

RANCANG BANGUN VIRTUAL GAMELAN MOBILE MENGGUNAKAN AUGMENTED REALITY

Tommi Suryanto¹⁾, Ema Utami²⁾, Hanif Al Fatta³⁾

^{1,2,3)} Magister Teknik Informatika STMIK AMIKOM Yogyakarta
email : tommisuryanto2000@gmail.com¹⁾, ema.u@amikom.ac.id²⁾, hanif.a@amikom.ac.id³⁾

Abstraksi

Gamelan merupakan alat musik tradisional yang harus tetap dilestarikan, semakin pesatnya perkembangan teknologi dan informasi, pada era globalisasi perkembangan gamelan diharapkan mampu menarik minat anak muda. Penelitian ini bertujuan untuk menganalisis faktor-faktor yang berpengaruh dalam pendeteksian interaksi dan bagaimana interaksi dapat dideteksi oleh sistem dan dengan adanya aplikasi ini anak muda yang memiliki media *smartphone* dapat memainkan aplikasi virtual gamelan ini dimana saja dengan media marker yang dapat langsung berinteraksi seolah-olah bermain gamelan. Prototype yang dibuat menggunakan *unity3D* dan *Vuforia*.

Kata Kunci :

Augmented Reality, Gamelan, Virtual

Pendahuluan

Augmented reality saat ini menjadi *trend* di industri *mobile*, yang memungkinkan penambahan data *eksternal* di atas masukan kamera. Perkembangan *augmented reality* saat ini sedang banyak diteliti di akademi, komputer *vision*, pengolahan citra, laboratorium *experience* dan industri oleh perusahaan. Banyak aplikasi *augmented reality* sekarang yang hanya menampilkan data ke penggunaannya tanpa berfokus untuk berinteraksi dengan penggunaannya[1].

Gamelan dalam era globalisasi yang sudah mulai terasa sejak akhir abad ke-20, telah membuat masyarakat dunia, termasuk bangsa Indonesia harus bersiap-siap menerima kenyataan masuknya pengaruh luar terhadap seluruh aspek kehidupan bangsa. Salah satu aspek yang terpengaruh adalah kebudayaan. Perkembangan gamelan Jawa pada era globalisasi terasa sangat mengawatirkan. Terlihat pada para pemain gamelan Jawa yang kebanyakan orang tua[2].

Teknologi virtual reality atau yang disebut dengan realitas maya adalah teknologi yang memungkinkan pengguna bisa berinteraksi terhadap objek nyata yang disimulasikan menggunakan komputer. Dengan melihat realita perkembangan teknologi tersebut maka dilakukan penelitian pembuatan *mobile virtual gamelan* menggunakan *Augmented Reality* dengan konsep interaktif sehingga nantinya menjadi sebuah perkembangan dalam dunia teknologi sekaligus juga dalam perkembangan alat

musik modern. Aplikasi ini dirancang berjalan pada *Android Mobile*.

Tinjauan Pustaka

Ewaldus Ambrosius Tukan pada tahun 2012 dalam jurnalnya "Penerapan *Augmented Reality* pada *Game Book*" membahas mengenai perancangan permainan ular tangga untuk anak-anak yang dimainkan oleh dua orang atau lebih. Dalam permainan ini sudah lebih terbentuk interaksi antara pemain dan marker. Penerapan teknologi *augmented reality* berjalan dengan baik, hanya saja belum sempurna dikarenakan jarak *marker* dengan kamera juga sangat berpengaruh [3]. Mochamad Fathoni, Eko Budi Cahyono, S.Kom, MT, Wahyu Andhyka Kusuma, S.Kom pada tahun 2011 dalam jurnalnya "Alat Musik Perkusi *Augmented Reality* Berbasis Android" membahas mengenai Perancangan alat musik perkusi dengan menggunakan *augmented reality* yang berjalan pada sistem operasi *android* hasil dari penelitian ini adalah alat musik perkusi yang dijalankan menggunakan *augmented reality* yang berjalan pada sistem operasi android dengan menggunakan sentuhan tangan untuk mengeluarkan suara [4]. Daniel Wagner, Lukas Gruber, Dieter Schmalstieg pada tahun 2011 dalam jurnalnya "*Augmented Reality on Mobile Phones*" artikel ini membahas berbagai macam teknik yang penting untuk membuat game dan lainnya, aplikasi grafis yang intensif yang akan dijalankan pada sejumlah besar ponsel saat ini digunakan untuk memberikan beberapa wawasan bagaimana pendekatan kinerja isu-isu sensitif dalam pengembangan *augmented*

reality terutama untuk pengembangan *augmented reality* pada ponsel, dan cara mendapatkan independensi *platform* yang memadai untuk memungkinkan diseminasi untuk menggambarkan kebutuhan khusus dan teknik pada aplikasi *augmented reality*, jurnal ini juga membahas masalah dan solusi ketika mengembangkan untuk *Symbian* dan *Windows Mobile*, *OpenGL ES* dan *Direct3D* tetapi tidak membahas mengenai segmentasi yang lebih dalam mengenai *software* yang digunakan [5].

Tobias Domhan pada tahun 2010 dalam jurnalnya “*Augmented Reality on Android Smartphones*” menunjukkan bahwa *smartphone Android* dapat digunakan untuk aplikasi *Augmented Reality*, meskipun mereka hanya memiliki kekuatan pemrosesan yang tersedia terbatas. Dalam konteks tulisan ini *Open Source Augmented Reality* kerangka kerjanya, berdasarkan perpustakaan ARTToolkit. Aplikasi ini mampu menampilkan model tiga dimensi pada *Augmented Reality*. Pada penelitian ini menghasilkan *Augmented Reality Model Viewer* [6].

