

PERANCANGAN KENDALI KEYBOARD VIRTUAL HEMOCS PADA PERANGKAT BERBASIS iOS DENGAN TEKNIK FREE MOVEMENT

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Firdaus Dwika Ainun Ilmi

NIM: 125150200111024



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016**

PENGESAHAN

PERANCANGAN KENDALI KEYBOARD VIRTUAL HEMOCS PADA PERANGKAT
BERBASIS iOS DENGAN TEKNIK FREE MOVEMENT

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Firdaus Dwika Ainun Ilmi
NIM: 125150200111024

Skripsi ini telah diuji dan dinyatakan lulus pada
24 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dr. Eng. Herman Tolle, S.T, M.T
NIP: 19740823 200012 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

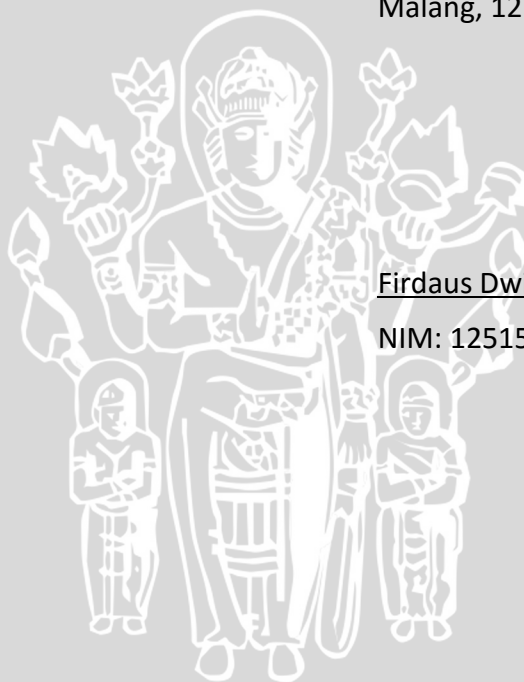
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Agustus 2016

Firdaus Dwika Ainun Ilmi

NIM: 125150200111024



KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Perancangan Kendali *Keyboard Virtual* Hemocs pada Perangkat Berbasis iOS dengan Teknik *Free Movement*” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Dr. Eng. Herman Tolle, S.T, M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Bapak Imam Cholissodin, S.Si., M.Kom. selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
3. Bapak Firman Abadi dan Ibu Hidayati Ningrum dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
4. Brilliant Chindy Maharani, Seluruh teman-teman di Fakultas Ilmu Komputer, rekan-rekan di Lembaga Pers Mahasiswa Fakultas Ilmu Komputer, rekan-rekan Student Employee Laboratorium Pemrograman Aplikasi Perangkat Bergerak dan kawan-kawan Kelas A Informatika 2012 atas bantuan dan dukungan sehingga terselesaikan skripsi ini.
5. Seluruh civitas academica Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 12 Agustus 2016

Penulis

firdausdwika@gmail.com

ABSTRAK

Firdaus Dwika Ainun Ilmi. 2016. Perancangan Kendali Keyboard Virtual Hemocs Pada Perangkat Berbasis iOS Dengan Teknik Free Movement. Skripsi Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing : Herman Tolle, Dr. Eng., S.T., M.T.

Teknologi interaksi manusia dan komputer khususnya *virtual reality* dan *augmented reality* menunjukkan perkembangan yang cukup pesat. Teknologi ini dapat dimanfaatkan dalam membantu penyandang disabilitas atau keterbatasan fisik dalam melakukan aktivitas layaknya manusia normal pada umumnya. Hal ini ditunjang dengan banyaknya perangkat bergerak yang dilengkapi dengan kamera serta sensor-sensor internal. Dalam penelitian ini dibuatlah sebuah kendali *keyboard virtual* dengan kontrol pergerakan kepala, yang memanfaatkan sensor-sensor pada perangkat bergerak seperti *accelerometer*, *gyroscope*, dan *magnetometer*. Nantinya pengguna hanya cukup dengan menggerakkan kepala untuk berinteraksi. Pergerakan kepala dilakukan secara bebas seperti pergerakan pada pointer, pergerakan kepala secara bebas ini dikenal dengan *free movement*. fungsi dari pergerakan ini yaitu memudahkan pengguna untuk bergerak kesegala arah saat melakukan pengetikan sebuah kalimat yang nantinya menjadi dasar dari pengujian akurasi. Pengujian akurasi dilakukan dengan membandingkan 3 ukuran keyboard, skala 0,3 cm, 0,4 cm dan 0,5 cm dihitung dari rata-rata jumlah error dan kecepatan pengetikan kalimat. keyboard dengan skala 0,3cm memiliki persentase paling tinggi dibanding yang lainnya.

Kata Kunci : *virtual reality*, *augmented reality*, disabilitas, pergerakan kepala, *keyboard virtual*, *free movement*.



ABSTRACT

Firdaus Dwika Ainun Ilmi. 2016. Perancangan Kendali Keyboard Virtual Hemocs Pada Perangkat Berbasis iOS Dengan Teknik Free Movement. Skripsi Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing : Herman Tolle, Dr. Eng., S.T., M.T.

Human computer interaction technology, especially virtual reality and augmented reality show fairly rapid growth. This technology can be utilized in helping persons with disabilities or physical limitations in performing normal activities of the human beings in general. It is supported by many mobile device equipped with a camera and internal sensors. In this study made a virtual keyboard to control the movement of the head, which utilizes sensors on mobile devices such as an accelerometer, gyroscope and magnetometer. Later, the user just simply by moving his head to interact. The movement of the head is conducted freely as the movement of the pointer, the head moves freely is known to free movement. the function of this movement that is easier for users to move in all directions when typing a phrase that later became the basis of testing accuracy. Accuracy testing is done by comparing 3 keyboard size, scale 0,3cm, 0,4cm and 0,5cm calculated from the average number of errors and speed typing sentences. keyboard with 0,3cm scale has the highest percentage compared to the other.

Keywords : virtual reality, augmented reality, disability, head movement, virtual keyboard, free movement.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	lii
KATA PENGANTAR.....	lv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	ix
DAFTAR KODE.....	x
DAFTAR LAMPIRAN	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Manfaat	2
1.5 Batasan Masalah	2
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Head Movement Control System.....	4
2.2.1 Kajian Pustaka	4
2.2 Sensor Pada iOS.....	5
2.2.1 Accelerometer	5
2.2.2 Gyroscope	6
2.2.3 Magnetometer.....	6
2.2.4 Core Motion	7
BAB 3 METODOLOGI	8
3.1 Studi Literatur	8
3.2 Perancangan Kendali Keyboard Virtual Hemocs.....	9
3.2.1 Gambaran Umum Sistem	9
3.2.2 Merancang Kendali Keyboard Virtual	9
3.3 Implementasi dan Pengujian.....	9

3.4 Kesimpulan dan Saran	9
BAB 4 PERANCANGAN KENDALI KEYBOARD VIRTUAL HEMOCS DENGAN TEKNIK FREE MOVEMENT	10
4.1 Gambaran Umum Sistem	10
4.2 Merancang Kendali Keyboard Virtual dengan Teknik Free Movement	11
4.3 Algoritma	14
4.3.1 Flowchart	15
BAB 5 IMPLEMENTASI DAN PENGUJIAN	16
5.1 Implementasi	16
5.1.1 Batasan-batasan Implementasi	16
5.1.2 Implementasi Kode Program	16
5.1.2.1 Fungsi Pembuatan Pointer	16
5.1.2.2 Fungsi Kendali Hemocs	18
5.1.2.3 Fungsi Pengecekan Pointer	20
5.1.2.4 Fungsi Penulisan Pada Textfield	21
5.1.3 Implementasi Antarmuka	22
5.2 Spesifikasi Perangkat Keras	22
5.3 Spesifikasi Perangkat Lunak	23
5.4 Pengujian	23
5.4.1 Pengujian Akurasi	24
5.4.2 Pengujian Usability	27
5.4.2.1 Hasil Pengujian Usability	28
5.4.2.2 Analisis Hasil Pengujian	28
BAB 6 KESIMPULAN DAN SARAN	30
6.1 Kesimpulan	30
6.2 Saran	30
DAFTAR PUSTAKA	31

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Sistem 3D Image Display	4
Gambar 2.2 Posisi Ponsel Untuk Game 3D Virtual Reality	5
Gambar 2.3 Menentukan Heading Data (yaw) menggunakan Magnetometer	6
Gambar 3.1 Diagram Alur Metodologi Penelitian.....	8
Gambar 4.1 Gambaran Umum Sistem	10
Gambar 4.2 Ilustrasi Teknik Free Movement.....	11
Gambar 4.3 Jenis Pergerakan Kepala.....	11
Gambar 4.4 Layout Keyboard	12
Gambar 4.5 Titik Awal Pointer	13
Gambar 4.6 Contoh Pemilihan Karakter Huruf B.....	13
Gambar 4.7 Variasi Percobaan Waktu Berdasarkan Kecepatan.....	15
Gambar 4.8 Variasi Percobaan Waktu Berdasarkan Jumlah Error	13
Gambar 4.9 Flowchart Perancangan Teknik Free Movement	15
Gambar 5.1 Antarmuka Kendali Keyboard Virtual	22
Gambar 5.2 Ukuran Huruf Skala Keyboard 0,3 cm	24
Gambar 5.3 Ukuran Huruf Skala Keyboard 0,4 cm	24
Gambar 5.4 Ukuran Huruf Skala Keyboard 0,5 cm	25
Gambar 5.5 Hasil Pengujian Akurasi Berdasarkan Waktu	26
Gambar 5.6 Hasil Pengujian Akurasi Berdasarkan Error.....	26
Gambar 5.7 Hasil Pilihan Ukuran Keyboard	29

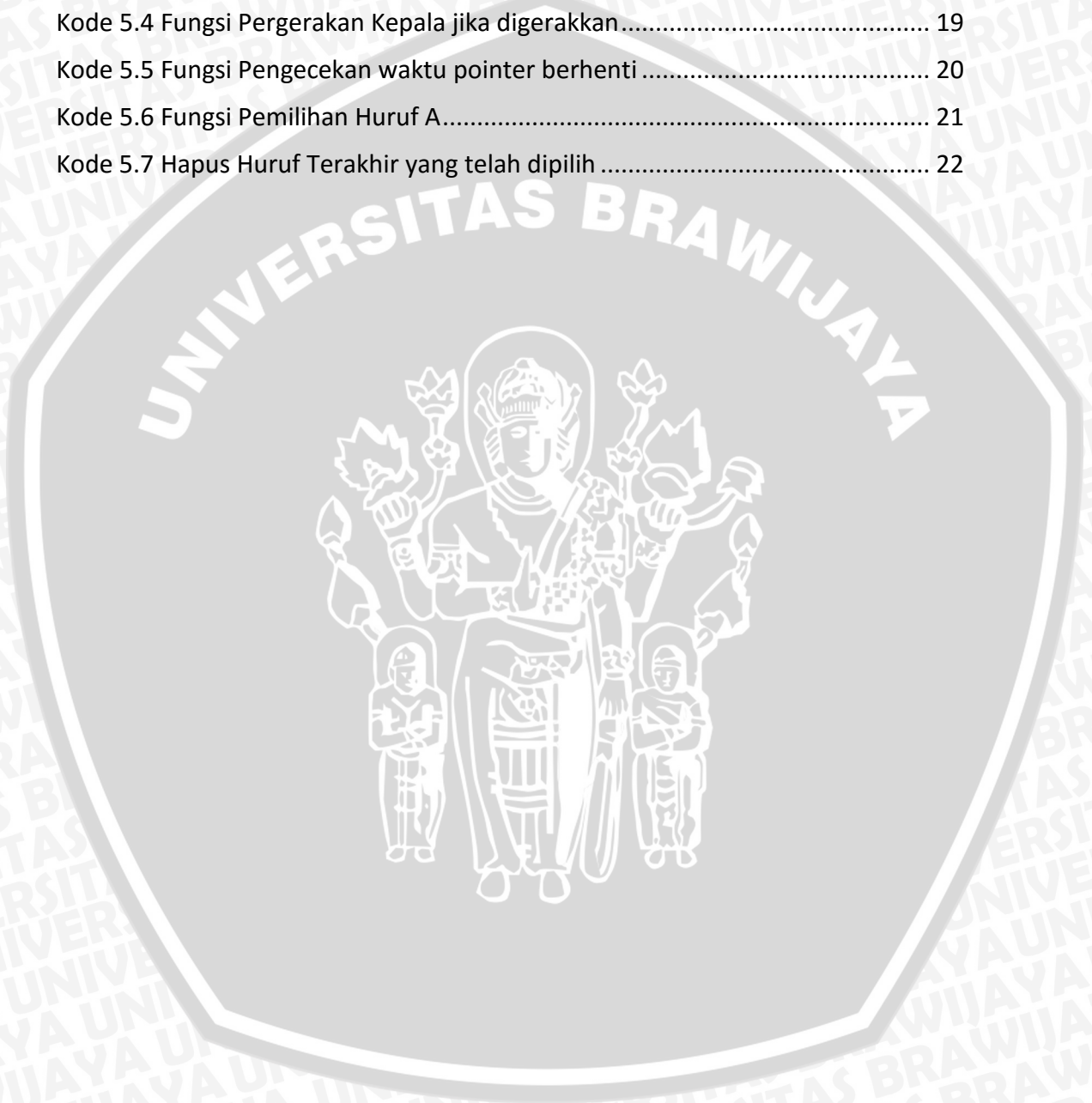
DAFTAR TABEL

Tabel 4.1 Percobaan Pointer Harus Berhenti.....	15
Tabel 5.1 Spesifikasi Perangkat Keras Komputer	22
Tabel 5.2 Spesifikasi Perangkat Keras Perangkat Bergerak	23
Tabel 5.3 Spesifikasi Perangkat Lunak Komputer	23
Tabel 5.4 Spesifikasi Perangkat Lunak Perangkat Bergerak.....	23
Tabel 5.5 Hasil Pengujian dengan Kalimat	25
Tabel 5.6 Hasil Pengujian Akurasi	27
Tabel 5.7 Daftar Pertanyaan Pengujian Usability	27
Tabel 5.8 Hasil Kuisisioner Pengujian Usability	28



