

TUGAS AKHIR - KI141502

INTEGRASI EKSTRAKSI FITUR STATIS DAN DINAMIS PADA GERAKAN TANGAN MENGGUNAKAN KINECT 2.0 UNTUK MENGENALI BAHASA ISYARAT INDONESIA

IGNATIUS BENEDICT
NRP 5113100044

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

INTEGRASI EKSTRAKSI FITUR STATIS DAN DINAMIS PADA GERAKAN TANGAN MENGGUNAKAN KINECT 2.0 UNTUK MENGENALI BAHASA ISYARAT INDONESIA

IGNATIUS BENEDICT
NRP 5113100044

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

INTEGRATION EXTRACTION OF STATIC AND DYNAMIC FEATURES ON HAND GESTURES USING KINECT 2.0 TO RECOGNIZE INDONESIAN SIGN LANGUAGE

IGNATIUS BENEDICT
NRP 5113100044

Advisor
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

INFORMATICS DEPARTMENT
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

INTEGRASI EKSTRAKSI FITUR STATIS DAN DINAMIS PADA GERAKAN TANGAN MENGGUNAKAN KINECT 2.0 UNTUK MENGENALI BAHASA ISYARAT INDONESIA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi Grafis dan Seni
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

IGNATIUS BENEDICT

NRP. 5113100044

Disetujui oleh Pembimbing Tugas Akhir

1. Wijayanti Nurul Khotimah, S.Kom, M.Sc.
NIP: 19860312 201212 2 004 (pembimbing 1)
2. Dr.Eng. Nanik Suciati, S.Kom, M.Kom.
NIP: 19710428 199412 2 001 (pembimbing 2)



**SURABAYA
JANUARI, 2018**

[Halaman ini sengaja dikosongkan]

INTEGRASI EKSTRAKSI FITUR STATIS DAN DINAMIS PADA GERAKAN TANGAN MENGGUNAKAN KINECT 2.0 UNTUK MENGENALI BAHASA ISYARAT INDONESIA

Nama Mahasiswa : Ignatius Benedict
NRP : 5113100044
Jurusan : Departemen Informatika FTIK-ITS
**Dosen Pembimbing I : Wijayanti Nurul Khotimah, S.Kom.,
M.Sc.**
Dosen Pembimbing II : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

ABSTRAK

Bahasa isyarat adalah sarana untuk berkomunikasi bagi orang yang menyandang tunarungu. Tidak semua orang dapat menguasai keahlian untuk memahami bahasa isyarat.

Pada Tugas Akhir sebelumnya, telah dikembangkan teknologi untuk mendukung pembelajaran bahasa isyarat Indonesia statis menggunakan Kinect 1.0 dan bahasa isyarat Indonesia dinamis menggunakan Kinect 2.0. Tugas Akhir ini bertujuan untuk membantu orang memahami bahasa isyarat statis dan dinamis dengan cara mengintegrasikan kedua kelompok bahasa isyarat dengan menggunakan teknologi Kinect 2.0.

Hasil pengujian dalam Tugas Akhir ini menunjukkan bahwa metode Decision Tree yang digunakan sebagai classifier gerakan bahasa isyarat mempunyai akurasi yang baik yaitu sekitar 83.67 persen. Hasil tersebut masih dapat ditingkatkan dengan menambahkan data training maupun menggunakan/membuat classifier lain untuk mengklasifikasikan bahasa isyarat.

Kata kunci: Bahasa Isyarat, Kinect 2, Fitur Statis, Fitur Dinamis, Pengenalan Bahasa Isyarat, Sistem Isyarat Bahasa Indonesia.

[Halaman ini sengaja dikosongkan]

INTEGRATION EXTRACTION OF STATIC AND DYNAMIC FEATURES ON HAND GESTURES USING KINECT 2.0 TO RECOGNIZE INDONESIAN SIGN LANGUAGE

Name : Ignatius Benedict
NRP : 5113100044
Major : Informatics Department, FTIf-ITS
Advisor I : Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Advisor II : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

ABSTRACT

Sign language is a way to communicate for deaf people. Not everyone have the skill to understand sign language.

In the previous final project, the use of the technology to learning of static Indonesian sign language using Kinect 1.0 and dynamic Indonesian sign language using Kinect 2.0. This final project goal is to help people understand both static and dynamic sign language by integrating two types of sign language using Kinect 2.0 technology.

The test results of this final project show that the Decision Tree method used as classifier have a good accuracy which is about 83.67 percent. These results can be improved by adding training data or using/creating another classifier to classify sign language.

Keywords: : *Sign Language, Kinect 2, Static Features, Dynamic Features, Sign Language Recognition, Indonesian Sign Language*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa atas berkat dan kasih-Nya, penulis dapat menyelesaikan Tugas Akhir ini sampai selesai. Penulis ingin menyampaikan rasa hormat dan terima kasih yang sebanyak-banyaknya kepada seluruh pihak yang telah membantu penulis dalam penyelesaian Tugas Akhir ini.

1. Bapak Carolus, Ibu Anastasia, dan seluruh keluarga yang selalu memberikan dukungan penuh dalam penyelesaian Tugas Akhir ini.
2. Ibu Wijayanti Nurul Khotimah, S.Kom., M.Sc. dan Ibu Dr.Eng. Nanik Suciati, S.Kom., M.Kom. yang telah bersedia untuk menjadi dosen pembimbing Tugas Akhir sehingga penulis dapat mengerjakan Tugas Akhir dengan arahan dan bimbingan yang baik dan jelas.
3. Bapak dan ibu dosen Departemen Informatika ITS yang banyak memberikan ilmu serta bimbingan selama masa perkuliahan.
4. Teman-teman “DICE” yang selalu memberi semangat dan bantuan pada saat pengerjaan Tugas Akhir ini.
5. Teman-teman Mahasiswa Teknik Informatika 2013 yang telah berjuang bersama-sama pada masa perkuliahan.
6. Teman-teman lain yang ikut serta membantu dalam terlaksananya Tugas Akhir ini.

Semoga Tuhan memberkati seluruh kegiatan dan memberikan balasan yang berlimpah kepada semua yang telah membantu pengerjaan Tugas Akhir ini. Penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan bagi semuanya.

Surabaya, Desember 2017

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi	3
1.7 Sistematika Penulisan.....	4
BAB II DASAR TEORI.....	7
2.1 Tunarungu	7
2.2 Bahasa Isyarat	7
2.3 Kinect 2.0	9
2.4 Kinect SDK	9
2.5 Microsoft Visual Studio	10
2.6 Ekstraksi Fitur	11
2.7 Decision Tree	13
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15
3.1 Analisis Perangkat Lunak.....	15
3.1.1 Deskripsi Umum Perangkat Lunak	16
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak.....	16
3.1.3 Identifikasi Pengguna	17
3.2 Perancangan Perangkat Lunak	18
3.2.1 Model Kasus Penggunaan	18
3.2.2 Definisi Aktor.....	19
3.2.3 Definisi Kasus Penggunaan.....	19

3.2.4 Arsitektur Umum Sistem	23
3.2.5 Rancangan Antarmuka Perangkat Lunak	23
3.2.6 Rancangan Proses Perangkat Lunak.....	24
BAB IV IMPLEMENTASI.....	35
4.1 Lingkungan Pembangunan	35
4.1.1 Lingkungan Pembangunan Perangkat Keras.....	35
4.1.2 Lingkungan Pembangunan Perangkat Lunak	35
4.2 Implementasi Antarmuka	36
4.3 Implementasi Perangkat Lunak	40
4.3.1 Implementasi Pendeteksian <i>Skeleton</i> Pengguna	40
4.3.2 Implementasi Proses Ekstraksi Fitur	43
4.3.4 Implementasi Proses <i>Testing</i> Data.....	50
BAB V PENGUJIAN DAN EVALUASI	59
5.1 Lingkungan Pembangunan	59
5.2 Skenario Pengujian.....	59
5.2.1 Pengujian Skenario 1 dan Analisis	62
5.2.2 Pengujian Skenario 2 dan Analisis	64
5.2.3 Pengujian Skenario 3 dan Analisis	66
5.2.4 Pengujian Skenario 4 dan Analisis	68
5.3 Evaluasi	70
BAB VI KESIMPULAN DAN SARAN.....	75
6.1 Kesimpulan	75
6.2 Saran	76
DAFTAR PUSTAKA.....	77
LAMPIRAN A KODE SUMBER.....	79
LAMPIRAN B <i>SCREENSHOT</i> PERANGKAT LUNAK.....	81
BIODATA PENULIS.....	93

DAFTAR GAMBAR

Gambar 1.1 Alur Aplikasi yang Akan Dibuat	3
Gambar 2.1 Contoh Bahasa Isyarat Statis (1) Gang; (2) Hamba dan Bahasa Isyarat Dinamis (3) Bola; (4) Kijang	8
Gambar 2.2 Kinect 2.0	9
Gambar 2.3 <i>Skeleton Joint</i> yang Dapat Dideteksi Oleh Kinect v2 [11]	11
Gambar 2.4 Pengelompokan Data Kuantisasi	13
Gambar 3.1 Diagram Kasus Penggunaan Perangkat Lunak.....	18
Gambar 3.2 Rancangan Antarmuka Perangkat Lunak	23
Gambar 3.3 Diagram Alir Proses Ekstraksi Fitur.....	24
Gambar 3.4 Posisi Gerakan Tangan pada Tubuh Mengacu pada Leher (Vektor N) dan Tulang Belakang Tengah (Vektor SM) ...	27
Gambar 3.5 Rancangan Proses <i>Training Dataset</i>	29
Gambar 3.6 Decision Tree Statis.....	31
Gambar 3.7 Decision Tree Dinamis	32
Gambar 3.8 Decision Tree Penentu Statis atau Dinamis.....	33
Gambar 3.9 Rancangan Proses <i>Testing Dataset</i>	33
Gambar 4.1 Antarmuka Perangkat Lunak	36
Gambar 5.1 Gambar Ke-20 Bahasa Isyarat yang Digunakan.....	61
Gambar 5.2 Hasil Uji Coba	70
Gambar 5.3 Kesalahan klasifikasi	71
Gambar 5.4 Kemiripan Gerakan Isyarat “Bingung” dan “Topeng”	72
Gambar 5.5 Kemiripan Gerakan Isyarat “Bola”, “Bingkai” dan “Badan”	72
Gambar 5.6 Kemiripan Gerak Isyarat “Ada”, “Botol” dan “Gang”	73
Gambar 5.7 Gerak Bahasa Isyarat “Jendral” dan “Wadah”	73
Gambar B.1 Penulis Melakukan Gerak Bahasa Isyarat Badan ...	81
Gambar B.2 Penulis Melakukan Gerak Bahasa Isyarat Besar.....	81
Gambar B.3 Penulis Melakukan Gerak Bahasa Isyarat Bingkai	82
Gambar B.4 Penulis Melakukan Gerak Bahasa Isyarat Bingung	82
Gambar B.5 Penulis Melakukan Gerak Bahasa Isyarat Bola	83

Gambar B.6 Penulis Melakukan Gerak Bahasa Isyarat Kijang ...	83
Gambar B.7 Penulis Melakukan Gerak Bahasa Isyarat Rujuk	84
Gambar B.8 Penulis Melakukan Gerak Bahasa Isyarat Samping	84
Gambar B.9 Penulis Melakukan Gerak Bahasa Isyarat Sempit ..	85
Gambar B.10 Penulis Melakukan Gerak Bahasa Isyarat Topeng	85
Gambar B.11 Penulis Melakukan Gerak Bahasa Isyarat Ada	86
Gambar B.12 Penulis Melakukan Gerak Bahasa Isyarat Botol ...	86
Gambar B.13 Penulis Melakukan Gerak Bahasa Isyarat Gang ...	87
Gambar B.14 Penulis Melakukan Gerak Bahasa Isyarat Geledag	87
Gambar B.15 Penulis Melakukan Gerak Bahasa Isyarat Hai	88
Gambar B.16 Penulis Melakukan Gerak Bahasa Isyarat Hamba	88
Gambar B.17 Penulis Melakukan Gerak Bahasa Isyarat Hormat	89
Gambar B.18 Penulis Melakukan Gerak Bahasa Isyarat Jendral	89
Gambar B.19 Penulis Melakukan Gerak Bahasa Isyarat Ketua ..	90
Gambar B.20 Penulis Melakukan Gerak Bahasa Isyarat Wadah.	90
Gambar B.21 Penguji 1 Melakukan Gerak Bahasa Isyarat Hai...	91
Gambar B.22 Penguji 2 Melakukan Gerak Bahasa Isyarat Gang	91

DAFTAR TABEL

Tabel 2.1 Fitur yang Digunakan Pada Fitur Statis	12
Tabel 3.1 Definisi Pengguna	19
Tabel 3.2 Definisi Kasus Penggunaan.....	19
Tabel 3.3 Spesifikasi Kasus Penggunaan Membuat Data Bahasa Isyarat Baru	20
Tabel 3.4 Spesifikasi Kasus Penggunaan <i>Testing</i> Data Bahasa Isyarat Baru	21
Tabel 3.5 Data Kuantisasi Hasil Fitur	26
Tabel 3.6 Seluruh Fitur yang Digunakan	28
Tabel 5.1 Skenario Pengujian 1.....	62
Tabel 5.2 Hasil Uji Coba Skenario 1.....	63
Tabel 5.3 Skenario Pengujian 2.....	64
Tabel 5.4 Hasil Uji Coba Skenario 2.....	65
Tabel 5.5 Skenario Pengujian 3.....	66
Tabel 5.6 Hasil Uji Coba Skenario 3.....	67
Tabel 5.7 Skenario Pengujian 4.....	68
Tabel 5.8 Hasil Uji Coba Skenario 4.....	69

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Tampilan Perangkat Lunak	39
Kode Sumber 4.2 Kode Sumber Integrasi Kinect	40
Kode Sumber 4.3 Kode Sumber Deteksi <i>Skeleton</i> Pengguna	43
Kode Sumber 4.4 Kode Sumber Ekstraksi Fitur Statis, Dinamis dan Menentukan Posisi Gerakan Tangan dari <i>Skeleton</i> Pengguna	48
Kode Sumber 4.5 Kode Sumber Menyimpan Hasil Ekstraksi Fitur Statis, Dinamis dan Posisi Gerak Tangan dari <i>Skeleton</i> Pengguna	50
Kode Sumber 4.6 Implementasi Decision Tree dari Hasil Klasifikasi.....	58
Kode Sumber A.1 Fungsi Create File	79
Kode Sumber A.2 Fungsi Start Testing.....	79
Kode Sumber A.3 Fungsi Stop.....	79

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan, batasan permasalahan, dan manfaat.

1.1 Latar Belakang

Para penyandang tunarungu memiliki keterbatasan dalam berkomunikasi menggunakan bahasa sehari-hari. Bahasa isyarat merupakan bahasa yang digunakan oleh penyandang tunarungu. Bahasa ini menggunakan gerak visual tubuh untuk menyampaikan maksud kepada lawan bicaranya. Sistem yang umum digunakan di Indonesia mengacu pada Sistem Isyarat Bahasa Indonesia (SIBI) .

Bahasa isyarat bagi penyandang tunarungu merupakan bahasa umum, tetapi bagi orang normal merupakan bahasa asing [1]. Hal ini dapat mengakibatkan kesenjangan komunikasi antara orang normal dengan penyandang tunarungu. Untuk mempermudah proses komunikasi, dibutuhkan penerjemah antara penyandang tunarungu dengan orang normal.

Sebelumnya sudah ada Tugas Akhir yang dibuat oleh Yohanes Aditya Sutanto dan Yahya Eka Nugyasa tentang pengenalan bahasa isyarat Indonesia menggunakan teknologi Kinect. Dalam Tugas Akhir yang dibangun oleh Yohanes menggunakan teknologi Kinect 1.0, sudah dapat mendeteksi bahasa isyarat statis [2]. Dalam Tugas Akhir yang dibangun oleh Yahya menggunakan teknologi Kinect 2.0, sudah dapat mendeteksi bahasa isyarat dinamis [3]. Oleh karena itu munculah ide untuk mengintegrasikan pendeteksi bahasa isyarat statis dan bahasa isyarat dinamis. Bahasa isyarat yang digunakan mengacu pada SIBI.

1.2 Rumusan Masalah

Rumusan masalah yang terdapat pada Tugas Akhir ini diantaranya:

1. Bagaimana mengintegrasikan fitur statis dan dinamis pada gerakan tangan dalam mendeteksi bahasa isyarat?
2. Bagaimana cara mendapatkan hasil ekstraksi fitur statis dan dinamis yang optimal?
3. Bagaimana hasil integrasi fitur statis dan dinamis untuk mengenali gerakan yang telah ditentukan oleh pengguna ?

1.3 Batasan Masalah

Batasan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut:

1. Teknologi yang dipakai adalah Kinect 2.0.
2. Bahasa isyarat statis yang dideteksi berjumlah 10 gerakan.
3. Bahasa isyarat dinamis yang dideteksi berjumlah 10 gerakan.
4. Klasifikasi yang digunakan untuk mengklasifikasi gerakan adalah Decision Tree.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah mengintegrasikan hasil ekstraksi fitur statis dan dinamis pada gerakan tangan menggunakan Kinect 2.0 untuk mengenali bahasa isyarat Indonesia.

1.5 Manfaat

Tugas Akhir ini diharapkan dapat mengenali bahasa isyarat statis dan bahasa isyarat dinamis sehingga dapat membantu orang tunarungu berkomunikasi dengan orang normal.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Pada tahap ini, dicari studi literatur yang relevan untuk dijadikan referensi dalam pengerjaan Tugas Akhir. Studi literatur dapat diambil dari buku, internet, maupun materi dalam suatu mata kuliah yang berhubungan dengan metode yang akan digunakan. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Tunarungu
2. Bahasa isyarat
3. Kinect
4. Kinect SDK
5. Decision Tree

B. Perancangan perangkat lunak



Gambar 1.1 Alur Aplikasi yang Akan Dibuat [4] [5] [6]

Analisa dimulai dari pengguna yang memberikan masukan berupa gerakan tangan dan posisi gerakan tersebut dari badan ke alat Kinect 2.0. Kemudian masukan tersebut diproses dengan *classifier* yang telah ditentukan yaitu Decision Tree. Dan pada akhirnya, keluaran dari proses tersebut ditampilkan menggunakan aplikasi perangkat lunak berbasis desktop. Seperti yang dapat dilihat pada Gambar 1.1.

- C. Implementasi dan pembuatan sistem
Pembangunan aplikasi dilakukan dengan menggunakan bahasa pemrograman C#, IDE (*Integrated Development Environment*) Microsoft Visual Studio dan Kinect for Windows SDK 2.0. Tahap awal dari pembangunan perangkat lunak ini ada integrasi aplikasi dengan perangkat Kinect. Berikutnya dilakukan implementasi ekstraksi fitur statis dan dinamis menggunakan Kinect 2.0. Dari data yang didapatkan, dilakukan proses *training* dan *testing* data untuk mendapatkan model. Aplikasi ini dibangun menggunakan Microsoft Visual Studio 2015.
- D. Uji coba dan evaluasi
Pengujian akan dilakukan oleh dua orang pengguna. Pengguna tersebut akan diminta untuk melakukan gerakan yang telah ditentukan kemudian dihitung akurasi gerakan tersebut dari aplikasi yang telah dibuat.
- E. Penyusunan laporan Tugas Akhir
Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan masalah, tujuan dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan Tugas Akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi antarmuka aplikasi dan pembuatan kebutuhan fungsional aplikasi

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan Tugas Akhir ini. Beberapa hal yang dibahas adalah penjelasan tentang tunarungu, bahasa isyarat secara umum, Kinect 2.0 dan SDK-nya, serta decision tree.

