

TUGAS AKHIR - KI141502

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

YAHYA EKA NUGYASA
NRP 5113100134

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

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

RBTC



TUGAS AKHIR - KI141502

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

YAHYA EKA NUGYASA
NRP 5113100134

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

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]

RBTC



FINAL PROJECT - KI141502

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

YAHYA EKA NUGYASA
NRP 5113100134

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

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

[Halaman ini sengaja dikosongkan]

RBTC

LEMBAR PENGESAHAN

EKSTRAKSI FITUR 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 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

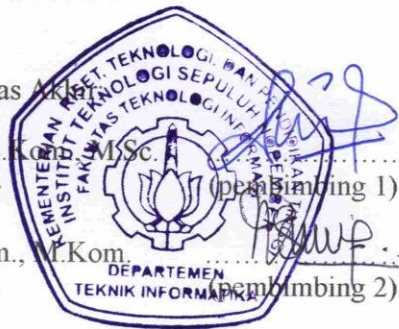
Oleh:

YAHYA EKA NUGYASA

NRP. 5113100134

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
JULI, 2017

[Halaman ini sengaja dikosongkan]

RBTC

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

Nama Mahasiswa : Yahya Eka Nugyasa
NRP : 5113100134
Jurusan : Teknik Informatika FTIf-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 merupakan media bagi penderita tuna rungu dan tuna wicara untuk berkomunikasi dengan sekitar. Mengacu pada Sistem Isyarat Bahasa Indonesia (SIBI), bahasa isyarat diklasifikasikan dalam dua kategori, yaitu bahasa isyarat statis dan dinamis.

Pada tugas akhir sebelumnya, penggunaan teknologi terkini guna mendukung pembelajaran pemakaian bahasa isyarat Indonesia sudah dilakukan dengan menggunakan perangkat Kinect 1.0, namun hanya sebatas bahasa isyarat dengan gerakan statis saja. Tugas Akhir ini menggunakan perangkat Kinect 2.0 dalam mendeteksi gerakan dan fitur dinamis untuk menentukan bahasa isyarat apa yang dimaksud oleh pengguna berdasarkan gerakan tangan yang dilakukan.

Hasil pengujian dalam Tugas Akhir ini menunjukkan bahwa metode Decision Tree yang digunakan sebagai classifier gerakan isyarat dinamis mempunyai akurasi yang baik yaitu sekitar 82,9 persen. Hasil tersebut masih dapat ditingkatkan dengan menambahkan data training maupun menggunakan/membuat classifier lain dalam mengklasifikasikan bahasa isyarat dinamis.

Kata kunci: Kinect 2, Fitur Dinamis, Bahasa Isyarat, SIBI.

[Halaman ini sengaja dikosongkan]

RBTC

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

Name : Yahya Eka Nugyasa
NRP : 5113100134
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 tool for deaf and speech impaired person to communicate with people around. Referring to the Indonesian sign language system (SIBI), sign language is classified into two categories, namely static and dynamic sign language.

In the previous final project, the use of the latest technology to learning of Indonesian sign language has been done using Kinect 1.0 device, but only limited to sign language with a static movement only. This final project uses Kinect 2.0 devices in motion detection and the dynamic features to determine what sign language the user is referring based on the hand movements performed.

The test results of this final project show that the Decision Tree method used as classifier have a good accuracy which is about 82,9 percent. These results can be improved by adding training data or using/creating another classifier in classifying dynamic sign language.

Keywords: : Kinect 2, Dynamic Features, Sign Language, SIBI

[Halaman ini sengaja dikosongkan]

RBTC

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ-

Puji syukur penulis panjatkan kehadiran Allah SWT karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “Ekstraksi Fitur Dinamis pada Gerakan Tangan Menggunakan Kinect 2.0 untuk Mengenali Bahasa Isyarat Indonesia”. Penyusunan tugas akhir ini diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana di Departemen Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan dan penulisan tugas akhir ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Bapak Usman Gunawan, Ibu Maya Susanti, Nabilah Dwi Gunasti, dan 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 Teknik Informatika ITS yang banyak memberikan ilmu serta bimbingan selama masa perkuliahan.
4. Saudara-saudari Edelweis 98 Angkatan XXIII yang tetap loyal menemani kesuntukan dalam menjalani masa perkuliahan di tanah rantau.
5. Teman-teman “Tidur Sek” yang selalu memberikan canda dan tawa dikala suntuk ketika masih aktif menjadi anggota himpunan.

6. Teman-teman “Piaraan” yang tidak hentinya memberi warna di tahun terakhir perkuliahan.
7. Teman-teman Pengurus Harian HMTTC Optimasi yang memberikan banyak pelajaran ketika bersama-sama berusaha membangun HMTTC menjadi lebih baik.
8. Seluruh teman-teman TC13 yang sangat sering saya repotkan di Surabaya.
9. Adik-adik perwalian yang selalu memberikan dorongan kepada untuk segera menyelesaikan tugas akhir.
10. Teman-teman Kabinet BEM ITS Wahana Juang 2016/2017 yang telah membuka lembaran baru di tahun terakhir dalam dunia organisasi.
11. *Squad* Perekonomian BEM ITS Wahana (J)uang yang selalu haus akan ilmu serta wawasan khususnya di bidang perekonomian.
12. Pihak-pihak lain yang turut memberikan coretan tinta kehidupan selama menempuh pendidikan di kampus perjuangan yang tidak dapat disebutkan satu persatu disini, baik itu teman-teman antar angkatan, antar jurusan, teman-teman kepanitiaan, dan yang lainnya.

Semoga Allah SWT memberikan balasan yang berlipat ganda kepada semuanya. Kritik dan saran yang membangun penulis terima dengan senang hati agar tugas akhir ini dapat bermanfaat khususnya bagi penulis dan umumnya bagi kita semua.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
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	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	4
1.7 Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1 Tunarungu	7
2.2 Bahasa Isyarat	7
2.3 Kinect 2.0	8
2.4 Kinect SDK	10
2.5 Microsoft Visual Studio	10
2.6 Ekstraksi Fitur Dinamis dari Hand Gesture.....	10
2.7 Decision Tree	12
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	17
3.1 Analisis Perangkat Lunak.....	17
3.1.1 Deskripsi Umum Perangkat Lunak	18
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak.....	18
3.1.3 Identifikasi Pengguna	19
3.2 Perancangan Perangkat Lunak	20
3.2.1 Model Kasus Penggunaan	20
3.2.2 Definisi Aktor.....	21
3.2.3 Definisi Kasus Penggunaan.....	21

3.2.4 Arsitektur Umum Sistem	25
3.2.5 Rancangan Antarmuka Perangkat Lunak	25
3.2.6 Rancangan Proses Perangkat Lunak.....	26
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	39
4.3.1 Implementasi Pendeteksian Skeleton Pengguna.....	39
4.3.2 Implementasi Proses Ekstraksi Fitur Dinamis dan Menentukan Posisi Gerakan Tangan	42
4.3.4 Implementasi Proses Testing Data	48
BAB V PENGUJIAN DAN EVALUASI	53
5.1 Lingkungan Pembangunan	53
5.2 Skenario Pengujian.....	53
5.2.1 Pengujian Skenario A1 dan Analisis	54
5.2.2 Pengujian Skenario A2 dan Analisis	55
5.2.3 Pengujian Skenario A3 dan Analisis	56
5.2.4 Pengujian Skenario B dan Analisis	60
5.3 Evaluasi	63
BAB VI KESIMPULAN DAN SARAN.....	67
6.1 Kesimpulan	67
6.2 Saran	68
DAFTAR PUSTAKA.....	69
LAMPIRAN A KODE SUMBER.....	71
LAMPIRAN B SCREENSHOT PERANGKAT LUNAK.....	77
BIODATA PENULIS.....	89

DAFTAR GAMBAR

Gambar 1.1 Contoh Pemakaian Bahasa Isyarat Statis yang Mempunyai Arti: (1) Baik; (2) Demam; (3) Maaf	2
Gambar 1.2 Contoh Pemakaian Bahasa Isyarat Dinamis yang Mempunyai Arti: (1) Awan; (2) Adik; (3) Agama	2
Gambar 1.3 Alur Aplikasi yang Dibuat.....	4
Gambar 2.1 Bahasa Isyarat yang Digunakan	9
Gambar 2.2 Kinect 2.0	10
Gambar 2.3 Pengelompokkan Data Kuantisasi (A) dan Contoh Hasil Kuantisasi (B)	11
Gambar 3.1 Diagram Kasus Penggunaan Perangkat Lunak.....	20
Gambar 3.2 Arsitektur Umum Sistem.....	25
Gambar 3.3 Rancangan Antarmuka Perangkat Lunak	26
Gambar 3.4 Diagram Alir Proses Ekstraksi Fitur Dinamis dan Proses Penentuan Posisi Gerakan Tangan	27
Gambar 3.5 Posisi Gerakan Tangan pada Tubuh Mengacu pada Skeleton Leher (N) dan Bagian Tengah Tulang Belakang (SM)	31
Gambar 3.6 Rancangan Proses Training Data.....	31
Gambar 3.7 Rancangan Proses Testing Data	32
Gambar 3.8 Decision Tree yang Didapatkan dari Hasil Klasifikasi WEKA	33
Gambar 4.1 Antarmuka Perangkat Lunak	36
Gambar 5.1 Bahasa Isyarat Statis yang Berarti Gang	61
Gambar 5.2 Grafik Tingkat Akurasi Klasifikasi Hasil Uji Coba	63
Gambar 5.3 Kemiripan Gerakan dari Beberapa Bahasa Isyarat ..	64
Gambar B.1 Penulis Mempraktekkan Bahasa Isyarat Topeng	77
Gambar B.2 Penulis Mempraktekkan Bahasa Isyarat Badan	77
Gambar B.3 Penulis Mempraktekkan Bahasa Isyarat Awan.....	78
Gambar B.4 Penulis Mempraktekkan Bahasa Isyarat Anak.....	78
Gambar B.5 Penulis Mempraktekkan Bahasa Isyarat Maklum...	79
Gambar B.6 Penulis Mempraktekkan Bahasa Isyarat Bola.....	79
Gambar B.7 Penulis Mempraktekkan Bahasa Isyarat Besar	80
Gambar B.8 Penulis Mempraktekkan Bahasa Isyarat Rujuk	80
Gambar B.9 Penulis Mempraktekkan Bahasa Isyarat Bingkai....	81

Gambar B.10 Penulis Mempraktekkan Bahasa Isyarat Bingung.	81
Gambar B.11 Penulis Mempraktekkan Bahasa Isyarat Selubung	82
Gambar B.12 Penulis Mempraktekkan Bahasa Isyarat Gelombang	82
Gambar B.13 Penulis Mempraktekkan Bahasa Isyarat Lengkung	83
Gambar B.14 Penulis Mempraktekkan Bahasa Isyarat Samping	83
Gambar B.15 Penulis Mempraktekkan Bahasa Isyarat Kijang ...	84
Gambar B.16 Penulis Mempraktekkan Bahasa Isyarat Sempit ...	84
Gambar B.17 Penulis Mempraktekkan Bahasa Isyarat Faedah...	85
Gambar B.18 Penulis Mempraktekkan Bahasa Isyarat Sama	85
Gambar B.19 Penulis Mempraktekkan Bahasa Isyarat Gang	86
Gambar B.20 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Sama	86
Gambar B.21 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Besar	87

DAFTAR TABEL

Tabel 2.1 Data Hasil Ekstraksi Fitur Dinamis.....	12
Tabel 3.2 Definisi Kasus Penggunaan.....	21
Tabel 3.3 Definisi Kasus Penggunaan.....	21
Tabel 3.4 Spesifikasi Kasus Penggunaan Membuat Data Bahasa Isyarat Baru	22
Tabel 3.5 Spesifikasi Kasus Penggunaan Training Dataset	23
Tabel 3.6 Pembagian Fitur Data.....	31
Tabel 5.1 Skenario Pengujian A1	55
Tabel 5.3 Skenario Pengujian A2.....	55
Tabel 5.2 Terjemahan Isyarat Kata Skenario Pengujian A1	57
Tabel 5.4 Terjemahan Isyarat Kata Skenario Pengujian A2	58
Tabel 5.6 Terjemahan Isyarat Kata Skenario Pengujian A3	59
Tabel 5.5 Skenario Pengujian A3.....	60
Tabel 5.7 Skenario Pengujian B	61
Tabel 5.8 Terjemahan Isyarat Kata Skenario Pengujian B.....	62

[Halaman ini sengaja dikosongkan]

RBTC

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 Skeleton Pengguna	42
Kode Sumber 4.4 Kode Sumber Ekstraksi Fitur Dinamis dan Menentukan Posisi Gerakan Tangan dari Skeleton Pengguna	46
Kode Sumber 4.5 Kode Sumber Menyimpan Hasil Ekstraksi Fitur Dinamis dari Skeleton Pengguna	47
Kode Sumber 4.6 Implementasi Decision Tree dari Hasil Klasifikasi.....	51
Kode Sumber A.1 Fungsi Create File	71
Kode Sumber A.2 Fungsi Start Testing.....	71
Kode Sumber A.3 Decision Tree 18 Gerakan Dinamis dan 1 Gerakan Statis	75

[Halaman ini sengaja dikosongkan]

RBTC

BAB I

PENDAHULUAN

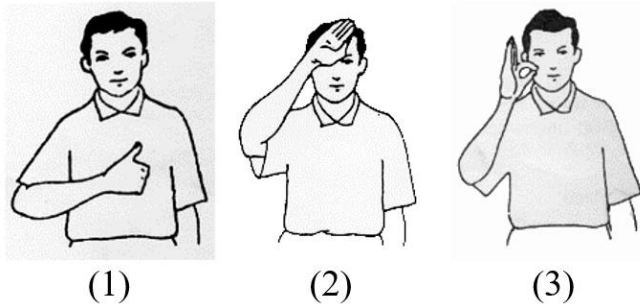
Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan, batasan permasalahan, dan manfaat.

1.1 Latar Belakang

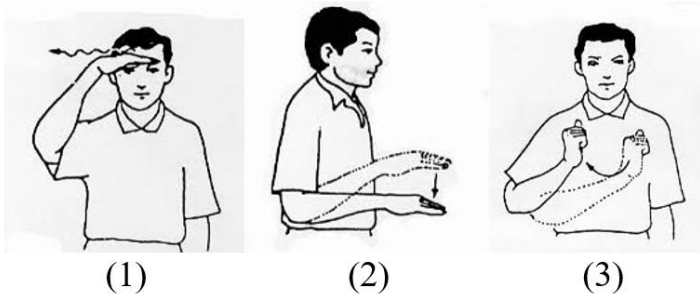
Bahasa isyarat merupakan media bagi para penderita tuna rungu dan tuna wicara untuk berkomunikasi dengan sekitarnya. Gerakan visual tubuh sangat membantu penderita agar yang ingin disampaikan lebih mudah dimengerti oleh pasangan komunikasinya [1]. Di Indonesia sendiri, para penderita tuna rungu dan tuna wicara berkomunikasi menggunakan bahasa isyarat yang mengacu pada dua sistem yaitu BISINDO (Berkenalan Dengan Sistem Isyarat Indonesia) dan SIBI (Sistem Isyarat Bahasa Indonesia). BISINDO dikembangkan oleh orang tuna rungu sendiri melalui GERKATIN (Gerakan Kesejahteraan Tuna Rungu Indonesia). SIBI dikembangkan oleh orang normal, bukan-penderita tuna rungu. SIBI sama dengan bahasa isyarat yang digunakan di Amerika yaitu ASL (*American Sign-Language*) [2].