DAFTAR KODE

Kode 5.1 Fungsi Inisialisasi Parameter dalam pembuatan Pointer	16
Kode 5.2 Fungsi Perhitungan Pembuatan Pointer	17
Kode 5.3 Inisialisasi Fungsi Pergerakan Kepala	18
Kode 5.4 Fungsi Pergerakan Kepala jika digerakkan	19
Kode 5.5 Fungsi Pengecekan waktu pointer berhenti	20
Kode 5.6 Fungsi Pemilihan Huruf A	21
Kode 5.7 Hapus Huruf Terakhir yang telah dipilih	22



DAFTAR LAMPIRAN

Lampiran A Data Kuisisioner Pengujian Usability.....	33
--	----





BAB 1 PENDAHULUAN

1.1 Latar belakang

Banyak inovasi baru dibidang *Human Computer Interaction*, dimana banyak perusahaan besar yang sedang berlomba-lomba mengembangkan berbagai macam *wearable devices* (The Daily Oktagon, 2016). Seperti contoh Google dengan Google Glassnya, Apple dengan Apple Watchnya serta Samsung dengan Samsung Gear Fitnya. Namun, semua itu tidak terlepas dari namanya *Wearable Computing*. *Wearable Computing* merupakan sebuah kajian atau latihan untuk menciptakan, mendesain, membangun atau menggunakan komputasi miniatur yang terdapat ditubuh dan alat sensor (Mann, 1996a).

Wearable Computing yang populer saat ini adalah *Google Cardboard*, yang mana teknologi dari google ini memanfaatkan *virtual reality* dan *augmented reality*. *Augmented reality* adalah penggabungan benda-benda nyata dan maya dilingkungan nyata, berjalan secara interaktif dalam waktu nyata dan terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata (Azuma, 1997). Namun untuk saat ini pemanfaatan *Google Cardboard* hanya sebatas untuk *virtual reality* atau *augmented reality*. Dengan adanya keterbatasan itu diperlukan pengembangan-pengembangan lain yang bermanfaat ditunjang dengan banyaknya masyarakat yang memiliki perangkat bergerak yang dilengkapi dengan kamera dan sensor-sensor internal.

Pada perangkat bergerak saat ini kebanyakan dilengkapi sensor-sensor internal antara lain seperti *accelerometer*, *gyroscope*, *compass* dan lain sebagainya yang digunakan untuk mendeteksi pergerakan yang dilakukan oleh perangkat tersebut. Dengan adanya sensor-sensor tersebut nantinya saat dikombinasikan dengan *Cardboard* ataupun perangkat yang mendukung *virtual reality* lainnya dapat menjadi sebuah kontrol atau kendali dengan memanfaatkan sensor-sensor pada perangkat bergerak untuk digunakan dalam pengembangan teknologi HCI pada masa mendatang. Misalnya untuk membantu para penyandang disabilitas dalam melakukan aktivitas seperti *chatting* dan *messaging* diperlukan sebuah kendali *keyboard virtual* menerapkan *head movement control system* yang memanfaatkan perangkat bergerak dimana pengguna dapat melakukan interaksi dengan sistem cukup dengan menggerakkan kepala.

Diperlukan sebuah penelitian untuk melakukan pengembangan metode *head movement controller system* dengan teknik *free movement*. Yang nantinya menggantikan fungsi jari dalam melakukan aktivitas dalam sebuah aplikasi seperti menekan tombol. Teknik *free movement* sendiri merupakan sebuah metode dimana pergerakan yang dilakukan menggunakan kepala sebagai kendali yang dapat bergerak secara bebas. Berbeda dengan metode sebelumnya yang memanfaatkan pergerakan kepala dengan pergerakan linear. Dalam pergerakan linear pengguna tidak bisa bergerak secara bebas, pengguna hanya bisa melakukan pergerakan secara vertical ataupun horizontal.

Dengan adanya perancangan kendali *keyboard virtual* hemocs dengan teknik *free movement* diharapkan mempermudah para penyandang disabilitas dalam melakukan komunikasi dengan orang normal dan untuk pengembangan lebih lanjut teknik *free movement* ini tidak hanya diimplementasikan untuk kendali *keyboard virtual* saja dan kendali *keyboard virtual* hemocs ini dapat dikembangkan menjadi aplikasi yang efektif dalam mempermudah manusia dalam kegiatan sehari-hari.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana rancangan kendali *keyboard virtual* hemocs dengan teknik *free movement*?
2. Bagaimana implementasi kendali *keyboard virtual* hemocs dengan teknik *free movement*?
3. Bagaimana performansi hasil implementasi kendali *keyboard virtual* hemocs dengan teknik *free movement* pada aplikasi dilihat dari akurasi dan usability?

1.3 Tujuan

Merancang dan mengimplementasikan kendali *keyboard virtual* hemocs dengan teknik *free movement* pada aplikasi berbasis iOS.

1.4 Manfaat

Manfaat dari penelitian ini adalah untuk mempermudah pengguna terutama kaum disabilitas dalam melakukan aktivitas yang menggunakan *keyboard virtual*, serta membantu game yang mengimplementasikan *virtual reality* dan dapat dimanfaatkan dalam bidang *virtual reality*, serta bidang teknologi yang terkait dengan *Human Computer Interaction*.

1.5 Batasan masalah

Agar penelitian ini lebih terfokus ke arah yang diharapkan, maka batasan masalah pada penelitian ini adalah:

1. Penelitian ini memanfaatkan data sensor pada perangkat bergerak dalam mengimplementasikan *head movement control system*.
2. Penelitian ini mengembangkan *head movement control system* dalam kegiatan game *virtual reality* ataupun berkomunikasi dengan orang lain.

1.6 Sistematika pembahasan

Sistematika penulisan dari penyusunan proyek akhir ini direncanakan sebagai berikut:

BAB I

PENDAHULUAN

Pendahuluan terdiri dari latar belakang, identifikasi dan pembatasan masalah, rumusan masalah, tujuan dan manfaat serta sistematika pembahasan dari penelitian ini.

BAB II

LANDASAN KEPUSTAKAAN

Bab ini membahas mengenai teori-teori yang berkaitan dan referensi yang menunjang dalam penyelesaian penelitian ini.

BAB III

METODOLOGI PENELITIAN

Menguraikan metode dan langkah kerja yang dilakukan dalam penulisan penelitian yang terdiri dari studi literatur, perancangan, implementasi dan pengujian, serta penulisan laporan.

BAB IV

PERANCANGAN

Pada bab perancangan membahas analisis kebutuhan dan perancangan kontrol *free movement* yang digunakan dalam implementasi pada aplikasi yang menerapkan virtual *keyboard*.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Pada bab implementasi dan pengujian akan dijelaskan implementasi kontrol *free movement* pada virtual *keyboard* serta memuat hasil pengujian terhadap aplikasi yang telah dibuat.

BAB VI

PENUTUP

Memuat kesimpulan serta saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

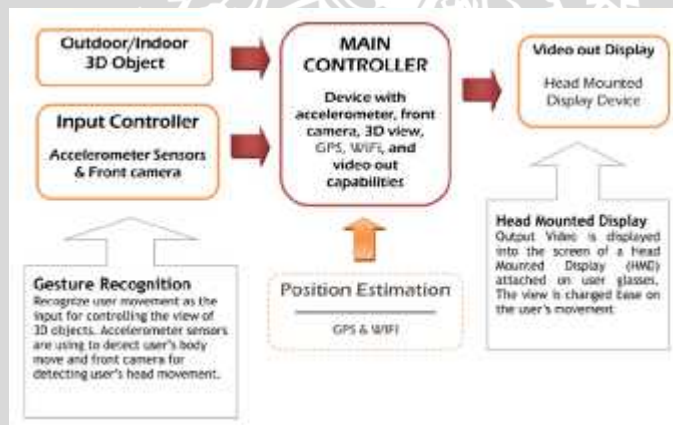
Pada Bab 2 ini dijelaskan tentang pendukung penelitian berupa kajian pustaka dan dasar teori yang digunakan. Kajian pustaka merupakan suatu pedoman yang membahas penelitian terkait yang sebelumnya telah dilakukan atau akan dilakukan. Sedangkan dasar teori sendiri menjelaskan tentang teori-teori dasar dalam menyusun penelitian ini.

2.1 Head movement control system (HEMOCS)

Head movement control system (HEMOCS) adalah sistem kontrol pada aplikasi perangkat bergerak dengan memanfaatkan sensor pergerakan kepala. Cara kerja hemocs ini sendiri memanfaatkan sensor-sensor pada perangkat bergerak misalnya seperti kamera, *accelerometer*, *gyroscope* dan *magnetometer*.

2.1.1 Kajian Pustaka

Pada penelitian sebelumnya berjudul "*Mobile Devices Based 3D Image Display Depending on User's Actions and Movements*", pada penelitian ini jelaskan bagaimana pergerakan pengguna mempengaruhi lokasi pada Gambar 3D yang ada pada perangkat bergerak (Arai, et al., 2013). pada Gambar 2.1. merupakan arsitektur sistem dari penelitian ini.

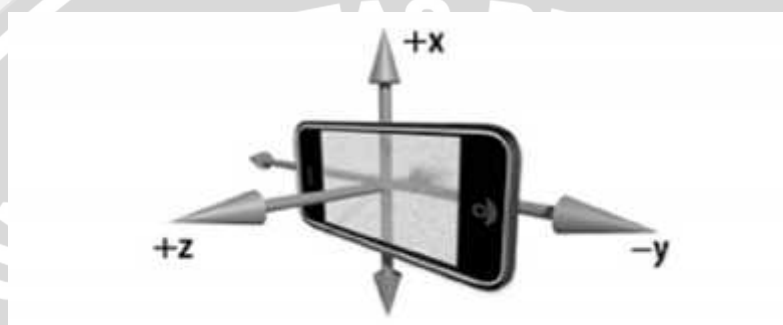


Gambar 2.1 Arsitektur Sistem 3D Image Display

Sumber : (Arai, et al., 2013)

Pada sisi input terdapat *input controller* yang meliputi sensor *accelerometer* dan kamera depan yang nantinya digunakan sebagai pendeteksi objek pergerakan kepala untuk melihat objek 3D *indoor* dan *outdoor*. Setelah itu data pada sisi input ditranslasikan pada sisi *main controller*, dimana pada sisi ini berfungsi sebagai kontrol utama perangkat bergerak yang dilengkapi dengan sensor pada sisi input. Sedangkan GPS dan WiFi mendeteksi lokasi dari *user*. Pada penelitian ini Sensor *accelerometer*, *gyro*, dan DeviceMotion merupakan sensor yang penting. Pada DeviceMotion ada 4 tipe sensor, diantaranya Attittude, UserAcceleration, RotationRate, dan Gravity.

Penelitian kedua yang berjudul “*Virtual reality Game Controlled With User’s Head and Body Movement Detection Using Smartphone Sensors*”. Penelitian ini menjelaskan metode yang memanfaatkan sensor *accelerometer* dan *gyroscope* digunakan dalam mendeteksi gerakan kepala dan tubuh dalam menentukan gerakan *virtual reality 3D game object* pada perangkat bergerak dengan Google Cardboard (Tolle, et al., 2015). Pada Gambar 2.2 gerakan dan aksi pengguna dideteksi oleh sensor *accelerometer* dan *gyroscope* atau *magnetometer* pada saat pengguna melakukan pergerakan pada saat itu juga. Untuk gerakan kepala ke kiri dan kanan menggunakan *gyroscope*, yang mana sensor ini nantinya akan menentukan nilai attitude nya, namun pada saat melihat ke atas dan kebawah menggunakan *accelerometer* dengan memanfaatkan sumbu x dan sumbu y.