2.1 Tunarungu

Tunarungu dapat diartikan sebagai keterbatasan yang dimiliki seseorang dalam mendengar sesuatu karena tidak berfungsinya organ pendengaran yang dimilikinya. Ketunarunguan dapat dibedakan menjadi dua kategori yaitu tuli (*deaf*) dan kurang dapat mendengar (*low hearing*) [7]. Tuli adalah keadaan dimana organ pendengaran telah mengalami kerusakan yang sangat parah dan mengakibatkan tidak berfungsinya pendengaran. Sedangkan kurang dapat mendengar adalah keadaan dimana organ pendengaran mengalami kerusakan tetapi masih dapat berfungsi untuk mendengar.

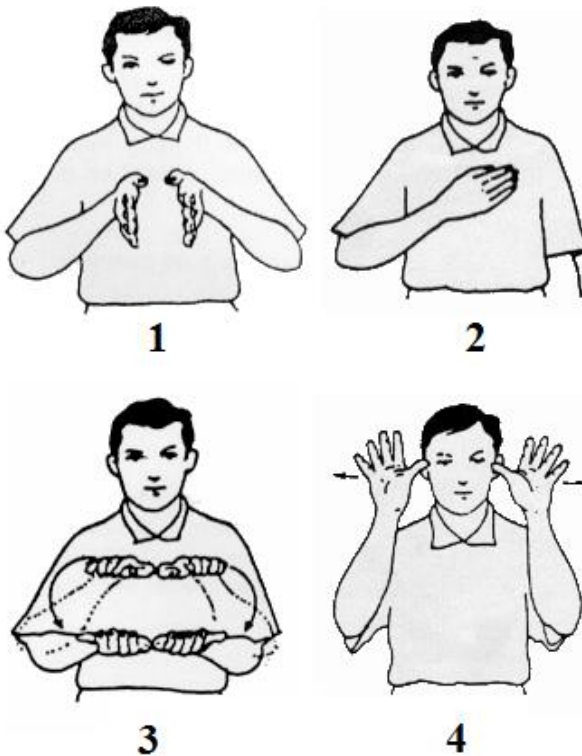
2.2 Bahasa Isyarat

Bahasa isyarat adalah bahasa yang digunakan untuk berkomunikasi oleh penyandang tunarungu. Bahasa isyarat yang sering digunakan di Indonesia berdasarkan pada Sistem Isyarat Bahasa Indonesia (SIBI). Ada 4 jenis bahasa isyarat dalam SIBI [8], yaitu:

1. Isyarat Pokok: melambangkan sebuah kata atau konsep;
2. Isyarat Tambahan: melambangkan awalan, akhiran, dan partikel (imbuhan);
3. Isyarat Bentuk: dibentuk dengan menggabungkan isyarat pokok dan isyarat tambahan;

4. Abjad Jari: dibentuk dengan jari-jari untuk mengeja huruf / angka.

Berdasarkan sifat gerakannya, bahasa isyarat dibagi menjadi 2 kelompok yaitu gerak bahasa isyarat statis dan gerak bahasa isyarat dinamis. Contoh gerak bahasa isyarat statis dan dinamis dapat dilihat pada Gambar 2.1.



Gambar 2.1 Contoh Bahasa Isyarat Statis (1) Gang; (2) Hamba dan Bahasa Isyarat Dinamis (3) Bola; (4) Kijang

2.3 Kinect 2.0

Generasi kedua dari Kinect yang dirilis oleh Microsoft pada tahun 2014 adalah versi terbaru Kinect dari yang pertama kali dikeluarkan pada tahun 2010. Perangkat Kinect 2.0 seperti yang terlihat pada Gambar 2.2, terdapat tiga lensa yaitu kamera RGB yang digunakan untuk menangkap spektrum warna, *infrared emitters* yang memproyeksikan spektrum inframerah dan sensor kedalaman yang menghasilkan gambar mendalam dari seseorang atau objek dengan menganalisis informasi inframerah. Dan sebuah *microphone* array yang dapat menemukan lokasi timbulnya suara. Alhasil, ada enam sumber data yang dihasilkan, termasuk warna, inframerah, kedalaman, indeks tubuh, tubuh, dan suara [9].



Gambar 2.2 Kinect 2.0

2.4 Kinect SDK

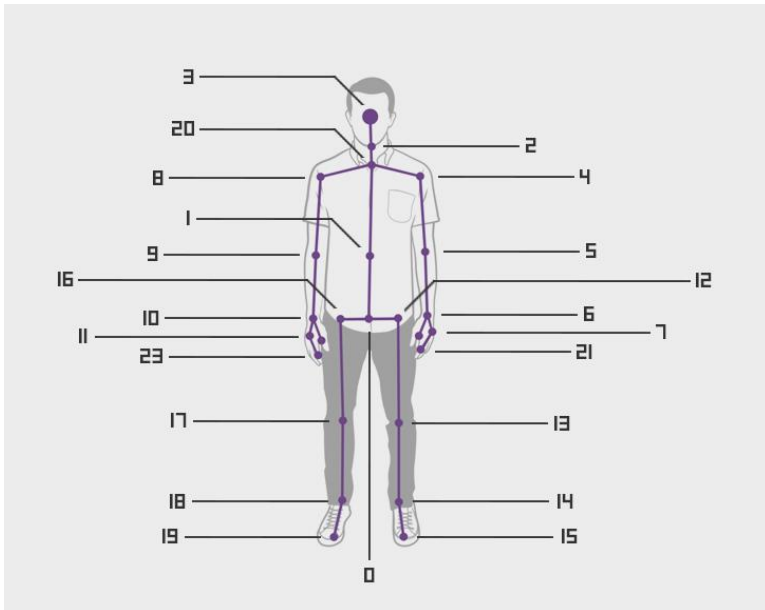
Kinect SDK adalah pustaka yang dibuat oleh Microsoft untuk pengembangan aplikasi perangkat lunak yang menggunakan Kinect sebagai alat input utama. Kinect SDK dapat diimplementasikan dengan bahasa pemrograman C#, C++, dan JavaScript. Pustaka ini memiliki beberapa fitur diantaranya *skeleton tracking*, *thumb tracking*, *end of hand tracking*, *open/close hand gesture* dan lainnya [5]. Kinect v2.0 ini dapat mendeteksi 25 *skeleton joint* yang dapat dilihat pada Gambar 2.3.

Berikut adalah ke 25 *skeleton joint* yang dapat dideteksi :

0. Dasar Tulang Belakang (SB)
1. Tulang Belakang Tengah (SM)
2. Leher (N)
3. Kepala (H)
4. Bahu Tangan Kiri (SL)
5. Siku Tangan Kiri (EL)
6. Pergelangan Tangan Kiri (WL)
7. Telapak Tangan Kiri (HL)
8. Bahu Tangan Kanan (SR)
9. Siku Tangan Kanan (ER)
10. Pergelangan Tangan Kanan (WR)
11. Telapak Tangan Kanan (HR)
12. Pinggul Kiri (HiL)
13. Lutut Kaki Kiri (KL)
14. Pergelangan Kaki Kiri (AL)
15. Kaki Kiri (FL)
16. Pinggul Kanan (HiR)
17. Lutut Kaki Kanan (KR)
18. Pergelangan Kaki Kanan (AR)
19. Kaki Kanan (FR)
20. Bahu Tulang Belakang (SS)
21. Jari Tangan Kiri (HTL)
22. Jempol Tangan Kiri (TL)
23. Jari Tangan Kanan (HTR)
24. Jempol Tangan Kanan (TR)

2.5 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan perangkat lunak, baik itu perangkat lunak bisnis, perangkat lunak pribadi, ataupun komponen perangkat lunak nya dalam bentuk perangkat lunak berbasis *console*, Windows, ataupun berbasis *website* [10].



Gambar 2.3 *Skeleton Joint* yang Dapat Dideteksi Oleh Kinect v2 [11]

2.6 Ekstraksi Fitur

Fitur yang digunakan dalam Tugas Akhir ini terdiri dari dua kelompok, yaitu fitur statis dan fitur dinamis. Fitur statis akan digunakan untuk mengklasifikasikan gerakan bahasa isyarat statis. Fitur dinamis akan digunakan untuk mengklasifikasikan gerak bahasa isyarat dinamis.

Skeleton joint yang akan digunakan untuk ekstraksi fitur terdapat sebanyak 11 *skeleton joints*. Sebelas *skeleton joints* tersebut adalah sebagai berikut:

1. Leher (N)
2. Bahu tulang belakang (SS)
3. Tulang belakang tengah (SM)
4. Bahu tangan kanan (SR)
5. Siku tangan kanan (ER)

6. Pergelangan tangan kanan (WR)
7. Telapak tangan kanan (HR)
8. Bahu tangan kiri (SL)
9. Siku tangan kiri (EL)
10. Pergelangan tangan kiri (WL)
11. Telapak tangan kiri (HL)

Fitur statis adalah fitur yang bersifat statis yang didapat dari hasil kalkulasi beberapa *skeleton joint* yang dideteksi oleh Kinect 2.0. Fitur yang digunakan terdiri dari 3 bagian yaitu *vector2*, *angel*, *distance*. Dari hasil perhitungan *skeleton joint*, didapatkan 21 buah fitur yang akan digunakan [12]. Fitur-fitur yang digunakan untuk fitur statis dapat di lihat pada Tabel 2.1.

Tabel 2.1 Fitur yang Digunakan Pada Fitur Statis

Vektor2 (x,y)	Sudut (float)	Jarak (float)
SR -> ER	$\angle$ SS – SR – ER	HR – HL
ER -> WR	$\angle$ SR – ER – WR	
WR -> HR	$\angle$ ER – WR – HR	
SL -> EL	$\angle$ SS – SL – EL	
EL -> WL	$\angle$ SL – EL – WL	
WL -> HL	$\angle$ EL – WL – HL	
HR -> HL		

Fitur dinamis merupakan fitur gerak atau fitur yang bersifat dinamis yang didapatkan dari deteksi Kinect 2.0 terhadap tangan. Fitur dinamis yang digunakan pada metode ini adalah fitur untuk mengolah gerakan tangan (*hand gesture*).

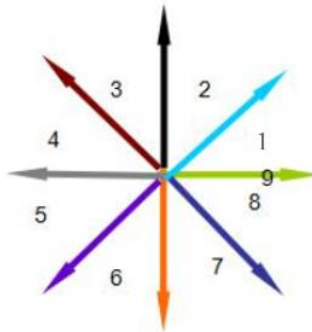
Terdapat 40 *frame* yang akan diproses. Setiap *frame* akan menghasilkan koordinat gerak dinamis yang diproyeksikan ke dalam bidang X0Y yang merupakan prinsip dalam bidang. Kemudian dari hasil setiap data tersebut, akan dicatat orientasi sudut mutlak $\alpha_t \in (0, 360^\circ)$ yang dapat dilihat pada persamaan (2.1) (2.2) dan (2.3) [9]. Orientasi sudut mutlak tersebut

kemudian diterjemahkan menjadi data kuantisasi hasil ekstraksi fitur dinamis seperti yang dapat dilihat pada Gambar 2.4.

$$\Delta y = y_t - y_{t-1} \quad (2.1)$$

$$\Delta x = x_t - x_{t-1} \quad (2.2)$$

$$\alpha_t = \begin{cases} \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right) + 180, & \Delta x < 0 \\ \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right) + 360, & \Delta y < 0 \\ \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right), & \Delta x > 0, \Delta y \geq 0 \end{cases} \quad (2.3)$$



Gambar 2.4 Pengelompokan Data Kuantisasi

2.7 Decision Tree

Pohon Keputusan atau dikenal dengan Decision Tree adalah salah satu metode klasifikasi yang menggunakan representasi suatu struktur pohon yang berisi alternatif-alternatif untuk pemecahan suatu masalah. Pohon ini juga menunjukkan faktor-faktor yang mempengaruhi hasil alternatif dari keputusan tersebut disertai dengan estimasi hasil akhir bila kita mengambil

keputusan tersebut. Peranan pohon keputusan ini adalah sebagai *Decision Support Tool* untuk membantu manusia dalam mengambil suatu keputusan. Manfaat dari Decision Tree adalah melakukan *break down* proses pengambilan keputusan yang kompleks menjadi lebih mudah sehingga orang yang mengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan. Konsep yang digunakan oleh Decision Tree adalah mengubah data menjadi suatu pohon keputusan dan aturan-aturan keputusan (*rules*).

Decision Tree menggunakan struktur hierarki untuk pembelajaran *supervised*. Proses dari Decision Tree dimulai dari *root node* hingga *leaf node* yang dilakukan secara rekursif. Dimana setiap percabangan menyatakan suatu kondisi yang harus dipenuhi dan pada setiap ujung pohon menyatakan kelas dari suatu data. Pada Decision Tree terdiri dari tiga bagian yaitu:

a. *Root Node*

Node ini merupakan *node* yang terletak paling atas dari suatu pohon.

b. *Internal Node*

Node ini merupakan *node* percabangan, hanya terdapat satu *input* serta mempunyai minimal dua *output*.

c. *Leaf Node*

Node ini merupakan *node* akhir, hanya memiliki satu *input*, dan tidak mempunyai *output*.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan Tugas Akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan oleh sistem. Selanjutnya dibahas mengenai perancangan sistem yang dibuat.

3.1 Analisis Perangkat Lunak

Bahasa isyarat merupakan alat komunikasi utama bagi penyandang tunarungu. Bahasa itu tidak mudah dipahami bagi orang pada umumnya, tetapi dapat sangat membantu penyandang tunarungu untuk berkomunikasi dengan sekitarnya. Namun masih banyak yang belum mengerti apa arti isyarat bagi yang diberi atau memberi isyarat. Perangkat lunak ini bertujuan untuk membantu pengguna mengartikan bahasa isyarat yang dilakukan. Perangkat lunak ini dibangun dengan Microsoft Visual Studio dan dengan bantuan Kinect 2.0 untuk mengekstraksi *skeleton joint*.

Proses *training* data dilakukan oleh pengguna untuk menambahkan bahasa isyarat yang diinginkan. Setelah proses pengambilan data *training* selesai, fitur-fitur didapatkan dari hasil kalkulasi *skeleton joint* yang disimpan dalam bentuk .csv untuk dikalsifikasikan menggunakan *classifier* WEKA. Hasil dari klasifikasi tersebut berupa model Decision Tree.

Untuk tahap *testing* data, pengguna melakukan pengujian dengan cara melakukan gerakan bahasa isyarat yang diinginkan. Kemudian perangkat lunak mengambil 40 *frame* hasil gerak pengguna. Dari 40 *frame* tersebut di hasilkan fitur-fitur yang akan diklasifikasikan kedalam model Decision Tree yang sudah didapatkan sebelumnya. Kemuadian perangkat lunak akan menampilkan hasil dari gerakan bahasa isyarat yang dilakukan oleh pengguna dalam bentuk tulisan dan gambar.

3.1.1 Deskripsi Umum Perangkat Lunak

Tugas Akhir yang dibangun ini adalah sebuah modul pengenalan bahasa isyarat Indonesia dengan menggunakan teknologi Kinect 2.0. Gerakan isyarat yang dideteksi dapat berupa gerakan isyarat statis dan gerakan isyarat dinamis.

Pengguna utama adalah semua orang yang ingin mempelajari gerakan isyarat yang mengacu pada SIBI. Pengguna dapat mempelajari isyarat yang sudah tersedia didalam perangkat lunak atau dapat memberikan isyarat baru yang mengacu pada SIBI.

3.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang akan dibuat ini melibatkan dua hal, yakni kebutuhan fungsional maupun kebutuhan non-fungsional. Dimana masing-masing berhubungan dengan keberhasilan dalam pembuatan Tugas Akhir ini.

3.1.2.1 Kebutuhan Fungsional Perangkat Lunak

Pada perangkat lunak ini, terdapat beberapa kebutuhan fungsional yang dapat mendukung jalannya perangkat lunak ini. Berikut adalah beberapa kebutuhan fungsional :

- a) Mendeteksi *skeleton* pengguna
Perangkat lunak dapat mendeteksi pengguna yang sedang berada di depan Kinect 2.0.
- b) Mengekstraksi fitur statis dan dinamis *skeleton*
Perangkat lunak dapat mendeteksi posisi dari *skeleton* yang akan diekstraksi menjadi fitur-fitur yang dibutuhkan. Fitur-fitur ini yang akan diklasifikasikan melalui proses *training* dan *testing*.

- c) Menerjemahkan Bahasa Isyarat
Perangkat lunak dapat menerjemahkan bahasa isyarat Indonesia yang dihasilkan melalui proses ekstraksi fitur statis dan dinamis *skeleton* pengguna.

3.1.2.2 Kebutuhan Non-Fungsional

Tidak hanya kebutuhan fungsional saja, terdapat juga beberapa kebutuhan non-fungsional dalam mendukung dan menambah performa jalannya perangkat lunak. Kebutuhan non-fungsional tersebut adalah sebagai berikut:

- a) Penyesuaian intensitas cahaya
Intensitas cahaya merupakan salah satu unsur yang perlu diperhatikan dalam penggunaan sensor Kinect 2.0. Jika intensitas cahaya pada saat pengambilan data kurang, maka dapat mengganggu stabilitas pengambilan data yang dapat berakibat pada hasil akhir yang ditampilkan oleh perangkat lunak. Maka dari itu, penggunaan perangkat lunak ini sebaiknya berada pada ruangan yang memiliki intensitas cahaya yang cukup.
- b) Posisi Kinect dengan pengguna
Posisi peletakan Kinect 2.0 disesuaikan dengan pengguna untuk mendapatkan fitur-fitur yang maksimal, baik saat pengambilan data *training* maupun *testing*. Jarak optimal Kinect 2.0 dengan pengguna adalah antara 0,6 sampai 1,8 meter.

3.1.3 Identifikasi Pengguna

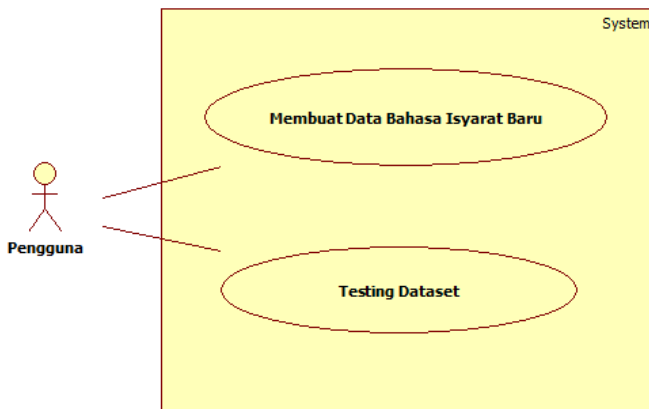
Dalam Tugas Akhir yang dibangun ini, pengguna yang akan terlibat dalam menjalankan perangkat lunak hanya satu orang saja, yaitu orang yang akan melakukan pengenalan bahasa isyarat Indonesia.

