

## BAB IV

### IMPLEMENTASI SISTEM DAN EVALUASI

#### 4.1 Implementasi Sistem

Tahap ini merupakan pembuatan perangkat lunak yang disesuaikan dengan rancangan atau desain sistem yang telah dibuat. Aplikasi yang dibuat akan diterapkan berdasarkan kebutuhan.

Aplikasi *Virtual Punch Training* ini membutuhkan Kinect sebagai media untuk menangkap gambar *user* yang nantinya setiap *user* melakukan gerakan akan diproses melalui Desktop ataupun Laptop, setelah itu *user* dapat melakukan interaksi dengan tampilan *GUI*.

Sensor 3-Dimensi yang dimiliki oleh Kinect mampu menangkap setiap gerakan yang dilakukan oleh *user*. Gerakan yang dideteksi tersebut diubah menjadi titik koordinat pada bagian tubuh tertentu secara 3-Dimensi (x,y,z). Titik-titik koordinat ini memiliki tipe data *float*. Dari data inilah nantinya komputer dapat memproses gerakan yang ditangkap oleh Kinect.

##### 4.1.1 Instalasi dan Kebutuhan Perangkat

###### A. Kebutuhan Perangkat Keras

Kebutuhan minimum perangkat keras yang diperlukan untuk menjalankan aplikasi ini adalah:

- Prosesor Quad Core 2.6 Ghz.
- RAM 4GB
- Monitor dengan resolusi minimal 1280 x 1024

- VGA dedicated 512MB.
- Dedicated USB 2.0 Port.
- Microsoft Kinect for Xbox 360 atau Microsoft Kinect for Windows

## **B. Kebutuhan Perangkat Lunak**

Adapun kebutuhan perangkat lunak yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

- Sistem Operasi Microsoft Windows 7 Ultimate.
- Microsoft Visual Studio 2010 Ultimate.
- Untuk dokumentasi menggunakan Microsoft Office Word 2007.
- Untuk mengakses Kinect menggunakan SDK Microsoft Kinect.
- .Net Framework 4.0

## **C. Instalasi Program dan Pengaturan Sistem**

Pegembangan aplikasi *Virtual Punch Training* membutuhkan perangkat lunak yang telah terinstall, adapun tahapan-tahapan instalasi dan pengaturan sistem adalah sebagai berikut:

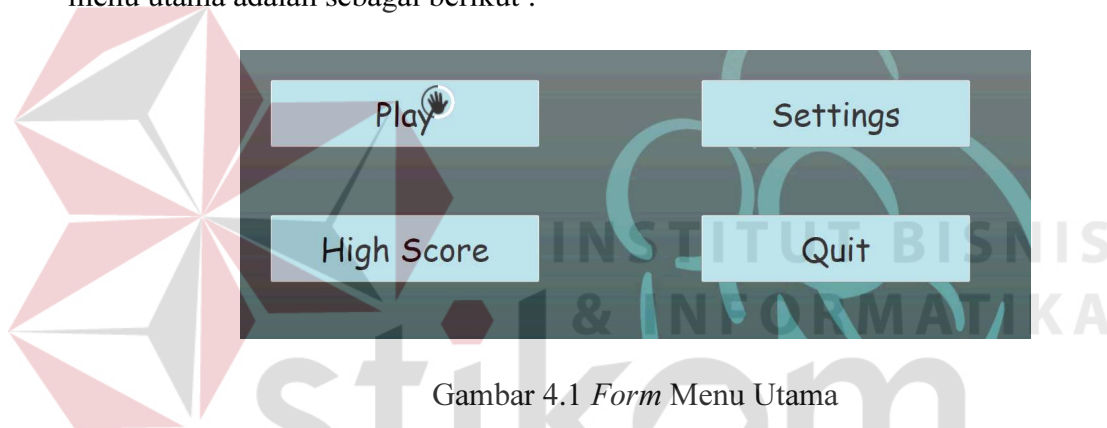
- . Install OS Microsoft Windows 7 Ultimate
- . Install Microsoft Visual Studio 2010 bahasa C#.
- . Install Kinect SDK versi 1.7

## 4.2 Penjelasan Hasil Implementasi

Setelah kebutuhan sistem terpenuhi langkah selanjutnya adalah mengimplementasikan rancangan sistem kedalam sebuah aplikasi *Virtual Punch Training*.