Gambar 2.2 Posisi Perangkat Bergerak dalam Game 3D Virtual reality

Sumber : (Tolle, et al., 2015)

Untuk deteksi pergerakan tubuh dapat dilakukan dengan integrasi pada *accelerometer* dalam menentukan jarak dan kecepatan bergerak, menentukan gerakan linier dengan memanfaatkan sensor-sensor yang ada, dan melakukan kombinasi data sensor dengan *framework CoreMotion*. Sedangkan untuk mendeteksi pergerakan tubuh metode yang digunakan memanfaatkan sinyal *RotationRate* pada *CoreMotion* untuk mengetahui gerakan tubuh saat berjalan dan berputar.

2.2 Sensor pada iOS

Beberapa perangkat bergerak yang menggunakan sistem operasi iOS, Misalnya saja pada iPhone 4 yang memiliki resolusi layar ukuran 960x640pixel sudah memiliki beberapa macam sensor-sensor internal seperti *Accelerometer*, *Magnetometer*, *Gyroscope*, GPS dan Camera (Allan, 2011).

2.2.1 Accelerometer

Sensor *accelerometer* digunakan sebagai sensor untuk mengukur percepatan gerakan linier sehingga didapatkan nilai *roll* dan *pitch* (sumbu X dan Y) dari pergerakan tersebut. Pada kombinasi antara sensor *accelerometer* dengan *magnetometer* nantinya dapat difungsikan dalam mengukur nilai *roll*, *pitch* dan *yaw* (Sumbu X, Y dan Z) (Allan, 2011). Fungsi *accelerometer* juga dapat digunakan dalam menentukan orientasi perangkat secara vertikal maupun horizontal. Pada

saat penggunaan secara vertikal, digunakan sumbu X dalam mengukur percepatan pergerakan ke kiri dan kanan, disisi lain pada sumbu Y digunakan untuk mengukur percepatan ke atas dan bawah. Namun ketika orientasi perangkat horizontal pengukuran percepatan akan sebaliknya, Sedangkan sumbu Z, percepatan perangkat ke arah depan dan belakang dan orientasi tidak berpengaruh.

2.2.2 Gyroscope

Gyroscope merupakan sensor yang digunakan untuk melakukan *auto screen rotation* pada perangkat bergerak (Allan, 2011). Sensor *gyroscope* biasanya digunakan dalam melakukan penghitungan gerakan rotasi menggunakan sumbu X, Y, dan Z. Hampir seperti *accelerometer* dimana orientasi perangkat mempengaruhi pergerakan dari rotasi, Saat orientasi perangkat vertikal, untuk menghitung pergerakan rotasi ke atas atau ke bawah sumbu X digunakan dan untuk menentukan pergerakan rotasi ke kanan atau ke kiri sumbu Y digunakan. Namun saat orientasi perangkat secara horizontal, Gerakan rotasi akan bergerak sebaliknya. Sedangkan pada sumbu Z mirip seperti *accelerometer*, dimana orientasi perangkat tidak berpengaruh. Disisi lain dalam menentukan rotasi perangkat secara miring ke kanan atau ke kiri perhitungan sumbu Z digunakan untuk fungsi *auto screen rotation* pada perangkat bergerak.

2.2.3 Magnetometer

Magnetometer digunakan untuk mengukur serta mendeteksi kekuatan medan magnet pada sekitar perangkat. Sensor ini menjadikan perangkat bergerak sebagai kompas digital. Cara kerja dari sensor ini menghitung kekuatan medan magnet mengacu geomagnetik kutub utara ketika terdeteksi tidak ada medan magnet yang kuat di sekitar ponsel (Allan, 2011). Sedangkan Kombinasi nilai yaw (sumbu Z) oleh magnetometer dengan nilai roll dan pitch pada *accelerometer* dapat ditentukan orientasi perangkat pada saat itu juga. Data sensor magnetomer biasanya dikenal dengan yaw atau *heading data*.



Gambar 2.3 Menentukan Yaw menggunakan Magnetometer

Sumber : (Allan, 2011)

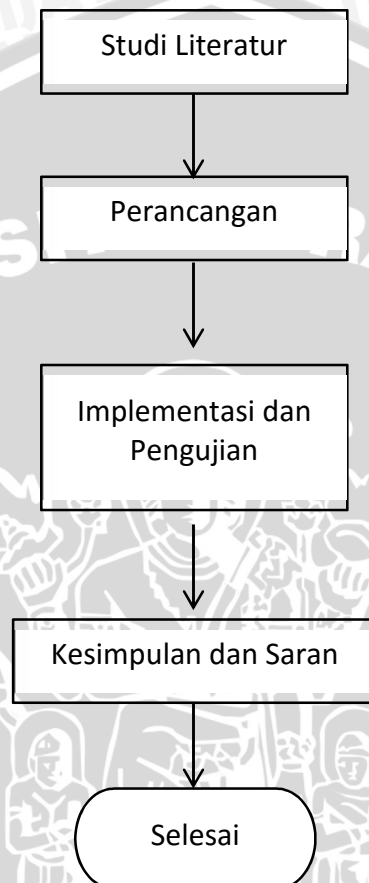
2.2.4 Core Motion

Core Motion *Framework* merupakan *framework* yang berguna mengambil data pergerakan dari sensor *accelerometer* dan *gyroscope*. Dari data-data tersebut, pengembang aplikasi yang memanfaatkan *framework* ini dapat mengamati respon dari perangkat iOS pada saat terjadi pergerakan ataupun perubahan orientasi (Allan, 2011). Dengan kombinasi data dengan DeviceMotion, Core Motion dapat menghitung dengan akurat data orientasi posisi perangkat, *rotation rate* merupakan data pergerakan dan perputaran pada perangkat yang lebih dikenal dengan *device attitude*, sedangkan *gravity* pada Core Motion digunakan menghitung gaya tarik terhadap medan magnet dan mendeteksi percepatan dari gerakan pengguna yang diberikan pada perangkat atau biasa dikenal dengan *user acceleration*.



BAB 3 METODOLOGI

Pada bab ini menjelaskan langkah-langkah yang akan dilakukan dalam melaksanakan penelitian, yaitu: studi literature, perancangan kontrol *free movement*, implementasi dan pengujian, serta kesimpulan dan saran. Dalam Gambar 3.1 merupakan alur dari tahapan penelitian.



Gambar 3.1 Diagram Alur Metodologi Penelitian

3.1 Studi literatur

Studi literatur pada penelitian ini bertujuan untuk mempelajari konsep yang digunakan dalam penulisan penelitian ini, pentingnya studi literature karena merupakan pengetahuan dasar yang menunjang sebelum melakukan implementasi. Sumber yang digunakan untuk referensi dapat bersumber dari buku, penelitian sebelumnya, jurnal, e-book maupun informasi dari internet.

Beberapa hal dilakukan untuk mendukung penelitian ini yaitu:

1. Pemahaman tentang *Head movement control system* meliputi konsep dasar.
2. Pemahaman tentang Sensor-sensor pada iOS.

3.2 Perancangan Kendali *Keyboard virtual* dengan Teknik *Free movement*

Perancangan kendali *keyboard virtual* dengan teknik *free movement* merupakan tahap untuk merancang sebuah kontrol yang digunakan untuk mendeteksi pola-pola data pada *virtual keyboard* sesuai dengan pergerakan kepala yang dilakukan oleh pengguna. Pada tahap ini terdapat beberapa tahap, yaitu merancang kendali *keyboard virtual*, dan pengambilan data dengan teknik *free movement*.

3.2.1 Gambaran Umum Sistem

Penjelasan tentang sistem tersebut bekerja baik dari teknik dan fungsi dari sistem yang akan dirancang pada penelitian ini.

3.2.2 Merancang Kendali *Keyboard virtual* dengan Teknik *Free movement*

Pengambilan data dalam penelitian ini bertujuan untuk mengumpulkan apakah karakter pada *virtual keyboard* dapat dideteksi dengan kontrol *free movement*. Untuk melakukan pengambilan data ini secara langsung menggunakan alat yang menunjang sebuah *virtual reality* dalam penelitian ini menggunakan *Cardboard*.

3.2.3 Algoritma

Penjelasan tentang alur kerja dari penggunaan teknik *free movement* pada kendali *keyboard virtual*.

3.3 Implementasi dan Pengujian

Implementasi dan pengujian merupakan tahapan dimana hasil perancangan kendali *keyboard virtual* dengan teknik *free movement* dilakukan implementasi pada sebuah aplikasi yang menerapkan teknik *free movement*. Kemudian dari aplikasi tersebut dilakukan pengujian berapa akurasi dari teknik *free movement* saat digunakan pada aplikasi tersebut.

Selain pengujian dalam hal akurasi diperlukan pengujian usability untuk mendapatkan data langsung dengan menggunakan teknik kuisioner dari para pengguna aplikasi tersebut.

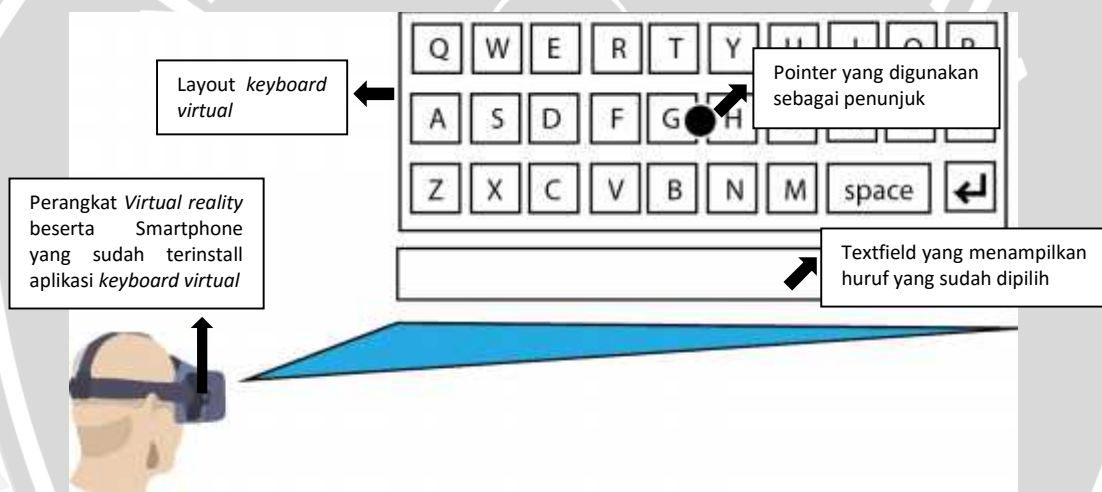
3.4 Kesimpulan dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah tahap perancangan, implementasi dan pengujian pada sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian sistem dan analisis yang dilakukan. Saran merupakan hal-hal yang bertujuan memperbaiki dan menyempurnakan penelitian menjadi lebih baik untuk pengembangan selanjutnya.

BAB 4 PERANCANGAN KENDALI KEYBOARD VIRTUAL DENGAN TEKNIK *FREE MOVEMENT*

Bab ini akan membahas mengenai perancangan sistem kendali pergerakan kepala dengan teknik *free movement* yang diimplementasikan pada *keyboard virtual*. Tahapan dalam perancangan kendali ini dibagi 3 macam. Tahap pertama yaitu gambaran umum dari sistem yang dibuat yang mana ditahap ini dijelaskan seperti apakah sistem yang dibuat ini. Tahap kedua yaitu merancang kendali *keyboard virtual* dengan teknik *free movement*, pada bagian ini dirancang sebuah virtual *keyboard* yang digunakan sebagai acuan pengguna untuk beraktivitas dengan teknik *free movement* pada tahapan ini nantinya akan menjadi dasar dari tahap ketiga. Pada tahapan ketiga ini dijelaskan bagaimana alur dari pergerakan teknik *free movement* itu sendiri dari mulai awal sistem dijalankan sampai pada saat huruf yang dipilih ditampilkan pada *textfield* yang disediakan.