3.2 Perancangan Perangkat Lunak

Subbab ini membahas bagaimana rancangan dari Tugas Akhir ini. Hal yang dibahas meliputi model kasus penggunaan, definisi aktor, definisi kasus penggunaan, arsitektur umum sistem, rancangan antarmuka perangkat lunak, dan rancangan proses perangkat lunak.

3.2.1 Model Kasus Penggunaan

Dari hasil analisa deskripsi umum perangkat lunak dan spesifikasi kebutuhan perangkat lunak yang telah dijelaskan, maka model kasus penggunaan untuk perangkat lunak pengenalan bahasa isyarat dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Kasus Penggunaan Perangkat Lunak

3.2.2 Definisi Aktor

Aktor yang terdapat dalam sistem aplikasi ini terlihat pada Tabel 3.1 .

Tabel 3.1 Definisi Pengguna

No	Nama	Deskripsi
1	Pengguna	Merupakan aktor yang bertugas untuk menambahkan data <i>training</i> dan melakukan <i>testing</i> gerakan isyarat statis dan dinamis, seluruh fungsionalitas yang ada di dalam sistem dapat digunakan oleh pengguna.

3.2.3 Definisi Kasus Penggunaan

Pada Tabel 3.1 telah dijelaskan bahwa aktor yang dalam hal ini disebut pengguna mempunyai dua kasus penggunaan, yakni membuat data bahasa isyarat baru dan melakukan *testing dataset*. Rincian mengenai kasus penggunaan tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Definisi Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-01	Membuat Data Bahasa Isyarat Baru	Pengguna membuat data bahasa isyarat yang baru
2	UC-02	<i>Testing Dataset</i>	Pengguna melakukan <i>testing dataset</i> dengan melakukan gerakan bahasa isyarat Indonesia yang tersedia.

3.2.3.1 Kasus Penggunaan Membuat Data Bahasa Isyarat Baru

Spesifikasi kasus penggunaan membuat data bahasa isyarat baru dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi Kasus Penggunaan Membuat Data Bahasa Isyarat Baru

Nama Kasus Penggunaan	Membuat Data Bahasa Isyarat Baru
Nomor	UC-01
Deskripsi	Kasus penggunaan aktor untuk membuat data bahasa isyarat baru
Aktor	Pengguna
Kondisi Awal	Pengguna sudah menjalankan perangkat lunak dan perangkat Kinect 2.0 telah tersambung
Alur Normal	1. Pengguna memasukkan nama bahasa isyarat yang akan dibuat di dalam <i>textbox</i> perangkat lunak 2. Pengguna menekan tombol “Create File” di dalam perangkat lunak 3. Perangkat lunak menerima inputan dari Kinect 2.0 dan ketika <i>skeleton</i> pengguna ditemukan, perangkat lunak memberikan waktu 5 detik untuk pengguna mempersiapkan gerakan A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna 4. Pengguna melakukan gerakan bahasa isyarat yang akan dibuat 5. Perangkat lunak mengekstrak

	data <i>skeleton</i> pengguna sebanyak 40 <i>frame</i> untuk dikalkulasi 6. Perangkat lunak menyimpan hasil ekstraksi ke dalam sebuah berkas berekstensi .csv
Alur Alternatif	A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna 1. Sistem memberikan notifikasi bahwa <i>skeleton</i> pengguna tidak ditemukan
Kondisi Akhir	Perangkat lunak membuat dataset bahasa isyarat yang baru

3.2.3.2 Kasus Penggunaan *Testing Data*

Spesifikasi kasus penggunaan *Testing Data* dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Penggunaan *Testing Data* Bahasa Isyarat Baru

Nama Kasus Penggunaan	<i>Testing Dataset</i>
Nomor	UC-02
Deskripsi	Kasus penggunaan aktor untuk melakukan <i>testing dataset</i> isyarat bahasa Indonesia
Aktor	Pengguna
Kondisi Awal	Pengguna dalam keadaan menjalankan perangkat lunak dan sudah ada model hasil <i>classifier</i> data isyarat bahasa Indonesia di dalam perangkat lunak
Alur Normal	1. Pengguna menekan tombol “Start Testing” di dalam

	perangkat lunak 2. Perangkat lunak menerima inputan dari Kinect 2.0 dan ketika <i>skeleton</i> pengguna ditemukan, perangkat lunak memberikan waktu 3 detik untuk pengguna mempersiapkan gerakan - A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna 3. Pengguna melakukan gerakan bahasa isyarat 4. Perangkat lunak mengekstrak data <i>skeleton</i> pengguna sebanyak 40 <i>frame</i> untuk dikalkulasi 5. Hasil ekstraksi data diklasifikasi menggunakan Decision Tree yang sudah dibuat di dalam perangkat lunak 6. Perangkat lunak menampilkan isyarat bahasa Indonesia hasil klasifikasi
Alur Alternatif	A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna 1. Sistem memberikan notifikasi bahwa <i>skeleton</i> pengguna tidak ditemukan
Kondisi Akhir	Perangkat lunak memberikan keluaran berupa bahasa isyarat yang dimaksud oleh pengguna

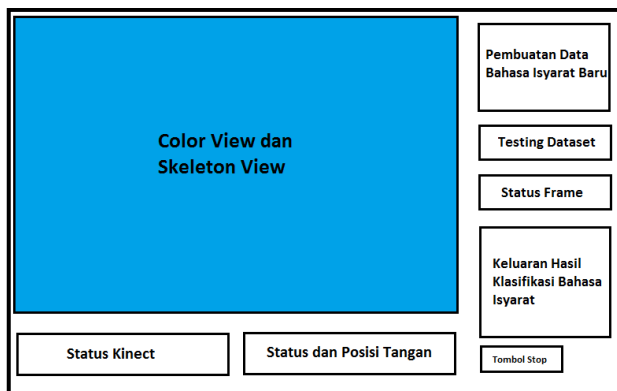
3.2.4 Arsitektur Umum Sistem

Arsitektur umum pada perangkat lunak ini memiliki perangkat tambahan Kinect 2.0 sebagai perangkat masukan. Implementasi aplikasi dibuat menggunakan Microsoft Visual Studio. Arsitektur umum perangkat lunak yang akan dibuat dapat dilihat pada Gambar 1.1.

3.2.5 Rancangan Antarmuka Perangkat Lunak

Rancangan antarmuka perangkat lunak diperlukan untuk memberikan gambaran umum kepada pengguna bagaimana sistem yang ada dalam perangkat lunak ini berinteraksi dengan pengguna. Selain itu, rancangan ini juga memberikan gambaran bagi pengguna tentang tampilan yang sudah disediakan didalam perangkat lunak, sehingga akan muncul kesan pengalaman pengguna yang baik dan mudah.

Rancangan antarmuka perangkat lunak ini hanya memiliki satu Windows dengan beberapa bagian seperti *color view*, *skeleton view* dan memiliki beberapa kontrol yang sekiranya dapat dipahami oleh pengguna. Rancangan antarmuka perangkat lunak dapat dilihat pada Gambar 3.2.



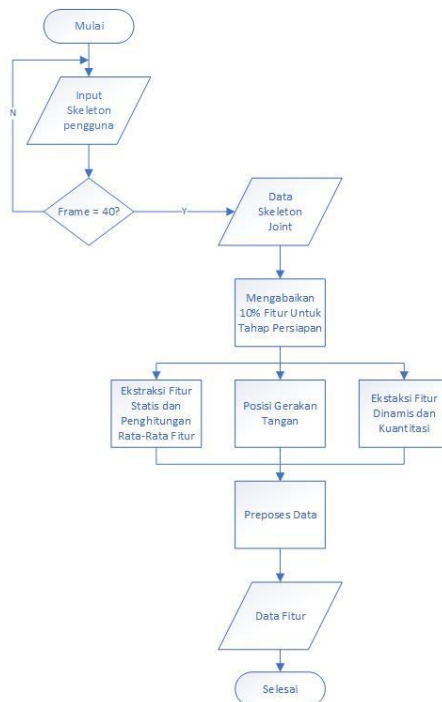
Gambar 3.2 Rancangan Antarmuka Perangkat Lunak

3.2.6 Rancangan Proses Perangkat Lunak

Pada rancangan proses perangkat lunak akan dijelaskan mengenai proses yang terjadi dalam sistem untuk memenuhi fungsionalitas yang ada pada perangkat lunak. Proses ini penting agar perangkat lunak dapat berjalan secara baik dan benar.

3.2.6.1 Rancangan Proses Ekstraksi Fitur

Proses ekstraksi fitur ini sangat dibutuhkan bagi pengguna untuk melakukan *training* dan *testing*. Rancangan proses pengambilan fitur dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Alir Proses Ekstraksi Fitur

1. *Input skeleton joint* pengguna oleh Kinect 2.0

Saat perekaman *skeleton joint* oleh Kinect 2.0 yang kemudian diolah menjadi fitur data, perangkat lunak akan memberikan waktu terlebih dahulu kepada pengguna selama 5 detik untuk mempersiapkan gerakan. Setelah itu, perangkat lunak mengambil total 40 *frame* untuk dikalkulasi dan dijadikan fitur.

Jumlah data tersebut ditentukan berdasarkan durasi gerakan bahasa isyarat yang paling lama dilakukan diantara seluruh *sample* gerakan yang sudah ditentukan sebelumnya. Apabila gerakan pengguna terlalu cepat atau terlalu lambat, maka pengambilan data harus diulang kembali guna mendapatkan hasil yang maksimal.

2. Ekstraksi fitur statis

Dalam proses ekstraksi fitur statis, didapatkan fitur yang berjumlah 21 buah [12]. Dua puluh satu fitur yang dimaksud dapat dilihat pada Tabel 2.1. Fitur yang digunakan berupa vektor2, sudut, dan jarak antara kedua telapak tangan.

3. Ekstraksi fitur dinamis dan kuantisasi

Dalam proses ekstraksi fitur dinamis, berdasarkan persamaan (2.1) dan (2.2), perangkat lunak terlebih dahulu menentukan Δx dan Δy yang didapatkan dari hasil selisih masing-masing koordinat X dan Y antar data ke- n dan $n+1$ *skeleton joints* HR dan HL. Nilai Δx dan Δy kemudian digunakan dalam menentukan orientasi sudut mutlak (α_t) berdasarkan persamaan (2.3) yang dapat dijelaskan sebagai berikut:

1. Jika Δx dan Δy lebih besar atau sama dengan daripada nol, maka nilai α_t adalah arctan hasil pembagian Δy dan Δx kemudian dikalikan dengan hasil pembagian 180 dengan PI (π).
2. Jika Δx lebih kecil daripada nol, maka nilai α_t adalah arctan hasil pembagian Δy dan Δx dikalikan dengan

hasil pembagian 180 dengan π kemudian ditambahkan 180.

3. Jika Δy lebih kecil daripada nol, maka nilai α_t adalah arctan hasil pembagian Δy dan Δx dikalikan dengan hasil pembagian 180 dengan π kemudian ditambah 360.

Hasil dari α_t tersebut kemudian diterjemahkan menjadi data kuantisasi hasil ekstraksi fitur dinamis (berdasarkan Gambar 2.4) yang dapat dijelaskan pada Tabel 3.5:

Tabel 3.5 Data Kuantisasi Hasil Fitur

Rule No.	α_t	Nilai Fitur
1	$\alpha_t = 0$	9
2	$\alpha_t > 314$	8
3	$\alpha_t > 269$	7
4	$\alpha_t > 224$	6
5	$\alpha_t > 179$	5
6	$\alpha_t > 134$	4
7	$\alpha_t > 89$	3
8	$\alpha_t > 44$	2
9	$\alpha_t > 0$	1

Data kuantisasi hasil ekstraksi fitur dinamis gerakan tangan yang didapatkan kemudian dijadikan fitur data. Terdapat 80 fitur dinamis yang diekstraksi. Fitur ke 1 s.d. 40 menginterpretasikan data kuantisasi 40 *frame* yang dihasilkan oleh tangan kiri. Sedangkan fitur ke 41 s.d. 80 menginterpretasikan data kuantisasi 40 *frame* yang dihasilkan oleh tangan kanan.

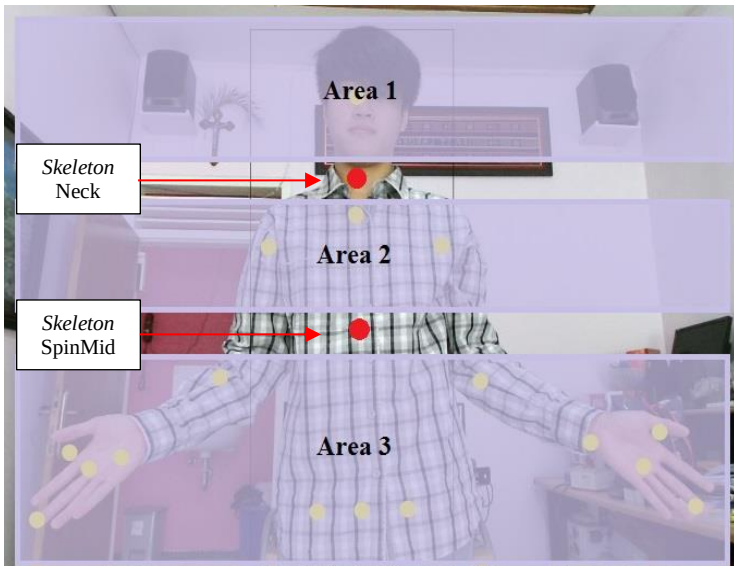
Setelah dikurangi dari 10% data yang diabaikan dalam tahap persiapan, perangkat lunak mengambil hanya data genap saja untuk mendapatkan data yang lebih akurat [9]. Maka didapatkanlah 18 fitur dinamis untuk tangan kanan dan 18 fitur dinamis untuk tangan kiri.

4. Posisi gerakan tangan

Pada *frame* ke-20, perangkat lunak mengidentifikasi posisi gerakan tangan yang dilakukan oleh pengguna. Seperti yang dapat

dilihat pada Gambar 3.4, posisi gerakan tangan dibagi menjadi tiga daerah, yaitu Area 1 (kepala), Area 2 (dada), dan Area 3 (perut). Daerah posisi gerakan tangan tersebut ditentukan oleh *skeleton joints* N dan SM yang dapat dijelaskan sebagai berikut:

1. Jika koordinat *skeleton joints* HR dan atau HL berada diatas (lebih besar daripada) koordinat *skeleton joints* N, maka posisi diidentifikasi berada pada Area 1.
2. Jika koordinat *skeleton joints* HR dan atau HL berada dibawah (lebih kecil daripada) koordinat *skeleton joints* N dan diatas (lebih besar daripada) koordinat *skeleton joints* SM, maka posisi diidentifikasi berada pada Area 2.
3. Jika koordinat *skeleton joints* HR dan atau HL berada dibawah (lebih kecil daripada) koordinat *skeleton joints* SM, maka posisi diidentifikasi berada pada Area 3.



Gambar 3.4 Posisi Gerakan Tangan pada Tubuh Mengacu pada Leher (Vektor N) dan Tulang Belakang Tengah (Vektor SM)

Posisi gerakan tangan yang sudah teridentifikasi kemudian dijadikan fitur data. Terdapat dua fitur data yang dihasilkan oleh penentuan posisi gerakan tangan. Masing-masing fitur mewakili tangan pengguna yaitu tangan kanan dan tangan kiri.

5. Fitur data akhir

Setelah seluruh fitur yang diperlukan sudah didapat. Fitur-fitur tersebut disatukan dan disimpan. Susunan dari fitur-fitur tersebut dapat dilihat pada Tabel 3.6.

Tabel 3.6 Seluruh Fitur yang Digunakan

Fitur	Indeks Fitur
Kuantisasi Tangan Kiri	Fitur ke 1 s.d. 18
Kuantisasi Tangan Kanan	Fitur ke 19 s.d. 36
Posisi Tangan Kiri	Fitur ke 37
Posisi Tangan Kanan	Fitur ke 38
Statis	Fitur ke 39 s.d. 59

3.2.6.2 Rancangan Proses *Training Dataset*

Seperti yang dapat dilihat pada

Gambar 3.5, proses *training dataset* dimulai dengan menyimpan fitur data yang didapatkan sebelumnya terlebih dahulu ke dalam sebuah berkas berekstensi .csv. Data *training* tersebut kemudian diolah pada perangkat lunak WEKA menggunakan algoritma Random Tree. Keluaran yang didapatkan dari hasil *training* data tersebut berupa model Decision Tree dan digunakan sebagai *classifier* ketika pengguna melakukan *testing* data. Proses pembentukan model Decision Tree dilakukan 2 bagian. Proses ini dilakukan baik pada fitur statis dan fitur dinamis, dikarenakan fitur-fitur yang dibutuhkan oleh bahasa isyarat statis dan dinamis berbeda. Hasil model Decision Tree Statis dapat dilihat pada Gambar 3.6 dan model Decision Tree Dinamis Gambar 3.7.

