



TUGAS AKHIR - KI141502

Pengenalan Bahasa Isyarat Indonesia dengan Metode *DYNAMIC TIME WARPING* (DTW) Menggunakan Kinect 2.0

TIARA ANGGITA
NRP 5114100176

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

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

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

Pengenalan Bahasa Isyarat Indonesia dengan Metode *DYNAMIC TIME WARPING* (DTW) Menggunakan Kinect 2.0

TIARA ANGGITA
NRP 5114100176

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

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

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

INDONESIAN SIGN LANGUAGE RECOGNITION USING DYNAMIC TIME WARPING (DTW) METHOD AND KINECT 2.0

TIARA ANGGITA
NRP 5114100176

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

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

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENGENALAN BAHASA ISYARAT INDONESIA DENGAN METODE *DYNAMIC TIME WARPING* (DTW) MENGUNAKAN KINECT 2.0

TUGAS AKHIR

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

Oleh:

TIARA ANGGITA

NRP. 5114100176

Disetujui oleh Pembimbing Tugas Akhir

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



**SURABAYA
JANUARI, 2018**

[Halaman ini sengaja dikosongkan]

**PENGENALAN BAHASA ISYARAT INDONESIA
DENGAN METODE *DYNAMIC TIME WARPING* (DTW)
MENGUNAKAN KINECT 2.0**

Nama Mahasiswa : Tiara Anggita
NRP : 5114100176
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing I : Wijayanti Nurul Khotimah, S.Kom.,
M.Sc.
Dosen Pembimbing II : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

ABSTRAK

Bahasa isyarat memungkinkan tuna rungu dan tuna wicara untuk saling berkomunikasi. Meski begitu, bahasa isyarat tidak dipahami oleh sebagian besar masyarakat umum. Perkembangan teknologi untuk menerjemahkan bahasa isyarat Indonesia secara real-time bisa bermanfaat bagi komunitas tuna rungu.

*Pada penelitian sebelumnya, pengenalan bahasa isyarat menggunakan teknologi Kinect sudah berhasil dibuat, namun penggunaan fiturnya hanya sebatas bahasa isyarat dengan gerakan statis saja. Selain itu, terdapat penelitian sebelumnya yang menggunakan ekstraksi fitur hanya untuk mengenali bahasa isyarat dinamis saja. Pada Tugas Akhir ini, penulis mengusulkan sebuah metode yang mampu mengenali secara real-time bahasa isyarat Indonesia dengan ekstraksi fitur untuk mengenali bahasa isyarat yang bersifat statis dan dinamis yang kemudian dicocokkan dengan menggunakan metode *Dynamic Time Warping* (DTW)*

Hasil pengujian dalam Tugas Akhir ini menunjukkan bahwa metode DTW yang digunakan untuk mengenali gerakan isyarat mencapai akurasi rata-rata 93%. Hasil tersebut masih dapat ditingkatkan dengan menambahkan data training maupun menggabungkan metode lain dalam mengenali bahasa isyarat.

Kata kunci – *Bahasa isyarat Indonesia, Dynamic Time Warping, Kinect 2.*

[Halaman ini sengaja dikosongkan]

INDONESIAN SIGN LANGUAGE RECOGNITION USING DYNAMIC TIME WARPING (DTW) METHOD AND KINECT 2.0

Name : Tiara Anggita
NRP : 5114100176
Major : Informatics Department, FTIK-ITS
Advisor I : Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Advisor II : Dr.Eng. Nanik Suciati, S.Kom., M.Kom.

ABSTRACT

Sign language allows hearing and speech impaired people to communicate each other. Nevertheless, sign language is not understood by most of common people. Technological development of a real-time Indonesian sign language translator could be beneficial to the speech impaired community.

In previous research, sign language recognition using Kinect technology has been done, but only limited to sign language with static movement only. There was also a previous research using feature extraction in sign language recognition, but only used for dynamic movement. In this final project, the author proposes a method capable of recognize in real-time a basic list of Indonesian sign language using feature extraction that can be use for both static and dynamic movement which then matched by using Dynamic Time Warping (DTW) method.

The test results of this final project show that the DTW method for gesture recognition achieve a mean accuracy of 93%. These results can still be improved by adding training data or using/combining other method in recognizing sign languages.

Keywords – Indonesian sign language, Dynamic Time Warping, Kinect 2.

[Halaman ini sengaja dikosongkan]

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 “Pengenalan Bahasa Isyarat Indonesia dengan Metode *Dynamic Time Warping* (DTW) menggunakan Kinect 2.0”. Penyusunan tugas akhir ini diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana di Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi 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. Allah SWT.
2. Keluarga yang selalu memberi dukungan penuh dan semangat untuk menyelesaikan Tugas Akhir ini.
3. 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.
4. Bapak dan ibu dosen Departemen Informatika ITS yang banyak memberikan ilmu serta bimbingan selama masa perkuliahan.
5. Mas Jono yang sudah meminjamkan Kinect kepada penulis
6. Sahabat-sahabat, yaitu Kania, Nay, Hilman, Vivi, Fikry, Gleen yang selalu memberikan canda tawa dan tidak hentinya memberi warna di masa perkuliahan penulis.