4.1 Gambaran Umum Sistem

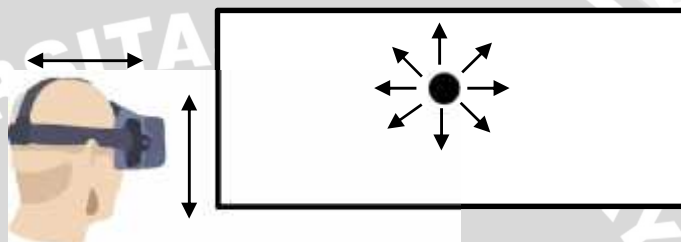


Gambar 4.1 Gambaran Umum Sistem

Sistem kendali *keyboard virtual* dengan teknik *free movement* ini dirancang untuk membantu para penyandang disabilitas ataupun digunakan dalam melakukan penulisan kata ataupun kalimat menggunakan *keyboard* pada saat memainkan game seperti pada Gambar 4.1. Untuk dapat menggunakan sistem ini diperlukan sebuah perangkat *virtual reality* beserta smartphone yang sudah terinstall aplikasi *keyboard virtual*. Ketika aplikasi tersebut berjalan pengguna akan langsung dihadapkan dengan *layout keyboard* qwerty beserta *textfield* yang nantinya berisi kata ataupun kalimat yang akan ditulis. Selain itu tampilan awal akan menampilkan pointer berwarna putih sebagai penunjuk dan membantu *user* dalam melakukan pemilihan huruf.

4.2 Merancang Kendali *Keyboard Virtual* dengan Teknik *Free Movement*

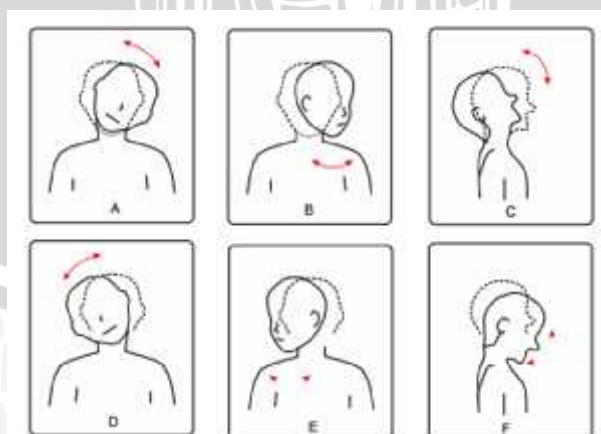
Pada Penelitian ini teknik yang digunakan yaitu adalah *free movement*, Teknik *free movement* merupakan teknik dimana pergerakan kepala dilakukan secara bebas, mirip seperti gerakan mouse pada komputer. Untuk membantu aktivitas *user* dibuatlah sebuah pointer. Dimana pointer ini digunakan sebagai penunjuk dan alat untuk melakukan interaksi dengan aplikasi yang ada. Dalam teknik *free movement* ini pointer sangatlah penting dikarenakan pada saat *user* melakukan pergerakan secara bebas, *user* tidak mengetahui pergerakan yang dilakukannya berada pada posisi mana serta *user* akan kebingungan jika tidak ada tanda yang digunakan pada saat diimplementasikan pada *keyboard virtual*.



Gambar 4.2 Ilustrasi Teknik *Free movement*

Pada Gambar 4.2 dijelaskan bahwa pointer bergerak sesuai dengan pergerakan kepala yang dilakukan *user*. Tanda panah pada ilustrasi diatas sebagai penanda jika *user* bergerak keatas ataupun bawah, pointer akan bergerak juga keatas maupun kebawah. Hal tersebut sama pada saat gerakan kepala *user* ke kanan atau ke kiri.

Dalam melakukan pergerakan kepala ada baiknya memperhatikan beberapa jenis pergerakan kepala diantaranya menoleh kanan/kiri, pergerakan kepala vertical atas/bawah dan untuk pergerakan menyamping kiri/kanan. Jenis pergerakan kepala tersebut yang nantinya sebagai acuan dalam pemanfaatan teknik *free movement* ini.



Gambar 4.3 Jenis Pergerakan Kepala

Sumber: (Safii, 2016)

1. Gerakan ke kanan dan kiri

Dalam gerakan ke kanan dan kiri Digunakan sumbu yaw. Sumbu yaw akan mendeteksi gerakan kepala dari *user* yang akan diikuti oleh pointer sesuai dengan arah yang digerakan.

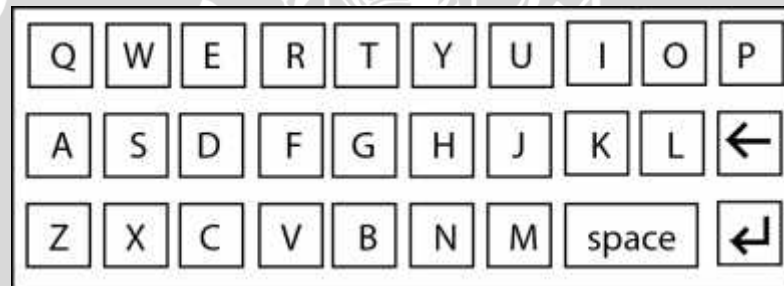
2. Gerakan ke atas dan bawah

Dalam gerakan ke atas dan bawah digunakan sumbu roll. Sumbu roll sendiri akan mendeteksi gerakan kepala dari *user* yang akan diikuti oleh pointer sesuai dengan arah yang digerakan.

3. Gerakan Menyamping

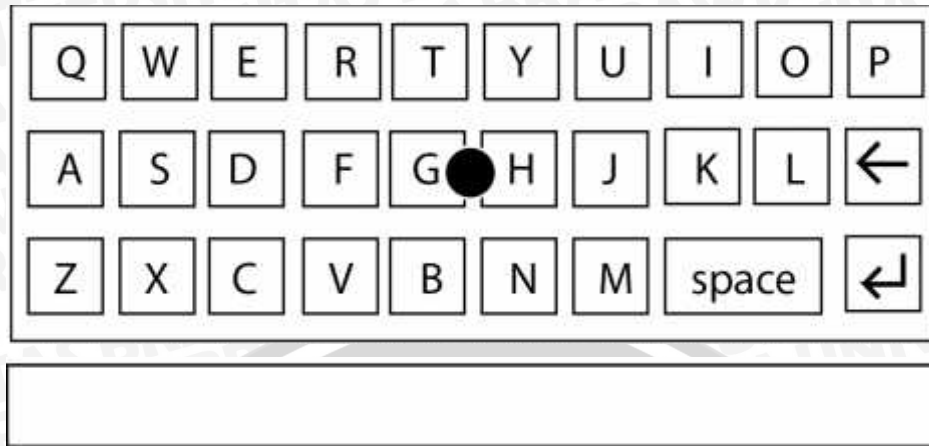
Dalam gerakan menyamping digunakan sumbu pitch. Sumbu pitch sendiri gerakan kepala dari *user* yang akan diikuti oleh pointer sesuai dengan arah yang digerakan.

Pada dasarnya gerakan dari teknik *free movement* ini mengikuti gerakan kepala dari *user* itu sendiri. Yang nantinya sensor pada perangkat bergerak yang digunakan akan mendeteksi gerakan dari pergerakan tersebut.



Gambar 4.4 Layout Keyboard

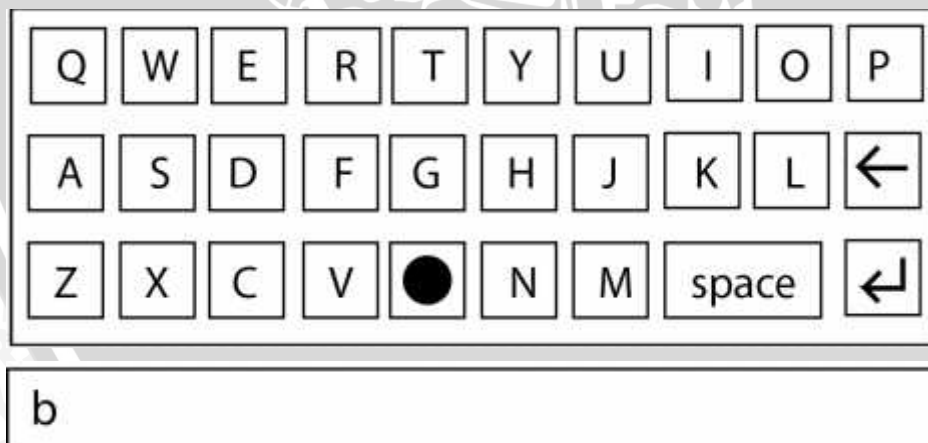
Pada Penelitian ini, *layout keyboard* yang digunakan adalah *layout keyboard* qwerty seperti pada Gambar 4.4 dengan perpaduan 10 kotak dalam 3 baris. Penggunaan *layout* ini didasari dengan populernya *layout* qwerty pada perangkat bergerak yang ada. Sehingga *user* dengan mudah menggunakan aplikasi pada penelitian ini karena sudah terbiasa dengan *layout* qwerty yang digunakan. Dalam penelitian ini karakter yang digunakan terbatas dikarenakan bahwa karakter ini merupakan karakter yang sering digunakan dalam melakukan *chatting* atau *messaging*.



Gambar 4.5 Titik Awal Pointer

Sementara untuk posisi awal pointer ditentukan terletak pada titik tengah antara huruf G dan H seperti pada Gambar 4.5. Yang mana titik tersebut merupakan titik pusat dari *layout keyboard* yang digunakan. Titik pusat tersebut nantinya akan menampilkan pointer utama yang digunakan sebagai penunjuk pada penelitian ini.

Karena penelitian ini menggunakan teknik *free movement*, saat melakukan aktivitas misalnya melakukan pemilihan huruf B pointer harus tepat berada dihuruf B dan akan ada lingkaran yang menunjukan dan menunggu selama 3 detik atau menekan tombol klik pada perangkat *cardboard* agar aplikasi dapat memproses aktivitas dari *user* seperti pada Gambar 4.6.



Gambar 4.6 Contoh Pemilihan Karakter Huruf B

4.3 Algoritma

Pada sub bab sebelumnya didapatkan bahwa perancangan teknik *free movement* untuk *keyboard virtual* dapat terjadi karena beberapa tahapan, dimana tahapan ini nantinya menjadi dasar penggunaan teknik *free movement* pada *keyboard virtual* hemocs. *Keyboard virtual* ini didesain khusus untuk penggunaan teknik *free movement*. Yang mana dalam perancangan teknik *free movement* dapat dilakukan dengan beberapa proses berikut ini :

1. Pergerakan Pointer dengan memperhatikan sumbu sensor data

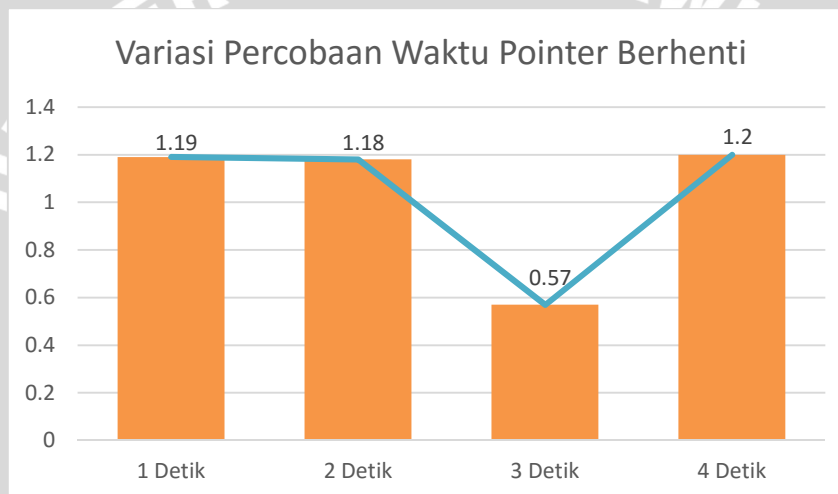
Pada tahapan ini dibuatlah sebuah pointer/reticle yang sudah terhubung dengan sensor-sensor internal pada perangkat bergerak, pointer ini dibuat berbentuk lingkaran berwarna putih yang mudah untuk dikenali. Nantinya pergerakan dari sensor dapat dideteksi apakah pointer mengalami perubahan letak dari posisi awal. Pada saat awal aplikasi berjalan, sistem akan menjalankan scene berupa *keyboard virtual* beserta pointer yang telah dibuat sebelumnya. Setelah itu *user* melakukan pergerakan kepala ke segala arah untuk melihat apakah terjadi perubahan pada pointer tersebut. Untuk pergerakan pada pointer, terjadi perubahan letak pointer akibat pergerakan kepala yang dilakukan oleh *user* dengan memperhatikan beberapa jenis pergerakan kepala pada bab sebelumnya.

