BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada bagian ini akan dijelaskan konsep sistem secara utuh. Tujuan dari bagian analisis sistem ini adalah untuk mengidentifikasi masalah, hambatan serta kebutuhan.

3.1.1 Analisis Masalah

Pada permainan *Augmented Wall* yang ada saat ini masih mengadopsi sistem permainan 2D, Dimana koordinat pergerakan player yang dideteksi hanya berada pada sumbu X dan Y saja. *Augmented Wall* yang ada masih memanfaatkan pengolahan citra dimana ketika objek tertutupi oleh tangan maka objek akan hilang. Pada permainan *Augmented Wall* pergerakan pemain sangat terbatas dikarenakan permainan ini masih mengadopsi sistem permainan 2D.

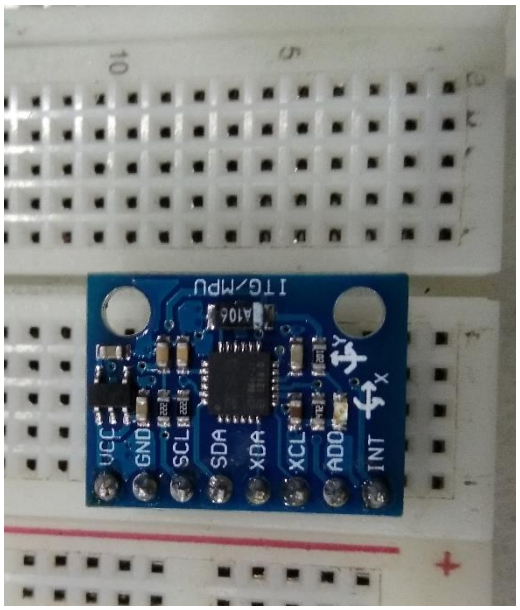
Menjawab permasalahan yang telah diuraikan maka akan ditambahkan sebuah fitur yang dimana akan membuat permainan *Augmented Wall* ini menjadi sebuah permainan 3D (tiga dimensi) yang akan menambah interaksi pemain dengan permainan, fitur ini juga akan memberikan informasi kekuatan ataupun pergerakan pemain dalam bentuk skor sehingga permainan ini akan memberikan nilai yang dinamis. Maka dari itu perlu dibuat sebuah kontroler berbasis sensor gyroscope dan accelerometer yang akan memberikan pengalaman bermain *game* 3D, sensor ini nantinya akan mendeteksi pergerakan pemain berdasarkan rotasi ataupun orientasi yang kemudian akan dilakukan perhitungan sehingga menghasilkan skor. Skor ini akan mewakili kekuatan ataupun kecepatan pukulan pemain saat sedang bermain.

3.1.2 Analisis Alat

Pada bagian ini akan dijelaskan mengenai alat-alat yang akan digunakan pasa pembangunan kontroler ini.

3.1.2.1 *Perangkat Masukan (Input)*

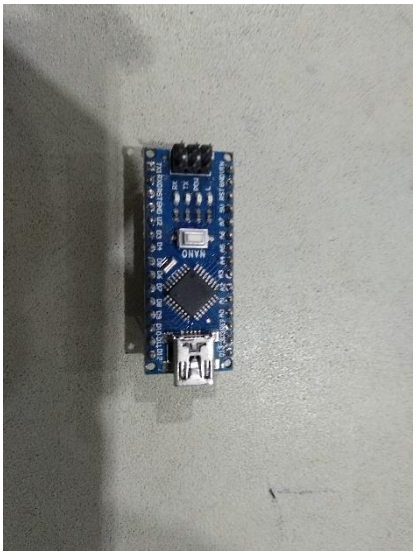
Perangkat masukan dari sistem ini adalah sensor MPU 6050. Sensor MPU 6050 sendiri merupakan sensor yang memiliki 2 sensor yang terintegrasi dalam satu chip yaitu *accelerometer* dan *gyroscope.* Sensor ini digunakan untuk mendeteksi pergerakan player terhadap gerak *accelerometer x, y, z* dan gerak *gyroscope x, y, z.* Dalam pengaplikasian alat ini digunakan *accelerometer x* dan *gyroscope z.*



Gambar 3.1 Sensor Mpu 6050

3.1.2.2 *Perangkat Proses*

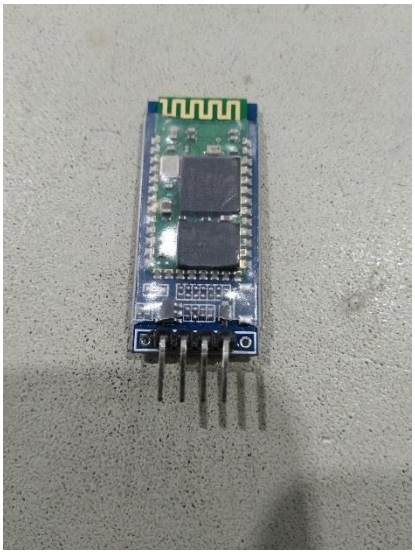
Perangkat proses dari sistem ini adalah arduino Nano. Arduino nano sendiri adalah papan mikrokontroler berbasis ATmega 328. Arduino nano digunakan sebagai alat pemrosesan data yang masuk dari sensor MPU 6050. Data dari sensor MPU 6050 yang di Proses Arduino nano akan menghasilkan data berupa koordinat dan respon player terhadap *game*, data ini akan dikirim oleh arduino nano langsung ke *game* dengan menggunakan modul Bluetooth HC-06.



Gambar 3.2 Arduino Nano

3.1.2.3 Output

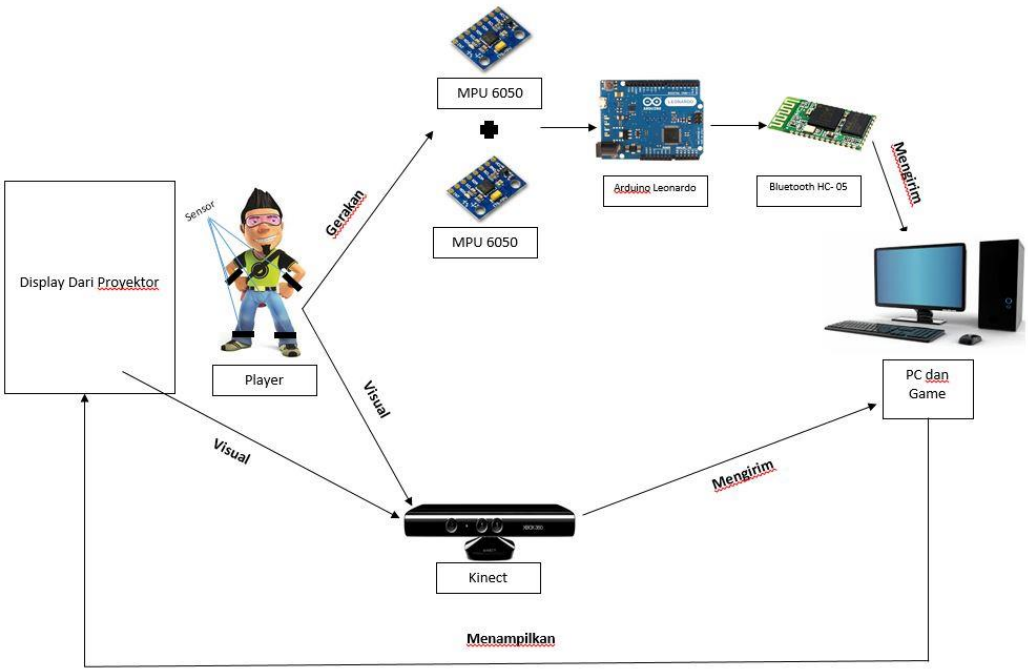
Keluaran dari hasil inputan dari sensor MPU6050 yang di proses oleh arduino nano berupa Skor. Data ini kemudian akan dikirim ke *game* dengan menggunakan modul Bluetooth HC-06.



