

## Penerapan Algoritma Surf Pendeteksi Objek Pada *Augmented Reality* Berbasis Android

Amir Alkodri<sup>1</sup>, Harrizki<sup>2</sup>, Suharno<sup>3</sup>

<sup>1,2</sup>STMIK Atma Luhur, Jl. Jend Sudirman, Selindung, Pangkalpinang. Kep. Bangka Belitung

<sup>3</sup> Program Studi Teknik Informatika, STMIK Atma Luhur, Bangka Belitung

e-mail: \*<sup>1</sup>arie\_a3@atmaluhur.ac.id, <sup>2</sup> harrizkiariep@atmaluhur.ac.id, <sup>3</sup> suharno@atmaluhur.ac.id

### *Abstrak*

Untuk penerapan konsep yang efektif dan efisien dalam pembelajaran menggunakan teknologi *augmented reality* yang interaktif. Pembelajaran multimedia merupakan bentuk pembelajaran secara interaktif. Pembelajaran dapat dilakukan dikelas maupun dalam bentuk tugas dirumah. Dampak dari pembelajaran interaktif ini sangat mempengaruhi kualitas pemahaman bagi pengguna. Berdasarkan pengujian dari teknologi *augmented reality*, penelitian, analisis dan analisis desain antarmuka interaktif, proses dan prinsip dilakukan melalui interaksi manusia dan komputer serta pengalaman pengguna. Metode desain antarmuka menggunakan metode *Object Oriented Programming* dan algoritma *Speeded Up Robust Features*. Hasil penelitian menunjukkan bahwa *augmented reality* interaktif dapat memberikan metode baru dalam proses pembelajaran dan memperbarui konten pembelajaran di kelas. Singkatnya, metode ini meningkatkan partisipasi dan antusiasme siswa, dan meningkatkan efektivitas pembelajaran.

**Kata kunci:** *Augmented Reality*, Interaktif, *Speeded Up Robust Features*

### *Abstract*

For the application of effective and efficient concepts in learning using interactive augmented reality technology. Multimedia learning is a form of interactive learning. Learning can be done in class or in the form of assignments at home. The impact of interactive learning greatly affects the quality of understanding for users. Based on testing from augmented reality technology, research, analysis and analysis of interactive interface design, processes and principles are carried out through human and computer interaction as well as user experience. The interface design method uses the Object-Oriented Programming method and the Speeded Up Robust Features algorithm. The results showed that interactive augmented reality can provide new methods in the learning process and update learning content in the classroom. In short, this method increases student participation and enthusiasm, and increases learning effectiveness.

**Keywords:** Augmented Reality, Interactive, Speeded Up Robust Features

## 1. PENDAHULUAN

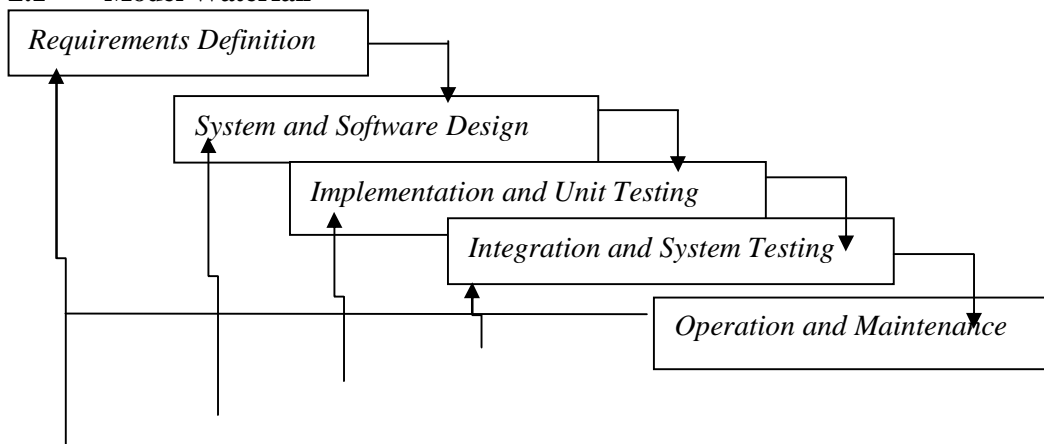
Pesatnya perkembangan teknologi komputer, teknologi interaksi antara manusia dan komputer telah diperbarui secara interaktif. Interaksi manusia terhadap komputer mengacu pada proses pertukaran informasi antara seseorang dan komputer menggunakan bahasa dialog tertentu dan berinteraksi satu sama lain untuk menyelesaikan tugas tertentu. Dalam proses desain interaksi saat ini, interaksi manusia terhadap komputer telah berjalan dan diteliti di dalam maupun di luar industri [1]. *Augmented Reality* (disebut sebagai AR) dapat dianggap sebagai salah satu dari *Virtual Reality* (VR). *Augmented reality* adalah penggunaan komputer untuk menciptakan lingkungan virtual dengan perasaan visi, pendengaran, kekuatan, sentuhan, dan

