



PROYEK AKHIR - VE180626

**SENSOR GYROSCOPE PADA SIMULASI VIRTUAL GAME SHOOTER
MENGUNAKAN UNITY BERBASIS ARDUINO**

Syaiful Huda
NRP 1031150000083

Dosen Pembimbing
Slamet Budiprayitno, S.T., M.T.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



ITS
Institut
Teknologi
Sepuluh Nopember

PROYEK AKHIR - VE180626

SENSOR GYROSCOPE PADA SIMULASI *VIRTUAL GAME SHOOTER* MENGGUNAKAN UNITY BERBASIS ARDUINO

Syaiful Huda
NRP 1031150000083

Dosen Pembimbing
Slamet Budiprayitno, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - VE180626

GYROSCOPE SENSOR IN ARDUINO-BASED VIRTUAL SHOOTER GAME SIMULATION USING UNITY

Syaiful Huda
NRP 10311500000083

Advisor
Slamet Budiprayitno, ST., MT.

DEPARTEMENT OF ELECTRICAL AUTOMATION ENGINEERING
Faculty of Vocations
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN PROYEK AKHIR

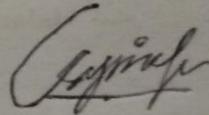
Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul "**Sensor Gyroscope Pada Simulasi *Virtual Game Shooter* Menggunakan Unity Berbasis Arduino**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 29 Januari 2019

Mahasiswa



Syaiful Huda

NRP 1031150000083

-----Halaman ini sengaja dikosongkan-----

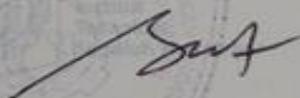
SENSOR GYROSCOPE PADA SIMULASI *VIRTUAL GAME SHOOTER* MENGGUNAKAN UNITY BERBASIS ARDUINO

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departement Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Slamet Budiprayitno, ST., MT.
NIP. 1978 11 13 2010 12 1 002

**SURABAYA
JANUARI, 2019**

-----Halaman ini sengaja dikosongkan-----

SENSOR GYROSCOPE PADA SIMULASI *VIRTUAL GAME SHOOTER* MENGGUNAKAN UNITY BERBASIS ARDUINO

Nama : Syaiful Huda
Pembimbing : Slamet Budiprayitno, ST., MT.

ABSTRAK

Virtual Reality (VR) adalah teknologi yang membuat manusia dapat berinteraksi dengan suatu lingkungan yang disimulasikan oleh komputer (*computer-simulated environment*) dan membuat pengguna merasakan seperti berada di dalam dunia maya. Teknologi VR ini digunakan oleh arsitek, pilot pesawat, pekerja medis, dan gamer yang menyukai sensasi bermain game yang nyata. Teknologi virtual reality juga memberikan banyak kontribusi terhadap dunia militer yang ada di Amerika Serikat. Tetapi di negara berkembang saat ini belum ada militer yang dapat membelinya dikarenakan mahalnya alat ini dan di Indonesia sendiri belum ada yang membuatnya.

Oleh karena itu dibutuhkan alat virtual reality pertama di Indonesia yang dikhususkan untuk militer. Alat itu ialah *Virtual Game Shooter* yang memerlukan sensor gyroscope dan kacamata khusus VR, yaitu VR Box. Sensor gyroscope dipasangkan pada senjata mainan sebagai controller menggerak-gerakkan senjata yang ada di dunia virtual sehingga pengguna merasa seperti sedang menebak sasaran tembak sungguhan. Untuk tampilan lingkungan *shooting* nya, senjata, sasaran tembak, peluru dan lain-lain dibuat dengan 3D menggunakan aplikasi yang bernama Unity. Dan diharapkan alat ini dapat dijadikan latihan militer di Indonesia.

Kata Kunci : *virtual reality*, *gyroscope* dan *unity*

-----Halaman ini sengaja dikosongkan-----

GYROSCOPE SENSOR IN ARDUINO-BASED VIRTUAL SHOOTER GAME SIMULATION USING UNITY

Name : Syaiful Huda
Advisor : Slamet Budiprayitno, ST., MT.

ABSTRACT

Virtual Reality (VR) is a technology that allows humans to be used with a computer-simulated environment (computer simulation environment) and make users like in cyberspace. This VR technology is used by pilots, medical workers, and gamers who play real games. Virtual reality technology also provides a lot of insight into the military world in the United States. However, in developing countries, currently, there is a military that can replace it with the high cost of this tool and no one has made it in Indonesia itself.

Therefore the first virtual reality tool in Indonesia is needed specifically for the military. The tool is a Virtual Game Shooter that demands a gyroscope sensor and special VR glasses, the VR Box. The gyroscope sensor is attached to a toy weapon as a motion controller in the virtual world. To display shooting environments, weapons, shooting targets, bullets and others made with 3D using an application called Unity. And hope this tool can shape military work in Indonesia.

Keywords : *virtual reality, gyroscope and unity*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Alhamdulillah kami panjatkan kepada Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan kemudahan dari Nya, hingga kami dapat menyelesaikan Proyek Akhir ini dengan baik, begitu pula dengan pembuatan buku Proyek Akhir ini.

Proyek Akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro Otomasi dengan judul :

SENSOR GYROSCOPE PADA SIMULASI *VIRTUAL GAME SHOOTER* MENGGUNAKAN UNITY BERBASIS ARDUINO

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Slamet Budiprayitno, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Proyek Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 29 Januari 2019

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN PROYEK AKHIR.....	vii
HALAMAN PENGESAHAN.....	ix
ABSTRAK.....	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL.....	xxii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Laporan.....	3
1.7 Relevansi.....	4
BAB II TEORI DASAR.....	5
2.1 Arduino.....	5
2.1.1 Konfigurasi Pin Arduino Pro Micro.....	6
2.1.2 Pin Power Arduino Pro Micro.....	7
2.1.3 I/O Arduino Pro Micro.....	7
2.2 <i>Virtual Reality</i>	8
2.3 <i>Unity</i>	9
2.3.1 Lisensi <i>Unity</i>	9
2.3.2 <i>Interface Unity</i>	9
2.4 Game.....	10
2.5 <i>FPS Controller</i>	10
2.6 <i>Gyroscope</i>	11
2.7 <i>HID</i>	12
2.8 <i>Komunikasi Wireless</i>	12
2.9 <i>Senjata Mainan</i>	14
2.10 <i>Pull Up Resistor</i>	14
BAB III PERANCANGAN DAN PEMBUATAN ALAT.....	15

3.1	Blok Fungsional Sistem	15
3.2	Perancangan Hardware	15
3.2.1	Perancangan Shield Mikrokontroler	16
3.2.2	Rangkaian Wireless dengan Arduino	17
3.3	Perancangan Software	18
3.3.1	Flowchart	18
3.3.2	Rancangan Program Arduino	19
3.3.2.1	Program <i>Bluetooth</i>	19
3.3.2.1	Program <i>Gyroscope</i>	20
BAB IV PENGUJIAN DAN ANALISA		23
4.1	Pembuatan Software Game	23
4.1.1	File New Project	23
4.1.2	Window Utama Unity	24
4.1.2.1	<i>Scene</i>	24
4.1.2.2	<i>Game View</i>	26
4.1.2.3	<i>Hierarchy Window</i>	27
4.1.2.4	<i>Project Window</i>	27
4.2	Pengujian Game	28
4.2.1	Pengujian Rigidbody	28
4.2.2	Uji Coba Menembak di Unity	31
4.2.3	Uji Coba Menyimpan <i>Game</i>	32
4.3	Pengujian Sensor <i>Gyroscope</i>	33
4.3.1	Pengujian Sensor <i>Gyroscope</i> Lewat Serial Monitor	34
4.3.2	Pengujian Sensor saat Diputar Berapa Derajat di Komputer	34
4.4	Pengujian <i>Bluetooth</i>	36
4.4.1	Pengujian AT Command	36
4.4.2	Pengujian Pairing Bluetooth	37
4.5	Pengujian Alat Keseluruhan	38
4.5.1	Pengujian Pergerakan Layar Game	38
4.5.2	Pengujian Menembak Game dengan Alat	39
BAB V PENUTUP		41
5.1	Kesimpulan	41
5.2	Saran	41
DAFTAR PUSTAKA		43
LAMPIRAN A		45

LAMPIRAN B	47
LAMPIRAN C	55
DAFTAR RIWAYAT HIDUP	57

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Konfigurasi Pin Arduino Pro Micro.....	6
Gambar 2.2	Unity	10
Gambar 2.3	FPS.....	11
Gambar 2.4	Bluetooth.....	13
Gambar 2.5	Senjata Mainan	14
Gambar 2.6	Pull Up Resistor	14
Gambar 3.1	Blok Diagram.....	15
Gambar 3.2	Rangkaian Shield Mikrokontroler.....	16
Gambar 3.3	Sambungan Arduino dengan Bluetooth HC-05.....	17
Gambar 3.4	Rancangan <i>Flowchart</i> Keseluruhan.....	19
Gambar 3.5	Program Serial Bluetooth.....	20
Gambar 3.6	Program Groscope	21
Gambar 4.1	Tampilan Pertama <i>Unity</i>	23
Gambar 4.2	<i>Project Unity</i>	24
Gambar 4.3	Memasukkan <i>Cube 3D</i>	25
Gambar 4.4	Mengedrag <i>Textures Object</i>	25
Gambar 4.5	<i>Box, Target, dan Wall</i>	25
Gambar 4.6	<i>FPS Character</i>	26
Gambar 4.7	Prefabs Handgun.....	26
Gambar 4.8	Game View	27
Gambar 4.9	<i>Project Window</i>	27
Gambar 4.10	Posisi, Rotasi, dan Skala (Objek Lantai).....	31
Gambar 4.11	Uji Coba <i>Rigidbody</i> pada Objek	31
Gambar 4.12	Ketika Terjadinya Error.....	33
Gambar 4.13	Pengujian Alat Menggunakan Busur.....	36
Gambar 4.14	Serial Monitor Arduino.....	36
Gambar 4.15	Serial Monitor Arduino 2.....	37

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

HALAMAN

Tabel 3.1 Konfigurasi Pin Arduino ke Modul-Modul.....	16
Tabel 3.2 Konfigurasi Pin HC-05 dengan Arduino.....	17
Tabel 4.1 Objek Yang Terdorong	28
Tabel 4.2 Pengaruhnya Ketinggian.....	29
Tabel 4.3 Uji Coba Menembak di Unity	32
Tabel 4.4 Perbedaan Platform Android dengan Platform PC.....	33
Tabel 4.5 Pengujian Sensor Gyroscope Pada Saat Diam	34
Tabel 4.6 Pengujian Sensor Saat digerakkan Searah Sumbu X.....	34
Tabel 4.7 Pengujian Sensor Saat digerakkan Searah Sumbu Y.....	35
Tabel 4.8 Uji Coba Pergerakan Layar Game Sumbu X	38
Tabel 4.9 Uji Coba Pergerakan Layar Game Sumbu Y	38
Tabel 4.10 Pengujian Menembak Game dengan Alat.....	39

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi semakin berkembang tiap waktunya dari dulu hingga sekarang dan kedepannya akan lebih berkembang lagi. *Virtual Reality* (VR) atau realitas maya adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan komputer (*computer-simulated environment*), suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi. Lingkungan virtual reality terkini umumnya menyajikan pengalaman visual (ex; *game* VR, Video, simulasi, dll), yang ditampilkan pada sebuah layar komputer atau melalui sebuah penampil seperti android. Tetapi beberapa simulasi mengikutsertakan tambahan informasi hasil pengindraan, seperti suara melalui speaker atau headphone.