Gambar 3.3 Arduino Nano

3.1.3 Deskripsi Umum Sistem

Pada gambar 3.4 dibawah merupakan gambaran umum sistem secara keseluruhan



Gambar 3.4 Arsitektur Sistem

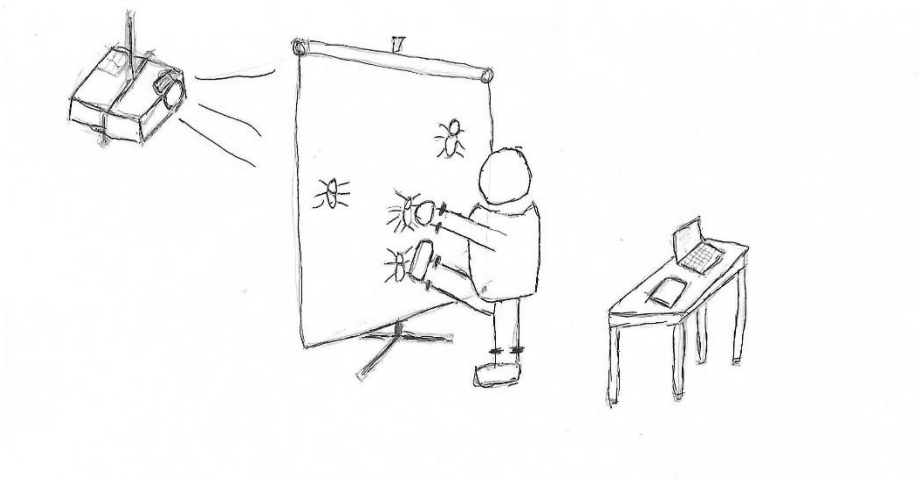
Cara kerja sistem.

a. *Game* akan menampilkan objek yang akan bergerak secara random.
b. *Game* akan ditampilkan dengan proyektor.
c. Lalu RGB Camera pada kinect akan mendeteksi jumlah dan koordinat dari objek tersebut.
d. Kinect juga akan melakukan *Skeleton* Tracking dengan memanfaatkan fitur yang dimiliki yaitu Depth Sensor. Kinect akan melakukan tracking *skeleton* pada bagian tangan dan menyimpan koordinat tangan.
e. Sensor MPU 6050 akan mendeteksi pergerakan pemain.
f. Arduino akan memproses data yang didapat dari MPU 6050 yaitu data Akselerasi data ini kemudian akan dihitung untuk menentukan nilai G.

g. Nilai G tersebut akan di hitung lagi agar menghasilkan score berdasarkan besaran nilai G.
h. Nilai G kemudian akan dikirim dengan menggunakan bluetooth HC-06.
i. Nilai G ini disimpan pada data base.
j. Software yang dibuat akan melakukan pengecekan terhadap koordinata tangan dan objek.
k. Jika koordinat objek dan tangan sama maka objek tersebut akan hilang.

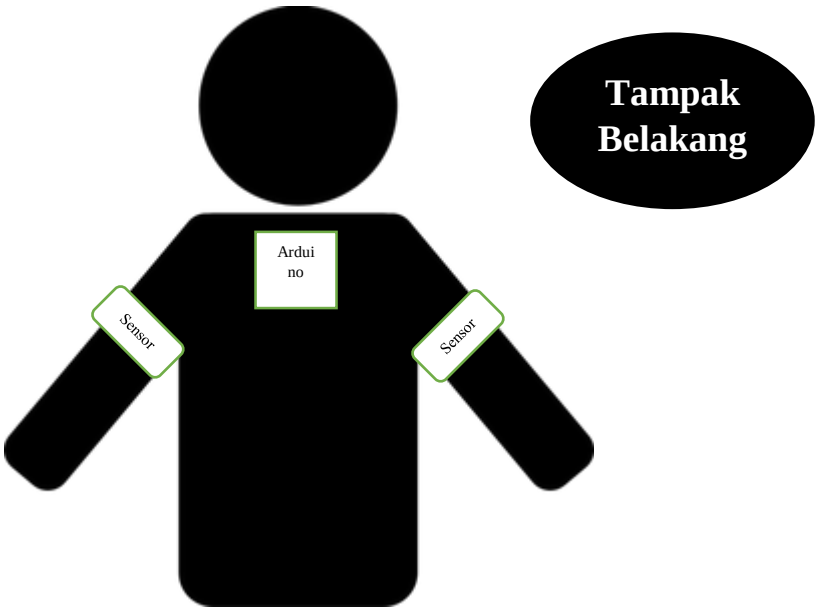
Selanjutnya akan dilakukan pengecekan terhadap nilai G untuk memberikan score terhadap aksi yang dilakukan pemain.

Gambar 3.5 dibawah merupakan gambaran simulasi sistem ketika pemain sedang bermain



Gambar 3.5 Simulasi Sistem

Pada gambar 3.5 adalah instalasi perangkat yang terhubung pada saat bermain *game*.



Gambar 3.6 Pemasangan Kontroler

Pada gambar 3.6 adalah instalasi kontroler pada pemain pada saat akan bermain *game*

3.1.4 Alur Program Pada Arduino

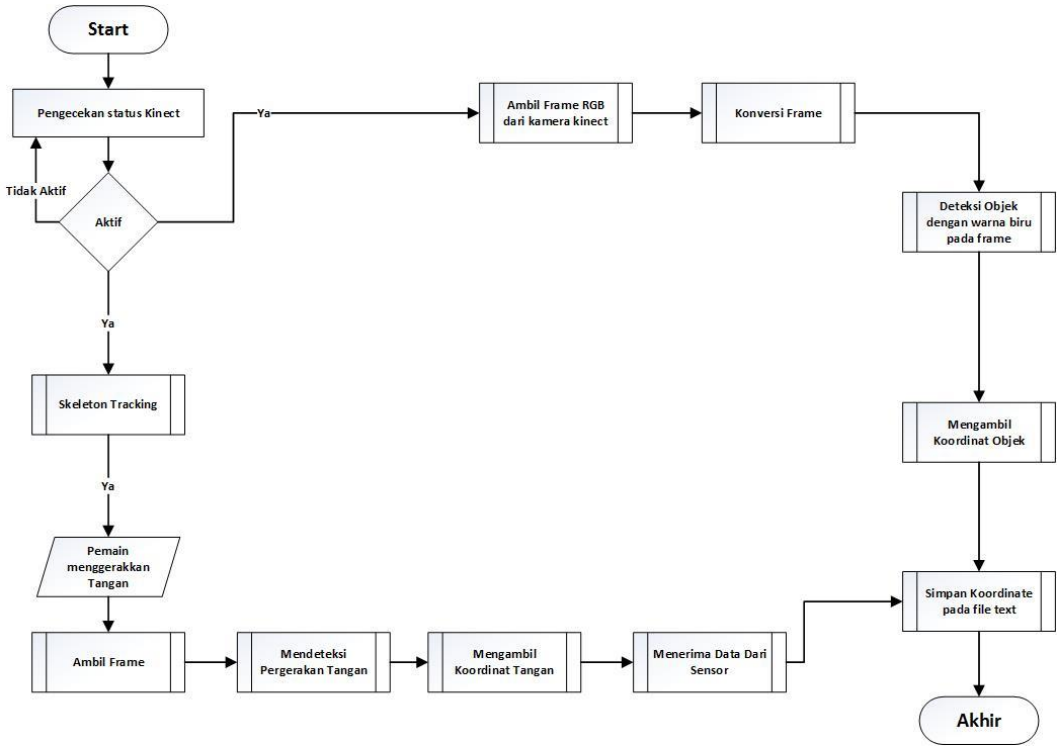
Pada gambar 3.4 dibawah adalah alur yang bekerja pada arduino.



Gambar 3.7 Alur Program pada Arduino

3.1.5 Alur Pada Program Deteksi

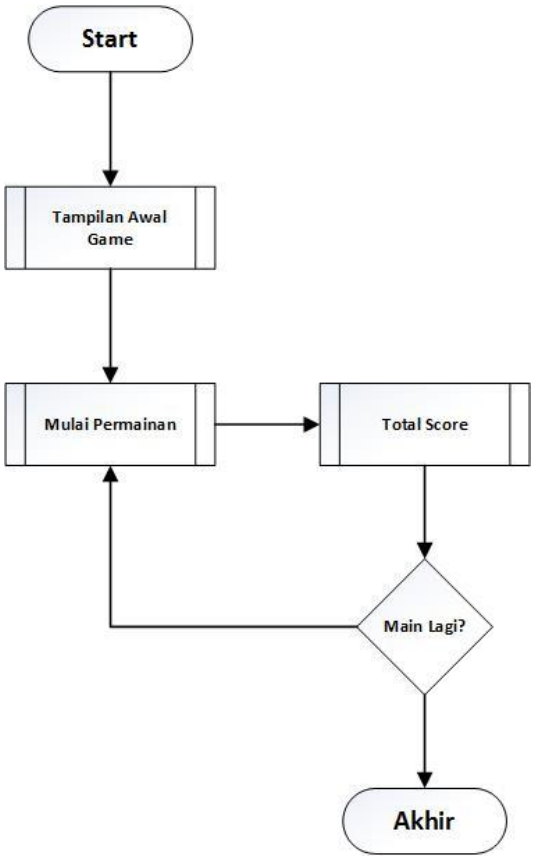
Gambar 3.5 dibawah ini menggambarkan alur kerja yang bekerja pada program deteksi.