gerakan yang realistis. Dengan melalui beberapa perangkat sensor dalam lingkungan *virtual* untuk mencapai interaksi alami langsung antara pengguna dan lingkungan. Ini dapat mensimulasikan lingkungan antarmuka interaksi manusia dan komputer yang canggih menggunakan komputer berdasarkan interaksi dan konsep. Pengguna tidak bisa hanya mengalami perasaan mendalam yang dialami dalam dunia fisik objektif melalui sistem realitas *virtual*, tetapi juga dapat menembus ruang, waktu, dan batasan objektif lainnya, dan merasakan pengalaman yang tidak dapat dialami di dunia nyata. *Augmented reality* menyediakan metode penggabungan dunia *virtual* dengan dunia nyata. Itu adalah salah satu mode pengembangan penting dari era antarmuka interaktif 3D.

Dalam penelitian ini, selama era interaksi manusia dan komputer itulah penulis berusaha menggunakan teknologi *augmented reality* sebagai titik awal untuk menganalisis persyaratan antarmuka interaktif berdasarkan *augmented reality* [2]. Dari aspek interaksi manusia dan komputer, pengalaman pengguna, komunikasi visual, desain emosional, dan aspek lainnya, desain, konten, aliran, dan prinsip-prinsip antarmuka interaktif dipelajari dan dianalisis untuk merangkum metode desain antarmuka interaktif berdasarkan *augmented reality* [3]. Dengan menggunakan teknologi *augmented reality* di bidang pendidikan mahasiswa sebagai latar belakang, metode desain antarmuka interaktif dipraktikkan dan hasil penelitiannya dilakukan pengujian. Dalam desain antarmuka yang bertujuan untuk mengeksplorasi dan memulai, sehingga pengguna dapat belajar dari pengembangan umum, dan terus memperdalam desain antarmuka di komputer pada bidang pendidikan, bidang pemerintahan, bidang keilmuan, dan perusahaan-perusahaan lainnya [4]. Dalam penelitian ini dilakukan pembuatan aplikasi *augmented reality* yang interaktif 3D pembelajaran berbasis android Penerapan Algoritma SURF Pendeteksi Objek pada *Augmented Reality*.

## 2. METODE PENELITIAN

### 2.1 Model Waterfall



Gambar 1. Model Waterfall

#### 1. Requirements Definition

Pada tahap pertama model *Waterfall* ini proses pencarian kebutuhan (*Requirements Definition*) bertujuan untuk mengetahui sifat dari program yang akan dibuat. Penulis mengumpulkan data bagaimana proses terjadinya gerhana matahari dan bulan, perputaran objek, bayangan, cahaya dan lain-lain. Serta fungsi yang dibutuhkan seperti *User Interface*, *output* suara yang akan menjelaskan proses bagaimana kejadian tersebut bisa terjadi dan lain-lain. *Software* pengolah objek 3D, bahasa pemrograman JavaScript, C# akan dibutuhkan pada pembuatan aplikasi *m-learning* ini.

