

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : BRYAN ANDI GERRARDO
NRP : 5110100152
DOSEN WALI : Imam Kuswardayan, S.Kom., M.T.
DOSEN PEMBIMBING : 1. Sarwosri, S.Kom., M.T.
2. Ridho Rahman Hariadi, S.Kom., M.Sc.

2. JUDUL TUGAS AKHIR

“Rancang Bangun Aplikasi Virtual Piano dengan Menggunakan Leap Motion Controller”

3. LATAR BELAKANG

Piano ialah alat musik yang dimainkan dengan jari-jemari tangan. Piano adalah salah satu alat musik tertua di dunia karena telah ada sejak abad 18. Piano merupakan alat musik yang menggunakan papan tuts dan bekerja dengan ditekan [1]. Piano merupakan salah satu alat musik paling digemari di dunia, akan tetapi harga sebuah piano relatif sangat mahal di pasaran. Bahkan banyak pelaku bisnis alat musik mengatakan bahwa piano merupakan alat musik termahal. Mempelajari piano seperti les piano sendiri terbilang cukup mahal karena butuh waktu yang cukup lama atau tidak bisa instan [2]. Sehingga banyak kalangan yang tidak dapat menikmati dan merasakan sensasi bermain piano.

Namun saat ini, teknologi telah banyak membantu manusia dalam banyak hal. Sehingga terdapat banyak cara untuk memperkenalkan piano. Seiring berkembangnya teknologi, manusia menemukan berbagai macam terobosan baru. Salah satunya dengan ditemukannya perangkat Leap Motion. Dengan menggunakan Leap Motion,

manusia dapat melakukan interaksi dengan komputer bahkan tanpa menyentuhnya sama sekali. Leap Motion yang mengusung metode baru untuk berinteraksi dengan komputer dapat diterapkan dalam pembuatan aplikasi perangkat lunak. Salah satunya dengan pembuatan piano virtual dengan menggunakan Leap Motion sebagai alat untuk mengoperasikannya dan juga sebagai *input controller* pada program ini. Bentuk Leap Motion dapat dilihat pada Gambar 1.



Gambar 1 Leap Motion

Diharapkan dengan adanya aplikasi ini, akan menambah sensasi dan pengalaman baru dalam bermain piano, dan juga diharapkan masyarakat akan lebih banyak mengetahui tentang alat musik piano serta mendalami dan mempelajari lebih banyak tentang piano. Dengan *input controller* berupa Leap Motion, sensasi dalam bermain piano diharapkan menjadi lebih menyenangkan untuk dimainkan.

Dalam aplikasi yang akan dibuat, pengguna dapat bermain piano secara virtual dengan nada asli sesuai dengan yang terdapat pada piano. Dengan Leap Motion, pengguna dapat lebih merasakan sensasi bermain piano. Leap Motion akan mendeteksi jari pengguna, lalu jari akan ditampilkan dalam bentuk virtual dalam aplikasi dengan pergerakan mengikuti gerak jari pengguna. Ketika jari pengguna melakukan tekanan/ketukan, gambaran jari virtual dalam aplikasi juga akan melakukan tekanan/ketukan, lalu aplikasi akan mengeluarkan suara tuts piano virtual yang ditekan oleh pengguna.

4. RUMUSAN MASALAH

Permasalahan-permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana membuat aplikasi virtual piano yang menyerupai piano asli.
2. Bagaimana memanfaatkan Leap Motion untuk mendeteksi *input* jari (*finger*) untuk mensimulasikan cara bermain piano pada aplikasi virtual piano di PC/laptop.
3. Bagaimana menyatukan masukan dari Leap Motion untuk menerima gerakan pengguna yang akan mempengaruhi hasil *output* program.

5. BATASAN MASALAH

Masalah yang dibahas pada tugas akhir ini dibatasi lingkupnya pada:

1. Aplikasi yang dibuat berupa aplikasi desktop.
2. Lingkungan pengembangan yang digunakan menggunakan bahasa C#.
3. Alat musik piano yang akan disimulasikan mempunyai jumlah tuts yang disesuaikan dengan radius deteksi Leap Motion.