Bahasa isyarat diklasifikasikan dalam dua kategori: statis dan dinamis. Isyarat statis adalah isyarat yang tidak melibatkan suatu pergerakan dalam penyampaianya seperti yang terlihat pada Gambar 1.1. Sedangkan isyarat dinamis adalah isyarat yang melibatkan pergerakan dalam penyampaianya seperti yang terlihat pada Gambar 1.2 [2].

Sebelumnya sudah ada Tugas Akhir yang dibuat oleh Yohanes Aditya Sutanto tentang pengenalan bahasa isyarat Indonesia menggunakan teknologi Kinect 1.0 [3]. Tetapi Tugas Akhir tersebut hanya dapat mendeteksi bahasa isyarat statis. Oleh karena itu, dengan kemampuan yang dimiliki oleh Kinect 2.0, munculah ide untuk membuat pengenalan bahasa isyarat Indonesia yang dapat mendeteksi bahasa isyarat dinamis.



Gambar 1.1 Contoh Pemakaian Bahasa Isyarat Statis yang Mempunyai Arti: (1) Baik; (2) Demam; (3) Maaf



Gambar 1.2 Contoh Pemakaian Bahasa Isyarat Dinamis yang Mempunyai Arti: (1) Awan; (2) Adik; (3) Agama

Untuk dapat mendeteksi bahasa isyarat dinamis, dibutuhkan pula fitur yang dinamis dalam memproses gerakan yang diberikan. Sehingga pada penelitian ini diusulkan suatu aplikasi yang dapat mengekstraksi fitur dinamis pada bahasa isyarat. Dengan adanya perangkat lunak ini, diharapkan dapat memudahkan komunikasi menggunakan bahasa isyarat dinamis yang digunakan oleh penderita tuna rungu dan tuna wicara dengan mengacu pada SIBI.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana mengekstraksi fitur dinamis pada gerakan tangan dalam mendeteksi bahasa isyarat?
2. Bagaimana menggunakan hasil klasifikasi gerakan tangan untuk mendeteksi bahasa isyarat?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Teknologi yang dipakai adalah Kinect 2.0.
2. Bahasa isyarat yang dideteksi hanya yang bersifat dinamis saja dengan menggunakan *sample* sebanyak 18 gerakan.
3. Klasifikasi yang digunakan untuk mengklasifikasi gerakan adalah Decision Tree.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah mengekstraksi fitur dinamis pada gerakan tangan menggunakan Kinect 2.0 untuk mengenali bahasa isyarat Indonesia.

1.5 Manfaat

Tugas Akhir ini diharapkan dapat mengenali bahasa isyarat menggunakan anggota tubuh yang bergerak (dinamis) sehingga dapat membantu orang berkebutuhan khusus yaitu tuna rungu dan tuna wicara dalam berkomunikasi dengan orang yang tidak berkebutuhan khusus.

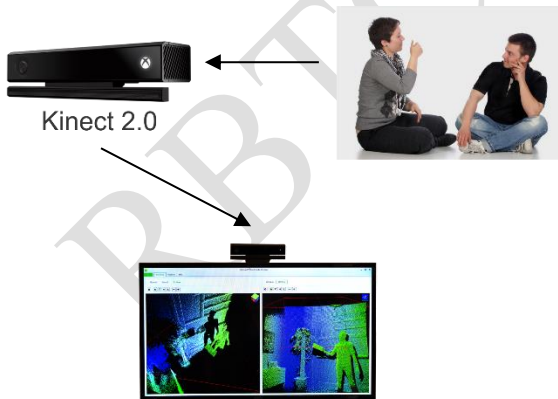
1.6 Metodologi

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.

B. Perancangan perangkat lunak



Gambar 1.3 Alur Aplikasi yang Dibuat

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 1.3.

- 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.
- 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.

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan Tugas Akhir ini. Pokok permasalahan yang akan di bahas mengenai teknologi yang mendukung dalam pembuatan tugas akhir seperti Kinect 2.0, Kinect SDK, Decision Tree, dan pengetahuan umum mengenai bahasa isyarat.

2.1 Tunarungu

Tuna rungu 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*) [4]. 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 sarana berkomunikasi bagi penderita tuna rungu. Bahasa isyarat dikembangkan dan memiliki karakteristik sendiri di berbagai negara. Di Indonesia, bahasa isyarat yang digunakan berdasarkan pada SIBI. Ada 4 jenis bahasa isyarat dalam SIBI [5] 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.

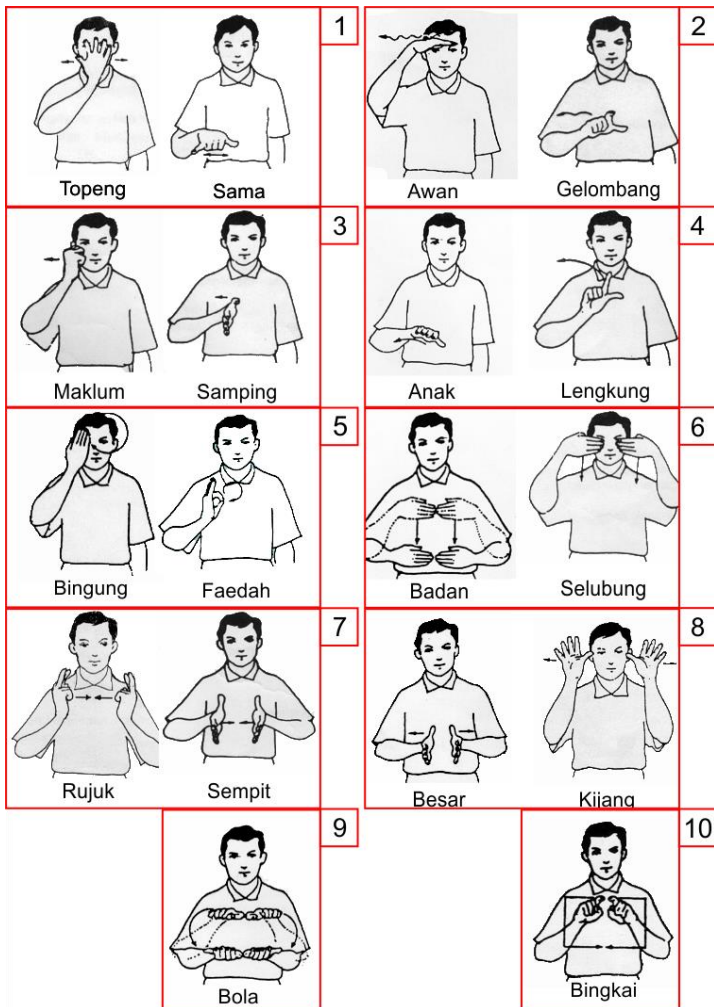
Pada Tugas Akhir ini terdapat 18 bahasa isyarat pokok yang digunakan seperti yang dapat dilihat pada Gambar 2.1. Bahasa isyarat tersebut kemudian dikelompokkan berdasarkan 10 gerakan dasar, yaitu:

- (1) gerakan satu tangan : membentuk garis lurus
(terhadap sumbu X)
- (2) gerakan satu tangan : membentuk garis lurus dua arah
(terhadap sumbu X)
- (3) gerakan satu tangan : membentuk gelombang
- (4) gerakan satu tangan : membentuk lengkungan
- (5) gerakan satu tangan : membentuk lingkaran
- (6) gerakan dua tangan : membentuk garis lurus
(terhadap sumbu Y)
- (7) gerakan dua tangan : membentuk garis lurus dan
kedua tangan saling menyatu
(terhadap sumbu X)
- (8) gerakan dua tangan : membentuk garis lurus dan
kedua tangan saling memisah
(terhadap sumbu X)
- (9) gerakan dua tangan : membentuk lingkaran
- (10) gerakan dua tangan : membentuk persegi

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 [6].



Gambar 2.1 Bahasa Isyarat yang Digunakan



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 [7].

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* [8].

2.6 Ekstraksi Fitur Dinamis dari *Hand Gesture*

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*).

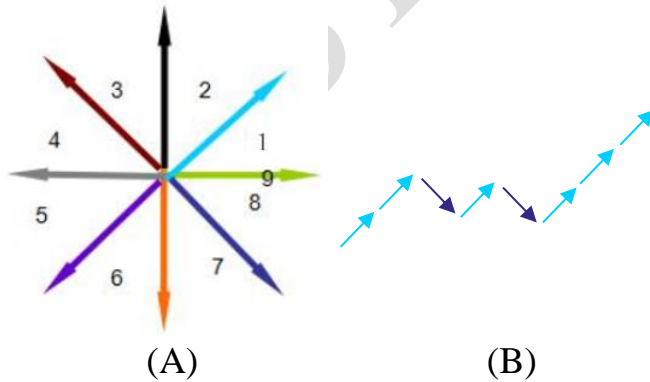
Sebuah fitur yang baik memainkan peran penting dalam pengenalan. Pada Tugas Akhir yang dibuat ini, terdapat 40 *data/frame* yang akan diproses. Setiap *data/frame* akan menghasilkan koordinat gerak dinamis yang diproyeksikan ke

dalam bidang XOY yang merupakan prinsip dalam bidang. Kemudian dari hasil setiap data/*frame* tersebut, akan dicatat orientasi sudut mutlak $\alpha_t \in (0, 360^\circ)$ yang dapat dilihat pada persamaan (1.0) (1.1) dan (1.2) [6]. Orientasi sudut mutlak tersebut kemudian diterjemahkan menjadi data kuantisasi hasil ekstraksi fitur dinamis seperti yang dapat dilihat pada Gambar 2.3.

$$\Delta y = y_t - y_{t-1} \quad (1.0)$$

$$\Delta x = x_t - x_{t-1} \quad (1.1)$$

$$\alpha_t = \begin{cases} \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right) + 180, & \Delta x < 0 \\ \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right) + 360, & \Delta y < 0 \\ \arctan\left(\frac{\Delta y}{\Delta x}\right) * \left(\frac{180}{\pi}\right), & \Delta x > 0, \Delta y \geq 0 \end{cases} \quad (1.2)$$



Gambar 2.3 Pengelompokkan Data Kuantisasi (A) dan Contoh Hasil Kuantisasi (B)

Tabel 2.1 Data Hasil Ekstraksi Fitur Dinamis

Frame	X	Y	Δx	Δy	α_t	Kuant
1	1	5	0	0	0	0
2	2	6	1	1	23	1
3	4	9	2	3	28	1
4	6	4	2	-5	326	7
5	7	5	1	1	23	1
6	9	8	2	3	28	1
7	7	13	-2	5	350	7
8	10	17	3	4	27	1
9	12	18	2	1	13	1
10	13	19	1	1	23	1

Dari contoh data hasil ekstraksi fitur dinamis yang ditunjukkan pada Tabel 2.1, didapatkan data kuantitasi 1, 1, 7, 1, 1, 7, 1, 1, dan 1 yang kemudian digunakan sebagai fitur dinamis dalam mengklasifikasi isyarat gerakan tangan yang diinginkan oleh pengguna. Frame pertama data tidak diolah karena merupakan inisiasi dari proses ekstraksi fitur dinamis.

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 keputusan pohon dan aturan-aturan keputusan (*rule*).

Decision Tree menggunakan struktur hierarki untuk pembelajaran *supervised*. Proses dari Decision Tree dimulai dari *root node* hingga *leaf node* yang dilakukan secara rekursif. Di mana 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 masukan, dan tidak mempunyai keluaran.

1). Algoritma Decision Tree C4.5

Pohon dibangun dengan cara membagi data secara rekursif hingga tiap bagian terdiri dari data yang berasal dari kelas yang sama. Bentuk pemecahan (*split*) yang digunakan untuk membagi data tergantung dari jenis atribut yang digunakan dalam *split*. Algoritma C4.5 dapat menangani data numerik (kontinu) dan diskrit. *Split* untuk atribut numerik yaitu mengurutkan contoh berdasarkan atribut kontinu A , kemudian membentuk minimum permulaan (*threshold*) M dari contoh-contoh yang ada dari kelas mayoritas pada setiap partisi yang bersebelahan, lalu menggabungkan partisi-partisi yang bersebelahan tersebut dengan

kelas mayoritas yang sama. *Split* untuk atribut diskrit A mempunyai bentuk $nilai(A) \in X$ dimana $X \subset domain(A)$.

Jika suatu set data yang mempunyai beberapa pengamatan dengan *missing value* (yaitu *record* dengan beberapa nilai variabel) tidak ada, dan jika jumlah pengamatan terbatas, maka atribut dengan *missing value* dapat diganti dengan nilai rata-rata dari variabel yang bersangkutan.

Dalam melakukan pemisahan obyek dilakukan tes terhadap atribut dengan mengukur tingkat ketidakmurnian pada sebuah *node*. Pada algoritma C4.5 menggunakan rasio perolehan (*gain ratio*). Sebelum menghitung *gain ratio*, perlu menghitung dulu nilai informasi dalam satuan bit dari suatu kumpulan objek. Cara menghitungnya dilakukan dengan menggunakan konsep entropi. Rumus untuk menghitung entropi adalah sebagai berikut:

$$Entropi(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (1.3)$$

S adalah ruang (data) sampel yang digunakan untuk pelatihan, p_+ adalah jumlah yang bersolusi positif atau mendukung pada data sampel untuk kriteria tertentu. Dan p_- adalah jumlah yang bersolusi negatif atau tidak mendukung pada data sampel untuk kriteria tertentu. $Entropi(S)$ sama dengan 0, jika semua contoh pada S berada dalam kelas yang sama. $Entropi(S)$ sama dengan 1, jika jumlah contoh positif dan negatif dalam S adalah sama. $Entropi(S)$ lebih dari 0 tetapi kurang dari 1, jika jumlah contoh positif dan negatif dalam S tidak sama.

Kemudian menghitung perolehan informasi dari keluaran data atau variabel dependen y yang dikelompokkan berdasarkan atribut A , dinotasikan dengan $Gain(y, A)$. Perolehan informasi $Gain(y, A)$ dari atribut A relatif terhadap keluaran data y adalah:

$$Gain(y, A) = entropi(y) - \sum_{c \in nilai(A)} \frac{y_c}{y} entropi(y_c) \quad (1.4)$$

$nilai(A)$ adalah semua nilai yang mungkin dari atribut A , dan y_c adalah subjek dari y dimana A mempunyai nilai c . *Term* pertama dalam persamaan diatas adalah entropi total y dan *term* kedua adalah entropi sesudah dilakukan pemisahan data berdasarkan atribut A .

Untuk menghitung rasio perolehan perlu diketahui suatu *term* baru yang disebut pemisahan informasi (*SplitInfo*). Pemisahan informasi dihitung dengan cara:

$$SplitInfo(S, A) = - \sum_{t=1}^c \frac{S_t}{S} \log_2 \frac{S_t}{S} \quad (1.5)$$

S_1 sampai S_c adalah c subset yang dihasilkan dari pemecahan S dengan menggunakan atribut A yang mempunyai sebanyak c nilai. Selanjutnya *gain ratio* dihitung dengan cara [9]:

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInformation(S, A)} \quad (1.6)$$

[Halaman ini sengaja dikosongkan]

RBTC

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 sarana komunikasi untuk penderita tunarungu. Walaupun agak sulit untuk mengartikan isyarat yang diberikan, hal tersebut telah membantu penderita tunarungu untuk berkomunikasi dengan sekitar. Namun masih banyak yang belum mengerti apa arti isyarat bagi yang diberi atau memberi isyarat.