### 4.2.1 Form Menu Utama

*Form* ini pertama kali ditampilkan setelah aplikasi di *start*. Pada *form* menu utama terdapat pilihan-pilihan menu yang dapat dipilih oleh *user*. Tampilan *form* menu utama adalah sebagai berikut :



Gambar 4.1 *Form* Menu Utama

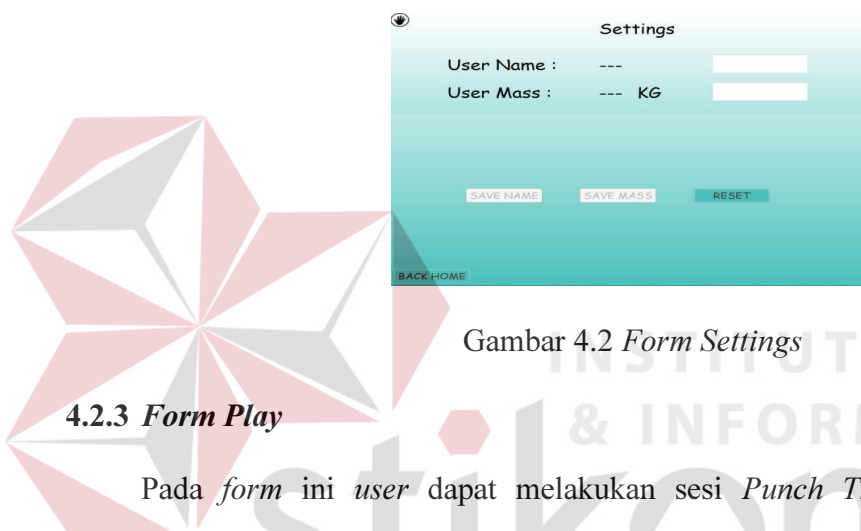
Berikut adalah penjelasan dari masing-masing tombol yang digunakan pada aplikasi *Virtual Punch Training*.

Tabel 4.1 Deskripsi Tombol

<i>Menu</i>	Deskripsi
<i>Play</i>	Berfungsi untuk memanggil <i>form play</i> , dan <i>user</i> dapat memulai sesi latihannya
<i>High Score</i>	Berfungsi untuk menampilkan <i>form</i> nilai terbaik dari sesi latihan
<i>Settings</i>	Berfungsi untuk menampilkan <i>form</i> pengaturan, dimana dapat mengisi nama dan berat badan <i>user</i>
<i>Quit</i>	Berfungsi untuk menutup aplikasi

### 4.2.2 Form Settings

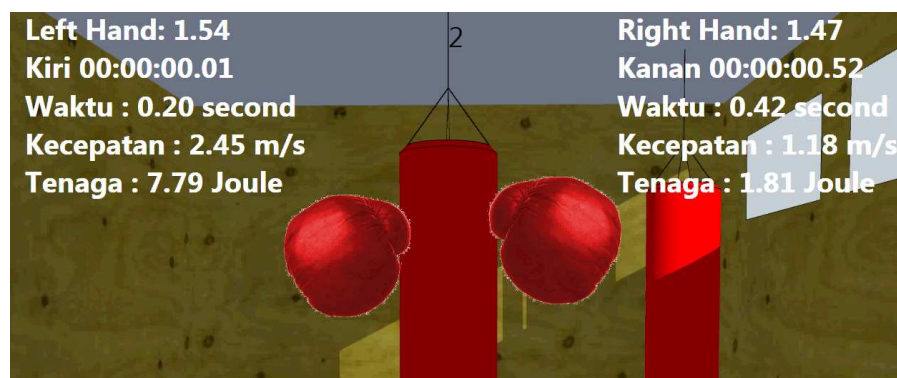
*Form Settings* adalah *form* yang wajib diisi datanya sebelum mulai melakukan latihan, *form* ini berfungsi untuk mengubah data nama dan data berat badan *user* yang dibutuhkan untuk perhitungan aplikasi dan pencatatan score selama latihan. Terdapat beberapa tombol diantaranya *Save Name* berfungsi untuk menyimpan nama *user*. *Save Mass* berfungsi untuk menyimpan berat badan *user*. *Reset* digunakan untuk mengembalikan ke nilai default seperti yang ada digambar



