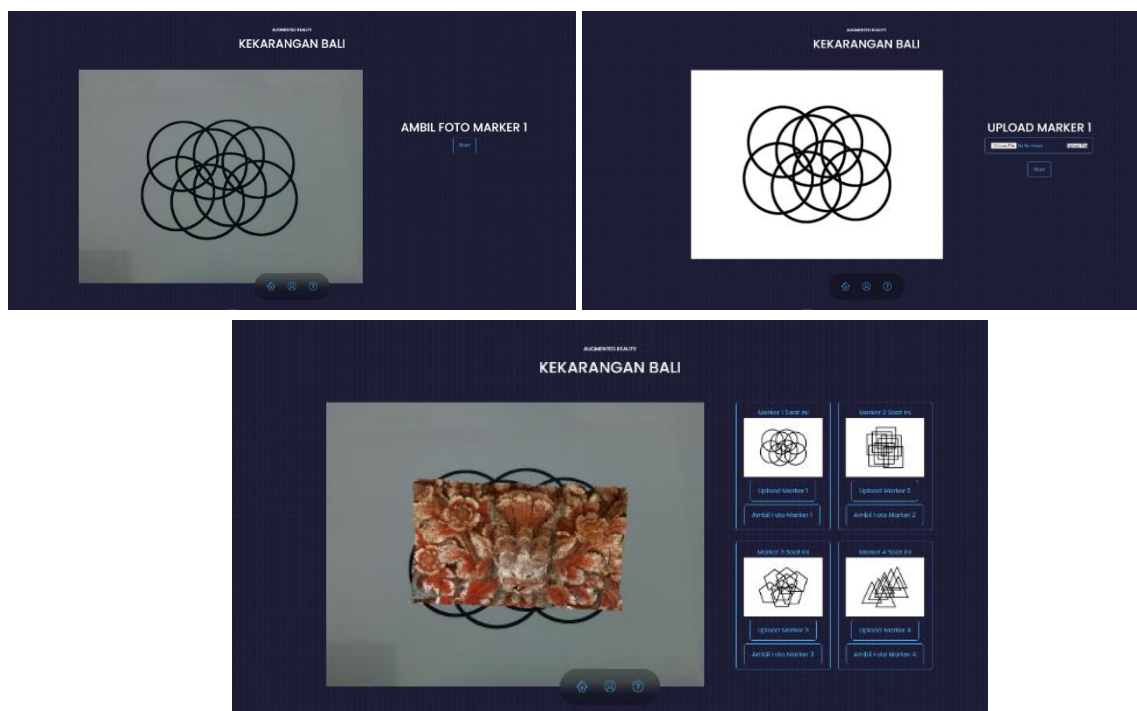






Gambar 7. Tampilan Antarmuka page Home, Help dan About Us.

b) Tampilan Antarmuka page Upload Marker, Ambil Foto Marker dan AR



Gambar 8. Tampilan Antarmuka page Upload Marker, Ambil Foto Marker dan AR

### 3.2 Pengujian Aplikasi

Pengujian aplikasi ini akan dilakukan dalam 3 tahap yaitu Pengujian Penampilan Objek 3D untuk mengetahui akurasi dan waktu dari penampilan objek 3D dengan jarak yang berbeda, Pengujian Citra Marker untuk mengetahui akurasi dan waktu yang dihasilkan dari proses ekstraksi citra dengan marker yang berbeda serta Evaluasi *Heuristik* untuk mengetahui tingkat kemudahan dalam menjalankan aplikasi ini.

#### 3.2.1 Pengujian Penampilan Objek 3D

Pengujian ini melakukan 2 tahapan yaitu dari Akurasi Tampil dan Waktu Tampil menggunakan 2 macam inputan marker yang berbeda dengan pola dan ukuran yang sama seperti pada gambar 4.7 yaitu dengan cara diambil foto secara langsung seperti gambar 4.5 (Ambil Foto Marker) serta menggunakan marker yang dibuat pada *Photoshop* lalu di *upload* seperti pada gambar 4.6 (*Upload* Marker).