Perangkat lunak ini bertujuan untuk membantu pengguna dalam mempelajari bahasa isyarat Indonesia yang sesuai dengan SIBI. Data isyarat tersebut didapatkan dari *training* data yang dilakukan oleh pengguna. Dengan menggunakan Microsoft Visual Studio dan Kinect 2.0, penulis mengekstraksi *skeleton joints* yang ditangkap oleh Kinect 2.0 kemudian dikalkulasi dan menghasilkan sebuah fitur. Setelah selesai dibuat, fitur tersebut diolah menggunakan perangkat lunak WEKA untuk membentuk Decision Tree yang akan digunakan sebagai *classifier* dalam menentukan arti bahasa isyarat yang diberikan.

Untuk tahap *testing* data, setelah perangkat lunak mengekstraksi fitur yang didapatkan dari kalkulasi *skeleton* pengguna, fitur tersebut akan diklasifikasikan menggunakan Decision Tree yang sebelumnya sudah dibuat kemudian dimasukkan ke dalam perangkat lunak sebagai sebuah kondisi. Setelah seluruh tahap sudah selesai dilakukan, perangkat lunak

akan memberikan keluaran berupa arti bahasa isyarat Indonesia yang dimaksud oleh pengguna.

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. Pengguna utama dalam perangkat lunak ini adalah semua orang yang ingin mempelajari bahasa isyarat Indonesia. Pengguna dapat menggunakan isyarat yang sudah ada di dalam perangkat lunak ataupun memberikan bahasa isyarat baru dengan 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

Dalam pengembangan perangkat lunak, terdapat beberapa kebutuhan fungsional yang mendukung jalannya perangkat lunak. Kebutuhan fungsional tersebut adalah sebagai berikut:

- a) Mendeteksi *skeleton* pengguna
Perangkat lunak dapat mendeteksi pengguna yang sedang berada di depan Kinect 2.0.
- b) Mengekstraksi fitur dinamis *skeleton* dan menentukan posisi gerakan
Perangkat lunak dapat mendeteksi bagian-bagian dari *skeleton* pengguna yang akan diekstraksi menjadi fitur dinamis dan posisi gerakan guna melakukan proses klasifikasi gerakan.

c) Menerjemahkan Bahasa Isyarat

Perangkat lunak dapat menerjemahkan bahasa isyarat Indonesia yang dihasilkan melalui proses ekstraksi fitur dinamis *skeleton* pengguna.

3.1.2.2 Kebutuhan Non-Fungsional

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

a) Penyesuaian intensitas cahaya

Intensitas cahaya merupakan salah satu unsur penting dalam penggunaan sensor di Kinect 2.0. Jika intensitas cahaya rendah, data *skeleton* yang diambil oleh Kinect 2.0 akan menjadi tidak stabil. Oleh karena itu, dalam menggunakan perangkat lunak ini, sebaiknya dilakukan di ruangan yang mempunyai intensitas cahaya yang cukup.

b) Posisi Kinect dengan pengguna

Dalam mencapai hasil oleh data perangkat lunak (baik itu pengambilan data maupun uji coba data) yang sempurna, jarak optimal Kinect 2.0 dengan pengguna adalah antara 0,6 sampai 1,8 meter.

3.1.3 Identifikasi Pengguna

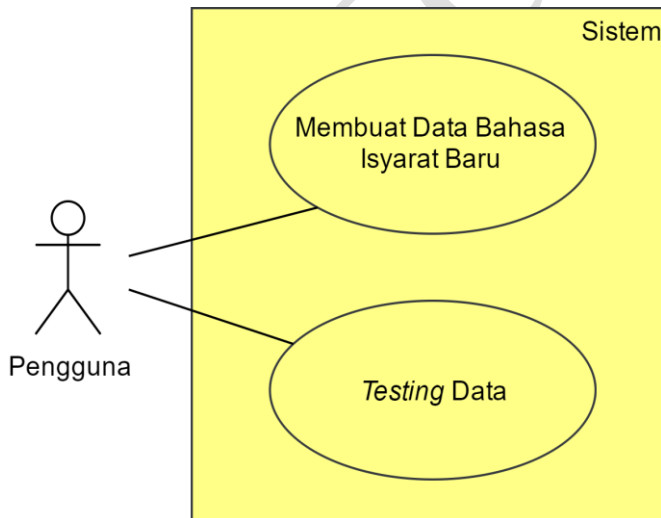
Dalam tugas akhir yang dibangun ini, pengguna yang akan terlibat dalam menjalankan perangkat lunak hanya satu orang saja, yaitu orang yang akan melakukan pengenalan bahasa isyarat Indonesia.

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 3.2.

Tabel 3.2 Definisi Kasus Penggunaan

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

3.2.3 Definisi Kasus Penggunaan

Pada Gambar 3.1 telah dijelaskan bahwa aktor yang dalam hal ini disebut pengguna mempunyai dua kasus penggunaan, yakni membuat data bahasa isyarat Indonesia yang baru dan melakukan *testing* data. Rincian mengenai kasus penggunaan tersebut dapat dilihat pada Tabel 3.3.

Tabel 3.3 Definisi Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-01	Membuat Data Bahasa Isyarat Baru	Pengguna membuat data bahasa isyarat Indonesia yang baru
2	UC-02	<i>Testing</i> Data	Pengguna melakukan <i>testing</i> data 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 3.4.

Tabel 3.4 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 aplikasi dan perangkat Kinect 2.0 telah tersambung
Alur Normal	<div><div>1. Pengguna memasukkan nama bahasa isyarat yang akan dibuat di dalam <i>textbox</i> perangkat lunak</div><div>2. Pengguna menekan tombol “Create File” di dalam perangkat lunak</div><div>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</div><div>A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna</div><div>4. Pengguna melakukan gerakan bahasa isyarat yang akan dibuat</div></div>

	5. Perangkat lunak mengekstrak data <i>skeleton</i> pengguna sebanyak 40 data/frame 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 data gerakan yang baru

3.2.3.2 Kasus Penggunaan *Testing Data*

Spesifikasi kasus penggunaan *Testing Data* dapat dilihat pada Tabel 3.5.

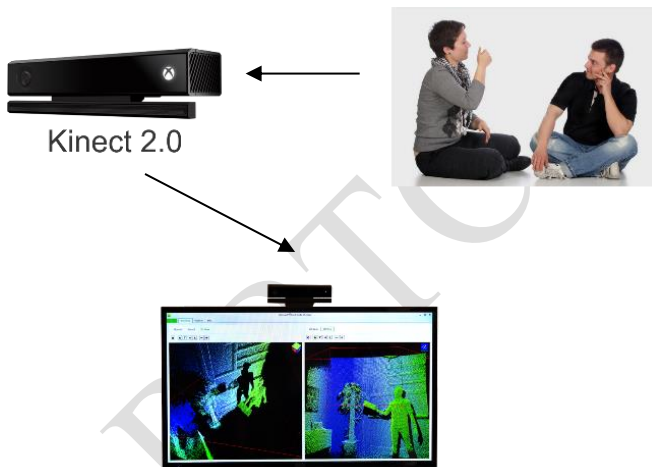
Tabel 3.5 Spesifikasi Kasus Penggunaan *Training Dataset*

Nama Kasus Penggunaan	<i>Testing Data</i>
Nomor	UC-02
Deskripsi	Kasus penggunaan aktor untuk melakukan <i>testing data</i> isyarat bahasa Indonesia yang telah dibuat
Aktor	Pengguna
Kondisi Awal	Pengguna dalam keadaan menjalankan perangkat lunak dan sudah ada <i>classifier</i> data isyarat bahasa Indonesia di dalam perangkat lunak

Alur Normal	<ol style="list-style-type: none"> 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 data/frame untuk dikalkulasi 5. Hasil ekstraksi data diklasifikasi menggunakan <i>Decision Tree</i> yang sudah dibuat di dalam perangkat lunak 6. Perangkat lunak menampilkan isyarat bahasa Indonesia hasil klasifikasi
Alur Alternatif	<p>A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna</p> <ol style="list-style-type: none"> 1. Sistem memberikan notifikasi bahwa <i>skeleton</i> pengguna tidak ditemukan
Kondisi Akhir	Perangkat lunak memberikan keluaran berupa bahasa isyarat yang dimaksud oleh pengguna baik itu dalam bentuk tulisan maupun dalam bentuk gambar

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 3.2.



Gambar 3.2 Arsitektur Umum Sistem

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 apakah tampilan yang sudah disediakan oleh perangkat lunak mudah untuk dipahami dan digunakan, sehingga akan muncul kesan pengalaman pengguna yang baik dan mudah.

Rancangan antarmuka perangkat lunak ini hanya memiliki satu Windows dan memiliki beberapa kontrol yang sekiranya dapat

dipahami oleh pengguna. Rancangan antarmuka perangkat lunak dapat dilihat pada Gambar 3.3.



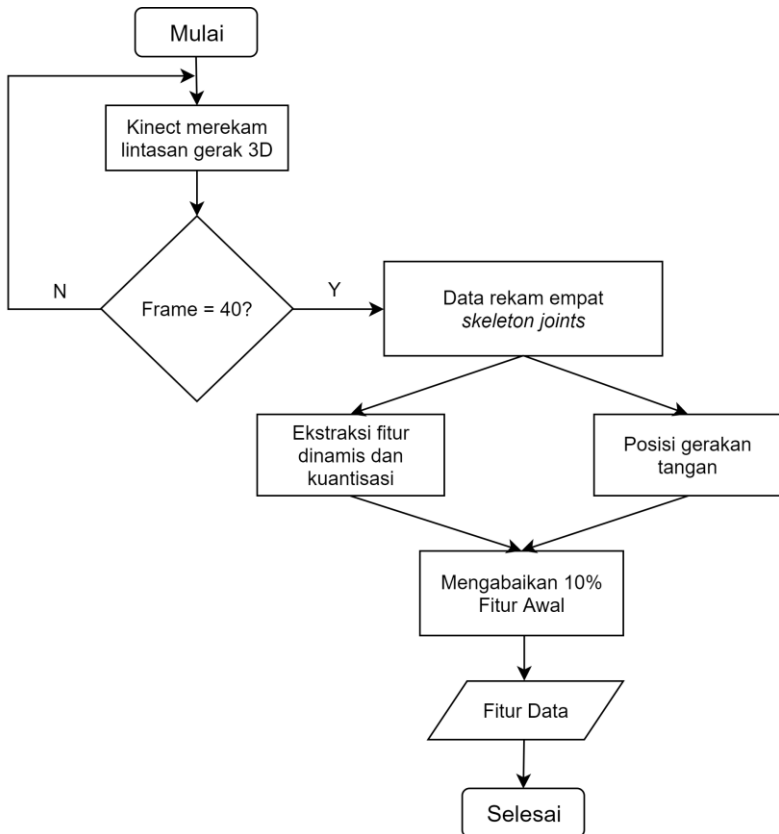
Gambar 3.3 Rancangan Antarmuka Perangkat Lunak

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 Dinamis dan Menentukan Posisi Gerakan Tangan

Proses ekstraksi fitur dinamis dari *skeleton* pengguna dan menentukan posisi gerakan merupakan hal yang terpenting dalam pembangunan perangkat lunak ini. Rancangan proses ekstraksi fitur dinamis dan menentukan posisi gerakan tangan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram Alir Proses Ekstraksi Fitur Dinamis dan Proses Penentuan Posisi Gerakan Tangan

1. Perekaman gerakan oleh Kinect 2.0

Dalam perekaman gerakan 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 data/frame untuk dikalkulasi dan dijadikan fitur dinamis dan juga mengidentifikasi posisi gerakan yang dilakukan.

Jumlah data/frame 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 sekali kembali guna mendapatkan hasil yang sesuai.

2. Penentuan *skeleton joints* yang digunakan

Terdapat empat *skeleton joints* yang diolah dalam Tugas Akhir ini. Masing-masing *skeleton joints* diinterpretasikan dalam bentuk koordinat X dan Y seperti yang sudah dijelaskan pada Subbab 2.6. Empat *skeleton joints* tersebut adalah:

1. Telapak tangan kanan (HR)
2. Telapak tangan kiri (HL)
3. Leher (N)
4. Bagian tengah tulang belakang (SM)

Data koordinat *skeleton joints* HR dan HL digunakan dalam mengekstraksi fitur dinamis berdasarkan persamaan (1.0), (1.1), dan (1.2). Sedangkan data koordinat *skeleton joints* N dan SM digunakan dalam menentukan posisi gerakan tangan pengguna.

3. Ekstraksi fitur dinamis dan kuantisasi

Dalam proses ekstraksi fitur dinamis, berdasarkan persamaan (1.0) dan (1.1), perangkat lunak terlebih dahulu menentukan Δx dan Δy yang didapatkan dari hasil selisih masing-masing koordinat X dan Y antar data/frame 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 (1.2) 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 π (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 2.3) yang dapat dijelaskan sebagai berikut:

1. Jika α_t sama dengan nol, maka kuantisasinya adalah 9.
2. Jika α_t lebih besar daripada 314, maka kuantisasinya adalah 8.
3. Jika α_t lebih besar daripada 269, maka kuantisasinya adalah 7.
4. Jika α_t lebih besar daripada 224, maka kuantisasinya adalah 6.
5. Jika α_t lebih besar daripada 179, maka kuantisasinya adalah 5.
6. Jika α_t lebih besar daripada 134, maka kuantisasinya adalah 4.
7. Jika α_t lebih besar daripada 89, maka kuantisasinya adalah 3.
8. Jika α_t lebih besar daripada 44, maka kuantisasinya adalah 2.
9. Jika α_t lebih kecil daripada 44, maka kuantisasinya adalah 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 data/frame yang dihasilkan oleh tangan kiri.

Sedangkan fitur ke 41 s.d. 80 menginterpretasikan data kuantisasi 40 data/frame yang dihasilkan oleh tangan kanan.

4. Posisi gerakan tangan

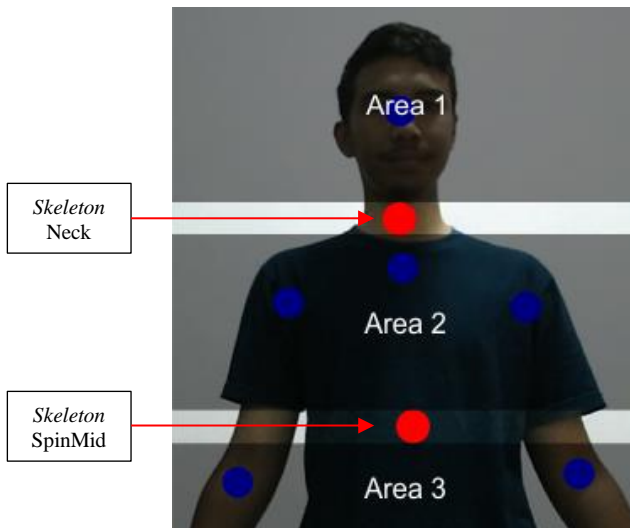
Pada data/frame ke-20, perangkat lunak mengidentifikasi posisi gerakan tangan yang dilakukan oleh pengguna. Seperti yang dapat dilihat pada Gambar 3.5, posisi gerakan tangan dibagi menjadi tiga daerah, yaitu Area 1 (kepala), Area 2 (dada), dan Area 3 (perut). Daerah posisi gerakan tangan tersebut ditentukan 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.