7. Teman-teman “Lodgiwetan” yang selalu menyemangati dan memberikan pengalaman dan cerita baru di akhir perkuliahan penulis.
8. Senior-senior, terutama Saddam, Lino, Nanang, Rifqi, Mas Fakhri, Mas Harry, dan Yahya yang telah banyak membantu dan mengajarkan banyak hal, serta memberikan pencerahan selama masa perkuliahan.
9. Teman-teman Admin MI yang sering penulis repotkan dan selalu menemani dikala suntuk ketika mengerjakan tugas-tugas perkuliahan.
10. Seluruh teman-teman TC14 yang sering penulis repotkan di Surabaya.
11. 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, Oktober 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	4
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Metodologi	4
1.7 Sistematika Penulisan	5
BAB II DASAR TEORI.....	7
2.1 Tuna Rungu.....	7
2.2 Bahasa Isyarat	7
2.3 Kinect 2.0.....	8
2.4 Kinect SDK.....	8
2.5 Microsoft Visual Studio	8
2.6 Euclidean Distance.....	9
2.7 Dynamic Time Warping (DTW).....	9
2.8 Rata-rata.....	13
2.9 Median	13
2.10 Modus.....	13
2.11 Varian	14
2.12 Standar Deviasi.....	14
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15
3.1 Analisis Perangkat Lunak	15
3.1.1 Deskripsi Umum Perangkat Lunak	16
3.1.2 Spesifikasi Kebutuhan Perangkat Lunak.....	18

3.1.3 Identifikasi Pengguna	19
3.2 Perancangan Perangkat Lunak	19
3.2.1 Model Kasus Penggunaan	19
3.2.2 Definisi Aktor	20
3.2.3 Definisi Kasus Penggunaan	20
3.2.4 Arsitektur Umum Sistem	23
3.2.5 Rancangan Antarmuka Perangkat Lunak	24
3.2.6 Rancangan Proses Perangkat Lunak	25
BAB IV IMPLEMENTASI	33
4.1 Lingkungan Pembangunan	33
4.1.1 Lingkungan Pembangunan Perangkat Keras	33
4.1.2 Lingkungan Pembangunan Perangkat Lunak	33
4.2 Implementasi Antarmuka	34
4.3 Implementasi Perangkat Lunak	37
4.3.1 Implementasi Pendeteksian <i>Skeleton</i> Pengguna	37
4.3.2 Implementasi Proses Ekstraksi Fitur	40
4.3.3 Implementasi Proses <i>Testing Data</i>	46
BAB V PENGUJIAN DAN EVALUASI	49
5.1 Lingkungan Pembangunan	49
5.2 Skenario Pengujian	49
5.2.1 Pengujian Skenario A dan Analisis	51
5.2.2 Pengujian Skenario B dan Analisis	51
5.2.3 Pengujian Skenario C dan Analisis	53
5.3 Evaluasi	57
BAB VI KESIMPULAN DAN SARAN	61
6.1 Kesimpulan	61
6.2 Saran	62
DAFTAR PUSTAKA	63
LAMPIRAN A KODE SUMBER	65
LAMPIRAN B <i>SCREENSHOT</i> PERANGKAT LUNAK	69
BIODATA PENULIS	81

DAFTAR GAMBAR

Gambar 1.1 Contoh Gerakan Bahasa Isyarat yang Mempunyai Arti: (a) Marah; (b) Ibu; (c) Maaf	2
Gambar 2.1 Kinect 2.0 [7].....	8
Gambar 2.2 Penyelarasan Antara Dua Urutan <i>Time-dependent</i> [10]	10
Gambar 2.3 Gerakan Bahasa Isyarat yang Digunakan	11
Gambar 2.4 Gerakan Bahasa Isyarat yang Digunakan	12
Gambar 3.1 Gerakan Bahasa Isyarat yang Digunakan	17
Gambar 3.2 Diagram Kasus Penggunaan Perangkat Lunak.....	20
Gambar 3.3 Arsitektur Umum Sistem.....	24
Gambar 3.4 Rancangan Antarmuka Perangkat Lunak	25
Gambar 3.5 <i>Joint</i> yang Digunakan [10]	26
Gambar 3.6 Diagram Alir Implementasi DTW	29
Gambar 3.7 Rancangan Proses <i>Training</i> Data	31
Gambar 3.8 Rancangan Proses <i>Testing</i> Data.....	31
Gambar 4.1 Antarmuka Perangkat Lunak	34
Gambar 5.1 Grafik Tingkat Akurasi Hasil Uji Coba.....	57
Gambar 5.2 Kemiripan Gerakan dari Beberapa Gerakan Isyarat	58
Gambar 5.3 Gerakan yang Memiliki Posisi Awal Serupa.....	59
Gambar B.1 Penulis Mempraktekkan Bahasa Isyarat Anak.....	69
Gambar B.2 Penulis Mempraktekkan Bahasa Isyarat Anda.....	69
Gambar B.3 Penulis Mempraktekkan Bahasa Isyarat Apa	70
Gambar B.4 Penulis Mempraktekkan Bahasa Isyarat Bagaimana	70
Gambar B.5 Penulis Mempraktekkan Bahasa Isyarat Laki-laki..	71
Gambar B.6 Penulis Mempraktekkan Bahasa Isyarat Jumpa.....	71
Gambar B.7 Penulis Mempraktekkan Bahasa Isyarat Baik.....	72
Gambar B.8 Penulis Mempraktekkan Bahasa Isyarat Salam	72
Gambar B.9 Penulis Mempraktekkan Bahasa Isyarat Marah.....	73
Gambar B.10 Penulis Mempraktekkan Bahasa Isyarat Keluarga	73
Gambar B.11 Penulis Mempraktekkan Bahasa Isyarat Jahat	74
Gambar B.12 Penulis Mempraktekkan Bahasa Isyarat Halo	74
Gambar B.13 Penulis Mempraktekkan Bahasa Isyarat Sekolah .	75

Gambar B.14 Penulis Mempraktekkan Bahasa Isyarat Masalah.	75
Gambar B.15 Penulis Mempraktekkan Bahasa Isyarat Saya	76
Gambar B.16 Penulis Mempraktekkan Bahasa Isyarat Rumah...	76
Gambar B.17 Penulis Mempraktekkan Bahasa Isyarat Kerja	77
Gambar B.18 Penulis Mempraktekkan Bahasa Isyarat Saudara .	77
Gambar B.19 Penulis Mempraktekkan Bahasa Isyarat Ibu.....	78
Gambar B.20 Penulis Mempraktekkan Bahasa Isyarat Pagi	78
Gambar B.21 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Halo	79
Gambar B.22 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Halo	79