Mas Ali Bahtiar pada tahun 2011 dalam jurnalnya “*Sistem Augmented Reality Untuk Animasi Games Menggunakan Camera Pada PC*” membahas mengenai sistem *Augmented Reality* untuk animasi games dengan menggunakan camera sebagai media untuk pembacaan simbol input dari animasi games 3D. Pada penelitian ini Ali Bahtiar melakukan perbandingan tingkat penggunaan artoolkit dan flartoolkit dalam perancangan *augmented reality* yang menunjang pemanfaatan *marker* serta melakukan analisis jarak dan pencahayaan yang digunakan untuk dapat meng-generate simbol *marker*. Penelitian ini terbatas hanya menampilkan *object augmented reality* tanpa ada interaksi [7].

Radhitya Wawan Yunarko, Imam Kuswardayan, Dwi Sunaryono pada tahun 2012 dalam jurnalnya “*Perancangan dan Pembuatan Aplikasi Kontrol Objek 3D Interaktif Melalui Sentuhan Tangan Berbasis Augmented Reality Dengan Library FLAR dan MotionTracker*” melakukan penelitian eksplorasi teknologi *augmented reality* di mana objek yang ditampilkan dapat berinteraksi dan merespons aksi pengguna. Respons objek ini dipicu oleh gerakan tangan dengan memanfaatkan teknologi pelacakan gerakan. Hasil Pendeteksian Tumbukan dan Penghasilan Respons sentuhan dapat berjalan, aplikasi telah berhasil melakukan pelacakan penanda dengan baik [8].

Gamelan Jawa menurut Panji Prasetyo pada tahun 2012 dalam jurnalnya “*Seni Gamelan Jawa sebagai Representasi dari Tradisi Kehidupan Manusia Jawa*” mempunyai dua sumber yaitu pertama gamelan jawa mempunyai latar belakang pra-historis atau sebelum kedatangan pedagang india ke

nusantara, yang nantinya akan menjadi sebuah mentalitas pendukung, atau yang akan ikut terbawa ke sejarah berikutnya dan kedua gamelan jawa yang sudah terbentuk secara utuh pada zaman Hindu – Buddha (kedatangan pedagang India di nusantara), yang budayanya menjadi sebuah inspirasi terbesar bagi masyarakat Jawa pada saat itu [9].

Azuma, R.T., et al. (2011) “*Indirect Augmented Reality*” mendefinisikan *augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif [10]. Paul Milgram dan Fumio. (1994) dalam jurnalnya “*Augmented Reality: A class of displays on the reality-virtuality continuum*” mereka menggambarkan sebuah kontinum yang membentang dari lingkungan nyata untuk untuk lingkungan *virtual* murni. Mereka menyimpulkan bahwa AR lebih dekat dengan dunia nyata dan *augmented virtuality* lebih dekat dengan dunia *virtual* [11].

Dalam penerapannya teknologi *augmented reality* memiliki beberapa komponen yang harus ada untuk mendukung kinerja dari proses pengolahan citra digital. Menurut R. Silva, J. (2005) dalam jurnalnya “*Introduction to Augmented Reality*” terdapat 4 komponen dalam *augmented reality*. Adapun komponen-komponen tersebut adalah sebagai berikut [12]:

1. *Scene Generator*
Scene Generator adalah komponen yang bertugas untuk melakukan *rendering* citra yang ditangkap oleh kamera. Objek virtual akan di tangkap kemudian diolah sehingga dapat kemudian objek tersebut dapat ditampilkan
2. *Tracking System*
Tracking system merupakan komponen yang terpenting dalam *Augmented reality*. Dalam proses tracking dilakukan sebuah pendeteksian pola objek virtual dengan objek nyata sehingga sinkron diantara keduanya dalam artian proyeksi virtual dengan proyeksi nyata harus sama atau mendekati sama sehingga mempengaruhi validitas hasil yang akan didapatkan.
3. *Display*
Dalam pembangunan sebuah sistem yang berbasis AR dimana sistem tersebut menggabungkan antara dunia virtual dan dunia nyata ada beberapa parameter mendasar yang perlu diperhatikan yaitu optik dan teknologi video. Keduanya mempunyai keterkaitan yang

tergantung pada faktor resolusi, fleksibilitas, titik pandang, *tracking area*. Ada batasan-batasan dalam pengembangan teknologi *Augmented reality* dalam hal proses menampilkan objek. Diantaranya adalah harus ada batasan pencahayaan, resolusi layar, dan perbedaan pencahayaan citra antara citra virtual dan nyata.

4. *AR Devices*

Ada beberapa tipe media yang dapat digunakan untuk menampilkan objek berbasis *Augmented reality* yaitu dengan menggunakan *optic*, sistem retina virtual, video penampil, monitor berbasis AR dan proyektor berbasis AR.

Vuforia qualcomm Menurut Michael R. Lyu. (2011) dalam jurnalnya “*Digital Interactive Game Interface Table Apps*” menjelaskan bahwa *vuforia qualcomm* merupakan *library* yang digunakan sebagai pendukung adanya *augmented reality* pada *Android*. *Vuforia* menganalisa gambar dengan menggunakan pendeteksi *marker* dan menghasilkan informasi 3D dari *marker* yang sudah dideteksi via API. Programmer juga dapat menggunakannya untuk membangun objek 3D virtual pada kamera. Pengembangan aplikasi menggunakan *platform Qualcomm AR*. *Platform* tersebut terdiri dari 2 komponen diantaranya adalah [13] :

1. *Target Management System*

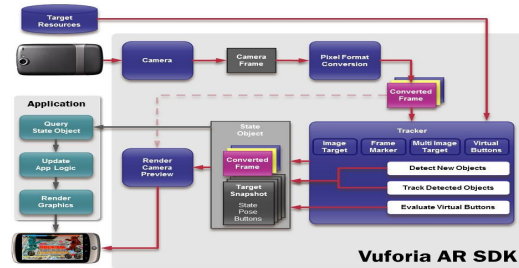
Mengijinkan pengembang melakukan *upload* gambar yang sudah diregistrasi oleh *marker* dan kemudian melakukan *download* target gambar yang akan dimunculkan.