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

Di dalam fitur data (gabungan antara fitur dinamis dan posisi gerakan tangan), setelah dilakukan seluruh proses yang sudah dijelaskan sebelumnya, perangkat lunak kemudian menyeleksi terlebih dahulu fitur dinamis yang didapatkan dengan hanya mengambil data/frame genap untuk diolah dan juga mengabaikan 10% data/frame awal dengan harapan agar mendapatkan hasil yang lebih akurat [6]. Pembagian fitur data dapat dilihat pada Tabel 3.6.



Gambar 3.5 Posisi Gerakan Tangan pada Tubuh Mengacu pada *Skeleton* Leher (N) dan Bagian Tengah Tulang Belakang (SM)

Tabel 3.6 Pembagian Fitur Data

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

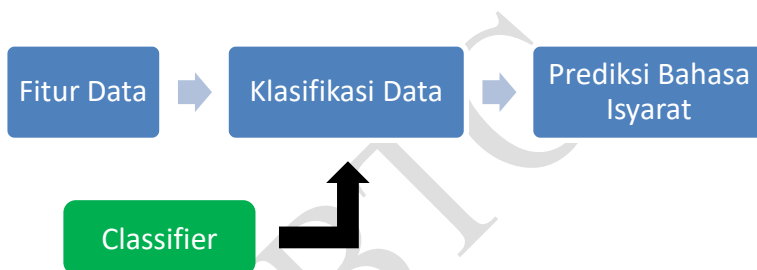
3.2.6.2 Rancangan Proses *Training* Data



Gambar 3.6 Rancangan Proses *Training* Data

Seperti yang dapat dilihat pada Gambar 3.6, proses *training* data dimulai dengan menyimpan fitur data yang didapatkan pada Subbab 3.2.6.1 terlebih dahulu ke dalam sebuah berkas berekstensi .csv. Data *training* tersebut kemudian diolah pada perangkat lunak WEKA menggunakan algoritma Decision Tree C4.5 (J48). Keluaran yang didapatkan dari hasil *training* data tersebut berupa Decision Tree dan digunakan sebagai *classifier* ketika pengguna melakukan *testing* data. *Classifier* berupa Decision Tree dapat dilihat pada Gambar 3.8.

3.2.6.3 Rancangan Proses *Testing* Data



Gambar 3.7 Rancangan Proses *Testing* Data

Seperti yang dapat dilihat pada Gambar 3.7, proses *testing* data dilakukan dengan mengklasifikasikan fitur data baru dengan *classifier* dalam bentuk Decision Tree 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.

[Halaman ini sengaja dikosongkan]

RBTC

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber dengan bahasa pemrograman C# untuk membangun program. Sebelum masuk ke penjelasan implementasi, akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Pembangunan

Dalam membangun aplikasi ini digunakan beberapa perangkat pendukung baik perangkat keras maupun perangkat lunak. Lingkungan pembangunan 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) 8,00 GB
- Kinect Sensor

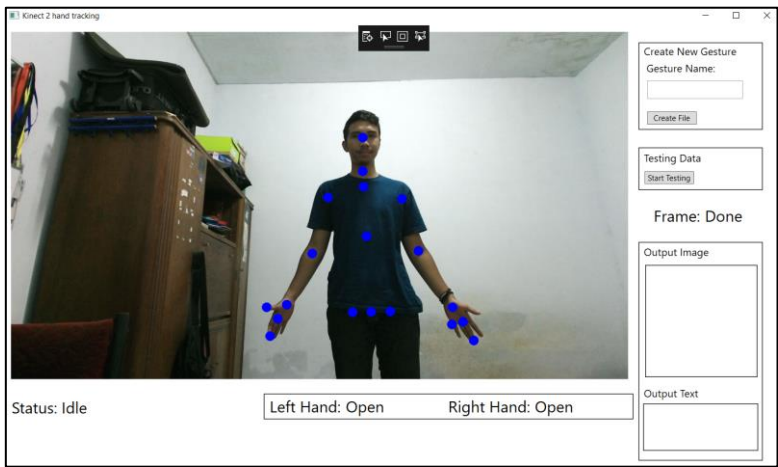
4.1.2 Lingkungan Pembangunan Perangkat Lunak

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

- Microsoft Visual Studio 2015
- Windows 10 Education 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
3	/xaml/presentation"
4	xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5	Title="Kinect 2 hand tracking"
6	Height="710" Width="1200"
7	Loaded="Window_Loaded" Closed="Window_Closed">
8	<Grid Margin="0,0,2,0">
9	<Border BorderBrush="Black"
10	BorderThickness="1" HorizontalAlignment="Left"
11	Height="39" Margin="395,561,0,0"
12	VerticalAlignment="Top" Width="563"/>
13	<Border BorderBrush="Black" BorderThickness="1"
14	HorizontalAlignment="Left" Height="171"

15	Margin="976,365,0,0" VerticalAlignment="Top"
16	Width="171"/>
17	<Border BorderBrush="Black" BorderThickness="1"
18	HorizontalAlignment="Left" Height="133"
19	Margin="966,26,0,0" VerticalAlignment="Top"
20	Width="189"/>
21	<Border BorderBrush="Black" BorderThickness="1"
22	HorizontalAlignment="Left" Height="65"
23	Margin="966,186,0,0" VerticalAlignment="Top"
24	Width="189"/>
25	<Border BorderBrush="Black" BorderThickness="1"
26	HorizontalAlignment="Left" Height="333"
27	Margin="966,330,0,0" VerticalAlignment="Top"
28	Width="189"/>
29	<Border BorderBrush="Black" BorderThickness="1"
30	HorizontalAlignment="Left" Height="72"
31	Margin="973,576,0,0" VerticalAlignment="Top"
32	Width="175"/>
33	
34	<Viewbox Width="940" Height="700" Margin="10,10,0,0"
35	HorizontalAlignment="Left"
36	VerticalAlignment="Top">
37	<Grid>
38	<Image Name="camera" Width="1920"
39	Height="1080" />
40	<Canvas Name="canvas" Width="1920"
41	Height="1080" ClipToBounds="True" />
42	</Grid>
43	</Viewbox>
44	
45	<Label Name="label3" Content="Testing Data"
46	HorizontalAlignment="Left" Margin="969,186,0,0"
47	VerticalAlignment="Top" Width="152"
48	FontSize="16"/>
49	<Button Name="OneTestButton" Content="Start Testing"
50	HorizontalAlignment="Left" Margin="975,222,0,0"
51	VerticalAlignment="Top" Width="75"
52	Click="OneTestButton_Click"/>
53	
54	<Label Name="label5" Content="Create New Gesture"
55	HorizontalAlignment="Left" Margin="969,23,0,0"
56	VerticalAlignment="Top" Width="152"
57	FontSize="16"/>
58	<Label Name="label6" Content="Gesture Name:"
59	HorizontalAlignment="Left" Margin="973,49,0,0"
60	VerticalAlignment="Top" Width="115"
61	FontSize="16"/>
62	<TextBox Name="fileName" HorizontalAlignment="Left"
63	Height="28" Margin="979,84,0,0"

64	TextWrapping="Wrap" VerticalAlignment="Top"
65	Width="146" FontSize="16"/>
66	<Button Name="createButton" Content="Create File"
67	HorizontalAlignment="Left" Margin="979,131,0,0"
68	VerticalAlignment="Top" Width="75"
69	Click="createButton_click"/>
70	
71	<Label Name="label7" Content="Output Image"
72	HorizontalAlignment="Left" Margin="969,329,0,0"
73	VerticalAlignment="Top" Width="152"
74	FontSize="16"/>
75	<Border HorizontalAlignment="Left" Height="171"
76	Margin="976,366,0,0" VerticalAlignment="Top"
77	Width="171" >
78	<Image x:Name="outputImage"
79	Stretch="UniformToFill"
80	Margin="0,-3,0.2,2.2"/>
81	</Border>
82	
83	<Label Name="label8" Content="Output Text"
84	HorizontalAlignment="Left" Margin="969,544,0,0"
85	VerticalAlignment="Top" Width="152"
86	FontSize="16"/>
87	<Label Name="outputText" Content=""
88	HorizontalAlignment="Left" Margin="973,585,0,0"
89	VerticalAlignment="Top" Height="48" Width="174"
90	FontSize="26.667"
91	RenderTransformOrigin="0.981,2.1"/>
92	
93	<Label Name="label9" Content="Status:"
94	HorizontalAlignment="Left" Margin="6,560,0,0"
95	VerticalAlignment="Top" Width="100"
96	FontSize="24" Height="47"/>
97	<Label Name="statusDetail"
98	Content="Kinect Not Connected"
99	HorizontalAlignment="Left" Height="47"
100	Margin="82,560,0,0" VerticalAlignment="Top"
101	Width="341" FontSize="24"/>
102	
103	<Label Name="label10" Content="Left Hand:"
104	HorizontalAlignment="Left" Margin="399,559,0,0"
105	VerticalAlignment="Top" Width="119"
106	FontSize="24" Height="47"/>
107	<Label Name="tblLeftHandState" Content="Not
108	Detected" HorizontalAlignment="Left"
109	Height="47" Margin="515,559,0,0"
110	VerticalAlignment="Top"
111	Width="187" FontSize="24"/>

112	<Label Name="label11" Content="Right Hand:"
113	HorizontalAlignment="Left" Margin="671,559,0,0"
114	VerticalAlignment="Top" Width="144"
115	FontSize="24" Height="47"/>
116	<Label Name="tblRightHandState" Content="Not
117	Detected" HorizontalAlignment="Left" Height="47"
118	Margin="803,559,0,0" VerticalAlignment="Top"
119	Width="187" FontSize="24"/>
120	
121	<Label Name="label12" Content="Frame:"
122	HorizontalAlignment="Left" Margin="984,268,0,0"
123	VerticalAlignment="Top" Width="144"
124	FontSize="24" Height="47"/>
125	<Label Name="ambilData" Content="Done"
126	HorizontalAlignment="Left" Height="47"
127	Margin="1062,268,-57.4,0"
128	VerticalAlignment="Top" Width="187"
129	FontSize="24"/>
130	</Grid>
131	</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 tentunya membutuhkan perangkat keras Kinect 2.0, sehingga dibutuhkan suatu proses untuk mendeteksi *skeleton* pengguna. 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 dinamis, perangkat lunak mendeteksi tubuh pengguna terlebih dahulu. Ketika tubuh pengguna sudah terdeteksi, perangkat lunak

kemudian menggambarkan *skeleton* pengguna secara keseluruhan termasuk 4 *skeleton joints* yang akan digunakan dalam proses ekstraksi fitur dinamis dan menentukan posisi gerakan. Kode sumber untuk mendeteksi *skeleton* pengguna dapat dilihat pada Kode Sumber 4.3.

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

Kode Sumber 4.2 Kode Sumber Integrasi Kinect

1	<code>async void Reader_MultiSourceFrameArrived(object sender,</code>
	<code>MultiSourceFrameArrivedEventArgs e)</code>
3	<code>{</code>
4	<code> var reference = e.FrameReference.AcquireFrame();</code>
5	
6	<code> using (var frame =</code>
	<code> reference.ColorFrameReference.AcquireFrame())</code>
7	<code> {</code>
8	<code> if (frame != null)</code>
9	<code> {</code>
10	<code> camera.Source = frame.ToBitmap();</code>
11	<code> }</code>
12	<code> }</code>
13	
14	<code> using (var frame =</code>
	<code> reference.BodyFrameReference.AcquireFrame())</code>
15	<code> {</code>
16	<code> if (frame != null)</code>
17	<code> {</code>
18	<code> canvas.Children.Clear();</code>
19	<code> _bodies = new</code>
	<code> Body[frame.BodyFrameSource.BodyCount];</code>
20	<code> frame.GetAndRefreshBodyData(_bodies);</code>
21	<code> }</code>

```

22         foreach (var body in _bodies)
23         {
24             if (body != null)
25             {
26                 if (body.IsTracked)
27                 {
28                     Joint handRight =
29                         body.Joints[JointType.HandRight];
30                     Joint handLeft =
31                         body.Joints[JointType.HandLeft];
32                     Joint neck =
33                         body.Joints[JointType.Neck];
34                     Joint spineMid =
35                         body.Joints[JointType.SpineMid];
36
37                     foreach (Joint joint in
38                         body.Joints.Values)
39                     {
40                         if (joint.TrackingState ==
41                             TrackingState.Tracked)
42                         {
43                             // 3D space point
44
45                             CameraSpacePoint
46                                 jointPosition =
47                                 joint.Position;
48
49                             // 2D space point
50
51                             Point point = new Point();
52
53                             ColorSpacePoint colorPoint =
54                                 _sensor.CoordinateMapper
55                                 .MapCameraPointToColorSpace
56                                 (jointPosition);
57                             point.X = float
58                                 .IsInfinity(colorPoint.X)
59                                 ? 0 : colorPoint.X;
60                             point.Y = float
61                                 .IsInfinity(colorPoint.Y)
62                                 ? 0 : colorPoint.Y;
63
64                             Ellipse ellipse = new Ellipse
65                             {
66                                 Fill = Brushes.Blue,
67                                 Width = 30,
68                                 Height = 30
69                             };

```

57		<code>Canvas.SetLeft(ellipse,</code>
		<code>point.X</code>
		<code>- ellipse.Width / 2);</code>
58		<code>Canvas.SetTop(ellipse,</code>
		<code>point.Y</code>
		<code>- ellipse.Height / 2);</code>
59		
60		<code>canvas.Children.Add(ellipse);</code>
61		<code>}</code>
62		<code>}</code>
63		<code>}</code>
64		<code>}</code>
65		<code>}</code>
66		<code>}</code>
67		<code>}</code>
68		<code>}</code>

Kode Sumber 4.3 Kode Sumber Deteksi *Skeleton* Pengguna

4.3.2 Implementasi Proses Ekstraksi Fitur Dinamis dan Menentukan Posisi Gerakan Tangan

Pada Kode Sumber 4.4 dijelaskan mengenai proses ekstraksi fitur dinamis pada setiap *data/frame* dan menentukan posisi gerakan tangan yang dilakukan oleh pengguna berdasarkan rancangan pada Subbab 3.2.6.1. Penjelasan kode sumber tersebut adalah sebagai berikut:

1. Variabel *i* menunjukkan *data/frame* ke-*i* yang sedang diolah.
2. Variabel *leher* menunjukkan koordinat *skeleton joints* N.
3. Variabel *tengah* menunjukkan koordinat *skeleton joints* SM.
4. Variabel *b* menunjukkan koordinat *skeleton joints* HL.
5. Variabel *y* menunjukkan koordinat *skeleton joints* HR.
6. Variabel *tanganKanan* menunjukkan posisi tangan kanan.
7. Variabel *tanganKiri* menunjukkan posisi tangan kiri.
8. Variabel *deltaa* menunjukkan Δx tangan kiri.
9. Variabel *deltab* menunjukkan Δy tangan kiri.
10. Variabel *alpha1* menunjukkan α_t tangan kiri.