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Membuat aplikasi virtual piano yang menyerupai piano asli.
2. Memanfaatkan Leap Motion untuk mendeteksi *input* jari (*finger*) untuk mensimulasikan cara bermain piano pada aplikasi virtual piano di PC/laptop.
3. Menyatukan masukan dari Leap Motion untuk menerima gerakan pengguna yang akan mempengaruhi hasil *output* program.

7. MANFAAT TUGAS AKHIR

Manfaat yang didapat dengan dikerjakannya tugas akhir ini antara lain:

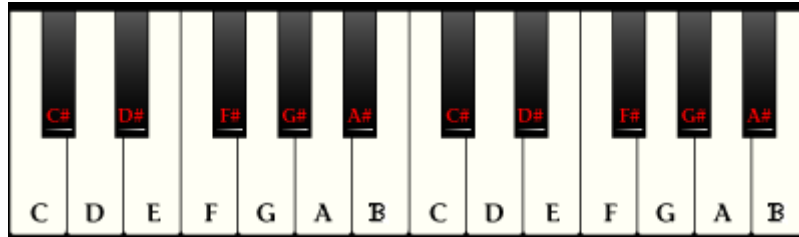
1. Bagi *user*: dapat mempelajari dan mensimulasikan alat musik piano yang lebih interaktif dan lebih murah dalam segi biaya, serta memberi sensasi dan pengalaman baru dalam bermain piano.
2. Bagi pengembang: dapat menjadi pioner pengembangan aplikasi lainnya yang menggunakan teknologi Leap Motion.

8. TINJAUAN PUSTAKA

1. Piano

Piano ialah alat musik yang dimainkan dengan jari-jemari tangan. Piano adalah salah satu alat musik tertua di dunia karena telah ada sejak abad 18 [1]. Piano merupakan alat musik yang menggunakan papan tuts dan bekerja dengan ditekan [1]. Bentuk piano dapat dilihat pada Gambar 2.

Asal mula kata piano sebenarnya berasal dari bahasa Italia, yaitu *pianoforte*. Piano itu sendiri dibuat oleh Bartolomeo Cristofori pada tahun 1720-an. Awal mula piano diciptakan, suaranya tidak sekeras piano yang dapat didengar pada abad 20-an. Pasalnya, tuts piano kala itu tidaklah sekuat piano yang sekarang. Piano juga bukan alat musik pertama yang menggunakan papan tuts dan bekerja dengan dipukul. Alat musik berprinsip kerja mirip piano telah ada sejak 1440 [2].



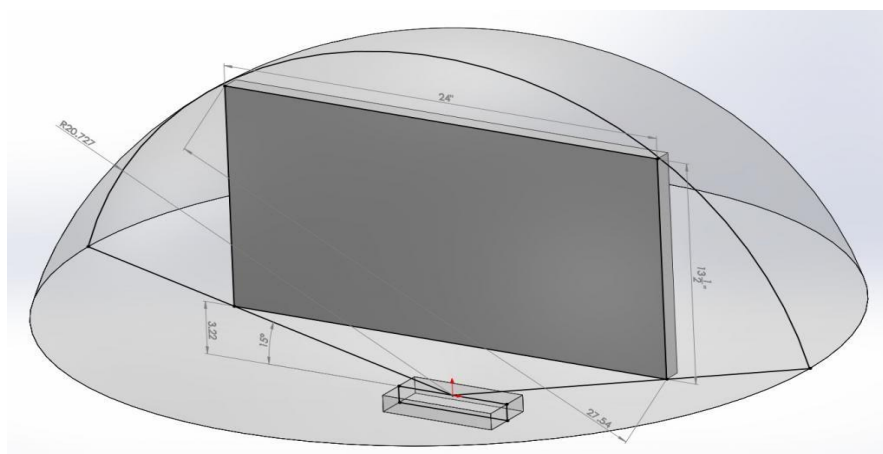
Gambar 2 Tuts Piano

Dalam perkembangannya sebelum memiliki 88 tuts seperti sekarang, piano memiliki lima oktaf dan 62 tuts. Ia juga dilengkapi dengan pedal. Semula pedal itu digerakkan dengan lutut. Namun, kemudian pedal kaki yang diperkenalkan di Inggris menjadi populer hingga sekarang. Dan saat ini piano merupakan salah satu alat musik yang paling digemari di dunia dan merupakan alat musik termahal di dunia [1].