Di VR yang umum saat ini adalah *game*/simulasi virtual. Biasanya untuk membuat *game* diperlukan aplikasi yang namanya Unity. Dia sebuah *tool* yang terintegrasi membuat tampilan-tampilan *game*, arsitektur bangunan, bahkan simulasi bentuk orang, dan lain lain. Setelah *game* itu sudah siap maka disimpan dalam bentuk yang diinginkan yaitu instalasi aplikasi android (apk) dan diinstal di android. HP android dimasukkan ke dalam *Googles* VR Box (kacamata khusus VR) kemudian mata didekatkan dan tampilan yang dibuat oleh Unity dapat terlihat dengan jelas. Dikarenakan tujuan saya membuat alat ini adalah membuat simulasi latihan shooting (tembak-menembak), maka komponen seperti sensor *gyroscope*, *bluetooth*, arduino di jadikan satu terus bisa ditempelkan pada senjata buatan atau senjata mainan seperti hal nya *joystick*.

Untuk program pengkalibrasian antara sensor dengan bluetooth menggunakan Arduino. Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Sensor *gyroscope* mempunyai gerakan *motion* mengirim data serial lewat *bluetooth* menuju android sehingga senjata yang telah digerak-gerakkan ataupun ditembakkan pelurunya bisa bekerja seperti dunia asli.

1.2 Permasalahan

Dalam halnya simulasi virtual sudah banyak alat yang bisa untuk berinteraksi secara langsung antara pengguna dengan komputer. Namun masih belum adanya alat di Indonesia yang disecara khususkan untuk penggunaan dalam kemiliteran. Makanya diperlukan alat yang bisa menunjang latihan kemiliteran Indonesia. Adapun pertanyaan penelitian pada Proyek Akhir ini adalah sebagai berikut:

1. Bagaimana cara menyinkronkan sensor *gyroscope* dengan simulasi game yang sedang dibuat ?
2. Bagaimana cara mengkoneksikan dari sensor menuju ke android?
3. Bagaimana cara membuat game simulasi dari aplikasi Unity?

1.3 Batasan Masalah

Didalam pembuatan Proyek Akhir ini terdapat batasan masalah yang meliputi:

1. Senjata di virtual game tidak ada fitur mereload/isi peluru.
2. Senjata di virtual game tidak ada fitur untuk mengganti senjata lain.
3. Senjata di virtual game tidak ada fitur angin/jarak antara peluru yang keluar dari senjata dengan sasaran tembakan tidak terpengaruh. Sehingga selalu lurus arah tembakannya.
4. Tidak ada sasaran yang terbuat dari manusia.
5. Jika peluru mengenai sasaran, maka tidak ada bekas tembakan.
6. Tidak ada suara tembakan didalamnya.

1.4 Tujuan

Tujuan utama Proyek Akhir ini adalah untuk merencanakan dan membangun sebuah alat yang simulasikan lapangan tembak yang ada di game. Dalam uraian tersebut, maka dapat dibagi menjadi tiga tujuan dalam Proyek akhir ini, yaitu:

1. Membuat game virtual shooter yang simple semua pengguna dapat mempelajarinya, khususnya dalam kemiliteran sebagai latihan tembak maupun dijadikan menjadi sebuah game permainan shooter.
2. Membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan computer.
3. Membuat komunikasi yang menjembantani antara sensor yang digunakan dengan game.

1.5 Metodologi Penelitian

Dalam pembuatan Proyek Akhir perancangan perangkat keras sensor *gyroscope* pada simulasi *virtual game shooter* menggunakan unity berbasis arduino, ada beberapa tahap kegiatan yaitu meliputi tahap persiapan (*study literature*), tahap perencanaan dan pembuatan alat, tahap pengujian dan analisa, serta penyusunan laporan.

Pada tahap studi literatur akan dipelajari mengenai konsep dasar untuk kontrol sensor *gyroscope* dengan arduino, mempelajari konsep untuk pembuatan game tersebut di Unity. Mempelajari cara mengkoneksikan antara game unity dengan sensor.

Tahap perancangan *hardware* dan *software* meliputi perancangan rangkaian sensor *gyroscope* dengan arduino yang dipasangkan pada senjata mainan, senjata mainan ini dijadikan sebagai kontroler pergerakan virtual yang ada di virtual game shooter tersebut. Kemudian tahap pembuatan game shooter nya akan dilakukan pemberian karakter FPS (*First Person Shooting*) untuk pergerakan arah ke atas ke bawah kiri kanan seperti halnya kita kalau ingin menggerakkan kursor *mouse*. Nah pergerakan kursor mouse tersebut digantikan dengan kontroler arduino dan sensor *gyroscope* tersebut.

Setelah itu dilakukan pengujian alat, menganalisa kesalahan atau kegagalan serta data sensor yang didapat ketika digerakkan sesuai sumbu pada alat dan mengatasi permasalahan tersebut. Tahapan ini dilakukan dengan melakukan pengujian kemampuan sistem dalam menggerakkan senjata yang ada pada simulasi secara otomatis. Data hasil pengujian tersebut akan dianalisa kemudian mencari tahu faktor apa saja yang menyebabkan alat tidak bekerja sesuai dengan keinginan atau terjadi *error*. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka yang mendukung dalam perencanaan dan pembuatan alat.

Bab III Perancangan Sistem

Bab ini membahas mengenai desain perancangan rangkaian *hardware* maupun *software* yang digunakan.

Bab IV Simulasi, Implementasi dan Analisis Sistem

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dari Proyek Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.7 Relevansi

Dengan adanya Proyek Akhir ini diharapkan dapat diaplikasikan pada simulasi *shooting* sesungguhnya, serta dapat dijadikan sebagai referensi atau ide dalam kemajuan teknologi, khususnya *virtual reality*.

BAB II TEORI DASAR

2.1 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para *hobbyist* atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus beberapa kelebihannya.

Sederhana dan mudah pemrogramannya dan mudah digunakan untuk pemula, dan cukup fleksibel bagi mereka yang sudah tingkat lanjut. Untuk guru/dosen, Arduino berbasis pada lingkungan pemrograman Processing, sehingga jika mahasiswa atau murid-murid terbiasa menggunakan Processing tentu saja akan mudah menggunakan Arduino.

Perangkat lunaknya *Open Source* – Perangkat lunak Arduino IDE dipublikasikan sebagai *Open Source*, tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut. Bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada Bahasa C untuk AVR.

Input/output digital atau *digital pin* adalah pin pin untuk menghubungkan arduino dengan komponen atau rangkaian digital. contohnya, jika ingin membuat LED berkedip, LED tersebut bisa dipasang pada salah satu pin *input* atau *output* digital dan *ground*. komponen lain yang menghasilkan *output digital* atau menerima *input digital* bisa disambungkan ke pin pin ini. Input analog atau *analog pin* adalah pin pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian analog. contohnya, potensiometer, sensor suhu, sensor cahaya, dll.

Pro Micro mirip dengan Pro Mini kecuali dengan papan ATmega32U4. *Transceiver* USB di dalam 32U4 memungkinkan kita

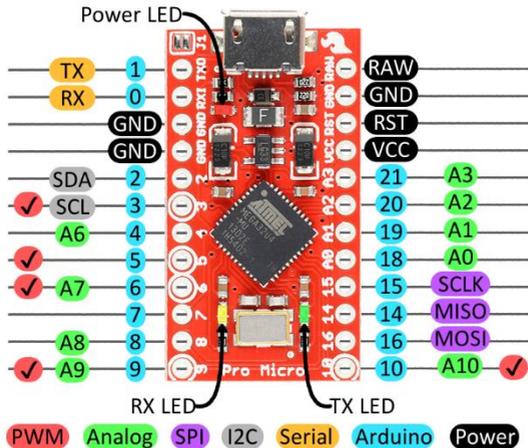
untuk menambahkan konektivitas USB *on-board* dan menghapus dengan antarmuka USB eksternal yang besar.

Papan kecil ini melakukan semua trik Arduino rapi yang Anda kenal: 9 saluran ADC 10-bit, 5 pin PWM, 12 DIO serta koneksi seri perangkat keras Rx dan Tx. Berjalan pada 16MHz dan 5V, board ini akan mengingatkan Anda banyak papan Arduino favorit Anda yang lain tetapi orang kecil ini bisa pergi ke mana saja. Ada regulator tegangan di papan sehingga dapat menerima tegangan hingga 12VDC. Jika Anda memasok kekuatan yang tidak diatur ke papan, pastikan untuk menyambungkan ke pin “RAW” di bukan VCC.

Revisi terbaru ini mengoreksi kesalahan sutra dari versi terakhir papan sehingga pin 14 diberi label dengan benar. Juga ditambahkan perlindungan sekering PTC dan dioda ke sirkuit daya dan memperbaiki sirkuit RX dan TX LED.

2.1.1 Konfigurasi Pin Arduino Pro Micro

Semua I / O dan pin daya Pro Micro dibagi menjadi dua, header paralel. Beberapa pin untuk input atau output daya, pin lainnya didedikasikan pin I / O. Selanjutnya, pin I / O dapat memiliki kemampuan khusus, seperti input analog. Berikut adalah peta pin mana dan di mana, dan fungsi perangkat keras khusus apa yang dimilikinya:



Gambar 2.1 Konfigurasi Pin

Sumber: learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-pro-micro

2.1.2 Pin Power Arduino Pro Micro

Ada berbagai aliran listrik dan yang terkait dengan daya yang pecah:

- a. RAW adalah input tegangan yang tidak diatur untuk Pro Micro. Jika papan didukung melalui USB, tegangan pada pin ini akan menjadi sekitar 4.8V (5V USB dikurangi penurunan dioda schottkey). Di sisi lain, jika papan diberi daya secara eksternal, melalui pin ini, tegangan yang diberikan bisa mencapai 12V.
- b. VCC adalah tegangan yang dipasok ke ATmega32U4 on-board. Tegangan ini akan bergantung pada apakah Anda menggunakan versi ProV 3.3V / 8MHz atau versi 5V / 16MHz, masing-masing akan menjadi 3.3V atau 5V. Tegangan ini diatur oleh tegangan yang diterapkan pada pin RAW. Jika papan diberi daya melalui pin 'RAW' (atau USB), pin ini dapat digunakan sebagai output untuk memasok perangkat lain.
- c. RST dapat digunakan untuk me-restart Pro Micro. Pin ini ditarik tinggi 10k & Ohm; resistor di papan, dan aktif-rendah, sehingga harus terhubung ke ground untuk memulai reset. Pro Micro akan tetap "mati" sampai garis reset ditarik kembali ke tinggi.
- d. GND, tentu saja adalah tegangan ground (referensi 0V) untuk sistem.

2.1.3 I/O Arduino Pro Micro

- a. Pin I / O Pro Micro - semuanya berjumlah 18 yang memiliki banyak talenta. Setiap pin dapat digunakan sebagai *input* atau *output digital*, untuk LED berkedip atau membaca tombol yang ditekan. Pin ini direferensikan dalam IDE Arduino melalui nilai *integer* antara 0 dan 21. (Pin A0-A3 dapat direferensikan secara *digital* menggunakan nomor pin *analog* atau *digital*).
- b. Fitur Sembilan pin fitur konverter *analog* ke *digital* (ADC) dan dapat digunakan sebagai *input analog*. Ini berguna untuk membaca potensiometer atau perangkat analog lainnya menggunakan fungsi `analogRead ([pin])`.
- c. Ada lima pin dengan fungsi *Pulse Width Modulation* (PWM), yang memungkinkan untuk bentuk *output analog* menggunakan fungsi `analogWrite ([pin], [value])`. Pin ini ditunjukkan di papan dengan lingkaran putih samar di sekitarnya.

