

TESIS

VIRTUAL PIANO BERBASIS AUGMENTED REALITY

VIRTUAL PIANO BASED AUGMENTED REALITY

WAHYUDDIN



**PROGRAM PASCASARJANA
STMIK HANDAYANI MAKASSAR
2017**

VIRTUAL PIANO BERBASIS AUGMENTED REALITY

TESIS

Sebagai salah satu syarat untuk mencapai gelar magister komputer
Program Studi Sistem Komputer

Disusun dan diajukan oleh

WAHYUDDIN

**PROGRAM STUDI SISTEM KOMPUTER
PASCASARJANA STMIK HANDAYANI MAKASSAR
2017**

TESIS

VIRTUAL PIANO BERBASIS AUGMENTED REALITY

Disusun dan diajukan oleh:

WAHYUDDIN
NOMOR POKOK : 2015130024


Telah dipertahankan didepan panitia ujian tesis

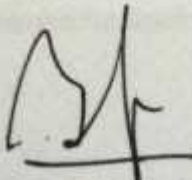
Pada tanggal 30 agustus 2017

Dan dinyatakan telah memenuhi syarat

Menyetujui

Komisi Penasihat,


Dr. Ir. Zahir Zainuddin, M.Sc.
Ketua


Prof. Dr. Ir. Syafruddin Syarif, M.T.
Anggota

Ketua Program Studi PPs
Sistem Komputer

Direktur Program Pascasarjana
STMIK Handyani


Dr. Ir. Zahir Zainuddin, M.Sc.


Dr. H. Moh. Alifuddin, MM., M. Kom.

PERNYATAAN KEASLIAN TESIS

Yang Bertanda Tangan Dibawah ini:

NAMA : WAHYUDDIN
NIM : 2015130024
PROGRAM STUDI : SISTEM KOMPUTER

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar merupakan hasil karya saya, bukan pengambilalihan tulisan atau pemikiran orang lain, adapun kutipan atau rujukan sebagai sumber yang saya gunakan dari penulisan orang lain, telah saya sebutkan pada daftar pustaka tesis ini. Apabila dikemudian hari terbukti dan telah memiliki kekuatan hukum yang sah dari lembaga yang berwenang bahwa tesis ini adalah hasil karya orang lain maka saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Makassar, Januari 2018

Yang Menyatakan,

WAHYUDDIN

ABSTRAK

WAHYUDDIN. virtual piano berbasis augmented reality, dibimbing oleh Zahir Zainuddin dan Syafruddin Syarif.

Musik merupakan nada atau suara yang disusun sedemikian rupa sehingga mengandung irama, lagu dan keharmonisan (terutama yang menggunakan alat-alat yang dapat menghasilkan bunyi-bunyi itu). Dalam bidang musik, banyak cara yang dapat dilakukan untuk belajar teori tentang musik. Dalam hal ini, terkhusus pada teori musik piano. Yang menjadi masalah, bagaimana membuat metode yang lebih menyenangkan dalam proses pembelajaran teori musik sehingga tidak terkesan monoton dan membosankan.

Sampai saat ini, untuk mempelajari alat musik piano paling tidak harus membutuhkan alat musik piano. Sedangkan, tidak semua orang dapat membeli alat musik tersebut. Dengan perkembangan teknologi yang sangat pesat, maka dilakukan penelitian untuk membuat virtual piano menggunakan augmented reality yang dijalankan pada android sehingga menjadi sebuah perkembangan dalam dunia teknologi dan sebagai perkembangan alat musik modern.

Dengan adanya aplikasi ini, diharapkan agar user dapat mempelajari alat musik piano dengan lebih menyenangkan dan mudah. Adapun konsep dari aplikasi ini adalah menjalankan aplikasi pada android dengan pendeteksian interaksi sentuhan tangan terhadap marker sehingga ketika marker tersebut disentuh maka akan mengeluarkan suara / nada.

Dari hasil pengujian Tingkat keberhasilan deteksi marker pada jarak 20-60 cm dan posisi kemiringan tingkat keberhasilannya pada 40-90⁰. Tingkat kegagalan deteksi marker pada jarak 65-70 cm dan posisi kemiringan tingkat kegagalannya pada 30-35⁰

Kata Kunci : Augmented Reality, Aplikasi, Android, Piano, Virtual Button

ABSTRACT

WAHYUDDIN. Virtual piano based augmented reality, guided by Zahir Zainuddin and Syafruddin Syarif.

Music is tone or sound which arranged in such a way thus contains rhythm, song, and harmony (mostly those use instruments that is able to produce the sounds). In music domain, there are many ways to learn about the music theory. In this case, particularly about the music theory of piano. The problem is how to create the more enjoyable method in a process of learning the music in order that it is not seemed monotonous or bored.

Until now, every person who want to learn playing piano at least have the piano itself. Meanwhile, not all people can buy it. By the rapid development of technology, then the research is established to create becomes a development in technology and also in modern music instrument.

By the existence of this application, it is hoped that the user can learn the piano more enjoyable and easier. The concept of this application is operating the application on android by sensing / detecting the interaction of hand-touched toward marker. When the marker is touched, it will produce sounds / tones.

From the test results The success rate of marker detection at a distance of 20-60 cm and the position of the slope of success rate at 40-90⁰. The failure rate of marker detection at a distance of 65-70 cm and its slope failure rate at 30-35⁰

Keywords : Augmented Reality, Application, Android, Piano, Virtual Button

KATA PENGANTAR

Alhamdulillah, segala puji syukur terlimpah kepada Tuhan kita, Allah SWT yang atas rahmat-Nya penulis dapat menyelesaikan tugas akhir ini dengan judul **“VIRTUAL PIANO BERBASIS AUGMENTED REALITY”**. penyusunan tugas akhir ini sebagai salah satu syarat guna menyelesaikan Studi pada STIMIK HANDAYANI MAKASSAR. Tak lupa shalawat serta salam selalu tercurah untuk Nabi kita, Muhammad SAW semoga kita mendapatkan syafa'atnya kelak.

Banyak kendala yang dihadapi penulis dalam melakukan penelitian dan penyusunan tesis ini, tetapi karena dukungan dan bimbingan dari semua pihak, maka tesis ini dapat disesuaikan tepat pada waktunya. Oleh karena itu, pada kesempatan ini penulis menyampaikan rasa terima kasih dan penghargaan yang tulus dan setinggi-tinggi kepada yang terhormat :

1. Teristimewa kepada kedua Orang Tua, Saudara dan keluarga yang selalu mendoakan juga dorongan moral dan bantuan materi hingga penulis dapat menyelesaikan tesis ini.
2. Penasehat utama Dr.Ir.Zahir Zainuddin, M.Sc. Atas bimbingan, arahan dan waktu yang telah diluangkan kepada penulis untuk memberikan masukan selama proses penyusunan tesis ini.

3. Penasehat kedua Prof.Dr.Ir.Syafruddin Syarif, M.T. Atas bimbingan, arahan dan waktu yang telah diluangkan kepada penulis untuk memberikan masukan selama proses penyusunan tesis ini.
4. Bapak Dr.Eng.Armin Lawi, M.Eng. Selaku ketua Program Pascasarjana STIMIK Handayani Makassar.
5. Ketua Prodi Pascasarjana Sistem Komputer Dr. Ir. Zahir Zainuddin, M.Sc.
6. Seluruh teman-teman angkatan 2015 Program Pascasarjana atas kerjasamanya selama ini.

Penulis menyadari bahwa apa yang telah dibahas dalam tesis ini masih jauh dari kesempurnaan oleh karena itu terbatasnya kemampuan, pengetahuan, dan pengalaman yang dimiliki sehingga kritik dan saran yang sifatnya membangun sangat kami butuhkan.

Akhir kata penulis mendoakan semoga semua pihak yang telah membantu mendapatkan balasan yang setimpal dari Allah SWT, dan semoga tesis ini dapat bermanfaat bagi semua pihak.

Makassar, Agustus 2017

Wahyuddin

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN TESIS	ii
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	iv
ABSTRAK	v
ABSTRACK.....	vi
KATA PENGANTAR	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Tujuan Penelitian	2
D. Manfaat Penelitian	3
E. Batasan Masalah	3
BAB II TINJUAN PUSTAKA	5
A. Definisi <i>Augmented Reality</i>	5
B. Marker	7
C. Natural Features Dan Rating.....	7
D. Android.....	8
E. Virtual.....	9

F. Piano.....	10
G. Bahasa C#	10
H. UML.....	11
I. Tahapan penelitian.....	25
J. Metode pengujian.....	26
K. Roadmap Penelitian	30
L. Kerangka Pikir Penelitian	33
BAB III METODOLOGI PENELITIAN.....	34
A. Lokasi Dan Waktu Penelitian	34
B. Tahapan Penelitian	34
C. Rancangan Sistem.....	36
D. Perangkat Pengembangan Sistem.....	37
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	39
A. Rancangan sistem yang diusulkan.....	39
B. Rancangan Output.....	44
C. Rancangan Input.....	45
D. Implementasi dan pengujian sistem	46
BAB V PENUTUP	56
A. Kesimpulan	56
B. Saran.....	56
DAFTAR PUSTAKA.....	58
LAMPIRAN.....	60

DAFTAR GAMBAR

Gambar 1. Lembar Kerja Unity	6
Gambar 2. Marker Piano.....	7
Gambar 3. Kerangka Pikir Penelitian	33
Gambar 4. Rancangan Sistem.....	36
Gambar 5 <i>Use Case Diagram</i>	40
Gambar 6 <i>Activity Diagram</i>	42
Gambar 7 <i>Sequence Diagram</i>	43
Gambar 8 <i>Class Diagram</i>	45
Gambar 9 Aplikasi dijalankan.....	46
Gambar 10 <i>Form Inspector</i>	46
Gambar 11 FlowChart Aplikasi	47
Gambar 12 FlowGraph Aplikasi	47
Gambar 13 FlowChart Pengaturan	49
Gambar 14 FlowGraph Pengaturan	50

DAFTAR TABEL

Tabel 1. Diagram <i>Use Case</i>	12
Tabel 2. Diagram Kelas	16
Tabel 3. Diagram Objek	18
Tabel 4. Diagram Status	19
Tabel 5. Diagram Aktivitas / Activity	20
Tabel 6. Diagram Komponen	22
Tabel 7. Diagram Deployment	24
Tabel 8. <i>Use Case</i> Deteksi Marker	40
Tabel 9. <i>Use Case</i> Mainkan Aplikasi Piano	41
Tabel 10. Grafik Matriks.....	49
Tabel 11. Grafik Matriks.....	52
Tabel 12. Pengujian Aplikasi.....	53
Tabel 13. Pengujian Akurasi	54
Tabel 14. Hasil Pengukuran Marker.....	54

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Dalam bidang komputer, terdapat teknologi yang disebut *Augmented Reality* atau yang biasa disingkat AR. Pada dasarnya AR memiliki kelebihan yaitu mampu memberikan pengalaman dan pemahaman yang mendalam bagi subjek pembelajaran.