2. Leap Motion

Leap Motion adalah sebuah perangkat keras pendeteksi gerakan yang diproduksi oleh Leap Motion, Inc. Alat ini telah dikembangkan sejak tahun 2010. Alat ini mendapat respon positif pada dunia teknologi dikarenakan mampu memberikan pengalaman baru kepada konsumen dalam mengendalikan komputer pribadi dengan komponen sederhana dan harga yang cukup terjangkau.

Leap Motion sendiri terdiri dari 2 buah *monochromatic* kamera CCD (*Charge-Coupled Device*) yang didukung dengan 3 buah lampu LED pemancar inframerah di atas sebuah *board* standar [3]. Cara kerjanya sendiri ialah sinar inframerah yang dipancarkan oleh 3 lampu LED (*Light-Emitting Diode*) menyebar secara konvergen sehingga mampu mendeteksi bagian yang lebih luas [3]. Bila ada objek yang melewati atau memasuki daerah tadi, pantulan tersebutlah yang akan ditangkap oleh dua buah kamera yang kemudian akan dikirimkan ke komputer untuk diproses [3]. Gambar 3 memperlihatkan jarak deteksi dari Leap Motion.



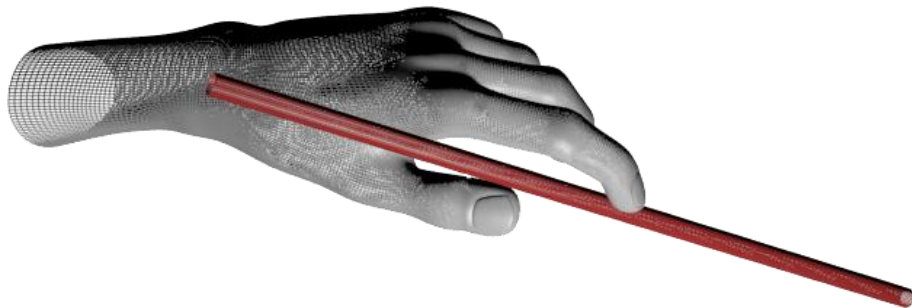
Gambar 3 Jarak Deteksi Leap Motion

3. Leap Motion SDK

Sebagai perangkat keras inovatif yang terjun dalam teknologi 3D *sense touchles*, dibutuhkan dukungan penggunaan dari para pengembang aplikasi. Untuk itulah Leap Motion, Inc selaku pengembang mengeluarkan Leap SDK yang dapat diunduh secara mudah dan gratis. Leap SDK sendiri memberikan fitur-fitur dan kemudahan kepada pengembang untuk mengembangkan aplikasinya, seperti bahasa yang digunakan disediakan untuk banyak bahasa diantaranya C#, Objective-C, Unity, C++, Java, dll [4].

4. Pointables

Pointables adalah objek yang terdeteksi oleh Leap Motion yang memiliki ujung sehingga bisa digunakan untuk menunjuk atau melakukan gerak isyarat tertentu seperti *key-tap*, *screen-tap*, *pointing*, *rotating*, *scaling*, *swipe*, dll. *Pointables* sendiri dapat dipecah menjadi dua objek berbeda, yaitu *tools* dan *fingers* [5].



Gambar 4 Representasi Objek Alat/Tools

Sebuah objek terdeteksi *tools* bila objek lurus, lebih kecil, dan tidak bengkok [6]. Gambar 4 merupakan gambaran dari representasi objek *tools*. Sedangkan objek terdeteksi *fingers* bila objek tidak lurus, mempunyai sendi kecil, lebih besar, dan lebih pendek [6]. Gambar 5 merupakan gambaran dari representasi objek *fingers*.



Gambar 5 Representasi Objek Finger

5. XNA Framework

Microsoft XNA adalah sebuah *platform* buatan Microsoft yang mengatur *runtime* dari permainan. *Platform* ini bekerja di atas *framework* .NET dan dapat berjalan di atas *platform* Microsoft Windows, XBOX, dan Windows Phone 7 [7].