- d. Ada perangkat keras UART (*serial*), I2C, dan pin SPI juga tersedia. Ini dapat digunakan untuk berinteraksi dengan perangkat digital seperti LCD *serial*, XBees, IMU, dan sensor *serial* lainnya.
- e. Pro Micro memiliki lima interupsi eksternal, yang memungkinkan Anda untuk secara instan memicu suatu fungsi ketika pin menjadi tinggi atau rendah (atau keduanya). Jika Anda memasang interupsi ke pin yang diaktifkan interupsi, Anda harus mengetahui interupsi spesifik yang dipicu pin: pin 3 untuk menginterupsi 0, pin 2 menginterupsi 1, pin 0 menginterupsi 2, pin 1 menginterupsi 3, dan pin 7 menginterupsi 4.

2.2 *Virtual Reality*

Virtual Reality (VR) atau realitas maya adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan komputer (*computer-simulated environment*), suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi. Lingkungan realitas maya terkini umumnya menyajikan pengalaman *visual*, yang ditampilkan pada sebuah layar komputer, tetapi beberapa simulasi mengikutsertakan tambahan informasi hasil pengindraan, seperti suara melalui *speaker* atau *headphone*.

Beberapa sistem *haptic* canggih sekarang meliputi informasi sentuh, biasanya dikenal sebagai umpan balik kekuatan pada aplikasi berjudi dan medis. Para pemakai dapat saling berhubungan dengan suatu lingkungan sebetulnya atau sebuah artifak maya baik melalui penggunaan alat masukan baku seperti a papan ketik dan tetikus, atau melalui alat multimodal seperti a sarung tangan terkabel, dan ban jalan segala arah. Lingkungan yang ditirukan dapat menjadi mirip dengan dunia nyata, sebagai contoh, simulasi untuk pilot atau pelatihan pertempuran, atau dapat sangat berbeda dengan kenyataan, seperti di VR *game*. Dalam praktik, sekarang ini sangat sukar untuk menciptakan pengalaman Realitas maya dengan kejernihan tinggi, karena keterbatasan teknis atas daya proses, resolusi citra dan lebar pita komunikasi. Bagaimanapun, pembatasan itu diharapkan untuk secepatnya diatasi dengan berkembangnya pengolah, pencitraan dan teknologi komunikasi data yang menjadi lebih hemat biaya dan lebih kuat dari waktu ke waktu.

2.3 Unity

Game Engine bisa didefinisikan sebagai perangkat lunak yang dirancang secara khusus untuk membuat sebuah game. *Game engine* umumnya terdiri dari dua fungsi dasar yang berkaitan dengan *Asset* (gambar, suara, dan film) dan *Logic* (fungsi-fungsi interaksi).

Unity ini memungkinkan seseorang atau tim untuk membuat sebuah *game* 3D dengan mudah dan cepat. Secara default, unity telah diatur untuk pembuatan game bergenre *First Person Shooting* (FPS), namun unity juga bisa digunakan untuk membuat *game* bergenre *Role Playing Game* (RPG), dan *Real Time Strategy* (RTS). Selain itu, unity merupakan sebuah *engine multiplatform* yang memungkinkan game yang anda bangun dipublish untuk berbagai *platform* seperti *Windows, Mac, Android, IOS, PS3* dan juga *Wii*. Unity bisa didownload secara gratis di <http://unity3d.com>.

2.3.1 Lisensi Unity

Untuk mengeluarkan 3 jenis lisensi yaitu : *Personal, Plus, dan Pro*. *Unity Personal* merupakan lisensi gratis yang bisa digunakan siapapun untuk membuat game gratis. *Unity Plus* memiliki lebih banyak fitur dihargai dengan harga \$35 per bulan. Sementara *Unity Pro* dibutuhkan minimal harga \$125 per bulan dengan fitur yang lebih banyak dari *Plus*.

2.3.2 Interface Unity

Tampilan awal Unity terdiri dari 9 tab utama yaitu: *Project, Console, Animation, Hierarchy, Scene, Game, Animator, Inspector, dan Services*. Dalam hal ini, *Tab Project, Console, dan Animation* yang berada dibawah merupakan tab di kelompok pertama yang harus diperhatikan karena pada kelompok inilah akan dimulai bekerja dengan memasukkan *Assets* berupa gambar, suara, atau bahkan *script*.

Selanjutnya, *Tab Hierarchy, Scene, Game, dan Animator* menjadi kelompok kedua yang harus diperhatikan. Karena pada kelompok inilah Anda akan mulai bekerja merancang game yang akan dibuat termasuk menentukan pergerakan animasi. Kelompok terakhir adalah *Tab Inspector dan Services* yang bertanggung jawab untuk melakukan pengaturan lebih rinci pada *game* yang dibuat serta pengaturan lainnya.



Gambar 2.2 *Unity*
Sumber: unity3d.com

2.4 *Game*

Game adalah sesuatu yang digunakan untuk bermain atau sesuatu yang dipergunakan. Setiap permainan terdapat alat dan aturan-aturan, sehingga pemain akan membutuhkan keterampilan strategi, kesempatan, ataupun keberuntungan. *Game* dapat dilakukan dengan dimainkan secara *multiple players* atau *single player*. *Game* dengan *single player* adalah *game* yang memiliki jenis tantangan yang unik, dimana pemain akan menggunakan keterampilannya sendiri untuk melawan waktu /kemungkinan.

2.5 *FPS (First Person Shooter)*

FPS disini bukanlah *Frame Per Second* tetapi sebuah genre game disebut *first person shooter*. FPS adalah *genre* permainan video yang ciri utamanya adalah penggunaan sudut pandang orang pertama dengan tampilan layar yang mensimulasikan apa yang dilihat melalui mata karakter yang dimainkan. Ciri utama lain adalah penggunaan senjata genggam jarak jauh.



Gambar 2.3 FPS

Sumber: www.rockpapershotgun.com/2018/05/21/best-fps-2018/1/

2.6 Gyroscope

Gyroscope adalah perangkat untuk mengukur atau mempertahankan orientasi, dengan prinsip ketetapan momentum sudut, alat ini bekerja sama dengan *accelerometer*. Mekanismenya adalah sebuah roda berputar dengan piringan didalamnya yang tetap stabil. Alat ini sering digunakan pada robot atau *drone* serta alat-alat canggih lainnya.

Pada *gyroscope* terdapat *gyro* sensor untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat pada sumbu. *Gyro* sensor sendiri memiliki fungsi untuk mendeteksi gerakan sesuai gravitasi, atau dengan kata lain mendeteksi gerakan pengguna.

Ketika *gyroscope* bergerak maka akan menghasilkan tegangan output. Pada saat *gyroscope* diam maka tegangan akan konstan. Ketika *gyroscope* berputar searah jarum jam terhadap sumbu Z maka tegangan output berkurang ($- Z$), dan ketika *gyroscope* berputar berlawanan arah jarum jam maka tegangan output bertambah ($+ Z$).

2.7 HID

HID (*Human Interface Devices*) adalah metode di mana manusia berinteraksi dengan sistem informasi elektronik baik dengan memasukkan data atau memberikan output. Ada segudang perangkat HID. Yang paling umum adalah keyboard, mouse, speaker komputer, webcam dan headset. Semua perangkat yang menyediakan antarmuka antara pengguna dan mesin komputer dianggap sebagai HID.

1. Keyboard dan perangkat penunjuk (seperti perangkat *mouse*, *joystick* dan *trackball*)
2. Kontrol panel depan (misalnya tombol, kenop, penggeser, dan sakelar)
3. Perangkat simulasi (seperti roda kemudi, pedal, perangkat *input* VR lainnya)
4. Kontrol jarak jauh dan *keypad* telepon
5. Perangkat laju data rendah lainnya yang menyediakan misalnya data lingkungan (seperti termometer, meter energi atau bahkan pembaca kode batang)

2.8 Komunikasi Wireless

Wireless Communication atau dalam bahasa Indonesia di biasa dikenal dengan istilah Telekomunikasi Nirkabel adalah transfer informasi antara dua atau lebih titik yang tidak terhubung secara fisik. Jarak bisa pendek, seperti beberapa meter untuk remote control televisi, atau sejauh ribuan atau bahkan jutaan kilometer untuk ruang-dalam komunikasi radio. Ini meliputi berbagai jenis tetap, *mobile*, dan portabel radio dua arah, telepon seluler, dan jaringan nirkabel. Salah satu modul yang cocok untuk menghubungkan dua mikrokontroler secara *wireless* atau tanpa kabel, yakni *Bluetooth*.

Modul *Bluetooth* HC-05 6 Pin yang siap pakai untuk membuat proyek tersebut memiliki kemampuan berkomunikasi secara serial *wireless* dengan protokol standar *Bluetooth* v2.0. *support* utk *master* maupun *slave*. Spesifikasi:

1. Ukuran: 37.3mm x 15.5mm.
2. Tegangan input: 3.6V – 6V (jangan lebih dari 7v)
3. Terdiri dari 6 pin: EN / VCC / GND / RXD / TXD / STATE
4. Dengan indikator status koneksi
5. Dengan regulator *chip* 3.3V *On-board*, arus sekitar 30mA dalam keadaan *unpair*. Setelah *pairing* berhasil, arus menjadi sekitar 10mA.
6. *Interface level* 3.3V dapat langsung dihubungkan ke berbagai *microcontroller* (Arduino / 51 / AVR / PIC / ARM / MSP430 dll) termasuk *microcontroller* 5V. Langsung dihubungkan ke *port serial microcontroller*, tanpa perlu melalui *chip* MAX232 lagi.
7. Jarak transmisi efektif pada ruang terbuka 10m. Diatas 10m masih memungkinkan namun kualitas koneksi menurun.

8. Setelah pairing berhasil, dapat digunakan sebagai *full-duplex serial interfaces* tanpa perlu mengerti *protocol Bluetooth*.
9. Dapat dijadikan mode *master* atau *slave*
10. *Default baudrate*: 9600, *Data bit*: 8, *Stop bit*: 1, *Parity*: *No Parity*, *Password default*: 1234, *Nama default*: HC-05

Metode pemasangan kabel:

VCC: positif *power supply*.

GND: negatif *power supply*.

RXD: penerimaan data dari TXD peralatan lain.

TXD: pengiriman data ke RXD peralatan lain.

EN: *enabled client*, masuk ke *mode AT 3.3V*.



Gambar 2.4 *Bluetooth HC-05*

Sumber: www.arduinoso.com/product/hc-05-bluetooth-module/

2.9 Senjata Mainan

Senjata mainan ini dibuat sebagai perumpamaan senjata yang ada di simulasi game virtual. Dengan baterai yang sudah terpasang pada senjata mainan maka bisa menyuplai sekitar 6V menuju pin “RAW” arduino pro micro. Disenjata mainan ini terdapat trigger seperti push button agar disambungkan ke salah satu pin digital arduino dan dijadikan high agar bisa menembak seperti yang ada pada game virtual tersebut.

Adapun fitur yang disajikan dalam senjata mainan ini adalah:

1. Terdapat 4 slot baterai AA yang masing-masing bertegangan 1.5V sehingga kalau digabungkan cukup untuk menyuplai tegangan 6V.
2. Mempunyai trigger seperti push button.