2. *QCAR SDK Vuforia*

Mengijinkan pengembang untuk melakukan koneksi antara aplikasi yang sudah dibuat dengan *library static* i.e *libQCAR.a* pada *iOS* atau *libQCAR.so* pada *Android*

SDK *Vuforia* mendukung berbagai jenis target 2D dan 3D termasuk Target Gambar 'markerless', 3D Multi target konfigurasi, dan bentuk *Marker Frame*. Fitur tambahan dari SDK termasuk Deteksi Oklusi lokal menggunakan 'Tombol virtual', *runtime* pemilihan gambar *target*, dan kemampuan untuk membuat dan mengkonfigurasi ulang set pemrograman pada saat *runtime*. *Vuforia* menyediakan *Application Programming Interfaces* (API) di C++, Java, Objective-C. SDK mendukung pembangunan untuk *iOS* dan *Android* menggunakan *Vuforia* karena itu kompatibel dengan berbagai perangkat *mobile* termasuk *iPhone* (4/4S), *iPad*, dan ponsel *Android* dan *tablet* yang menjalankan *Android OS* versi 2.2 atau yang lebih besar dan prosesor *ARMv6* atau 7 dengan *FPU (Floating Point Unit)* kemampuan pengolahan. *Qualcomm Augmented Reality* memberikan beberapa keuntungan seperti [14] :

1. Teknologi *computer vision* untuk menyelaraskan gambar yang tercetak dan *object 3D*.
2. Mendukung beberapa alat development seperti *Eclipse*, *Android*, *Xcode*. Selain itu, *QCAR* juga menawarkan *development* dan distribusi yang gratis (*Qualcomm*, 2012).

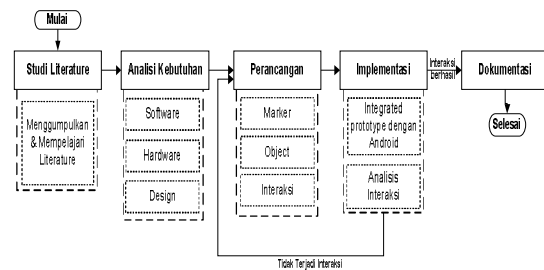


Gambar 1. DFD of the Vuforia SDK in an application environment

Sumber: <https://developer.vuforia.com/resources/devguide/vuforia-ar-architecture>

Metode Penelitian

Berikut adalah gambar diagram alir langkah-langkah penelitian yang akan dilakukan.



Gambar 2. Diagram Alir Penelitian

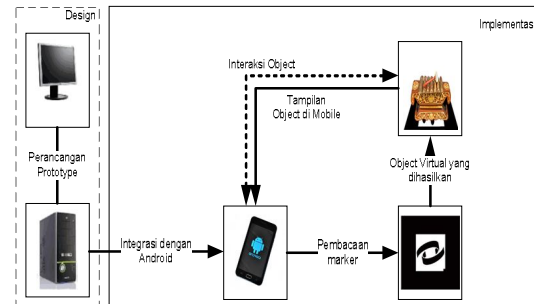
Berdasarkan gambar 2 diagram alir penelitian diatas terdapat beberapa langkah, berikut adalah penjelasan dari tiap-tiap langkah tersebut

1. Melakukan studi pustaka dengan mempelajari literatur-literatur terkait dengan penelitian ini dan untuk mencari solusi atas permasalahan yang dialami dalam melakukan penelitian.
2. Berdasarkan data yang telah dikumpulkan melalui observasi maupun melalui studi pustaka maka dilakukan analisis kebutuhan perancangan *virtual* gamelan. Analisis kebutuhan ini meliputi: bentuk *virtual* gamelan yang akan dibangun, suara di tiap-tiap bagian, iteraksi yang digunakan dalam virtual dan
3. kebutuhan perangkat-perangkat apa saja yang akan digunakan dalam membangun virtual gamelan menggunakan *augmented reality* berbasis *android* yang direncanakan.
4. Berdasarkan data yang telah dikumpulkan dan analisis kebutuhan yang telah dilakukan maka

- dilakukan perancangan object yang meliputi : perancangan marker, perancangan object, dan perancangan interaksi.
5. Perancangan *marker* dimana design *marker* di masukan kedalam sistem untuk nantinya digunakan sebagai media *virtual*
 6. Perancangan object dimana object yang akan ditempatkan diatas *marker* dibuat dan disatukan kedalam aplikasi yang nantinya dikombinasikan untuk dapat dipanggil dengan sistem dan dapat muncul pada bidang *marker*.
 7. Perancangan interaksi dimana design sistem yang dihasilkan ditambahkan koding agar dapat berinteraksi dengan sentuhan.
 8. Setelah itu masuk ketahap *integrated* ke perangkat Android mobile pada tahap ini penelitian akan melakukan itegerasi aplikasi keperangkat *android* untuk dapat dijalankan melalui media *mobile*.
 9. Dalam tahap ini akan dilakukan implementasi sentuhan terhadap desain *virtual* gamelan yang telah dibuat, tahap pengujian ini meliputi: melakukan uji *marker* untuk mengetahui apakah *marker* dapat dikenali oleh perangkat *mobile*, pengujian interaksi dengan *virtual* apakah suara dapat menghasilkan suara jika disentuh dan analisis setuhan terhadap object dan multi object pada *marker*. Apabila dalam tahap implementasi ditemukan kesalahan dalam melakukan konfigurasi maka akan dilakukan evaluasi atau perbaikan terhadap kesalahan-kesalahan konfigurasi tersebut.
 10. Tahap terakhir ialah tahap dokumentasi dilakukan untuk mendokumentasikan perancangan virtual gamelan augmented reality yang telah dilakukan dan membuat laporan hasil implementasi.

Hasil dan Pembahasan

Aplikasi mobile virtual gamelan yang dirancang dengan *augmented reality* ini dibangun dengan menggunakan bahasa pemrograman C#. Bahasa pemrograman C# ini dipilih karena bahasa ini cukup *powerfull* untuk mengembangkan aplikasi *mobile android*. Aplikasi yang digunakan untuk membangun *mobile virtual* gamelan ini adalah Unity 4 Version 4.2.1f4 Developer Preview dimana aplikasi ini mendukung untuk pembuatan Aplikasi *augmented reality* dan sudah mendukung *library vuforia* yang sudah terintegrasi dengan berbagai jenis *platform*. Seri *device Android* yang disarankan minimal mendukung proses *multi touch* karena mengingat proses kerja aplikasi ini harus mendukung sentuhan tangan secara *simultan*. Arsitektur aplikasi yang akan di bangun dapat dilihat pada Gambar 3.