Gambar 4.2 *Form Settings*

### 4.2.3 Form Play

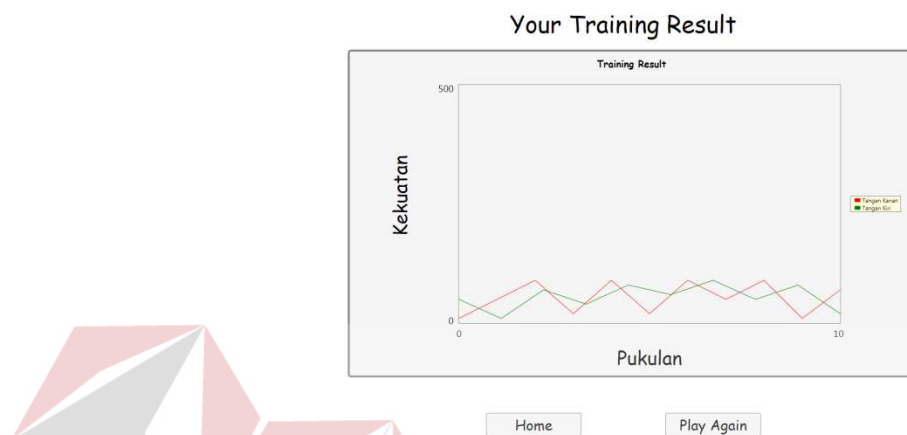
Pada *form* ini *user* dapat melakukan sesi *Punch Training*, *user* dapat memukul objek virtual yang ditampilkan. Setiap kali pukulan *user* mengenai target maka akan diberi poin nilai sesuai hasil perhitungan yang telah ditentukan sebelumnya. Waktu akan berjalan dan apabila waktu telah habis maka sesi latihan berakhir.



Gambar 4.3 *Form Play*

#### 4.2.4 Form Hasil Latihan

Setelah waktu untuk satu sesi berakhir maka akan ditampilkan sebuah *form* yang berisi informasi grafik tenaga yang dihasilkan setiap pukulan yang dilakukan pada sesi tersebut.



Gambar 4.4 Form Hasil Latihan

#### 4.2.5 Form High Score

Pada *Form High Score* akan ditampilkan nilai terbaik saat melakukan *training*, terdapat beberapa informasi yang akan ditampilkan dalam *form* ini diantaranya nomor urut, nama *user*, pukulan terbaik dari tangan kiri dan kanan, lalu jumlah score selama latihan.

No.	NAME	L-HAND		R-HAND	
		MAX	MIN	MAX	MIN
1	wira	146.7434	0.950488	3241.563	5.064229
2	wira	66.18691	3.469313	135.0089	3.461876
3	wira	42.82683	0.0002027598	47.05418	5.581107
4	wira	20.746	0.1981588	26.78977	1.071591
5	wira	20.418	2.572158	23.2804	0.2442584

Gambar 4.5 Form High Score

### 4.3 Uji Coba Fungsionalitas Aplikasi dan Pembahasan

Uji coba fungsionalitas aplikasi dan pembahasan dilakukan untuk mengetahui apakah fungsi-fungsi yang ada pada aplikasi *Virtual Punch Training* telah berjalan dengan baik atau tidak. Selain itu juga akan membahas masalah-masalah yang mungkin masih terdapat pada aplikasi. Adapun fungsi-fungsi yang akan diujikan adalah sebagai berikut:

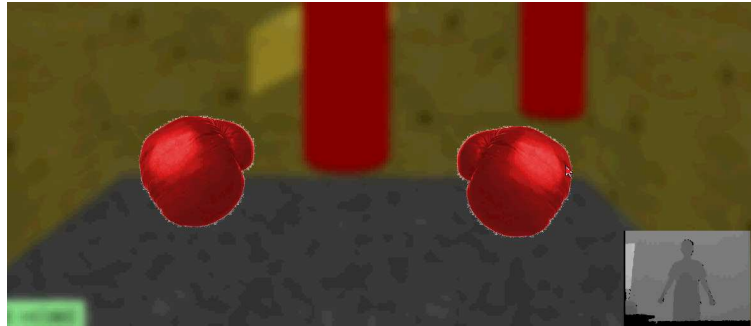
#### 4.3.1 Proses Mendeteksi *User*

Proses ini dilakukan untuk menguji apakah aplikasi dapat mendeteksi *user* dengan baik. Pengujian dilakukan dengan cara mendeteksi *user* yang berdiri tegak didepan Kinect pada ruangan seluas 2x2 meter dan *user* berdiri dengan jarak minimal 1,5 meter di depan Kinect. Hasil uji coba pada tabel berikut

Tabel 4.2 Uji Coba Pendeteksian *User*

Test Case ID	Tujuan	Masukkan	Keluaran yang diharapkan	Status
1.	Mendeteksi jarak <i>user</i> dari Kinect dalam ruangan 2x2 meter.	<i>User</i> berdiri tegak didepan Kinect dengan posisi siap.	Jarak <i>user</i> ditampilkan oleh aplikasi.	Sesuai
2.	Penempatan ID setiap tangan.	<i>User</i> berdiri tegak didepan Kinect dengan posisi siap.	Tampilan sarung tangan tinju pada posisi tangan <i>user</i> .	Sesuai

Pada uji coba ini aplikasi dapat mendeteksi *user*, namun pendeteksian masih belum 100% akurat, karena terdapat beberapa kesalahan deteksi saat *user* menggerakkan tangannya. Selain itu terdapat juga *respond delay* gerakan antara gerakan *pointer* dengan gerakan *user*. Hal ini disebabkan oleh keterbatasan sensor kamera Kinect. Solusi yang dapat diberikan adalah melakukan *restart form* agar Kinect dapat mendeteksi ulang *user*.



Gambar 4.6 Proses pendeteksian *User*

### 4.3.2 Proses Navigasi Menu

Proses ini dilakukan untuk menguji apakah menu yang terdapat dalam aplikasi dapat berfungsi sesuai yang diinginkan. Pengujian dilakukan dengan cara memilih menu dari *Play*, *Settings*, *Back*, *Home*, *Play Again* dan *Quit*. Pada uji coba ini semua tombol menu dapat dipilih dan bernavigasi ke *form* yang dituju dengan semestinya.



Gambar 4.7 Proses navigasi Menu

### 4.3.3 Proses Mengubah *Settings*

Proses ini dilakukan untuk menguji apakah menu *Settings* yang terdapat dalam aplikasi dapat berfungsi sesuai yang diinginkan. Pengujian dilakukan terhadap 2 *textbox* dan 3 *button*, *textbox* pertama yaitu *textbox* nama, dilakukan pemasukan data berupa *text* dan *number* maka *number* tidak muncul kedalam *textbox*, kemudian melakukan uji coba *button save* nama yang berfungsi untuk menyimpan masukan pada *textbox* nama dan berhasil dengan sukses. Selanjutnya

uji coba pada *textbox* kedua yaitu *textbox* massa, dilakukan pemasukan data berupa *text* dan *number* maka *text* tidak muncul kedalam *textbox*, kemudian melakukan uji coba *save* massa yang berfungsi menyimpan masukan pada *textbox* massa dan berhasil dengan sukses. Yang terakhir melakukan uji coba pada *button* *reset* yang berfungsi mengembalikan nilai dari nama dan massa ke asal, saat *button* di klik maka data nama dan massa tersebut terhapus dengan sukses. Hasil coba pengujian pada tabel berikut.