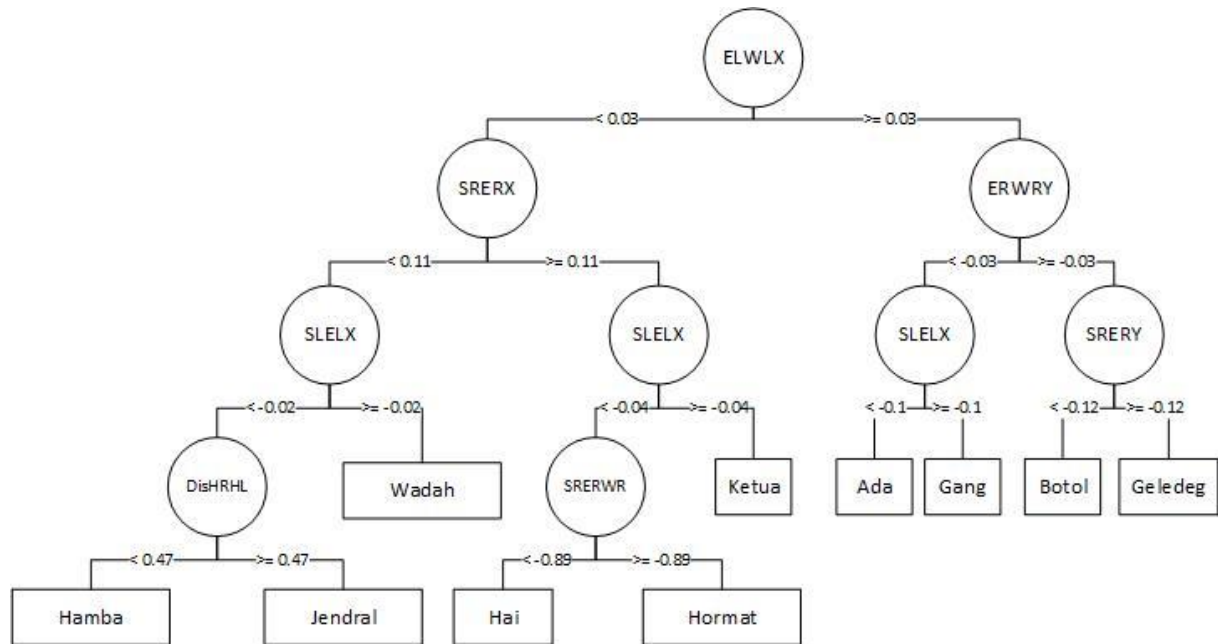


Gambar 3.5 Rancangan Proses *Training Dataset*

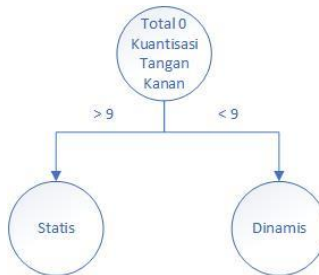
3.2.6.3 Rancangan Proses *Testing Dataset*

Proses *testing* data dilakukan dengan mengenali gerakan isyarat yang dilakukan merupakan gerakan isyarat statis atau dinamis. Berdasarkan uji coba yang telah dilakukan penulis, pemisah gerak bahasa isyarat statis dan dinamis dapat dilihat pada Gambar 3.8. Setelah hasil data fitur baru diambil, data tersebut dimasukkan kedalam *classifier* yang sudah didapatkan pada proses *training* data di Subbab 3.2.6.2. Keluaran hasil klasifikasi yang dilakukan tersebut berupa prediksi gerakan bahasa isyarat yang

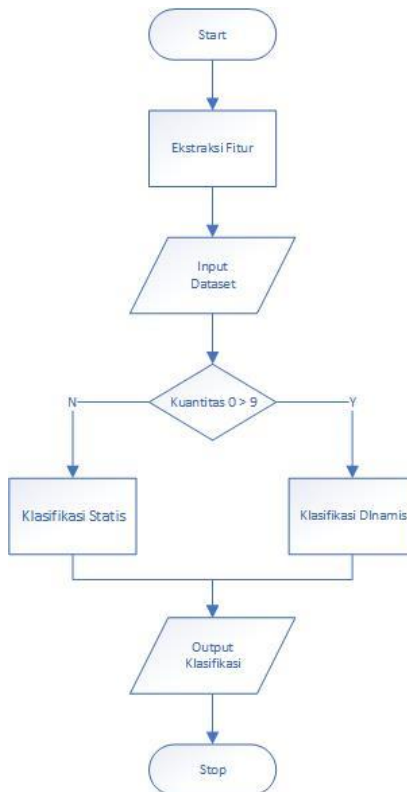
dimaksud oleh pengguna dalam bentuk gambar dan juga tulisan bahasa isyarat. Dapat dilihat pada Gambar 3.9



Gambar 3.6 Decision Tree Statis



Gambar 3.8 Decision Tree Penentu Statis atau Dinamis



Gambar 3.9 Rancangan Proses *Testing Dataset*

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan sistem. Implementasi berupa kode sumber dengan bahasa pemrograman C# dengan menggunakan Kinect SDK. Berikut ini akan diperkenalkan lingkungan yang digunakan untuk mengimplementasikan perangkat lunak ini.

4.1 Lingkungan Pembangunan

Dalam membangun aplikasi ini digunakan beberapa perangkat pendukung, baik perangkat keras maupun perangkat lunak. Lingkungan pembangunan aplikasi ini dijelaskan sebagai berikut.

4.1.1 Lingkungan Pembangunan Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan aplikasi ini adalah sebuah perangkat laptop dengan spesifikasi sebagai berikut:

- Prosesor Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
- Memori (RAM) 16,00 GB

Dengan tambahan perangkat lunak Kinect Sensor Xbox One.

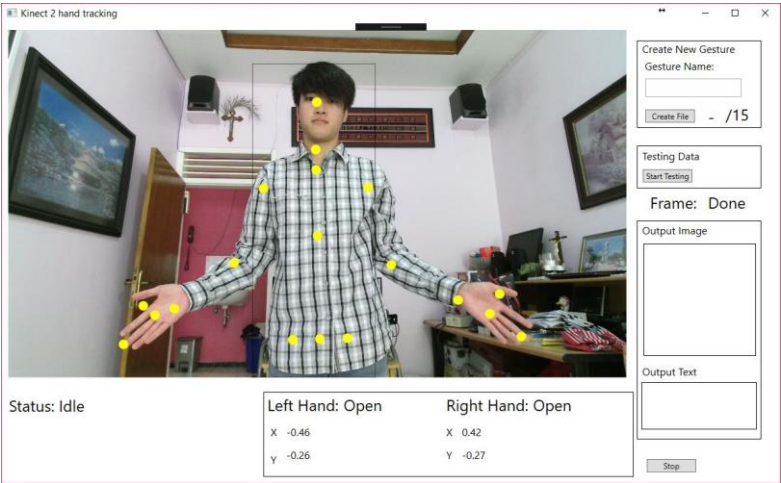
4.1.2 Lingkungan Pembangunan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membuat aplikasi ini sebagai berikut.

- Microsoft Visual Studio 2015
- Windows 10 64 bit sebagai sistem operasi
- Kinect SDK

4.2 Implementasi Antarmuka

Seperti yang telah dijelaskan pada Subbab 3.2.5, modul pengenalan bahasa isyarat yang akan dibuat hanya akan memiliki satu window utama yang sudah mencakup semua fungsionalitas perangkat lunak yang dibutuhkan. Tampilan antarmuka perangkat dapat dilihat pada Gambar 4.1. Sedangkan kode sumber untuk antarmuka perangkat lunak dapat dilihat pada Kode Sumber 4.1.



Gambar 4.1 Antarmuka Perangkat Lunak

1	<Window x:Class="KinectHandTracking.MainWindow"
2	xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3	xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4	Title="Kinect 2 hand tracking" Height="738.5" Width="1200"
	Loaded="Window_Loaded" Closed="Window_Closed">
5	
6	<Grid Margin="0,0,2,0">
7	<Border BorderBrush="Black" BorderThickness="1"
	HorizontalAlignment="Left" Height="129" Margin="398,561,0,0"
	VerticalAlignment="Top" Width="563"/>
8	<Border BorderBrush="Black" BorderThickness="1"

```

HorizontalAlignment="Left" Height="171" Margin="976,335,0,0"
VerticalAlignment="Top" Width="171"/>
9         <Border BorderBrush="Black" BorderThickness="1"
HorizontalAlignment="Left" Height="133" Margin="966,26,0,0"
VerticalAlignment="Top" Width="189"/>
10        <Border BorderBrush="Black" BorderThickness="1"
HorizontalAlignment="Left" Height="65" Margin="966,186,0,0"
VerticalAlignment="Top" Width="189"/>
11        <Border BorderBrush="Black" BorderThickness="1"
HorizontalAlignment="Left" Height="333" Margin="966,300,0,0"
VerticalAlignment="Top" Width="189"/>
12        <Border BorderBrush="Black" BorderThickness="1"
HorizontalAlignment="Left" Height="72" Margin="973,546,0,0"
VerticalAlignment="Top" Width="175"/>
13
14        <Viewbox Width="940" Height="700" Margin="10,10,0,0"
HorizontalAlignment="Left" VerticalAlignment="Top">
15            <Grid>
16                <Image Name="camera" Width="1920"
Height="1080" />
17                <Canvas Name="canvas" Width="1920"
Height="1080" ClipToBounds="True" Margin="-10,10,10,-10" />
18                <Border BorderBrush="Black"
BorderThickness="1" HorizontalAlignment="Left" Height="976"
VerticalAlignment="Top" Width="382" Margin="758,104,0,0"/>
19            </Grid>
20        </Viewbox>
21
22        <Label Name="label3" Content="Testing Data"
HorizontalAlignment="Left" Margin="969,186,0,0"
VerticalAlignment="Top" Width="152" FontSize="16"/>
23        <Button Name="OneTestButton" Content="Start Testing"
HorizontalAlignment="Left" Margin="975,222,0,0"
VerticalAlignment="Top" Width="75"
Click="OneTestButton_Click"/>
24
25        <Label Name="label5" Content="Create New Gesture"
HorizontalAlignment="Left" Margin="969,23,0,0"
VerticalAlignment="Top" Width="152" FontSize="16"/>
26        <Label Name="label6" Content="Gesture Name:"
HorizontalAlignment="Left" Margin="973,49,0,0"
VerticalAlignment="Top" Width="115" FontSize="16"/>
27        <TextBox Name="fileName" HorizontalAlignment="Left"
Height="28" Margin="979,84,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="146" FontSize="16"/>
28        <Button Name="createButton" Content="Create File"
HorizontalAlignment="Left" Margin="979,131,0,0"
VerticalAlignment="Top" Width="75"
Click="createButton_click"/>

```

29	<Label Name="ambil" Content="-" HorizontalAlignment="Left" Height="47" Margin="1069,120,62,0" VerticalAlignment="Top" Width="59" FontSize="24"/>
30	<Label Name="label13" Content="/15" HorizontalAlignment="Left" Height="47" Margin="1096,117,0,0" VerticalAlignment="Top" Width="59" FontSize="24"/>
31	
32	<Label Name="label7" Content="Output Image" HorizontalAlignment="Left" Margin="969,299,0,0" VerticalAlignment="Top" Width="152" FontSize="16"/>
33	<Border HorizontalAlignment="Left" Height="171" Margin="976,336,0,0" VerticalAlignment="Top" Width="171" >
34	<Image x:Name="outputImage" Stretch="UniformToFill" Margin="0,-3,0.2,2.2"/>
35	</Border>
36	
37	<Label Name="label8" Content="Output Text" HorizontalAlignment="Left" Margin="969,514,0,0" VerticalAlignment="Top" Width="152" FontSize="16"/>
38	<Label Name="outputText" Content="" HorizontalAlignment="Left" Margin="973,555,0,0" VerticalAlignment="Top" Height="48" Width="174" FontSize="26.667" RenderTransformOrigin="0.981,2.1"/>
39	
40	<Label Name="label9" Content="Status:" HorizontalAlignment="Left" Margin="6,560,0,0" VerticalAlignment="Top" Width="100" FontSize="24" Height="47"/>
41	<Label Name="statusDetail" Content="Kinect Not Connected" HorizontalAlignment="Left" Height="47" Margin="82,560,0,0" VerticalAlignment="Top" Width="341" FontSize="24"/>
42	
43	<Label Name="label10" Content="Left Hand:" HorizontalAlignment="Left" Margin="399,559,0,0" VerticalAlignment="Top" Width="119" FontSize="24" Height="47"/>
44	<Label Name="tblLeftHandState" Content="Not Detected" HorizontalAlignment="Left" Height="47" Margin="515,559,0,0" VerticalAlignment="Top" Width="187" FontSize="24"/>
45	<Label Name="labelxL" Content="X" HorizontalAlignment="Left" Height="42" Margin="404,606,0,0" VerticalAlignment="Top" FontSize="16"/>
46	<Label Name="labelyL" Content="Y" HorizontalAlignment="Left" Height="42" Margin="404,648,0,- 11" VerticalAlignment="Top" FontSize="16"/>
47	<Label Name="xL" Content="-"

48	HorizontalAlignment="Left" Height="42" Margin="428,606,0,0" VerticalAlignment="Top" FontSize="16"/> <Label Name="yL" Content="-"
49	HorizontalAlignment="Left" Height="42" Margin="428,641,0,0" VerticalAlignment="Top" FontSize="16"/>
50	<Label Name="label11" Content="Right Hand:" HorizontalAlignment="Left" Margin="671,559,0,0" VerticalAlignment="Top" Width="144" FontSize="24" Height="47"/>
51	<Label Name="tblRightHandState" Content="Not Detected" HorizontalAlignment="Left" Height="47" Margin="803,559,0,0" VerticalAlignment="Top" Width="187" FontSize="24"/>
52	<Label Name="labelxR" Content="X" HorizontalAlignment="Left" Height="42" Margin="671,606,0,0" VerticalAlignment="Top" FontSize="16"/>
53	<Label Name="labelyR" Content="Y" HorizontalAlignment="Left" Height="42" Margin="671,641,0,0" VerticalAlignment="Top" FontSize="16"/>
54	<Label Name="xR" Content="-" HorizontalAlignment="Left" Height="42" Margin="695,606,0,0" VerticalAlignment="Top" FontSize="16"/>
55	<Label Name="yR" Content="-" HorizontalAlignment="Left" Height="42" Margin="695,641,0,0" VerticalAlignment="Top" FontSize="16"/>
56	
57	<Label Name="label12" Content="Frame:" HorizontalAlignment="Left" Margin="981,251,0,0" VerticalAlignment="Top" Width="144" FontSize="24" Height="47"/>
58	<Label Name="ambilData" Content="Done" HorizontalAlignment="Left" Height="47" Margin="1069,251,- 66,0" VerticalAlignment="Top" Width="187" FontSize="24"/>
59	
60	<Button Name="stopButton" Content="Stop" HorizontalAlignment="Left" Margin="981,663,0,0" VerticalAlignment="Top" Width="75" Click="stopButton_Click"/>
61	
62	</Grid>
63	</Window>

Kode Sumber 4.1 Tampilan Perangkat Lunak

4.3 Implementasi Perangkat Lunak

Pada subbab ini akan dibahas mengenai implementasi perangkat lunak dari kasus penggunaan ke dalam baris kode. Dijelaskan juga dengan fungsi yang dibutuhkan untuk menunjang perangkat lunak ini agar dapat berjalan sebagaimana mestinya. Implementasi ini dilakukan menggunakan Microsoft Visual Studio 2015 dengan bahasa pemrograman C#.

4.3.1 Implementasi Pendeteksian *Skeleton* Pengguna

Untuk menjalankan perangkat lunak ini diperlukan bantuan untuk mendeteksi *skeleton* pengguna, sehingga dibutuhkan bantuan dari perangkat keras Kinect 2.0. Sebelum mendeteksi *skeleton*, Kinect 2.0 harus diintegrasikan dengan program terlebih dahulu.

Kode sumber proses integrasi Kinect 2.0 dapat dilakukan seperti pada Kode Sumber 4.2. Untuk melakukan ekstraksi fitur, perangkat lunak mendeteksi tubuh pengguna terlebih dahulu. Ketika tubuh pengguna sudah terdeteksi, perangkat lunak kemudian menggambarkan *skeleton* pengguna secara keseluruhan termasuk 11 *skeleton joints* yang akan digunakan dalam proses ekstraksi fitur statis dan dinamis serta menentukan posisi gerakan. Kode sumber untuk mendeteksi *skeleton* pengguna dapat dilihat pada Kode Sumber 4.3.

1	<code>_sensor = KinectSensor.Default();</code>
2	<code>if (_sensor != null)</code>
3	<code>{</code>
4	<code> _sensor.Open();</code>
5	<code> statusDetail.Content = "Idle";</code>
6	<code> _reader =</code>
	<code> _sensor.OpenMultiSourceFrameReader(FrameSourceTypes.Color </code>
	<code> FrameSourceTypes.Depth FrameSourceTypes.Infrared </code>
	<code> FrameSourceTypes.Body);</code>
7	<code> _reader.MultiSourceFrameArrived +=</code>
	<code> Reader_MultiSourceFrameArrived;</code>
8	<code>}</code>

Kode Sumber 4.2 Kode Sumber Integrasi Kinect

```

1  async void Reader_MultiSourceFrameArrived(object sender,
MultiSourceFrameArrivedEventArgs e)
2  {
3      var reference = e.FrameReference.AcquireFrame();
4
5      #region Acquire Frame Color
6      // Color
7      using (var frame =
reference.ColorFrameReference.AcquireFrame())
8      {
9          if (frame != null)
10         {
11             camera.Source = frame.ToBitmap();
12         }
13     }
14     #endregion
15
16     using (var frame =
reference.BodyFrameReference.AcquireFrame())
17     {
18         if (frame != null)
19         {
20             canvas.Children.Clear();
21             _bodies = new
22             Body[frame.BodyFrameSource.BodyCount];
23             frame.GetAndRefreshBodyData(_bodies);
24
25             var csv = new StringBuilder();
26             string filePath = "E:\\Kuliah\\TUGAS
AKHIR\\TABI\\Data\\DataSet\\dataori.csv";
27             string imagePath = "";
28
29             foreach (var body in _bodies)
30             {
31                 if (body != null)
32                 {
33                     if (body.IsTracked)
34                     {
35                         Joint handRight =
body.Joints[JointType.HandRight];
36                         Joint handLeft =
body.Joints[JointType.HandLeft];
37                         Joint wristRight =
body.Joints[JointType.WristRight];
38                         Joint wristLeft =
body.Joints[JointType.WristLeft];
39                         Joint elbowRight =
body.Joints[JointType.ElbowRight];

```

```

39         Joint elbowLeft =
body.Joints[JointType.ElbowLeft];
40         Joint shoulderRight =
body.Joints[JointType.ShoulderRight];
41         Joint shoulderLeft =
body.Joints[JointType.ShoulderLeft];
42         Joint spineMid =
body.Joints[JointType.SpineMid];
43         Joint spineShoulder =
body.Joints[JointType.SpineShoulder];
44         Joint neck =
body.Joints[JointType.Neck];
45
46         foreach (Joint joint in
body.Joints.Values)
47         {
48             if (joint.TrackingState ==
TrackingState.Tracked)
49             {
50                 CameraSpacePoint
jointPosition = joint.Position;
51                 Point point = new Point();
52
53                 ColorSpacePoint colorPoint =
_sensor.CoordinateMapper.MapCameraPointToColorSpace(jointPosi
tion);
54                 point.X =
float.IsInfinity(colorPoint.X) ? 0 : colorPoint.X;
55                 point.Y =
float.IsInfinity(colorPoint.Y) ? 0 : colorPoint.Y;
56
57                 Ellipse ellipse = new Ellipse
58                 {
59                     Fill = Brushes.Yellow,
60                     Width = 30,
61                     Height = 30
62                 };
63
64                 Canvas.SetLeft(ellipse,
point.X - ellipse.Width / 2);
65                 Canvas.SetTop(ellipse,
point.Y - ellipse.Height / 2);
66
67                 canvas.Children.Add(ellipse);
68             }
69         }
70     }
71 }
72

```


73	}
74	}
75	}

Kode Sumber 4.3 Kode Sumber Deteksi *Skeleton* Pengguna

4.3.2 Implementasi Proses Ekstraksi Fitur

Pada Kode Sumber 4.4 dijelaskan mengenai proses ekstraksi fitur statis dan dinamis pada setiap data serta menentukan posisi gerakan tangan yang dilakukan oleh pengguna. Penjelasan kode sumber tersebut adalah sebagai berikut:

1. Variabel *i* menunjukkan *frame* ke-*i* yang sedang diolah.
2. Variabel *Ny* menunjukkan koordinat *y skeleton joints* N.
3. Variabel *SMy* menunjukkan koordinat *y skeleton joints* SM.
4. Variabel *SS* menunjukkan koordinat *skeleton joints* SS.
5. Variabel *SL* menunjukkan koordinat *skeleton joints* SL.
6. Variabel *EL* menunjukkan koordinat *skeleton joints* EL.
7. Variabel *WL* menunjukkan koordinat *skeleton joints* WL.
8. Variabel *HL* menunjukkan koordinat *skeleton joints* HL.
9. Variabel *SR* menunjukkan koordinat *skeleton joints* SR.
10. Variabel *ER* menunjukkan koordinat *skeleton joints* ER.
11. Variabel *WR* menunjukkan koordinat *skeleton joints* WR.
12. Variabel *HR* menunjukkan koordinat *skeleton joints* HR.