Gambar 3.8 Alur Pada Program Deteksi

3.1.6 Alur Pada *Game*

Pada gambar 3.6 dibawah ini adalah alur kerja yang bekerja pada *game*.



Gambar 3.9 Alur Pada *Game*

3.1.7 Analisis kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah analisis yang dibutuhkan untuk menentukan spesifikasi yang dibutuhkan sistem. Spesifikasi ini juga meliputi komponen-komponen apa saja yang dibutuhkan untuk membangun sistem ini.

3.1.7.1 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras pada penelitian ini adalah perangkat keras yang digunakan dalam pembangunan kontroler ini. Adapun spesifikasi perangkat keras yang digunakan tertera pada tabel 3.1 berikut.

Tabel 3.1 Tabel Analisis Kebutuhan Perangkat Keras

No	Perangkat Keras	Keterangan
1	*Processor*	Intel Core i5-7200U 2.5 GHz
2	RAM	8 GB
3	VGA	NVIDIA GEFORCE 940MX 2 GB
4	Resolusi Layar	1920X1080
5	Kinect	Microsoft Kinect Xbox 360 v1

3.1.7.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak pada penelitian ini adalah perangkat lunak yang digunakan dalam pembangunan kontroler ini. Adapun spesifikasi perangkat lunak tertera pada tabel 3.2 sebagai berikut.

Tabel 3.2 Tabel Analisis Kebutuhan Perangkat Keras

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 10 Pro
2	Integrated Development Environment (IDE)	Visual Studio 2015
		JetBrains PyCharm Community Edition 2017.3.4
3	Pemodelan	Visio 2013
4	Software Pendukung	-Kinect SDK 1.8 -EmguCV - Py*game*

3.1.7.3 *Analisis Kebutuhan Pengguna*

Analisis kebutuhan pengguna pada penilitian ini merupakan kondisi pengguna yang dapat menggunakan kontroler ini. Adapun kondisi pengguna yang dapat menggunakan kontroler ini sebagai berikut.

Tabel 3.3 Tabel Analisis Kebutuhan Pengguna

No	Pengguna	Keterangan
1	Fisik	- Pengguna yang tidak mempunyai keterbatasan fisik.
2	Komputer	- Dapat mengoprasikan komputer

No	Pengguna	Keterangan
		- Memiliki koneksi bluetooth - Terinstal Kinect Management Service

3.1.8 Analisis kebutuhan Fungsional

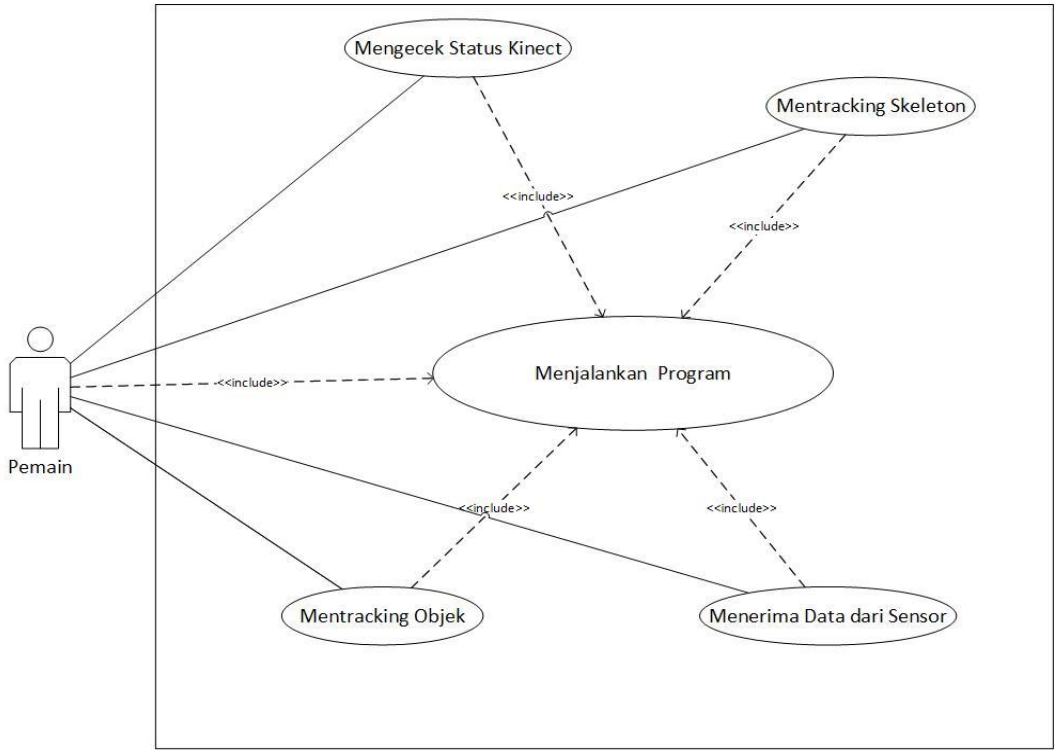
Analisis kebutuhan fungsional dibutuhkan untuk mendefinisikan pemodelan sistem yang digunakan untuk pembangunan perangkat ini. Pada penelitian ini pemodelan sistem yang digunakan yaitu *Unified Modelling Language* (UML) untuk memodelkan program deteksi dan *game*.

3.1.8.1 UML Program Deteksi

UML Program deteksi menjelaskan tentang alur dari program deteksi yang akan dibangun

3.1.8.1.1 *Use case* Diagram

Aktor yang berinteraksi dengan program deteksi hanya satu orang. Program ini adalah program deteksi dimana seorang pemain akan menjalankan program ini untuk bisa menggunakan kontroler ini.



Gambar 3.10 *Use case* Program Deteksi

3.1.8.1.2 *Use case* Skenario

Pada bagian ini akan dijelaskan skenario dari *use case* yang merupakan urutan interaksi aktor dan program deteksi. *Use case* skenario terdapat pada table 3.4 berikut.

Tabel 3.4 *Use case* Skenario Program Deteksi

No	1
Usecase	Menjalankan Program
Tujuan	Untuk mendapatkan footage dari Kinect
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Program menampilkan footage yang ditangkap oleh kinect
Skenario Utama	
1.	Pengguna menjalankan program deteksi
No	2
Usecase	Mengecek Status Kinect
Tujuan	Untuk mendapatkan status dari kinect
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Status Kinect aktif
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukan pengecekan status kinect
No	3
Usecase	Mentracking Objek
Tujuan	Untuk melakukan tracking terhadap *skeleton* pemain
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Frame *skeleton* diambil dan di filter bagian tangan kanan, dan tangan kiri
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukan tracking dan filter dibelakang layar
No	4
Usecase	Objek Tracking
Tujuan	Untuk melakukan tracking objek berwarna biru pada frame yang di tangkap kinect
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Mendapatkan koordinat objek biru yang ada pada frame
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukan tracking dibelakang layar