Framework XNA dirancang untuk serupa dengan *framework* .NET dalam hal pola-pola rancang dan idiom. Dengan XNA Game Studio, pengembang mampu menggunakan fungsi-fungsi baik dari *framework* XNA maupun *framework* .NET pada aplikasi. *Framework* XNA baik digunakan untuk lebih kepada hal-hal yang bersifat aplikasi video game seperti mencetak grafis dan menangani *input* [7].

6. Unity

Unity merupakan sebuah ekosistem pengembangan permainan dengan mesin *rendering* yang kuat dan terintegrasi, serta disuplai dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat muatan 3D interaktif. Unity juga dilengkapi penerbitan *multiplatform* dengan mudah serta menyediakan ribuan aset berkualitas yang siap pakai [8].

Unity menyediakan banyak fungsionalitas untuk membuat permainan dan muatan 3D lainnya. Unity dapat digunakan untuk merangkai aset yang dibuat ke dalam lingkungan, menambahkan pencahayaan, suara, efek, fisika, dan animasi. Unity juga dapat dilakukan secara bersamaan untuk mengetes maupun mengubah permainan yang dibuat. Dan ketika sudah siap, permainan dapat dipublikasikan ke berbagai macam *platform* seperti Mac, PC, Linux, Windows Store, iOS, Android, Windows Phone 8, Blackberry 10, Wii U, PS3 dan Xbox 360 dengan 3 bahasa pemrograman yang dapat digunakan dalam *scripting*-nya yaitu C#, JavaScript dan Boo [9].

9. RINGKASAN ISI TUGAS AKHIR

Dalam tugas akhir ini akan dibuat sebuah sistem aplikasi virtual piano dengan menggunakan Leap Motion *controller*. Diagram kasus penggunaan dalam program ini dapat dilihat pada Gambar 6.



Gambar 6 Diagram Kasus Penggunaan Aplikasi

Gambar 6 menjelaskan bahwa dalam aplikasi yang akan dibuat, pengguna dapat bermain piano secara virtual dengan nada asli sesuai dengan yang terdapat pada piano. Tidak hanya nada yang sesuai dengan nada piano sebenarnya, akan tetapi tekanan tuts oleh pengguna juga diperhitungkan dalam aplikasi ini. Aplikasi virtual piano ini dapat melakukan pendeteksian durasi waktu tekanan pengguna pada tuts yang ditekan. Jika tuts yang ditekan lama, maka nada yang dikeluarkan juga mempunyai durasi yang sama lamanya dengan tekanan pengguna pada tuts. Tentunya hal ini memberikan nilai lebih dalam segi *reliability* dari aplikasi ini. Deskripsi kasus penggunaan yang ada dalam sistem dapat dilihat pada Tabel 1.