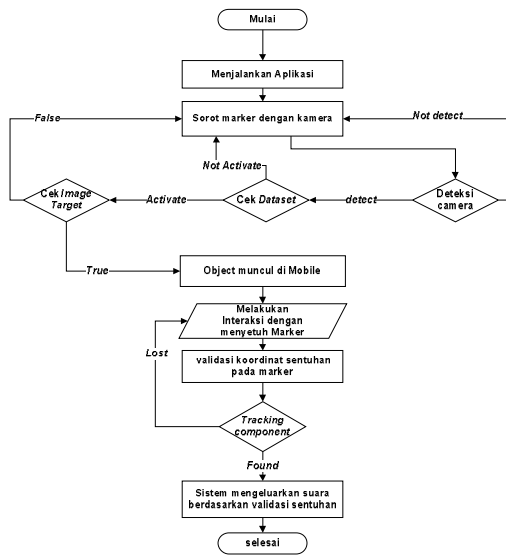
Gambar 3. Arsitektur Sistem

Mobile virtual gamelan ini diakses oleh satu *user* saja (*Single User*). Dimana *user* tersebut dapat menjalankan aplikasi ini dengan menggunakan media *marker* yang sudah ditentukan. Secara umum alur sistem aplikasi yang dibuat adalah sebagai berikut :

1. *User* membuka aplikasi menggunakan *android mobile* yang sudah terinstall aplikasi *mobile virtual* gamelan.
2. *User* Meletakkan *marker* yang sudah diregistrasi tepat di bawah kamera sehingga secara otomatis kamera *mobile* akan melacak *marker* yang sudah diregistrasi tersebut dan object gamelan akan muncul pada *device android* yang digunakan.
3. *User* menyentuh bagian *marker* yang diidentifikasi koordinatnya dimana kamera *device android* akan memvalidasi koordinat tersebut.
4. Sentuhan tangan pada koordinat *marker* yang telah ditentukan akan dideteksi sebagai sebuah sentuhan(interaksi) yang nantinya oleh sistem akan diolah datanya.
5. Koordinat sentuhan oleh sistem kemudian akan dicocokkan dengan spesifikasi suara yang didefinisikan dan suara akan dikeluarkan sebagai parameter keberhasilan pencocokkan dengan interaksi yang dilakukan.

User melakukan gerakan dinamis secara *simultan* pada koordinat *marker* tersebut sehingga menimbulkan irama musik gamelan yang dipukul. Pada gambar 4 dapat diketahui bahwa *user* memiliki peran sebagai pengontrol jalannya aplikasi *virtual gamelan augmented reality*. Dalam aplikasi ini hanya ada satu *actor* yaitu *user*. Karena sifat dari aplikasi ini *stand alone* dimana tidak ada interaksi *client-server* sehingga dalam proses menjalankannya hanya membutuhkan satu *user* saja. Dalam pendeteksian *marker* sekaligus object yang ada tersebut harus sudah didefinisikan dalam sistem pada tahap implementasi kode program. Karena letak koordinat tersebut bersifat tetap dan tidak dapat diubah-ubah sehingga dalam proses deteksi koordinat dan *event* yang ditimbulkan berdasarkan koordinat tersebut akan lebih mudah diatur.

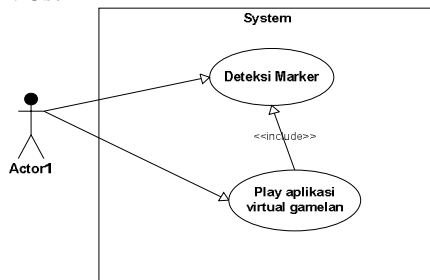
Langkah-langkah tersebut dapat digambarkan ke dalam sebuah *flow chart* seperti pada gambar dibawah ini.



Gambar 4. Flowchart Sistem

Merujuk pada *flowchart* sistem aplikasi *virtual gamelan augmented reality* yang sudah digambarkan sebelumnya maka dapat dibuat *use case diagram* berikut juga spesifikasi dari *use case diagram*.

Nama : aplikasi *virtual gamelan augmented reality*
Aktor : User



Gambar 5. Use Case Diagram prototype

Skenario *Use Case* mendeskripsikan urutan langkah-langkah dalam proses prototype, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor seperti tabel berikut :

Tabel 1. Use Case Deteksi Marker

User	Sistem
1.Membuka aplikasi	2.Tampil aplikasi
3.Meyorot marker dengan kamera	4.Deteksi kamera
	5.Jika tidak terdeteksi , maka kembali ke no3
	6.Jika terdeteksi, lanjut no.7
	7. cek data set
	8.Jika tidak aktive , maka kembali ke no3

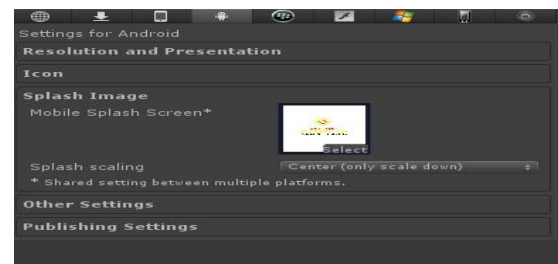
	9.Jika active, lanjut no10 10. cek image target 11.Jika false, maka kembali ke no 3 12.Jika true, maka muncul object gamelan
--	---

Tabel 2. Use Case Play Aplikasi Virtual Gamelan

User	Sistem
1.menyentuh marker	2.Deteksi sentuhan 3.Validasi koordinat marker yang disentuh 4. <i>Tracking component</i> 5.Jika <i>tracking lost</i> , maka kembali ke 1. 6.Jika <i>tracking found</i> , maka muncul suara sesuai koordinat yang disentuh.