Sampai saat ini untuk mendalami musik piano ini paling tidak harus memiliki alat tersebut. Seiring dengan kemajuan teknologi hal ini dirasa kurang efisien karena mengingat pada saat ini teknologi berkembang sangat pesat. Dengan melihat realita perkembangan teknologi tersebut maka dilakukan penelitian pembuatan alat musik piano berbasis *Augmented Reality* sehingga nantinya menjadi sebuah perkembangan dalam dunia teknologi sekaligus juga dalam perkembangan alat musik modern. Adapun konsep dari aplikasi ini adalah pendeteksian interaksi sentuhan tangan pada *marker* sehingga ketika *marker* tersebut tersentuh tangan maka akan menimbulkan bunyi. Aplikasi ini berjalan pada *Android Mobile*.

Musik merupakan sesuatu yang sudah tidak asing lagi bagi kita. Hampir setiap hari kita mendengarkan musik, baik dari televisi, komputer, handphone, suara seseorang atau dari media-media lainnya. Musik adalah sebuah fenomena yang sangat unik yang bisa dihasilkan oleh

beberapa alat musik. Musik juga bisa berperan sebagai salah satu media ungkapan kesenian, musik mencerminkan kebudayaan masyarakat pendukungnya. Di dalam musik terkandung nilai dan norma-norma yang menjadi bagian dari proses enkulturasi budaya, baik dalam bentuk formal maupun informal.

Pada penelitian sebelumnya ada beberapa penelitian yang dilakukan dengan menggunakan augmented reality seperti:

1. **'Alat Musik Perkusi Augmented Reality Berbasis Android'** pada penelitian ini memiliki kekurangan pada saat efek perubahan objek 3D ketika suara muncul, dan juga suara nada yang dikeluarkan hanya 4 jenis nada.
2. **'sebuah aplikasi pengenalan dan simulasi alat musik tradisional Betawi menggunakan teknologi Augmented Reality'** pada penelitian ini memiliki kekurangan yaitu hanya menampilkan jenis-jenis alat tradisional betawi dalam bentuk 3D.

Sedangkan penelitian yang saya angkat ini adalah virtual piano berbasis augmented reality dimana penelitian ini memiliki kelebihan dari beberapa penelitian yang ada diatas yaitu virtual piano ini kelebihanannya adalah

1. Memiliki nada yang lengkap yaitu nada rendah sampai nada paling tertinggi.
2. Suara yang dihasilkan sesuai dengan nada piano yang sesungguhnya.

B. RUMUSAN MASALAH

Setelah melihat permasalahan yang terdapat pada latar belakang di atas, maka penulis merumuskan :

1. Bagaimana membuat Virtual Piano sebagai sarana yang menarik dan saat dimainkan seperti piano yang sesungguhnya.
2. Bagaimana memanfaatkan teknologi *augmented reality* bermanfaat dibidang musik.

C. TUJUAN PENELITIAN

Adapun tujuan dari penelitian ini adalah membangun aplikasi yang berfungsi sebagai media dan metode dalam proses pembelajaran alat musik piano bagi para *user* khususnya bagi pemula.

D. MANFAAT PENELITIAN

Adapun manfaat dari penelitian yang diharapkan penulis dalam penelitian ini adalah sebagai berikut :

1. Bagi Mahasiswa

Melatih mahasiswa dalam menerapkan pengetahuan dan keterampilan yang diperoleh di bangku kuliah.

2. Bagi Kampus

Penelitian ini diharapkan dapat memberikan manfaat berupa sumbangan ilmu pengetahuan.

3. Bagi Masyarakat Umum

Penelitian ini dapat dijadikan sebagai hiburan yang dimainkan dalam android.

E. BATASAN MASALAH

Adapun batasan masalah dalam ruang lingkup penelitian ini, yaitu :

1. Aplikasi ini dibangun menggunakan *Augmented Reality* dengan *unity* sebagai *software* pendukung.
2. Aplikasi Virtual Piano ini tidak dapat di tekan lebih dari satu tombol/*tuts* dalam waktu yang bersamaan.
3. *User* menggunakan aplikasi untuk belajar *tuts* piano (do, re, mi, fa, so, la, si, do').
4. Aplikasi yang di rancang merupakan berbasis dekstop yang dimana pengguna akan berinteraksi dengan sebuah kertas yang akan berisi *marker*
5. Kualitas *texture* dan pola warna sangat mempengaruhi berhasil tidaknya pendeteksian pada *marker/image* serta kualitas dari pencahayaan.
6. Kecepatan dari aplikasi bergantung pada RAM dan *processor*.
7. Penelitian ini akan membuat alat musik yang dibuat adalah alat musik piano.

BAB II

TINJAUAN PUSTAKA

A. DEFINISI *AUGMENTED REALITY*

Augmented Reality (AR) merupakan teknologi yang dapat menampilkan informasi yang bersifat virtual namun disajikan pada pandangan dunia nyata.

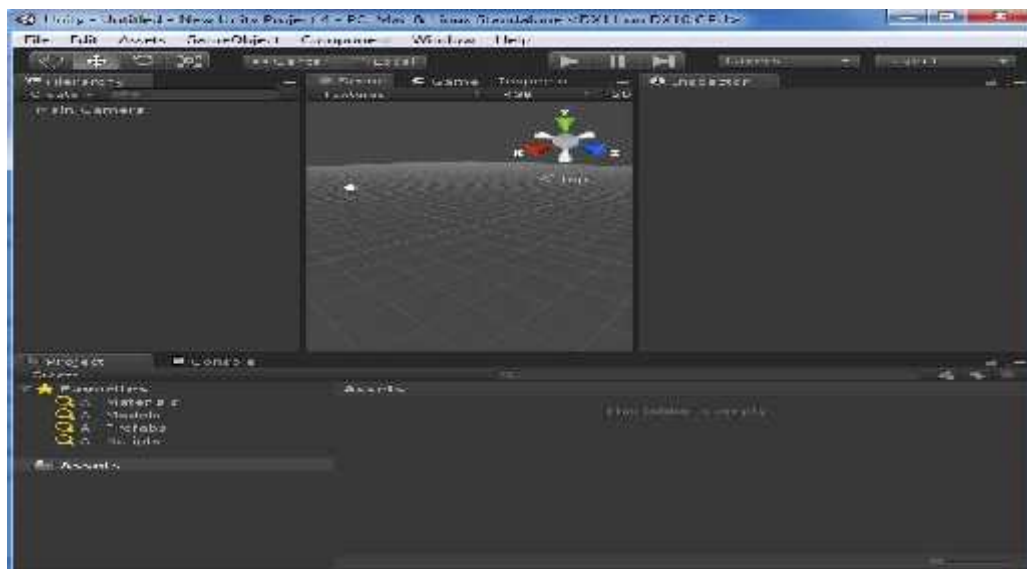
Menurut seorang pakar Ronal T Azuma (1997), mendefinisikan bahwa AR sebagai penggabungan benda-benda nyata dan maya pada lingkungan nyata. Pada intinya benda dunia maya terintegrasi pada dunia nyata. (*Febriyan M. Fajar, 2014*)

Menurut Suryawinata (2010), Augmented Reality (AR) adalah kombinasi antara dunia maya (virtual) dan dunia nyata (real) yang dibuat oleh komputer. Obyek virtual dapat berupa teks, animasi, model 3D atau video yang digabungkan dengan lingkungan sebenarnya sehingga pengguna merasakan obyek virtual berada dilingkungannya. (*Mario Fernando, 2014*)

1. DEFINISI *UNITY*

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan *game* multi *platform* yang di desain untuk mudah digunakan. Editor pada *unity* dibuat dengan *user interface* yang sederhana. Editor ini dibuat setelah ribuan jam ang mana telah dihabiskan

untuk membuatnya menjadi nomor satu dalam urutan ranking teratas untuk editor *game*. Grafis pada *unity* dibuat dengan grafis tingkat tinggi untuk *OpenGL* dan *directX*. *Unity* mendukung semua format *file*, terutamanya format umum seperti semua format dari *art applications*. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada *Mac OS x* dan *windows* dan dapat menghasilkan *game* untuk *Mac*, *Windows*, *Wii*, *iPhone* dan *Android*. (<http://www.hermantolle.com/class/docs/unity-3d-game-engine/>, diakses 16 April 2015)



Gambar 1. Lembar Kerja Unity

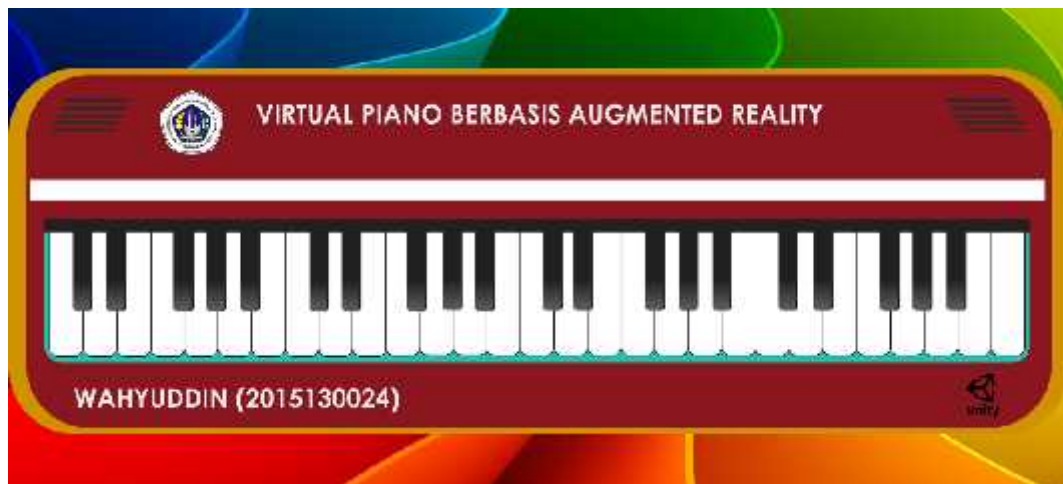
B. VUFORIA

Vuforia merupakan *software* untuk augmented reality yang dikembangkan oleh Qualcomm, yang menggunakan sumber yang konsisten mengenai computer vision yang fokus pada image recognition. Vuforia mempunyai banyak fitur-fitur dan kemampuan, yang dapat

membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknikal. (Mario Fernando, 2014)

C. MARKER

Marker merupakan perangkat keras lainnya yang digunakan untuk membuat suatu aplikasi *Augmented Reality*. Marker diperlukan sebagai penanda untuk menampilkan suatu objek. Marker ini dicetak dengan menggunakan printer untuk diarahkan langsung pada kamera.



Gambar 2. Marker Piano

D. NATURAL FEATURES DAN RATING

Rating dalam pengertian disini adalah sebuah nilai, seberapa baiknya gambar dapat dideteksi oleh sistem Vuforia. Rating pada gambar dapat dilihat pada Target Manager. Rating dapat dinilai antara 0 sampai 5 pada tiap gambar. Lebih tinggi nilai akan memudahkan sistem dalam mendeteksi gambar. Nilai diangka 3 sudah sangat sulit dikenal menggunakan device dengan kamera standard dan kondisi cahaya

kurang. Maka saya sarankan untuk mendapatkan nilai 4 atau 5 agar dapat dikenali menggunakan berbagai device dan keadaan cahaya.