Tabel 4.3 Uji Coba *Settings*

Test Case ID	Tujuan	Masukkan	Keluaran yang diharapkan	Status
1.	Mengubah nama user "wira"	Nama user "wira"	Nama user "wira" tampil pada <i>textblock</i> .	Sesuai
2.	Mengubah massa user dengan "55" kilogram	Massa user "55"	Massa user "55" tampil pada <i>textblock</i>	Sesuai
3.	Menghapus semua data setingan yang ada didalam aplikasi	Klik tombol Reset	Nama dan Massa user dihapus dari aplikasi	Sesuai

Settings  
SURABAYA

User Name : wira

User Mass : 55 KG

SAVE NAME SAVE MASS RESET

Gambar 4.8 Uji Coba *Settings*



#### 4.3.4 Proses *Training Session*

Proses ini dilakukan untuk menguji apakah fungsi *timer*, fungsi perhitungan kekuatan pukulan yang terdapat dalam aplikasi dapat berfungsi sesuai yang diinginkan. Pengujian dilakukan dengan cara *user* berdiri didepan Kinect siap untuk berlatih, kemudian memilih menu yang tampil, saat *timer* berjalan *user* melakukan pukulan kearah target. Pada uji coba ini semua fungsi berjalan sebagaimana mestinya, tetapi pada pukulan cepat terdapat *respond delay* pada Kinect dan aplikasi sehingga terkadang beberapa pukulan tidak terdeteksi dengan akurat waktu dan tenaganya. Untuk mengatasi hal tersebut jeda antar masing-masing pukulan harus dibuat lama sekitar 1-2 detik sebelum memukul kembali agar Kinect dan aplikasi dapat kembali sinkron selama terjadi jeda. Jeda tersebut biasanya muncul apabila pukulan terlalu intensif, penerangan yang kurang, terlalu banyak objek dibelakang *user* baik itu berupa orang ataupun benda, jarak *user* terhadap Kinect semakin mendekat tidak sesuai dengan jarak yang sudah ditentukan yaitu 150 cm. Seperti yang disajikan dalam gambar 4.9.

#### 4.3.5 Proses Menghitung Kecepatan Pukulan

Proses ini berjalan disaat proses *Training Session* berlangsung, saat salah satu tangan telah terdeteksi oleh Kinect maka program akan memberi ID pada tangan tersebut dan membuat sebuah *stopwatch*, *stopwatch* tersebut akan mulai berjalan pada saat tangan yang melakukan pukulan berjarak 150 cm dari Kinect dan akan berhenti setelah tangan berjarak 100 cm dari Kinect, kemudian waktu tersebut dicatat dan dikalkulasikan dengan cara yang telah ditentukan dibagian

analisa. Pada saat tangan kembali ke posisi awal dengan jarak 150 cm atau lebih maka *stopwatch* akan direset. Berikut tabel keluaran kecepatan pukulan:

Tabel 4.4 Keluaran Kecepatan Pukulan

Pukulan Ke-	Waktu Tangan (detik)		Kecepatan Tangan (m/s)	
	Kanan	Kiri	Kanan	Kiri
1	0,18	0,22	2,84	2,29
2	0,40	0,17	1,26	3,03
3	0,25	0,23	1,98	2,21
4	0,22	0,25	2,30	2,02
5	0,32	0,23	1,58	2,16
6	0,19	0,20	2,58	2,45
7	0,16	0,19	3,21	2,67
8	0,12	0,14	4,10	3,47
9	0,15	0,17	3,40	2,92
10	0,19	0,20	2,63	2,70

Pengujian terhadap perhitungan kecepatan dilakukan dengan cara melakukan pukulan secara bervariasi untuk melihat apakah *timer* dari *stopwatch* berfungsi dengan semestinya. Seperti yang disajikan dalam gambar 4.9. Tabel pengujian sebagai berikut.