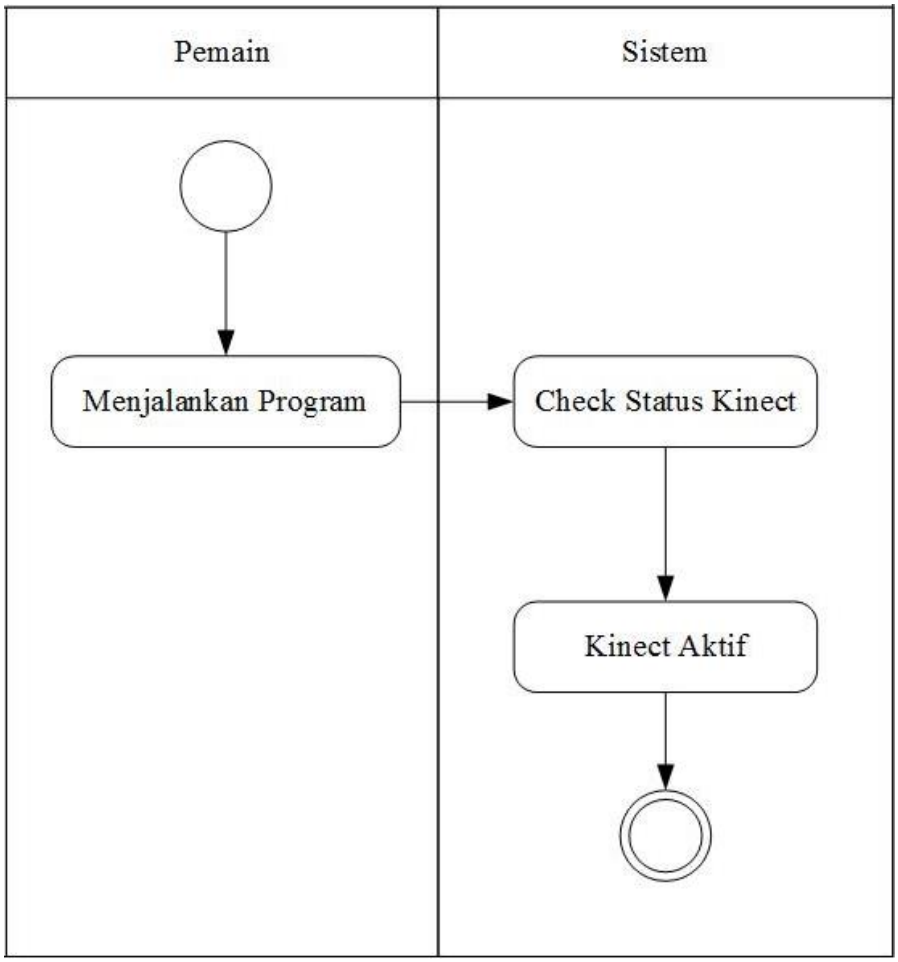
No	5
Usecase	Menerima Data dari Sensor
Tujuan	Untuk mendapatkan data kekuatan pergerakan tangan pemain dalam bentuk *score*
Aktor	Pengguna
Kondisi Awal	Pengguna berada pada tampilan desktop
Kondisi Akhir	Mendapatkan kekuatan pergerakan tangan pemain berupa *score*
Skenario Utama	
1.	Pengguna menjalankan program deteksi
2.	Sistem melakukan tracking dibelakang layar

3.1.8.1.3 *Activity Diagram*

Activity Diagram menggambarkan alur aktifitas dari sistem, diagram ini digunakan untuk melihat aktifitas yang dilakukan oleh sistem.

3.1.8.1.3.1 *Activity* Mengecek Status Kinect

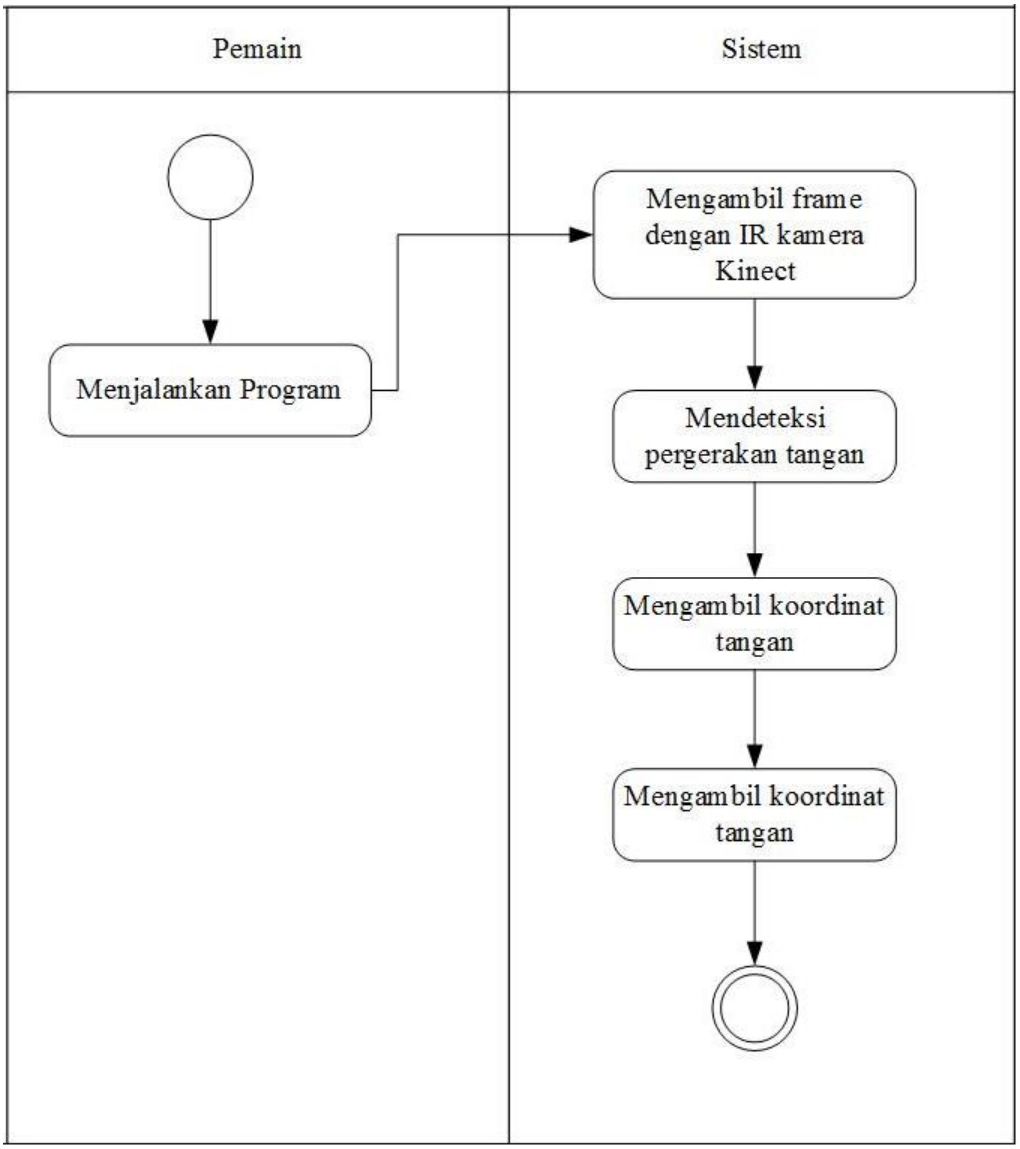
Berikut ini merupakan alur aktifitas yang dilakukan pemain saat melakukan check status Kinect.



Gambar 3.11 *Activity* Mengecek Status Kinect

3.1.8.1.3.2 *Activity* Mentracking *Skeleton*

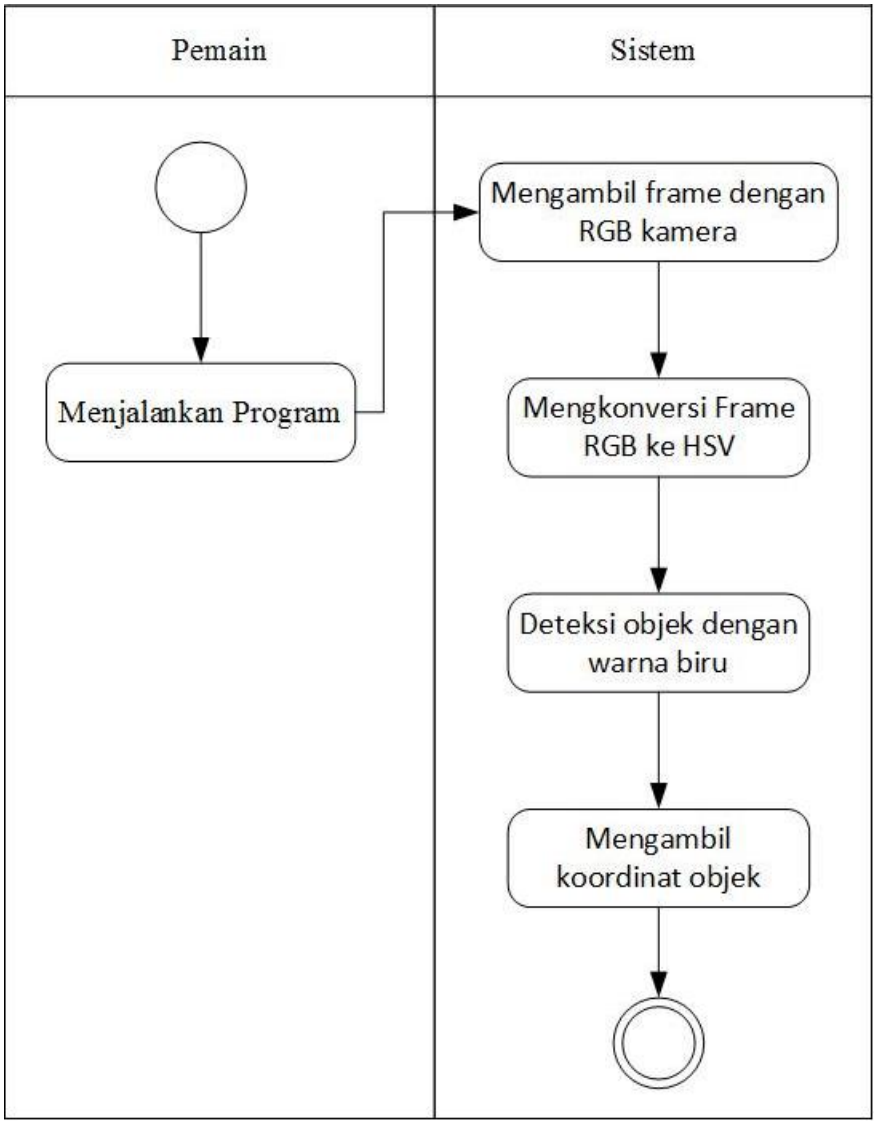
Berikut ini adalah alur aktifitas *skeleton tracking* yang dilakukan ketika melakukan pemain menjalankan program deteksi. Aktifitas yang berjalan dibelakang layar dapat dilihat pada gambar 3.11 berikut.