## 2. *System and Software Design*

Tahap kedua adalah pembuatan *design* sistem sebelum proses *coding* mulai, tahap ini dapat dimulai dari membuat objek yang akan ditampilkan dalam bentuk tiga dimensi, persiapan *output* suara, *user interface*, fitur-fitur yang ada ini akan membantu *user* dalam menggunakan aplikasi.

### 3. *Implementation and Unit Testing*

Pada tahap ketiga *waterfall* ini *design* harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap *design*.

### 4. *Integration and System Testing*

Tahap ke empat adalah proses uji coba aplikasi dari kebutuhan setiap tahap sebelumnya untuk memastikan tidak ada lagi *error* atau *bug* dalam aplikasi yang dibuat.

### 5. *Operation and Maintenance*

Perbaikan dibutuhkan bila aplikasi mengalami *error* yang tidak ditemukan sebelumnya serta pengembangan aplikasi seperti penambahan fitur baru.

## 2.2 *Metode Object Oriented Programming*

Pemrograman berorientasi objek digunakan dalam penulisan kali ini karena memiliki banyak keunggulan dalam penanganan tugas yang kompleks. Pemrograman menggunakan bahasa berorientasi objek mengikuti konsep-konsep berorientasi objek terutama dalam hal fleksibilitas, kegunaan, dan kemudahan pemeliharaan. Metode ini akan mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang dilakukan terhadapnya.

## 2.3 *Tools Pengembangan Sistem*

Pada penulisan kali ini alat atau *tools* yang digunakan untuk mengembangkan sistem adalah *tools* UML (*Unified Modeling Language*), yang terdiri dari:

### 1. *Use Case Diagram*

Berfungsi untuk menggambarkan hubungan antara *user* (pengguna) dengan sistem aplikasi. Melalui diagram *usecase* dapat diketahui fungsi apa saja yang terdapat pada sistem aplikasi *augmented reality* ini.

### 2. *Activity Diagram*

*Activity diagram* menampilkan aktifitas yang terjadi pada *usecase*, bukan apa yang dilakukan aktor, namun aktivitas yang dapat dilakukan oleh sistem aplikasi.

### 3. *Class Diagram*

*Class Diagram* merupakan inti dari pengembangan dan *design* berorientasi objek, serta bagaimana caranya agar pengguna dengan sistem aplikasi bisa saling berhubungan untuk mendapatkan hasil yang diinginkan.

### 4. *Sequence Diagram*

*Sequence Diagram* menggambarkan interaksi antar objek berupa pengiriman data antar objek dalam urutan waktu. Interaksi antar objek tersebut dapat berupa pesan (*messages*), pengguna (*user*), dan tampilan (*display*).

## 2.4 *Algoritma Speeded Up Robust Features*

Algoritma ini penulis gunakan karena algoritma SURF memiliki kemampuan deteksi citra yang cepat dan dapat mendeskripsikan citra yang terdeteksi secara unik. Memiliki ketahanan terhadap transformasi citra seperti perubahan rotasi, skala, pencahayaan, gangguan *noise* dengan intensitas tertentu, dan perubahan sudut pandang. Algoritma SURF merupakan pengembangan dari algoritma SIFT dimana SURF memanfaatkan kecepatan komputasi tapis kotak dengan menggunakan citra integral. Berikut adalah tahap-tahap bagaimana algoritma SURF berjalan [5]:

1. Konversi citra RGB ke *Grayscale*

Tahap awal dari algoritma SURF adalah mempersiapkan citra masukan dengan format *grayscale 32-bit*. Citra integral  $I$  adalah representasi tengah (*intermediate*) untuk citra dan terdiri dari jumlah nilai keabuan dari citra  $N$  dengan tinggi =  $y$  dan lebar =  $x$ . Perumusannya pada gambar 2.