DAFTAR TABEL

Tabel 3.1 Definisi Aktor.....	20
Tabel 3.2 Definisi Kasus Penggunaan.....	21
Tabel 3.3 Spesifikasi Kasus Penggunaan Membuat Data Bahasa Isyarat Baru	21
Tabel 3.4 Spesifikasi Kasus Penggunaan <i>Training Dataset</i>	22
Tabel 5.1 Skenario Pengujian A.....	51
Tabel 5.3 Skenario Pengujian B	52
Tabel 5.5 Skenario Pengujian C	53
Tabel 5.2 Terjemahan Isyarat Kata Skenario Pengujian A	54
Tabel 5.4 Terjemahan Isyarat Kata Skenario Pengujian B.....	55
Tabel 5.6 Terjemahan Isyarat Kata Skenario Pengujian C.....	56

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Tampilan Perangkat Lunak	37
Kode Sumber 4.2 Kode Sumber Deteksi <i>Skeleton</i> Pengguna	40
Kode Sumber 4.3 Kode Sumber Ekstraksi Fitur	43
Kode Sumber 4.4 Kode Sumber Ekstraksi Fitur dan Penyimpanan Data Hasil Ekstrasi Fitur <i>Skeleton</i> Pengguna	45
Kode Sumber 4.5 Implementasi DTW untuk Pengenalan Gerakan Tangan	48
Kode Sumber A.1 Fungsi <i>Start Testing</i>	65
Kode Sumber A.2 Fungsi <i>Create File</i>	66
Kode Sumber A.3 <i>Testing Data</i>	67

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

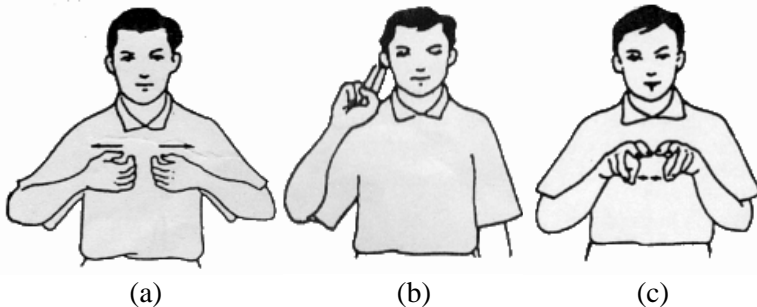
Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, rumusan, batasan permasalahan, manfaat, manfaat, dan metodologi.

1.1 Latar Belakang

Kekurangan pada pendengaran sering berdampak pada kemampuan verbal pada orang dengan gangguan pendengaran, sehingga mereka menggunakan bahasa isyarat dan bahasa tubuh untuk berkomunikasi. Bahasa isyarat merupakan media bagi para penderita tuna rungu dan tuna wicara untuk berkomunikasi dengan sekitarnya. Penderita tuna rungu dan tuna wicara mengalami kesulitan dalam berkomunikasi dengan orang normal. Hal ini dikarenakan ketidakpahaman orang normal dengan system isyarat ini [1]. Dengan demikian, proses pertukaran informasi sulit terjadi. Dimana kita tahu bahwa informasi tidak dapat terlepas dari diri kita. Untuk itu dibutuhkan sistem untuk menerjemahkan bahasa isyarat ke dalam Bahasa Indonesia agar dapat tercipta komunikasi yang lebih baik.

Bahasa isyarat adalah bahasa yang mengutamakan komunikasi manual, bahasa tubuh, dan gerak bibir untuk berkomunikasi. Penyandang tuna rungu adalah kelompok utama yang menggunakan bahasa ini, biasanya mengkombinasikan bentuk tangan, orientasi dan gerak tangan, lengan dan tubuh, serta ekspresi wajah untuk mengungkapkan pikiran mereka [2]. Bahasa Isyarat sangat dipengaruhi oleh latar belakang budaya dan kebiasaan dimana orang tersebut tinggal dan berasal. Di Indonesia, bahasa isyarat diterapkan dalam dua bentuk, yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Bahasa Isyarat Indonesia adalah sistem komunikasi yang praktis dan efektif untuk penyandang tuna rungu Indonesia yang telah dikembangkan oleh kaum tuna rungu, sedangkan Sistem Bahasa Isyarat Indonesia (SIBI) adalah sistem hasil rekayasa dan ciptaan

dari orang normal untuk berkomunikasi dengan penyandang difabel tuna rungu dan bukan berasal dari penyandang difabel tuna rungu. Dalam kehidupan sehari-hari, penderita tuna rungu wicara berkomunikasi dengan bahasa isyarat yang mengacu pada SIBI [1]. Contoh bahasa isyarat SIBI dapat dilihat pada Gambar 1.1.



Gambar 1.1 Contoh Gerakan Bahasa Isyarat yang Mempunyai Arti: (a) Marah; (b) Ibu; (c) Maaf

Bahasa Isyarat tidak memiliki lingkup pengguna yang besar seperti bahasa lisan, sehingga tidak banyak orang yang dapat mengenali atau mengerti bahasa isyarat yang disampaikan lawan bicaranya. Hal ini menimbulkan kebutuhan alat bantu untuk mempelajari bahasa isyarat dengan mudah.

Pada tahun 2010, Microsoft meluncurkan teknologi baru berupa perangkat keras sensor Kinect. Pada awalnya, teknologi tersebut ditujukan sebagai konsol Xbox 360 sehingga pemain dapat menggunakan gerakan tubuhnya sebagai pengendali permainan. Teknologi sensor Kinect ini dapat dikembangkan lebih lanjut agar dapat digunakan untuk mengenali bahasa isyarat [3].

Penelitian mengenai pengenalan bahasa isyarat Indonesia masih terbatas dan masih membutuhkan pengembangan. Sebelumnya, sudah ada penelitian yang dibuat oleh Wijayanti Nurul Khotimah tentang pengenalan bahasa isyarat Indonesia menggunakan Kinect 1.0. Dalam penelitian tersebut, bahasa yang dapat dideteksi, yaitu Alquran, Bentuk, Gang, Hai, Hamba, Hormat, Ketua, dan Wadah. Bahasa yang dapat dikenali dalam

penelitian ini hanya berupa gerakan statis dan kosa katanya pun masih terbatas, sehingga perlu ditelusuri lebih lanjut [4].