Gambar 3.12 *Activity* Mentracking *Skeleton*

3.1.8.1.3.3 *Activity* Mentracking Objek

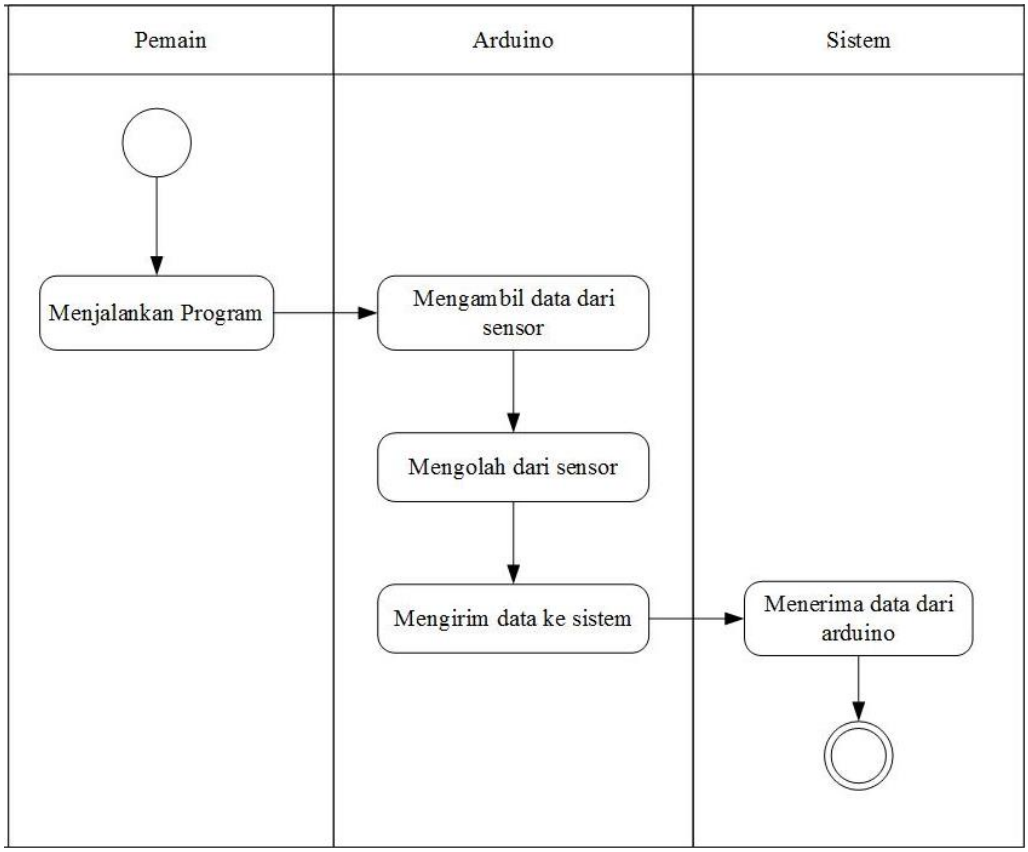
Gambar 3.12 berikut menggambarkan alur aktifitas *Object Tracking* yang dilakukan pada saat pemain menjalankan program deteksi. Aktifitas dibelakang layar dapat dilihat pada gambar berikut.



Gambar 3.13 *Activity* Mentracking Objek

3.1.8.1.3.4 Menerima Data Dari Sensor

Gambar 3.13 berikut menggambarkan alur aktifitas penerimaan data data dari sensor yang dilakukan pada saat pemain menjalankan program deteksi. Aktifitas dibelakang layar dapat dilihat pada gambar berikut.



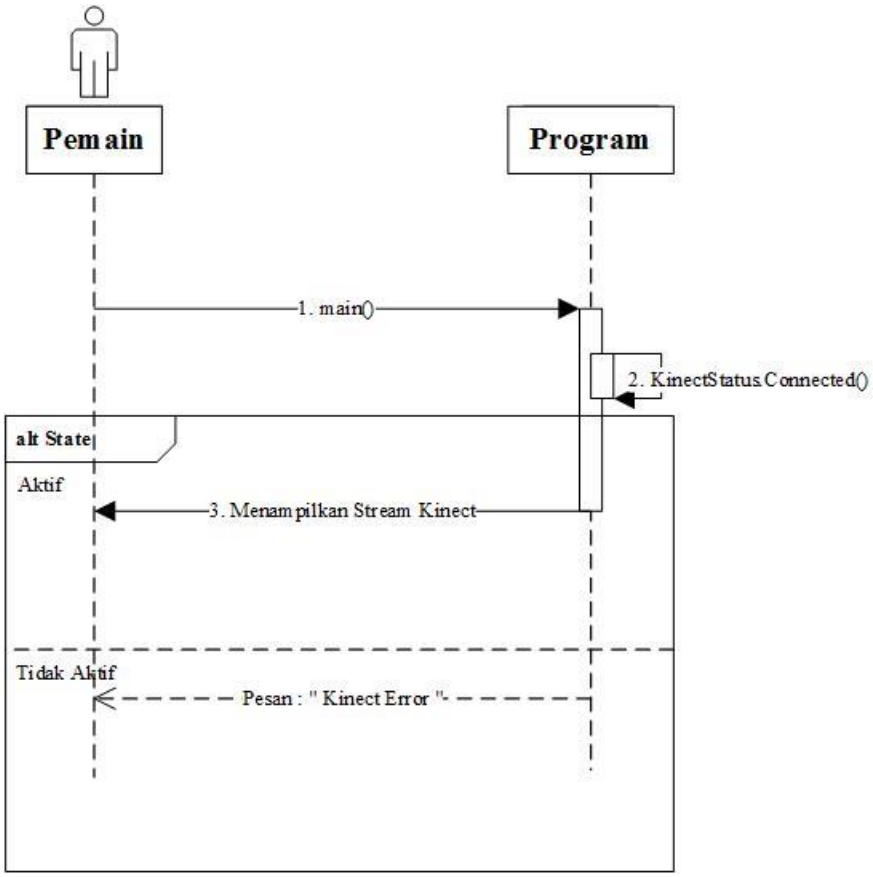
Gambar 3.14 *Activity* Menerima Data dari Sensor

3.1.8.1.4 *Sequence Diagram*

Sequence Diagram adalah diagram yang menggambarkan rangkaian pesan dan interaksi antara objek. Berikut adalah *Sequence Diagram* pada program deteksi.