$$I(x,y) = \sum_{x'=0}^x \sum_{y'=0}^y N(x',y')$$

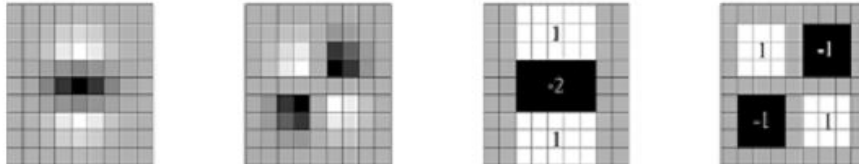
Gambar 2. Konversi Citra RGB ke *Grayscale* [6]

2. Mendeteksi Titik Fitur

Dari fitur *Haar-like* maka hanya dipilih fitur gelembung (*blob-like feature*) untuk mendeteksi titik-titik fitur. Adapun pertimbangan pemilihan fitur gelembung adalah untuk mengurangi jumlah fitur yang terdeteksi dan lebih mempercepat waktu komputasi. Pada tahap kedua ini terbagi lagi menjadi beberapa langkah yang terdiri dari [6]:

a. Pembentukan Piramid Citra

Untuk membentuk piramid citra digunakan *box filter* sebagai aproksimasi dari turunan parsial kedua dari *Gaussian* pada gambar 3.



Gambar 3. Turunan Parsial Kedua dari *Gaussian*

b. Mencari ekstrema dari determinan matriks Hessian

Citra dengan skala yang lebih kecil dibentuk dari konvolusi citra  $I$  dengan *up-scaling box filter* sebelumnya. Pada tahap pendeteksian fitur digunakan matriks *Hessian* karena memiliki performa yang baik dalam kecepatan waktu komputasi dan akurasi. Matriks *Hessian* di titik  $x = (x, y)$  dari citra  $I$  dengan skala  $\sigma$  pada gambar 4.

$$D = (A+B+C+D)-(A+B)-(A+C)+A$$

Gambar 4. Matriks *Hessian*

Agar fitur yang terdeteksi tahan terhadap penskalaan maka dicari ekstrema dari matriks *Hessian* dengan perumusan pada gambar 5.

$$\det(H_{approx} = D_{(xx)}D_{yy} - (0,9D_{xy})^2)$$

Gambar 5. Perumusan Ekstrema Matriks *Hessian*

Kemudian ekstrema dari determinan matriks *Hessian* di interpolasikan pada skala ruang dengan metode yang diajukan oleh Brown. Metode ini akan diterapkan pada setiap calon fitur untuk mencari lokasi ekstrema setelah di interpolasi. *3 Dquadratic* menggunakan ekspansi Taylor terhadap fungsi *scale-space*,  $D(x, y, \sigma)$ , yang digeser sedemikian rupa sehingga titik aslinya digunakan sebagai titik uji pada gambar 6.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$

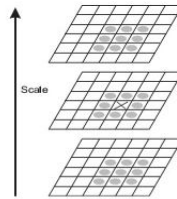
Gambar 6. Perumusan Titik Uji

Dimana  $D$  dan turunannya dihitung pada titik uji dan  $x = (x, y, \sigma)^T$  adalah simpangan dari titik uji. Sedangkan lokasi ekstremum dapat dihitung, pada gambar 7.

$$\hat{x} = \frac{\partial^2 D^{-1}}{\partial x^2} \times \frac{\partial D}{\partial x}$$

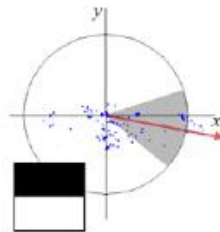
Gambar 7. Perumusan Lokasi Ekstremum

SURF beroperasi dengan cara mengurangi ruang pencarian dari kemungkinan titik penting. Pada gambar 8 memperlihatkan proses *non-max suppression* untuk mendapatkan *keypoints*.