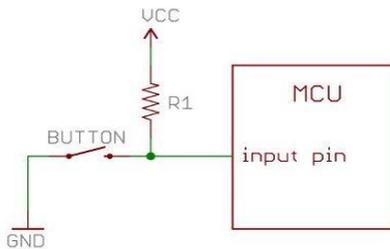


Gambar 2.5 Senjata Mainan

Sumber: gamebrott.com/mengenal-lebih-dekat-senjata-favorit-kamu-di-pubg-dan-pubg-mobile

2.10 Pull Up Resistor

Pull Up Resistor berfungsi sebagai mencegah kondisi *floating* atau kondisi mengambang tidak mengetahui itu logika “0”/”1”. Pull Up Resistor bisa digunakan dalam pin pin digital yang mana memerlukan logika HIGH atau LOW.



Gambar 2.6 Pull Up Resistor

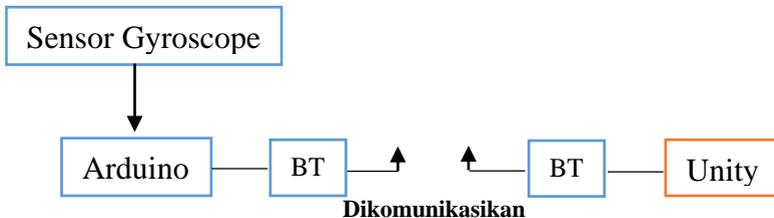
Sumber: learn.sparkfun.com/tutorials/pull-up-resistors/all

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan Sensor Gyroscope Pada Simulasi Virtual Game Shooter Menggunakan Unity Berbasis Arduino, yang meliputi blok diagram yang akan menjelaskan proses kerja alat dalam bentuk alur diagram seperti gambar, perancangan perangkat lunak (*hardware*), dan perancangan perangkat lunak (*software*) yang akan menjelaskan mengenai pembuatan program kendali senjata virtual game shooter dengan sensor *gyroscope* menggunakan Arduino, dan perancangan *virtual game shooter* menggunakan *Unity*.

3.1 Blok Diagram

Sebelum melakukan pembuatan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*), diperlukan sebuah perencanaan sistem berupa blok diagram yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok Diagram

Dari Gambar 3.1 di atas merupakan gambaran kerja dari sistem bagian pada senjatanya. Di dalam blok diagram ini senjata sebagai kontroler pergerakan arah pada game shooting meliputi sensor gyroscope di program lewat arduino, kemudian dikirim menuju game shooting Unity lewat Bluetooth serta juga membuat virtual game shooting tersebut. Cara kerja sistem diatas, yaitu sensor gyrocope mengirim data pergerakan ke arduino dan arduino mengirim data serial menggunakan Bluetooth yang ada.

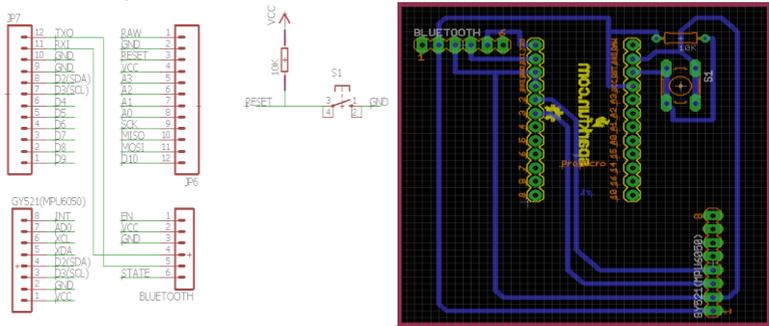
3.2 Perancangan Hardware

Pada bab perancangan hardware akan dibahas mengenai rangkaian elektrik beserta komponen-komponen pendukungnya. Sehingga Sensor

Gyroscope Pada Simulasi *Virtual Game Shooter* Menggunakan Unity Berbasis Arduino dapat berfungsi dengan baik dan benar. Pembahasan bab ini meliputi perancangan shield mikrokontroler, rangkaian wireless dengan arduino.

3.2.1 Perancangan Shield Mikrokontroler

Perancangan *shield* ini digunakan untuk mempermudah koneksi antara arduino dengan modul – modul yang digunakan dalam Proyek Akhir ini. Pada perancangan ini Arduino Pro Micro sebagai mikrokontroler yang digunakan untuk mengatur pergerakan layar *virtual game shooter* menggunakan data sensor *gyroscope*. Pada rangkaian ini ditambahkan *shield* untuk komunikasi *wireless*, sensory gyroscope, push button sebagai reset. Adapun rangkaian *shield* Mikrokontroler dengan sistem ditunjukkan oleh Gambar 3.4.



Gambar 3.2 Rangkaian *Shield* Mikrokontroler

Tabel 3.1 Konfigurasi Pin Arduino ke Modul-modul

No.	Pin Arduino	Keterangan
1.	TXD	Pin 5
2.	RXD	Pin 4
3.	VCC +5	VCC (Sensor, <i>Bluetooth</i> , & Pull Up Resistor)
4.	Gnd	Gnd (Sensor, <i>Bluetooth</i> , & Push Button)
5.	RESET	Push Button
6.	TX	RX (<i>Bluetooth</i>)
7.	RX	TX (<i>Bluetooth</i>)
8.	Pin 2	SDA (Sensor)
9.	Pin 3	SCL (Sensor)

3.2.2 Rangkaian Wireless dengan Arduino

Wireless yang digunakan untuk komunikasi dalam Proyek Akhir ini adalah Bluetooth HC-05.



Gambar 3.3 Sambungan Arduino dengan Bluetooth HC-05.

Konfigurasi pin HC-12 dengan Mikrokontroler tertera pada Tabel 3.2.

Tabel 3.2 Konfigurasi Pin HC-05 dengan Arduino.

No.	Pin HC-05	Keterangan
1.	TXD	Pin RX Arduino
2.	RXD	Pin TX Arduino
3.	Vcc +5	(+)VCC
4.	Gnd	(-) Gnd Arduino

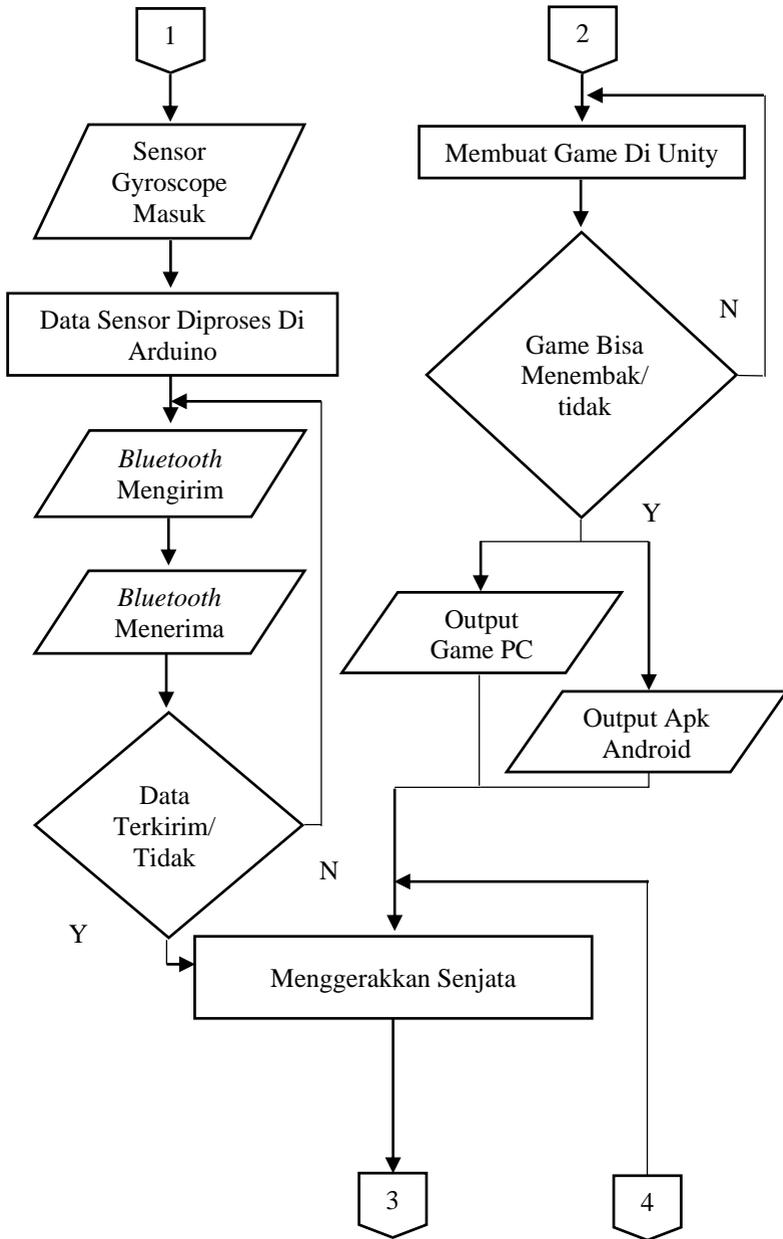
TX dan Rx menggunakan protokol yang diimplementasikan dalam sebuah perangkat bernama UART (*Universal Asynchronous Receiver / Transmitter*). Rx adalah jalur penerimaan data (perpindahan data) dari satu komputer ke komputer lain. Rx biasa disebut *received*, yang menerima data yang dikirim oleh (Tx) nya Arduino. Tx disebut *Transmitter* yang berfungsi untuk mengirim data/mengeluarkan data, atau merupakan jalan yang dilalui dalam mengirim data antar device. Data akan dikirim melalui Tx (*Transmitter*) dan di ujung lainnya data akan diterima melalui Rx (*Received*).

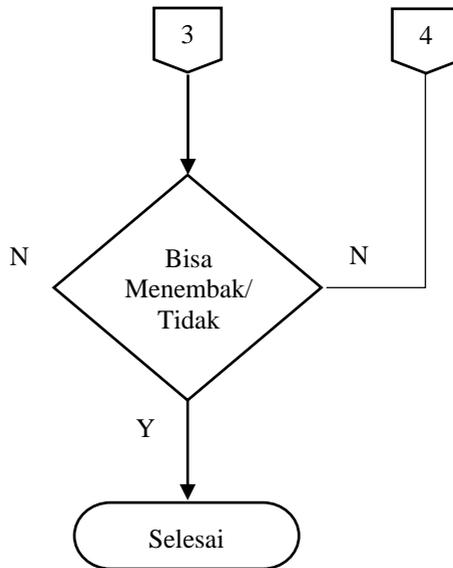
3.3 Perancangan Software

Setelah semua *hardware* telah terangkai maka dilakukan pembuatan rancangan program agar sistem dapat bekerja sebagaimana mestinya. Dalam Proyek Akhir ini *software* yang digunakan adalah Unity dan Arduino. Pada bab perancangan *software* dibahas tentang *flowchart* untuk mendukung kerja sistem secara keseluruhan.

3.3.1 Flowchart







Gambar 3.4 Rancangan *Flowchart* Keseluruhan

Untuk mengatur pergerakan senjata didalam *Software game* diperlukan pengaturan mekanis seperti *gyroscope*, Arduino mengirim data *serial* ke *Game Unity*. Perintah yang dikirim oleh Arduino berupa perintah untuk menggerakkan *mouse* menggunakan sistem HID. Sebelum simulasi tersimpan dalam *game*, diperlukan tahapan proses dalam membuat *game* tersebut. Setelah game siap digunakan, program sensor di masukkan dan dicoba disambung dengan game VR tersebut.