3.1.8.1.4.1 *Sequence Diagram* Mengecek Status Kinect

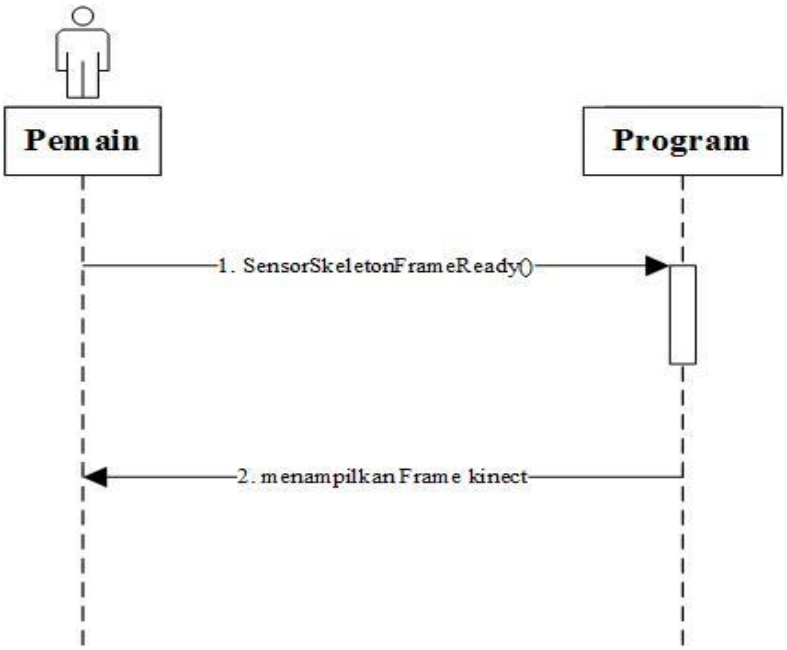
Berikut adalah *sequence diagram* dari proses check status Kinect pada program deteksi.



Gambar 3.15 *Sequence Diagram* Mengecek Status Kinect

3.1.8.1.4.2 *Sequence Diagram* Mentracking *Skeleton*

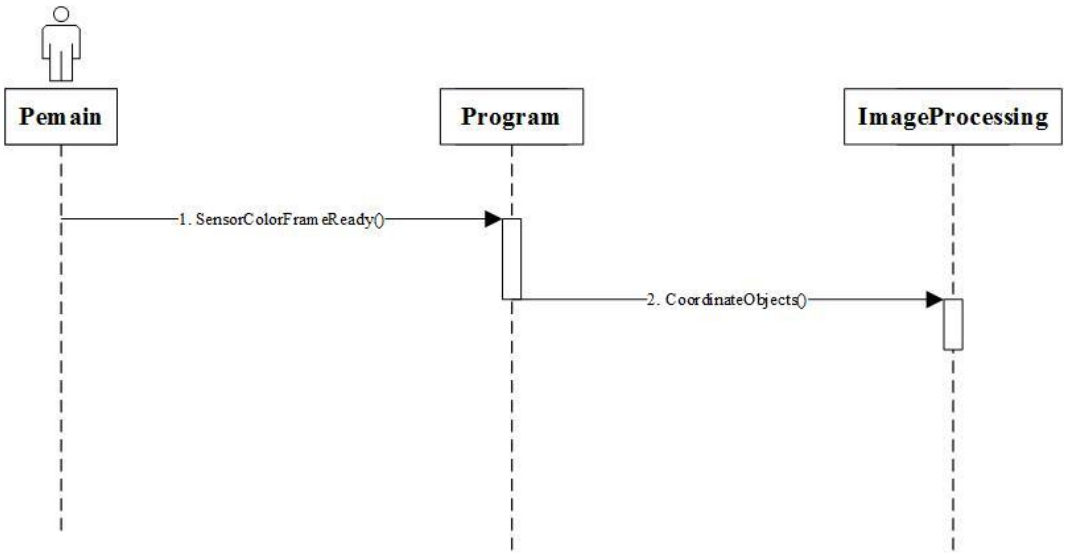
Berikut adalah *sequence diagram* dari proses *skeleton tracking* pada program deteksi.



Gambar 3.16 *Sequence Diagram* Mentracking *Skeleton*

3.1.8.1.4.3 *Sequence Diagram* Mentracking Objek

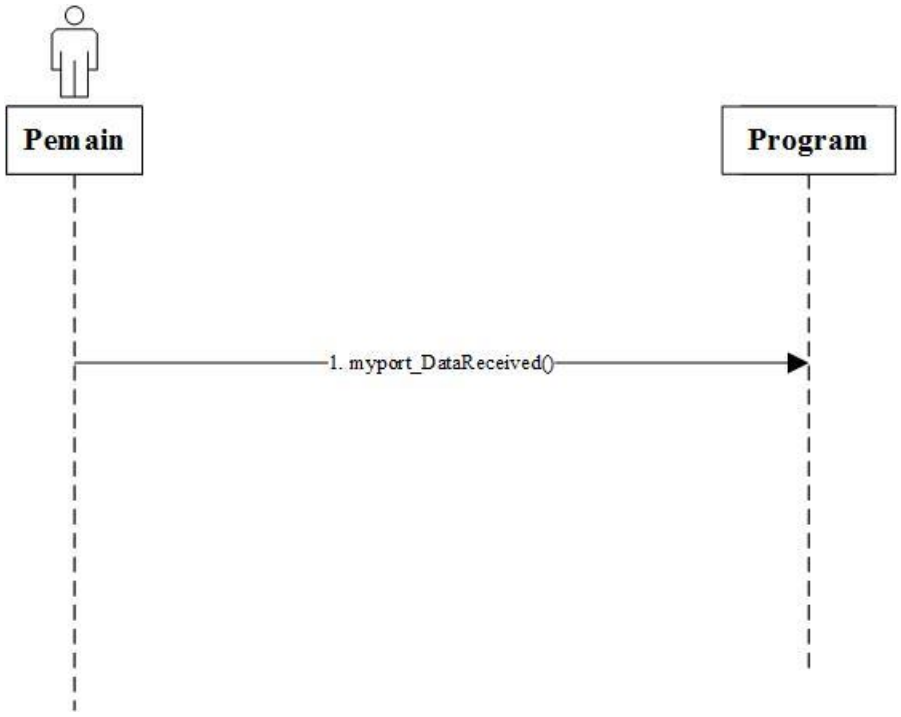
Berikut adalah *sequence diagram* dari proses object tracking pada program deteksi.



Gambar 3.17 *Sequence Diagram* Mentracking Objek

3.1.8.1.4.4 *Sequence Diagram* Menerima Data dari Sensor

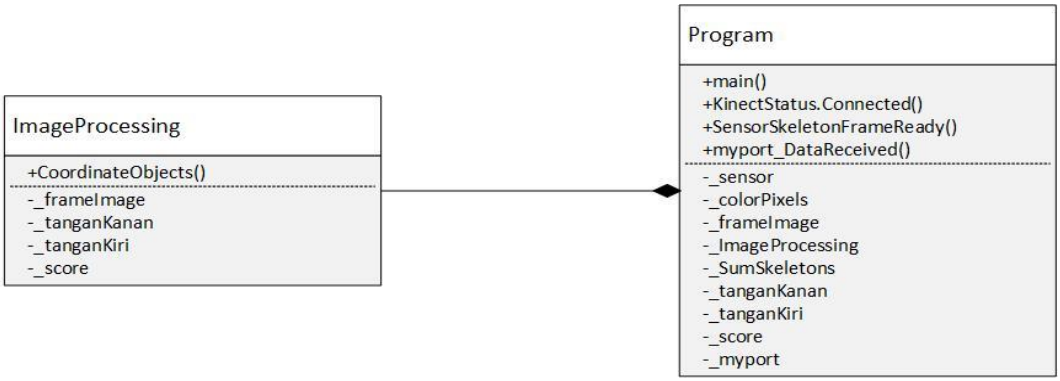
Berikut ini adalah *sequence diagram* pada proses menerima data dari sensor pada program deteksi.



Gambar 3.18 *Sequence Diagram* Menerima Data dari Sensor

3.1.8.1.5 *Class Diagram*

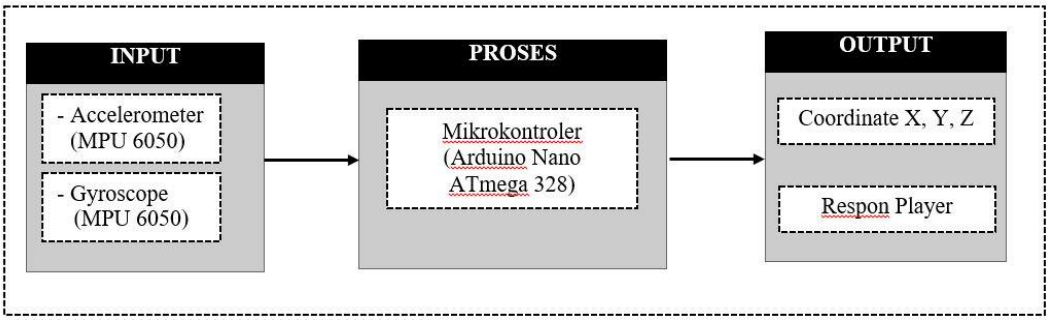
Class Diagram menggambarkan objek-objek yang berada dalam sistem dalam kasus ini program deteksi, pada gambar 3.18 berikut adalah class diagram pada sistem deteksi.