Gambar 8. *Non-max Suppression* [7]

Agar tahan terhadap rotasi, maka setiap *keypoints* yang terdeteksi akan diberikan orientasi. Pertama akan dihitung respon *Haar-wavelet* terhadap sumbu  $-x$  dan sumbu  $-y$  dengan titik-titik di lingkungan tetangganya seperti pada gambar 9.



Gambar 9. Pemberian Orientasi [8]

Proses awal yang akan dilakukan adalah mencocokkan orientasi yang dihasilkan berdasarkan informasi dari daerah yang berbentuk lingkaran disekitar pixel. Kemudian membuat daerah berbentuk kotak pada orientasi yang terpilih dan mengekstrak deskriptor SURF dari daerah tersebut, selanjutnya proses *matching* fitur antara dua gambar dilakukan.

### 3. Inisiasi

Langkah pertama inisiasi adalah aplikasi akan melakukan pengaturan pada tampilan *device* atau *smartphone*, seperti pengaturan *layout*, penempatan *background*, penempatan *button* dan lain-lain. Setelah itu sistem akan memulai proses pembuatan deskriptor terhadap *marker* yang telah penulis siapkan dalam *database* aplikasi EclipseAR. Saat proses inialisasi berjalan, sistem akan mengolah *marker* pada *database* satu kali saja pada saat aplikasi sedang di *install*.

Dalam proses pembuatan deskriptor pertama citra atau gambar yang didapat akan diproses oleh algoritma SURF untuk mencari *keypoints* dari citra tersebut. Setelah didapatkan, proses selanjutnya adalah mencari deskriptor dari citra tersebut, deskriptor inilah yang dibutuhkan untuk proses pencocokan *marker* pada *database* dengan citra yang di *tracking* oleh ARCamera [9].

### 4. Tracking Marker

Pendeteksian gambar dan perhitungan posisi gambar dilakukan dengan menggunakan algoritma SURF, dimana aplikasi akan menjalankan ARCamera dari *smartphone* untuk mengambil gambar *marker*. Informasi posisi yang didapatkan nantinya akan digunakan untuk menempatkan objek atau model pada *marker*. Ada 4 tahapan dalam proses *tracking marker*, yaitu [10]:

- a. *Tracking marker* dari kamera *smartphone*.
- b. Ekstraksi *keypoints* menjadi deskriptor menggunakan SURF.
- c. *Corner detection*.
- d. Pencocokan pola.

Proses pembentukan deskriptor dilakukan dengan membuat kotak persegi disekitar *keypoint*. Sehingga akan dihasilkan area yang memiliki bentuk segi empat dan pola-pola citra yang sudah ditandai. Pola tersebut dibentuk untuk menghasilkan deskriptor yang siap diolah untuk pencocokan citra.

Deskriptor tersebut mengandung informasi terkait koordinat *keypoint* dan jumlah *keypoint* yang dihasilkan dari hasil komputasi. Tahap terakhir adalah pencocokan pola setelah area persegi dan pola gambar ditandai, citra yang berada di dalam persegi dianalisa dan dibandingkan polanya dengan sekumpulan pola yang telah ditentukan sebelumnya.

#### 5. Render Objek

Setelah mendapatkan hasil dari proses perhitungan menggunakan algoritma SURF, maka langkah selanjutnya adalah menampilkan hasil *augmented reality* berupa objek tiga dimensi [11]. Citra yang ditangkap oleh *ARCamera* dikatakan cocok ketika variabel *FoundState* yang diproses saat *tracking marker* memiliki status *true*. Setelah diketahui bahwa status *tracking marker* tersebut adalah *true* maka informasi mengenai objek tersebut akan ditampilkan berdasarkan variabel *index Proses* [12].