Selain itu, penelitian sebelumnya mengenai pengenalan bahasa isyarat Indonesia juga pernah dilakukan oleh Eka. Penelitian tersebut membahas tentang bagaimana mengekstraksi fitur untuk mengenali bahasa isyarat dinamis. Sehingga fitur bahasa isyarat yang dapat dikenali pada penelitian tersebut masih terbatas pada bahasa isyarat yang bersifat dinamis saja. Hasil rata-rata akurasi yang didapatkan pada penelitian tersebut, yaitu 82,9% [5].

Dalam pengenalan bahasa isyarat, terdapat beberapa metode yang digunakan untuk menangani gerakan dinamis, yaitu *Artificial Neural Networks*, *Hidden Markov Models*, dan *Dynamic Time Warping* (DTW). DTW diperkenalkan untuk komunitas *data mining* oleh Berndt and Clifford [6]. Metode ini digunakan untuk menghitung kesamaan antara dua data deret waktu (*time series*) yang dapat berbeda dalam waktu dan kecepatan [3].

Dalam Tugas Akhir ini, akan dikembangkan suatu program aplikasi untuk pembelajaran bahasa isyarat dengan menggunakan sensor Kinect untuk mendeteksi gerakan tangan dan kemudian membandingkan pola gerakan tangan tersebut dengan data-data referensi menggunakan metode *Dynamic Time Warping* (DTW).

1.2 Rumusan Masalah

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

1. Bagaimana menentukan titik yang representatif untuk bahasa isyarat?
2. Bagaimana mengimplementasikan algoritma *Dynamic Time Warping* (DTW) berdasarkan titik *skeleton* yang sesuai?
3. Apakah posisi pengguna berpengaruh terhadap akurasi pada saat pengenalan gerakan?
4. Bagaimana tinggi badan pengguna mempengaruhi identifikasi posisi gerakan?

5. Apakah perbedaan posisi Kinect dengan pengguna mempengaruhi akurasi pada saat pengenalan gerakan?

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. Posisi titik *skeleton* yang dilacak berjumlah 12 posisi, yaitu kepala, bahu kiri-tengah-kanan, siku kanan-kiri, pergelangan tangan kanan-kiri, punggung, telapak tangan kanan-kiri, dan pinggul
3. Gerakan tangan yang digunakan tidak melibatkan pergerakan jari tangan

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini, yaitu untuk mengimplementasikan algoritma *Dynamic Time Warping* (DTW) untuk mengenali bahasa isyarat menggunakan Kinect 2.0.

1.5 Manfaat

Manfaat yang diperoleh dari pembuatan Tugas Akhir ini adalah untuk mengenali bahasa isyarat yang diharapkan dapat membantu orang berkebutuhan khusus, yaitu tuna rungu dan tuna wicara, dalam berkomunikasi dengan orang yang tidak memiliki kebutuhan khusus.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan menggunakan metodologi sebagai berikut:

- A. Studi literatur
Tugas Akhir ini akan menggunakan literatur paper yang berasal dari jurnal internasional bereputasi, yaitu IEEE.

Selain itu, akan digunakan sejumlah referensi yang diperlukan dalam pembuatan aplikasi, yaitu mengenai *Dynamic Time Warping* (DTW) yang berasal dari buku, internet, ataupun materi dalam suatu mata kuliah yang berhubungan dengan metode yang akan digunakan.

- B. Perancangan perangkat lunak
Analisa dimulai dari pengguna yang memberikan masukan berupa gerakan tangan dan posisi gerakan tersebut dari badan ke alat Kinect 2.0. Kemudian, dari masukan tersebut diproses dengan menghitung ekstraksi fitur yang telah ditentukan. Setelah itu, dilakukan pengenalan gerakan tangan dengan menggunakan metode *Dynamic Time Warping* (DTW). Dan pada akhirnya, keluaran dari proses tersebut ditampilkan menggunakan aplikasi perangkat lunak berbasis *desktop*.
- C. Implementasi dan pembuatan sistem
Aplikasi ini dibangun menggunakan perangkat lunak Bahasa Pemrograman C#, *Integrated Development Environment* (IDE) Microsoft Visual Studio, dan Kinect 2.0
- D. Uji coba dan evaluasi
Pengujian akan dilakukan oleh tiga 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, *Dynamic Time Warping* (DTW), dan pengetahuan umum mengenai bahasa isyarat.

2.1 Tuna Rungu

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*) [7]. Tuli adalah keadaan dimana organ pendengaran telah mengalami kerusakan yang sangat parah dan mengakibatkan tidak berfungsinya pendengaran. Sedangkan kurang dapat mendengar adalah keadaan dimana organ pendengaran mengalami kerusakan tetapi masih dapat berfungsi untuk mendengar.

2.2 Bahasa Isyarat

Bahasa isyarat adalah 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 [8], yaitu:

1. Isyarat Pokok: melambangkan sebuah kata atau konsep;
2. Isyarat Tambahan: melambangkan awalan, akhiran, dan partikel (imbuhan);
3. Isyarat Bentuk: dibentuk dengan menggabungkan isyarat pokok dan isyarat tambahan;
4. Abjad Jari: dibentuk dengan jari-jari untuk mengeja huruf.

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 2.1, terdapat tiga lensa yaitu kamera RGB yang digunakan untuk menangkap spektrum warna, *infrared emitters* yang memproyeksikan spektrum inframerah dan sensor kedalaman yang menghasilkan gambar mendalam dari seseorang atau objek dengan menganalisis informasi inframerah. Dan sebuah *microphone array* yang dapat menemukan lokasi timbulnya suara. Alhasil, ada enam sumber data yang dihasilkan, termasuk warna, inframerah, kedalaman, indeks tubuh, tubuh, dan suara [9].