Untuk mengembangkan aplikasi *augmented reality* pada unity harus melakukan *load library vuforia*. *library vuforia* unity dapat didownload di situs <https://ar.qualcomm.at> yang merupakan situs resmi *vuforia qualcomm*. Setelah file berhasil didownload langkah selanjutnya adalah melakukan instalasi *library*. Setelah melakukan instalasi *library vuforia* unity akan di *generate* pada direktori yang bersangkutan. Kemudian *library vuforia* unity tersebut di *import* ke folder Unity\Editor\Standard Packages agar Unity dapat menjalankan aplikasi berbasis *augmented reality*.

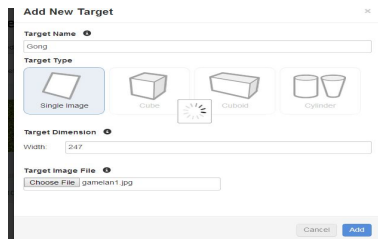
Tampilan *splash screen* di unity sudah disediakan fitur *upload image splash screen*, adapun caranya adalah dengan masuk ke *File - Build Setting - Player Setting* - pada inspector pilih *Splash Image*. Namun sebelumnya *image* yang dibuat harus dimasukkan ke dalam folder *Assets \ Editor \ QCAR \ Frame Marker Textures* seperti pada gambar 6.



Gambar 6. Mengubah *Splash Screen*

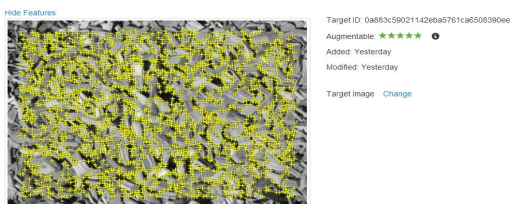
Menampilkan *marker* yang akan digunakan untuk memunculkan object maka *marker* harus ditracking terlebih dahulu agar *image* yang akan dijadikan marker tersebut teregistrasi di *engine unity*. Untuk melakukan *registrasi marker qualcomm* sudah menyediakan fitur registrasi *marker* yaitu dengan

cara *login* sebagai *user* jika belum terdaftar maka harus melakukan registrasi akun terlebih dahulu. Setelah melakukan *login* langkah selanjutnya adalah melakukan *upload image* yang digunakan sebagai *marker* dengan cara klik *My Trackables* dan tekan *New Project* untuk membuat *project* baru seperti pada gambar 7 dibawah ini.



Gambar 7. Form Upload Image

Kemudian mengisi nama *project* dan tekan *save*. Kemudian *upload file image* pada form *upload* yang telah disediakan. Kemudian secara otomatis sistem akan melacak *marker* yang telah diupload. Jika *image* sudah terdeteksi maka akan muncul keterangan seperti ditunjukkan pada gambar 8 di bawah ini.



Gambar 8. Hasil Registrasi Marker

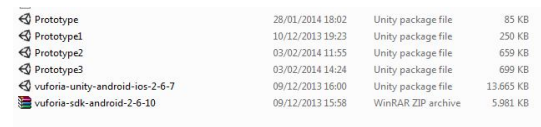
Pada gambar 8 ditunjukkan bahwa *marker* yang dideteksi memiliki banyak *node* sehingga dalam pelacakan objek 3D lebih mudah karena *node* yang dimunculkan pada gambar tersebut sangat banyak dan ditandai dengan bintang lima yang menandakan bahwa *tracking marker* sangat bagus. setelah melakukan registrasi *marker*. *Marker* yang telah diregistrasi, adapun *output* dari *marker* yang telah diregistrasi akan digenerate ke dalam bentuk *Unity Package*. Untuk melakukan *download asset marker* yang telah diregistrasi tekan “*download selected trackables*” dan pilih “*unity editor*” agar file *output marker* dapat dideteksi di unity seperti pada gambar 10.



Gambar 10. Download Marker

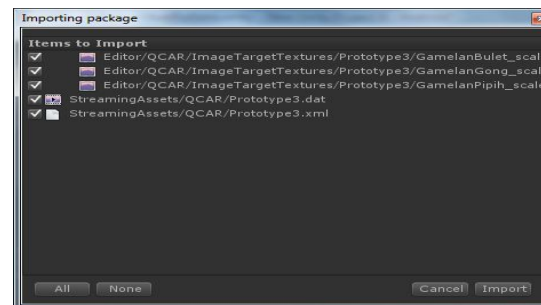
Export file marker yang telah didownload ke dalam *project unity* yang telah dibuat. Untuk melakukan

export file cukup menekan file tersebut dua kali dan secara otomatis objek *marker* yang sudah diregistrasi akan masuk ke dalam *project* di Unity seperti pada gambar 11 dibawah ini.



Gambar 11. File Output Marker yang Terregistrasi

Pada tahap ini *package* yang sudah di *download* dari *web* di masukan kedalam aplikasi dengan menggunakan perintah *asset - import package*, *import package* dapat dilihat pada gambar 12 dibawah ini.



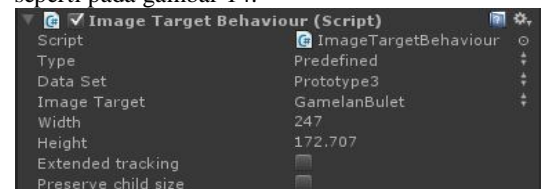
Gambar 12. Import Package

Setelah *package* diimportkan selanjutnya adalah memilih *data set* yang akan digunakan seperti gambar 13 *Data set* yang digunakan adalah *prototype 3*.



Gambar 13. Setting Data Set

Data set yang dipilih memiliki *image target* yang akan digunakan sebagai deteksi *marker* oleh sistem dimana *image target* yang akan dijalankan harus diaktifkan agar sistem dapat mengenali *marker* pada saat di *tracking* oleh kamera, pemilihan *image target* harus sesuai dan tidak boleh sama dengan *image target* yang lain dalam satu *data set* karena sistem tidak akan bisa melakukan pelacakan atau *tracking*, pemilihan *image target* dapat dilihat seperti pada gambar 14.