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Tampilan Layar

##### 1. Splash Screen

*Splash screen* adalah tampilan awal pada aplikasi yang menampilkan logo Unity kemudian disusul dengan tampilan *main menu* dan lain-lain seperti pada Gambar 10.



Gambar 10. Tampilan Layar *Splash Screen*

##### 2. Main Menu

Fungsi *button* pertama dan kedua adalah untuk beralih ke *scene* gerhana matahari atau *scene* gerhana bulan kemudian menampilkan objek tiga dimensi suara dan animasi jika *marker* yang digunakan cocok. *Quit button* berfungsi untuk menutup aplikasi. Adapun Main Menu bisa dilihat pada Gambar 3.2.



Gambar 11. Tampilan layar *Main Menu*

##### 3. Gerhana Matahari

Tampilan utama pada layar adalah area kamera AR yang nantinya digunakan untuk *tracking marker* menggunakan seluruh layar. *Scene* gerhana matahari menampilkan animasi tiga dimensi bumi, bulan, matahari, suara dan bagaimana proses terjadinya gerhana matahari dari awal sampai selesai. Pada tampilan ini terdapat dua *button* yaitu *play* dan *menu*, *play button* berfungsi untuk memulai animasi dan memunculkan suara sedangkan *menu button* berfungsi untuk kembali ke *menu* utama. Adapun gambar gerhana matahari bisa dilihat pada Gambar 3.3.



Gambar 12. Tampilan Layar Gerhana Matahari

#### 4. Gerhana Bulan

*Scene* gerhana bulan memiliki pola yang sama dengan *scene* gerhana matahari, tiga objek 3D dan dua *button*. Yang membedakan kedua *scene* ini adalah letak objek tiga dimensi dan *output* suara yang dihasilkan. Untuk Gerhana Bulan bisa dilihat pada Gambar 13.



Gambar 13. Tampilan Layar Gerhana Bulan

### 3.2 Algoritma SURF

#### 1. *Script* Algoritma SURF

Berikut adalah *script* algoritma *Speeded Up Robust Features* yang penulis gunakan pada pembuatan aplikasi *mobile learning* gerhana. *Script* ini terletak pada Vuforia dan berfungsi sebagai inisiasi awal *ARCamera*, menangkap pola *marker*, menentukan *status tracking*, eksekusi fungsi *tracking* ditemukan atau *tracking* tidak ditemukan.

##### a. Inisiasi Awal

*Script* dibawah berfungsi sebagai inisiasi awal saat *tracking* dijalankan kemudian mengambil pola *marker* yang ada pada sistem. Untuk *Script* algoritma SURF dilihat pada gambar 3.5.

```
protected virtual void Start()
{
    mTrackableBehaviour = GetComponent<TrackableBehaviour>();
    if (mTrackableBehaviour)
        mTrackableBehaviour.RegisterTrackableEventHandler(this);
}

protected virtual void OnDestroy()
{
    if (mTrackableBehaviour)
        mTrackableBehaviour.UnregisterTrackableEventHandler(this);
}
```

Gambar 14. *Script* Inisiasi Awal

##### b. *Status Tracking*

*Script* ini adalah langkah selanjutnya dari *script* inisiasi, setelah selesai menjalankan perintah pertama maka langkah selanjutnya adalah menentukan status *tracking*. Jika sistem menemukan *marker* yang berada pada *database* dan *marker* tersebut cocok maka fungsi *On Tracking Found* akan dijalankan, sebaliknya jika *marker* tidak ditemukan atau tidak cocok dengan *database* maka fungsi *On Tracking Lost* akan digunakan oleh sistem.