Gambar 2.1 Kinect 2.0 [7]

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 [10].

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* [11].

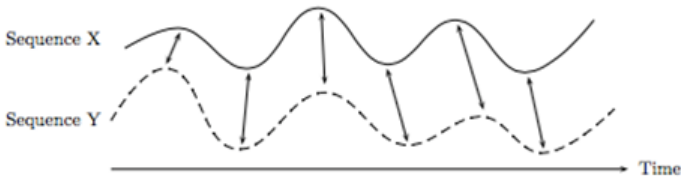
2.6 Euclidean Distance

Euclidean Distance adalah perhitungan jarak dari dua buah titik dalam *Euclidean space*. Perhitungan ini biasanya diterapkan pada dua dimensi dan tiga dimensi. Pada sistem tiga dimensi, *Euclidean Distance* dihitung berdasarkan persamaan (2.1), dimana d merupakan jarak Euclidean, sedangkan p dan q merupakan titik yang ingin dihitung jaraknya dari masing-masing koordinat x , y , dan z .

$$d = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \quad (2.1)$$

2.7 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) merupakan teknik yang terkenal untuk mencari penyelarasan yang optimal antara dua urutan yang diberikan (*time-dependent*) di bawah batasan tertentu. Penyelarasan antara dua urutan dapat dilihat pada Gambar 2.2. Algoritma ini juga disebut sebagai *non-linear sequence alignment*, sehingga algoritma ini lebih realistis untuk digunakan dalam mengukur kemiripan suatu pola (*pattern/template matching*) ketimbang hanya menggunakan algoritma pengukuran linier seperti *Euclidean Distance*, *Manhattan*, *Canberra*, *Mexican Hat*, dan lain-lain. Algoritma ini secara luas digunakan untuk aplikasi pengenalan suara (*speech recognition*), pengenalan tulisan tangan dan tandatangan, *data mining*, pengklusteran, pengolahan isyarat, musik, dan masih banyak lagi. Dalam bidang seperti pengumpulan data dan pengambilan informasi, DTW telah berhasil diterapkan untuk mengatasi deformasi waktu secara otomatis dan kecepatan yang berbeda yang terkait dengan data *time-dependent* [6].



Gambar 2.2 Penyelarasan Antara Dua Urutan *Time-dependent* [10]

Asumsikan kita mempunyai dua deret numerik sebagai berikut (a_1, a_2, \dots, a_n) dan (b_1, b_2, \dots, b_m) . Algoritma memulai dengan kalkulasi jarak lokal antara elemen dari dua deret dengan menggunakan tipe jarak yang berbeda. Metode yang paling sering digunakan untuk kalkulasi jarak adalah rumus *Euclidean Distance* yang menghasilkan matriks dengan n baris dan m kolom. Kalkulasi jarak lokal dihitung dengan mencari nilai absolut dari selisih kedua deret data yang dirumuskan seperti pada persamaan (2.2).

$$c_{ij} = |a_i - b_j|, i = 1 \text{ s. d. } n, j = 1 \text{ s. d. } m \quad (2.2)$$

Memulai dengan matriks jarak lokal, langkah selanjutnya merupakan nilai-nilai yang harus kita lewati untuk menemukan *warping path*. Terdapat tiga rumus yang digunakan untuk mengisi matriks DTW, yaitu rumus untuk mencari baris pertama, kolom pertama, dan elemen lainnya. Untuk menghitung elemen baris pertama pada matriks DTW dirumuskan seperti pada persamaan (2.3).

$$D_{(1,j)} = D_{(1,j-1)} + c_{(1,j)} \quad (2.3)$$

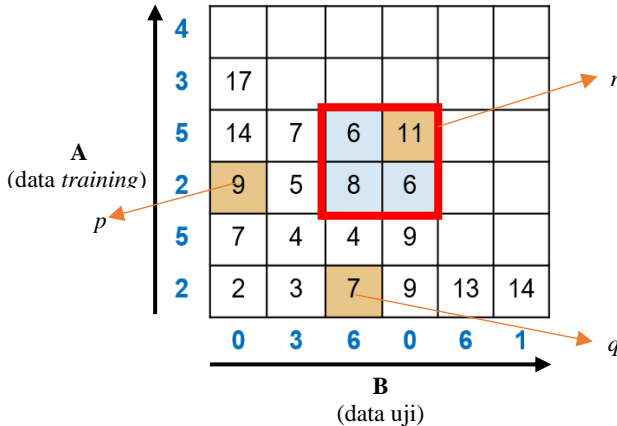
Selanjutnya, untuk menghitung kolom pertama pada matriks DTW dirumuskan pada persamaan (2.4).

$$D_{(i,1)} = D_{(i,j-1)} + c_{(i,1)} \quad (2.4)$$

Untuk menghitung elemen lainnya pada matriks DTW, dirumuskan seperti pada persamaan (2.5), dimana D_{ij} adalah jarak minimal antara deret bagian (a_1, a_2, \dots, a_i) dan (b_1, b_2, \dots, b_j) .

$$D_{ij} = c_{ij} + \min(D_{i-1,j-1}, D_{i-1,j}, D_{i,j-1}) \quad (2.5)$$

Sebuah *warping path* adalah jalur lintasan atau *path* yang melalui matriks yang berisi jarak minimal dari elemen D_{ij} hingga elemen D_{nm} yang terdiri dari elemen-elemen D_{ij} itu sendiri. Contoh matriks DTW dapat dilihat pada Gambar 2.3.