Natural Features, features merupakan titik sudut siku-siku atau tajam dalam sebuah komponen gambar. Semakin banyak jumlah features dalam satu gambar, semakin tinggi pula rating yang akan didapat. *(Febriyan M. Fajar, 2014)*

Ponsel adalah platform kinerja rendah dengan sumber daya yang sangat terbatas dibandingkan dengan komputer. Akibatnya, keterbatasan ponsel dalam setiap aspek harus diperhitungkan ketika mengembangkan sebuah teknologi AR. Banyak teknologi AR, yang dirancang khusus untuk ponsel, menggunakan penanda berbasis pelacakan untuk menimbulkan estimasi dan ekstraksi. Pendekatan ini bekerja dengan cukup baik dan menyederhanakan proses pelacakan, sehingga bagi pengguna akhir, lingkungan harus dipersiapkan terlebih dahulu, hal ini membuat prosedur pengembangan menjadi rumit tetapi membawa fleksibilitas sehingga setiap benda dengan tekstur yang cukup rinci dapat dengan mudah dilacak. *(Mario Fernando, 2014)*

E. ANDROID

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Android berjalan pada *smartphone* saat ini dan menyesuaikan spesifikasi di kelas *low-end* hingga *high-end*.

Hampir semua vendor saat ini mengembangkan produknya dengan sistem operasi *Android*, karena peminatnya yang semakin meningkat tajam. (*Anis Ramadhani, 2013*).

F. VIRTUAL

Virtual button adalah sebuah area persegi di dalam area penanda, yang digunakan untuk men-*trigger* sebuah *event* tertentu dengan menyentuhnya. *Virtual button* dapat digunakan untuk mengimplementasikan *event* seperti *OnButtonPress* atau untuk mendeteksi apakah area tertentu dari *image target* ditutupi sebuah objek tertentu. *Event* pada *virtual button* hanya dapat dieksekusi ketika tampilan dari kamera dan kamera tersebut stabil. Artinya, jika terjadi perpindahan kamera secara cepat, maka *virtual button* akan di-*disable*.

Virtual button didefinisikan dalam konfigurasi *dataset .unitypackage* yang menjadi *property* dari *image target* (penanda). Dengan menggunakan *virtual button* interaksi antara objek 3 dimensi dengan pengguna dapat dilakukan.

G. PIANO

Piano adalah alat musik yang dimainkan dengan jari-jemari tangan dengan menggunakan *keyboard*. Untuk membuat rangkaian nada harmonis pada piano dapat menggunakan *chord*. *Chord* atau kunci nada adalah rangkaian dua nada atau lebih secara bersama – sama sehingga

menimbulkan kesan indah. Permainan *chord* digunakan untuk mengiringi jalannya suatu lagu [2].

Pada piano terdapat kunci not yang merupakan deretan tuts yang jika ditekan, maka akan mengeluarkan suara berdasarkan kunci note yang ada. Dimulai dari kunci paling kecil yang terletak di sebelah kiri berurutan hingga kunci paling besar di sebelah kanan. Tuts berwarna putih mengandung kunci *C Mayor*, yaitu: C,D,E,F,G,A,B. sedangkan tuts berwarna hitam mengandung kunci selain *C Mayor*, yaitu C#, D#, F#, G#, A#.

H. BAHASA C#

Pemrograman C# atau sering disebut *C Sharp* merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka. Seperti halnya bahasa pemrograman yang lain, C# bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis *windows (desktop)* dan aplikasi berbasis *web*, serta aplikasi berbasis *web services*. (*Yulius Eka Agung Seputra, Buku Pintar Pemrograman C#*).

I. UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik / gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan

standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software* (<http://www.omg.org>).

UML adalah standar dunia yang dibuat oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object-oriented* dan *software component*.

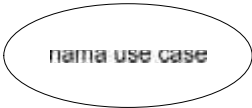
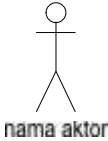

Diagram-diagram pada UML beserta urutan pembuatannya :

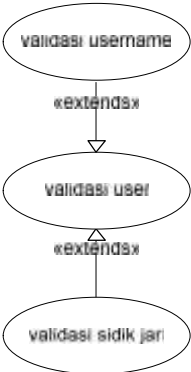
a. Diagram *Use Case*

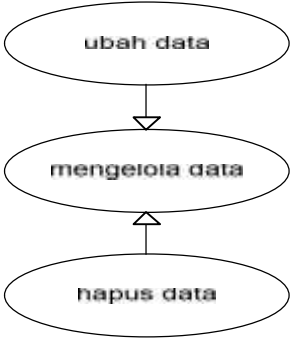
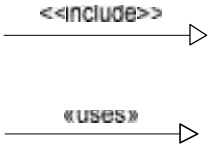
Use Case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

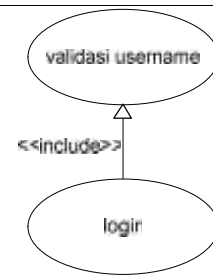
Diagram *Use Case* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Hal yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* menyatakan sebuah aktivitas atas pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja, dan lain sebagainya. Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. (Hamim Tohari, 2014)

Tabel 1. Diagram *Use Case*

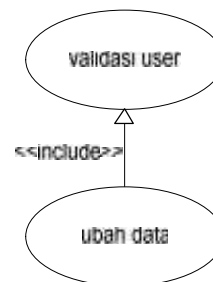
SIMBOL	DESKRIPSI
<p data-bbox="359 421 502 450"><i>Use Case</i></p> 	<p data-bbox="880 421 1362 815">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>Use Case</i></p>
<p data-bbox="359 862 539 891"><i>Aktor / actor</i></p> 	<p data-bbox="880 862 1362 1480">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, benda di awal frase nama aktor</p>
<p data-bbox="359 1527 671 1556"><i>Asosiasi / association</i></p> 	<p data-bbox="880 1527 1362 1776">Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan aktor</p>
<p data-bbox="359 1818 608 1848"><i>Ekstensi / extend</i></p>	<p data-bbox="880 1818 1362 1921">Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use</i></p>

<p style="text-align: center;">«extends» —————▶</p>	<p>Case yang ditambahkan dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek, biasanya <i>Use Case</i> tambahan memiliki nama depan yang sama dengan <i>Use Case</i> yang ditambahkan, misal :</p> <div style="text-align: center;">  <pre> graph TD A([validasi username]) -- «extends» --> B([validasi user]) C([validasi sidik jari]) -- «extends» --> B </pre> </div> <p>arak panah mengarah pada <i>Use Case</i> yang ditambahkan</p>
<p>Generalisasi / <i>generalization</i> —————▶</p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>

	 <p>arah panah mengarah pada <i>Use Case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan / <i>include</i> / <i>users</i></p> 	<p>Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>Use Case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>Use Case</i> :</p> <p>Include berarti <i>Use Case</i> yang ditambahkan akan selalu dipanggil saat <i>Use Case</i> tambahan dijalankan, misal pada kasus berikut :</p>



Tambahan akan selalu melakukan pengecekan apakah *Use Case* yang ditambahkan telah dijalankan sebelum *Use Case* tambahan dijalankan, misal pada kasus berikut :

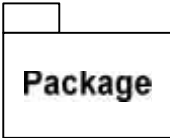
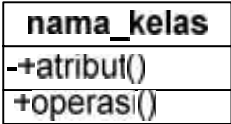




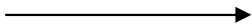
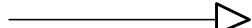


Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan. Arah panah include mengarah pada *Use Case* yang dipakai.

b. Diagram Kelas

Kelas (*class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek. Kelas menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode / fungsi). (*Hamim Tohari, 2014*)

Tabel 2. Diagram Kelas

SIMBOL	DESKRIPSI
<p><i>Package</i></p>  <p style="text-align: center;">Package</p>	<p><i>Package</i> merupakan sebuah bungkusan dari satu atau lebih kelas</p>
<p>Kelas</p> 	<p>Kelas pada struktur sistem</p>
<p>Antarmuka / <i>interface</i></p>  <p style="text-align: center;">Nama_interface</p>	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek</p>
<p>Asosiasi / <i>association</i></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i></p>
<p>Asosiasi berarah / <i>directed association</i></p>	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh</p>

	kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

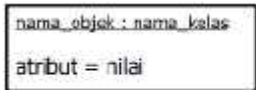

c. Diagram Objek

Objek Diagram adalah diagram yang memberikan gambaran atas model *instances* dari sebuah *class*. Objek-objek diidentifikasi dengan cara meletakkan nama *instance*-nya kemudian diikuti oleh tanda titik dua di depan nama *class*-nya. Nilai properti / atribut dituliskan berpasangan seperti "nama_atribut=nilai". Sedangkan notasi sebuah objek digambarkan segi empat yang terbagi atas 2 bagian.

Asosiasi antar dua buah objek biasanya dinotasikan dengan sebuah garis yang menghubungkan kedua objek diagram. Garis ini digunakan untuk menggambarkan sebuah sistem pada sebuah sudut pandang waktu tertentu. Dengan menggunakan diagram ini, anda dapat memeriksa keabsahan kelas-kelas diagram berikut aturan-aturan

multiplisitasnya dengan “*real data*” dan mengujinya dengan skenario-skenario tertentu. (Hamim Tohari, 2014)

Tabel 3. Diagram Objek

Simbol	Deskripsi
Objek 	objek dari kelas yang berjalan saat sistem dijalankan
Link 	relasi antar objek

d. Diagram Interaksi


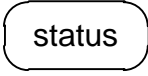


Interaction Diagram merupakan model yang menjelaskan bagaimana sejumlah objek bekerja sama dalam beberapa “kelakuan”. *Interaction* Diagram menerangkan kelakuan dari suatu *Use Case*. Diagram ini menggambarkan sejumlah objek dan pesan yang dijalankan antara objek dengan *Use Case*. Ketika kita memberikan pesan, aksi yang dihasilkan adalah sebuah pernyataan tereksekusi yang membentuk abstraksi dari prosedur komputasi. Sebuah aksi mungkin menghasilkan perubahan kondisi. (Hamim Tohari, 2014)


e. Diagram Status

Diagram status menjelaskan aliran kontrol dari satu status ke status lain. Status didefinisikan sebagai suatu kondisi dari suatu objek yang ada, dan perubahan yang terjadi sekiranya ada *event* yang terpicu.