2. Mendeteksi aksi yang dilakukan oleh Pointer

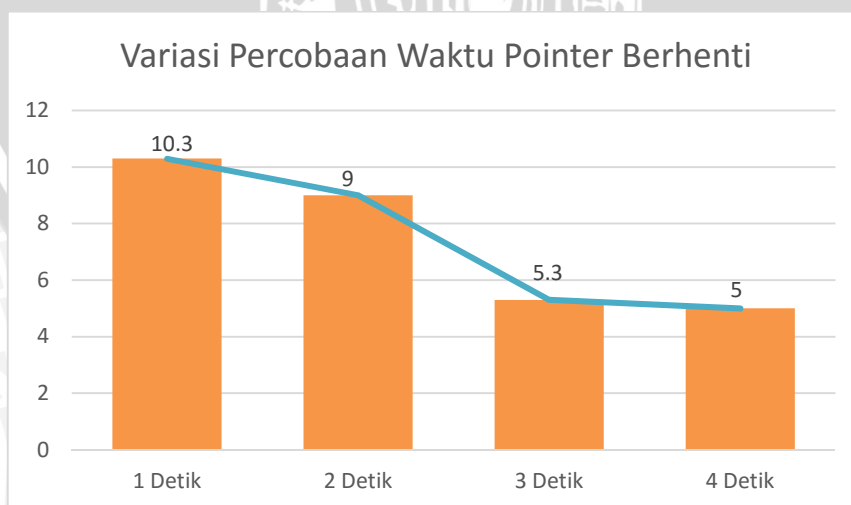
Pada tahapan sebelumnya pointer melakukan perubahan sesuai dengan pergerakan kepala dari *user*, pada tahapan ini dijelaskan bahwa pada saat pointer mengarah ke objek dalam aplikasi *keyboard virtual*. Pointer akan melakukan event atau aksi ketika pointer yang awalnya berbentuk lingkaran putih penuh, menjadi pointer yang memiliki lubang seperti bentuk donat. Setelah itu pointer akan melakukan aksi *onEnter* jika pointer berada tepat pada objek yang dipilih dan diharuskan menunggu selama 3 detik, Agar sistem dapat mendeteksi objek apa yang telah dipilih oleh pointer dan objek yang dipilih akan ditampilkan pada textfield yang sudah disediakan. Namun jika pointer berada pada luar objek, pointer akan melakukan event *onExit* yang mana event ini mengatur dari awal waktu pemilihan huruf serta melakukan set awal huruf yang dipilih ketika huruf sebelumnya dipilih kurang dari 3 detik. Berdasarkan percobaan pointer diharuskan berhenti 3 detik dikarenakan waktu 1 detik dan 2 detik terlalu cepat dan banyak terjadi kesalahan pemilihan huruf, sedangkan untuk 4 detik terlalu lama meskipun minim terjadi kesalahan seperti pada tabel 4.1. Skenario dari percobaan ini dengan melakukan pengetikan huruf "informatika" untuk kecepatan pengetikan menggunakan stopwatch sebagai alat ukurnya.

Tabel 4.1 Percobaan Pointer Harus Berhenti

Pengujian	1 Detik		2 Detik		3 Detik		4 Detik	
	Error	Kecepatan	Error	Kecepatan	Error	Kecepatan	Error	Kecepatan
1	10 kali	1 Menit 20 detik	9 kali	1 Menit 14 detik	5 kali	1 Menit	5 kali	1 Menit 3 detik
2	12 kali	1 Menit 25 detik	10 kali	1 Menit 22 detik	6 kali	58 detik	5 kali	1 Menit 5 detik
3	9 kali	1 Menit 12 detik	8 kali	1 Menit 10 detik	5 kali	55 detik	5 kali	1 Menit
Rata – Rata	10,3 kali	1 Menit 19 detik	9 kali	1 Menit 18 detik	5,3 kali	57 detik	5 kali	1 Menit 2 detik



Gambar 4.7 Variasi Percobaan Waktu Pointer Berhenti berdasarkan Kecepatan

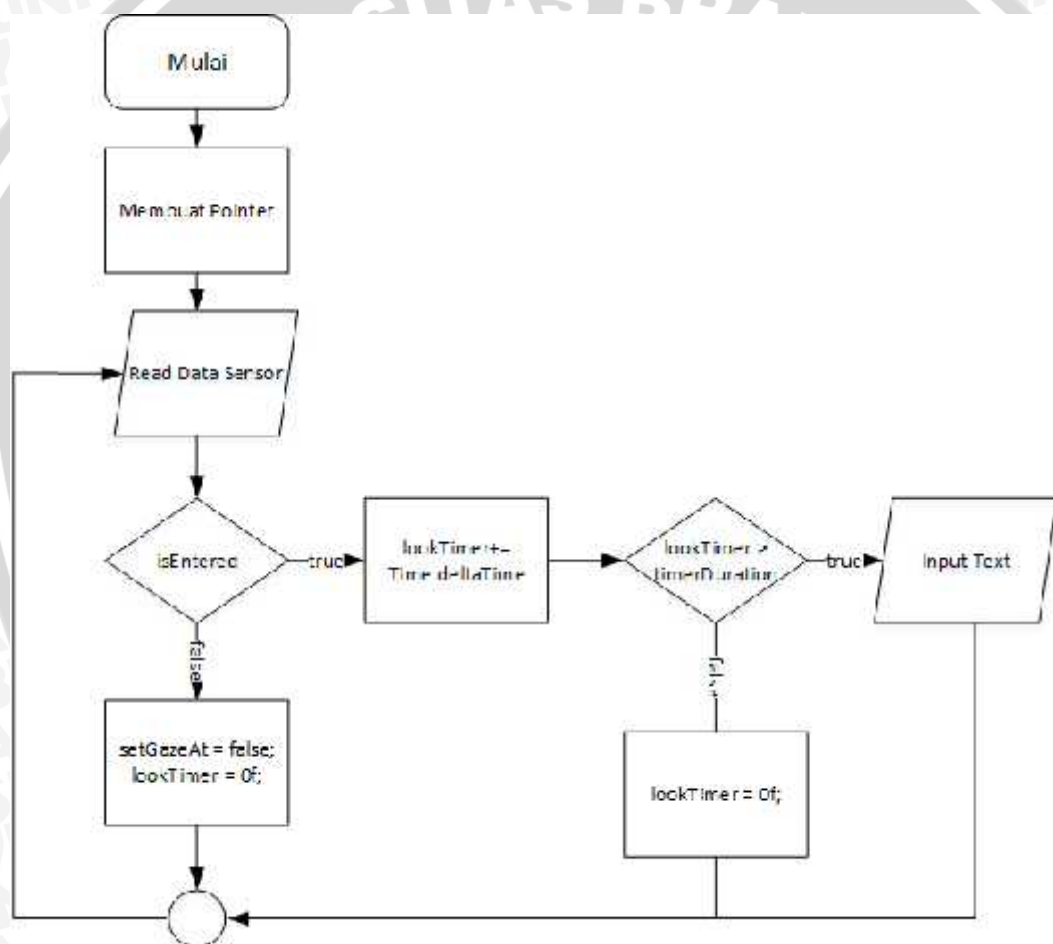


Gambar 4.8 Variasi Percobaan Waktu Pointer Berhenti Berdasarkan Jumlah Error

Pada Gambar 4.7 dan 4.8 dapat disimpulkan bahwa dengan waktu pointer berhenti selama 3 detik dalam melakukan pengetikan sebuah kata lebih cepat dibanding yang lainnya, namun dengan waktu berhenti selama 4 detik jumlah error lebih sedikit dari pada yang lainnya akan tetapi jumlah error waktu berhenti 3 detik dan 4 detik terjadi selisih yang sedikit yaitu 0,3. Maka dari itu waktu pointer berhenti selama 3 detik dipilih dikarenakan kecepatan dan jumlah error yang terjadi cukup sedikit dibandingkan dengan waktu 1 detik ataupun 2 detik.

Berdasarkan perancangan algoritma pada sub bab sebelumnya pada Gambar 4.9 ditampilkan alur perancangan teknik *free movement* mulai dari pergerakan pointer dengan memperhatikan sumbu sensor data sampai deteksi aksi yang dilakukan pada pointer.

4.3.1 Flowchart



Gambar 4.9 Flowchart Perancangan Teknik *Free movement*

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Untuk mendukung hasil penelitian ini diperlukan sebuah aplikasi sederhana yang mengimplementasikan teknik *free movement* yang mana aplikasi tersebut digunakan dalam melakukan pengetikan kalimat atau huruf pada 3 ukuran *keyboard*. Pada aplikasi ini akan dilakukan pengujian akurasi dan usabilitas dari perancangan kendali dengan pergerakan kepala menggunakan teknik *free movement*.

5.1 Implementasi

Pada subbab ini menjelaskan implementasi dari teknik *free movement* menggunakan hemocs pada kendali *keyboard virtual* berdasarkan hasil perancangan yang dibahas pada bab sebelumnya.

5.1.1 Batasan-batasan Implementasi

Batasan-batasan dari kendali *keyboard virtual* hemocs menggunakan teknik *free movement* ini berupa :

1. Sistem kendali ini diimplementasikan untuk perangkat bergerak yang mendukung *virtual reality*.
2. Sistem Kendali ini menggunakan Bahasa pemrograman C-Sharp dengan tools Unity.
3. Lingkungan pengembangan dan implementasi dari kendali ini ada pada *Mobile Apps*.

5.1.2 Implementasi Kode Program

Pada subbab ini menjelaskan beberapa kode program yang digunakan dalam implementasi teknik *free movement* pada kendali *keyboard virtual*.

5.1.2.1 Fungsi Pembuatan Pointer

```

1  [AddComponentMenu("GoogleVR/UI/GvrReticle")]
2  [RequireComponent(typeof(Renderer))]
3  public class GvrReticle : MonoBehaviour, IGvrGazePointer {
4      public int reticleSegments = 20;
5      public float reticleGrowthSpeed = 8.0f;

6      private Material materialComp;
7      private GameObject targetObj;

8      private float reticleInnerAngle = 0.0f;
9      private float reticleOuterAngle = 0.5f;
10     private float reticleDistanceInMeters = 10.0f;

11     private const float kReticleMinInnerAngle = 0.0f;
12     private const float kReticleMinOuterAngle = 0.5f;
13     private const float kReticleGrowthAngle = 1.5f;

14     private const float kReticleDistanceMin = 0.45f;
15     private const float kReticleDistanceMax = 10.0f;

```

```

16     private float reticleInnerDiameter = 0.0f;
17     private float reticleOuterDiameter = 0.0f;

18     void Start () {
19         CreateReticleVertices();

20     materialComp=gameObject.GetComponent<Renderer>().material;
21     }

22     void OnEnable() {
23         GazeInputModule.gazePointer = this;
24     }

```

Kode 5.1 Fungsi inialisasi parameter dalam pembuatan pointer

Pada kode 5.1 dijelaskan kode dalam tahap pembuatan pointer dimana pada fungsi ini melakukan inialisasi dengan beberapa parameter sebagai acuan besar ukuran dari pointer yang akan dibuat. Penjelasan kode program diatas adalah sebagai berikut :

4. Baris 1-2 : menambahkan komponen menu yang terletak pada "GoogleVR/UI/GvrReticle" dan merubah komponen dalam bentuk renderer.
5. Baris 3 : inialisasi kelas GvrReticle yang mengimplement MonoBehaviour dan IGvrGazePointer
6. Baris 4-17 : inialisasi parameter berupa diameter dari pointer yang akan digunakan, angle, ukuran seberapa besar retricle yang akan dibuat.
7. Baris 18-24 : merupakan tahap ketika program dijalankan yaitu dengan melakukan pembuatan pointer.

```

1     private void CreateReticleVertices() {
2         Mesh mesh = new Mesh();
3         gameObject.AddComponent<MeshFilter>();
4         GetComponent<MeshFilter>().mesh = mesh;

5         int segments_count = reticleSegments;
6         int vertex_count = (segments_count+1)*2;

7         #region Vertices

8         Vector3[] vertices = new Vector3[vertex_count];

9         const float kTwoPi = Mathf.PI * 2.0f;
10        int vi = 0;
11        for (int si = 0; si <= segments_count; ++si) {
12            float angle = (float)si / (float)(segments_count) *
kTwoPi;

13            float x = Mathf.Sin(angle);
14            float y = Mathf.Cos(angle);

15            vertices[vi++] = new Vector3(x, y, 0.0f); // Outer
vertex.
16        }

```

```

17     vertices[vi++] = new Vector3(x, y, 1.0f); // Inner
18     }
19     #endregion
20     #region Triangles
21     int indices_count = (segments_count+1)*3*2;
22     int[] indices = new int[indices_count];
23
24     int vert = 0;
25     int idx = 0;
26     for (int si = 0; si < segments_count; ++si) {
27         indices[idx++] = vert+1;
28         indices[idx++] = vert;
29         indices[idx++] = vert+2;
30
31         indices[idx++] = vert+1;
32         indices[idx++] = vert+2;
33         indices[idx++] = vert+3;
34
35         vert += 2;
36     }
37     #endregion
38
39     mesh.vertices = vertices;
40     mesh.triangles = indices;
41     mesh.RecalculateBounds();
42     mesh.Optimize();
43 }

```

Kode 5.2 Fungsi perhitungan pembuatan pointer

Pada kode 5.2 dijelaskan alur dari pembuatan objek pointer dalam bentuk 3 dimensi. Penjelasan kode program diatas adalah sebagai berikut :

1. Baris 1-6 : pembuatan objek 3 dimensi berupa mesh object dan melakukan penambahan komponen meshfilter.
2. Baris 7-18 : wilayah pembuatan vektor 3 dimensi berbentuk vertices.
3. Baris 19-33 : wilayah pembuatan indices berbentuk triangles.
4. Baris 34-38 : memasukkan data vertices,indices dan melakukan optimasi pada mesh object pada inisialisasi awal.