11. Variabel *deltax* menunjukkan Δx tangan kanan.
12. Variabel *deltay* menunjukkan Δy tangan kanan.
13. Variabel *alpha2* menunjukkan α_t tangan kanan.
14. Variabel *kuant1* menunjukkan hasil kuantisasi fitur dinamis tangan kiri.
15. Variabel *kuant2* menunjukkan hasil kuantisasi fitur dinamis tangan kanan.

Fitur dinamis dan fitur gerakan tangan digabungkan menjadi fitur data yang kemudian disimpan ke dalam sebuah berkas berekstensi .csv dengan pembagian fitur data berdasarkan Tabel 3.6. 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)
2  {
3      //flag ambil data
4
5      if (i < 38)
6      {
7          ambilData.Content = (i + 1).ToString();
8      }
9      else ambilData.Content = "Done";
10
11     //posisi Neck dan SpinerMid
12
13     if (i == 1)
14     {
15         leher = (neck.Position.Y);
16         tengah = (spineMid.Position.Y);
17     }
18
19     //flag posisi gerakan
20
21     if (i == 20)
22     {
23         if (b > leher) tanganKiri = "Kepala";
24         else if (b < tengah) tanganKiri = "Perut";
25         else tanganKiri = "Dada";
26
27         if (y > leher) tanganKanan = "Kepala";
28         else if (y < tengah) tanganKanan = "Perut";
29         else tanganKanan = "Dada";

```

```

30     }
31
32     //ekstraksi fitur tangan kiri
33
34     if (deltaa >= 0 && deltab >= 0)
35     {
36         alpha1 = (Math.Atan(deltab / deltaa)) * (180 /
37             Math.PI);
38     }
39     else if (deltaa < 0)
40     {
41         alpha1 = (Math.Atan(deltab / deltaa)) * (180 /
42             Math.PI) + 180;
43     }
44     else { alpha1 = (Math.Atan(deltab / deltaa)) * (180 /
45         Math.PI) + 360; }
46
47     //ekstraksi fitur tangan kanan
48
49     if (deltax >= 0 && deltay >= 0)
50     {
51         alpha2 = (Math.Atan(deltay / deltax)) * (180 /
52             Math.PI);
53     }
54     else if (deltax < 0)
55     {
56         alpha2 = (Math.Atan(deltay / deltax)) * (180 /
57             Math.PI) + 180;
58     }
59     else
60     {
61         alpha2 = (Math.Atan(deltay / deltax)) * (180 /
62             Math.PI) + 360;
63     }
64
65     //kuantisasi hasil ekstraksi tangan kiri
66
67     if (alpha1 > 314)
68     {
69         kuant1[i] = 8;
70     }
71     else if (alpha1 > 269)
72     {
73         kuant1[i] = 7;
74     }
75     else if (alpha1 > 224)
76     {
77         kuant1[i] = 6;
78     }
79

```

```
73     else if (alpha1 > 179)
74     {
75         kuant1[i] = 5;
76     }
77     else if (alpha1 > 134)
78     {
79         kuant1[i] = 4;
80     }
81     else if (alpha1 > 89)
82     {
83         kuant1[i] = 3;
84     }
85     else if (alpha1 > 44)
86     {
87         kuant1[i] = 2;
88     }
89     else
90     {
91         kuant1[i] = 1;
92     }
93
94     //kuantisasi hasil ekstraksi tangan kanan
95
96     if (alpha2 > 314)
97     {
98         kuant2[i] = 8;
99     }
100    else if (alpha2 > 269)
101    {
102        kuant2[i] = 7;
103    }
104    else if (alpha2 > 224)
105    {
106        kuant2[i] = 6;
107    }
108    else if (alpha2 > 179)
109    {
110        kuant2[i] = 5;
111    }
112    else if (alpha2 > 134)
113    {
114        kuant2[i] = 4;
115    }
116    else if (alpha2 > 89)
117    {
118        kuant2[i] = 3;
119    }
120    else if (alpha2 > 44)
121    {
```

```

123         kuant2[i] = 2;
124     }
125     else
126     {
127         kuant2[i] = 1;
128     }
129 }
130
131 //increment penanda untuk data/frame selanjutnya
132
133 i++;

```

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

```

1  if (i == 38)
2  {
3      var stringkuant4 = kuant1[4].ToString();
4      var stringkuant6 = kuant1[6].ToString();
5      var stringkuant8 = kuant1[8].ToString();
6      var stringkuant10 = kuant1[10].ToString();
7      var stringkuant12 = kuant1[12].ToString();
8      var stringkuant14 = kuant1[14].ToString();
9      var stringkuant16 = kuant1[16].ToString();
10     var stringkuant18 = kuant1[18].ToString();
11     var stringkuant20 = kuant1[20].ToString();
12     var stringkuant22 = kuant1[22].ToString();
13     var stringkuant24 = kuant1[24].ToString();
14     var stringkuant26 = kuant1[26].ToString();
15     var stringkuant28 = kuant1[28].ToString();
16     var stringkuant30 = kuant1[30].ToString();
17     var stringkuant32 = kuant1[32].ToString();
18     var stringkuant34 = kuant1[34].ToString();
19     var stringkuant36 = kuant1[36].ToString();
20     var stringkuant38 = kuant1[38].ToString();
21
22     var stringkuant43 = kuant2[4].ToString();
23     var stringkuant45 = kuant2[6].ToString();
24     var stringkuant47 = kuant2[8].ToString();
25     var stringkuant49 = kuant2[10].ToString();
26     var stringkuant51 = kuant2[12].ToString();
27     var stringkuant53 = kuant2[14].ToString();
28     var stringkuant55 = kuant2[16].ToString();
29     var stringkuant57 = kuant2[18].ToString();
30     var stringkuant59 = kuant2[20].ToString();
31     var stringkuant61 = kuant2[22].ToString();
32     var stringkuant63 = kuant2[24].ToString();
33     var stringkuant65 = kuant2[26].ToString();
34     var stringkuant67 = kuant2[28].ToString();
35     var stringkuant69 = kuant2[30].ToString();

```

36	var stringkuant71 = kuant2[32].ToString();
37	var stringkuant73 = kuant2[34].ToString();
38	var stringkuant75 = kuant2[36].ToString();
39	var stringkuant77 = kuant2[38].ToString();
40	var stringtangankiri = tanganKiri;
41	var stringtangankanan = tanganKanan;
42	var stringnamagerakan = namaGerakan;
43	
44	if (statusAmbil == 1)
45	{
46	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},{34},
	{35},{36},{37},{38}",
47	stringkuant4, stringkuant6,
48	stringkuant8, stringkuant10,
49	stringkuant12, stringkuant14,
50	stringkuant16, stringkuant18,
51	stringkuant20, stringkuant22,
52	stringkuant24, stringkuant26,
53	stringkuant28, stringkuant30,
54	stringkuant32, stringkuant34,
55	stringkuant36, stringkuant38,
56	stringkuant43, stringkuant45,
57	stringkuant47, stringkuant49,
58	stringkuant51, stringkuant53,
59	stringkuant55, stringkuant57,
60	stringkuant59, stringkuant61,
61	stringkuant63, stringkuant65,
62	stringkuant67, stringkuant69,
63	stringkuant71, stringkuant73,
64	stringkuant75, stringkuant77,
65	stringtangankiri, stringtangankanan,
66	stringnamagerakan);
67	
68	//memasukkan ke dalam baris
69	
70	csv.AppendLine(newLine);
71	}
72	}

Kode Sumber 4.5 Kode Sumber Menyimpan Hasil Ekstraksi Fitur Dinamis dari *Skeleton* Pengguna

4.3.4 Implementasi Proses *Testing* Data

Tree hasil dari *training* data yang dapat dilihat pada Gambar 3.8 kemudian diimplementasikan 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  if (stringtangankiri == "Perut")
2  {
3      if(stringtangankan == "Kepala")
4      {
5          if (kuant2[34] <= 3)
6          {
7              if (kuant2[22] <= 4)
8              {
9                  outputText.Content = "Maklum";
10             }
11             else
12             {
13                 if (kuant2[14] <= 7)
14                 {
15                     outputText.Content = "Bingung";
16                 }
17                 else
18                 {
19                     outputText.Content = "Awan";
20                 }
21             }
22         }
23     else
24     {
25         if (kuant2[34] <= 6)
26         {
27             outputText.Content = "Topeng";
28         }
29         else
30         {
31             if (kuant2[18] <= 3)
32             {
33                 outputText.Content = "Maklum";
34             }
35             else
36             {
37                 outputText.Content = "Awan";
```

```

38         }
39     }
40 }
41 }
42 else if(stringtangankanan == "Perut")
43 {
44     if (kuant1[14] <= 6)
45     {
46         if (kuant1[34] <= 6)
47         {
48             if (kuant1[16] <= 6)
49             {
50                 outputText.Content = "Besar";
51             }
52             else
53             {
54                 outputText.Content = "Badan";
55             }
56         }
57         else
58         {
59             outputText.Content = "Bola";
60         }
61     }
62     else
63     {
64         if (kuant1[34] <= 6)
65         {
66             if(kuant2[14] <= 5)
67             {
68                 outputText.Content = "Sempit";
69             }
70             else
71             {
72                 outputText.Content = "Badan";
73             }
74         }
75         else
76         {
77             if (kuant2[28] <= 6)
78             {
79                 if (kuant2[28] <= 2)
80                 {
81                     outputText.Content = "Samping";
82                 }
83                 else
84                 {
85                     if (kuant2[30] <= 5)
86                     {

```

87	if (kuant1[26] <= 6)
88	{
89	outputText.Content =
	"Sempit";
90	}
91	else
92	{
93	outputText.Content = "Sama";
94	}
95	}
96	else
97	{
98	outputText.Content = "Sempit";
99	}
100	}
101	}
102	else
103	{
104	if(kuant2[10] <= 6)
105	{
106	if(kuant2[34] <= 4)
107	{
108	outputText.Content = "Samping";
109	}
110	else
111	{
112	outputText.Content = "Anak";
113	}
114	}
115	else
116	{
117	outputText.Content = "Samping";
118	}
119	}
120	}
121	}
122	}
123	else
124	{
125	if (kuant2[16] <= 3)
126	{
127	if (kuant2[30] <= 4)
128	{
129	outputText.Content = "Lengkung";
130	}
131	else
132	{
133	outputText.Content = "Gelombang";
134	}


```

135     }
136     else
137     {
138         if (kuant2[18] <= 6)
139         {
140             outputText.Content = "Faedah";
141         }
142         else
143         {
144             if (kuant2[16] <= 5)
145             {
146                 outputText.Content = "Faedah";
147             }
148             else
149             {
150                 outputText.Content = "Gelombang";
151             }
152         }
153     }
154 }
155 }
156 else if (stringtangankiri == "Dada")
157 {
158     if (kuant1[22] <= 6 )
159     {
160         outputText.Content = "Bingkai";
161     }
162     else
163     {
164         outputText.Content = "Rujuk";
165     }
166 }
167 else
168 {
169     if (kuant2[36] <= 7)
170     {
171         if (kuant2[36] <= 3)
172         {
173             outputText.Content = "Kijang";
174         }
175         else {
176             outputText.Content = "Selubung";
177         }
178     }
179     else {
180         outputText.Content = "Kijang";
181     }
182 }

```

Kode Sumber 4.6 Implementasi Decision Tree dari Hasil Klasifikasi

[Halaman ini sengaja dikosongkan]

RBTC

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) 8,00 GB
- Kinect Sensor

5.2 Skenario Pengujian

Pengujian dilakukan terhadap 18 bahasa isyarat yang dipilih oleh penulis seperti yang dapat dilihat pada Gambar 2.1. Bahasa isyarat tersebut adalah sebagai berikut:

1. Topeng
2. Badan
3. Awan
4. Anak
5. Maklum
6. Bola
7. Besar
8. Rujuk

9. Bingkai
10. Bingung
11. Selubung
12. Gelombang
13. Lengkung
14. Samping
15. Kijang
16. Sempit
17. Faedah
18. Sama

Skenario pengujian yang dilakukan dibagi menjadi dua bagian, yaitu skenario A dan skenario B. Pada skenario A, dilakukan pengujian bahasa isyarat menggunakan 18 gerakan seperti yang sudah dijelaskan pada Subbab 2.2. Kemudian pada skenario B, dilakukan penambahan satu gerakan bahasa isyarat statis dalam pengujian bahasa isyarat guna mengetahui apakah bahasa isyarat statis dapat diidentifikasi menggunakan fitur dinamis yang digunakan di dalam Tugas Akhir ini.

1. Pengujian skenario A1 merupakan pengujian akurasi yang dilakukan oleh penulis dengan menggunakan data *training* yang sudah ada sebelumnya.
2. Pengujian skenario A2 dan A3 merupakan pengujian akurasi yang dilakukan oleh pengguna lain menggunakan data *training* yang sudah ada sebelumnya.
3. Pengujian skenario B merupakan pengujian yang dilakukan oleh penulis dengan menambahkan satu data gerakan bahasa isyarat statis di dalam data *training*.

5.2.1 Pengujian Skenario A1 dan Analisis

Pada pengujian skenario A1, uji coba dilakukan sendiri oleh penulis yang mempunyai karakteristik tinggi badan 171 cm dengan menggunakan data gerakan sebanyak 269 data/frame hasil

ekstraksi fitur dinamis dan posisi gerakan. Skenario dapat dilihat pada Tabel 5.1.

Tabel 5.1 Skenario Pengujian A1

Nama Skenario Pengujian	Pengujian Akurasi A1
Kode	SP-A1
Algoritma	Decision Tree
Jumlah Data/Frame	269 data/frame
Penguji	Penulis memiliki tinggi 171 cm
Prosedur Pengujian	Penulis melakukan uji coba 18 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 86,6%

Hasil yang didapatkan pada skenario A1 cukup baik, seperti yang dapat dilihat pada Tabel 5.2. Kesalahan klasifikasi terjadi pada beberapa gerakan. Menurut pengamatan penulis berdasarkan skenario A1 yang dilakukan, kesalahan klasifikasi tersebut terjadi karena posisi dan gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan.

5.2.2 Pengujian Skenario A2 dan Analisis

Pada pengujian skenario A2, uji coba dilakukan oleh pengguna yang mempunyai karakteristik tinggi badan 165 cm dengan menggunakan data gerakan sebanyak 269 data/frame hasil ekstraksi fitur dinamis dan posisi gerakan yang dilakukan oleh penulis. Skenario dapat dilihat pada Tabel 5.3.

Tabel 5.3 Skenario Pengujian A2

Nama Skenario Pengujian	Pengujian Akurasi A2
Kode	SP-A2

Algoritma	Decision Tree
Jumlah Data/<i>Frame</i>	269 data/ <i>frame</i>
Penguji	Pengguna memiliki tinggi 165
Prosedur Pengujian	Pengguna melakukan uji coba 18 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 80%

Hasil yang didapatkan pada skenario A2 cukup baik namun akurasi yang didapatkan lebih rendah dibandingkan dengan skenario A1, seperti yang dapat dilihat pada Tabel 5.4. Kesalahan klasifikasi terjadi pada beberapa gerakan.