3.3.2 Rancangan Program Arduino

3.3.2.1 Program *Bluetooth*

Program *Bluetooth* ini adalah rancangan untuk mengirim data satu sama lain agar bisa terkirim lewat tanpa kabel. *Bluetooth* ini bisa diatur fungsinya menjadi *slave* atau *master* dengan AT+Command. Dengan pengaturan ini *bluetooth* yang satu bisa diseting jadi *master* yang mengirim data dan satunya lagi di setting menjadi *slave* untuk menerima data.

```

/* Serial Loop */

char myChar ;

void setup() {
  Serial.begin(9600);
  Serial.println("AT");
  Serial1.begin(38400);
  Serial1.println("AT");
}
void loop() {
  while (Serial1.available()) {
    myChar = Serial1.read();
    Serial.print(myChar);
  }
  while (Serial.available()) {
    myChar = Serial.read();
    Serial.print(myChar); //echo
    Serial1.print(myChar);
  }
}
}

```

Gambar 3.5 Program serial *bluetooth*

Gambar diatas adalah program untuk mengirim serial biasa yang akan di gabungkan sama program. Sebelum di gabungkan bluetooth harus di atur data (*Address*, UART, ROLE, dan lain-lain) lewat AT+Command agar bluetooth yang satu berfungsi sebagai master dan satunya lagi sebagai slave kemudian pairing satu sama lain. *Address* disini adalah nomor alamat yang sudah ada pada *bluetooth*, dengan perintah “AT+ADDR?” inilah alamatnya keluar. UART juga diperlukan untuk mengatur parameter-parameter yang digunakan oleh *bluetooth*, yaitu: *baudrate*, start-stop bit, dan parity. ROLE adalah kondisi dia menjadi *master* atau *slave*.

3.3.2.2 Program *Gyroscope*

Program *gyroscope* ini ada untuk memerintahkan/ mengendalikan layar game tersebut agar sensor bisa bergerak sesuai dengan *software* game tersebut dan juga bisa untuk menembak.

```

#include <Wire.h>
#include <I2Cdev.h>
#include <MPU6050.h>
#include <Mouse.h>

MPU6050 mpu;
int16_t ax, ay, az, gx, gy, gz;
int X, Y;
const int sensitivity = 50;

void setup() {
  pinMode(4, INPUT);
  Serial.begin(9600);
  Wire.begin();
  Mouse.begin();
  mpu.initialize();
  if (!mpu.testConnection()) { while (1); }
}

```

Gambar 3.6 Program *Gyroscope*

-----Halaman ini sengaja dikosongkan-----

BAB IV PENGUJIAN DAN ANALISA

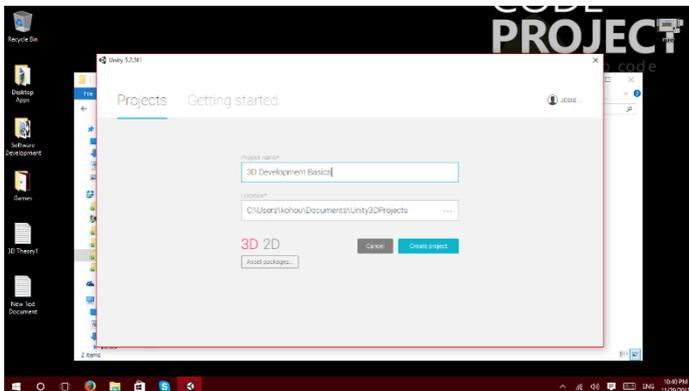
Setelah tahap perancangan Sensor Gyroscope Pada Simulasi *Virtual Game Shooter* Menggunakan Unity Berbasis Arduino selesai, berikutnya akan dilakukan pengujian dan analisa untuk mengetahui kinerja sistem yang telah dirancang. Pengujian dilakukan secara terpisah, yaitu Pembuatan *Software Game*, Pengujian *Game* dan Pengujian Alat Keseluruhan. Dari pengujian ini akan dilihat apakah sistem yang dibuat dapat berjalan sesuai dengan yang diinginkan.

4.1 Pembuatan *Software Game*

Di dalam pembuatan *software game* ini meliputi beberapa tahapan, yaitu Membuat *File New Project*, *Window Utama Unity* dan Menyimpan *Game* dengan Konfigurasi *Platform* sebagai berikut:

4.1.1 *File New Project*

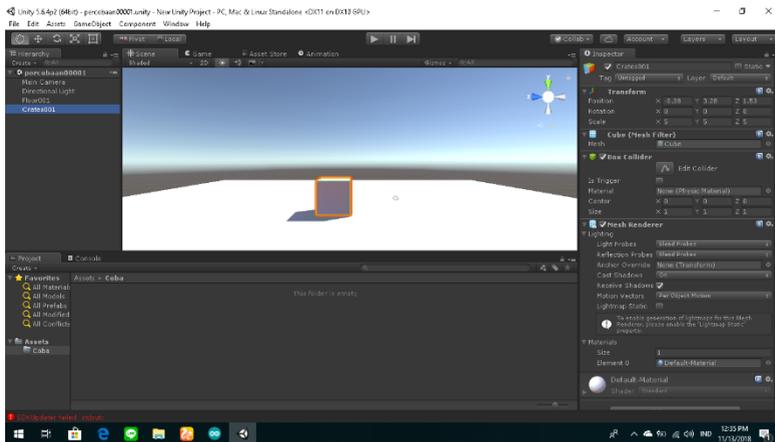
Ketika pertama kali menggunakan Unity diperlukan untuk membuat lembar kerja baru. Dan yang pertama kali yang dilakukan yaitu log in terlebih dahulu agar Unity bisa mendeteksi bahwa lisensi yang digunakan sudah benar atau tidak dan menggunakan lisensi yang bisa digunakan sewaktu pertama menginstall (*Personal, Plus, dan Pro*). Dan saya menggunakan yang personal, personal ini hanya digunakan untuk orang-orang ingin menggunakan *unity* pertama kali (*Beginners*).



Gambar 4.1 Tampilan Pertama *Unity*

Pada tampilan itu kita diminta memasukkan lokasi dimana kita menyimpan *project* tersebut, dan *Package* apa saja yang dibutuhkan pada *project* anda. Jika ragu untuk memilih *Package* yang akan di *import*, maka lebih baik tidak meng*import package* sama sekali dan ketika butuh *package* bisa di*download* lewat toko/*store* pada *Unity* nanti.

Setelah selesai menentukan lokasi dan memilih *package* yang ada kemudian *create project*. Tunggu beberapa saat maka *project* baru bisa dibuat.



Gambar 4.2 Project Unity

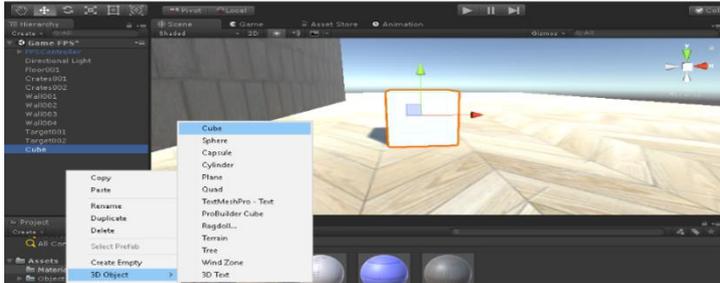
4.1.2 Window Utama Unity

Tampilan utama *Unity* ada macam-macam, yaitu: *Scene*, *Game*, *Hierarchy*, *Project* dan *Inspector*.

4.1.2.1 Scene

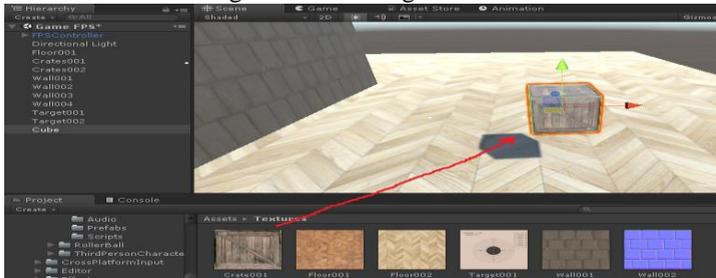
Scene itu apa? Ialah tampilan yang berisi objek permainan. Mereka dapat digunakan untuk mengatur tata letak objek, membuat objek yang berperan sebagai barang atau pun sebagai individual yang dapat bergerak, dan apa pun. Setiap membuat *file Scene* yang unik maka hasil *game* nantinya cukup unik. Di setiap adegan, ada kondisi bisa menempatkan lingkungan, hambatan, dan dekorasi, pada dasarnya merancang dan membangun *game* tersebut menjadi potongan-potongan. Dan *scene* yang dibuat dalam *game* ini adalah Tembok, sasaran target, *box*, lantai, *FPS Controller*, serta senjatanya. Pembuatannya sebagai berikut:

- a. Pertama *cube* 3D dimunculkan dan ditaruh didalam tampilan *Scene*. Cube ini berfungsi sebagai bentuk fisik dari suatu object yang nanti nya akan terisi misalnya: *Box*, *Wall*, *Target*, dll.



Gambar 4.3 Memasukkan *Cube* 3D

- b. *Cube* tersebut diatur akan sebagai *object* bagian apa, misal yang pertama akan dijadikan sebagai sebuah *box*. Dan proses pertama yaitu *texture* yang *box* didownload di internet terus dimasukkan/didrag ke *cube* kosong. Gambar 4.4 dibawah:



Gambar 4.4 Mengedrag *textures object*

- c. Dimasukkan semua benda *object* yang dibutuhkan, tembok, target, lantai:



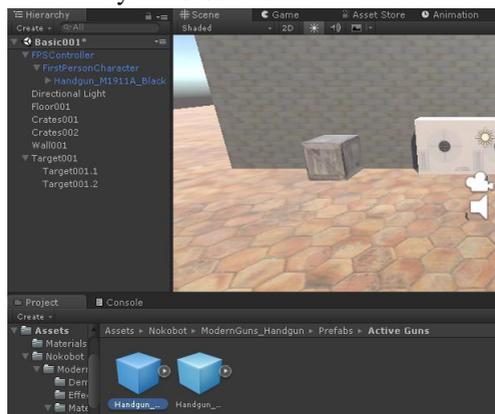
Gambar 4.5 *Box*, *Target*, dan *Wall*.

- d. juga perlu dimasukkan FPS (*First Person Shooter*) didalamnya, karena FPS ini berpengaruh besar sebagai tampilan untuk menggerakkan *character* yang bisa loncat, berjalan dan sebagainya.



Gambar 4.6 FPS Character

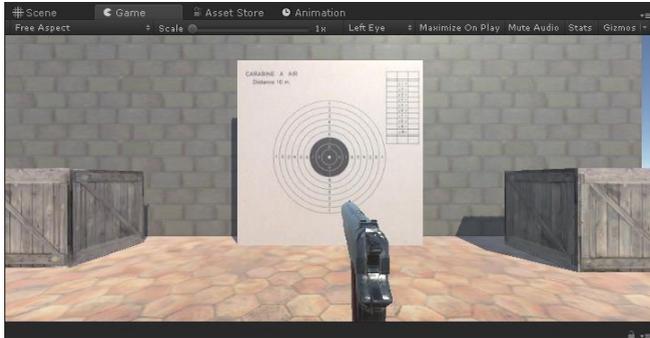
- e. Perlu juga di tambahkan pistol untuk senjatanya, pistolnya di *import* dari file *asset* yang terdapat di internet, tetapi downloadnya harus melewati *asset store* yang berada di Unity dimana harus *log in* terlebih dahulu. Setelah di *import* file yang ada di dalam nya dibuka dan cari *Prefabs* dari pistol tersebut. Dengan drag & drop di FPS Controller senjata tersebut dapat masuk ke view nya FPS Controller.