Penekanan diagram status diberikan pada perubahan status tas beberapa *event* yang terjadi baik internal maupun eksternal. Transisi merupakan *event* sebagai hubungan antara beberapa status yang menunjukkan kapan sebuah objek dapat bergerak dari satu status ke status lainnya, saat bertemu dengan suatu kondisi. (Hamim Tohari, 2014)

Tabel 4. Diagram Status


SIMBOL	DESKRIPSI
Status awal / kondisi awal 	Status awal alur sebuah objek, sebuah diagram status memiliki sebuah status awal
Status 	Status yang dialami objek selama hidupnya
Status akhir / kondisi akhir 	Kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
Transisi 	Garis transisi antar status pada daur hidup objek, transisi biasanya diberi nama pesan yang ada pada diagram sekuen sehingga pesan pada diagram sekuen menjadi transisi bukan sebagai status, status merupakan





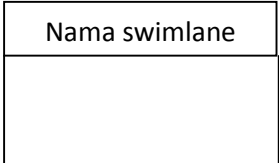
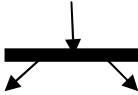
	<p> kondisi yang dialami objek, bukan merupakan pesan (<i>message</i>), transisi juga bisa memutar pada sebuah status</p>
	<p>Transisi internal melingkar / ke status sendiri</p>

f. Diagram Aktivitas

Activity Diagram memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Membuat *activity* diagram pada awal pemodelan proses cukup menguntungkan untuk membantu memahami keseluruhan proses. *Activity* diagram juga bermanfaat untuk menggambarkan paralel *behavior* atau menggambarkan interaksi antara beberapa *Use Case*. (Hamim Tohari, 2014)

Tabel 5. Diagram Aktivitas / Activity

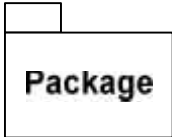
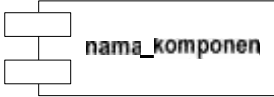



SIMBOL	DESKRIPSI
<p>Status awal</p> 	<p>Sebuah awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal</p>
<p>Aktivitas</p>	<p>Aktivitas yang dilakukan sistem,</p>

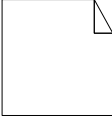
	<p>aktivitas biasanya diawali dengan kata kerja</p>
<p>Percabangan / <i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
<p>Penggabungan / <i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir</p>
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
	<p><i>Fork</i>, digunakan untuk menunjukkan kegiatan yang dilakukan</p>

g. Diagram Komponen

Komponen diagram digunakan untuk memvisualisasikan organisasi dan hubungan antar komponen dalam suatu sistem. Diagram ini juga digunakan untuk menjelaskan cara sistem dapat dieksekusi. (Hamim Tohari, 2014)

Tabel 6. Diagram Komponen

SIMBOL	DESKRIPSI
<p><i>Package</i></p> 	<p><i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen</p>
<p>Komponen</p> 	<p>Komponen sistem</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai</p>
<p>Antar muka / <i>interface</i></p> 	<p>Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen</p>
<p>Link </p>	<p>Relasi antar komponen</p>

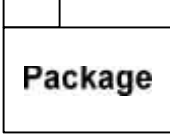
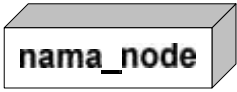

	Dokumen, dapat berupa file, library
---	-------------------------------------

h. Diagram *Deployment*

Diagram *deployment* menunjukkan tata letak sebuah sistem secara fisik. Diagram ini akan menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut.

Penggambaran arsitektur fisik sebuah aplikasi melibatkan dua perangkat yaitu, perangkat lunak dan perangkat keras. Perangkat-perangkat tersebut disebut Node. Node akan menunjukkan bagaimana komponen perangkat lunak dan keras ini bekerja sama. Node digambarkan dalam diagram *deployment* ini. (Hamim Tohari, 2014)

Tabel 7. Diagram Deployment

SIMBOL	DESKRIPSI
<p><i>Package</i></p> 	<p><i>Package</i> merupakan sebuah bungkus dari satu atau lebih node</p>
<p>Node</p> 	<p>Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Kebergantungan antar node, arah panah mengarah pada node yang dipakai</p>

J. TAHAPAN PENELITIAN

Adapun tahapan yang dilakukan oleh penulis dalam perancangan aplikasi ini adalah sebagai berikut :

a. Pengumpulan Data

Penulis mengumpulkan data-data dan informasi yang terkait untuk dapat dijadikan sebagai acuan dalam perancangan aplikasi.

b. Analisis Sistem

Menganalisis masalah yang ada kemudian mencari alternatif pemecahannya.

c. Desain Sistem

Setelah masalah ditentukan dan analisis data telah dilakukan, maka perlu dilakukan desain sistem terhadap masalah yang telah ditentukan.

d. *Coding*

Tahap penulisan kode (*listing*) program sistem.

e. Pengujian Sistem

Setelah proses *coding* selesai, maka dilakukan pengujian sistem menggunakan metode pengujian *black box*.

f. Implementasi Sistem

Tahap ini digunakan untuk penerapan dan pengujian sistem ke dalam ruang lingkup yang sebenarnya agar dapat diketahui kelebihan dan kekurangan dari sistem/aplikasi yang telah dibangun.

g. Penulisan Laporan

Tahapan ini digunakan oleh penulis dalam pembuatan laporan.

K. METODE PENGUJIAN

a. Pengujian *White Box*

Pengujian *white-box*, yang kadang-kadang disebut pengujian *glass-box*, adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white-box*, perancang sistem dapat melakukan *test case* yang ;

- 1) Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
- 2) Menggunakan semua keputusan logis pada sisi *true* dan *false*.
- 3) Mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka.
- 4) Menggunakan struktur data internal untuk menjamin validitasnya.

Pada titik ini, dapat diajukan pertanyaan yang berbatasan, yaitu :
“Mengapa menghabiskan waktu dan energi untuk mencemaskan (dan menguji) logika jika kita dapat dengan lebih baik memperluas kerja yang dapat memastikan bahwa persyaratan program telah dipenuhi?” Bila dinyatakan dengan cara lain, mengapa kita tidak

menguji semua energi kita untuk melakukan pengujian *black-box*? Jawabannya ada pada sifat cacat perangkat lunak, seperti :

- a) Kesalahan logis dan asumsi yang tidak benar berbanding terbalik dengan probabilitas jalur program yang akan dieksekusi. Kesalahan cenderung muncul dalam kerja kita pada saat kita mendesain dan mengimplementasi fungsi, kondisi, atau kontrol yang berada di luar mainstream. Pemrosesan setiap hari cenderung dipahami dengan baik (dan diteliti dengan baik) sementara pemrosesan “kasus khusus” cenderung berantakan.
- b) Kita sering percaya bahwa jalur logis mungkin tidak akan dieksekusi bila pada kenyataannya akan dieksekusi pada basis reguler. Aliran logika dari suatu program kadang-kadang bersifat konterintuitif, yang berarti asumsi kita yang tidak kita sadari mengenai aliran dan data kontrol dapat menyebabkan kita membuat kesalahan desain yang akan terungkap hanya setelah pengujian jalur dimulai.
- c) Kesalahan tipografis adalah random. Bila sebuah program diterjemahkan ke dalam kode sumber bahasa pemrograman maka dimungkinkan akan terjadi banyak kesalahan pengetikan. Beberapa akan ditemukan dengan mekanisme pengecekan sintaks, tetapi yang lainnya akan tetap tidak terdeteksi sampai pengujian dimulai.

Dimungkinkan bahwa suatu type akan ada pada suatu jalur logika tidak jelas seperti pada jalur mainstream.

Masing-masing atasan tersebut memberikan suatu argumen untuk melakukan pengujian *white-box*. Pengujian *black-box*, tidak peduli seberapa cermat dilakukan, dapat tidak menangkap bentuk kesalahan tersebut. Seperti dikatakan oleh Beizer (BEI90): “*Bug* bersembunyi di sudut-sudut dan berkerumun di perbatasan-perbatasan.” Tetapi pengujian *white-box* sangat mungkin untuk menemukannya.

b. Pengujian *Black-box*

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekrutan perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan dari pada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- 1) Fungsi-fungsi yang tidak benar atau hilang.
- 2) Kesalahan *interface*.

- 3) Kesalahan dalam struktur data atau akses *database* eksternal.
- 4) Kesalahan kinerja.
- 5) Inisialisasi dan kesalahan terminasi.

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi. Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut :

- a) Bagaimana validitas fungsional diuji?
- b) Kelas input apa yang akan membuat *test case* menjadi baik?
- c) Apakah sistem sangat sensitif terhadap harga input tertentu?
- d) Bagaimana batasan dari suatu data diisolasi?
- e) Kecepatan data apa dan volume data apa yang dapat ditolerir oleh sistem?
- f) Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

Dengan mengaplikasikan teknik *black-box*, maka kita menarik serangkaian *test case* yang memenuhi kriteria berikut ini : (1) *test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test*

case tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan, dan (2) *test case* yang memberi tahu kita sesuatu mengenai kehadiran atau ketidakhadiran kelas kesalahan, dari pada memberi tahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada.

L. ROADMAP PENELITIAN

Penelitian ini di tulis oleh Mochamad Fathoni, Eko Budi Cahyono, S.Kom., MT, Wahyu Andhyka Kusuma, S.Kom pada tahun 2012 dengan judul “**Alat Musik Perkusi Augmented Reality Berbasis Android**” Penelitian ini menggunakan bahasa pemrograman C#. Perkusi adalah sebutan bagi semua instrumen musik yang teknik permainannya dipukul, baik menggunakan tangan maupun stik. Beberapa instrumen musik yang tergolong dalam alat musik perkusi adalah gamelan, arumba, kendang, kolintang, tifa, talempong, rebana, bedug, jimbe dan lain sebagainya.

Penelitian ini di tulis oleh Rudi Syaifudin, Eko Budi Cahyono, Wahyu Andhyka Kusuma dengan judul “**Game Gitar Virtual Berbasis Augmented Reality**” Penelitian ini menggunakan bahasa pemrograman *ActionScript*. *Game* ini menantang *player* untuk menekan / menutup *marker* yang tepat dan pada waktu yang tepat. Pada *game* ini juga menampilkan *marker* mana yang harus ditutup dan jika tepat maka *point* akan ditambahkan, jika tidak, *point* akan dikurangi.

Berbeda dengan 2 penelitian di atas, penulis mengangkat judul “*Virtual Piano menggunakan Augmented Reality*”, bahasa program yang digunakan adalah C#. Adapun tujuan dari penelitian ini adalah membangun aplikasi yang berfungsi sebagai media dan metode dalam proses pembelajaran alat musik piano bagi para *user* khususnya bagi pemula. Konsep dari aplikasi ini adalah pendeteksian interaksi sentuhan tangan pada *marker* sehingga ketika *marker* tersebut tersentuh tangan maka akan menimbulkan bunyi. Aplikasi ini berjalan pada *Android Mobile*.

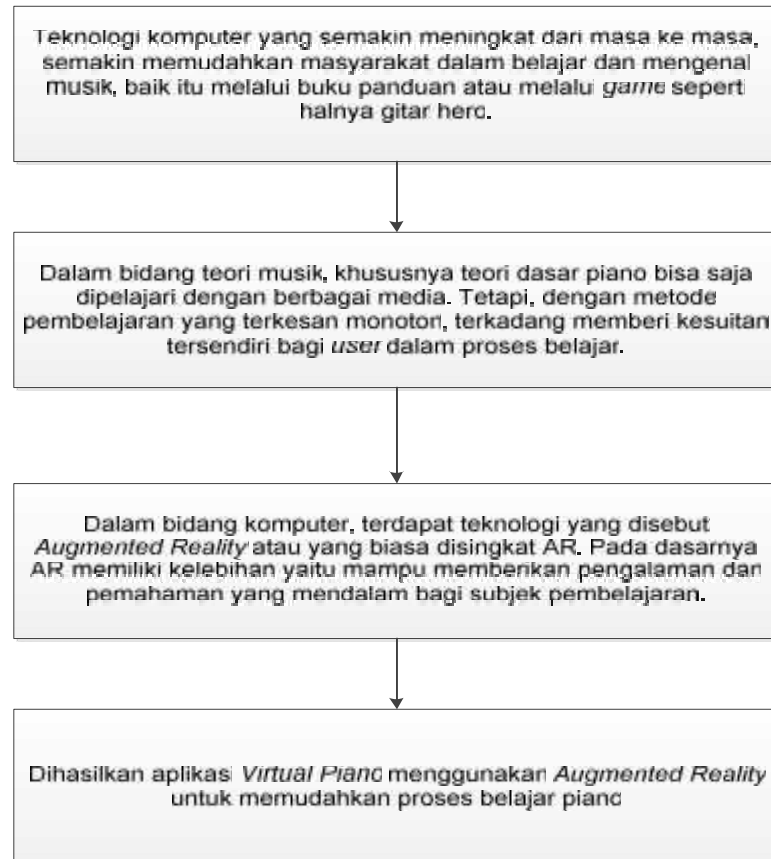
Penelitian ini ditulis oleh Irman, Muhamad Tahun 2012 dengan judul “**sebuah aplikasi pengenalan dan simulasi alat musik tradisional Betawi menggunakan teknologi Augmented Reality**” pada penelitian Penulis juga akan hanya membuat satu marker yang terdapat beberapa objek 3D alat musik Betawi pada aplikasi ini.