##### 1) Akurasi Tampil

Pengujian akurasi tampil 4 objek 3D kecharangan dilakukan pada masing-masing 20 *frame* pada jarak 20cm, 25cm, 30cm, 35cm dengan menggunakan 2 inputan marker yaitu Ambil Foto Marker dan *Upload* Marker. Pengujian ini akan menghitung jumlah poin yang diperoleh lalu akan ditotal dan dibagi dengan jumlah keseluruhan pengujian (20 *frame*) lalu dikali 100% untuk mendapatkan kesimpulan yaitu jarak terbaik dalam akurasi tampil objek 3D. Berikut

merupakan tabel hasil pengujian akurasi tampil dari 4 model 3D kekarangan untuk mendapatkan jarak dengan akurasi terbaik.

**Tabel 1.** Hasil Pengujian Akurasi Tampil (Ambil Foto Marker)

No	Hasil	Jarak				Total Poin	Persentase
		20cm	25cm	30cm	35cm		
1	Objek tampil dengan posisi benar	6	18	20	20	64	80%
2	Objek tampil dengan posisi salah	2	2	0	0	4	5%
3	Objek tidak tampil	12	0	0	0	12	15%

**Tabel 2.** Hasil Pengujian Akurasi Tampil (*Upload* Marker)

No	Hasil	Jarak				Total Poin	Persentase
		20cm	25cm	30cm	35cm		
1	Objek tampil dengan posisi benar	8	13	17	17	55	68.75%
2	Objek tampil dengan posisi salah	1	1	0	0	4	2.50%
3	Objek tidak tampil	11	6	3	3	23	28.75%

2) Waktu Tampil

Pengujian waktu tampil 4 objek 3D kekarangan akan dilakukan pada masing-masing jarak 20cm, 25cm, 30cm, 35cm dengan menggunakan 2 inputan marker yaitu Ambil Foto Marker dan *Upload* Marker. Pengujian ini akan menghitung perolehan waktu dalam menampilkan 4 objek 3D kekarangan untuk menentukan jarak tercepat dalam proses ekstraksi citra. Berikut merupakan tabel hasil pengujian waktu tampil dari 4 model 3D kekarangan untuk mendapatkan jarak dengan waktu tercepat.

**Tabel 3.** Hasil Pengujian Waktu Tampil (Ambil Foto Marker)

	20cm	25cm	30cm	35cm
Karang Boma	8.03 detik	8.36 detik	8.79 detik	9.04 detik
Karang Goak	7.59 detik	7.84 detik	8.09 detik	8.34 detik
Karang Tapel	7.21 detik	7.55 detik	7.72 detik	7.97 detik
Karang Gajah	8.22 detik	8.63 detik	9.01 detik	9.30 detik

**Tabel 4.** Hasil Pengujian Waktu Tampil (*Upload* Marker)

	20cm	25cm	30cm	35cm
Karang Boma	8.12 detik	9.08 detik	10 detik	10.70 detik
Karang Goak	7.37 detik	7.86 detik	8.42 detik	9 detik
Karang Tapel	7 detik	7.57 detik	8.16 detik	8.48 detik
Karang Gajah	8.18 detik	8.83 detik	9.44 detik	10 detik

**3.2.2 Pengujian Citra Marker**

Pengujian ini dilakukan pada 4 macam marker dengan inputan yang berbeda yaitu dengan cara diambil foto secara langsung seperti pada gambar 4.5 (Ambil Foto Marker) serta menggunakan marker yang dibuat pada Photoshop lalu diupload seperti pada gambar 4.6 (*Upload* Marker) untuk menguji akurasi kecocokan dari marker yang digunakan sehingga akan didapat marker terbaik yang dilihat dari jumlah *matches* dan *inliers* serta untuk menguji waktu yang dibutuhkan dari masing-masing marker yang digunakan sehingga akan didapat marker tercepat untuk dideteksi. Dari pengujian penampilan objek 3D sebelumnya dipilih jarak 30cm karena jarak tersebut memiliki akurasi paling besar untuk menampilkan objek 3D dengan benar serta memerlukan waktu paling sedikit. Berikut merupakan 4 macam marker yang akan digunakan dengan inputan yang berbeda.