Berdasarkan skenario A2 yang dilakukan, disamping gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan (terlebih penguji belum terbiasa mempraktikkan bahasa isyarat), menurut pengamatan penulis perbedaan karakteristik tinggi badan yang signifikan antara penulis sebagai objek dalam pembuatan *dataset* dan penguji mempunyai pengaruh terhadap tingkat akurasi dari klasifikasi fitur dinamis.

Hal tersebut disebabkan karena tinggi badan pengguna turut andil dalam menentukan posisi gerakan seperti yang dijelaskan pada Subbab 4.3.2. Dan perbedaan karakteristik tinggi badan juga mempunyai efek terhadap identifikasi *skeleton joints* oleh Kinect 2.0 karena adanya perbedaan sudut Kinect 2.0 dalam mendeteksi *skeleton joints*.

5.2.3 Pengujian Skenario A3 dan Analisis

Pada pengujian skenario A3, uji coba dilakukan oleh pengguna yang mempunyai karakteristik tinggi badan 168 cm dengan menggunakan data gerakan sebanyak 269 data/*frame* hasil ekstraksi fitur dinamis dan posisi gerakan yang dilakukan oleh penulis. Skenario dapat dilihat pada Tabel 5.5.

Tabel 5.2 Terjemahan Isyarat Kata Skenario Pengujian A1

[illegible]

Tabel 5.4 Terjemahan Isyarat Kata Skenario Pengujian A2

[illegible]

Tabel 5.6 Terjemahan Isyarat Kata Skenario Pengujian A3

[illegible]

Tabel 5.5 Skenario Pengujian A3

Nama Skenario Pengujian	Pengujian Akurasi A3
Kode	SP-A3
Algoritma	Decision Tree
Jumlah Data/Frame	269 data/frame
Penguji	Pengguna memiliki tinggi 168 cm
Prosedur Pengujian	Pengguna melakukan uji coba 18 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 82,2%

Hasil yang didapatkan pada skenario A3 cukup baik. Walaupun akurasi yang didapatkan lebih rendah dibandingkan dengan skenario A1, namun skenario A3 mempunyai akurasi lebih tinggi dibandingkan skenario A2, seperti yang dapat dilihat pada Tabel 5.6. Menurut pengamatan penulis, hasil klasifikasi yang rendah terjadi karena posisi dan gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan. Disamping kurang terbiasanya pengguna menggunakan perangkat lunak yang dibangun ini.

5.2.4 Pengujian Skenario B dan Analisis

Pada pengujian skenario B, uji coba kembali dilakukan sendiri oleh penulis dengan menambahkan satu gerakan bahasa isyarat statis yaitu Gang seperti yang dapat dilihat pada Gambar 5.1. Hal ini dilakukan untuk mengetahui apakah fitur dinamis juga dapat digunakan untuk mendeteksi gerakan bahasa isyarat statis. Dengan jumlah data gerakan sebanyak 284 data/frame hasil ekstraksi menggunakan fitur dinamis yang dilakukan oleh penulis, skenario dapat dilihat pada Tabel 5.7.



Gambar 5.1 Bahasa Isyarat Statis yang Berarti Gang

Tabel 5.7 Skenario Pengujian B

Nama Skenario Pengujian	Pengujian Akurasi B
Kode	SP-B
Algoritma	Decision Tree
Jumlah <i>Dataset</i>	284 data/ <i>frame</i>
Penguji	Penulis
Prosedur Pengujian	Penulis melakukan uji coba 19 gerakan bahasa isyarat pokok (18 dinamis dan 1 statis) dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 84,2%

Hasil yang didapatkan pada skenario B cukup baik, walaupun tingkat akurasi berkurang dibandingkan dengan skenario A1 yang sama-sama dilakukan oleh penulis seperti yang dapat dilihat pada Tabel 5.8. Menurut pengamatan penulis, hasil klasifikasi yang lebih rendah tersebut disebabkan oleh menambahnya data gerakan baru yang tidak diikuti oleh penambahan data *training* di masing-masing gerakan yang sudah ada sebelumnya.

Tabel 5.8 Terjemahan Isyarat Kata Skenario Pengujian B

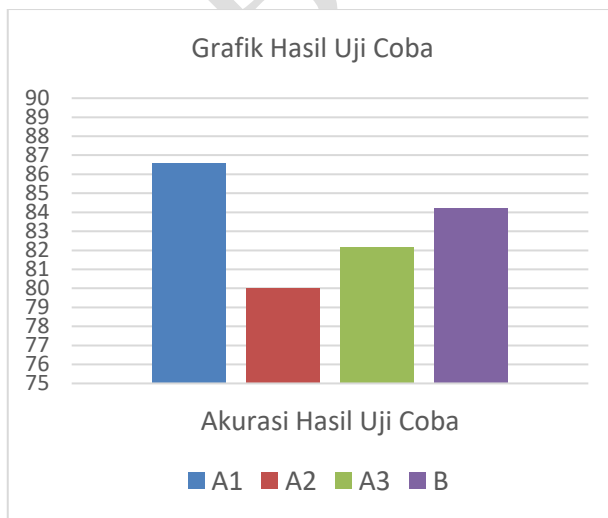
[illegible]

Di sisi lain, berdasarkan skenario B yang , penulis menarik kesimpulan bahwa gerakan isyarat statis dapat diklasifikasikan menggunakan fitur dinamis yang digunakan di dalam Tugas Akhir ini. Dengan persentase berhasil sekitar 60%, gerakan Gang tersebut juga tidak mempengaruhi klasifikasi data training yang dilakukan penulis menggunakan perangkat lunak WEKA secara signifikan.

5.3 Evaluasi

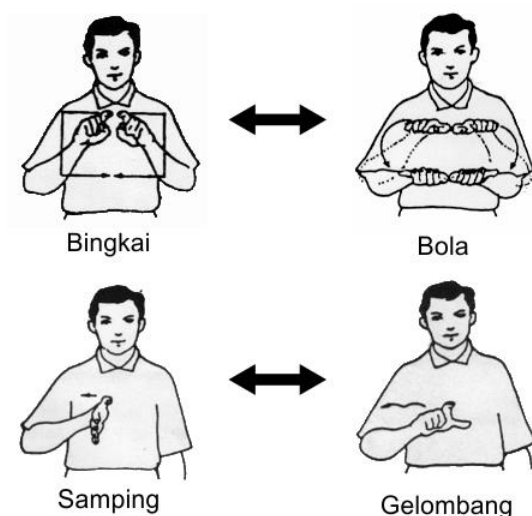
Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini evaluasi menunjukkan data hasil pengujian yang telah dipaparkan pada Subbab 5.2. Evaluasi disampaikan dalam bentuk analisis hasil secara keseluruhan sebagai berikut:

1. Klasifikasi fitur dinamis di dalam perangkat lunak yang dibuat mempunyai akurasi rata-rata yang cukup baik yaitu 82,9%. Grafik hasil akurasi dari masing-masing percobaan dapat dilihat pada Gambar 5.2.



Gambar 5.2 Grafik Tingkat Akurasi Klasifikasi Hasil Uji Coba

2. Adanya beberapa gerakan bahasa isyarat dinamis yang hampir sama antara satu dengan yang lain menyebabkan kesalahan pada proses klasifikasi. Seperti yang dapat dilihat pada Gambar 5.3, gerakan bahasa isyarat yang menunjukkan kata Bingkai mempunyai kemiripan dengan gerakan bahasa isyarat yang menunjukkan kata Bola. Kemudian gerakan bahasa isyarat yang menunjukkan kata Samping mempunyai kemiripan dengan gerakan bahasa isyarat yang menunjukkan kata Gelombang.



Gambar 5.3 Kemiripan Gerakan dari Beberapa Bahasa Isyarat

3. Selain adanya beberapa gerakan bahasa isyarat dinamis yang hampir sama antara satu dengan yang lain, sensitifitas Kinect 2.0 dalam merekam koordinat masing-masing *skeleton joints* juga mempengaruhi tingkat akurasi klasifikasi gerakan bahasa isyarat.
4. Bahasa isyarat yang tidak mempunyai kemiripan gerakan secara signifikan antara gerakan satu dengan yang lainnya tidak menutup kemungkinan menimbulkan kesalahan

klasifikasi gerakan bahasa isyarat. Analisis dapat dijelaskan sebagai berikut:

- 1) Pada gerakan topeng, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan awan dan maklum. Hal ini disebabkan karena kedua gerakan tersebut mempunyai gerakan awal yang serupa dengan gerakan topeng. Begitu pun dengan sebaliknya.
- 2) Pada gerakan badan, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan besar, sempit, bingkai, dan bola. Kesalahan klasifikasi gerakan bingkai dan bola disebabkan karena kedua gerakan tersebut mempunyai gerakan di tengah yang serupa dengan gerakan badan. Kemudian untuk gerakan besar dan sempit, kesalahan klasifikasi disebabkan karena tidak adanya *threshold* jika terjadi gerakan yang minim. Begitu pun dengan sebaliknya.
- 3) Pada gerakan awan, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan maklum. Hal ini disebabkan karena gerakan maklum mempunyai gerakan yang hampir serupa dengan gerakan awan. Begitu pun dengan sebaliknya.
- 4) Pada gerakan anak, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan samping dan gelombang. Hal ini disebabkan karena kedua gerakan tersebut mempunyai gerakan yang hampir serupa dengan gerakan anak. Begitu pun dengan sebaliknya.
- 5) Pada gerakan bola dan bingkai, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil besar. Hal ini disebabkan karena gerakan besar mempunyai gerakan awal yang hampir serupa dengan gerakan bola dan bingkai.
- 6) Pada gerakan bingung, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan awan dan topeng. Hal ini disebabkan karena kedua gerakan

tersebut mempunyai gerakan awal yang hampir serupa dengan gerakan bingung. Begitu pun dengan sebaliknya.

- 7) Pada gerakan gelombang, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan samping dan sama. Hal ini disebabkan karena kedua gerakan tersebut mempunyai gerakan awal yang hampir serupa dengan gerakan gelombang. Begitu pun dengan sebaliknya.
- 8) Pada gerakan lengkung, terdapat beberapa kesalahan klasifikasi yaitu mendapatkan hasil gerakan faedah, gelombang, dan maklum. Hal ini disebabkan karena ketiga gerakan tersebut mempunyai gerakan awal yang hampir serupa dengan gerakan lengkung. Begitu pun dengan sebaliknya.

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 yang berhasil digunakan dalam proses ekstraksi fitur dinamis berjumlah 38 fitur yaitu fitur 1 s.d. 18 merupakan ekstraksi fitur dinamis tangan kiri, fitur 19 s.d. 36 merupakan ekstraksi fitur dinamis tangan kanan, fitur 37 dan 38 merupakan posisi tangan kiri dan kanan ketika melakukan gerakan bahasa isyarat. Semua fitur tersebut didapatkan dari hasil olah data koordinat 4 *skeleton joints* yang ditentukan oleh penulis, yaitu telapak tangan kanan (HR), telapak tangan kiri (HL), leher (N), dan bagian tengah tulang belakang (SM).
2. Identifikasi posisi gerakan yang dilakukan ketika melakukan *training* dan *testing* data sangat berpengaruh terhadap akurasi klasifikasi fitur dinamis.
3. Perbedaan karakteristik tinggi badan yaitu diluar *range* antara 166 cm – 176 cm berefek 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 82,9%.

5. Walaupun akurasi yang didapatkan tidak sama baik, berdasarkan percobaan yang dilakukan, dapat disimpulkan bahwa fitur dinamis yang digunakan dalam Tugas Akhir ini juga dapat mendeteksi bahasa isyarat statis.
6. Sedikit perbedaan gerakan bahasa isyarat yang dilakukan oleh pengguna, dapat mengakibatkan kesalahan keluaran bahasa isyarat yang diinginkan.

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* dari berbagai macam pengguna yang mempunyai karakteristik tubuh dan sudut pengambilan Kinect 2.0 yang berbeda-beda.
2. Identifikasi *skeleton joints* tidak hanya dipergelangan tangan saja. Karena pada beberapa gerakan bahasa isyarat dinamis, terdapat kesamaan dalam gerakan yang dihasilkan namun berbeda jika dilihat dari bentuk tangan.
3. Menggunakan/membuat *classifier* lain dalam mengklasifikasikan fitur dinamis guna mendapatkan hasil klasifikasi gerakan bahasa isyarat dinamis yang lebih baik.
4. Melakukan normalisasi data koordinat masing-masing *skeleton joints* sebelum dilakukan proses ekstraksi fitur dinamis agar mendapatkan hasil lebih akurat.
5. Ekstraksi fitur dinamis tidak hanya terbatas pada 40 data/*frame* saja (dalam satu kali identifikasi gerakan), melainkan dapat dilakukan secara dinamis mengikuti gerakan tangan yang dibuat oleh pengguna.

DAFTAR PUSTAKA

- [1] E. Rakun, M. Andriani, I. W. Wiprayoga, K. Danniswara and A. Tjandra, "Combining depth image and skeleton data from Kinect for recognizing words in the sign system for Indonesian language (SIBI [Sistem Isyarat Bahasa Indonesia])," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Bali, 2013.
- [2] 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.
- [3] Y. A. Sutanto, "Rancang Bangun Modul Pengenalan Bahasa Isyarat Indonesia Menggunakan Teknologi Kinect dan Metode Back Propagation Genetic Algorithm Neural Network," Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [4] 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.
- [5] M. Iqbal, "Pengenalan Bahasa Isyarat Indonesia Berbasis Sensor Flex dan Accelerometer Menggunakan Dynamic Time Warping," Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [6] 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.
- [7] "Kinect," Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Kinect>. [Accessed 7 June 2016].

- [8] "Microsoft Visual Studio," Wikipedia, [Online]. Available: http://id.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 7 Juny 2016].
- [9] R. Ariadni and I. Ariesianti, "Implementasi Metode Pohon Keputusan untuk Klasifikasi Data Dengan Nilai Fitur yang Tidak Pasti," Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
- [10] M. Zbakh, Z. Haddad and J. L. Krahe, "classification, An online reversed French Sign Language dictionary based on a learning approach for signs," *Pattern Recognition Letters*, vol. 67, pp. 28-38, 2015.