5.1.2.2 Fungsi Kendali Hemocs

```

1  [AddComponentMenu("GoogleVR/GvrHead")]
2  public class GvrHead : MonoBehaviour {
3      public bool trackRotation = true;
4      public bool trackPosition = true;
5      public Transform target;
6      public bool updateEarly = false;
7
8      public Ray Gaze {
9          get {
10             UpdateHead();
11             return new Ray(transform.position, transform.forward);
12         }
13     }
14 }

```

```

13     public delegate void HeadUpdatedDelegate(GameObject head);
14     public event HeadUpdatedDelegate OnHeadUpdated;

15     void Awake() {
16         GvrViewer.Create();
17     }

18     private bool updated;

19     void Update() {
20         updated = false; // OK to recompute head pose.
21         if (updateEarly) {
22             UpdateHead();
23         }
24     }

25     // Normally, update head pose now.
26     void LateUpdate() {
27         UpdateHead();
28     }

```

Kode 5.3 Inisialisasi fungsi pergerakan kepala

Pada Kode 5.3 dijelaskan inisialisasi dari fungsi kendali dengan pergerakan kepala berupa track rotation dan track position. Berikut penjelasan kode programnya :

1. Baris 1 : menambahkan komponen menu yang terletak pada "GoogleVR/GvrHead".
2. Baris 2 : inisialisasi kelas GvrHead yang mengimplemen kelas MonoBehaviour.
3. Baris 3-6 : inisialisasi gerakan kepala berupa trackrotation, trackposition dan updateEarly.
4. Baris 7-12 : mendapatkan data pergerakan berupa transformasi dari posisi pointer.
5. Baris 13-14 : merupakan variabel untuk mendapatkan aksi jika kepala telah bergerak.
6. Baris 15-17 : merupakan method untuk menampilkan tampilan *virtual reality*.
7. Baris 19-28 : merupakan deteksi jika terjadi pergerakan dari kepala.

```

1     private void UpdateHead() {
2         if (updated) { // Only one update per frame, please.
3             return;
4         }
5         updated = true;
6         GvrViewer.Instance.UpdateState();

7         if (trackRotation) {
8             var rot = GvrViewer.Instance.HeadPose.Orientation;

```

```

9      if (target == null) {
10         transform.localRotation = rot;
11     } else {
12         transform.rotation = target.rotation * rot;
13     }
14 }

15     if (trackPosition) {
16         Vector3 pos = GvrViewer.Instance.HeadPose.Position;
17         if (target == null) {
18             transform.localPosition = pos;
19         } else {
20             transform.position = target.position +
21             target.rotation * pos;
22         }
23     }

24     if (OnHeadUpdated != null) {
25         OnHeadUpdated(gameObject);
26     }
27 }

```

Kode 5.4 Fungsi pergerakan kepala jika digerakkan

Pada kode 5.4 dijelaskan kode program untuk fungsi pergerakan kepala jika terjadi pergerakan. Penjelasan programnya adalah sebagai berikut :

1. Baris 2-6 : merupakan penjelasan jika dalam program yang berjalan terjadi perubahan status gerakan.
2. Baris 7-14 : terjadi perubahan berupa pergerakan rotasi yang menggunakan kepala.
3. Baris 15-22 : terjadi perubahan pergerakan berupa posisi dengan menggunakan kepala.
4. Baris 23-27 : Update pergerakan kepala jika pada kondisi tidak null, maka object tersebut berpindah sesuai dengan gerakan.

5.1.2.3 Fungsi Pengecekan pointer harus berhenti selama 3 detik

```

1  Float lookTimer = 0f;
2  bool isEntered = false;
3  if (isEntered)
4      {
5          Renderer rend = GetComponent<Renderer>();
6          a = rend.ToString();
7          string coba = a.Substring(0,7);
8          lookTimer += Time.deltaTime;
9          Debug.Log("Waktu : " + lookTimer);
10         if (lookTimer > timerDuration)
11         {
12             lookTimer = 0f;
13             SetGazedAt(false);
14         }
15     }

```

Kode 5.5 Fungsi pengecekan waktu pointer berhenti selama 3 detik

Pada kode 5.5 dijelaskan kode program untuk fungsi pergerakan kepala jika pointer harus berhenti 3 detik untuk melakukan aksi. Penjelasan programnya adalah sebagai berikut :

1. Baris 1-2 : merupakan inisialisasi waktu awal dan kondisi pointer awal sebelum melakukan aksi.
2. Baris 3-8 : terjadi aksi dan penambahan waktu untuk mencapai waktu 3 detik.
3. Baris 9-11 : kondisi ketika waktu awal melebihi 3 detik dan sistem melakukan pengetikan huruf.

5.1.2.4 Fungsi Penulisan pada Textfield setelah huruf dipilih

```

1 public void SendMessageA()
2 {
3     Text InputText =
    GameObject.Find("InputField").GetComponentInChildren<Text>(
    );
4     if (InputText==null)
5     {
6         InputText.text = "a";
7     }
8     else
9     {
10        InputText.text = InputText.text+"a";
11    }
12 }
```

Kode 5.6 Fungsi Pemilihan Huruf sebagai Contoh Pemilihan Huruf A

Pada kode 5.5 dijelaskan kode program untuk fungsi pemilihan huruf serta penulisan pada textfield pada saat huruf dipilih. Penjelasan programnya adalah sebagai berikut :

1. Baris 3 : inisialisasi objek textfield dan textview dan disimpan dalam variabel.
2. Baris 4-8 : jika pada kondisi textfield kosong maka huruf yang dipilih akan mengisi textfield tersebut.
3. Baris 8-12 : jika pada kondisi textfield tidak kosong maka huruf yang telah dipilih akan berada setelah huruf terakhir dipilih.

```

1 public void DeleteData()
2 {
3     Text InputText =
4     GameObject.Find("InputField").GetComponentInChildren<Text>(
5     );
6     InputText.text =
    InputText.text.Remove(InputText.text.Length-1);
7 }
```

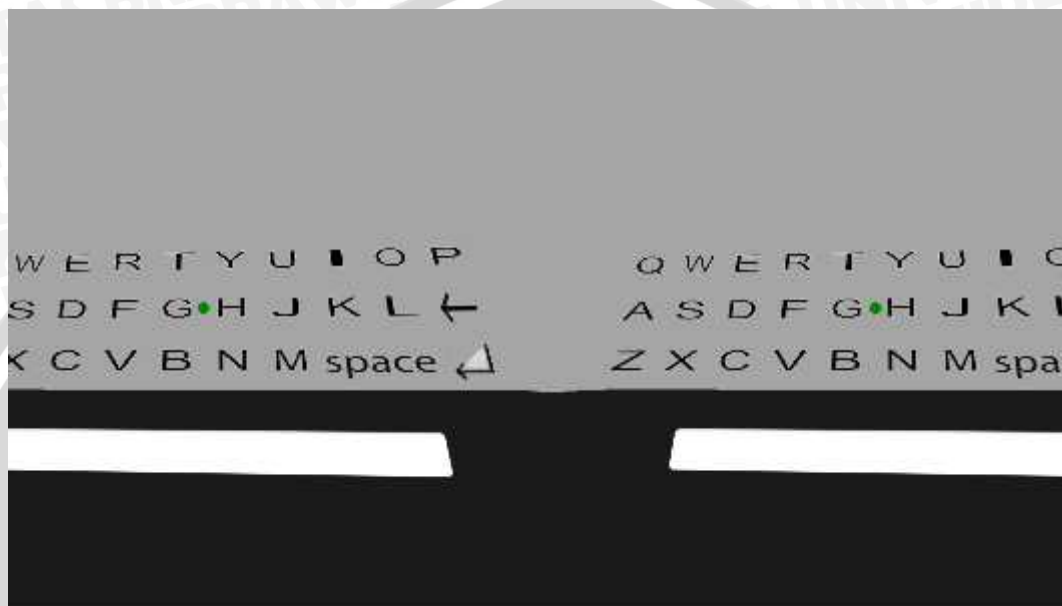
Kode 5.7 Hapus huruf terakhir yang telah dipilih

Pada Kode 5.6 dijelaskan kode program untuk menghapus huruf yang dipilih sebelumnya pada textfield. Penjelasan programnya adalah sebagai berikut :

1. Baris 3 : inialisasi objek input text dan menyimpannya dalam variabel InputText.
2. Baris 4 : penghapusan data terakhir yang diinputkan pada input text.

5.1.3 Implementasi Antarmuka

Implementasi antarmuka disini memiliki 1 scene atau halaman tampilan dengan latar belakang dari terrain default dari unity.



Gambar 5.1 Antarmuka Kendali *Keyboard virtual*

Gambar 5.1 menampilkan antarmuka dari kendali *keyboard virtual* dengan teknik *free movement* yang mana memiliki 2 objek awal diantaranya :

1. *Layout keyboard virtual* yang berguna sebagai interaksi *user* dalam memilih karakter yang ada.
2. Pointer yang berguna sebagai penunjuk berada pada karakter apa *user* saat ini.

5.2 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Komponen	Spesifikasi
Model Name	Asus X4550
Processor	Intel Core i3 1.8Ghz
Memory	10 GB
Graphics/Display	NVIDIA GeForce 720M 2GB

Untuk proses implementasi dan pengujian diterapkan pada perangkat bergerak dengan spesifikasi sebagai berikut.

Tabel 5.2 Spesifikasi Perangkat Keras Perangkat Bergerak

Komponen	Spesifikasi
Model Name	Xiaomi Mi4i, Iphone 4s
Processor	Qualcomm Snapdragon
Memory	2 GB

5.3 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

Tabel 5.3 Spesifikasi Perangkat Lunak Komputer

Komponen	Spesifikasi
Sistem Operasi	Windows 10 Pro
Tools Development	Unity Version 5.3.5, Microsoft Visual Studio 2015
Bahasa Pemrograman	C-Sharp

Untuk proses implementasi dan pengujian diterapkan pada perangkat bergerak dengan spesifikasi sebagai berikut.

Tabel 5.4 Spesifikasi Perangkat Lunak Perangkat Bergerak

Komponen	Spesifikasi
Sistem Operasi	MIUI 7, Android Lollipop 5.0.1, iOS 7.0

5.4 Pengujian

Pada subbab pengujian ini menjelaskan tentang hasil pengujian dari penggunaan teknik *free movement* dengan hemocs pada sebuah *keyboard virtual*. Pengujian pada penelitian ini menggunakan 2 jenis diantaranya pengujian akurasi dan pengujian usabilitas.

Pengujian akurasi dilakukan untuk mengetahui seberapa akurat teknik *free movement* dalam *keyboard virtual* serta apakah *user* nyaman dengan ukuran seperti apa dalam menunjang kecepatan *user* mengetik sebuah kalimat dan berapakah persentase kesalahan penulisan. Kemudian untuk pengujian usabilitas dilakukan dengan perbandingan ukuran huruf tersebut dalam penggunaan teknik *free movement* dengan memberikan testing serta beberapa pertanyaan responden yang telah melakukan testing.

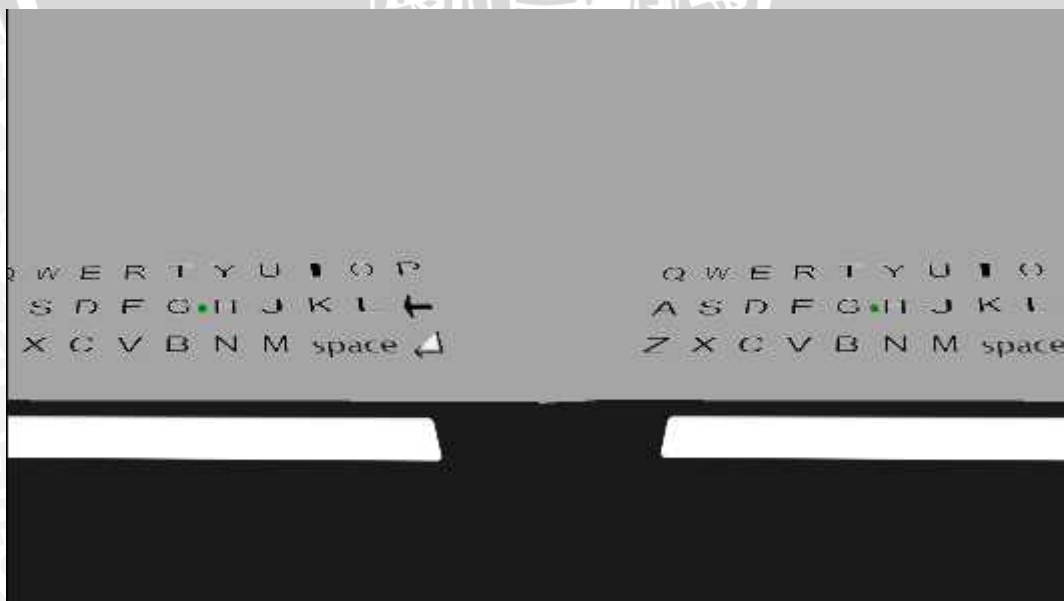
5.4.1 Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui seberapa akurat dan cepat penggunaan teknik *free movement* ini dan pengujian ini menjadi sebuah landasan dalam pengujian tahap selanjutnya.