Gambar 3. 19 *Class Diagram* Program Deteksi

3.2 Perancangan Sistem

Pada perancangan Sistem Kontroler *Game* Berbasis Sensor Gyroscope dan Accelerometer ini, secara umum ada 3 bagian utama yaitu input, proses, dan output. Fungsi dari setiap bagian akan saling berkaitan dalam sistem ini.

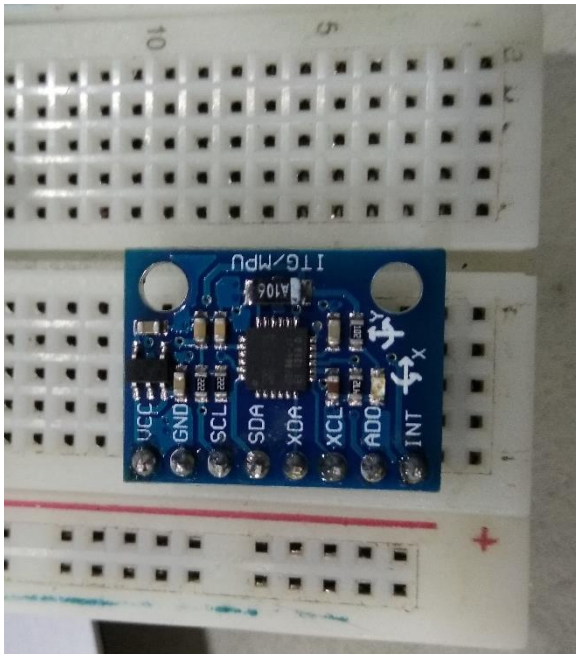


Gambar 3.20 Blok Diagram Sistem Kontroler *Game* Berbasis Sensor Gyroscope dan Accelerometer

Secara umum cara kerja dari Sistem Kontroler *Game* Berbasis Sensor Gyroscope dan Accelerometer ini adalah mendeteksi pergerakan dari player. Pergerakan player dideteksi dengan menggunakan sensor MPU 6050 untuk menghitung gerak tangan terhadap sumbu *accelerometer x* dan *gyroscope z*. hasil dari deteksi ini akan memberi nilai masukan ke arduino. Selanjutnya akan diproses pengolahan data sehingga menghasilkan koordinat pergerakan player dan respon terhadap *game*.

3.2.1 Perangkat Masukan (Input)

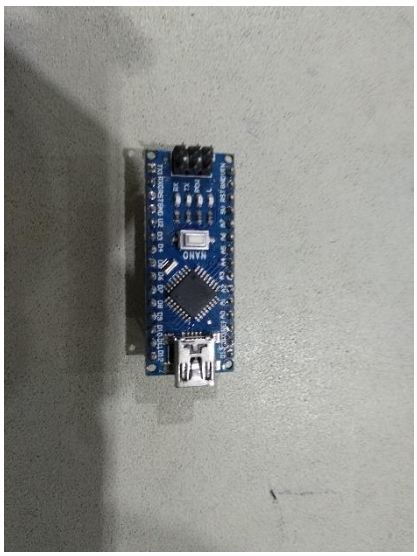
Perangkat masukan dari sistem ini adalah sensor MPU 6050. Sensor MPU 6050 sendiri merupakan sensor yang memiliki 2 sensor yang terintegrasi dalam satu chip yaitu accelerometer dan gyroscope. Sensor ini digunakan untuk mendeteksi pergerakan player terhadap gerak accelerometer x, y, z dan gerak gyroscope x, y, z. Dalam pengaplikasian alat ini digunakan accelerometer x dan gyroscope z.



Gambar 3.21 Sensor Mpu 6050

3.2.2 Perangkat Proses

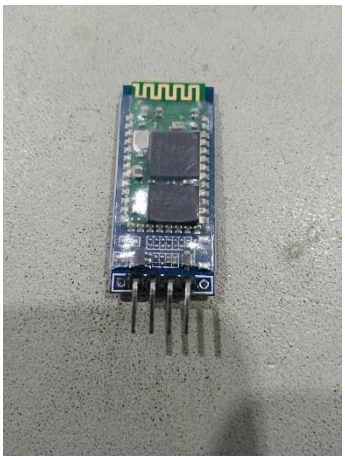
Perangkat proses dari sistem ini adalah arduino Nano. Arduino nano sendiri adalah papan mikrokontroler berbasis ATmega 328. Arduino nano digunakan sebagai alat pemrosesan data yang masuk dari sensor MPU 6050. Data dari sensor MPU 6050 yang di Proses Arduino nano akan menghasilkan data berupa koordinat dan respon player terhadap *game*, data ini akan dikirim oleh arduino nano langsung ke *game* dengan menggunakan modul Bluetooth HC-06.



Gambar 3.22 Arduino Nano

3.2.3 Output

Keluaran dari hasil inputan dari sensor MPU6050 yang di proses oleh arduino nano berupa koordinat X, Y, Z dan respon player. Data ini kemudian akan dikirim ke *game* dengan menggunakan modul Bluetooth HC-06.



Gambar 3.23 Bluetooth HC-06

3.3 Perancangan Permainan

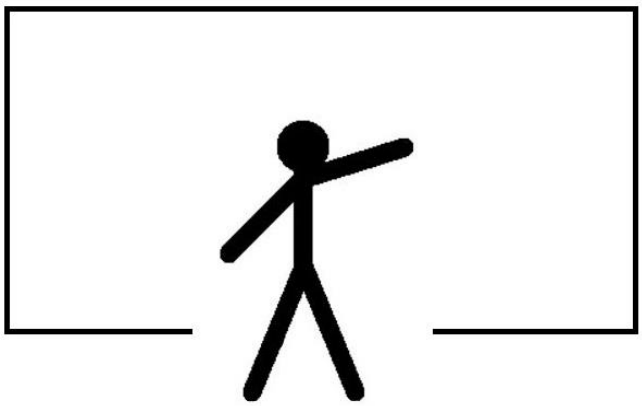
Perancangan *game* disini bertujuan untuk membuat sebuah *game* yang bertujuan untuk melakukan pengujian terhadap alat yang dibuat. *Game* yang dibuat disini adalah *game* sederhana hal ini dikarenakan *game* dalam penelitian

ini hanya untuk pengujian namun tetap mengadopsi konsep *augmented climbing wall*.

3.3.1 Gerakan Yang Akan Dideteksi

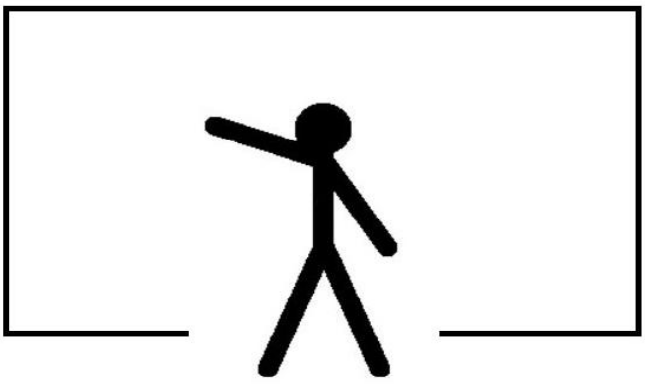
Adapun gerakan-gerakan yang akan dideteksi pada permaianan ini adalah pukul kanan atas, pukul kiri atas, pukul kanan tengah, pukul kiri tengah, pukul kanan bawah, pukul kiri bawah, pukul kanan depan, dan pukul kiri depan.