LAMPIRAN A KODE SUMBER

1	private void OneTestButton_Click(object sender,
2	RoutedEventArgs e)
3	{
4	statusDetail.Content = "Testing Data";
5	statusAmbil = 2;
	}

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	namaGerakan = fileName.Text;
7	}

Kode Sumber A.2 Fungsi Start Testing

1	if (stringtangankiri == "Perut")
2	{
3	if (stringtangankan == "Kepala")
4	{
5	if (kuant2[34] <= 3)
6	{
7	if (kuant2[22] <= 4)
8	{
9	outputText.Content = "Maklum";
10	}
11	else
12	{
13	if (kuant2[14] <= 7)
14	{
15	outputText.Content = "Bingung";
16	}
17	else
18	{
19	outputText.Content = "Awan";
20	}
21	}
22	}
23	else
24	{
25	if (kuant2[34] <= 6)
26	{

27	outputText.Content = "Topeng";
28	}
29	else
30	{
31	if (kuant2[18] <= 3)
32	{
33	outputText.Content = "Maklum";
34	}
35	else
36	{
37	outputText.Content = "Awan";
38	}
39	}
40	}
41	}
42	else if (stringtangankanan == "Perut")
43	{
44	if (kuant1[14] <= 6)
45	{
46	if (kuant1[34] <= 6)
47	{
48	if (kuant1[16] <= 6)
49	{
50	outputText.Content = "Besar";
51	}
52	else
53	{
54	outputText.Content = "Badan";
55	}
56	}
57	else
58	{
59	outputText.Content = "Bola";
60	}
61	}
62	else
63	{
64	if (kuant1[34] <= 6)
65	{
66	if (kuant2[14] <= 5)
67	{
68	outputText.Content = "Sempit";
69	}
70	else
71	{
72	outputText.Content = "Badan";
73	}
74	}
75	else

76	{
77	if (kuant2[28] <= 6)
78	{
79	if (kuant2[28] <= 2)
80	{
81	outputText.Content = "Sampling";
82	}
83	else
84	{
85	if (kuant2[30] <= 5)
86	{
87	if (kuant1[26] <= 6)
88	{
89	outputText.Content =
	"Sempit";
90	}
91	else
92	{
93	outputText.Content = "Sama";
94	}
95	}
96	else
97	{
98	outputText.Content = "Sempit";
99	}
100	}
101	}
102	else
103	{
104	if (kuant2[10] <= 6)
105	{
106	if (kuant2[34] <= 4)
107	{
108	outputText.Content = "Sampling";
109	}
110	else
111	{
112	outputText.Content = "Anak";
113	}
114	}
115	else
116	{
117	outputText.Content = "Sampling";
118	}
119	}
120	}
121	}
122	}
123	else

```
124 {
125     if (kuant2[16] <= 3)
126     {
127         if (kuant2[30] <= 4)
128         {
129             outputText.Content = "Lengkung";
130         }
131         else
132         {
133             outputText.Content = "Gelombang";
134         }
135     }
136     else
137     {
138         if (kuant2[18] <= 6)
139         {
140             outputText.Content = "Faedah";
141         }
142         else
143         {
144             if (kuant2[16] <= 5)
145             {
146                 outputText.Content = "Faedah";
147             }
148             else
149             {
150                 outputText.Content = "Gelombang";
151             }
152         }
153     }
154 }
155 }
156 else if (stringtangankiri == "Dada")
157 {
158     if (kuant2[24] <= 6)
159     {
160         if (kuant2[14] <= 3)
161         {
162             outputText.Content = "Gang";
163         }
164         else
165         {
166             if (kuant2[16] <= 6)
167             {
168                 outputText.Content = "Rujuk";
169             }
170             else
171             {
172                 outputText.Content = "Gang";
```



```

173     }
174     }
175 }
176 else
177 {
178     if (kuant2[22] <= 6)
179     {
180         outputText.Content = "Gang";
181     }
182     else
183     {
184         outputText.Content = "Bingkai";
185     }
186 }
187 }
188 else
189 {
190     if (kuant2[36] <= 7)
191     {
192         if (kuant2[36] <= 3)
193         {
194             outputText.Content = "Kijang";
195         }
196         else
197         {
198             outputText.Content = "Selubung";
199         }
200     }
201     else
202     {
203         outputText.Content = "Kijang";
204     }
205 }

```

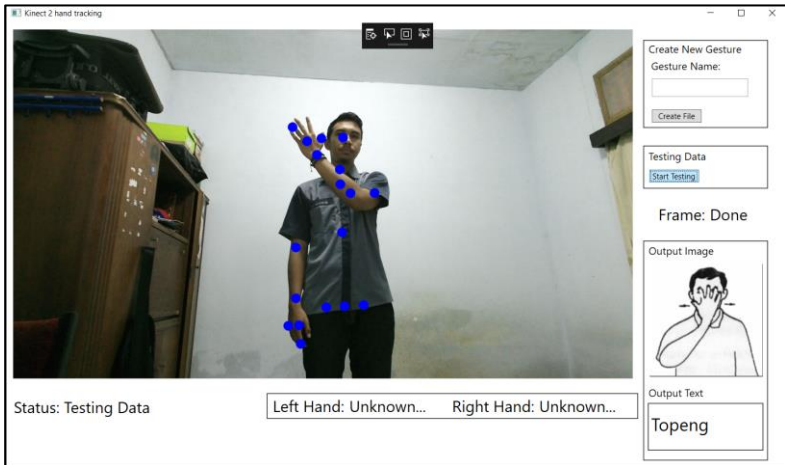
Kode Sumber A.3 Decision Tree 18 Gerakan Dinamis dan 1 Gerakan Statis

[Halaman ini sengaja dikosongkan]

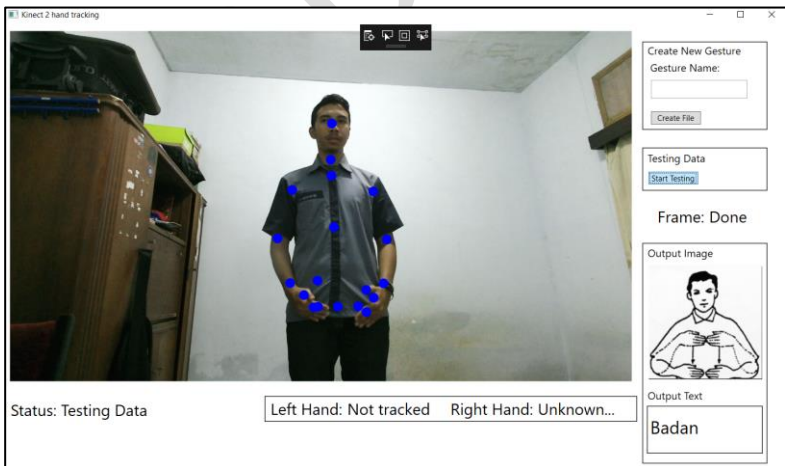
RBTC

LAMPIRAN B

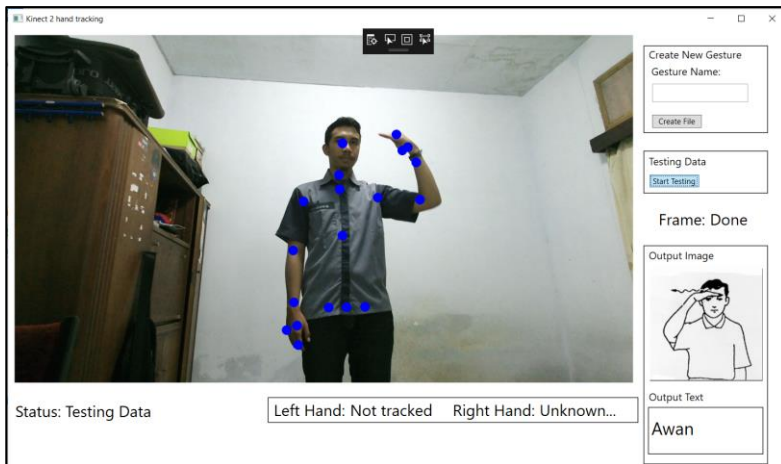
SCREENSHOT PERANGKAT LUNAK



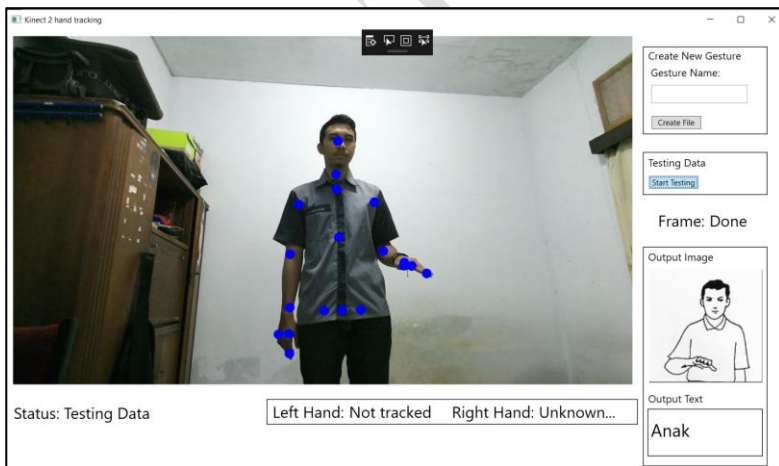
Gambar B.1 Penulis Mempraktekkan Bahasa Isyarat Topeng



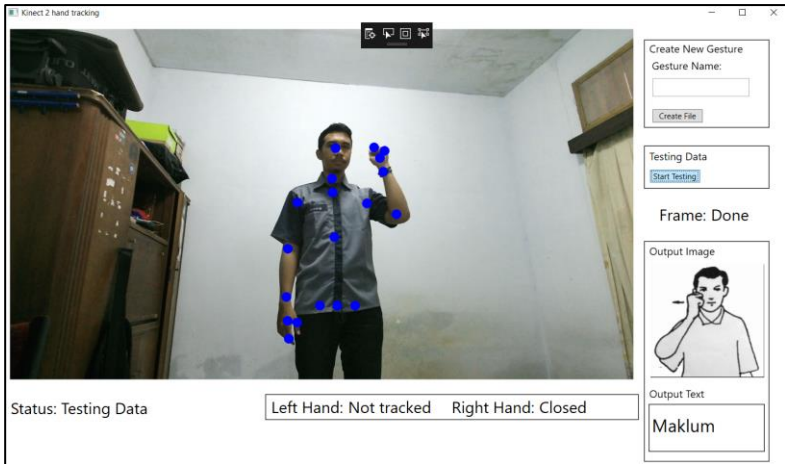
Gambar B.2 Penulis Mempraktekkan Bahasa Isyarat Badan



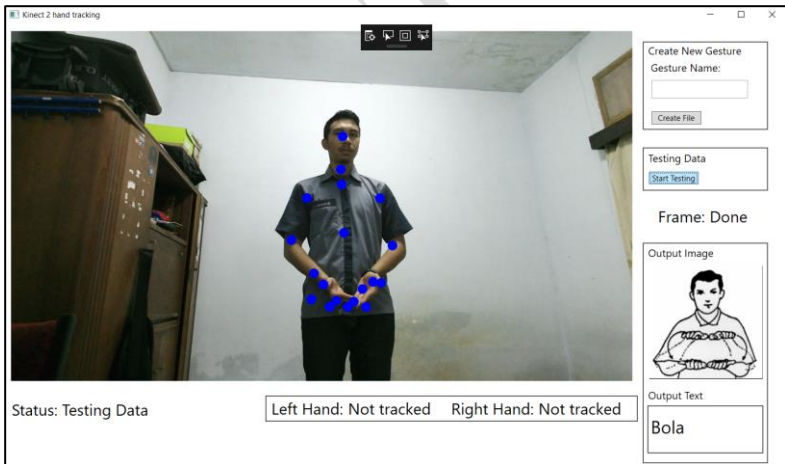
Gambar B.3 Penulis Mempraktekkan Bahasa Isyarat Awan



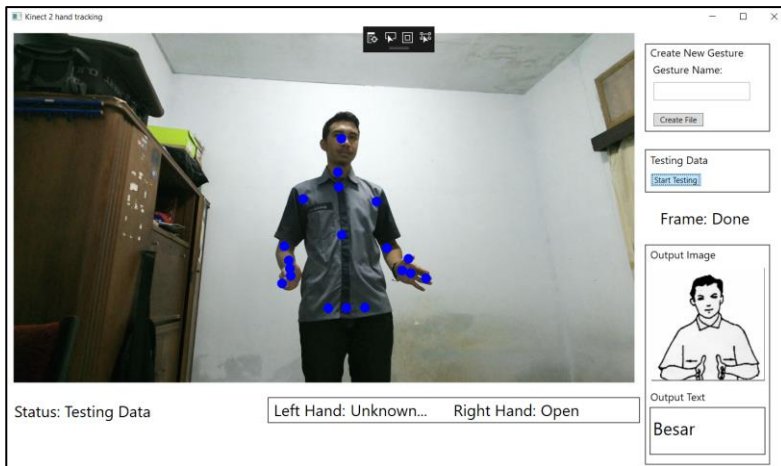
Gambar B.4 Penulis Mempraktekkan Bahasa Isyarat Anak



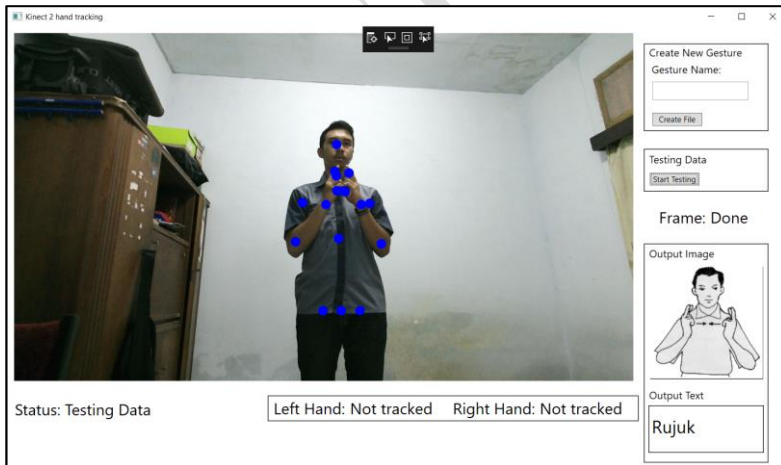
Gambar B.5 Penulis Mempraktekkan Bahasa Isyarat Maklum



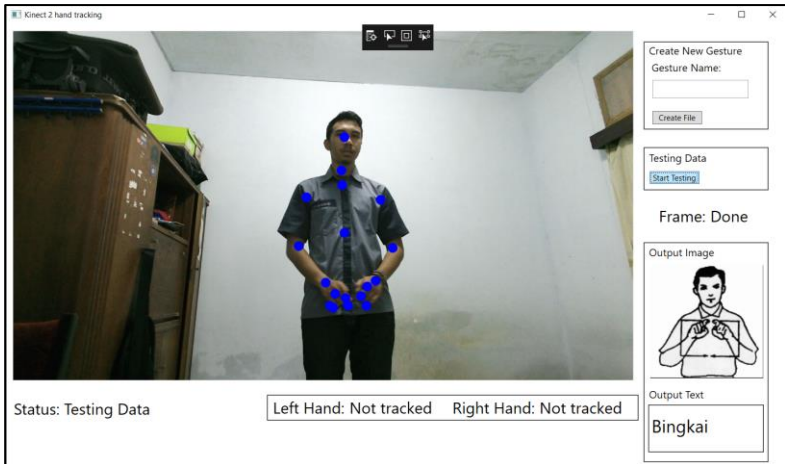
Gambar B.6 Penulis Mempraktekkan Bahasa Isyarat Bola