13. Variabel *tanganKanan* menunjukkan posisi tangan kanan.
14. Variabel *tanganKiri* menunjukkan posisi tangan kiri.
15. Variabel *deltaHLx* menunjukkan Δx tangan kiri.
16. Variabel *deltaHLy* menunjukkan Δy tangan kiri.
17. Variabel *alphaHL* menunjukkan α_t tangan kiri.
18. Variabel *deltaHRx* menunjukkan Δx tangan kanan.
19. Variabel *deltaHRy* menunjukkan Δy tangan kanan.
20. Variabel *alphaHR* menunjukkan α_t tangan kanan.
21. Variabel *kuantKiri* menunjukkan hasil kuantisasi fitur dinamis tangan kiri.
22. Variabel *kuantKanan* menunjukkan hasil kuantisasi fitur dinamis tangan kanan.
23. Variabel *SRER* menunjukkan vektor bahu menuju siku tangan kanan.
24. Variabel *ERWR* menunjukkan vektor siku menuju pergelangan tangan kanan.
25. Variabel *WRHR* menunjukkan vektor pergelangan menuju telapak tangan kanan.
26. Variabel *SLEL* menunjukkan vektor bahu menuju siku tangan kiri.
27. Variabel *ELWL* menunjukkan vektor siku menuju pergelangan tangan kiri.
28. Variabel *WLHL* menunjukkan vektor pergelangan menuju telapak tangan kiri.
29. Variabel *HRHL* menunjukkan vektor telapak tangan kanan menuju telapak tangan kiri.
30. Variabel *SSSRER* menunjukkan sudut pada bahu tangan kanan.
31. Variabel *SRERWR* menunjukkan sudut pada siku tangan kanan.
32. Variabel *ERWRHR* menunjukkan sudut pada pergelangan tangan kanan.
33. Variabel *SSSLEL* menunjukkan sudut pada bahu tangan kiri.

34. Variabel *SLELWL* menunjukkan sudut pada siku tangan kiri.
35. Variabel *ELWLHL* menunjukkan sudut pada pergelangan tangan kiri.
36. Variabel *DisHRHL* menunjukkan jarak antara telapak tangan kanan dan telapak tangan kiri.

Fitur statis dan dinamis serta posisi gerakan tangan digabungkan menjadi fitur data yang kemudian disimpan ke dalam sebuah berkas berekstensi .csv dengan pembagian fitur data berdasarkan Tabel 2.1. Sebelum disimpan, seluruh atribut yang ada di dalam fitur data diubah terlebih dahulu menjadi atribut tipe string. Kode sumber penyimpanan fitur data dapat dilihat pada Kode Sumber 4.5.

```
1  if (i < 40 && statusAmbil != 0)
2  {
3      if (i < 38)
4      {
5          ambilData.Content = (i + 1).ToString();
6      }
7      else ambilData.Content = "Done";
8
9      if (i == 1)
10     {
11         Ny = (neck.Position.Y);
12         SMy = (spineMid.Position.Y);
13     }
14     if (i == 20)
15     {
16         if (HL.Y > Ny) tanganKiri = "Kepala";
17         else if (HL.Y < SMy) tanganKiri = "Perut";
18         else tanganKiri = "Dada";
19
20         if (HR.Y > Ny) tanganKanan = "Kepala";
21         else if (HR.Y < SMy) tanganKanan = "Perut";
22         else tanganKanan = "Dada";
23     }
24
25     // tangan kanan
26     if (deltaHRx == 0 && deltaHRY == 0)
27     {
28         alphaHR = -1;
29     }
```

```

30     else if (deltaHRx >= 0 && deltaHRy >= 0)
31     {
32         alphaHR = (Math.Atan(deltaHRy / deltaHRx)) * (180 /
Math.PI);
33     }
34     else if (deltaHRx < 0)
35     {
36         alphaHR = (Math.Atan(deltaHRy / deltaHRx)) * (180 /
Math.PI) + 180;
37     }
38     else
39     {
40         alphaHR = (Math.Atan(deltaHRy / deltaHRx)) * (180 /
Math.PI) + 360;
41     }
42
43     // tangan kiri
44     if (deltaHLx == 0 && deltaHLy == 0)
45     {
46         alphaHL = -1;
47     }
48     else if (deltaHLx >= 0 && deltaHLy >= 0)
49     {
50         alphaHL = (Math.Atan(deltaHLy / deltaHLx)) * (180 /
Math.PI);
51     }
52     else if (deltaHRx < 0)
53     {
54         alphaHL = (Math.Atan(deltaHLy / deltaHLx)) * (180 /
Math.PI) + 180;
55     }
56     else
57     {
58         alphaHL = (Math.Atan(deltaHLy / deltaHLx)) * (180 /
Math.PI) + 360;
59     }
60
61     // tangan kanan
62     if (alphaHR >= 315) { kuantKanan[i] = 8; }
63     else if (alphaHR >= 270) { kuantKanan[i] = 7; }
64     else if (alphaHR >= 225) { kuantKanan[i] = 6; }
65     else if (alphaHR >= 180) { kuantKanan[i] = 5; }
66     else if (alphaHR >= 135) { kuantKanan[i] = 4; }
67     else if (alphaHR >= 90) { kuantKanan[i] = 3; }
68     else if (alphaHR >= 45) { kuantKanan[i] = 2; }
69     else if (alphaHR >= 0) { kuantKanan[i] = 1; }
70     else if (alphaHR < 0) { kuantKanan[i] = 0; }
71
72     // tangan kiri

```

73	if (alphaHL >= 315) { kuantKiri[i] = 8; }
74	else if (alphaHL >= 270) { kuantKiri[i] = 7; }
75	else if (alphaHL >= 225) { kuantKiri[i] = 6; }
76	else if (alphaHL >= 180) { kuantKiri[i] = 5; }
77	else if (alphaHL >= 135) { kuantKiri[i] = 4; }
78	else if (alphaHL >= 90) { kuantKiri[i] = 3; }
79	else if (alphaHL >= 45) { kuantKiri[i] = 2; }
80	else if (alphaHL >= 0) { kuantKiri[i] = 1; }
81	else if (alphaHL < 0) { kuantKiri[i] = 0; }
82	
83	if (1 < i && i < 38)
84	{
85	SRER += ER - SR;
86	ERWR += WR - ER;
87	WRHR += HR - WR;
88	SLEL += EL - SL;
89	ELWL += WL - EL;
90	WLHL += HL - WL;
91	HRHL += HL - HR;
92	
93	Vector v1, v2;
94	double res;
95	
96	//SS-SR-ER
97	v1 = SS - SR;
98	v2 = ER - SR;
99	res = Vector.AngleBetween(v1, v2);
100	SSSRER += (double)res * Math.PI / 180;
101	
102	//SR-ER-WR
103	v1 = SR - ER;
104	v2 = WR - ER;
105	res = Vector.AngleBetween(v1, v2);
106	SRERWR += (double)res * Math.PI / 180;
107	
108	//ER-WR-HR
109	v1 = ER - WR;
110	v2 = HR - WR;
111	res = Vector.AngleBetween(v1, v2);
112	ERWRHR += (double)res * Math.PI / 180;
113	
114	//SS-SL-EL
115	v1 = SS - SL;
116	v2 = EL - SL;
117	res = Vector.AngleBetween(v1, v2);
118	SSSLEL += (double)res * Math.PI / 180;
119	
120	//SL-EL-WL
121	v1 = SL - EL;

123	v2 = WL - EL;
124	res = Vector.AngleBetween(v1, v2);
125	SLELWL += (double)res * Math.PI / 180;
126	
127	//EL-WL-HL
128	v1 = EL - WL;
129	v2 = HL - WL;
130	res = Vector.AngleBetween(v1, v2);
131	ELWLHL += (double)res * Math.PI / 180;
132	
133	//Distance HR - HL
	DisHRHL += Math.Sqrt((HR.X - HL.X) * (HR.X - HL.X)
	+ (HR.Y - HL.Y) * (HR.Y - HL.Y)); // sqrt from dotproduct
	= length
134	}
135	}

Kode Sumber 4.4 Kode Sumber Ekstraksi Fitur Statis, Dinamis dan Menentukan Posisi Gerakan Tangan dari *Skeleton* Pengguna

1	var stringSRER = SRER.ToString();
2	var stringERWR = ERWR.ToString();
3	var stringWRHR = WRHR.ToString();
4	var stringSLEL = SLEL.ToString();
5	var stringELWL = ELWL.ToString();
6	var stringWLHL = WLHL.ToString();
8	var stringHRHL = HRHL.ToString();
8	var stringSSSRER = SSSRER.ToString();
9	var stringSRERWR = SRERWR.ToString();
10	var stringERWRHR = ERWRHR.ToString();
11	var stringSSSLEL = SSSLEL.ToString();
12	var stringSLELWL = SLELWL.ToString();
13	var stringELWLHL = ELWLHL.ToString();
14	var stingDisHRHL = DisHRHL.ToString();
15	
16	var stringkuant4 = kuantKiri[4].ToString();
17	var stringkuant6 = kuantKiri[6].ToString();
18	var stringkuant8 = kuantKiri[8].ToString();
19	var stringkuant10 = kuantKiri[10].ToString();
20	var stringkuant12 = kuantKiri[12].ToString();
21	var stringkuant14 = kuantKiri[14].ToString();
22	var stringkuant16 = kuantKiri[16].ToString();
23	var stringkuant18 = kuantKiri[18].ToString();
24	var stringkuant20 = kuantKiri[20].ToString();
25	var stringkuant22 = kuantKiri[22].ToString();
26	var stringkuant24 = kuantKiri[24].ToString();
27	var stringkuant26 = kuantKiri[26].ToString();
28	var stringkuant28 = kuantKiri[28].ToString();
29	var stringkuant30 = kuantKiri[30].ToString();
30	var stringkuant32 = kuantKiri[32].ToString();

31	var stringkuant34 = kuantKiri[34].ToString();
32	var stringkuant36 = kuantKiri[36].ToString();
33	var stringkuant38 = kuantKiri[38].ToString();
34	
35	var stringkuant43 = kuantKanan[4].ToString();
36	var stringkuant45 = kuantKanan[6].ToString();
37	var stringkuant47 = kuantKanan[8].ToString();
38	var stringkuant49 = kuantKanan[10].ToString();
39	var stringkuant51 = kuantKanan[12].ToString();
40	var stringkuant53 = kuantKanan[14].ToString();
41	var stringkuant55 = kuantKanan[16].ToString();
42	var stringkuant57 = kuantKanan[18].ToString();
43	var stringkuant59 = kuantKanan[20].ToString();
44	var stringkuant61 = kuantKanan[22].ToString();
45	var stringkuant63 = kuantKanan[24].ToString();
46	var stringkuant65 = kuantKanan[26].ToString();
47	var stringkuant67 = kuantKanan[28].ToString();
48	var stringkuant69 = kuantKanan[30].ToString();
49	var stringkuant71 = kuantKanan[32].ToString();
50	var stringkuant73 = kuantKanan[34].ToString();
51	var stringkuant75 = kuantKanan[36].ToString();
52	var stringkuant77 = kuantKanan[38].ToString();
53	
54	var stringtangankiri = tanganKiri;
55	var stringtangankanan = tanganKanan;
56	var stringnamagerakan = namaGerakan;
57	
58	if (statusAmbil == 1)
59	{
60	var newLine =
	string.Format("{0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10}
	, {11},{12},{13},{14},{15},{16},{17},{18},{19},{20},{21},{22}
	},{23},{24},{25},{26},{27},{28},{29},{30},{31},{32},{33},{3}
	4},{35},{36},{37},{38},{39},{40},{41},{42},{43},{44},{45},{
	46},{47},{48},{49},{50},{51},{52}",
61	stringkuant4, stringkuant6,
62	stringkuant8, stringkuant10,
63	stringkuant12, stringkuant14,
64	stringkuant16, stringkuant18,
65	stringkuant20, stringkuant22,
66	stringkuant24, stringkuant26,
67	stringkuant28, stringkuant30,
68	stringkuant32, stringkuant34,
69	stringkuant36, stringkuant38,
70	stringkuant43, stringkuant45,
71	stringkuant47, stringkuant49,
72	stringkuant51, stringkuant53,
73	stringkuant55, stringkuant57,
74	stringkuant59, stringkuant61,

75	stringkuant63, stringkuant65,
76	stringkuant67, stringkuant69,
77	stringkuant71, stringkuant73,
78	stringkuant75, stringkuant77,
79	stringtangankiri, stringtangankan,
80	stringSRER, stringERWR, stringWRHR,
81	stringSLEL, stringELWL, stringWLHL,
82	stringHRHL,
83	stringSSSRER, stringSRERWR, stringERWRHR,
84	stringSSSLEL, stringSLELWL, stringELWLHL,
85	stingDisHRHL,
86	stringnamagerakan
87	);
88	
89	csv.AppendLine(newLine);
90	}

Kode Sumber 4.5 Kode Sumber Menyimpan Hasil Ekstraksi Fitur Statis, Dinamis dan Posisi Gerak Tangan dari *Skeleton* Pengguna

4.3.4 Implementasi Proses *Testing* Data

Hasil dari *training* data yang dapat berupa *tree* yang dipisahkan dalam 2 bagian, yaitu *tree* statis dan dinamis yang dilihat pada Gambar 3.6 dan Gambar 3.7. Model *tree* tersebut kemudian diimplentasikan ke dalam perangkat lunak seperti yang dapat dilihat pada Kode Sumber 4.6. Keluaran dari kode sumber tersebut berupa prediksi gerakan yang dilakukan oleh pengguna ketika melakukan proses *testing* baik itu dalam bentuk tulisan maupun dalam bentuk gambar.

1	else if (statusAmbil == 2)
2	{
3	// Statis or Dinamis
4	flag4 = 0;
5	for (j = 2; j < 39; j = j+2)
6	{
7	if (kuantKanan[j] == 0)
8	{
9	flag4++;
10	}

11	}
12	
13	if (flag4 > 9)
14	{
15	// Statis
16	if (ELWL.X < 0.03)
17	{
18	if (SRER.X < 0.11)
19	{
20	if (SLEL.X < -0.02)
21	{
22	if (DisHRHL < 0.47)
23	{
24	outputText.Content = "Hamba";
25	}
26	else if (DisHRHL >= 0.47)
27	{
28	outputText.Content = "Jendral";
29	}
30	}
31	else if (SLEL.X >= -0.02)
32	{
33	outputText.Content = "Wadah";
34	}
35	}
36	else if (SRER.X >= 0.11)
37	{
38	if (SLEL.X < -0.04)
39	{
40	if (SRERWR < -0.84)
41	{
42	outputText.Content = "Hai";
43	}
44	else if (SRERWR >= -0.84)
45	{
46	outputText.Content = "Hormat";
47	}
48	}
49	else if (SLEL.X >= -0.04)
50	{
51	if (SRERWR < -0.84)
52	{
53	outputText.Content = "Hai";
54	}
55	else if (SRERWR >= -0.84)
56	{
57	outputText.Content = "Ketua";
58	}
59	}

```

60     }
61 }
62 else if (ELWL.X >= 0.03)
63 {
64     if (ERWR.Y < -0.03)
65     {
66         if (SLEL.X < -0.1)
67         {
68             outputText.Content = "Ada";
69         }
70         else if (SLEL.X >= -0.1)
71         {
72             outputText.Content = "Gang";
73         }
74     }
75     else if (ERWR.Y >= -0.03)
76     {
77         if (SRER.Y < -0.12)
78         {
79             outputText.Content = "Botol";
80         }
81         else if (SRER.Y >= -0.12)
82         {
83             outputText.Content = "Geledeg";
84         }
85     }
86 }
87 }
88 else if (flag4 <= 9)
89 {
90     // Dinamis
91     if (kuantKiri[20] < 6)
92     {
93         if (kuantKiri[24] < 5.5)
94         {
95             if (kuantKiri[34] < 0.5)
96             {
97                 if (kuantKanan[18] < 2)
98                 {
99                     if (kuantKanan[8] < 0.5)
100                     {
101                         outputText.Content = "Samping";
102                     }
103                     else if (kuantKanan[8] >= 0.5)
104                     {
105                         outputText.Content = "Sempit";
106                     }
107                 }
108                 else if (kuantKanan[18] >= 2)

```

109	{
110	if (kuantKanan[26] < 4.5)
111	{
112	if (kuantKanan[36] < 2)
113	{
114	if (kuantKanan[18] < 6.5)
115	{
116	outputText.Content =
"Topeng";	
117	}
118	else if (kuantKanan[18] >=
6.5)	
119	{
120	outputText.Content =
"Samping";	
121	}
122	}
123	else if (kuantKanan[36] >= 2)
124	{
125	if (kuantKiri[32] < 0.5)
126	{
127	if (kuantKiri[18] < 0.5)
128	{
129	outputText.Content =
"Topeng";	
130	}
131	else if (kuantKiri[18]
>= 0.5)	
132	{
133	outputText.Content =
"Bingung";	
134	}
135	}
136	else if (kuantKiri[32] >=
0.5)	
137	{
138	outputText.Content =
"Bingung";	
139	}
140	}
141	}
142	else if (kuantKanan[26] >= 4.5)
143	{
144	if (kuantKiri[16] < 3.5)
145	{
146	outputText.Content =
"Sempit";	
147	}
148	else if (kuantKiri[16] >= 3.5)

149	{
150	outputText.Content =
151	"Bingkai";
152	}
153	}
154	}
155	else if (kuantKiri[34] >= 0.5)
156	{
157	if (tanganKiri == "Perut")
158	{
159	if (kuantKanan[26] < 4.5)
160	{
161	if (kuantKiri[30] < 0.5)
162	{
163	if (kuantKanan[32] < 2)
164	{
165	if (tanganKanan ==
166	{
167	outputText.Content =
168	}
169	else if (tanganKanan ==
170	"Perut")
171	{
172	outputText.Content =
173	}
174	else if (tanganKanan ==
175	"Dada")
176	{
177	outputText.Content =
178	}
179	else if (kuantKanan[32] >=
180	2)
181	{
182	outputText.Content =
183	}
184	else if (kuantKanan[30] >= 0.5)
185	{
186	if (kuantKanan[32] < 3)
187	{
	outputText.Content =
	"Topeng";