Untuk status *tracking* bisa dilihat pada Gambar 15.

```
public void OnTrackableStateChanged(
    TrackableBehaviour.Status previousStatus,
    TrackableBehaviour.Status newStatus)
{
    if (newStatus == TrackableBehaviour.Status.DETECTED ||
        newStatus == TrackableBehaviour.Status.TRACKED ||
        newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
    {
        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
        OnTrackingFound();
    }
    else if (previousStatus == TrackableBehaviour.Status.TRACKED &&
        newStatus == TrackableBehaviour.Status.NO_POSE)
    {
        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
        OnTrackingLost();
    }
    else
    {
        // For combo of previousStatus=UNKNOWN + newStatus=UNKNOWN|NOT_FOUND
        // Warning is starting, but tracking has not been lost or found yet
        // Call OnTrackingLost() to hide the augmentation.
        OnTrackingLost();
    }
}
```

Gambar 15. Status Tracking

c. *On Tracking Found*

Setelah proses status *tracking* selesai dan *marker* ditemukan maka sistem akan memberi nilai *true* pada tiga komponen yaitu *rendered*, *collider* dan *canvas*. Langkah ini bertujuan untuk memunculkan ketiga objek tersebut diatas *marker* dalam bentuk objek dua atau tiga dimensi pada saat *ARCamera* dijalankan dan *marker* ditemukan. Sedangkan untuk *on tracking found* bisa dilihat pada gambar 16.

```
protected virtual void OnTrackingFound()
{
    var rendererComponents = GetComponentsInChildren<Renderer>(true);
    var colliderComponents = GetComponentsInChildren<Collider>(true);
    var canvasComponents = GetComponentsInChildren<Canvas>(true);

    // Enable rendering:
    foreach (var component in rendererComponents)
        component.enabled = true;

    // Enable colliders:
    foreach (var component in colliderComponents)
        component.enabled = true;

    // Enable canvas:
    foreach (var component in canvasComponents)
        component.enabled = true;
}
```

Gambar 16. Script On Tracking Found

d. *On Tracking Lost*

Kebalikan dari fungsi *On Tracking Found*, fungsi *script* ini adalah jika pada saat *status tracking* sistem tidak menemukan *marker* yang cocok pada *database* atau *marker* belum ditempatkan dengan benar maka sistem akan menjalankan perintah ini. Dengan cara memberi nilai *false* pada ketiga komponen, fungsi tersebut membuat tampilan *ARCamera* tidak menampilkan objek apapun saat dijalankan. Untuk tampilan *on tracking lost* bisa dilihat pada Gambar 17.

```
protected virtual void OnTrackingLost()
{
    var rendererComponents = GetComponentsInChildren<Renderer>(true);
    var colliderComponents = GetComponentsInChildren<Collider>(true);
    var canvasComponents = GetComponentsInChildren<Canvas>(true);

    // Disable rendering:
    foreach (var component in rendererComponents)
        component.enabled = false;

    // Disable colliders:
    foreach (var component in colliderComponents)
        component.enabled = false;

    // Disable canvas:
    foreach (var component in canvasComponents)
        component.enabled = false;
}
```

Gambar 17. Script On Tracking Lost

#### 4. KESIMPULAN

Dari rumusan masalah yang telah ditentukan dalam penerapan algoritma SURF pendeteksi objek pada augmented reality berbasis android, maka dapat disimpulkan bahwa dalam membuat lima objek tiga dimensi menggunakan aplikasi Blender 3D sebagai penentuan implementasi, dalam proses gunakan pada pembuatan aplikasi *mobile learning* gerhana. *Script*



ini terletak pada Vuforia dan berfungsi sebagai inisiasi awal ARCamera. Pengujian kinerja algoritma SURF dalam kemampuan mendeteksi objek *Grayscale* berdasarkan cahaya lebih optimal, tingkat intensitas cahaya sangat mempengaruhi terhadap pendeteksi objek 3 dimensi oleh kamera.

Semakin tinggi tingkat intensitas cahaya, maka semakin banyak pula interest point yang tampil pada objek.