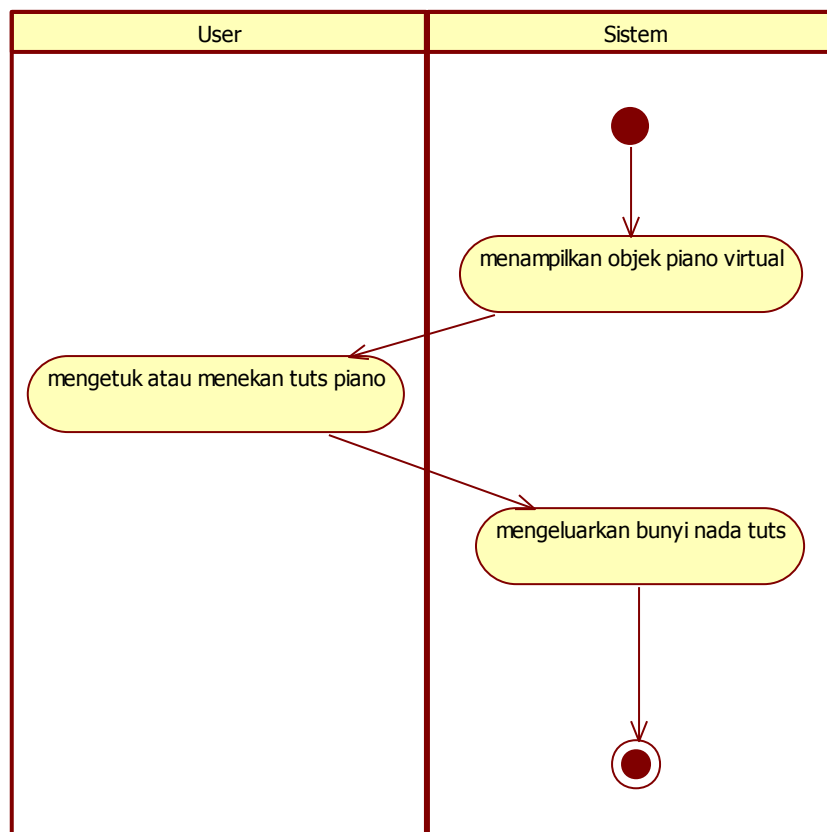
Tabel 1 Deskripsi Kasus Penggunaan

No	Kode Kasus Pengguna	Nama Kasus Penggunaan	Deskripsi
1	UC-001	Memainkan alat musik	Pengguna dapat memainkan alat musik dari piano yang tersedia dalam aplikasi

Selain deskripsi kasus penggunaan, disertakan pula diagram aktivitas yang merupakan ilustrasi alur penggunaan aplikasi pada Gambar 7. Diagram aktivitas beserta spesifikasi dari kasus penggunaan sebelumnya dapat dilihat pada Tabel 2.

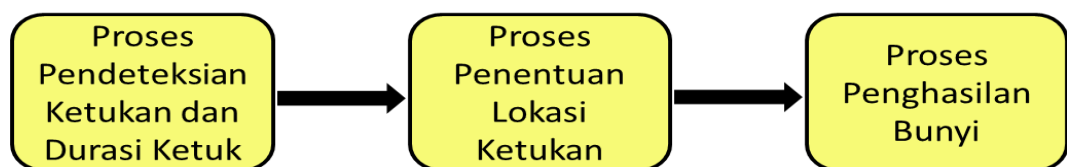
Tabel 2 Spesifikasi Kasus Penggunaan

Aktor	Pengguna aplikasi / User
Kondisi Awal	Pengguna berada pada layar permainan
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan objek piano virtual pada layar. 2. Pengguna melakukan <i>input</i> mengetuk atau menekan salah satu tuts pada alat bantu visual. 3. Sistem merespon dengan mengeluarkan bunyi nada sesuai dengan tuts yang diketuk atau ditekan dengan durasi waktu yang sesuai ketukan/tekanan pengguna.
Alur Alternatif	-
Kondisi Akhir	-



Gambar 7 Diagram Aktivitas Memainkan Alat Musik

Aplikasi ini mempunyai beberapa proses yang penting agar unsur fungsionalitas aplikasi dapat berjalan dengan baik dan benar. Proses-proses tersebut antara lain proses pendeteksian ketukan, proses penentuan lokasi ketukan, dan proses mengeluarkan bunyi, seperti digambarkan pada Gambar 8.



Gambar 8 Blok Proses Aplikasi

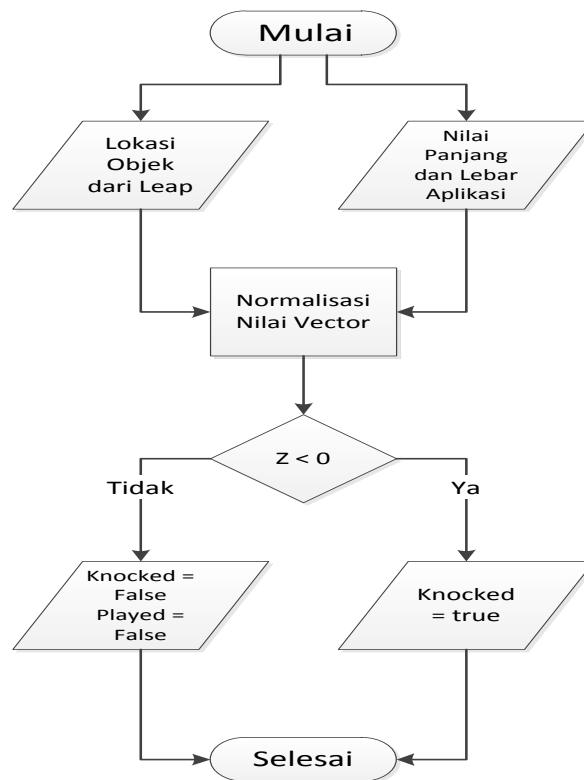
Setiap proses pada Gambar 8 mempunyai detail yang berbeda. Berikut penjelasan detail dari tiap proses.