Peneliti ini ditulis oleh Tommi Suryant, Ema Utami dan Hanif Al Fatta pada tahun 2014 dengan judul “**Rancang Bangun Virtual Gamelan Mobile Menggunakan Augmented Reality**” 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*.

Penelitian ini di tulis oleh Meylisa Rasjid, Rizal Sengkey dan Stanley Karouw pada tahun 2016 dengan judul **“Rancang Bangun Aplikasi Alat Musik Kolintang menggunakan *Augmented Reality*”** Penelitian yang berjudul “Rancang Bangun Aplikasi Alat Musik Kolintang menggunakan *Augmented Reality* berbasis Android” ini merupakan sebuah perancangan yang bertujuan untuk menghasilkan sebuah aplikasi *Augmented Reality* Kolintang yang dapat berjalan pada sistem operasi android. Metode yang digunakan sebagai tahapan penelitian ini adalah metode *Prototype* yang merupakan metode proses perangkat lunak yang bisa memahami kebutuhan *user* dan dapat diperbaiki secara terus menerus sampai sesuai dengan kebutuhan *user*. Aplikasi yang di bangun menggunakan beberapa *tools* yaitu UML (*Unified Modelling Language*), Unity, Vuforia dan Blender3D. Maka aplikasi yang di hasilkan yaitu aplikasi *augmented reality* yang dapat menampilkan alat musik kolintang berbentuk 3D serta jenis suara pada setiap jenis kolintang. Membuat model alat musik Kolintang tiga dimensi dalam aplikasi blender yang akan di jadikan objek dari pembuatan aplikasi *Augmented Reality* alat musik Kolintang serta cara meng*export* hasil modeling dan hasil desain *marker*.

M. KERANGKA PIKIR PENELITIAN



Gambar 3. Kerangka Pikir Penelitian

BAB III

METODOLOGI PENELITIAN

A. LOKASI DAN WAKTU PENELITIAN

Pelaksanaan penelitian ini direncanakan selama 2 bulan. Penelitian yang dilakukan oleh penulis berlokasi di Studio Teknik Informatika Universitas Muhammadiyah Parepare, bertempat di Jl. Jend. Ahmad Yani Km.6 Parepare.

B. TAHAPAN PENELITIAN

Dalam membangun aplikasi augmented reality dengan virtual piano sebagai topik penelitian, ada beberapa tahapan yang akan dilakukan yakni :

1. Pengumpulan data.

Pengumpulan data untuk penelitian ini dilakukan studi pustaka, yaitu pengumpulan data dengan membaca buku-buku referensi yang terkait dengan penelitian ini. Studi pustaka antara lain mencari jurnal-jurnal dan buku tentang augmented reality dengan virtual piano menggunakan augmented reality.

2. Rancang bangun aplikasi virtual piano dengan augmented . reality terdiri dari empat tahap :

- a. Analisis sistem aplikasi

Pada tahap ini akan dilakukan analisa kebutuhan perangkat lunak yang akan dikembangkan, yaitu apa saja yang dibutuhkan untuk merancang dan membangun aplikasi ini.

b. Desain sistem aplikasi

Pada tahap ini akan dilakukan perancangan model perangkat lunak yang akan dikembangkan, yaitu menggunakan android SDK dan vuforia.

c. Perkodean sistem aplikasi

Pada tahap ini akan dilakukan proses penulisan program untuk merealisasikan rancangan sistem dengan menggunakan bahasa pemrograman atau alat bantu framework aplikasi.

d. Pengujian sistem aplikasi

Pada tahap ini akan dilakukan proses pengujian fungsionalitas sistem aplikasi yang telah dikembangkan. Pengujian yang dilakukan adalah menghasilkan bunyi not piano pada marker.

3. Pembuatan jurnal

Pada tahap ini akan dilakukan pembuatan makalah untuk jurnal yang merupakan uraian singkat dari hasil penelitian.

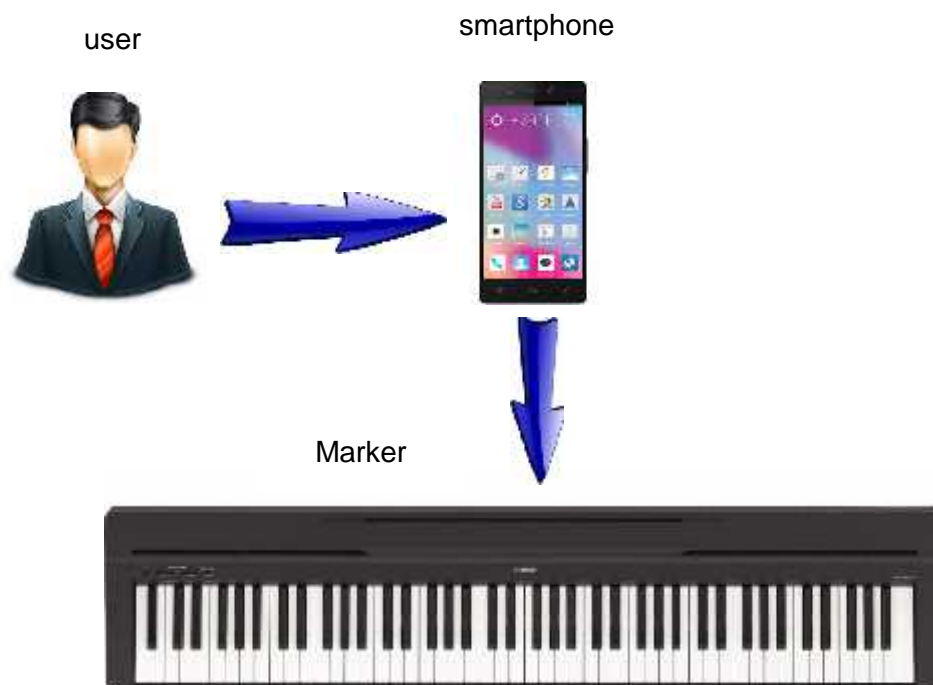
4. Pendistribusian angket kepuasan pengguna

Pada tahap ini akan dilakukan distribusi angket yang berfungsi untuk mengukur tingkat kepuasan pengguna terhadap

perangkat lunak yang dikembangkan yang dalam hal ini pengguna virtual piano ini dengan menggunakan augmented reality (AR) yang berbasis android di studio informatika kota parepare.

C. RANCANGAN SISTEM

Rancangan sistem di sini adalah untuk menggambarkan setiap komponen, *database* maupun *platform* teknologi yang akan dirancang.



Gambar 4. Rancangan Sistem

Pada rancangan sistem yang diatas, setiap user harus memiliki kamera mobile (android) setelah itu user mengarahkan kameranya kegambar piano (marker), setelah itu marker akan diidentifikasi dan ketika gambar piano (marker) di kenali maka gambar tersebut akan menghasilkan bunyi piano ketika ditekan.

D. PERANGKAT PENGEMBANGAN SISTEM

Dalam melakukan penelitian, penulis memerlukan alat dan bahan penelitian yang mendukung kegiatan penelitian tersebut. Alat dan bahan yang diperlukan antara lain adalah sebagai berikut :

1. Alat Penelitian

a. Alat penelitian yang digunakan selama proses penelitian, yaitu :

1) *Hardware*

a) *Personal notebook* dengan spesifikasi *Hardware* sebagai berikut:

- *Processor* : Intel® Core™2 Duo CPU processor
- *Memory* : 2 GB
- *Monitor* : 14 inci
- *Printer* : Canon PIXMA MP 258

b) *HP Android* dengan spesifikasi *Hardware* sebagai berikut :

- *Chipset* : *Intel Atom Z2520*
- *Memory* : 4 GB
- *Display Size* : 4 inci (-59,6% *screen-to-body ratio*)
- *OS* : *Android OS v4.4.2 (KitKat)*

2) *Software*

Alat penelitian yang digunakan berupa *software* yaitu :

- *Sistem Operasi* : *Windows 7*

- Aplikasi Program : *Unity, SDK Vuforia, SDK Android, JDK (Java Development Kit)*
- Bahasa pemrograman : Bahasa C#

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

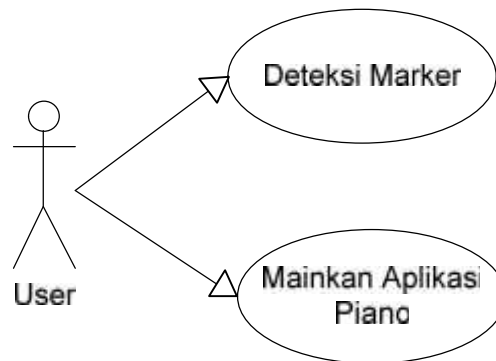
A. RANCANGAN SISTEM YANG DIUSULKAN

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

Analisis Sistem Yang diusulkan adalah sebuah sistem yang dapat memudahkan *user* dalam belajar alat musik piano tanpa harus menggunakan alat musik piano yang sebenarnya. Dengan adanya aplikasi ini diharapkan dapat lebih menarik minat masyarakat untuk belajar alat musik khususnya piano. Namun demikian, sistem ini hanya untuk pemula mengingat keterbatasan tuts nada yang disediakan.