Gambar 2.3 Gerakan Bahasa Isyarat yang Digunakan

Terdapat matriks 6×6 antara dua deret data yaitu data *training* $A = \{2, 5, 2, 5, 3, 4\}$ dan data uji $B = \{0, 3, 6, 0, 6, 1\}$. Untuk menghitung elemen matriks, terdapat tiga rumus yang digunakan untuk mengisi seluruh elemen matriks, yaitu seperti berikut:

- Baris pertama, digunakan rumus sesuai dengan persamaan (2.3). Sebagai contoh, untuk mengisi elemen p dilakukan perhitungan sebagai berikut:

$$\begin{aligned}
 p &= D_{(1-1,3)} + c_{(1,3)} \\
 p &= D_{(0,3)} + |a_1 - b_3| \\
 p &= 7 + |2 - 0| \\
 p &= 7 + 2 \\
 p &= 9
 \end{aligned}$$

- Kolom pertama, digunakan rumus sesuai dengan persamaan (2.4). Sebagai contoh, mendapatkan elemen q , dilakukan perhitungan sebagai berikut:

$$q = D_{(3,1-1)} + c_{(3,1)}$$

$$q = D_{(3,0)} + |a_3 - b_1|$$

$$q = 3 + |2 - 6|$$

$$q = 3 + 4$$

$$q = 7$$

- c. Elemen lain dalam matriks, gunakan rumus pada persamaan (2.5). Contoh, untuk mengisi elemen r dilakukan perhitungan sebagai berikut:

$$r = c_{4,4} + \min(D_{4-1,4-1}, D_{4-1,4}, D_{4,4-1})$$

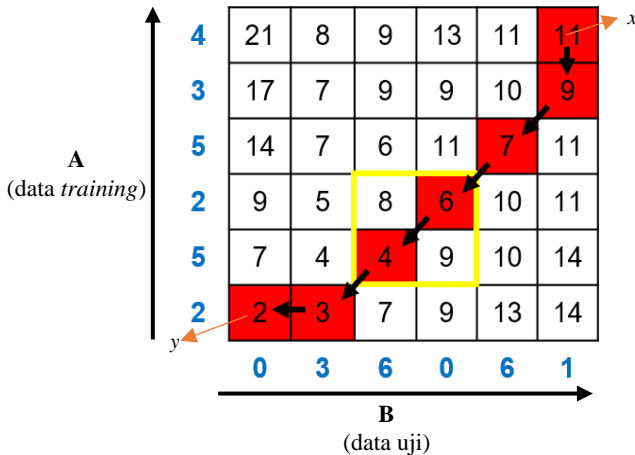
$$r = |a_4 - b_4| + \min(D_{3,3}, D_{3,4}, D_{4,3})$$

$$r = |5 - 0| + \min(8, 6, 6)$$

$$r = 5 + 6$$

$$r = 11$$

Setelah semua elemen matriks terisi, selanjutnya dihitung nilai *optimal warping cost* dengan menghitung total dari *cost* jalur yang dilewati. Contoh *warping path* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Gerakan Bahasa Isyarat yang Digunakan

Untuk menghitung *warping path*, dilakukan perhitungan yang dimulai dari elemen matriks paling akhir, yaitu x . Kemudian, dijumlahkan sesuai dengan rumus pada persamaan (2.5) sampai dengan elemen matriks yang paling awal, yaitu y . Jalur pembengkokan ditandai dengan elemen-elemen yang berwarna merah.

2.8 Rata-rata

Nilai rata-rata atau *mean* merupakan jumlah dari sekelompok data dibagi banyaknya data. Untuk mencari rata-rata pada suatu data digunakan rumus pada persamaan (2.6).

$$\text{Rata - rata} = \frac{\text{Jumlah data}}{\text{Banyak data}} \quad (2.6)$$

2.9 Median

Median merupakan nilai pertengahan dari suatu kelompok data yang telah diurutkan menurut besarnya. Median biasanya dilambangkan dengan *Me*. Jika data ganjil, maka rumus mediannya dapat dilihat pada persamaan (2.7), dimana n merupakan banyaknya data dan X merupakan nomor urut data.

$$Me = X_{(n+1)/2} \quad (2.7)$$

Jika banyak datanya genap maka nilai median dapat dihitung dengan persamaan (2.8).

$$Me = \frac{\left(X_{\frac{n}{2}} + X_{\left(\frac{n}{2}\right)+1} \right)}{2} \quad (2.8)$$

2.10 Modus

Modus atau *mode* adalah penjelasan tentang suatu kelompok data dengan menggunakan nilai yang sering muncul dalam kelompok data tersebut. Jika dalam suatu kelompok data memiliki lebih dari satu nilai data yang sering muncul, maka sekumpulan

data tersebut memiliki lebih dari satu modus. Modus biasanya dilambangkan dengan Mo .

2.11 Varian

Varian adalah ukuran seberapa tersebar data. Varian yang rendah menandakan data yang berkelompok dekat satu sama lain. Varian yang tinggi menandakan data yang lebih tersebar. Konsep ini memiliki banyak kegunaan di dalam statistik. Varian juga berguna saat membuat model statistik, karena varian yang rendah menandakan data yang *over-fitting*. Nilai varian (S^2) dapat dihitung dengan persamaan (2.9), dimana n merupakan banyaknya data, x_i merupakan nomor urut data ke- i , dan \bar{x} merupakan rata-rata.

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (2.9)$$

2.12 Standar Deviasi

Standar deviasi adalah nilai statistik yang digunakan untuk menentukan bagaimana sebaran data dalam sampel, dan seberapa dekat titik data individu ke mean atau rata-rata nilai sampel. Sebuah standar deviasi dari kumpulan data sama dengan nol menunjukkan bahwa semua nilai-nilai dalam himpunan tersebut adalah sama. Sebuah nilai deviasi yang lebih besar akan memberikan makna bahwa titik data individu jauh dari nilai rata-rata. Untuk menghitung standar deviasi (S), digunakan perhitungan pada persamaan (2.10), dimana n merupakan banyaknya data, x_i merupakan nomor urut data ke- i , dan \bar{x} merupakan rata-rata.

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (2.10)$$

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 tuna rungu. Walaupun agak sulit untuk mengartikan isyarat yang diberikan, hal tersebut telah membantu penderita tuna rungu 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 disimpan ke dalam sebuah file ekstensi .csv untuk dijadikan *training* data.