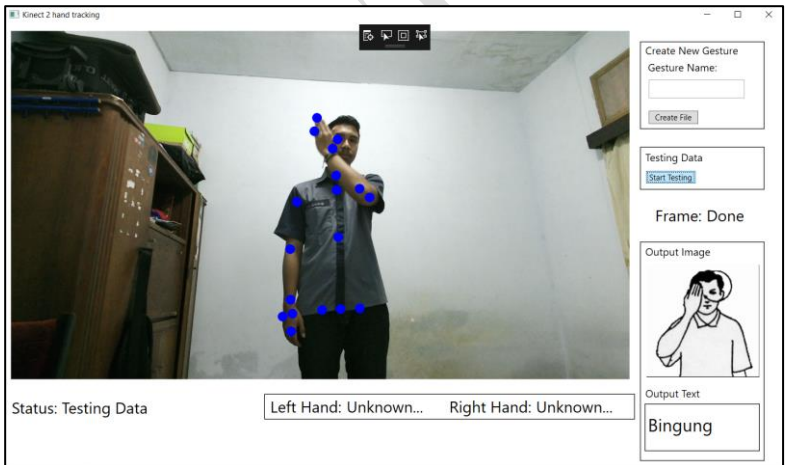
Gambar B.7 Penulis Mempraktekkan Bahasa Isyarat Besar



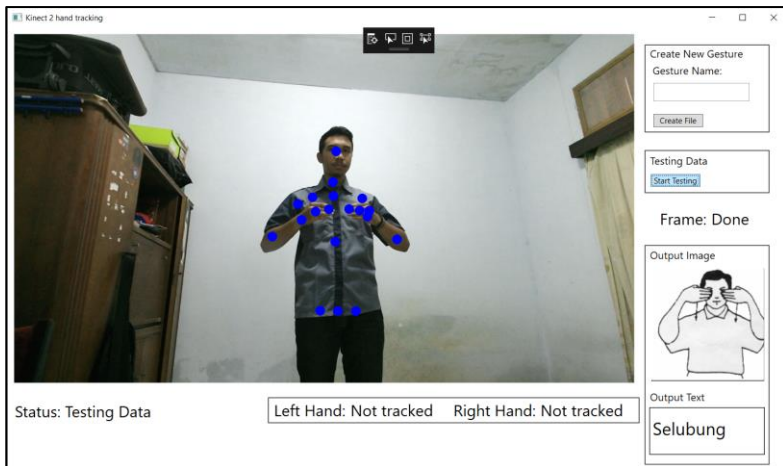
Gambar B.8 Penulis Mempraktekkan Bahasa Isyarat Rujuk



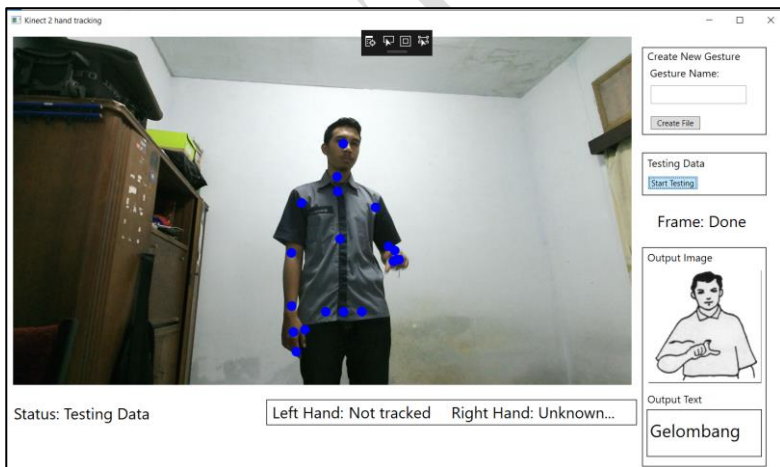
Gambar B.9 Penulis Mempraktekkan Bahasa Isyarat Bingkai



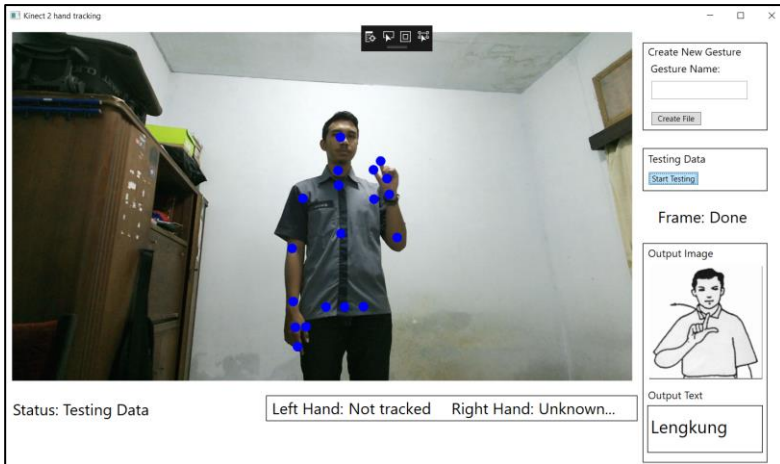
Gambar B.10 Penulis Mempraktekkan Bahasa Isyarat Bingung



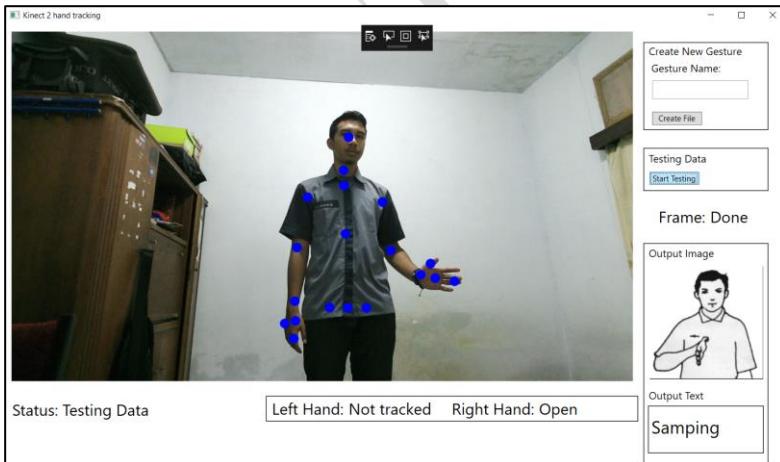
Gambar B.11 Penulis Mempraktekkan Bahasa Isyarat Selubung



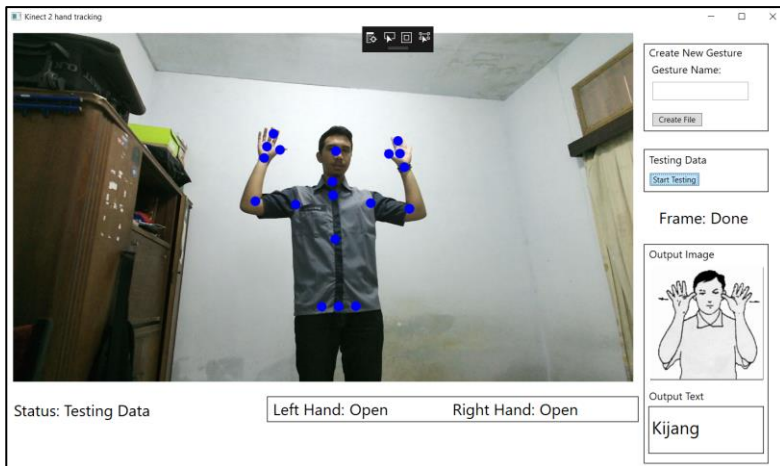
Gambar B.12 Penulis Mempraktekkan Bahasa Isyarat Gelombang



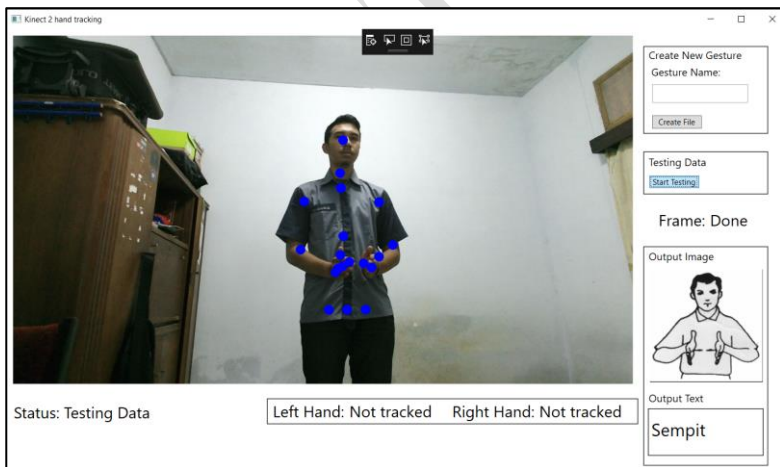
Gambar B.13 Penulis Mempraktekkan Bahasa Isyarat Lengkung



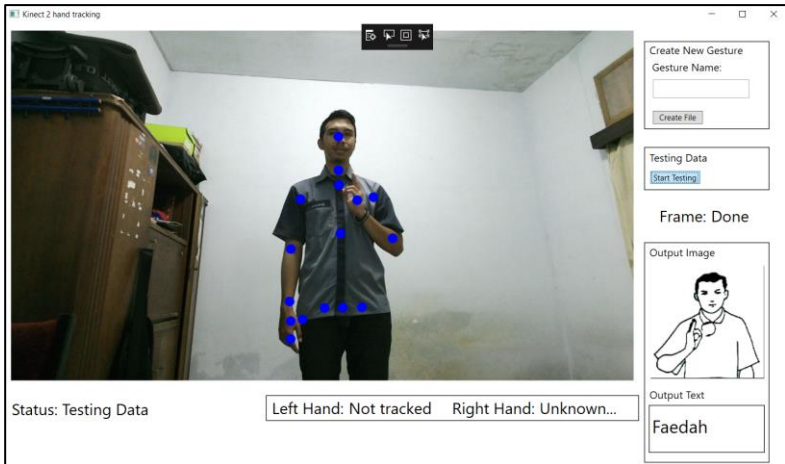
Gambar B.14 Penulis Mempraktekkan Bahasa Isyarat Samping



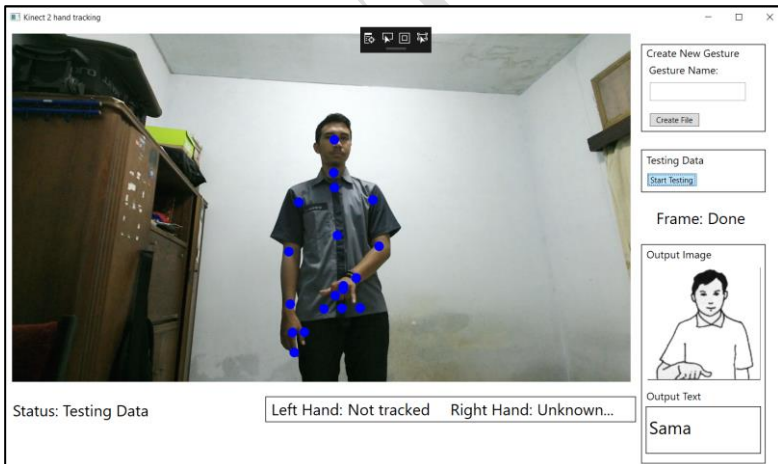
Gambar B.15 Penulis Mempraktekkan Bahasa Isyarat Kijang



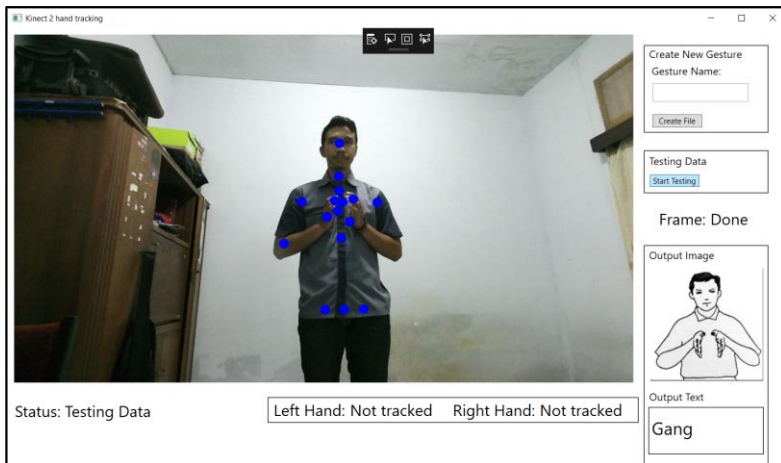
Gambar B.16 Penulis Mempraktekkan Bahasa Isyarat Sempit



Gambar B.17 Penulis Mempraktekkan Bahasa Isyarat Faedah



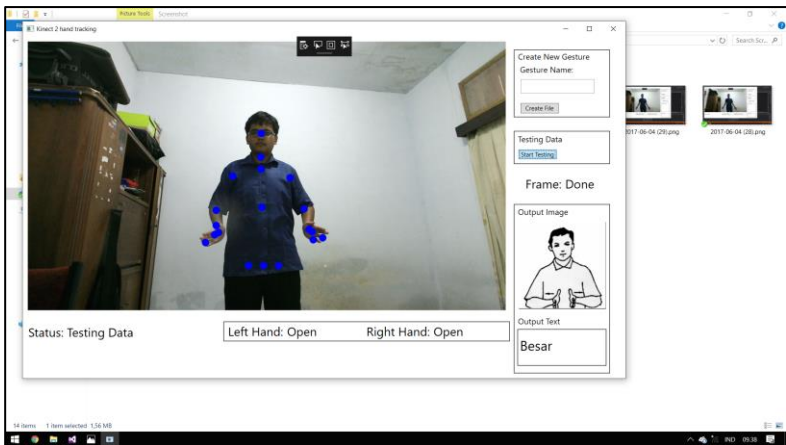
Gambar B.18 Penulis Mempraktekkan Bahasa Isyarat Sama



Gambar B.19 Penulis Mempraktekkan Bahasa Isyarat Gang



Gambar B.20 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Sama



Gambar B.21 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Besar

[Halaman ini sengaja dikosongkan]

RBTC

BIODATA PENULIS



Penulis lahir di kota Jakarta pada 13 November 1995. Merupakan anak dari dua bersaudara yang mempunyai hobi mendaki gunung dan fotografi. Penulis telah menempuh pendidikan formal yaitu TK Islam Kemerdekaan (2000 – 2001), SDN Sumur Batu 05 Pagi (2001 – 2003), SDN Pekayon 07 Pagi (2003 – 2007), SMPN 217 Jakarta (2007 – 2010), SMAN 98

Jakarta (2010 – 2013), dan mahasiswa S1 Departemen Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya rumpun mata kuliah Interaksi, Grafika, dan Seni (IGS).

Dalam dunia kampus, penulis mengikuti beberapa organisasi yaitu Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS sebagai Staf Departemen Media Informasi (2014 – 2015) serta Kepala Departemen Kewirausahaan (2015 – 2016) dan organisasi Badan Eksekutif Mahasiswa (BEM) ITS sebagai Staf Perekonomian (2014 – 2015) dan Menteri Perekonomian (2016 – 2017). Selain itu, penulis juga mengikuti beberapa kepanitiaan di ITS seperti SCHEMATICS ITS sebagai staf (2014) dan staf ahli (2015), ITS EXPO sebagai staf ahli (2015), YES Summit sebagai staf (2014) dan staf ahli (2015), PAMMITS sebagai *Organizing Committee* (2014) dan Koordinator Sie (2015), dan GERIGI ITS sebagai *Organizing Committee* (2015). Selain aktif dalam pengembangan softskill, dalam bidang akademis penulis pernah menjadi asisten dosen Interaksi Manusia dan Komputer (IMK). Penulis dapat dihubungi melalui surel n.yahyaeka@gmail.com.