Gambar 4.7 Prefabs Handgun

4.1.2.2 Game View

Game view adalah tampilan untuk mensimulasikan *game* ketika dijalankan. Dimana kita bisa mereview *game* tersebut bekerja atau tidak.

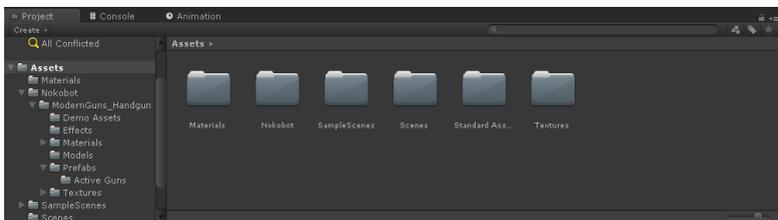


Gambar 4.8 *Game View*

4.1.2.3 *Hierarchy Window*

Hierarchy adalah berisikan seluruh *game object* yang ada di dalam di dalam *game scene*. Jika di dalam *project window* adalah aset yang ada di dalam *harddisk* kita, *hierarchy* berisikan list-list *game object* yang kita gunakan di dalam *scene*. Apa yang ada pada dalam *scene* akan ditampilkan secara list di *window* ini. Kita bisa memasukan aset dari *project window* ke *hierarchy* dengan cara *drag and drop* atau kita juga bisa *create* aset menggunakan *drop down create* yang ada di *toolbar* *hierarchy*. Seperti contoh-contoh pada Gambar sebelumnya: 4.3, 4.5, 4.6, dan 4.7.

4.1.2.4 *Project Window*



Gambar 4.9 *Project Window*

Project window ini digunakan untuk mengorganisir aset yang kita gunakan. Struktur file yang kita buat di dalam *project window* akan disimpan dalam struktur yang sama juga di dalam *harddisk* kita. Di dalam *project window*, kita bisa memasukan aset yang akan kita gunakan.

Seperti memasukkan *folder*, *file*, *textures*, dan lain-lain yang mana semua penting untuk dimasukkan ke game tersebut.

Dan sebelah kanan dari list kiri adalah tempat hasil yang di *import* dari internet termasuk pistol, foto gambar-gambar *object* dimasukkan ke folder yang bernama *textures*, *material*, *skrips*, dan lain-lain. Semua itu bisa disebutkan sebagai *package* kumpulan bahan-bahan dari *game* tersebut.

4.2 Pengujian Game

Tujuan pengujian *game* adalah untuk mengetahui apakah VR *game shooter* ini dapat bekerja dengan baik atau tidak. Serta tahap-tahap terpengaruhnya kondisi ketika diberi *rigidbody*. Pengujian ini dilakukan dengan memainkan *game* tersebut pada *Unity*. Dengan tekan tombol *Play* diatas apapun yang ada pada *scene* akan disimulasikan.

4.2.1 Pengujian Rigidbody

Rigidbody mengaktifkan *GameObjects* tersebut untuk bertindak di bawah kendali fisika. *Rigidbody* dapat menerima kekuatan dan torsi untuk membuat obyek tersebut bergerak secara realistis. Berikut adalah hasil penelitian ketika objek yang dikasih komponen *rigidbody* dengan massa yang yang berbeda serta ketinggian yang berbeda. Dengan ketentuan hanya *Use Gravity* nya saja yang di centang sisanya *default* aja.

Tabel 4.1 Objek yang terdorong

No.	Kondisi <i>Object</i> (<i>Box/Crates</i>)	Massa
1.	Bergerak	1
2.	Bergerak	5
3.	Bergerak	15
4.	Tidak Bergerak	20
5.	Tidak Bergerak	25
6.	Tidak Bergerak	30
7.	Tidak Bergerak	35
8.	Tidak bergerak	40
9.	Tidak bergerak	45
10.	Tidak bergerak	50
11.	Tidak bergerak	55
12.	Tidak bergerak	60
13.	Tidak bergerak	65
14.	Tidak bergerak	70

15.	Tidak bergerak	75
16.	Tidak bergerak	80
17.	Tidak bergerak	85
18.	Tidak bergerak	90
19.	Tidak bergerak	95
20.	Tidak bergerak	100

Jadi pengujian pendorongan *object* ini di lakukan pada objek *Box*. Untuk menguji seberapa pengaruhnya berat massa yang diberi oleh *rigidbody*.

Tabel 4.2 Pengaruhnya Ketinggian

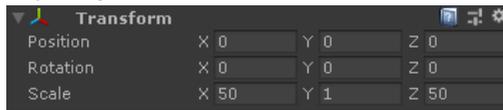
No.	Posisi Ketinggian (Sumbu Y)	Massa	Pengaruh Kondisi Ketika di Play serta Hasil Posisi yang telah Berubah (X,Y,Z)	Posisi Lantai (Sumbu Y)
1.	1	1	Kembali ke atas permukaan lantai (-4,498 . 1,5 . 4,254)	0
2.	1,5	1	Diam ditempat (-4,5 . 1,5 . 4,25)	0
3.	2	1	Jatuh ke lantai secara normal (-4,5 . 1,5 . 4,25)	0
4.	3	1	Jatuh ke lantai secara normal (-4,5 . 1,5 . 4,25)	0
5.	5	1	Jatuh ke lantai secara normal (-4,5 . 1,5 . 4,25)	0
6.	10	1	Jatuh ke lantai secara normal (-4,5001 . 1,5 . 4,25009)	0

7.	15	1	Jatuh ke lantai secara normal (-4,5001 . 1,5 . 4,25009)	0
8.	20	1	Jatuh ke lantai (-4,50012 . 1,5 . 4,2507)	0
9.	25	1	Jatuh ke lantai secara normal (-4,5 . 1,5 . 4,25)	0
10.	50	1	Jatuh ke lantai secara normal (-4,499 . 1,5 . 4,25)	0
11.	75	1	Jatuh ke lantai secara normal (-4,5036 . 1,5 . 4,265)	0
12.	100	1	Jatuh ke lantai secara normal (-4,500 . 1,5 . 4,250)	0
13.	150	1	Jatuh ke lantai secara normal (-4,499 . 1,5 . 4,251)	0
14.	200	1	Jatuh ke lantai secara normal (-4,486 . 1,5 . 4,268)	0
15.	300	1	Jatuh ke lantai secara normal (-4,502 . 1,5 . 4,2689)	0
16.	400	1	Jatuh ke lantai secara normal (-5,477 . 1,5 . 4,117)	0
17.	500	1	Jatuh ke lantai secara normal (-4,5528 . 1,5 . 4,243)	0

18.	700	1	Jatuh secara tidak normal terus tembus (-4,5 . -9999 . 4,25)	0
19.	800	1	Jatuh ke lantai secara normal (-5,161 . 1,5 . 4,0589)	0
20.	1000	1	Jatuh ke lantai secara normal (-4,5003 . 1,5 . 4,2504)	0

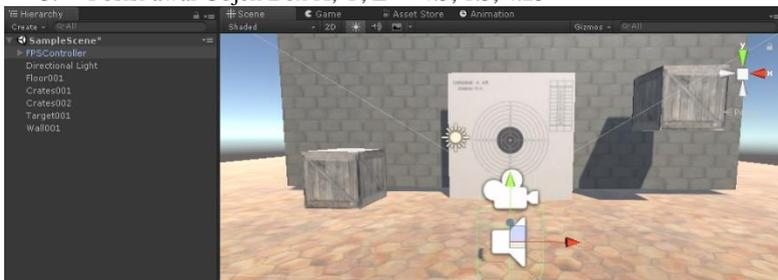
Ket:

1. Skala ukuran lantainya X, Y, Z = 50, 1, 50.
2. Rotasinya Object di set X, Y, Z = 0, 0, 0.



Gambar 4.10 Posisi, Rotasi, & Skala (Objek Lantai)

3. Posisi awal Objek Box X, Y, Z = -4.5, 1.5, 4.25



Gambar 4.11 Uji Coba *Rigidbody* pada Objek

Jadi pada uji coba *Rigidbody* ini untuk mencari bagaimana pengaruhnya gravitasi didalam game ketika di dorong dan ketika ketinggian beratus-ratus sampai ribuan.

4.2.2 Uji Coba Menembak di Unity

Uji Coba Menembak ini adalah uji coba untuk melihat bahwa peluru yang dikeluarkan hingga meluncur sampai suatu benda object menghilang

atau kah membekas, atau juga memantul. Dan juga menguji coba ketika benda tersebut ada *rigidbody* peluru tersebut terpengaruh apa tidak.

Tabel 4.3 Uji Coba Menembak di Unity

No.	Kondisi Berapa Peluru yang di keluarkan	Hasil Kondisi Objek Target (<i>Rigidbody</i>)	Hasil Kondisi Objek Target (Tanpa <i>Rigidbody</i>)
1.	1	Peluru Memantul	Peluru terlewat
2.	2	Peluru memantul tetapi objek mulai bergerak	Peluru terlewat
3.	3	Terpental dan langsung jatuh	Peluru terlewat
4.	4	Terpental, jatuh, dan peluru sudah tidak mempengaruhi ketika jatuh	Peluru terlewat
5.	5	Terpental, jatuh, dan peluru sudah tidak mempengaruhi ketika jatuh	Peluru terlewat

Ket: Massa objek Target tersebut hanya 5

4.2.3 Uji Coba Penyimpanan *Game*

Uji coba penyimpanan *Game* adalah uji coba yang dilakukan untuk proses menyimpan *game* tersebut dari awal kegagalan sampai keberhasilannya. Kenapa dilakukan uji coba penyimpanan ini?

Dikarenakan sewaktu menyimpan suatu *game*. Diperlukan suatu penyesuaian yang ditentukan oleh *Unity* apakah ada *error* pada animasi ataupun *error* pada pengaturan dari *Unity* tersebut.

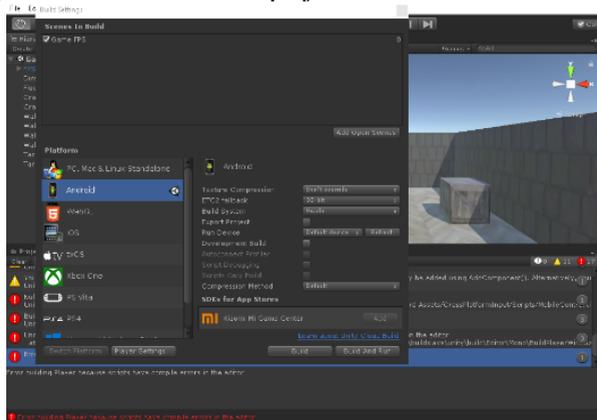
Error yang pertama adalah *error* ketika setelah semua hasil *project game* sudah selesai maka perlu disimpan lewat langkah langkah: *Build Setting*, *Switch Platform*, dan *Build*. *Build Setting* ada pada *toolbar* pojok sebelah kiri atas terus kemudian menuju *switch platform*. *Switch Platform* ini berfungsi merubah tipe kondisi *game* tersebut akan *support* jadi apa. Karena VR diperlukan lewat *android* ganti platform ke *android*

system. Berikut perubahan dalam proyek yang dibuat sebelum di ganti platform dan sesudah diganti platform ke android system:

Tabel 4.4 Perbedaan Platform Android dengan Platform PC

	Platform PC	Platform Android
Mencoba pergerakan layar sebelum di Build	Layar FPS bisa digerak-gerakkan	Layar FPS tidak bisa digerak-gerakkan
Mencoba pergerakan layar setelah di Build	Layar FPS bisa digerak-gerakkan	Layar FPS tidak bisa digerak-gerakkan

Setelah diganti platform bisa dicoba untuk menyimpan game ini atau juga bisa disebut membuild proyek tersebut.