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

1. Use Case Diagram



Gambar 5 Use Case Diagram

Deskripsi :

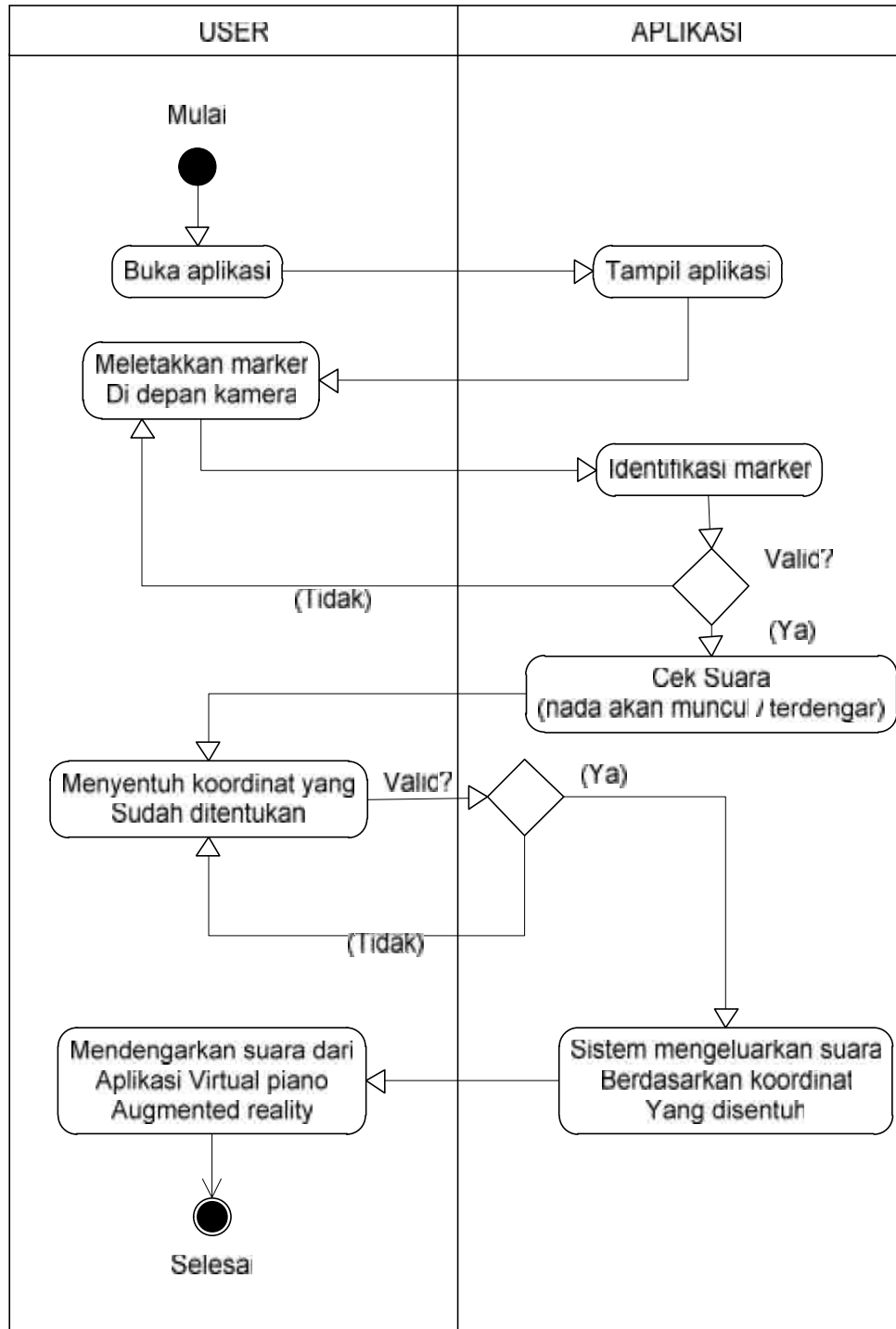
Tabel 8 Use Case Deteksi Marker

USER	SISTEM
1. Membuka aplikasi	2. Tampil Aplikasi
3. Meletakkan marker di depan kamera	4. Validasi marker
	5. Jika tidak valid, maka kembali ke no.3
	6. Jika valid, maka muncul suara

Tabel 9 *Use Case* Mainkan Aplikasi Piano

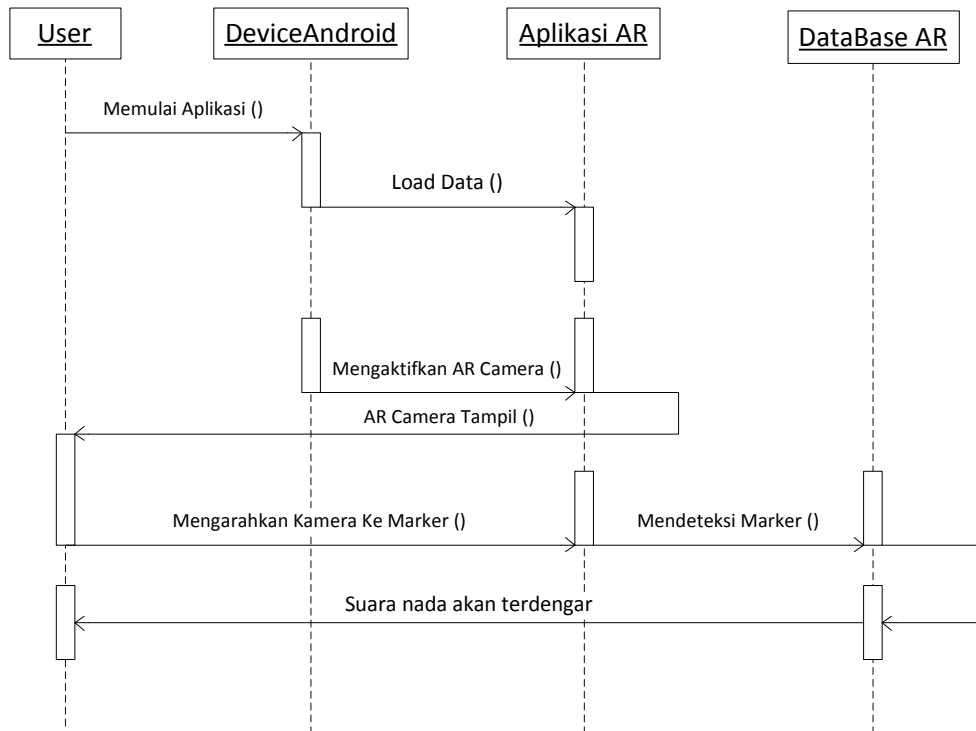
USER	SISTEM
1. Menyentuh koordinat tuts piano pada marker	2. Deteksi sentuhan
	3. Validasi koordinat marker yang disentuh
	4. Jika tidak valid, maka kembali ke no.1
	5. Jika valid, maka muncul suara sesuai koordinat yang disentuh

2. Activity Diagram



Gambar 6 Activity Diagram

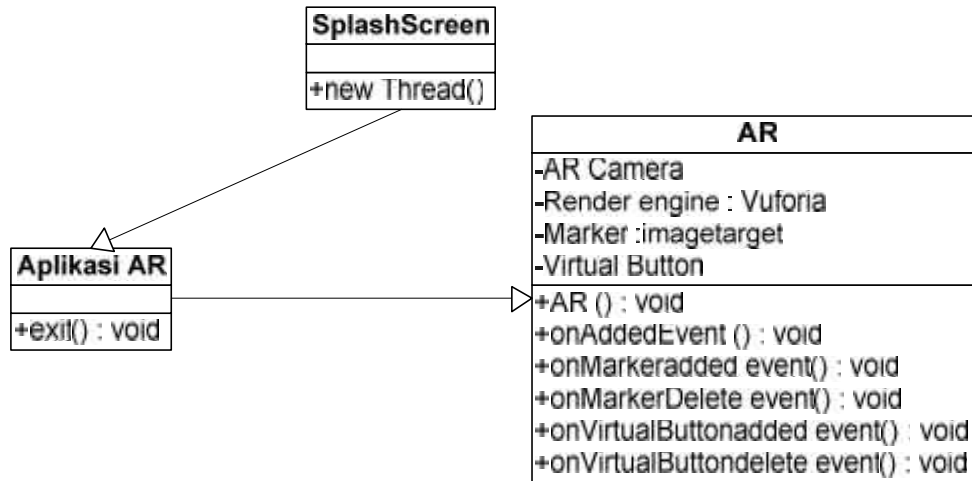
3. Sequence Diagram



Gambar 7 Sequence Diagram

Diagram *Sequence* menjelaskan bahwa ketika *user* memulai aplikasi maka *device* Android akan me-load data untuk selanjutnya memulai aplikasi. Setelah aplikasi terbuka, maka AR camera akan aktif. Langkah selanjutnya adalah *user* mengarahkan kamera ke marker yang telah disediakan ketika AR camera tampil. Selanjutnya, aplikasi akan melakukan proses pendeteksian marker untuk melakukan pencocokan dengan *database*, dan jika sesuai maka nada akan terdengar.

4. Class Diagram

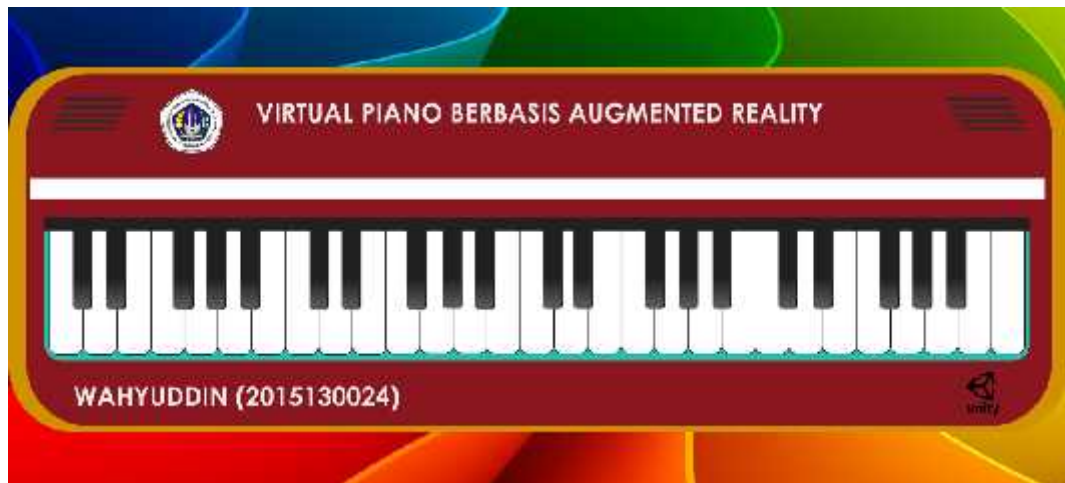


Gambar 8 Class Diagram

Diagram Kelas (*class*) menjelaskan mengenai keterkaitan antara fungsi-fungsi yang ada didalamnya, yakni ketika pertama kali memulai aplikasi maka yang muncul pertama adalah *splashscreen* sebelum masuk ke *AR camera*. Ketika *AR camera* telah tampil, maka disaat itu akan terjadi proses pendeteksian marker dengan menyentuh titik koordinat tertentu.

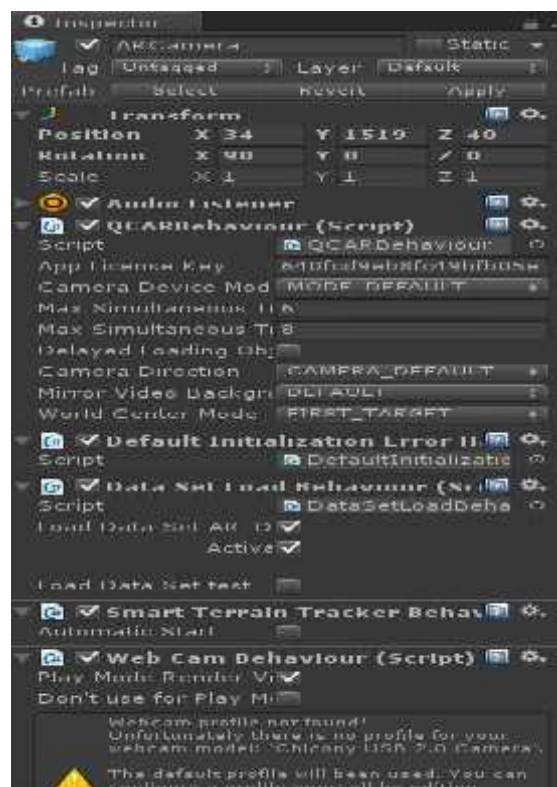
B. RANCANGAN OUTPUT

Output merupakan produk dari sistem informasi yang dapat dilihat. Output ini dapat berupa hasil yang dikeluarkan ke media keras seperti kertas dan *output* yang berupa hasil dikeluarkan ke media lunak seperti tampilan di layar. Berikut adalah *output* dari rancangan sistem :



Gambar 9 Aplikasi dijalankan

C. RANCANGAN INPUT



Gambar 10 Form Inspector

D. IMPLEMENTASI DAN PENGUJIAN SISTEM

1. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Pada tahap ini, desain sistem yang telah dirancang diterapkan ke dalam bahasa pemrograman yang kemudian dilakukan pengujian sistem.