Untuk tahap *testing* data, setelah perangkat lunak mengekstraksi fitur yang didapatkan dari kalkulasi *skeleton* pengguna, fitur tersebut akan dicocokkan dengan *training* data yang telah dibuat sebelumnya menggunakan metode *Dynamic Time Warping* (DTW). Setelah seluruh tahap sudah selesai dilakukan, perangkat lunak akan memberikan keluaran berupa arti bahasa isyarat Indonesia yang dimaksud oleh pengguna dalam bentuk tulisan dan gambar.

3.1.1 Deskripsi Umum Perangkat Lunak

Tugas Akhir yang dibangun ini adalah sebuah modul pengenalan bahasa isyarat Indonesia dengan menggunakan teknologi Kinect 2.0. 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.

Pada Tugas Akhir ini terdapat 20 bahasa isyarat pokok yang digunakan seperti yang dapat dilihat pada Gambar 3.1. Bahasa isyarat tersebut kemudian dikelompokkan berdasarkan enam semantik kategori:

1. Makna kata ganti orang, yang terdiri dari dua gerakan
2. Makna kata tanya, yang terdiri dari dua gerakan
3. Makna bangunan, yang terdiri dari dua gerakan
4. Makna keluarga, yang terdiri dari lima gerakan
5. Makna sapaan, yang terdiri dari empat gerakan
6. Makna lain-lain, yang terdiri dari lima gerakan

1					
2			3		
4					
5					
6					

Gambar 3.1 Gerakan Bahasa Isyarat yang Digunakan

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) Menentukan ekstraksi fitur
Perangkat lunak dapat mendeteksi bagian-bagian dari *skeleton* pengguna yang akan diekstraksi menjadi fitur data guna melakukan proses pengenalan gerakan.
- c) Menerjemahkan bahasa isyarat
Perangkat lunak dapat menerjemahkan bahasa isyarat Indonesia yang dihasilkan melalui proses ekstraksi fitur *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 perangkat lunak (baik itu pengambilan data maupun uji coba data) yang sempurna, jarak optimal Kinect 2.0 dengan pengguna adalah antara 1 sampai 1,5 meter.

3.1.3 Identifikasi Pengguna

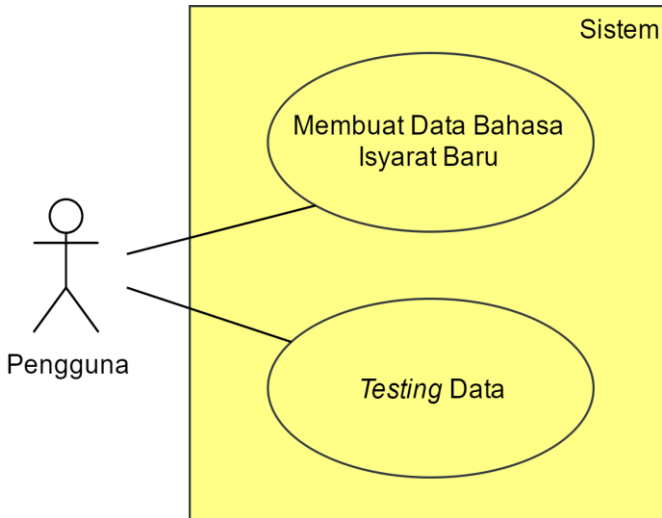
Dalam Tugas Akhir yang dibangun ini, pengguna yang akan terlibat dalam menjalankan perangkat lunak hanya satu orang saja, yaitu orang yang akan melakukan pengenalan bahasa isyarat Indonesia.

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 3.2.



Gambar 3.2 Diagram Kasus Penggunaan Perangkat Lunak

3.2.2 Definisi Aktor

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

Tabel 3.1 Definisi Aktor

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

3.2.3 Definisi Kasus Penggunaan

Pada Gambar 3.2 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 3.2.

Tabel 3.2 Definisi Kasus Penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-01	Membuat Data Bahasa Isyarat Baru	Pengguna membuat data bahasa isyarat 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 3.3.

Tabel 3.3 Spesifikasi Kasus Penggunaan Membuat Data Bahasa Isyarat Baru

Nama Kasus Penggunaan	Membuat Data Bahasa Isyarat Baru
Nomor	UC-01
Deskripsi	Kasus penggunaan aktor untuk membuat data bahasa isyarat baru
Aktor	Pengguna
Kondisi Awal	Pengguna sudah menjalankan aplikasi dan perangkat Kinect 2.0 telah tersambung
Alur Normal	1. Pengguna memasukkan nama bahasa isyarat yang akan dibuat di dalam <i>textbox</i> perangkat lunak

	<ol style="list-style-type: none"> 2. Pengguna menekan tombol “Create File” di dalam perangkat lunak 3. Perangkat lunak menerima inputan dari Kinect 2.0 dan ketika <i>skeleton</i> pengguna ditemukan 4. Pengguna melakukan gerakan bahasa isyarat yang akan dibuat 5. Perangkat lunak mengekstrak data <i>skeleton</i> pengguna sebanyak 39 data/<i>frame</i> untuk dikalkulasi 6. Perangkat lunak menyimpan hasil ekstraksi ke dalam sebuah berkas berekstensi .csv
Alur Alternatif	A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna <ol style="list-style-type: none"> 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 3.4.