Gambar 4.12 Ketika terjadinya Error

4.3 Pengujian Sensor Gyroscope

Tujuan pengujian sensor *gyroscope* adalah untuk mengetahui apakah sensor tersebut berfungsi dengan baik. Tiap bergerak apakah ada suatu kejanggalan atau bagaimana. Sesudah di konfigurasi dengan program yang memerlukan *compatible* dengan *mouse* atau HID, *mouse* bisa bekerja dengan baik atau tidak.

4.3.1 Pengujian Sensor *Gyroscope* Lewat Serial Monitor

Pengujian sensor *gyroscope* lewat serial monitor ini berfungsi melihat bahwa sumbu x dan y yang diambil tersebut.

1. Pengujian Sensor *Gyroscope* pada Saat Diam

Tabel 4.5 Pengujian sensor *gyroscope* pada saat diam

No.	Kondisi	Sumbu X	Sumbu Y	Hasil X	Hasil Y
1.	Diam	-496	-126	0	1
2.	Diam	-446	-131	0	1
3.	Diam	-446	-90	0	0
4.	Diam	-426	-52	0	0
5.	Diam	-460	-84	0	0
7.	Diam	-383	-95	0	0
8.	Diam	-382	-106	0	1
9.	Diam	-346	-87	0	0
10.	Diam	-360	-90	0	0
11.	Diam	-357	-102	0	1
12.	Diam	-336	-95	0	0
13.	Diam	-351	-43	0	0
14.	Diam	-372	-79	0	0
15.	Diam	-395	-103	0	1

Ket: Hasil X, Y disebelah kanan ini adalah hasil dari pembagian dari sumbu x, y sensor *gyroscope* dengan nilai yang sekitar 150 agar nilai yang didapat bisa mengecilkan nilai gx & gy.

4.3.2 Pengujian Sensor saat Diputar Berapa Derajat di Komputer

Pengujian ini adalah untuk mengetahui berapa maksimal kursor komputer atau berapa *pixel* yang didapat di komputer ketika belok beberapa derajat.

Tabel 4.6 Pengujian Sensor Saat digerakkan Searah Sumbu X

No.	Sudut Derajat	Sudut Awal	Posisi Awal Kursor	Hasil Pergerakan <i>Pixel Game</i>
1.	90°	90°	Ditengah	Diam
2.	40°	90°	Ditengah	Bergeser Kekanam hampir sampai ujung kanan layar komputer

3.	150°	90°	Ditengah	Bergeser Kekiri hampir sampai ujung kiri layar komputer
4.	0°	90°	Ditengah	Bergeser sampai ujung kanan layar komputer
5.	180°	90°	Ditengah	Bergeser sampai ujung kiri layar komputer

Tabel 4.7 Pengujian Sensor Saat digerakkan searah Sumbu Y

No.	Sudut Derajat	Sudut Awal	Posisi Awal Kursor	Hasil Pergerakan <i>Pixel Game</i>
1.	90°	90°	Ditengah	Diam
2.	40°	90°	Ditengah	Bergeser Kekanam hampir sampai ujung atas layar komputer
3.	150°	90°	Ditengah	Bergeser Kekiri hampir sampai ujung bawah layar komputer
4.	0°	90°	Ditengah	Bergeser sampai ujung atas layar komputer
5.	180°	90°	Ditengah	Bergeser sampai ujung bawah layar komputer

4.4 Pengujian *Bluetooth*

Tujuan pengujian pada *Bluetooth* ini adalah untuk mengetahui bahwa dia bisa mengirim data serial dengan baik apa tidak. Dari mulai pengujian *AT Command*, pengujian *Pairing* satu sama lain.

4.4.1 Pengujian AT Command

Pengujian AT Command ini adalah pengujian *bluetooth* dalam menyesuaikan, kalibrasi, atau juga pengaturan bahwa *bluetooth* HC-05 berfungsi sebagai *Master/Slave*, mendapatkan alamat/*address*, setting nama modul *bluetooth*nya atau hanya melihat nama modul tersebut, dan lain-lain. Dan itu semua bisa di perintah lewat serial monitor hanya dengan mengetik perintah yang ada pada datasheet maka pasti bisa. Akan tetapi dengan catatan bahwa ketika perintah tersebut dijawab, dan jawabannya pasti OK atau kalau gagal ERROR [0]. Perlu juga di cek dari segi *baudrate* yang *bluetooth* digunakan dan disesuaikan di *baudrate* serial monitor arduino sebelah kanan bawah seperti di gambar 4.2.



Gambar 4.14 Serial Monitor Arduino

Terus kemudian dicek lagi yang sebelumnya baudratanya serial monitor tersebut, disitu ada pilihan *no line ending*, *both NL & CR*, dan dua lainnya. Langsung pilih aja yang *Both NL & CR* dikarenakan dia memiliki dua fungsi NL (Newline) & CR (Carriage Return). Carriage Return berfungsi ketika mengirim kode/kata” maka dia akan memunculkan kembali yang telah di kirim tadi. Newline berfungsi sebagai terusan dari jawaban nya kode perintah yang pertama dan ditaruh dibawahnya. Dan contoh hasil nya seperti pada gambar 4.3. Setelah itu dilakukan perintah-perintah AT+Command nya.



Gambar 4.15 Serial Monitor Arduino 2

Ada kondisi dimana *bluetooth* tidak mau mengirim ada beberapa hal penyebabnya yaitu dari *bluetooth* HC-05 yang ada banyak versi. Biasanya orang akan mencari *bluetooth* HC-05 yang paling mudah settingannya tanpa mencoba-coba hal-hal lainnya. Tapi juga terkadang ada yang salah juga memilih salah satu dari 2 *Bluetooth* tersebut hanyalah sebuah *chipnya* saja. Kondisi kedua yang terjadi hingga *bluetooth* tidak mau mengirim yaitu wiringnya atau pengkabelannya ada yang salah dan bisa saja terjadi *short*.

4.4.2 Pengujian *Pairing Bluetooth*

Pengujian *Pairing Bluetooth* ini adalah mengecek kondisi dimana antara dua *bluetooth* bisa tersambung apa tidak dengan sistem AT *Command*:

1. Pertama di cek “AT” seperti pada gambar 4.3.
2. Kedua cek “AT+Role?” dan atur AT+Role yang diinginkan. AT Role berisi tentang parameter *Slave* dan *Master*. Salah satu *bluetooth* bisa di jadikan *Master* dan pasangannya *Slave* begitu sebaliknya pun sama. Parameter *Slave* = 0 dan parameter *Master* = 1.
3. Ketiga cek alamat *address* dengan memberi perintah “ADDR?” dapat mengetahui berapa alamatnya.
4. *Bluetooth* yang kedua dikonfigurasi seperti nomor 1 dan 2. Kemudian beri perintah untuk mengkaitkan atau mempairing *bluetooth* yang pertama tadi dengan perintah “AT+BIND=<alamat yang bluetooth satunya>”.
5. Setelah semua OK, itu adalah bukti kalau *bluetooth* sudah di pairing dan siap untuk dikirim data. Dengan catatan lampu di modul

berjalan menyala mati dengan normal dan juga sudah diprogram yang sesuai dengan yang diinginkan.

4.5 Pengujian Alat Keseluruhan

Pengujian alat secara keseluruhan ini adalah alat yang sudah tersambung sensor dan juga tersambung langsung pada *Game*.

4.5.1 Pengujian Pergerakan Layar *Game*

Tujuan pengujian pergerakan layar *game* ini adalah untuk mengetahui seberapa efektifnya sensor *gyroscope* ketika digerakkan dengan *Game* yang dibuat. Pengujian ini bisa dilihat dari *Game View* atau layar *game* yang sudah terinstall.

Tabel 4.8 Uji Coba Pergerakan Layar *Game* Sumbu X

No.	Sudut Sensor	Sudut Awal	Posisi Awal Senjata Di <i>Game</i>	Hasil Sudut & Pergerakan View Di VR <i>Game Shooter</i>
1.	90°	90°	Tengah	Diam
2.	40°	90°	Tengah	Posisi View berubah 0,5
3.	150°	90°	Tengah	Posisi View berubah -0,5
4.	0°	90°	Tengah	Posisi View berubah 1
5.	180°	90°	Tengah	Posisi View berubah -1

Tabel 4.9 Uji Coba Pergerakan Layar *Game* Sumbu Y

No.	Sudut Sensor	Sudut Awal	Posisi Awal Senjata Di <i>Game</i>	Hasil Sudut & Pergerakan View Di VR <i>Game Shooter</i>
1.	90°	90°	Tengah	Diam
2.	40°	90°	Tengah	Posisi View berubah -5,3
3.	150°	90°	Tengah	Posisi View berubah 7,1

4.	0°	90°	Tengah	Posisi View berubah -11,6
5.	180°	90°	Tengah	Posisi View berubah 14,1

4.5.2 Pengujian Menembak Game dengan Alat

Pengujian menembak game dengan alat ini bertujuan trigger yang di pakai untuk menembak di layar game yang telah ter install bisa bekerja dengan baik apa tidak.

Tabel 4.10 Pengujian Menembak Game dengan Alat

No.	Kondisi Berapa Peluru yang di keluarkan	Hasil Kondisi
1.	1	Peluru ditembakkan tetapi resporn cukup lambat
2.	2	Peluru tidak keluar
3.	3	Peluru ditembakkan tetapi resporn sangat lambat
4.	4	Peluru ditembakkan tetapi resporn cukup lambat
5.	5	Peluru tidak keluar

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Bab penutup ini berisi kesimpulan yang diperoleh selama proses pembuatan alat Sensor Gyroscope Pada Simulasi *Virtual Game Shooter* Menggunakan Unity Berbasis Arduino kesimpulan dari hasil pengujian dan analisa data, serta saran untuk pengembangan alat ini kedepannya.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari Proyek Akhir dengan judul Sensor Gyroscope Pada Simulasi *Virtual Game Shooter* Menggunakan Unity Berbasis Arduino adalah :

1. Dalam hal membuat *game* 3D komponen *rigidbody* di dalam objek ketika dimainkan pasti ada kesalahan (*bug*) di *game* tersebut.
2. Dalam menyimpan *game* diperlukan nama perusahaan, nama project *game* tersebut agar bisa menyimpan.
3. Sensor *gyroscope* ketika dicoba dengan layar komputer tidak bisa diam semestinya pasti ditiap berapa *pixel* layar sumbu y berubah.
4. Respon sensor *gyroscope* ketika bergerak sumbu x dan y cukup lambat.
5. Respon senjata untuk menembak butuh jeda peluru dikeluarkan

5.2 Saran

Saran untuk Sensor Gyroscope Pada Simulasi *Virtual Game Shooter* Menggunakan Unity Berbasis Arduino adalah:

1. Sebaiknya selanjutnya mencoba virtual langsung tanpa menggunakan tampilan 3D terlebih dahulu.
2. Sebaiknya selanjutnya tembakan *game* tersebut ada suara nya.
3. Sebaiknya selanjutnya mencoba berbagai macam animasi didalamnya dan mempelajari komponen-komponen lainnya.
4. Sebaiknya selanjutnya alat tersebut bisa digunakan untuk berjalan maju mundur dengan sensor juga.
5. Sebaiknya selanjutnya sensor yang di gunakan cukup sensitif untuk menggerakkan layar komputer maupun layar *game shooter* dengan pergerakan yang sedikit.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Carbin, Eddie. 2017. “*Quick Draw Vr Shooting Gallery – Arduino Toy Gun Vr Controller*” <http://www.eddiecarbin.com/wp/quick-draw-vr/> (diakses tanggal 27 September 2018).
- [2] Gabry295, 2017, Head Mouse With MPU6050 and Arduino Micro” <https://www.instructables.com/id/Head-Mouse-With-MPU6050-and-Arduino-Micro/> (diakses tanggal 28 Maret 2017).
- [3] Jimblom. 1996. “*Pro Micro & Fio V3 Hookup Guide*”. <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-pro-micro> (diakses pada tanggal 2 Desember 2018).
- [4] Kadir, Abdul. 2018 “ From Zero to a PRO Arduino Panduan Mempelajari Pembuatan Aneka Proyek Berbasis Mikrokontroler (Edisi Revisi)” ANDI : Yogyakarta.
- [5] matthewh8,....., “*Arduino VR Skateboard Tutorial for Beginners*” <https://www.instructables.com/id/Arduino-VR-Skateboard-Tutorial-for-Beginners/> (diakses tanggal 27 September 2018)
- [6] Roedavan, Rickman, 2018, “Unity Tutorial Game Engine (Revisi Kedua)”, Informatika : Bandung.
- [7] Zucconi, Alan. 2015. “How to integrate Arduino with Unity”. <https://www.alanzucconi.com/2015/10/07/how-to-integrate-arduino-with-unity/#step1>. (diakses pada tanggal 4 Desember 2018).