1. Proses Pendeteksian Ketukan dan Durasi Ketuk

Proses pendeteksian ketukan adalah salah satu proses utama yang berjalan dalam memainkan alat musik piano virtual. Proses ini mendeteksi jari pengguna apakah telah melakukan proses ketukan yang dibutuhkan oleh aplikasi. Proses ini juga

mendeteksi mengenai lamanya tuts yang ditekan oleh pengguna, yang mempengaruhi lamanya durasi waktu dari nada tuts yang dimainkan.

Untuk melakukan pendeteksian ketukan, aplikasi mendapatkan data objek dari hasil tracking Leap *controller*. Leap nantinya akan mendeteksi jari pengguna apakah telah melewati nilai tertentu sehingga dapat mendeteksi sebagai sebagai sebuah ketukan atau tidak. Variabel Z merupakan variabel parameter untuk objek apakah objek tersebut melakukan ketukan atau tidak, dan juga sekaligus menjadi parameter apakah tuts masih ditekan oleh pengguna atau telah dilepas yang nantinya akan mempengaruhi lamanya durasi waktu *output* nada/suara. sebagaimana ditunjukkan pada Gambar 9.

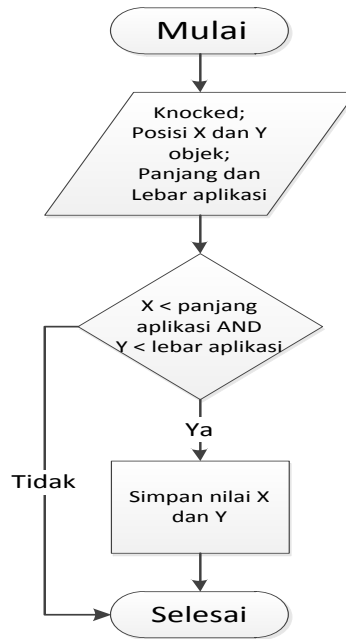


Gambar 9 Diagram Alir Proses Pendeteksian Ketukan dan Durasi Ketuk

2. Proses Penentuan Lokasi Ketukan

Proses selanjutnya yang berhubungan dengan pendeteksian ketukan adalah mendeteksi lokasi ketukan. Proses ini sangat penting karena lokasi ketukan akan mempengaruhi *output* bunyi pada proses selanjutnya.

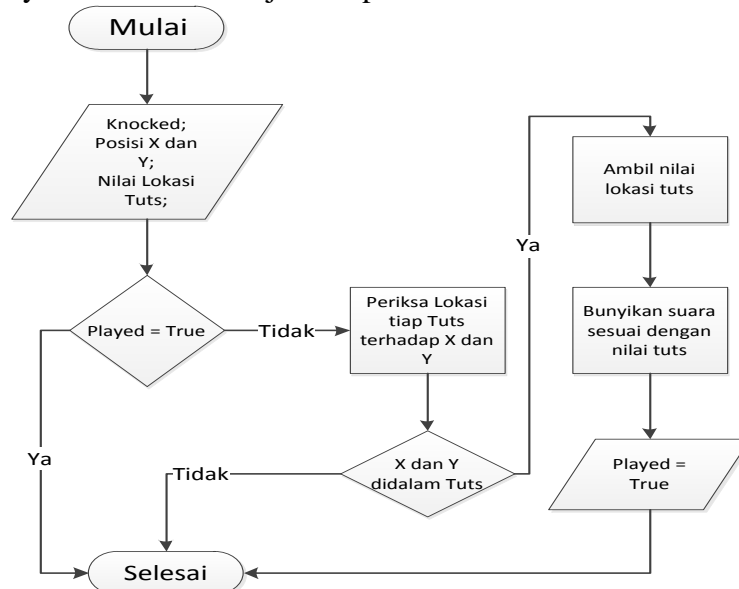
Bila jari pengguna terdeteksi melakukan ketukan maka saat itu disimpan datanya termasuk koordinat x dan y . Koordinat ini akan disimpan untuk dinormalisasi untuk menentukan tuts piano mana yang diketuk oleh pengguna. Proses ini ditunjukkan pada Gambar 10.