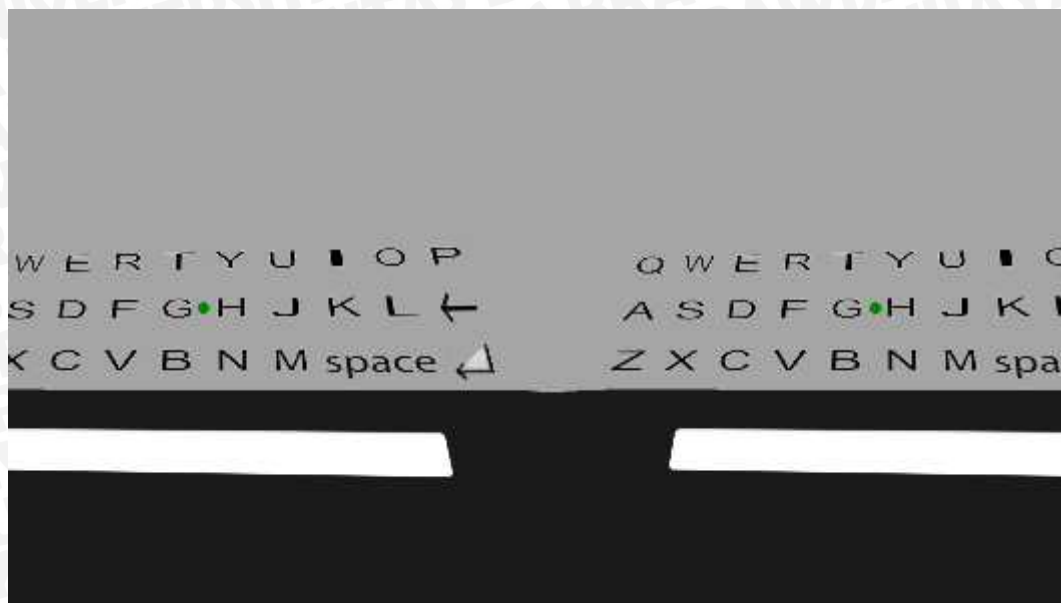
Pengujian dilakukan dengan menggunakan 3 ukuran huruf, yaitu ukuran huruf dengan skala 0.3 cm, 0.4 cm dan 0.5 cm seperti pada Gambar 5.2, 5.3 dan 5.4. Dari ukuran huruf tersebut scenario yang dijalankan yaitu dengan melakukan pengetikan kalimat “*The quick brown fox jumps over the lazy dog*” dengan menggunakan *keyboard virtual* yang tersedia dan dihitung berapa kecepatan dari user melakukan pengetikan kalimat tersebut dengan menggunakan stopwatch.



Gambar 5.2 Ukuran huruf *keyboard* skala 0,3 cm



Gambar 5.3 Ukuran huruf skala *keyboard* 0,4 cm

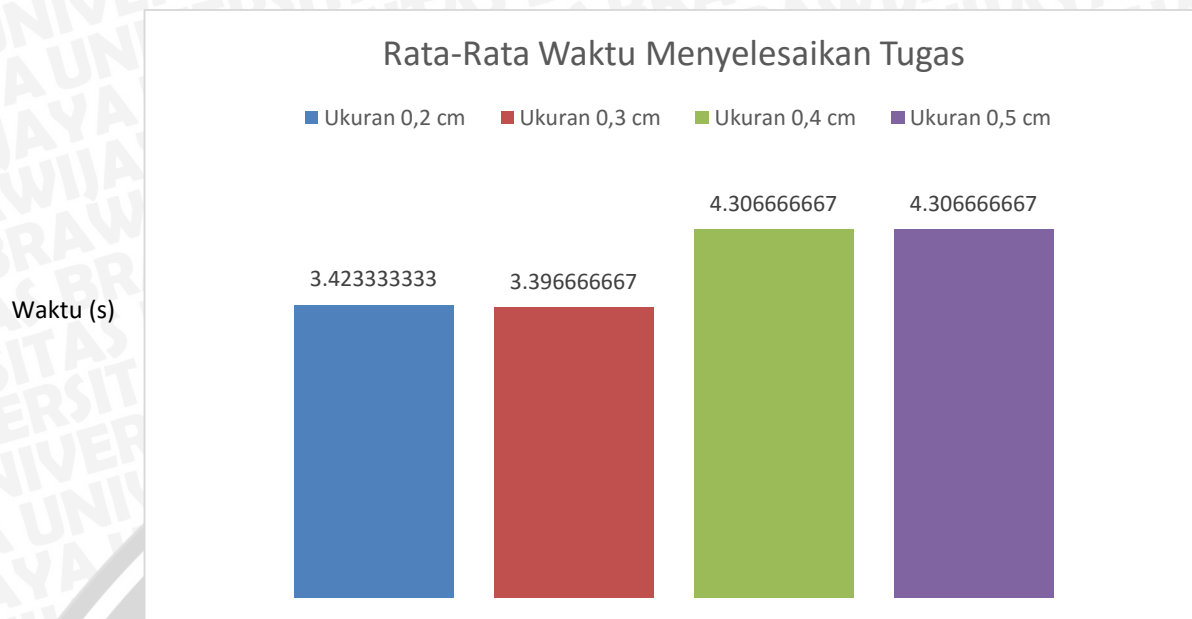


Gambar 5.4 Ukuran huruf skala keyboard 0,5 cm

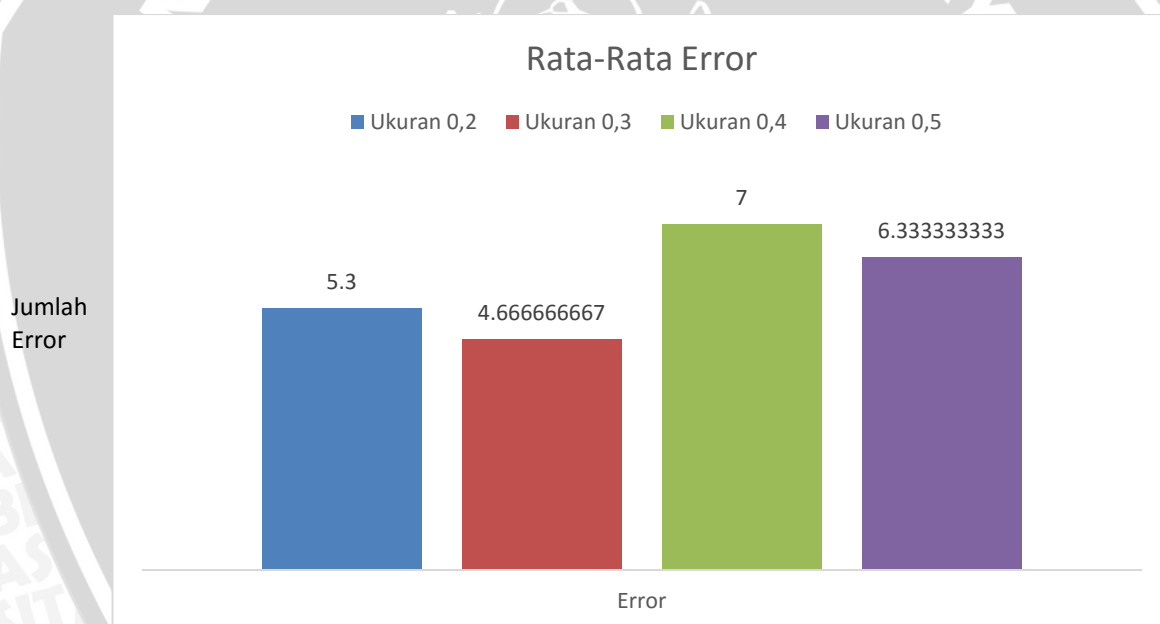
Tabel 5.5 Hasil Pengujian dengan kalimat *“the quick brown fox jumps over the lazy dog”* menggunakan 3 Ukuran huruf Keyboard

Pengujian	Skala 0,2 cm		Skala 0,3 cm		Skala 0,4 cm		Skala 0,5 cm	
	Waktu	Error	Waktu	Error	Waktu	Error	Waktu	Error
1	3 Menit 27 Detik	5 kali	3 Menit 29 Detik	4 kali	4 Menit 36 Detik	7 kali	4 Menit 42 Detik	7 kali
2	3 Menit 53 Detik	6 kali	3 Menit 40 Detik	4 kali	4 Menit 26 Detik	8 kali	4 Menit 35 Detik	6 kali
3	3 Menit 47 Detik	6 kali	4 Menit	6 kali	4 Menit 30 Detik	6 kali	4 Menit 38 Detik	6 kali
Rata - Rata	3 Menit 39 Detik	5.3 kali	3 Menit 39 Detik	4.67 kali	4 Menit 30 Detik	7 kali	4 Menit 38 Detik	6.3 kali

Tabel 5.5 Menampilkan data hasil pengujian kecepatan pengguna dalam mengetik kalimat *“the quick brown fox jumps over the lazy dog”* dengan menggunakan 3 ukuran huruf keyboard. Didapatkan hasil bahwa perbedaan ukuran dari huruf keyboard mempengaruhi kecepatan dan kesalahan pengetikan dimana dengan ukuran skala 0,3 cm lebih cepat dan lebih efektif daripada ukuran yang lain.



Gambar 5.4 Hasil Pengujian Akurasi Berdasarkan Waktu



Gambar 5.5 Hasil Pengujian Akurasi Berdasarkan Error

Perbedaan hasil dalam ukuran waktu dikarenakan pergerakan dari huruf satu ke lainnya membutuhkan waktu, semakin jauh jarak antar huruf semakin lama pergerakan yang dilakukan sedangkan semakin dekat jarak huruf akan sebaliknya lebih cepat pergerakan yang dilakukan. Namun untuk error banyak terjadi ketika pada penulisan spasi dan waktu berhenti yang dilakukan kurang dari 3 detik sehingga banyak terjadi error. Akan tetapi pada ukuran 0,2 cm error banyak terjadi karena jarak antar huruf terlalu dekat sehingga mengakibatkan kesalahan pada saat memilih huruf.

Untuk hasil akurasi dari penggunaan 3 ukuran *keyboard*, didapatkan dari jumlah error yang terjadi dibandingkan dengan banyaknya kata yang diketik pada

saat melakukan pengetikan kalimat “*the quick brown fox jumps over the lazy dog*” yang berjumlah 42 karakter. Untuk nilai akurasi didapatkan dari jumlah karakter dikurangi dengan jumlah huruf lalu dibagi jumlah huruf dan dikalikan 100%. Data Pengujian akurasi ditampilkan pada Tabel 5.6.

Tabel 5.6 Data Pengujian Akurasi

Pengujian	Skala 0,2 cm		Skala 0,3 cm		Skala 0,4 cm		Skala 0,5 cm	
	Error	Akurasi	Error	Akurasi	Error	Akurasi	Error	Akurasi
1	5 kali	88 %	4 kali	90,4 %	7 kali	83 %	7 kali	83 %
2	6 kali	85%	4 kali	90,4 %	8 kali	80,9 %	6 kali	85 %
3	6 kali	85%	6 kali	85 %	6 kali	85%	6 kali	85 %
Rata – Rata	5.3 kali	86 %	4.67 kali	88,6 %	7 kali	82,9 %	6.3 kali	84,3 %

Dari hasil pengujian pada Tabel 5.6, dapat disimpulkan bahwa akurasi penggunaan ukuran huruf dengan skala 0,3 cm lebih tinggi dibandingkan dengan yang lainnya. Karena menggunakan ukuran skala 0,3 cm dapat mempercepat waktu melakukan pengetikan huruf, error yang terjadi lebih sedikit dibanding dengan ukuran lainnya dan tingkat akurasi lebih dari 85%.