Gambar 14. Setting *Image Target*

Pada gambar 14 dilakukan pemilihan image target yang akan digunakan oleh sistem, adapun kode program agar tracking oleh sentuhan dapat dijalankan seperti gambar 15 dibawah ini.

```

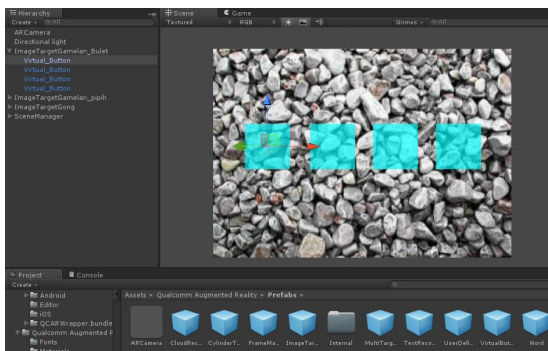
void Start()
{
    mTrackableBehaviour = GetComponent<TrackableBehaviour>();
    if (mTrackableBehaviour)
    {
        mTrackableBehaviour.RegisterTrackableEventHandler(this);
    }
}

public void OnTrackableStateChanged(
    TrackableBehaviour.Status previousStatus,
    TrackableBehaviour.Status newStatus)
{
    if (newStatus == TrackableBehaviour.Status.DETECTED ||
        newStatus == TrackableBehaviour.Status.TRACKED ||
        newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
    {
        OnTrackingFound();
    }
    else
    {
        OnTrackingLost();
    }
}

```

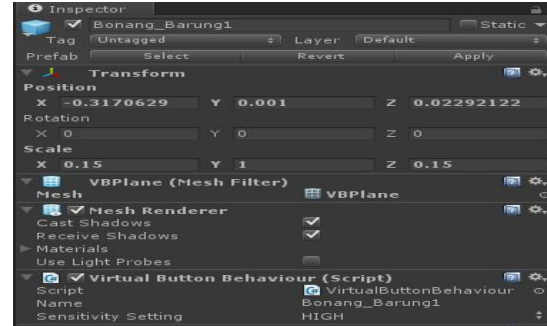
Gambar 15. *Traking Script*

Library *vuforia* yang telah menyediakan *prefab virtual button*. Menampilkan virtual button yang ada di *prefabs* ke dalam *ImageTarget* seperti pada gambar 16.



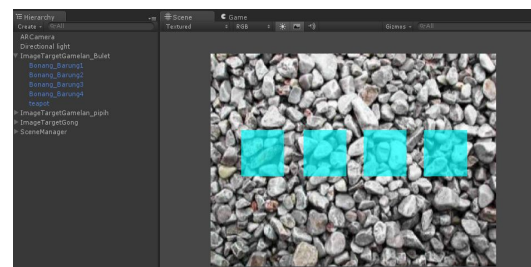
Gambar 16. Memasukkan *Virtual Button* ke dalam *ImageTarget*

Setting ukuran dan koordinat sehingga terlihat seperti pada gambar 17 Untuk mengatur ukuran dapat diatur di inspector.



Gambar 17. *Set Name Virtual Button*

Mengubah nama dari *virtual button* agar mudah untuk diingat. Dalam pembuatan aplikasi ini dibuat empat *virtual button* yang digunakan untuk mengeluarkan efek suara ketika *virtual button* tersebut disentuh.

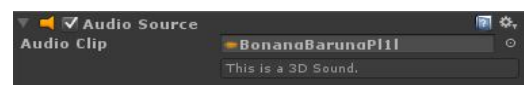


Gambar 18. *Virtual Button Marker*

Pada gambar 18 dapat dilihat terdapat empat virtual button yang digunakan pada image target masing-masing virtual button akan memiliki satu event handler sebagai triger jika pada koordinat virtual button disentuh atau ditutup.

Event yang terjadi jika pada koordinat virtual button ditutup adalah aplikasi akan mentrigger ke dalam fungsi virtual button yang mana akan memicu event atau reaksi sebagai mana telah dibuat dalam pengaturan script pada *virtualbuttoneventhendler*.

File suara ke dalam *virtual button* hampir sama seperti memasukkan objek *virtual button* ke dalam *ImageTarget*. File suara disimpan ke dalam satu folder (nama folder tidak ditentukan/bebas) dalam penelitian ini dibuat sebuah folder untuk menyimpan file-file suara yang nantinya akan dipanggil di *virtual button*. Dibuat sebuah folder dengan nama "*sound*" yang terdapat di direktori *\Assets\sound*. Setelah folder terbentuk langkah selanjutnya memasukkan file-file suara ke dalam folder tersebut. Kemudian drug file suara tersebut ke masing *virtual button* yang telah dibuat seperti pada gambar 19.

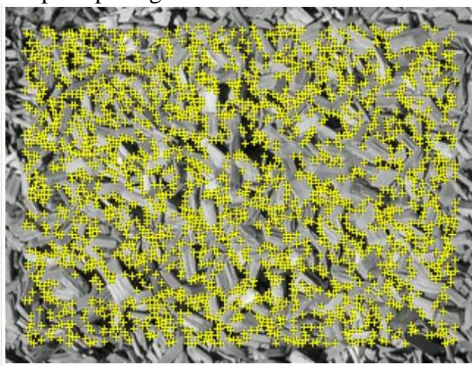


Gambar 19. Memasukkan File Suara ke dalam *Virtual Button*

Hal yang perlu diingat dalam implementasi file suara ke dalam *virtual button* adalah mengatur 3D *Sound Setting* yang terdapat pada inspector dengan tidak mencentang *Bypass Effects* dan *Play On Awake* kemudian memposisikan *Pan Level* pada posisi nol agar suara dapat terdengar.