Tabel 4.5 Uji Coba Menghitung Kecepatan Pukulan

Test Case ID	Tujuan	Masukkan	Keluaran yang diharapkan	Status
1.	Mendeteksi variasi pukulan di antara 0,5-0.9 detik.	Gerakan pukulan tangan dari jarak 150 cm sampai 100 cm, waktu dari <i>stopwatch</i>	- <i>Timer</i> sesuai data masukkan. - <i>Timer</i> Direset - Tidak ada delay pembacaan pukulan.	Ada delay apabila jeda antar pukulan terlalu cepat, mengakibatkan data kecepatan tidak sesuai masukan dari <i>stopwatch</i> .
2.	Mendeteksi variasi pukulan di antara 0,1-0.5 detik.	Gerakan pukulan tangan dari jarak 150 cm sampai 100 cm, waktu dari <i>stopwatch</i>	- <i>Timer</i> sesuai data masukkan. - <i>Timer</i> Direset - Tidak ada delay pembacaan pukulan.	Ada delay apabila jeda antar pukulan terlalu cepat, mengakibatkan data kecepatan tidak sesuai masukan dari <i>stopwatch</i> .

#### 4.3.6 Proses Menghitung Kekuatan Pukulan

Proses ini berjalan setelah data dari kecepatan pukulan tersimpan kedalam memori program. Yaitu dengan memasukkan parameter berupa kecepatan dan berat tangan yang sudah ditentukan besarnya lalu dikalkulasikan menggunakan rumus Energi Kinetik. Hasil dari perhitungan tersebut akan disimpan dan digunakan untuk pembuatan grafis performa latihan serta digunakan untuk mencari pukulan dengan tenaga terkuat dan tenaga terlemah dalam satu sesi latihan yang bisa dicapai. Pengujian dilakukan dengan cara mengubah parameter berat tangan dan melakukan pukulan dengan kecepatan yang bervariasi untuk melihat apakah perhitungan berjalan sesuai harapan. Seperti yang disajikan dalam gambar 4.9.