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1. Listing Program pada Arduino

```
#include <Wire.h>
#include <I2Cdev.h>
#include <MPU6050.h>
#include <Mouse.h>

MPU6050 mpu;
int16_t ax, ay, az, gx, gy, gz;
int X, Y;
const int sensitivity = 50;

void setup() {
  pinMode(4,INPUT);
  Serial.begin(9600);
  Wire.begin();
  Mouse.begin();
  mpu.initialize();
  if (!mpu.testConnection()) { while (1); }
}

void loop() {
  if(digitalRead(4) == HIGH){
    Mouse.press();
  }
  if(digitalRead(4) == LOW){
    Mouse.release();
  }
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  X = (gx+300)/150;
  Y = (gy+150)/150;

  Serial.print("gx = ");
  Serial.print(gx);
  Serial.print(" | gy = ");
  Serial.print(gy);
  Serial.print(" | gz = ");
```

```
Serial.print(gz);  
  
Serial.print(" | X = ");  
Serial.print(X);  
Serial.print(" | Y = ");  
Serial.println(Y);
```

```
Mouse.move(X, Y);
```

```
delay(200);  
}
```

LAMPIRAN B

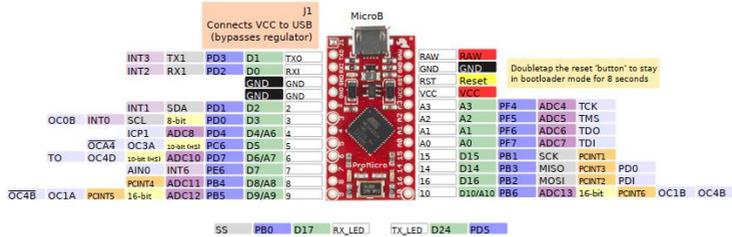
B.1 DATASHEET ARDUINO PRO MICRO

Name	ADC
Power	PWM
GND	Serial
Control	Ext Interrupt
Arduino	PC Interrupt
Port	Misc

The Arduino IDE renders all PWM pins as 8-bit

Pro Micro (Dev-12640)

16MHz/5V



Power

RAW: 6V-16V
VCC: 5V at 500mA

USB

HID enabled
VID: 0x1B4F
PID: 0x9205 (bootloader); 0x9206 (sketch)

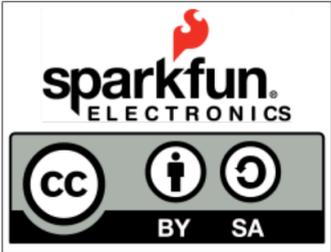
ATmega32U4
Built in USB 2.0
Absolute maximum VCC: 6V
Maximum current for chip: 200mA
Maximum current per pin: 40mA
Recommended current per pin: 20mA
8-bit Atmel AVR
Flash Program Memory: 32kB
EEPROM: 1kB
Internal SRAM 2.5kB
ADC: 10-bit
PWM: 8bit
High Speed PWM with programmable resolution from 2-11 bits

LEDs

Power: Red
RX: Yellow
TX: Green

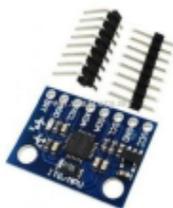
Serial

Use Serial for the USB connection
Use Serial1 for the hardware serial connection



B.2. DATASHEET GY-521/MPU6050

MPU 6050 GY-521 3 Axis Gyro Accelerometer Sensor Module Arduino



The MPU-6050 sensor module contains an accelerometer and a gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino.

Application:

- Motion-enabled game and application framework
- Location based services, points of interest
- Handset and portable gaming
- Motion-based game controllers
- Wearable sensors for health, fitness and sports
- Toys

Features:

- Use the chip: MPU-6050.
- Power supply: 3-5v (internal low dropout regulator).
- Communication modes: standard IIC communications protocol.
- Chip built-in 16bit AD converter, 16-bit data output.
- Immersion Gold PCB machine welding process to ensure quality.
- Tri-Axis angular rate sensor (gyro) with a sensitivity up to 131 LSBs/dps and a full-scale range of ± 250 , ± 500 , ± 1000 , and ± 2000 dps
- Tri-Axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$

B.3. DATASHEET HC-05

HC-05

-Bluetooth to Serial Port Module

Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-

3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

```
AT+VERSION?\r\n
```

```
+VERSION:2.0-20100601
```

4. Restore default

Command	Respond	Parameter
AT+ORGL	OK	-

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01 ,Baud 38400bits/s.

5. Get module address

Command	Respond	Parameter
AT+ADDR?	+ADDR:<Param> OK	Param: address of Bluetooth module

Bluetooth address: NAP: UAP : LAP

Example:

```
AT+ADDR?\r\n
```

```
+ADDR:1234:56:abcdef
```

```
OK
```

6. Set/Check module name:

Command	Respond	Parameter
AT+NAME=<Param>	OK	Param: Bluetooth module name (Default :HC-05)
AT+NAME?	+NAME:<Param> OK (/FAIL)	

Example:

```
AT+NAME=HC-05\r\n    set the module name to "HC-05"
```

```
OK
```

```
AT+NAME=ITeadStudio\r\n
```

```
OK
```

```
AT+NAME?\r\n
```

```
+NAME: ITeadStudio
```

```
OK
```

7. Get the Bluetooth device name:

Command	Respond	Parameter
AT+RNAME?<Param1>	1. +NAME:<Param2> OK 2. FAIL	Param1,Param 2 : the address of Bluetooth device

Example: (Device address 00:02:72:0d:22:24, name: ITead)

AT+RNAME? 0002, 72, 0d2224\r\n

+RNAME:ITead

OK

8. Set/Check module mode:

Command	Respond	Parameter
AT+ROLE=<Param>	OK	Param:
AT+ ROLE?	+ROLE:<Param>	0- Slave

HC-05 Bluetooth module

iteadstudio.com

06.18.2010



Tech Support: info@iteadstudio.com

	OK	1-Master 2-Slave-Loop
--	----	--------------------------

14. Set/Check connect mode:

Command	Respond	Parameter
AT+CMODE=<Param>	OK	Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop
AT+CMODE?	+CMODE:<Param> OK	

15. Set/Check fixed address:

Command	Respond	Parameter
AT+BIND=<Param>	OK	Param: Fixed address (Default 00:00:00:00:00:00)
AT+BIND?	+BIND:<Param> OK	

Example:

AT+BIND=1234, 56, abcdef\r\n

OK

AT+BIND?\r\n

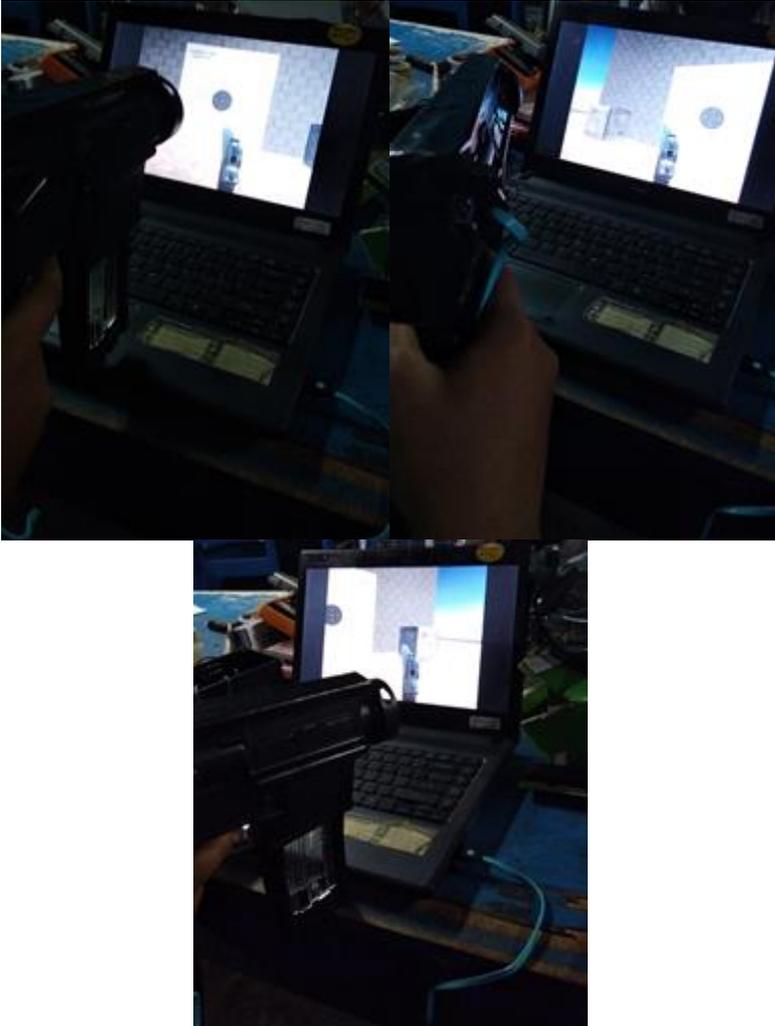
+BIND:1234:56:abcdef

OK

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C PENGUJIAN ALAT

C.1. PENGUJIAN PERGERAKAN ARAH SENJATA DI GAME



-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Syaiful Huda
TTL : Surabaya, 02 Februari
1998
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Perum Western Village
C4-12, Sememi, Benowo,
Surabaya
Telp/HP : 085706951410
E-mail : syaifulh349@gmail.com

RIWAYAT PENDIDIKAN

1. 2003 – 2009 : MI Putat Lor Menganti Gresik
2. 2009 – 2012 : SMP Negeri 1 Menganti Gresik
3. 2012 – 2015 : SMA Negeri 1 Menganti Gresik
4. 2015 – 2019 : D3 Teknik Elektro Otomasi, Program Studi Komputer Kontrol – Fakultas Vokasi Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT PLN (Persero) APB Jawa Timur

PENGALAMAN ORGANISASI

1. Staff UKM ITS Foreign Language Society 2015-2017
2. Volunteer Lego NXT IARC 2015-2016
3. Staff KWU ITS Foreign Language Society 2016-2017
4. Staff Konsum IARC 2016-2017