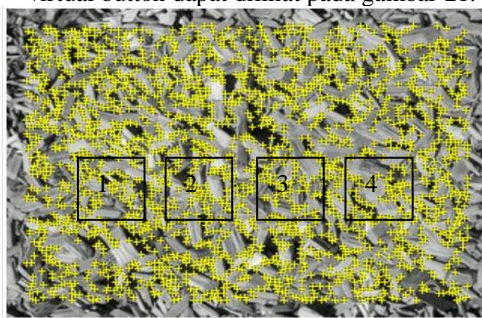
Konsep kerja dari interaksi yang dibuat dalam virtual button yang dipasang pada koordinat yang telah ditentukan pada marker bertujuan untuk mendapatkan interaksi sesuai dengan yang diinginkan. Prinsip kerja dari virtual button;

1. Marker yang telah diregistrasi kedalam vuforia akan memiliki image tunes/ node yang mana akan menjadi parameter pelacakan oleh kamera seperti pada gambar 20.



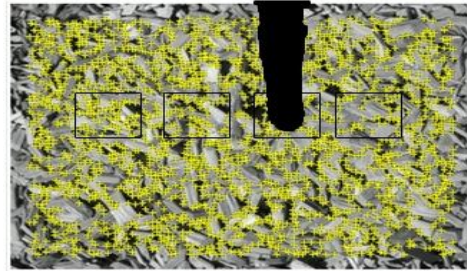
Gambar 20. Imagetunes/Node pada Marker

2. Virtual button yang dipasang di atas marker akan berada diatas node yang menjadi parameter pelacakan, peletakan koordinat virtual button dapat dilihat pada gambar 21.



Gambar 21. Koordinat Virtual Button pada Marker

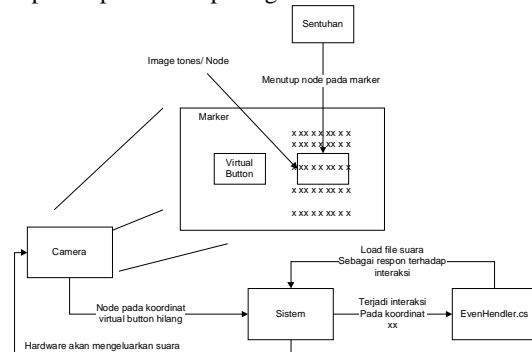
3. Node pada koordinat virtual button akan bereraksi jika node tersebut ditutup dalam hal ini sistem akan mengartikan node yang hilang pada koordinat virtual button sebagai interaksi/sentuhan dengan melakukan pengecekan pada action handler yang ada, dalam hal ini melakukan reaksi dengan mengeluarkan suara sebagai akibat dari tertutupnya node-node pada koordinat virtual button. Proses pelacakan dapat dilihat pada gambar 22.



Gambar 22. Tampilan Interaksi Pada Kamera

4. Node yang terletak diluar dari koordinat virtual button tidak akan bereaksi jika ditutup.

Pada event handler telah dipasang respon dari interaksi yang dilakukan dalam hal ini node yang hilang pada koordinat virtual button. Adapun skema dari sentuhan sampai dengan suara keluar sebagai respon dapat dilihat pada gambar 23 dibawah ini



Gambar 23. Skema sentuhan sampai dengan suara keluar sebagai respon

Pengujian pada aplikasi virtual gamelan ini meliputi pengujian jarak, pencahayaan, sudut pelacakan dan respon time sentuhan.

Tabel 3. Tabel Pengujian Jarak

No	Jarak (cm)	Keterangan
1	0	Gagal
2	10	Berhasil
3	20	Berhasil
4	30	Berhasil
5	40	Berhasil
6	50	Berhasil
7	60	Berhasil
8	70	Berhasil
9	80	Berhasil
10	90	Berhasil
11	100	Gagal
12	110	Gagal

Dari pengujian jarak pada tabel 3 yang dilakukan dapat diketahui aplikasi virtual gamelan ini dapat digunakan pada rentang jarak antara 10 sampai dengan 90 cm, lebih dari itu kamera tidak

dapat lagi melakukan deteksi marker sehingga object dan interaksi tidak dapat berjalan.

Tabel 4. Tabel Pengujian Pencahayaan

No	Intensitas Cahaya (Cd)	Keterangan
1	0	Gagal
2	1	Gagal
3	2	Berhasil
4	3	Berhasil
5	4	Berhasil
6	5	Berhasil
7	6	Berhasil
8	7	Berhasil
9	8	Berhasil
10	9	Berhasil
11	10	Berhasil
12	11	Berhasil
13	12	Gagal
14	13	Gagal
15	14	Gagal

Dari pengujian cahaya pada tabel 4 yang dilakukan pada aplikasi ini dapat diketahui intensitas cahaya yang dapat diberikan pada penggunaan aplikasi ini berada diantara 2 sampai dengan 11 caldela, lebih dari itu kamera tidak dapat melakukan pelacakan marker karena marker yang terlalu gelap dan image marker terbias oleh cahaya jika terlalu terang.

Tabel 5. Tabel Pengujian Sudut dengan sumbu x

No	Sudut (%)	Keterangan
1	0	Gagal
2	10	Gagal
3	15	Gagal
4	20	Berhasil
5	30	Berhasil
6	45	Berhasil
7	70	Berhasil
8	80	Berhasil
9	90	Berhasil
10	110	Berhasil
11	155	Berhasil
12	165	Gagal
13	170	Gagal
14	180	Gagal

Tabel 6. Tabel Pengujian Sudut dengan sumbu y

No	Sudut (%)	Keterangan
1	0	Gagal
2	10	Gagal
3	15	Gagal
4	20	Berhasil
5	30	Berhasil

6	45	Berhasil
7	70	Berhasil
8	80	Berhasil
9	90	Berhasil
10	110	Berhasil
11	155	Berhasil
12	165	Gagal

Dari pengujian sudut diketahui dalam penggunaannya aplikasi virtual gamelan ini dapat digunakan pada sudut antara 20⁰ sampai 155⁰ pada penggunaan pada sumbu x dan y.