a. Pukul Kanan Atas



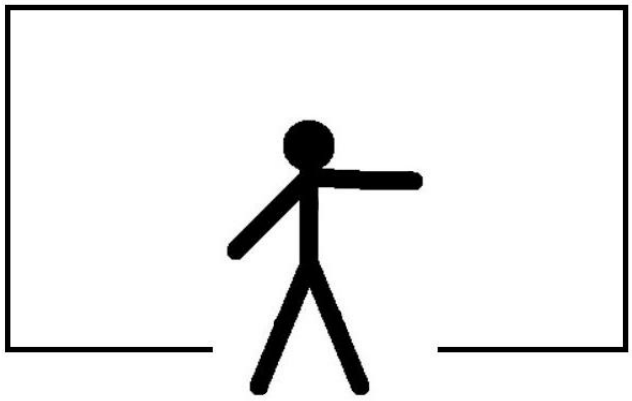
Gambar 3.24 Pukul Kanan Atas

b. Pukul Kiri Atas



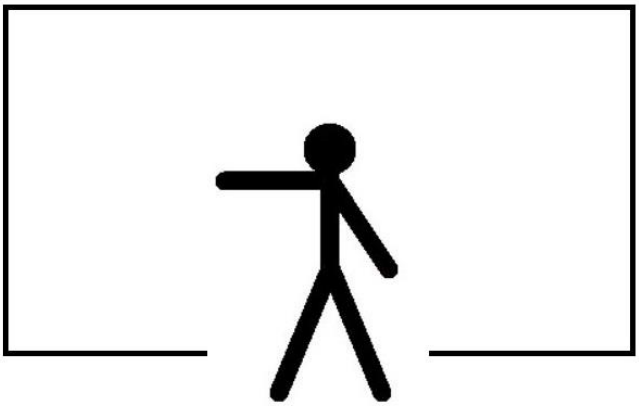
Gambar 3.25 Pukul Kiri Atas

c. Pukul Kanan Tengah



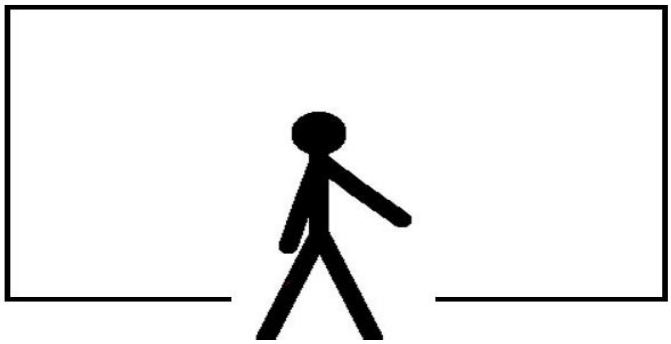
Gambar 3.26 Pukul Kanan Tengah

d. Pukul Kiri Tengah



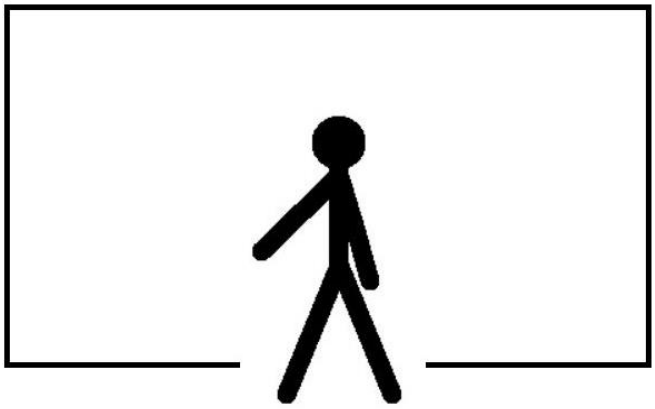
Gambar 3.27 Pukul Kiri Tengah

e. Pukul Kanan Bawah



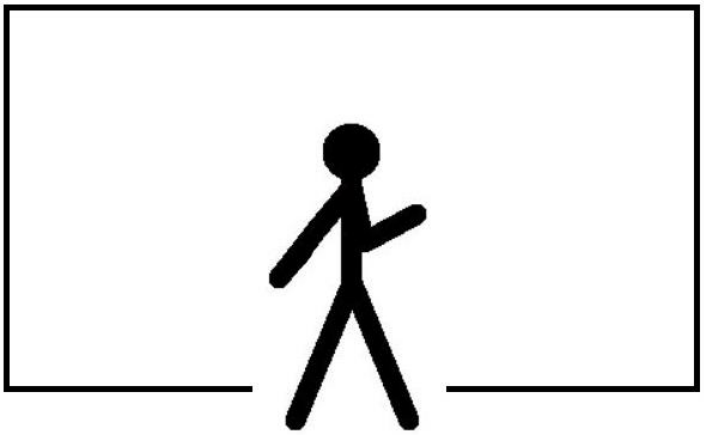
Gambar 3.28 Pukul Kanan Bawah

f. Pukul Kiri Bawah



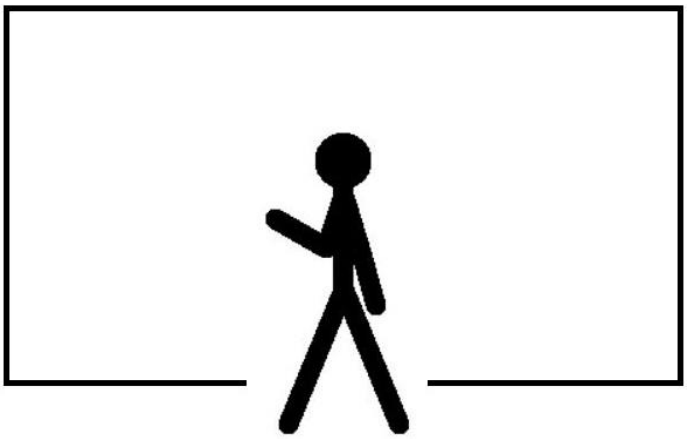
Gambar 3.29 Pukul Kiri Bawah

g. Pukul Kanan Depan



Gambar 3.30 Pukul Kanan Depan

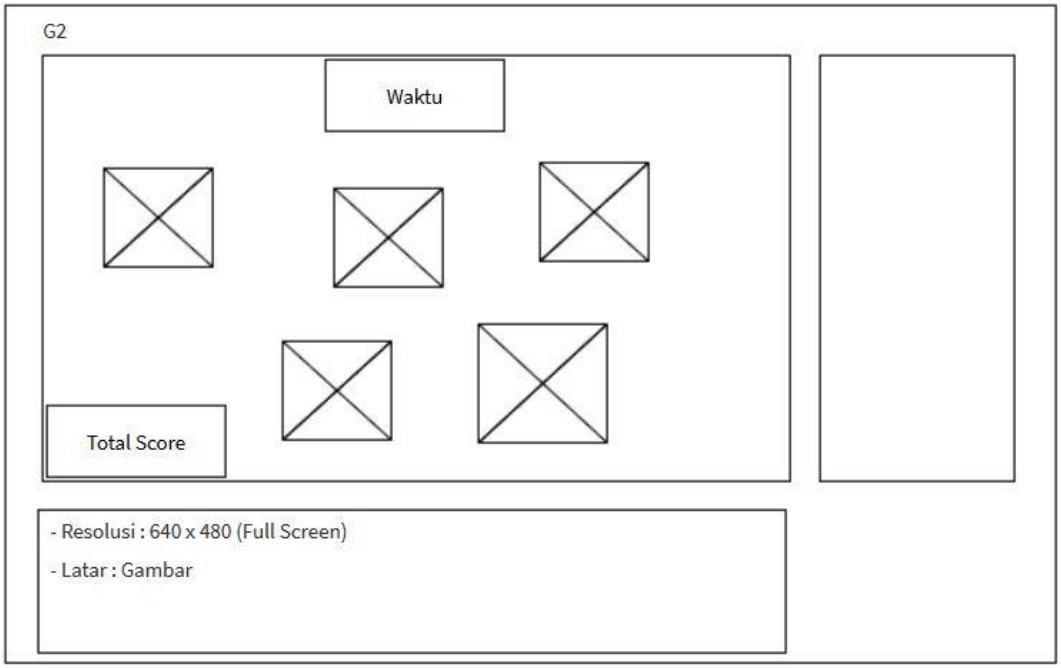
h. Pukul Kiri Depan



Gambar 3.31 Pukul Kiri Depan

3.3.2 Mockup Permainan

Resolusi *game* disesuaikan dengan resolusi Kinect yaitu 640 x 480, objek pada *game* yaitu laba-laba yang akan bergerak secara random. Tampilan *game* pada saat *game* sedang dimainkan dapat dilihat pada gambar 3.24 berikut.



Gambar 3.32 Tampilan *Game*