Adapun bahasa pemrograman yang digunakan untuk membuat aplikasi *Virtual Piano* menggunakan *Augmented Reality* adalah bahasa pemrograman *C#*.

2. Pengujian Sistem

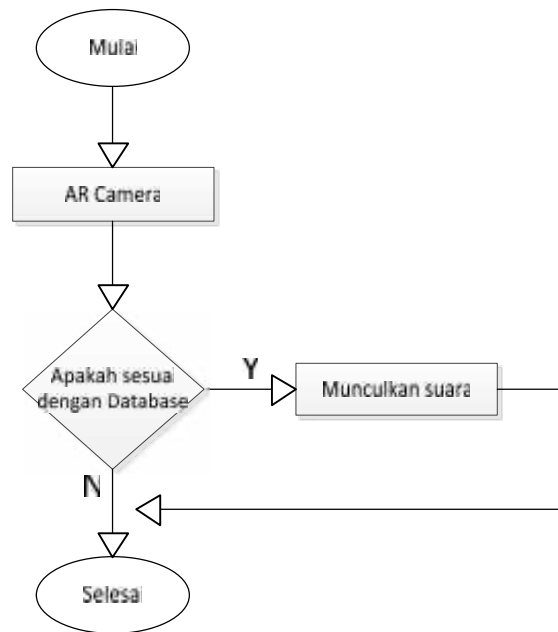
Pengujian aplikasi bertujuan untuk memastikan bahwa aplikasi telah memiliki fungsi seperti yang diharapkan dan menguji aplikasi bebas dari kesalahan logika.

Pengujian black-box berkaitan dengan pengujian yang dilakukan pada interface perangkat lunak. Sedangkan, pengujian white-box berfokus pada efektifitas aplikasi yang dirancang.

1. Pengujian *White Box*

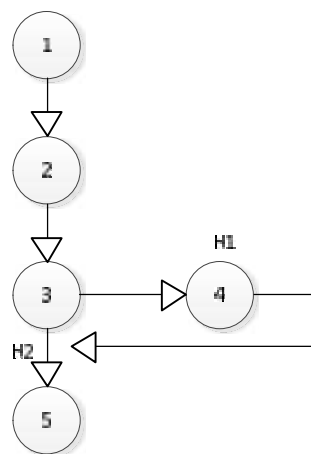
a. White Box Aplikasi

1) FlowChart Aplikasi



Gambar 11 FlowChart Aplikasi

2) FlowGraph Aplikasi



Gambar 12 FlowGraph Aplikasi

3) Proses Perhitungan

Dari Gambar FlowGraph di atas dapat dilakukan proses perhitungan sebagai berikut :

a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$N(\text{node}) = 5$

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

$P(\text{predikat node}) = 1$

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

$$= 5 - 5 + 2$$

$$= 2$$

Predikat Node (N) = $P + 1$

$$= 1 + 1$$

$$= 2$$

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent Path* pada *flowgraph* diatas adalah :

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

Path 2 = 1 - 2 - 3 - 5

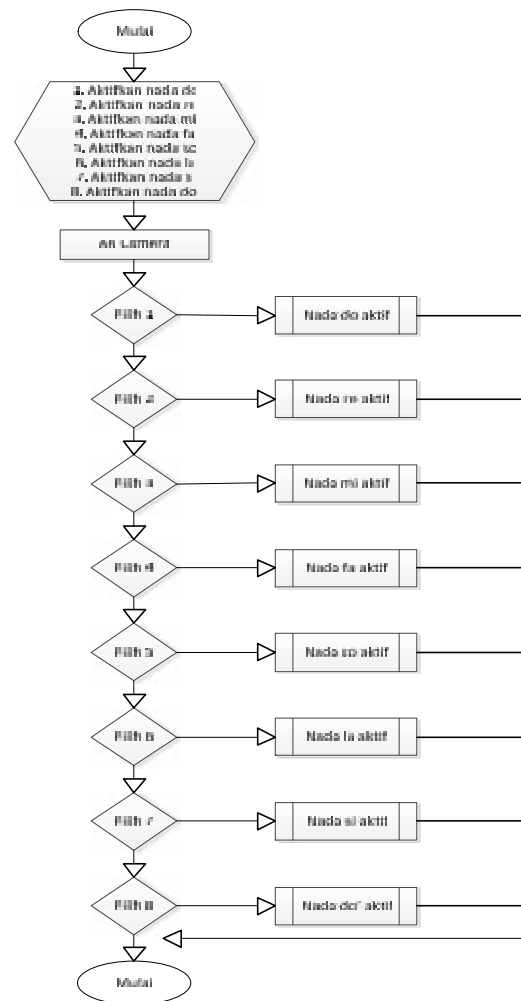
d) Grafik Matriks

Tabel 10 Grafik Matriks

	1	2	3	4	5	E-1
1		1				$1-1=0$
2			1			$1-1=0$
3				1	1	$2-1=1$
4					1	$1-1=0$
5						
Zum (E+1)						$1+1=2$

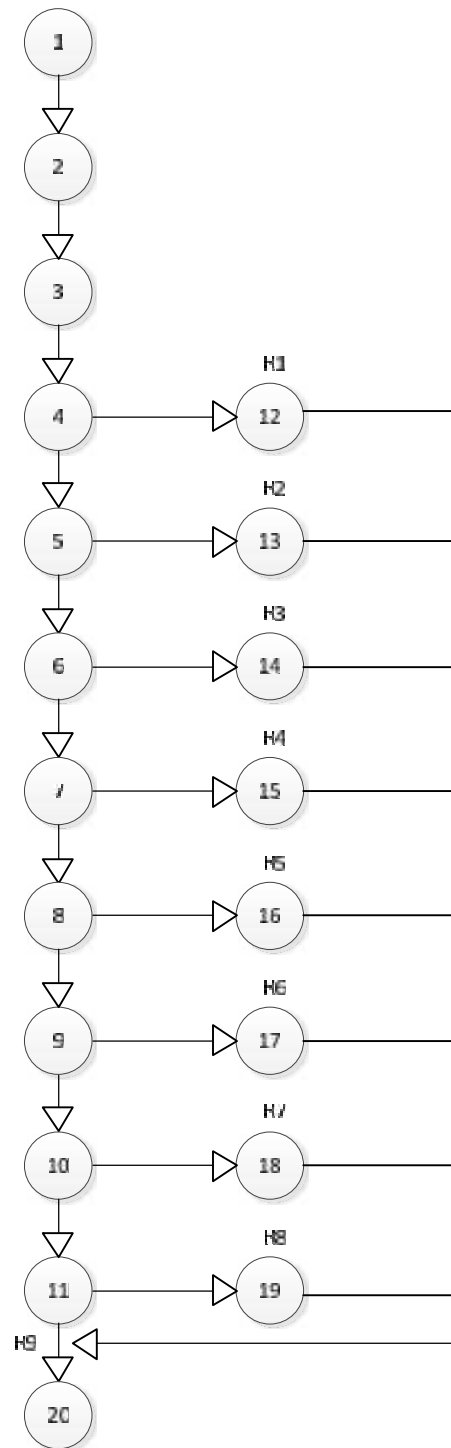
b. White Box Pengaturan

1) FlowChart Pengaturan



Gambar 13 FlowChart Pengaturan

2) FlowGraph Pengaturan



Gambar 14 FlowGraph Pengaturan

3) Proses Perhitungan

Dari Gambar FlowGraph di atas dapat di lakukan proses perhitungan sebagai berikut :

- a) Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*:

Dengan Rumus : $V(G) = E - N + 2$

$N(\text{node}) = 20$

$E(\text{edge}) = 20$

$P(\text{predikat node}) = 8$

Penyelesaian : $V(G) = E - N + 2$
 $= 20 - 20 + 2$
 $= 2$

Predikat Node (N) $= P + 1$
 $= 8 + 1$
 $= 9$

- b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 9

- c) *Independent Path* pada *flowgraph* diatas adalah :

Path 1 = 1 – 2 – 3 – 4 – 12 – 20

Path 2 = 1 – 2 – 3 – 4 – 5 – 13 – 20

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 14 – 20

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 15 – 20

Path 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 16 – 20

Path 6 = 1 – 2 – 3 – 4– 5– 6 – 7– 8 – 9 – 17 – 20

Path 7 = 1 – 2 – 3 – 4– 5– 6 – 7– 8 – 9 – 10 – 18 – 20

Path 8 = 1 – 2 – 3 – 4– 5– 6 – 7– 8 – 9 – 10 – 11 – 19 – 20

Path 9 = 1 – 2 – 3 – 4– 5– 6 – 7– 8 – 9 – 10 – 11 – 20

d) Grafik Matriks

Tabel 11. Grafik Matriks

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	E-1		
1		1																				1-1=0	
2			1																				1-1=0
3				1																			1-1=0
4					1							1											2-1=1
5						1							1										2-1=1
6							1							1									2-1=1
7								1							1								2-1=1
8									1							1							2-1=1
9										1							1						2-1=1
10											1							1					2-1=1
11																			1	1			2-1=1
Zum (E+1)																					8+1=9		

2. Pengujian *Black Box*

Pengujian aplikasi ini meliputi pengujian semua not (do, re, mi, fa, so, la, si, do') yang tersedia melalui pendeteksian marker. Durasi tiap nada adalah 1 detik dan jarak perpindahan antar nada adalah ± 2 detik. Hasil pengujian dapat dilihat pada tabel Pengujian Aplikasi :

Tabel 12. Pengujian Aplikasi

Nama Pengujian	Bentuk Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
Cek Koneksi	Menjalankan aplikasi pertama kali dengan tanpa koneksi internet	Muncul pemberitahuan untuk pengecekan koneksi	Berhasil
Pengujian kamera mendeteksi not "do"	Mengarahkan kamera ke marker lalu tekan not "do"	Muncul suara nada "do"	Berhasil
Pengujian kamera mendeteksi not "re"	Mengarahkan kamera ke marker lalu tekan not "re"	Muncul suara nada "re"	Berhasil
Pengujian kamera mendeteksi not "mi"	Mengarahkan kamera ke marker lalu tekan not "mi"	Muncul suara nada "mi"	Berhasil
Pengujian kamera mendeteksi not "fa"	Mengarahkan kamera ke marker lalu tekan not "fa"	Muncul suara nada "fa"	Berhasil
Pengujian kamera mendeteksi not "so"	Mengarahkan kamera ke marker lalu tekan not "so"	Muncul suara nada "so"	Berhasil
Pengujian kamera mendeteksi not "la"	Mengarahkan kamera ke marker lalu tekan not "la"	Muncul suara nada "la"	Berhasil
Pengujian kamera mendeteksi not "si"	Mengarahkan kamera ke marker lalu tekan not "si"	Muncul suara nada "si"	Berhasil
Pengujian kamera mendeteksi not "do' "	Mengarahkan kamera ke marker lalu tekan not "do' "	Muncul suara nada "do' "	Berhasil

Adapun Hasil Uji Akurasi dari Pendeteksian Marker adalah sebagai berikut :

Tabel 13. Pengujian Akurasi

Marker	Jarak minimum	Jarak maximum	Jarak optimal	Derajat kemiringan	Ket.
Piano	±30 cm	±60 cm	±45 cm	±90 ⁰	berhasil

Berdasarkan pengujian, dapat diambil kesimpulan bahwa kesalahan dari sintaks dan kualitas marker secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan. Selain itu kualitas texture dan pola warna sangat mempengaruhi berhasil tidaknya proses pendeteksian pada marker/image. Dan hal yang paling utama yaitu kualitas dari pencahayaan pada saat melakukan pengujian.