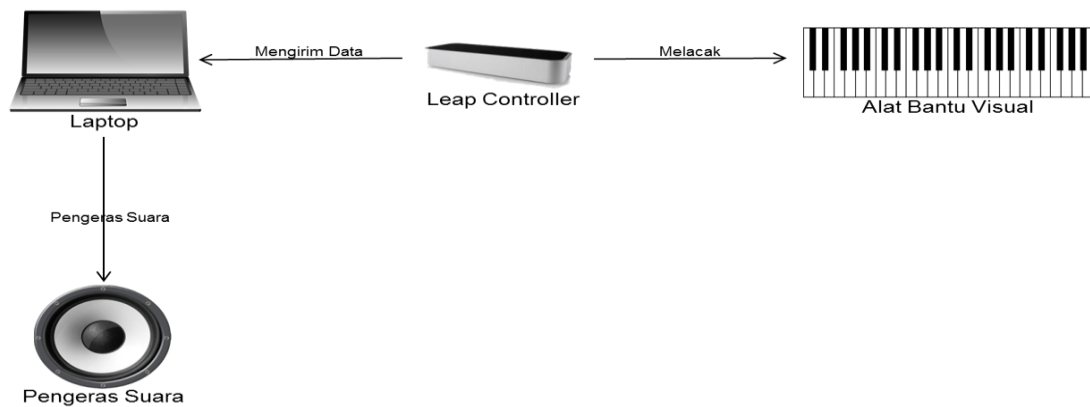
Gambar 10 Diagram Alir Proses Penentuan Lokasi Ketukan

3. Proses Penghasilan Bunyi

Proses penghasilan bunyi mempunyai peran yang penting dalam jalannya proses aplikasi ini, karena proses ini memberikan hasil *output* kepada pengguna bahwa ketukan jari yang dilakukan telah berhasil dan diterima dengan baik oleh sistem. *Output* yang dihasilkan adalah *output* bunyi yang disesuaikan dengan tuts piano virtual yang diketuk oleh jari pengguna. Karenanya proses penghasilan bunyi ini sangat bergantung erat dengan posisi ketukan yang terjadi pada proses sebelumnya. Proses ini ditunjukkan pada Gambar 11.



Gambar 11 Diagram Alir Proses Penghasilan Bunyi



Gambar 12 Arsitektur Sistem

Gambar 12 menjelaskan mengenai arsitektur sistem dari aplikasi yang akan dibuat dimana Leap *controller* akan melacak posisi *pointables* berupa *finger* atau *tool* pada tuts dalam alat bantu visual. Dari situ posisi *pointables* yang berhasil dilacak akan dikirim sebagai data ke aplikasi simulasi virtual piano di laptop atau PC. Dari Aplikasi tersebut maka hasil *output* berupa suara tuts yang dipilih akan dikeluarkan melalui pengeras suara.

10. METODOLOGI

a. Penyusunan proposal tugas akhir

Proposal ini berisi tentang rancang bangun aplikasi virtual piano yang dioperasikan menggunakan Leap Motion sebagai *input* utamanya. Leap Motion adalah teknologi baru yang memungkinkan interaksi dengan komputer dilakukan dengan menggerakkan tangan dan jari. Tugas akhir ini bertujuan untuk mengeksplorasi teknologi tersebut dengan menciptakan sebuah media interaktif berupa aplikasi virtual piano yang digunakan sebagai media simulasi dan pembelajaran bermain piano. Integrasi Leap Motion dengan kontrol gerakan jari yang natural diharapkan dapat mendukung memberikan sensasi dan pengalaman baru dalam bermain dan mensimulasikan piano.

b. Studi literatur

Pada tahap ini dilakukan pencarian informasi dan literatur yang diperlukan dalam proses perancangan dan implementasi sistem yang akan dibangun. Literatur yang digunakan adalah sebagai berikut:

1. Pengoperasian Leap Motion.
2. Pengintegrasian *input* jari menggunakan pustaka Leap Motion SDK.
3. Teknik pemrograman untuk pengembangan aplikasi menggunakan Unity dengan bahasa pemrograman C#.