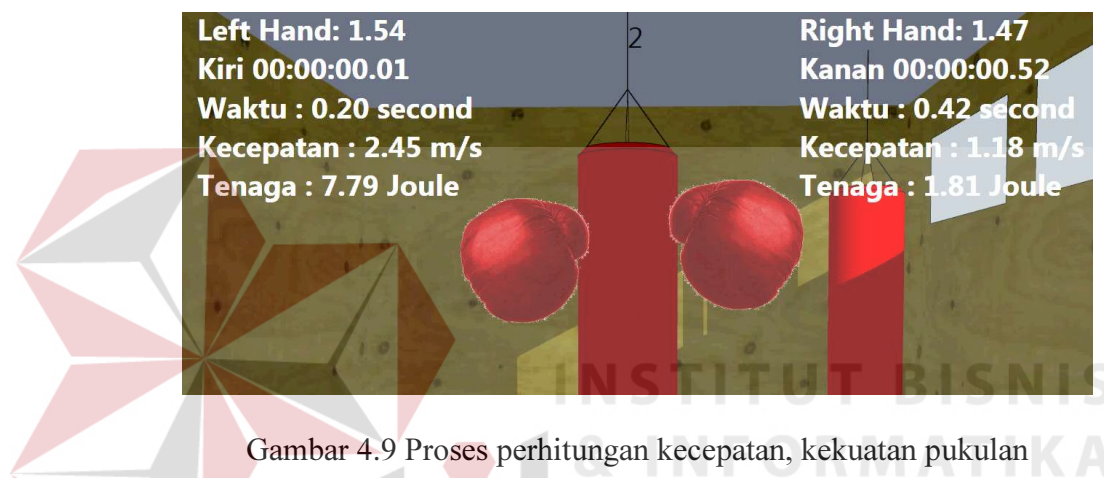
Tabel 4.6 Keluaran Kekuatan Pukulan

Pukulan Ke-	Kecepatan Tangan (m/s)		Kekuatan Pukulan (Joule)	
	Kanan	Kiri	Kanan	Kiri
1	2,84	2,29	10,46	6,82
2	1,26	3,03	2,07	11,91
3	1,98	2,21	5,06	6,35
4	2,30	2,02	6,88	5,31
5	1,58	2,16	3,25	6,07
6	2,58	2,45	8,61	7,79
7	3,21	2,67	13,32	9,27
8	4,10	3,47	21,78	15,63
9	3,40	2,92	15,00	11,09
10	2,63	2,70	8,98	9,47

Tabel 4.7 Uji Coba Menghitung Kekuatan Pukulan

Test Case ID	Tujuan	Masukkan	Keluaran yang diharapkan	Status
1.	Mendeteksi kekuatan pukulan di antara kecepatan 1-3 m/s	Gerakan pukulan tangan dari jarak 150 cm sampai 100 cm dengan kecepatan 1-4 m/s. Berat badan 55 Kg	- Kekuatan dalam Joule ditampilkan.  -Kekuatan harus diantara 1-15 Joule.	Sesuai

			-Tidak ada keluaran berupa 'Infinity'.	
2.	Mendeteksi kekuatan pukulan di antara kecepatan 3-5 m/s atau lebih.	Gerakan pukulan tangan dari jarak 150 cm sampai 100 cm dengan kecepatan 3-5 m/s atau lebih. Berat badan 55 Kg	- Kekuatan dalam Joule ditampilkan. -Kekuatan harus lebih besar dari 10-15 Joule. -Tidak ada keluaran berupa 'Infinity'.	Sesuai



Gambar 4.9 Proses perhitungan kecepatan, kekuatan pukulan

#### 4.3.7 Proses Menampilkan Grafis Latihan

Proses ini dilakukan untuk menguji apakah grafis hasil latihan yang terdapat dalam aplikasi dapat berfungsi sesuai yang diinginkan. Pengujian dilakukan setelah *user* selesai berlatih dan tampilan grafis akan muncul dengan mengambil parameter nilai yang telah diperoleh pada sesi *training*. Pada pengujian ini grafis dapat muncul sesuai yang diharapkan, tetapi terdapat beberapa kekurangan diantaranya apabila pada sesi *training* terdapat kesalahan perhitungan akibat jeda antara Kinect dan aplikasi membuat grafis yang ditampilkan tidak sesuai parameter dan mengacaukan grafik tersebut.



Gambar 4.10 Grafis Latihan

### 4.3.8 Proses Menampilkan Nilai Tertinggi

Proses ini dilakukan untuk menguji apakah fungsi perhitungan nilai tertinggi yang terdapat dalam aplikasi dapat berfungsi sesuai yang diinginkan. Pengujian dilakukan dengan cara memilih menu *High Score* pada aplikasi dan akan disajikan 5 nilai terbaik dari sesi latihan. Pada pengujian ini 5 nilai terbaik dapat ditampilkan sesuai yang diinginkan, akan tetapi apabila terdapat nilai akibat jeda pada sesi *training* dapat mengacaukan hasil 5 nilai terbaik yang telah ada.

**Your Best Result**

No.	NAME	L-HAND		R-HAND	
		MAX	MIN	MAX	MIN
1	wira	146.7434	0.9504488	3241.563	5.064229
2	wira	66.18691	3.469313	135.0089	3.461876
3	wira	42.82683	0.0002027598	47.05418	5.581107
4	wira	20.746	0.1981588	26.78977	1.071591
5	wira	20.418	2.572158	23.2804	0.2442584

Gambar 4.11 Proses menampilkan nilai tertinggi

#### 4.4 Evaluasi

Setelah melakukan uji coba diatas maka berikut adalah hasil evaluasi dari aplikasi *Virtual Punch Training*.