Tabel 14. Hasil Pengukuran Marker

Jarak	Marker	KET	
20 cm	✓	✓=bunyi x= tidak bunyi	
25 cm	✓		
30 cm	✓		
35 cm	✓		
40 cm	✓		
45 cm	✓		
50 cm	✓		
55 cm	✓		
60 cm	✓		
65 cm	x		
70 cm	x		
Derajat Kemiringan	Marker		
30 ⁰	x		
35 ⁰	x		
40 ⁰	✓		
45 ⁰	✓		
50 ⁰	✓		
55 ⁰	✓		

60 ⁰	✓	
65 ⁰	✓	
70 ⁰	✓	
75 ⁰	✓	
80 ⁰	✓	
85 ⁰	✓	
90 ⁰	✓	

3. Pengujian nada

Nada	1 Nada	Nada1 dan Nada2 bersamaan	Nada1, Nada2 dan Nada3 bersamaan	Ket.
Waktu	00.00.00.09	00.00.00.9, 00.00.00.17	00.00.00.9, 00.00.00.17, 00.00.00.22,	berhasil

Pada tabel diatas menjelaskan bahwa ketika 1 nada yang ditekan akan akan menghasilkan bunyi dengan waktu 0,9 detik sedangkan nada1 dan nada2 ditekan secara bersamaan itu memiliki jeda waktu 0,17 detik dan ketika nada1, nada2 dan nada3 ditekan secara bersamaan memiliki jeda waktu 0,22 detik.

Pada pengujian penelitian telah memiliki beberapa kekurangan yaitu, ketika ingin menekan 1 buah tuts, posisi tangan harus tidak menutupi tuts yang lain ketika kita menutupi tuts yang lain maka nada yang dikeluarkan akan menghasilkan 2 nada

BAB V

PENUTUP

A. KESIMPULAN

Berdasarkan hasil pengujian dan analisis aplikasi *Virtual Piano* menggunakan *Augmented Reality*, maka dapat disimpulkan hal-hal sebagai berikut :

1. *Virtual Piano* merupakan sarana yang menarik dan saat dimainkan seperti piano yang sesungguhnya.
2. Ketika 1 nada yang ditekan akan akan menghasilkan bunyi dengan waktu 0,9 detik sedangkan nada1 dan nada2 ditekan secara bersamaan itu memiliki jeda waktu 0,17 detik dan ketika nada1, nada2 dan nada3 ditekan secara bersamaan memiliki jeda waktu 0,22 detik.
3. Tingkat keberhasilan deteksi marker pada jarak 20-60 cm dan posisi kemiringan tingkat keberhasilannya pada $40-90^{\circ}$
4. Tingkat kegagalan deteksi marker pada jarak 65-70 cm dan posisi kemiringan tingkat kegagalannya pada $30-35^{\circ}$
5. Pemanfaatan teknologi *Augmented Reality* pada aplikasi ini dapat berjalan sesuai dengan perancangan, yaitu dapat menggabungkan objek *virtual* dengan lingkungan nyata secara *realtime*.

B. SARAN

Setelah melakukan penelitian, maka penulis memberikan saran sebagai berikut :

1. Penulis menyarankan agar kedepannya aplikasi *Virtual Piano* ini dapat lebih berkembang dengan menambahkan lebih banyak tuts-tuts piano.
2. Penulis menyarankan agar kedepannya aplikasi ini dapat lebih meningkatkan kualitas dari segi tampilan aplikasi dan kualitas audio yang dimasukkan ke dalam *virtual button*.
3. Penulis menyarankan agar kedepannya aplikasi ini dapat lebih responsive.
4. Penulis menyarankan agar aplikasi ini dijalankan pada ram dan processor yang tinggi karena hal tersebut ikut menentukan kecepatan dari aplikasi ini.

DAFTAR PUSTAKA

- Fajar, M.Febriyan. 2014. *Mobile Interactive Augmented Reality*. Bandung : Buku AR Online.
- Fernando, Mario. 2014. *Membuat Aplikasi Android Augmented Reality Menggunakan Vuforia SDK dan Unity*. Bandung : Buku AR Online.
- Jogiyanto. 2005. *Pengenalan Komputer*. Yogyakarta : Andi.
- Ramadhani, Anis. 2013. *Jurus Rahasia Pintar Menguasai Android Untuk Pemula*. Jakarta : Kir Direction.
- Rh, Sianipar. 2014. *Pemrograman C#*. Bandung : Informatika.
- Roedavan, Rickman. 2014. *Unity Tutorial Game Engine*. Bandung : Informatika.
- TIM. 2011. *Microsoft Visual C# 2010*. Yogyakarta : Andi.
- Tohani, Hamim. 2014. *Analisis serta Perancangan Sistem Informasi melalui Pendekatan UML*. Yogyakarta : Andi.
- Franz, Annafi.,dkk. 2014. *Augmented reality untuk pengenalan satwa pada kebun binatang gembira loka yogyakarta*. Jurnal SCRIPT, (Online), Vol.1, No.2,

L A M P I R A N

```

using UnityEngine;
using System.Collections;
using Vuforia;
public class nada : MonoBehaviour, IVirtualButtonEventHandler
{
    //public Material[] m_TeapotMaterials;
    public AudioSource sentuh1;
    public AudioSource sentuh2;
    public AudioSource sentuh3;
    public AudioSource sentuh4;
    public AudioSource sentuh5;
    public AudioSource sentuh6;
    public AudioSource sentuh7;
    public AudioSource sentuh8;
    public AudioSource sentuh9;
    public AudioSource sentuh10;
    public AudioSource sentuh11;
    public AudioSource sentuh12;
    public AudioSource sentuh13;
    public AudioSource sentuh14;
    public AudioSource sentuh15;
    public AudioSource sentuh16;
    public AudioSource sentuh17;
    public AudioSource sentuh18;
    public AudioSource sentuh19;
    public AudioSource sentuh20;
    public AudioSource sentuh21;
    public AudioSource sentuh22;
    public AudioSource sentuh23;
    public AudioSource sentuh24;
    public AudioSource sentuh25;
    public AudioSource sentuh26;
    public AudioSource sentuh27;
    public AudioSource sentuh28;
    public AudioSource sentuh29;

    // Use this for initialization
    void Start ()
    {
        VirtualButtonBehaviour[] vbs = GetComponentInChildren<VirtualButtonBehaviour> ();
        for (int i=0; i < vbs.Length; i++)

```

```

        {
            vbs[i].RegisterEventHandler(this);
        }
        //whale = transform.FindChild ("while").gameObject;
    }

    public void OnButtonPressed(VirtualButtonAbstractBehavior vb){
        switch (vb.VirtualButtonName) {

            case "nada1":
                sentuh1.Play();
                break;
            case "nada2":
                sentuh2.Play();
                break;
            case "nada3":
                sentuh3.Play();
                break;
            case "nada4":
                sentuh4.Play();
                break;
            case "nada5":
                sentuh5.Play();
                break;
            case "nada6":
                sentuh6.Play();
                break;
            case "nada7":
                sentuh7.Play();
                break;
            case "nada8":
                sentuh8.Play();
                break;
            case "nada9":
                sentuh9.Play();
                break;
            case "nada10":
                sentuh10.Play();
                break;
            case "nada11":
                sentuh11.Play();
                break;
            case "nada12":
                sentuh12.Play();

```

```
        break;
    case "nada13":
        sentuh13.Play();
        break;
    case "nada14":
        sentuh14.Play();
        break;
    case "nada15":
        sentuh15.Play();
        break;
    case "nada16":
        sentuh16.Play();
        break;
    case "nada17":
        sentuh17.Play();
        break;
    case "nada18":
        sentuh18.Play();
        break;
    case "nada19":
        sentuh19.Play();
        break;
    case "nada20":
        sentuh20.Play();
        break;
    case "nada21":
        sentuh21.Play();
        break;
    case "nada22":
        sentuh22.Play();
        break;
    case "nada23":
        sentuh23.Play();
        break;
    case "nada24":
        sentuh24.Play();
        break;
    case "nada25":
        sentuh25.Play();
        break;
    case "nada26":
        sentuh26.Play();
        break;
    case "nada27":
        sentuh27.Play();
```

```

        break;
    case "nada28":
        sentuh28.Play();
        break;
    case "nada29":
        sentuh29.Play();
        break;
    }
}

public void OnButtonReleased(VirtualButtonAbstractBehavior vb){
    switch (vb.VirtualButtonName) {
    case "nada1":
        sentuh1.Stop();
        break;
    case "nada2":
        sentuh2.Stop();
        break;
    case "nada3":
        sentuh3.Stop();
        break;
    case "nada4":
        sentuh4.Stop();
        break;
    case "nada5":
        sentuh5.Stop();
        break;
    case "nada6":
        sentuh6.Stop();
        break;
    case "nada7":
        sentuh7.Stop();
        break;
    case "nada8":
        sentuh8.Stop();
        break;
    case "nada9":
        sentuh9.Stop();
        break;
    case "nada10":
        sentuh10.Stop();
        break;
    }
}

```

```
case "nada11":
    sentuh11.Stop();
    break;
case "nada12":
    sentuh12.Stop();
    break;
case "nada13":
    sentuh13.Stop();
    break;
case "nada14":
    sentuh14.Stop();
    break;
case "nada15":
    sentuh15.Stop();
    break;
case "nada16":
    sentuh16.Stop();
    break;
case "nada17":
    sentuh17.Stop();
    break;
case "nada18":
    sentuh18.Stop();
    break;
case "nada19":
    sentuh19.Stop();
    break;
case "nada20":
    sentuh20.Stop();
    break;
case "nada21":
    sentuh21.Stop();
    break;
case "nada22":
    sentuh22.Stop();
    break;
case "nada23":
    sentuh23.Stop();
    break;
case "nada24":
    sentuh24.Stop();
    break;
case "nada25":
    sentuh25.Stop();
    break;
```

```
case "nada26":  
    sentuh26.Stop();  
    break;  
case "nada27":  
    sentuh27.Stop();  
    break;  
case "nada28":  
    sentuh14.Stop();  
    break;  
case "nada29":  
    sentuh29.Stop();  
    break;
```

```
    }  
}  
}
```