## 5. SARAN

Setelah mengevaluasi keseluruhan, beberapa saran yang dapat penulis berikan untuk pengembangan aplikasi ini kedepannya adalah sebagai berikut:

1. Jumlah objek tiga dimensi dalam aplikasi masih tergolong sedikit, diharapkan kedepannya agar objek tiga dimensi lainnya dapat ditambahkan agar informasi yang disampaikan jadi lebih banyak.
2. Penambahan fitur untuk menyalakan lampu *flash* pada kamera sehingga *marker* dapat terdeteksi oleh ARCamera dan mengembangkan *platform* agar dapat berjalan pada sistem operasi lainnya.

## DAFTAR PUSTAKA

- [1] Gallayanee Yaoyuneyong, Jamye K. Foster & Leisa R. Flynn 2014, *Factors Impacting The Efficacy of Augmented Reality Virtual Dressing Room Technology As a Tool for Online Visual Merchandising*, *Journal of Global Fashion Marketing*, 5:4, 283-296, DOI: 10.1080/20932685.2014.926129
- [2] Nielsen, J. and Molich, R. "Heuristic Evaluation of User Interfaces", *In Proceedings of ACM CHI'90 Conference on Human in Computing Systems*, pp.25-62.
- [3] Chang, K. E., Chang, C. T., Hou, H. T., Sung, Y. T., Chao, H. L., & Lee, C. M. 2014. *Development and Behavioral Pattern Analysis of A Mobile Guide System with Augmented Reality for Painting Appreciation Instruction in an art Museum*, *Computers & Education*, 71(1): 185-197.
- [4] Preece, J., Rodger, Y. & Sharp, H. "Interaction Design: Beyond Human-Computer Interaction". John Wiley & Sons, Inc.
- [5] Cheng, K. H., & Tsai, C. C 2014, *Children and Parents' Reading of An Augmented Reality Picture Book: Analyses of Behavioral Patterns and Cognitive Attainment*. *Computers & Education*, 72(C):.https://doi.org/10.1016/j.compedu.2013.12.003
- [6] Cai, S., Wang, X., & Chiang, F. K 2014, *A Case Study of Augmented Reality Simulation System Application in A Chemistry Course*, *Computers in Human Behavior*, 37(37): 31-40, https://doi.org/10.1016/j.chb.2014.04.018
- [7] Falahah, Iwan Rijayana, *Evaluasi Implementasi Sistem Informasi Dengan Pendekatan Utility System (Studi Kasus Sistem E-Campus Universitas Widyatama)*, *Program Studi Teknik Universitas Widyatama, Fakultas Teknik, Universitas Widyatama Jurnal Ilmiah Kursor*, Vol. 6, No. 2, Juli 2011 ISSN 0216 – 0544:1

- [8] Parlangei, O., Marchigiani, E. & Bagnara, S. “*Multimedia System in Distance Education: Effects of Usability on Learning*”, *Journal of Interacting with Computer*, Vol 12, Elsevier.
- [9] Chen, X., Wu, H., Li, X., Luo, X., dan Qiu, T 2012, *Real-time Visual Object Tracking via CamShift-Based Robust Framework*, *International Journal of Fuzzy Systems*, Vol 14, June 2012, No 2.
- [10] Rinci Kembang Hapsari, Nur Sulaiman, Luky Agus Hermanto 2016, *Aplikasi Findgo-ITATS Berbasis Android Dengan Algoritma SURF Untuk Menampilkan Informasi Lokasi Di ITATS*, *Integer Journal*, No 1, Vol 1.
- [11] Escobedo, L., Tentori, M., Quintana, E., Favela, J., & Garciarosas, D 2014, *Using Augmented Reality to Help Children with Autism Stay Focused*, *IEEE Pervasive Computing*, 13(1).<https://doi.org/10.1109/MPRV.2014.19>
- [12] Wang, Z., Yang, F 2012, *Object Tracking Algorithm Based on Camshift and Grey Prediction Model in Occlusions*, *The 2<sup>nd</sup> International Conference on Computer Application and System Modeling*.