188		}
189		else if (kuantKanan[32] >=
190	3)	{
191		outputText.Content =
192	"Sempit";	
193		}
194		}
195		else if (kuantKanan[26] >= 4.5)
196		{
197		if (kuantKiri[20] < 0.5)
198		{
199		if (kuantKiri[18] < 2)
200		{
201		outputText.Content =
202	"Sempit";	
203		}
204	0.5)	else if (kuantKiri[18] >=
205		{
206		outputText.Content =
207	"Bingung";	
208		}
209		else if (kuantKiri[20] >= 0.5)
210		{
211	"Sempit";	outputText.Content =
212		}
213		}
214		else if (tanganKiri == "Dada")
215		{
216		outputText.Content = "Rujuk";
217		}
218		else if (tanganKiri == "Kepala")
219		{
220		outputText.Content = "Kijang";
221		}
222		}
223		}
224		else if (kuantKiri[24] >= 5.5)
225		{
226		if (kuantKiri[18] < 2.5)
227		{
228		if (kuantKanan[14] < 3)
229		{
230		outputText.Content = "Kijang";

231	}
232	else if (kuantKanan[14] >= 3)
233	{
234	outputText.Content = "Bola";
235	}
236	}
237	else if (kuantKanan[18] >= 2.5)
238	{
239	outputText.Content = "Badan";
240	}
241	}
242	}
243	else if (kuantKiri[20] >= 6)
244	{
245	if (kuantKanan[30] < 3)
246	{
247	if (kuantKanan[36] < 3)
248	{
249	if (kuantKanan[32] < 4.5)
250	{
251	if (kuantKanan[28] < 1.5)
252	{
253	if (kuantKanan[24] < 2)
254	{
255	outputText.Content =
256	"Besar";
257	}
258	else if (kuantKanan[24] >= 2)
259	{
260	outputText.Content =
261	"Kijang";
262	}
263	else if (kuantKanan[28] >= 1.5)
264	{
265	if (kuantKanan[6] < 1)
266	{
267	outputText.Content =
268	"Kijang";
269	}
270	else if (kuantKanan[6] >= 1)
271	{
272	outputText.Content =
273	"Besar";
274	}
275	}
	}
	}
	else if (kuantKanan[32] >= 4.5)
	{

276	outputText.Content = "Kijang";
277	}
278	}
279	else if (kuantKanan[36] >= 3)
280	{
281	if (kuantKanan[26] < 4)
282	{
283	outputText.Content = "Kijang";
284	}
285	else if (kuantKanan[26] >= 4)
286	{
287	if (kuantKiri[22] < 7.5)
288	{
289	outputText.Content = "Bingkai";
290	}
291	else if (kuantKiri[22] >= 7.5)
292	{
293	outputText.Content = "Badan";
294	}
295	}
296	}
297	}
298	else if (kuantKanan[30] >= 3)
299	{
300	if (kuantKiri[12] < 7.5)
301	{
302	if (kuantKanan[34] < 6.5)
303	{
304	if (kuantKiri[14] < 0.5)
305	{
306	outputText.Content = "Bingung";
307	}
308	else if (kuantKiri[14] >= 0.5)
309	{
310	if (kuantKiri[22] < 4)
311	{
312	outputText.Content =
313	"Rujuk";
314	}
315	else if (kuantKiri[22] >= 4)
316	{
317	outputText.Content = "Bola";
318	}
319	}
320	else if (kuantKanan[34] >= 6.5)
321	{
322	if (kuantKanan[18] < 7.5)
323	{

```

324         outputText.Content = "Badan";
325     }
326     else if (kuantKanan[18] >= 7.5)
327     {
328         outputText.Content = "Bingung";
329     }
330 }
331 }
332 else if (kuantKiri[12] >= 7.5)
333 {
334     if (kuantKiri[14] < 5.5)
335     {
336         if (kuantKanan[20] < 7.5)
337         {
338             outputText.Content = "Badan";
339         }
340         else if (kuantKanan[20] >= 7.5)
341         {
342             outputText.Content = "Bola";
343         }
344     }
345     else if (kuantKiri[14] >= 5.5)
346     {
347         if (tanganKiri == "Perut")
348         {
349             outputText.Content = "Bola";
350         }
351         else if (tanganKiri == "Dada")
352         {
353             outputText.Content = "Bingkai";
354         }
355         else if (tanganKiri == "Kepala")
356         {
357             outputText.Content = "Kijang";
358         }
359     }
360 }
361 }
362 }
363 }
364 }

```

Kode Sumber 4.6 Implementasi Decision Tree dari Hasil Klasifikasi

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada perangkat yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsional secara keseluruhan. Pengujian dilakukan dengan beberapa skenario. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini.

5.1 Lingkungan Pembangunan

Dalam membangun perangkat lunak ini digunakan beberapa perangkat pendukung baik perangkat keras maupun perangkat lunak lainnya. Perangkat keras yang digunakan dalam pembuatan perangkat lunak ini adalah sebuah laptop yang memiliki spesifikasi sebagai berikut:

- Prosesor Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
- Memori (RAM) 16,00 GB
- Kinect Sensor

5.2 Skenario Pengujian

Pengujian dilakukan terhadap 20 bahasa isyarat yang dipilih oleh penulis seperti yang dapat dilihat pada Gambar 5.1. Bahasa isyarat tersebut adalah sebagai berikut:

1. Badan
2. Besar
3. Bingkai
4. Bingung
5. Bola
6. Kijang
7. Rujuk
8. Samping
9. Sempit

10. Topeng
11. Ada
12. Botol
13. Gang
14. Geledeg
15. Hai
16. Hamba
17. Hormat
18. Jendral
19. Ketua
20. Wadah

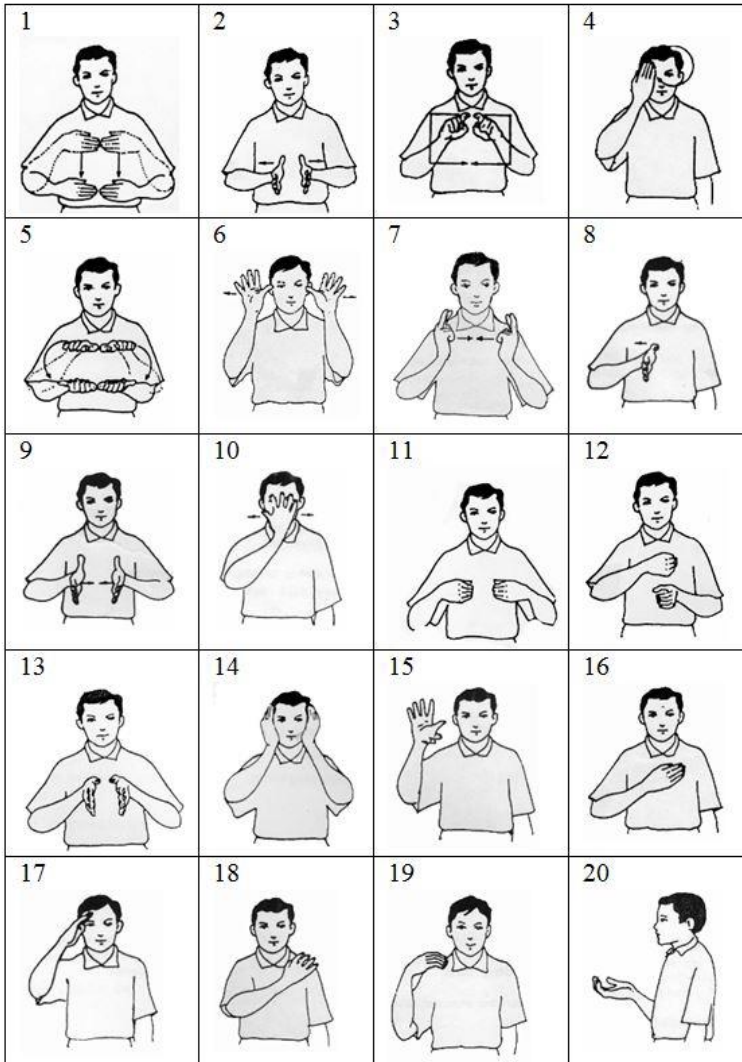
Dari ke 20 bahasa isyarat diatas, dibagi menjadi 2 jenis yaitu bahas isyarat statis dan bahasa isyarat dinamis. Bahasa isyarat statis yang berisikan Ada, Botol, Gang, Geledeg, Hai, Hamba, Hormat, Jendral, Ketua, dan Wadah. Bahasa isyarat dinamis terdiri dari Badan, Besar, Bingkai, Bingung, Bola, Kijang, Rujuk, Samping, Sempit, dan Topeng.

Pada tahap pengujian ini, dilakukan menggunakan dua data *training*. Penjelasan sebagai berikut :

1. Model *tree* A didapat dari bahasa isyarat statis dan dinamis yang diklasifikasikan secara langsung.
2. Model *tree* B didapat dari bahasa isyarat statis dan dinamis yang dikelompokkan terlebih dahulu, setelah itu digabungkan pada *tree* utama.

Pengujian dilakukan pada 10 gerakan bahasa isyarat statis dan 10 gerakan isyarat dinamis. Pengujian yang dilakukan sebagai berikut:

1. Pengujian skenario 1 merupakan pengujian akurasi yang dilakukan oleh penulis dengan menggunakan model *tree* A.



Gambar 5.1 Gambar Ke-20 Bahasa Isyarat yang Digunakan

2. Pengujian skenario 2 merupakan pengujian akurasi yang dilakukan oleh penulis dengan menggunakan model *tree* B.
3. Pengujian skenario 3 dan 4 merupakan pengujian akurasi yang dilakukan oleh pengguna lain menggunakan model *tree* B.

5.2.1 Pengujian Skenario 1 dan Analisis

Pada pengujian skenario 1, uji coba dilakukan sendiri oleh penulis yang mempunyai karakteristik tinggi badan 179 cm. Klasifikasi dilakukan menggunakan data yang telah diambil oleh penulis sebelumnya. Skenario dapat dilihat pada Tabel 5.1

Tabel 5.1 Skenario Pengujian 1

Nama Skenario Pengujian	Pengujian Akurasi 1
Kode	SP-1
Algoritma	Decision Tree
Model Bahasa Isyarat	Model <i>tree</i> A
Penguji	Penulis memiliki tinggi 179 cm
Prosedur Pengujian	Penulis melakukan uji coba 20 gerakan bahasa isyarat dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 58%

Hasil yang didapatkan dari skenario 1 tidak optimal. Banyak hasil klasifikasi yang tidak sesuai dengan kelas yang dituju. Detail hasil percobaan ini dapat dilihat pada Tabel 5.2.

Menurut hasil pengamatan penulis, banyak terjadi kesalahan klasifikasi diakibatkan karena kurangnya stabilitas data yang diterima. Selain data yang tidak tetap, klasifikasi bahasa isyarat statis dan dinamis yang digabungkan secara langsung mengakibatkan data yang diklasifikasi menjadi tidak sesuai.

Tabel 5.2 Hasil Uji Coba Skenario 1

		Target Kelas																			
		Badan	Besar	Bingkai	Bingung	Bola	Kijang	Rujuk	Samping	Sempit	Topeng	Ada	Botol	Gang	Geledeg	Hai	Hamba	Hormat	Jendral	Ketua	Wadah
Hasil Uji Coba	Badan	1	1	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Besar	2	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingkai	0	0	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingung	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
	Bola	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Kijang	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Rujuk	0	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
	Samping	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	2	0	0	0	0
	Sempit	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	Topeng	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
	Ada	1	0	0	0	0	0	0	0	2	0	1	0	1	0	0	0	0	0	0	0
	Botol	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0
	Gang	1	2	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	Geledeg	0	0	0	0	0	1	0	0	0	0	0	0	0	4	0	0	0	0	0	0
	Hai	0	0	0	0	0	0	0	0	0	2	0	0	0	0	3	0	0	0	0	0
	Hamba	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	2	0	0
	Hormat	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	3	0	0	0
	Jendral	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
	Ketua	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	2	0
	Wadah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	2
Akurasi (%)		20	40	40	60	100	100	0	60	100	100	20	80	20	80	60	60	60	80	40	40

Hal ini dikarenakan fitur-fitur yang dibutuhkan oleh bahasa isyarat statis dan dinamis berbeda. Beberapa faktor tersebut yang mengakibatkan hasil klasifikasi dari perangkat lunak ini kurang optimal.

5.2.2 Pengujian Skenario 2 dan Analisis

Pada pengujian skenario 2, uji coba dilakukan sendiri oleh penulis yang mempunyai karakteristik tinggi badan 179 cm. Klasifikasi dilakukan menggunakan data yang telah diambil oleh penulis sebelumnya. Skenario dapat dilihat pada Tabel 5.3.

Tabel 5.3 Skenario Pengujian 2

Nama Skenario Pengujian	Pengujian Akurasi 2
Kode	SP-2
Algoritma	Decision Tree
Model Bahasa Isyarat	Model <i>tree</i> B
Penguji	Penulis memiliki tinggi 179 cm
Prosedur Pengujian	Penulis melakukan uji coba 20 gerakan bahasa isyarat dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 89%

Hasil yang didapatkan pada skenario 2 cukup baik, seperti yang dapat dilihat pada Tabel 5.4. Kesalahan klasifikasi terjadi pada beberapa gerakan. Menurut pengamatan penulis, kesalahan klasifikasi tersebut terjadi karena posisi berdiri yang kurang tepat dan gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan.

Tabel 5.4 Hasil Uji Coba Skenario 2

		Target Kelas																			
Hasil Uji Coba		Badan	Besar	Bingkai	Bingung	Bola	Kijang	Rujuk	Samping	Sempit	Topeng	Ada	Botol	Gang	Geledeg	Hai	Hamba	Hormat	Jendral	Ketua	Wadah
	Badan	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Besar	0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingkai	0	0	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingung	0	0	0	3	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
	Bola	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Kijang	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Rujuk	0	0	0	0	0	0	3	0	2	0	0	0	0	0	0	0	0	0	0	0
	Samping	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0
	Sempit	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	Topeng	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0
	Ada	0	0	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0
	Botol	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	Gang	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
	Geledeg	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	Hai	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	Hamba	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	Hormat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0
	Jendral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
	Ketua	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
	Wadah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4
Hasil (%)		80	80	80	60	100	100	60	80	100	80	80	100	100	100	100	100	100	100	100	80

5.2.3 Pengujian Skenario 3 dan Analisis

Pada pengujian skenario 3, uji coba dilakukan oleh pengguna yang mempunyai karakteristik tinggi badan 172 cm. Klasifikasi dilakukan menggunakan data yang telah diambil oleh penulis sebelumnya. Skenario dapat dilihat pada Tabel 5.3.

Tabel 5.5 Skenario Pengujian 3

Nama Skenario Pengujian	Pengujian Akurasi 3
Kode	SP-3
Algoritma	Decision Tree
Model Bahasa Isyarat	Model <i>tree</i> B
Penguji	Pengguna memiliki tinggi 172
Prosedur Pengujian	Pengguna melakukan uji coba 20 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 81%

Hasil yang didapatkan pada skenario 3 cukup baik namun akurasi yang didapatkan lebih rendah dibandingkan dengan skenario 2, seperti yang dapat dilihat pada Tabel 5.6.

Berdasarkan skenario 3 yang telah dilakukan, terdapat lebih banyak gerakan yang kurang sesuai dengan yang diinginkan. Dapat dilihat pada gerak bahasa isyarat “Ada” memiliki tingkat akurasi yang kurang yaitu 2 dari 5 kali uji coba. Dari percobaan ini, menurut pengamatan penulis, kesalahan terjadi dikarenakan perbedaan postur tubuh penguji yang cukup berbeda dan kurang terbiasanya penguji dalam melakukan gerak bahasa isyarat ini.

Tabel 5.6 Hasil Uji Coba Skenario 3

		Target Kelas																			
		Badan	Besar	Bingkai	Bingung	Bola	Kijang	Rujuk	Samping	Sempit	Topeng	Ada	Botol	Gang	Geledeg	Hai	Hamba	Hormat	Jendral	Ketua	Wadah
Hasil Uji Coba	Badan	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Besar	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingkai	0	0	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingung	0	0	0	3	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
	Bola	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Kijang	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Rujuk	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	Samping	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
	Sempit	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
	Topeng	0	0	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
	Ada	0	0	0	0	0	0	0	0	0	0	2	2	1	0	0	0	0	0	0	0
	Botol	0	0	0	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0
	Gang	0	0	0	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0	0
	Geledeg	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	Hai	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	Hamba	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	Hormat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0
	Jendral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
Ketua	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3	
Wadah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3	
Akurasi (%)		80	100	60	60	80	80	100	100	100	60	40	80	80	100	100	100	80	100	60	60

5.2.4 Pengujian Skenario 4 dan Analisis

Pada pengujian skenario 4, uji coba dilakukan oleh pengguna yang mempunyai karakteristik tinggi badan 161 cm. Klasifikasi dilakukan menggunakan data yang telah diambil oleh penulis sebelumnya. Skenario dapat dilihat pada Tabel 5.7.

Tabel 5.7 Skenario Pengujian 4

Nama Skenario Pengujian	Pengujian Akurasi 4
Kode	SP-4
Algoritma	Decision Tree
Model Bahasa Isyarat	Model <i>tree</i> B
Penguji	Pengguna memiliki tinggi 161 cm
Prosedur Pengujian	Pengguna melakukan uji coba 20 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 81%

Hasil yang didapatkan pada skenario 4 cukup baik. Walaupun akurasi yang didapatkan lebih rendah dibandingkan dengan skenario 2, seperti yang dapat dilihat pada Tabel 5.8.

Menurut pengamatan penulis, hasil klasifikasi yang lebih rendah terjadi karena posisi dan gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan serta belum terbiasanya penguji menggunakan perangkat lunak yang dibangun. Pada uji coba ini juga dapat dilihat bahwa hasil gerak bahasa isyarat “Wadah” hanya memiliki benar 1 dari 5 kali pengujian, menandakan hasil klasifikasi wadah yang kurang baik.