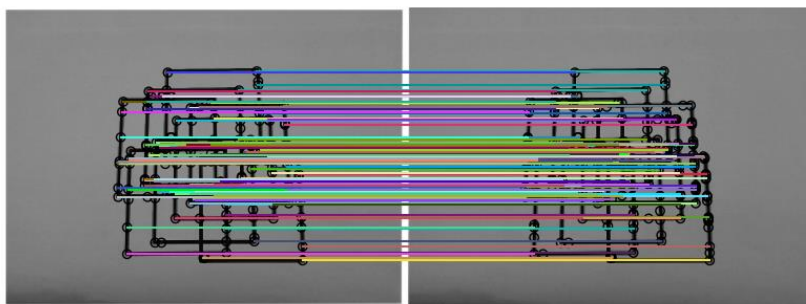
1) Pengujian Marker

Pengujian marker ini akan menghitung akurasi kecocokan serta waktu yang dibutuhkan dengan menggunakan 4 macam marker dengan menggunakan 2 inputan marker yaitu Ambil Foto Marker dan *Upload* Marker dengan jarak 30cm. Pengujian ini akan menghasilkan marker terbaik yang dilihat dari jumlah *matches* dan *inliers* serta untuk menguji waktu yang dibutuhkan dari masing-masing marker yang digunakan sehingga akan didapat marker

tercepat untuk dideteksi. Berikut merupakan tabel hasil pengujian marker dengan input Ambil Foto Marker dan *Upload Marker*.

**Tabel 5.** Hasil Pengujian Marker (Ambil Foto Marker)

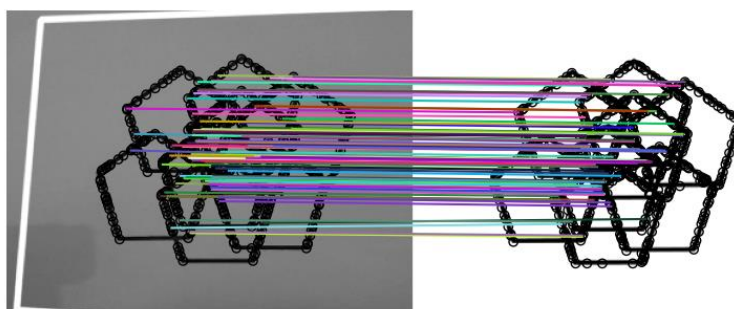
No	Hasil	Matches	Inliers	Persentase Kecocokan	Waktu
1	Marker 1	422	394	93.36%	8.79 detik
2	Marker 2	157	150	95.54%	4.32 detik
3	Marker 3	354	324	91.52%	7.61 detik
4	Marker 4	317	269	84.85%	4.87 detik



**Gambar 9.** Pengujian Marker (Ambil Foto Marker)

**Tabel 6.** Hasil Pengujian Marker (*Upload Marker*)

No	Hasil	Matches	Inliers	Persentase Kecocokan	Waktu
1	Marker 1	235	87	37.02%	10 detik
2	Marker 2	59	21	35.59%	4.20 detik
3	Marker 3	231	88	38.09%	6.32 detik
4	Marker 4	196	93	47.44%	6.09 detik



**Gambar 10.** Pengujian Marker (*Upload Marker*)

### 3.2.3 Evaluasi *Heuristik*

Evaluasi *Heuristik* bertujuan untuk mengevaluasi *design* tampilan aplikasi dengan menguji permasalahan *usability* terkait dengan tingkat kemudahan dalam menjalankan aplikasi yang ditinjau dari responden mahasiswa Informatika Udayana yang di *rating* dengan menggunakan *severity rating*.

**Tabel 7.** Hasil Evaluasi *Heuristik*

Aspek <i>Heuristik</i>	Rata-Rata <i>Severity Rating</i>	Nilai Pembulatan Skala 0-4
1	0.8	1
2	1.9	2
3	1.9	2
4	1.3	1
5	2	2
6	2.6	3
7	1.6	2
8	1.3	1
9	2.2	2
10	1.9	2
Rata-Rata	1.75	2