5.4.2 Pengujian Usability

Pengujian Usability dalam penelitian ini bertujuan untuk mengetahui respon dari pengguna setelah menggunakan *keyboard virtual* yang mengimplementasikan teknik *free movement*. Pengujian ini dilakukan dengan mengambil 5 orang responden. Karena hasil terbaik berasal dari percobaan tidak lebih dari 5 orang (Nielsen, 2000). Responden merupakan mahasiswa Fakultas Ilmu Komputer yang sudah terbiasa menggunakan aplikasi berbasis *virtual reality*. Skenario dari pengujian ini Responden melakukan beberapa kegiatan dengan *keyboard virtual* dengan menggunakan 3 ukuran *keyboard* pada pengujian sebelumnya. Sebagai pengecualian untuk ukuran keyboard 0,2 cm tidak digunakan dalam pengujian ini dikarenakan ukuran keyboard terlalu kecil. Setelah melakukan percobaan responden mengisi pertanyaan-pertanyaan dari kuisioner yang digunakan dalam pengujian ini dengan nilai kisaran 1 untuk sangat tidak setuju sampai 9 untuk sangat setuju.

Berikut 8 pertanyaan yang dari penelitian dalam menguji usability dari aplikasi yang menerapkan *augmented reality* (Budhiraja,2013) seperti pada Tabel 5.6.

Tabel 5.6 Daftar Pertanyaan Pengujian Usability

No.	Pertanyaan
Q1	Aplikasi Berjalan dengan baik
Q2	Interface yang diberikan berguna untuk menyelesaikan pekerjaan

Q3	Interface yang diberikan mudah untuk digunakan
Q4	Interface yang diberikan mudah untuk dipelajari
Q5	Interface yang diberikan mudah untuk dipahami tanpa mengetahui sebelumnya
Q6	Interface yang diberikan bersifat natural
Q7	Interface yang diberikan tidak memberikan tekanan mental berupa tingkat stress yang tinggi
Q8	Interface yang diberikan tidak memberikan pengaruh pada fisik

5.4.2.1 Hasil Pengujian Usability

Dari hasil pengujian dengan 5 responden Mahasiswa Fakultas Ilmu Komputer yang terbiasa menggunakan aplikasi *virtual reality* didapatkan data pengujian menggunakan 3 ukuran *keyboard* seperti pada Tabel 5.7. Untuk Data lebih lengkap tentang hasil kuisioner yang pada responden terdapat pada Lampiran 1.

Tabel 5.7 Hasil Kuisioner Pengujian Usability

No.	Skala 0,3 cm	Skala 0,4 cm	Skala 0,5 cm
Q1	8.4	8	8.2
Q2	7.2	7	7.2
Q3	7.8	7.4	7.4
Q4	7.8	7.6	8
Q5	7.6	7.2	7
Q6	7.8	7.6	7.2
Q7	5.2	4.6	4
Q8	3.8	3.6	3.2

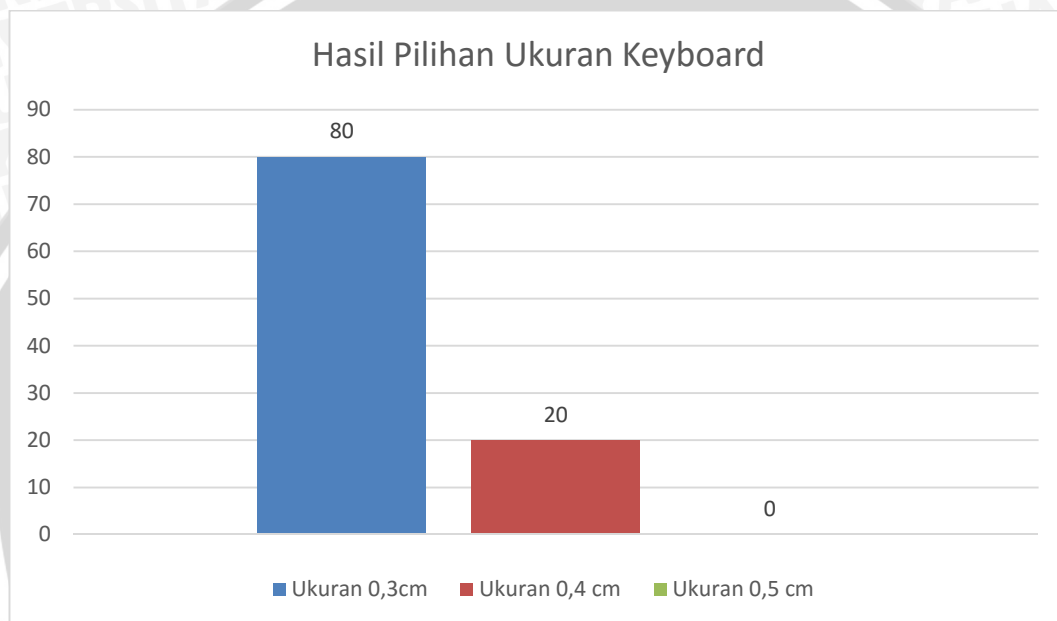
Setelah didapatkan data pengujian dari 5 responden tersebut Proses selanjutnya dilakukan perbandingan antar ketiga ukuran *keyboard* berdasarkan hasil rata-rata pengujian pada Tabel 5.7.

5.4.2.2 Analisis Hasil Pengujian

Dari Hasil Pengujian pada Tabel 5.7, bahwa rata-rata responden sangat setuju aplikasi dapat berjalan secara baik. Sementara itu dari segi interface yang diberikan, responden lebih setuju menggunakan skala ukuran 0,3 cm dimulai dari interface yang mudah digunakan, dan dapat digunakan dalam menyelesaikan sebuah pekerjaan. Sedangkan untuk interface yang diberikan kepada responden ukuran 0,5 cm lebih mudah untuk dipelajari. Dari hasil analisis berupa interface yang diberikan kepada responden didapatkan hasil bahwa ukuran 0,3 cm memiliki nilai yang besar dibandingkan dengan ukuran lainnya.

Namun dari segi pengaruh mental (Q7) rata-rata responden kurang setuju dengan beberapa ukuran yang ada. Dari hasil analisis pengaruh baik fisik maupun mental rata-rata responden ukuran 0,5 cm lebih memiliki pengaruh yang besar dikarenakan rata-rata nilai dari Q7 dan Q8 lebih kecil dari pada nilai pada ukuran yang lain.

Setelah melakukan analisis dari hasil perbandingan pada Tabel 5.7. Responden diminta untuk memilih ukuran berapa yang lebih disukai. Rata-rata dari 5 responden yang memilih ukuran 0,3 cm 80 persen, sedangkan untuk ukuran 0,4 cm 20 persen dan 0 persen untuk skala 0,5 cm seperti yang ditampilkan pada Gambar 5.5.



Gambar 5.5 Hasil Pilihan Ukuran Keyboard

Mengacu dari analisis hasil pengujian yang telah dilakukan bahwa rata-rata keyboard dengan ukuran 0,3 cm memiliki banyak kelebihan dibanding yang lainnya seperti lebih mudah digunakan, mempercepat pengerjaan tugas dan lebih memiliki tingkat kenyamanan bagi penggunaanya dan tidak mempengaruhi secara mental maupun fisik.

BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan dari hasil perancangan, implementasi dan pengujian didapatkan beberapa kesimpulan diantaranya :

1. Rancangan kendali *keyboard virtual* hemocs dengan teknik *free movement* dapat dilakukan dengan menggunakan sensor-sensor internal yang ada pada perangkat bergerak. Pergerakan dari sensor tersebut dalam teknik *free movement* ditunjang dengan adanya pointer yang memudahkan *user* bergerak.
2. Kendali *keyboard virtual* hemocs dengan teknik *free movement* dapat diimplementasikan pada pengetikan nama atau kalimat dengan menggunakan 3 ukuran skala *keyboard*.
3. Dari analisis hasil yang telah dilakukan, rata-rata dari 3 ukuran *keyboard* yang digunakan dalam pengujian memiliki tingkat akurasi lebih dari 80%.
4. Dalam pengujian usability didapatkan hasil bahwa rata-rata responden lebih menyukai ukuran skala 0,3 cm, dikarenakan ukuran tersebut lebih banyak disukai oleh responden baik dari tingkat kenyamanan maupun mempercepat pengerjaan tugas dan tingkat kenyamanan dari ukuran 0,3 cm dari segi fisik maupun mental.

6.2 Saran

Saran yang nantinya dapat digunakan dalam pengembangan penggunaan teknik *free movement* pada *keyboard virtual* ini yaitu:

1. *Keyboard virtual* dengan teknik *free movement* ini dapat digunakan dalam game berbasis *virtual reality*, misalkan pada saat melakukan penulisan nama high score ataupun game yang tidak menggerakkan tangan.
2. Dapat diimplementasikan aplikasi berbentuk *Augmented reality* yang menggunakan text-to-speech.

DAFTAR PUSTAKA

- Allan, A., 2011. *Basic Sensor in iOS*. Sebastopol: O'Reilly Media.
- Arai, K., 2013. Wearable Computing System with Input-Output Devices Based on Eye-Based *Human Computer Interaction* Allowing Location Based Web Services. *International Journal of Advanced Research in Artificial Intelligence*, 2(8), page 34-39. Tersedia melalui: <<http://www.ijarai.thesai.org>> [Diakses 25 Januari 2016].
- Arai, K., Tolle, H. & Serita, A., 2013. Mobile Devices Based 3D Image Display Depending on *User's* Actions and Movements. *International Journal of Advanced Computer Science and Application*, 2(6), page 71-78. Tersedia melalui: <<http://www.ijarai.thesai.org>> [Diakses 22 Januari 2016].
- Budhiraja, R., Lee, G.A., Billingham, M., 2014. Using a HHD with a HMD for Mobile AR Interaction. *International Symposium on Mixed and Augmented reality*. [Diakses 26 Juli 2016].
- Mann, S., The Encyclopedia of Human-Computer Interaction, 2nd. Tersedia di : Interaction Design Foundation <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/wearable-computing> [diakses 1 Februari 2016].
- Nielsen, J., *Why You Only Need to Test with 5 Users*. Nielsen Norman Group. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> [Diakses 24 Agustus 2016]
- Oktagon, T. D., *Tren Wearable Device Semakin Memudahkan Gaya Hidup Digital di 2016*. <http://daily.oktagon.co.id/tren-wearable-device-semakin-memudahkan-gaya-hidup-digital-di-2016/> [Diakses 24 Agustus 2016].
- Lancioni, G.E., Singh, N.N., O'Reilly, M.F., Sigafoos, J., Green, V., Chiapparino, C., Stasolla, F. & Oliva, D., 2009. A voice-detecting sensor and a scanning keyboard emulator to support word writing by two boys with extensive motor disabilities. *Research in Developmental Disabilities*, [E-Journal] 30(2009). Tersedia Melalui: ScienceDirect <<http://www.sciencedirect.com>> [Diakses 22 Januari 2016].
- Safii, I. 2016. *Pengembangan Metode Pendeteksian Pergerakan Kepala Berbasis Sensor Internal Pada Perangkat Bergerak Berbasis iOS*. S1. Universitas Brawijaya.
- Tolle, H., Pinandito, A., Adams J, E.M. & Arai, K., 2015. *Virtual reality* Game Controlled With *User's* Head And Body Movement Detection Using Smartphone Sensors. *ARPN Journal Engineering and Applied Sciences*, 10(20), page 9776-9782. Tersedia melalui: ARPN <<http://www.arpnjournals.com>> [Diakses 03 Februari 2016].

Wikipedia, 2015. *Augmented reality*. [Online] Diperbarui 17 Oktober 2014, Tersedia di: <https://id.wikipedia.org/wiki/Realitas_tetambah> [Diakses 03 Februari 2016].



LAMPIRAN

Lampiran A Data Kuisiener Pengujian Usability

1. Skala 0,3 cm

No.	R1	R2	R3	R4	R5
Q1	8	9	8	8	9
Q2	7	8	7	7	7
Q3	7	8	7	8	9
Q4	7	8	7	8	9
Q5	7	8	7	8	8
Q6	7	8	7	8	9
Q7	7	2	7	2	1
Q8	7	4	7	7	1

2. Skala 0,4 cm

No.	R1	R2	R3	R4	R5
Q1	8	9	6	8	9
Q2	7	7	6	8	7
Q3	7	8	6	8	8
Q4	7	8	6	8	9
Q5	7	8	6	7	8
Q6	7	8	7	7	9
Q7	7	4	6	2	3
Q8	7	5	6	6	3

3. Skala 0,5 cm

No.	R1	R2	R3	R4	R5
Q1	8	9	8	7	9
Q2	8	6	8	7	7
Q3	7	8	8	7	7
Q4	8	8	7	8	9
Q5	7	8	5	7	8
Q6	7	8	5	7	9
Q7	8	5	5	2	5
Q8	7	6	5	6	5

Keterangan :**R1 : Responden Pertama****R2 : Responden Kedua****R3 : Responden Ketiga****R4 : Responden Keempat****R5 : Responden Kelima**