Tabel 4.8 Hasil Evaluasi

Proses	Memenuhi Kriteria	Faktor Hambatan
Mendeteksi gerakan <i>user</i>	Ya	- <i>Delay</i> apabila ruangan banyak objek, atau <i>user</i> terlalu dekat dengan sensor.  -Eror apabila jarak tidak sesuai dengan yang ditentukan yaitu 100 centimeter sampai 150 centimeter dari Kinect
Navigasi Menu	Ya	-
Mengubah <i>Settings</i>	Ya	-
<i>Training Session</i>	Ya	-Tidak bisa melakukan pukulan cepat dan intensif  -Apabila jarak terlalu dekat maka akan terjadi jeda pada Kinect dan aplikasi
Menghitung Kecepatan Pukulan	Ya	-Tidak bisa melakukan pukulan cepat dan intensif  -Apabila jarak terlalu dekat maka akan terjadi jeda pada Kinect dan aplikasi
Menghitung Kekuatan Pukulan	Ya	-Apabila masukan kecepatan terjadi <i>delay</i> maka hasil perhitungan tidak akurat
Menampilkan Grafis	Ya	-Apabila salah satu tangan tidak terdeteksi saat melakukan latihan maka grafis tangan tersebut tidak dapat ditampilkan
Menampilkan Nilai	Ya	-

Berdasarkan pada tabel hasil evaluasi diatas maka dapat disimpulkan sebagai berikut:

1. Aplikasi *Virtual Punch Training* dapat digunakan sebagai alat ukur kekuatan pukulan *virtual* secara *realtime*. Dengan batasan kelemahan Kinect, yaitu tidak dapat mendeteksi gerakan yang intensif dan cepat, tidak bisa mendeteksi dengan baik apabila dibelakang *user* terdapat objek ataupun *user* lain dibelakang *user* utama.
2. Aplikasi *Virtual Punch Training* dapat menampilkan informasi kecepatan gerakan pukulan, waktu tempuhnya dan kekuatan yang dihasilkan dari pukulan tersebut.
3. Jeda dari Kinect mengakibatkan hilangnya ID pada setiap tangan yang telah diberikan sebelumnya oleh aplikasi *Virtual Punch Training* sehingga pencatatan kecepatan, kekuatan dan grafis pukulan tidak sesuai yang diharapkan.
4. Besarnya waktu jeda dari Kinect bervariasi tergantung kecepatan dan intensitas pukulan yang dilakukan, umumnya jeda tersebut terjadi apabila *user* terlalu dekat dengan Kinect, pukulan terlalu intensif atau memukul menggunakan kedua tangan bersamaan.
5. Untuk mengatasi jeda yang diakibatkan oleh Kinect, *user* harus memberi waktu beberapa detik sampai posisi kursor kembali ke posisi awal atau benar-benar melakukan *tracking* terhadap tangan, atau dengan *restart form* pada aplikasi agar Kinect bisa mendeteksi ulang *user* utama, disaat Kinect sedang melakukan pendeteksian diharapkan *user* berdiri tegak posisi siap.
6. Dari hasil evaluasi dan data tabel keluaran bahwa berat badan dan berat tangan *user* sangat berpengaruh pada hasil kecepatan dan kekuatan pukulan.

Semakin berat bobot *user* dan semakin cepat pukulan tersebut maka semakin besar pula kekuatan yang dihasilkan.

7. Apabila terjadi kesalahan perhitungan kecepatan akibat jeda dari Kinect ataupun salah mengisi data berat badan maka akan merusak hasil perhitungan kekuatan pada aplikasi, sehingga nilai yang ditampilkan terkadang tidak sesuai dari yang seharusnya.

8. Aplikasi ini masih memiliki *delay* yang cukup tinggi dalam menangkap respon *user* yang disebabkan oleh Kinect dalam menangkap dan mengenali *joint* dari *user* yang bergerak cepat (Micosoft, 2012).