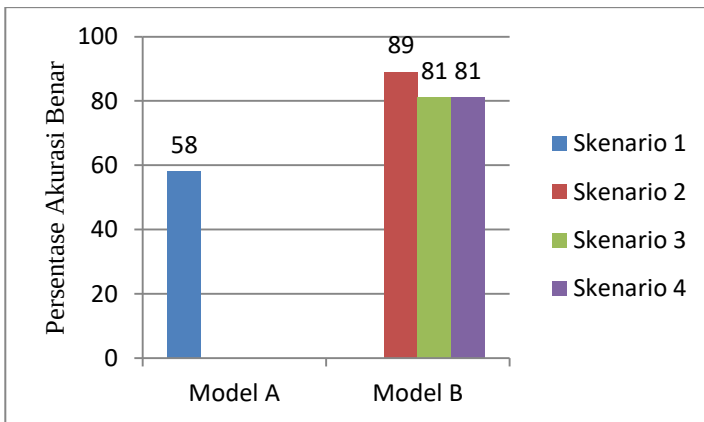
Tabel 5.8 Hasil Uji Coba Skenario 4

		Target Kelas																			
		Badan	Besar	Bingkai	Bingung	Bola	Kijang	Rujuk	Samping	Sempit	Topeng	Ada	Botol	Gang	Geledag	Hai	Hamba	Hormat	Jendral	Ketua	Wadah
Hasil Uji Coba	Badan	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Besar	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingkai	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Bingung	0	0	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
	Bola	1	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Kijang	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Rujuk	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
	Samping	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
	Sempit	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
	Topeng	0	0	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
	Ada	0	0	0	0	0	0	0	0	0	0	3	1	1	0	0	0	0	0	0	0
	Botol	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
	Gang	0	0	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0
	Geledag	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
	Hai	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
	Hamba	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
	Hormat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0
Jendral	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	
Ketua	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
Wadah	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	0	1
Akurasi (%)		80	100	80	40	40	100	100	100	80	60	60	100	80	100	100	100	80	100	100	20

5.3 Evaluasi

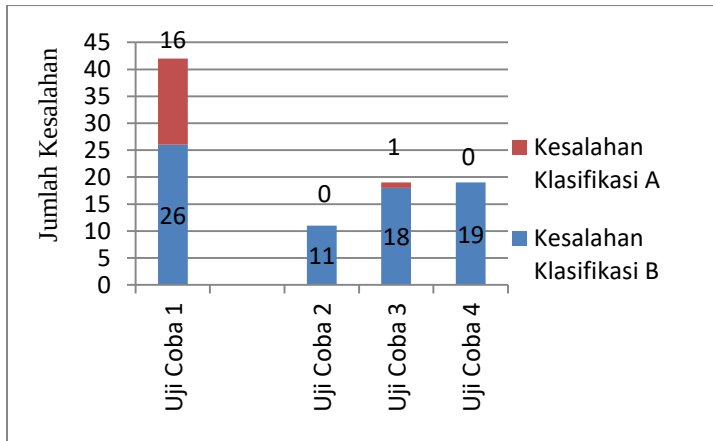
Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Evaluasi disampaikan dalam bentuk analisis hasil secara keseluruhan sebagai berikut:

1. Klasifikasi fitur statis dan dinamis menggunakan model *tree* B (uji coba 2 – 4) di dalam perangkat lunak yang dibuat mempunyai akurasi rata-rata 83.67%. Grafik hasil akurasi dari masing-masing percobaan dapat dilihat pada Gambar 5.2.



Gambar 5.2 Hasil Uji Coba

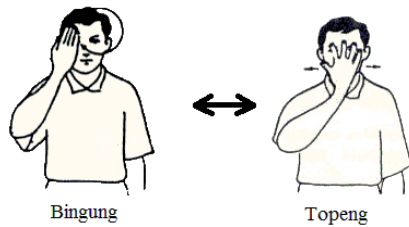
2. Pada percobaan skenario 1 dapat dilihat hasil dari klasifikasi kurang memuaskan. Dari hal ini dilakukan peningkatan dengan cara mengelompokkan data *training* terlebih dahulu. Dapat dilihat dari hasil skenario 2 bahwa hasil yang didapat lebih optimal.
3. Pada hasil uji coba menggunakan model B dapat dilihat hasil klasifikasi penentuan antara gerakan statis dan dinamis telah lebih baik daripada model A. Hasil dari percobaan ini dapat dilihat pada Gambar 5.3



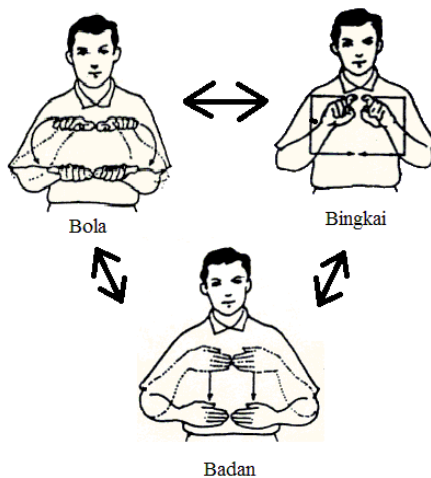
Gambar 5.3 Kesalahan klasifikasi

Kesalahan Klasifikais A adalah kesalahan klasifikasi bahasa isyarat statis menjadi dinamis atau sebaliknya. Kesalahan Klasifikasi B adalah kelasahan klasifikasi bahasa isyarat statis dengan bahasa isyarat statis lainnya atau bahasa isyarat dinamis dengan bahasa isyarat dinamis lainnya. Dari data diatas dapat di simpulkan dari hasil klasifikasi Model B (uji coba 2-4), memiliki tingkat akurasi penentuan antara gerakan bahasa isyarat statis dan dinamis sebesar 97.96%.

4. Terdapat beberapa gerakan bahasa isyarat yang memiliki kemiripan posisi gerak dan model gerakan. Seperti gerakan “Bingung” dan “Topeng” yang dapat dilihat pada Gambar 5.4, kedua gerakan ini memiliki posisi gerak pada area 1 (area kepala). Ada juga gerakan “Badan”, “Bingkai” dan “Bola” yang dapat dilihat pada Gambar 5.5, ketiga gerakan ini memiliki posisi gerak yang menyerupai dan memiliki arah gerak menuju kebawah. Hal ini dapat mengakibatkan kesalahan pada hasil klasifikasi.

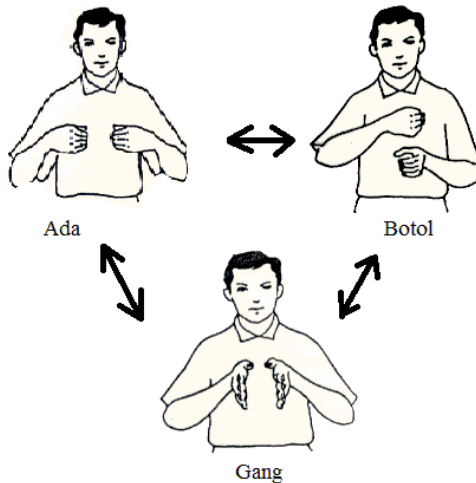


Gambar 5.4 Kemiripan Gerakan Isyarat “Bingung” dan “Topeng”



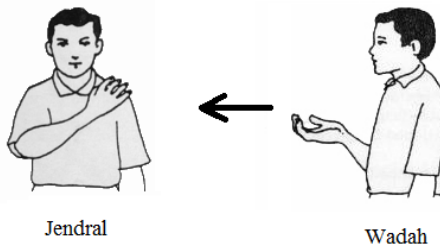
Gambar 5.5 Kemiripan Gerakan Isyarat “Bola”, “Bingkai” dan “Badan”

5. Ada beberapa gerakan bahasa isyarat statis yang juga memiliki kemiripan posisi. Contoh pada bahasa isyarat “Gang” ,“Ada” dan “Botol” dapat dilihat pada Gambar 5.6, ketiga gerakan ini memiliki posisi yang dekat dan menyerupai, sehingga terjadi beberapa kesalahan pengartian pada saat uji coba.



Gambar 5.6 Kemiripan Gerak Isyarat “Ada”, “Botol” dan “Gang”

6. Bahasa isyarat statis “Wadah” memiliki tingkat akurasi yang sangat kurang, banyak kesalahan pengartian menjadi “Jendral” seperti pada Gambar 5.7. Menurut pengamatan penulis penggunaan klasifikasi decision tree yang mengakibatkan pengklasifikasikan yang kurang akurat. Dalam perangkat lunak ini menggunakan decision tree sehingga memungkinkan sebuah fitur dianggap lebih penting dibandingkan dengan fitur yang lainnya. Hal ini dapat mengurangi tingkat akurasi dari klasifikasi perangkat lunak.



Gambar 5.7 Gerak Bahasa Isyarat “Jendral” dan “Wadah”

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari proses pengerjaan selama perancangan, implementasi, dan proses pengujian aplikasi yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Fitur-fitur yang digunakan dalam perangkat lunak ini berjumlah 59 fitur. Fitur yang digunakan dalam proses ekstraksi fitur statis ada 21 fitur yang digunakan yaitu 7 fitur vektor 2D, 6 fitur *angle*, dan 1 fitur *distance*. Fitur yang digunakan dalam proses ekstraksi fitur dinamis berjumlah 38 fitur yaitu 18 fitur awal merupakan ekstraksi fitur dinamis tangan kiri, 18 fitur berikutnya merupakan ekstraksi fitur dinamis tangan kanan, dan 2 fitur untuk menentukan posisi tangan kiri dan kanan. Semua fitur tersebut didapatkan dari hasil olah data koordinat 11 *skeleton joints* yang ditentukan oleh penulis, yaitu :
 - a. Leher (vector N)
 - b. Bahu tulang belakang (Vektor SS)
 - c. Tulang belakang tengah (Vektor SM)
 - d. Bahu tangan kanan (Vektor SR)
 - e. Siku tangan kanan (Vektor ER)
 - f. Pergelangan tangan kanan (Vektor WR)
 - g. Telapak Tangan kanan (Vektor HR)
 - h. Bahu tangan kiri (Vektor SL)
 - i. Siku tangan kiri (Vektor EL)
 - j. Pergelangan tangan kiri (Vektor WL)
 - k. Telapak tangan kiri (Vektor HL)

2. Ketepatan posisi berdiri dan melakukan gerakan bahasa isyarat yang dilakukan ketika mengambil data *training* dan *testing* sangat berpengaruh terhadap akurasi klasifikasi fitur.
3. Perbedaan karakteristik tinggi badan berpengaruh terhadap penentuan koordinat *skeleton joints* dimana perbedaan tersebut mempengaruhi identifikasi posisi gerakan yang dilakukan oleh perangkat lunak.
4. Perangkat lunak yang dibangun pada Tugas Akhir ini dapat menerjemahkan bahasa isyarat pokok dengan akurasi rata-rata 83.67% dan ketepatan akurasi klasifikasi statis dan dinamis sebesar 97.96%.

6.2 Saran

Berikut saran-saran untuk pengembangan dan perbaikan sistem di masa yang akan datang. Di antaranya adalah sebagai berikut:

1. Memperbanyak data *training* guna meningkatkan akurasi dari klasifikasi perangkat lunak ini.
2. Identifikasi *skeleton joints* tidak hanya posisi tangan saja. Karena ada beberapa gerakan bahasa isyarat memiliki kesamaan dalam gerakan atau posisi namun memiliki bentuk atau model tangan yang berbeda.
3. Melakukan normalisasi data koordinat masing-masing *skeleton joints* sebelum dilakukan proses ekstraksi fitur agar mendapatkan hasil lebih akurat.
4. Menggunakan *classifier* yang lebih baik dapat meningkatkan akurasi perangkat lunak ini.
5. Fitur yang digunakan sebaiknya ditambahkan menjadi vektor 3D untuk meningkatkan akurasi hasil klasifikasi perangkat lunak.

DAFTAR PUSTAKA

- [1] N. Sugianto and F. Samopa, "Analisa Manfaat dan Penerimaan Terhadap Implementasi Bahasa Isyarat Indonesia pada Latar Belakang Komplek Menggunakan Kinect dan Jaringan Syaraf Tiruan (Studi Kasus SLB Karya Mulia 1)," *JUISI*, vol. 1, pp. 56-72, 2015.
- [2] Y. A. Sutanto, "Rancang Bangun Modul Pengenalan Bahasa Isyarat Indonesia Menggunakan Teknologi Kinect dan Metode Back Propagation Genetic Algorithm Neural Network," Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [3] Y. E. Nugyasa, "Ekstraksi Fitur Dinamis Pada Gerakan Tangan Menggunakan Kinect 2.0 Untuk Mengenali Bahasa Isyarat Indonesia," Institut Teknologi Sepuluh Nopember, Surabaya, 2017.
- [4] "Rick Hansen Foundation," [Online]. Available: <https://www.rickhansen.com/portals/0/WhatWeDo/SchoolProgram/Disability%20images/ASL.jpg>. [Accessed 7 June 2017].
- [5] "Kinect," Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Kinect>. [Accessed 7 June 2016].
- [6] "Developing with Kinect for Windows," Microsoft, [Online]. Available: <https://developer.microsoft.com/en-us/windows/kinect/develop>. [Accessed 7 June 2017].
- [7] A. W. Yanuardi, S. Prasetyo and P. P. Johannes Adi, "Indonesian Sign Language Computer Application for the Deaf," in *International Conference on Education Technology and Computer*, Shanghai, 2010.
- [8] M. Iqbal, "Pengenalan Bahasa Isyarat Indonesia Berbasis Sensor Flex dan Accelerometer Menggunakan Dynamic Time Warping," Institut Teknologi Sepuluh Nopember,

Surabaya, 2011.

- [9] Y. Chen, B. Luo, Y.-L. Chen, G. Liang and X. Wu, "A Real-time Dynamic Hand Gesture Recognition System Using Kinect Sensor," in *IEEE Conference on Robotics and Biomimetics*, Zhuhai, 2015.
- [10] "Microsoft Visual Studio," Wikipedia, [Online]. Available: http://id.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 7 Juny 2016].
- [11] "vvvv - a multipurpose toolkit," [Online]. Available: <https://vvvv.org/documentation/kinect>. [Accessed 7 June 2017].
- [12] C. Sun, T. Zhang, B.-K. Bao and C. Xu, "Discriminative Exemplar Coding for Sign Language," in *IEEE TRANSACTIONS ON CYBERNETICS*, 2013.

LAMPIRAN A KODE SUMBER

1	private void OneTestButton_Click(object sender,
2	RoutedEventArgs e)
3	{
4	statusDetail.Content = "Testing Data";
5	statusAmbil = 2;
6	flag2 = 0;
7	flag3 = 0;
8	i = 0;
9	InitStaticFeatures();
	}

Kode Sumber A.1 Fungsi Create File

1	private void createButton_click(object sender,
2	RoutedEventArgs e)
3	{
4	statusDetail.Content = "Create Dataset";
5	statusAmbil = 1;
6	flag2 = 0;
7	flag3 = 0;
8	i = 0;
9	InitStaticFeatures();
10	namaGerakan = fileName.Text;
11	}

Kode Sumber A.2 Fungsi Start Testing

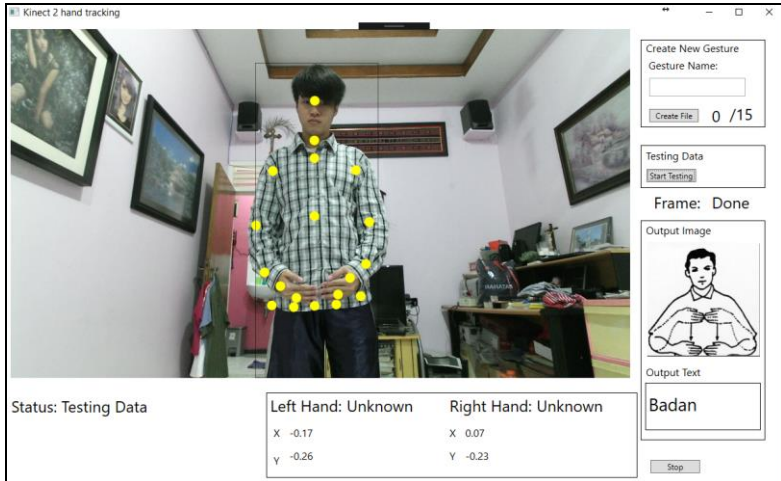
1	private void stopButton_Click(object sender, RoutedEventArgs
2	e)
3	{
4	statusDetail.Content = "Idle";
5	statusAmbil = 0;
6	flag2 = 0;
7	ambilData.Content = "Done";
8	outputText.Content = "";
9	outputImage.Source = null;
	}

Kode Sumber A.3 Fungsi Stop

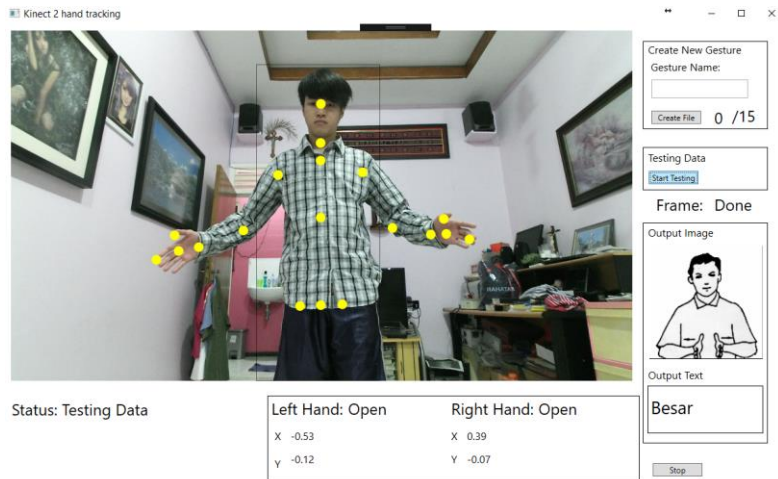
[Halaman ini sengaja dikosongkan]

LAMPIRAN B

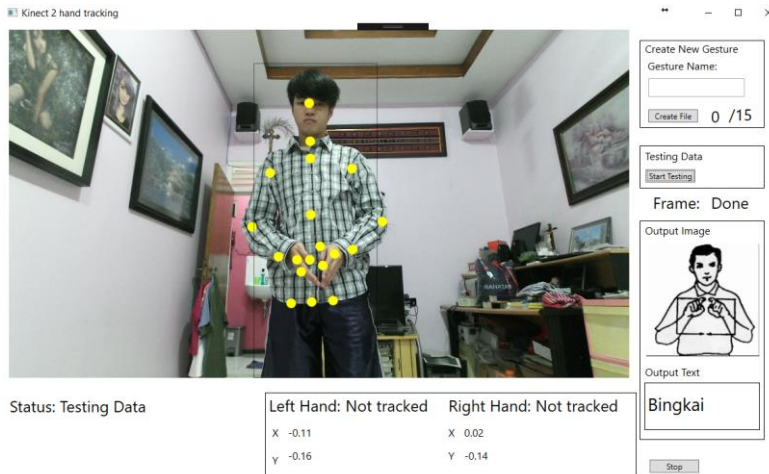
SCREENSHOT PERANGKAT LUNAK



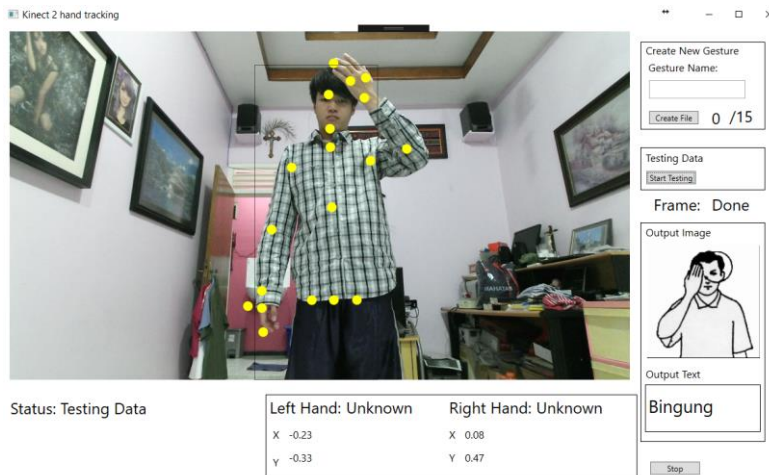
Gambar B.1 Penulis Melakukan Gerak Bahasa Isyarat Badan