#### 4. Kesimpulan

- 1) Pengujian penampilan 4 objek 3D menggunakan pola yang sama dengan inputan marker yang berbeda pada masing-masing 20 *frame* dengan jarak 20cm, 25cm, 30cm, 35cm dapat dilihat bahwa jarak terbaik adalah 30cm dan 35cm dengan inputan Ambil Foto Marker dengan persentase tampil dengan posisi benar sebesar 80% pada model mamapun dengan perolehan waktu rata-rata pada setiap objek 3D adalah 8.40 detik pada jarak 30cm dan 8.66 detik pada jarak 35cm. Jarak tercepat adalah 20cm dengan inputan *Upload* Marker dengan perolehan waktu 7 detik pada model karang tapel.
- 2) Pengujian 4 citra marker menggunakan inputan marker berbeda dengan jarak yang sama yaitu 30cm dapat dilihat bahwa marker terbaik adalah marker 2 yang dibangun dengan garis berbentuk segiempat sebanyak 8 buah dengan inputan Ambil Foto Marker dengan persentase kecocokan sebesar 95.54% dengan waktu 4.32 detik. Marker tercepat adalah marker 2 dengan inputan *Upload* Marker dengan perolehan waktu 4.20 detik dengan persentase kecocokan sebesar 35.59%.
- 3) Dari evaluasi *heuristic* dapat dilihat bahwa aspek *usability* dari aplikasi termasuk kedalam skala 2 *severity rating* atau termasuk dalam kategori *minor usability problem* yang artinya permasalahan kegunaan kecil dengan prioritas yang rendah.

#### Referensi

- [1] A. Arifianto, "Rancang Bangun Aplikasi Pendeteksi Plagiarisme Kode Program Bahasa C Menggunakan algoritma Levenshtein Distance dan Brute Force," *UMN Universitas Multimedia Nusantara.*, 2011.
- [2] A. Jakubovic and J. Velagic, "Image Feature Matching and Object Detection Using Brute-Force Matchers," *Conference: 2018 International Symposium ELMAR. Zadar: Institute of Electrical and Electronics Engineers (IEEE).*, 2018.
- [3] C. Luo, W. Yang, P. Huang and J. Zhou, "Overview of Image Matching Based on ORB Algorithm," *Journal of Physics: Conference Series.*, Vol. 1237, Issue 3, 2019.
- [4] G. Khoharja, M. Liliana and M. A. N. Purbowo, "Aplikasi Deteksi Nilai Uang pada Mata Uang Indonesia," *Jurnal INFRA.*, 2017.
- [5] I. G. L. T. Sumantara, I. P. A. Bayupati and N. K. A. Wardani, "Rancang Bangun Aplikasi Pengenalan Ukiran Bali dengan Metode ORB," *MERPATI: Menara Penelitian Akademika Teknologi Informasi.*, Vol 5. Hal 51-56, 2017.
- [6] I. M. J. Waisnawa and T. H. Yupardhi, "PENGEMBANGAN ORNAMEN TRADISIONAL BALI," *ISI Denpasar | Institutional Repository.*, 2014.
- [7] I. M. Suparta, "Jenis Hiasan Tatahan Bade," *IMAJI: Jurnal Seni Dan Pendidikan Seni.*, Vol. 8, Hal 81-91, 2010.
- [8] I. M. W. Permadi, "Aplikasi Desktop Pendeteksi Jenis Ukiran Khas Bali dengan Metode Invariant Moments," *e-perpus.unud.ac.id.*, 2014.
- [9] J. Carmigniani and B. Furht, "Handbook Of Augmented Reality," *Florida: Springer-Verlag.*, 2011.
- [10] M. Hamidia, N. Zenati-Henda, H. Belghit and A. Bellarbi, "Object Recognition Based on ORB Descriptor for Markerless Augmented Reality," *9ème Conférence sur le Génie Electrique.*, 2015.
- [11] N.K. Sutriyanti, "PELINGGIH PADMATIGA PENATARAN AGUNG BESAKIH (Analisis Bentuk, Fungsi, Dan Makna)," *Vidya Samhita: Jurnal Penelitian Agama.*, Hal 76-96, 2015.
- [12] R. Abdeen, "An Algorithm for String Based on Brute Force Algorithm," *Internasional Journal of Computer Science and Network Security.*, Vol. 11 : 7, 2011.
- [13] S. Čuković, M. Gattullo, F. Pankratz and G. Devedžić, "Marker Based vs. Natural Feature Tracking Augmented Reality Visualization of the 3D Foot Phantom," *ResearchGate.*, 2015.