4. Teknik pemrograman *framework* XNA untuk pengembangan aplikasi.

c. Analisis dan desain perangkat lunak

Pada tahap ini dilakukan analisa awal dan pendefinisian kebutuhan sistem. Selanjutnya, dirumuskan rancangan sistem yang nantinya digunakan untuk membangun aplikasi. Perincian dalam tahap ini adalah sebagai berikut:

1. Perancangan aturan nada piano.
2. Perancangan diagram kasus penggunaan, diagram aktivitas, blok-blok proses beserta diagram alir proses yang merupakan analisis awal kebutuhan pada aplikasi yang akan dibangun.
3. Perancangan antarmuka aplikasi.
4. Perancangan integrasi Leap Motion dengan Unity pada aplikasi.
5. Perancangan integrasi Leap Motion dengan *framework* XNA pada aplikasi.

d. Implementasi perangkat lunak

Pada tahap ini desain perangkat lunak diwujudkan ke dalam bentuk program dari rancangan yang telah dibuat sebelumnya. Perincian tahap ini adalah sebagai berikut:

1. Implementasi pemodelan objek dengan tekstur, efek, suara dan animasi menggunakan Unity dengan bahasa pemrograman C#.
2. Implementasi pemodelan objek dengan tekstur, efek, suara dan animasi pada *framework* XNA.
3. Implementasi pustaka Leap Motion SDK untuk kontrol simulasi gerakan jari.

e. Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang dibuat. Tujuan uji coba perangkat lunak adalah untuk menemukan kesalahan-kesalahan (*bugs*) dalam pengembangan aplikasi secara langsung agar dapat diperbaiki sesegera mungkin. Pengujian dilakukan untuk menguji fungsionalitas secara keseluruhan berjalan sesuai keinginan atau tidak.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah

- c. Batasan Tugas Akhir
- d. Tujuan
- e. Metodologi
- f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Jadwal kegiatan tugas akhir ini dibuat dalam 6 tahapan dan dengan jangka waktu pengerjaan yang bervariasi. Tahapan-tahapan dan waktu pengerjaan lebih detail dapat dilihat pada Tabel 3.

Tabel 3 Jadwal Pengerjaan

Tahapan	2014																	
	Februari				Maret				April				Mei				Juni	
Penyusunan Proposal																		
Studi Literatur																		
Perancangan sistem																		
Implementasi																		
Pengujian dan evaluasi																		
Penyusunan buku																		

12. DAFTAR PUSTAKA

- [1] Wikipedia, "Piano," wikipedia.org, 2013. [Online]. Available: <http://id.wikipedia.org/wiki/Piano>. [Accessed 26 February 2014].
- [2] Milka, t.s, "Sejarah Alat Musik Piano," 2012. [Online]. Available: <http://trixiepiano.blogspot.com>. [Accessed 26 February 2013].
- [3] Wikipedia, "Leap Motion," wikipedia.org, 2014. [Online]. Available: <http://id.wikipedia.org/wiki/Piano>. [Accessed 26 February 2014].
- [4] L. Motion, "Leap Motion," Leap Motion, 2013. [Online]. Available: <https://developer.leapmotion.com/docs>. [Accessed 26 February 2014].

- [5] L. Motion, "Tracking Hands, Fingers, and Tools," Leap Motion, 2013. [Online]. Available: https://developer.leapmotion.com/documentation/python/devguide/Leap_Tracking.html. [Accessed 26 February 2014].
- [6] L. Motion, "API Overview," Leap Motion, 2013. [Online]. Available: https://developer.leapmotion.com/documentation/java/devguide/Leap_Overview.html. [Accessed 26 February 2014].
- [7] Kamal, m, "Mengenai XNA," 2012. [Online]. Available: <http://students.netindonesia.net/blogs/mustofakamal/archive/2012/02/17/mengenai-xna.aspx>. [Accessed 26 February 2014].
- [8] Unity, "Unity - Game engine, tools and multiplatform," Unity3D, 2013. [Online]. Available: <http://unity3d.com/unity/>. [Accessed 26 February 2014].
- [9] Unity, "Unity - Create Games with Unity," Unity3D, 2013. [Online]. Available: <http://unity3d.com/pages/create-games>. [Accessed 26 February 2014].