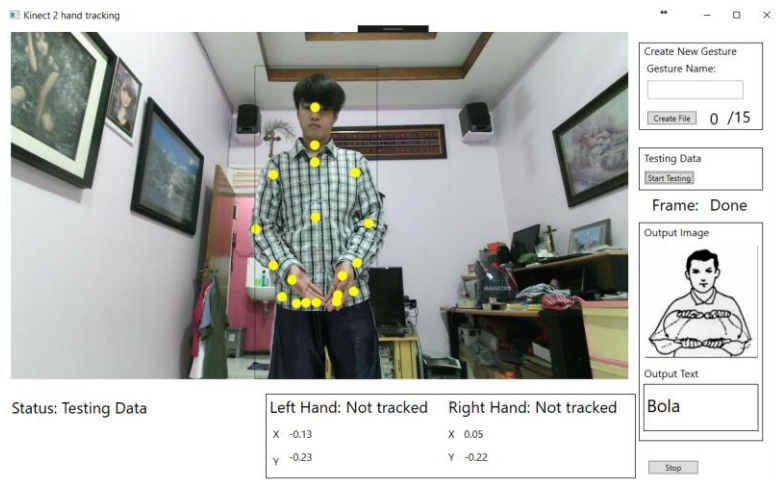
Gambar B.2 Penulis Melakukan Gerak Bahasa Isyarat Besar



Gambar B.3 Penulis Melakukan Gerak Bahasa Isyarat Bingkai



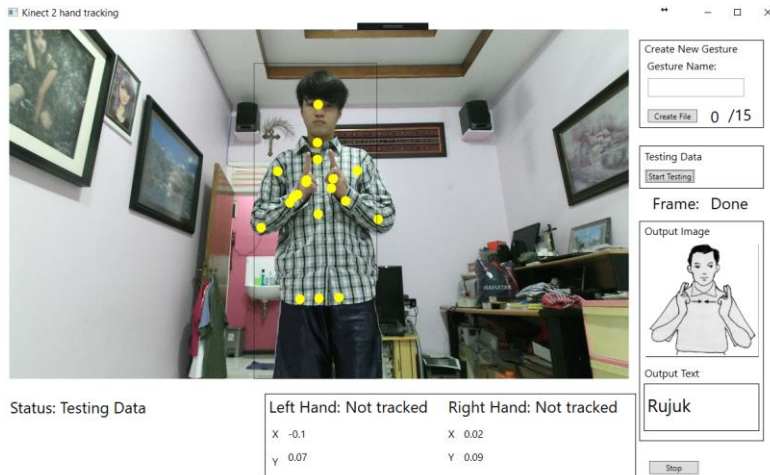
Gambar B.4 Penulis Melakukan Gerak Bahasa Isyarat Bingung



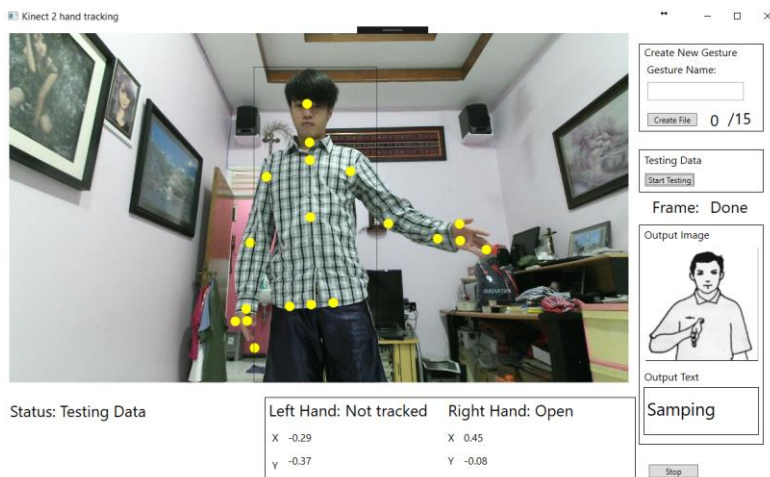
Gambar B.5 Penulis Melakukan Gerak Bahasa Isyarat Bola



Gambar B.6 Penulis Melakukan Gerak Bahasa Isyarat Kijang



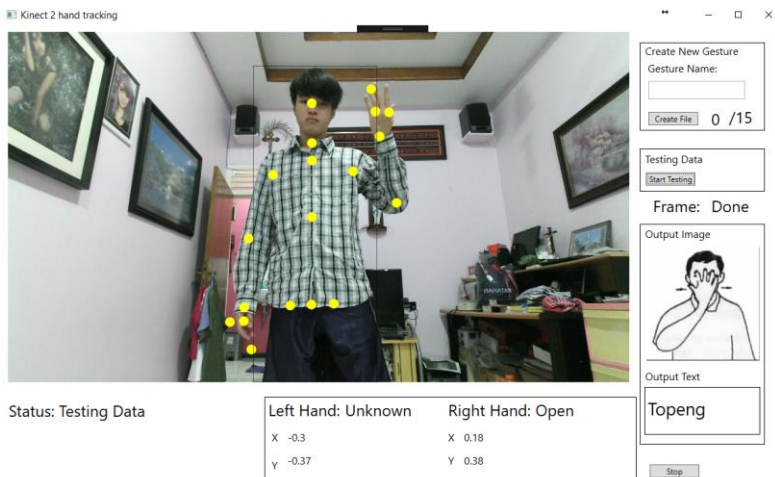
Gambar B.7 Penulis Melakukan Gerak Bahasa Isyarat Rujuk



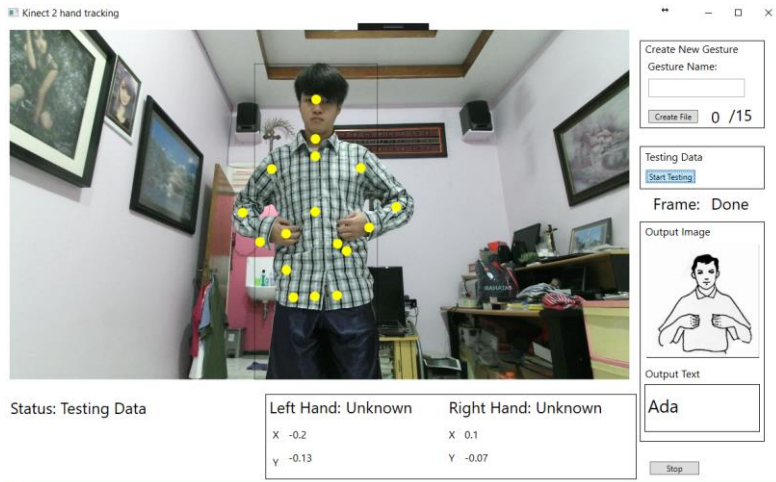
Gambar B.8 Penulis Melakukan Gerak Bahasa Isyarat Samping



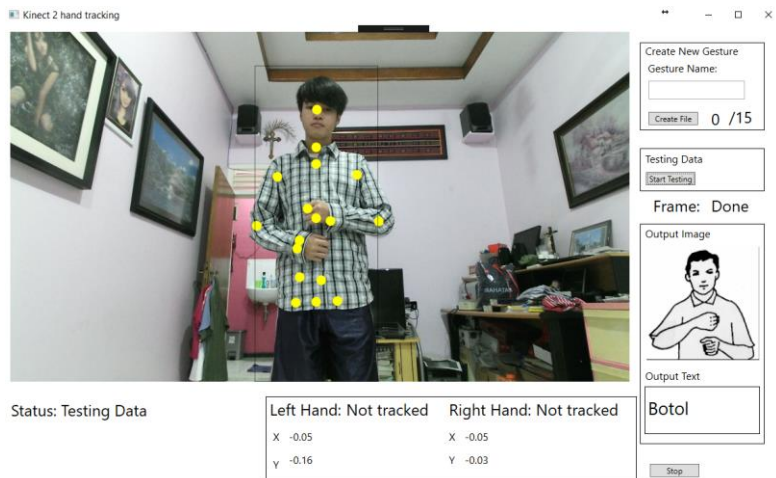
Gambar B.9 Penulis Melakukan Gerak Bahasa Isyarat Sempit



Gambar B.10 Penulis Melakukan Gerak Bahasa Isyarat Topeng



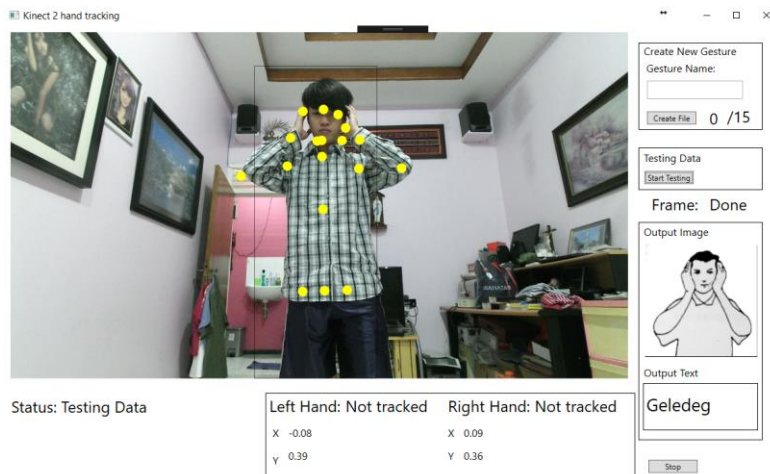
Gambar B.11 Penulis Melakukan Gerak Bahasa Isyarat Ada



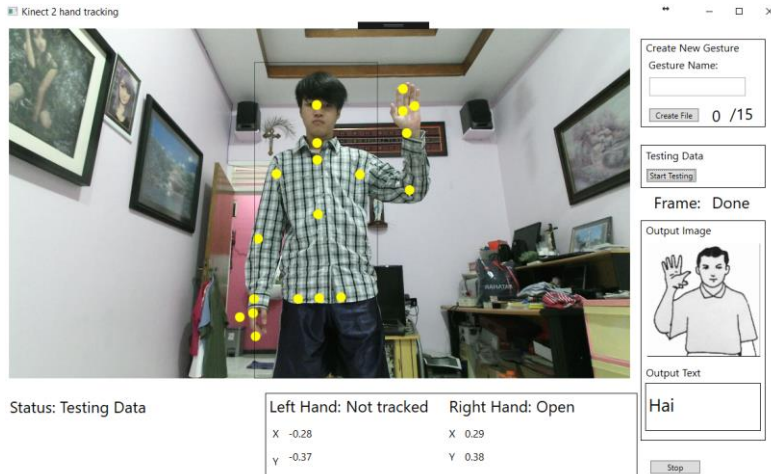
Gambar B.12 Penulis Melakukan Gerak Bahasa Isyarat Botol



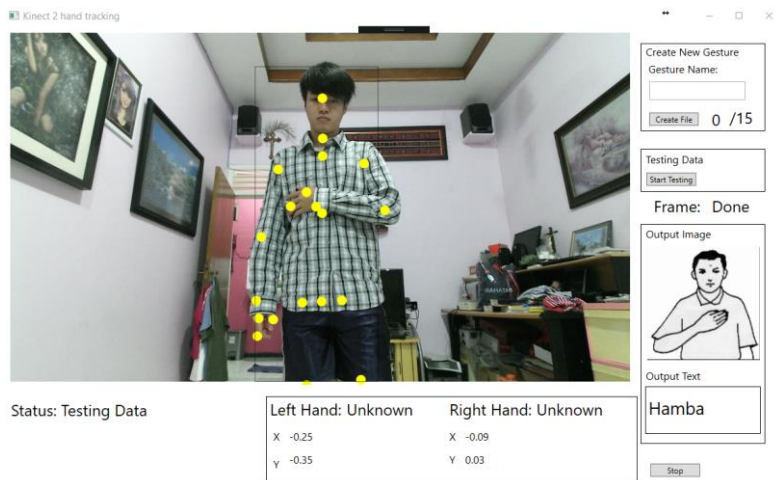
Gambar B.13 Penulis Melakukan Gerak Bahasa Isyarat Gang



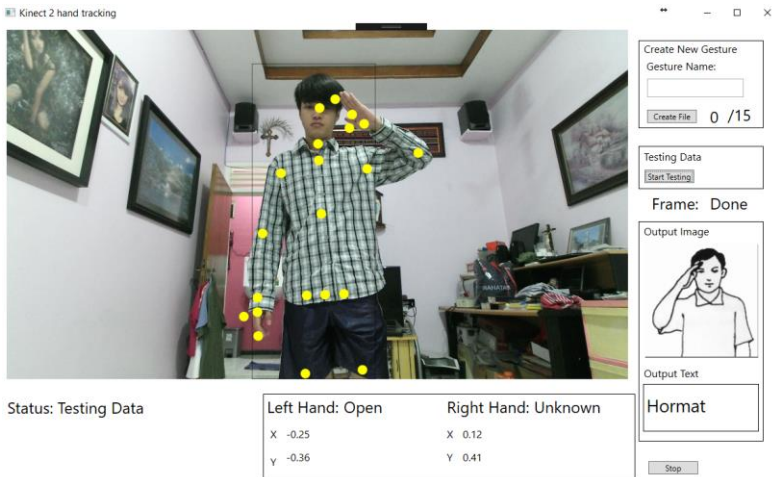
Gambar B.14 Penulis Melakukan Gerak Bahasa Isyarat Geledeg



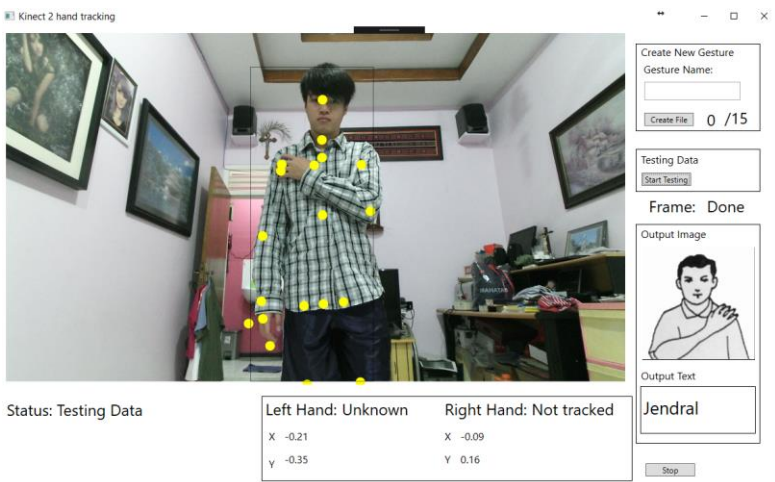
Gambar B.15 Penulis Melakukan Gerak Bahasa Isyarat Hai



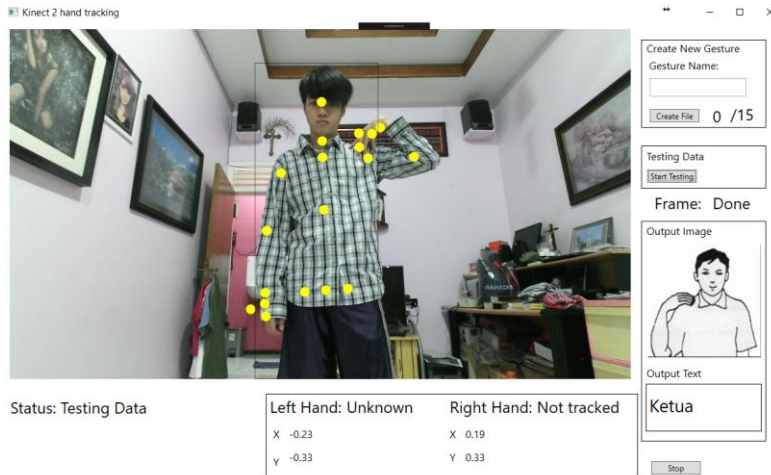
Gambar B.16 Penulis Melakukan Gerak Bahasa Isyarat Hamba



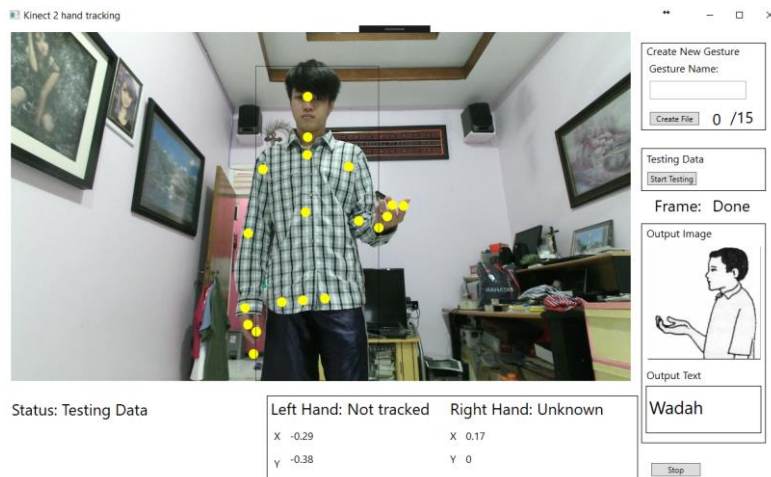
Gambar B.17 Penulis Melakukan Gerak Bahasa Isyarat Hormat



Gambar B.18 Penulis Melakukan Gerak Bahasa Isyarat Jendral



Gambar B.19 Penulis Melakukan Gerak Bahasa Isyarat Ketua



Gambar B.20 Penulis Melakukan Gerak Bahasa Isyarat Wadah



Create New Gesture
Gesture Name:

Create File 0 /15

Testing Data
Start Testing

Frame: Done

Output Image


Output Text

Hai

Stop

Status: Testing Data

Left Hand: Not tracked	Right Hand: Open
X -0.3	X 0.24
Y -0.37	Y 0.46

Gambar B.21 Penguji 1 Melakukan Gerak Bahasa Isyarat Hai



Create New Gesture
Gesture Name:

Create File 0 /15

Testing Data
Start Testing

Frame: Done

Output Image


Output Text

Gang

Stop

Status: Testing Data

Left Hand: Unknown	Right Hand: Unknown
X -0.23	X 0
Y -0.27	Y -0.25

Gambar B.22 Penguji 2 Melakukan Gerak Bahasa Isyarat Gang

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis lahir di kota Surabaya pada 05 September 1995. Merupakan anak tunggal yang mempunyai hobi bermain game, mendaki gunung dan fotografi. Penulis telah menempuh pendidikan formal yaitu TKK Maria Regina (2000 – 2001), SDK Maria Regina (2001 – 2007), SMPK Carolus (2007 – 2010), SMAK Frateran (2010 – 2013), dan mahasiswa S1 Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember Surabaya rumpun mata kuliah Interaksi, Grafika, dan Seni (IGS).

Penulis pernah mengikuti beberapa organisasi dan kepanitiaan yaitu ITS EXPO sebagai staf (2015), dan GERIGI ITS sebagai *Organizing Committee* (2014) dan *Instructor Committee* (2015). Penulis dapat dihubungi melalui surel inyas.benedict@gmail.com.