Tabel 7. Tabel Pengujian Respon time sentuhan

Perangkat Jarak	20 cm	30 cm	40 cm	50 cm	60 cm	70 cm	80 cm	90 cm
Advan Tablet	0,5	0,5	0,7	0,7	0,9	1,1	1,1	1,1
Lenovo	0,6	0,6	0,7	0,8	1	1,1	1,3	1,3

Dari pengujian respon time sentuhan yang dilakukan dengan dua hardware yang berbeda dapat diketahui rata-rata respon time yang diperlukan oleh sistem untuk mengolah koordinat sentuhan sampai dengan mengeluarkan suara adalah 0,9 detik dan semakin jauh jarak marker dari hardware maka akan semakin lama respon time yang dibutuhkan.

Kesimpulan dan Saran

Berdasarkan hasil analisa dan pengujian pada penelitian *mobile virtual gamelan*, maka dapat diambil kesimpulan sebagai berikut :

Pendeteksian *marker* pada aplikasi *mobile virtual gamelan* telah berjalan dengan baik. *Objek* 3D gamelan dapat muncul pada *mobile* dan dapat divalidasi sesuai dengan marker dan virtual button berhasil mengeluarkan suara ketika tangan diletakkan tepat di koordinat bidang virtual button yang bersangkutan dan mengeluarkan suara yang berbeda-beda berdasarkan koordinat bidang virtual button yang telah ditentukan.

1. Pemilihan jarak dan pencahayaan sangat berpengaruh pada aplikasi ini, pemilihan jarak tidak terlalu dekat dan tidak terlalu jauh yaitu antara 10 sampai 90 cm serta pemilihan pencahayaan harus sesuai yaitu antara 2 sampai dengan 10 cd sangat baik untuk aplikasi ini.
2. Penggunaan virtual button dapat berjalan lancar dan aplikasi dapat mendeteksi tiga koordinat sentuhan langsung tanpa melakukan kesalahan dan suara yang dihasilkan sesuai dengan koordinat sentuhan yang disentuh.
3. Pemanfaatan teknologi *Augmented Reality* pada aplikasi ini dapat berjalan sesuai dengan perancangan, yaitu dapat menggabungkan objek *virtual* dengan lingkungan nyata secara *real-time*.

Dalam pengembangan aplikasi *mobile virtual gamelan* masih banyak terdapat kekurangan, maka dari itu perlu banyak pembenahan yang harus dilakukan pada penelitian selanjutnya. Beberapa hal yang disarankan yaitu :

1. Penambahan tipe gamelan yang dapat dijalankan dengan aplikasi ini.
2. Penambahan fitur record pada aplikasi ini.
3. Penambahan efek perubahan objek 3D ketika suara muncul.

Daftar Pustaka

- [1] Inger, Yaron. 2012. Real-time Image Blending for Augmented Reality on Mobile Phones. University of Jerusalem
- [2] Andewi, I. 2012. Perkembangan Kebudayaan Gamelan Jawa Di Tengah Arus Globalisasi. IKIP PGRI Madiun.
- [3] Tukan, Ewaldus Ambrosius. 2012 Penerapan *Augmented Reality* Pada Game Book. Yogyakarta : STM IK AMIKOM
- [4] Fathoni, Mochamad. Cahyono, Eko Budi, S.Kom, MT. Kusuma, Wahyu Andhyka, S.Kom. 2011. Alat Musik Perkusi *Augmented Reality* Berbasis Android. Malang : Universitas Muhammadiyah Malang
- [5] Wagner, Daniel. Gruber, Lukas. Schmalstieg, Dieter. 2011. *Augmented Reality on Mobile Phones*. Graz University of Technology
- [6] Domhan, Tobias. 2010. Augmented Reality on Android Smartphones. Stuttgart
- [7] Bahtiar, Mas Ali. 2011. Sistem *Augmented Reality* Untuk Animasi Games Menggunakan Camera Pada PC. Institut Teknologi Sepuluh Nopember
- [8] Yunarko., et al. 2012. Perancangan dan Pembuatan Aplikasi Kontrol Objek 3D Interaktif Melalui Sentuhan Tangan Berbasis *Augmented Reality* Dengan *Library FLAR* dan *MotionTracker*. Institut Teknologi Sepuluh Nopember
- [9] Prasetyo,P. 2012. Seni Gamelan Jawa sebagai Representasi dari Tradisi Kehidupan Manusia Jawa. Jakarta : Universitas Indonesia
- [10] Azuma, R.T., et al. 2011. *Indirect Augmented Reality*. United State : Nokia Research Center Hollywood.
- [11] Milgram, Paul., et al. 1994. *Augmented Reality: A class of displays on the reality-virtuality continuum*. Japan : ATR Communication Systems Research Laboratories
- [12] Sylva, R., et al. 2005. *Introduction to Augmented Reality*. Brazil : National Laboratory of Scientific Computation.
- [13] R.Lyu, Michael. 2012. *Digital Interactive Game Interface Table Apps*. Hongkong : Chinese University of Hongkong

Biodata Penulis

Tommi Suryanto, memperoleh gelar Sarjana Teknik Informatika (S.Kom), STM IK AMIKOM lulus tahun 2012. Saat ini sebagai Mahasiswa Magister Teknik Informatika STM IK AMIKOM Yogyakarta.

Em a Utami, memperoleh gelar Sarjana Sains (S.Si), Program Studi Ilmu Komputer FMIPA UGM, lulus tahun 1997. Tahun 2002 memperoleh gelar Magister Komputer (M.Kom) dari Program Ilmu Komputer UGM. Program Doktor pada Ilmu Komputer UGM, lulus tahun 2010. Saat ini sebagai Staf Pengajar program Magister Teknik Informatika STM IK AMIKOM Yogyakarta.

Hanif Al Fatta, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Ilmu Komputer FMIPA UGM, lulus tahun 2002. Tahun 2007 memperoleh gelar Magister Komputer (M.Kom) dari Program Ilmu Komputer UGM. Saat ini sebagai Staf Pengajar program Magister Teknik Informatika STM IK AMIKOM Yogyakarta.