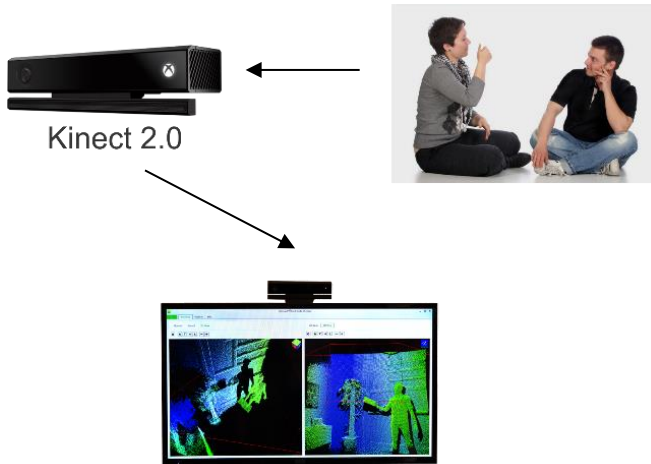
Tabel 3.4 Spesifikasi Kasus Penggunaan *Training Dataset*

Nama Kasus Penggunaan	<i>Testing Data</i>
Nomor	UC-02
Deskripsi	Kasus penggunaan aktor untuk melakukan <i>testing</i> data 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 <ol style="list-style-type: none"> 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 39 data/frame untuk dikalkulasi 5. Hasil ekstraksi data diklasifikasi menggunakan DTW yang sudah dibuat di dalam perangkat lunak 6. Perangkat lunak menampilkan isyarat bahasa Indonesia hasil klasifikasi
Alur Alternatif	<ol style="list-style-type: none"> A1. Kinect 2.0 tidak menemukan <i>skeleton</i> pengguna <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 3.3.



Gambar 3.3 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 3.4.



Gambar 3.4 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

Ekstraksi fitur diperlukan untuk meningkatkan kecepatan dan akurasi pada saat pengenalan gerakan. Selain itu, fitur data juga merepresentasikan nilai-nilai penting yang terdapat pada sebuah gerakan. Proses ekstraksi fitur dilakukan dengan menghitung jarak antara *joint* telapak tangan kanan terhadap *joint* lain. Keseluruhan *joint* yang digunakan terdapat pada Gambar 3.5.



Gambar 3.5 Joint yang Digunakan [10]

Dari perhitungan tersebut akan diambil nilai minimum pada masing-masing perhitungan jarak, maka akan didapatkan 45 elemen yang dapat digunakan sebagai fitur data. Elemen-elemen tersebut dibagi menjadi empat kelompok dengan rincian sebagai berikut:

1. Kelompok 1, ekstraksi fitur yang merepresentasikan posisi telapak tangan terhadap bidang vertikal:
 - a. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* kepala.
 - b. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* kepala.
 - c. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* pinggul.
 - d. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* pinggul.
2. Kelompok 2, ekstraksi fitur yang merepresentasikan posisi telapak tangan terhadap *joint* lain:

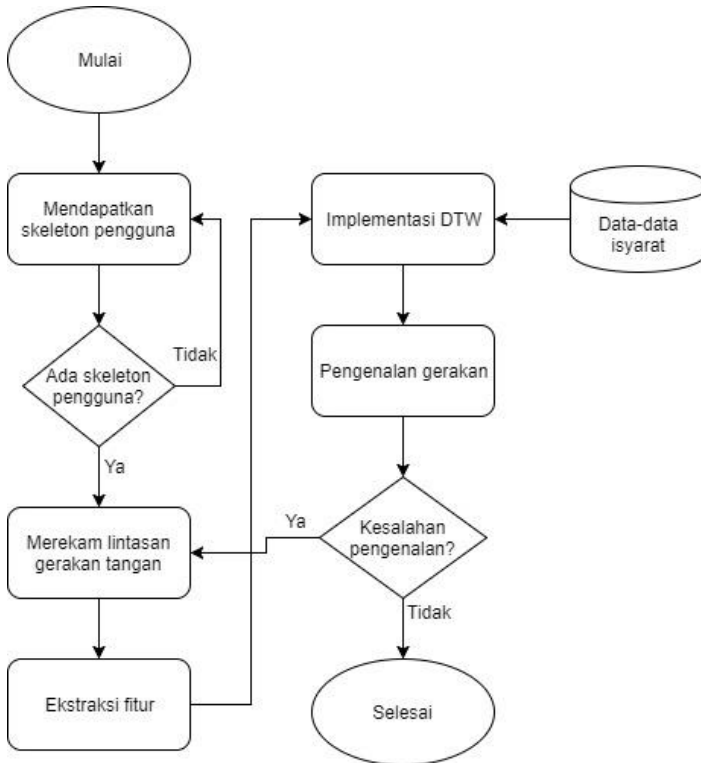
- a. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* bahu kanan.
- b. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* bahu kiri.
- c. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* bahu tengah.
- d. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* siku kanan.
- e. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* siku kiri.
- f. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* pergelangan tangan kiri.
- g. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* pergelangan tangan kanan.
- h. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kanan dengan *joint* tulang belakang.
- i. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* bahu kanan.
- j. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* bahu kiri.
- k. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* bahu tengah.
- l. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* siku kanan.
- m. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* siku kiri.
- n. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* pergelangan tangan kanan.
- o. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* pergelangan tangan kiri.
- p. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* tulang belakang.
- q. 1 nilai minimum perhitungan jarak antara *joint* telapak tangan kiri dengan *joint* telapak tangan kiri.

3. Kelompok 3, ekstraksi fitur yang merepresentasikan fitur statistik dari fitur yang ada pada kelompok 1.
Ada lima fitur statistik yang dihitung, yaitu rata-rata, median, modus, varian, dan standar deviasi. Untuk menghitung kelima fitur tersebut, dilakukan perhitungan berdasarkan data dari kelompok 1, dengan penjabaran seperti berikut:
 - a. 4 nilai minimum perhitungan rata-rata dari data kelompok 1.
 - b. 4 nilai minimum perhitungan median dari data kelompok 1.
 - c. 4 nilai minimum perhitungan modus dari data kelompok 1.
 - d. 4 nilai minimum perhitungan varians dari data kelompok 1.
 - e. 4 nilai minimum perhitungan standar deviasi dari data kelompok 1.
4. Kelompok 4, ekstraksi fitur yang merepresentasikan posisi telapak tangan terhadap sumbu Z (kedalaman).
Untuk mendapatkan nilai kedalaman minimum, dilakukan perhitungan jarak Euclidean kedalaman atau *depth* (sumbu Z) yang didapat dari data awal pada saat gerakan direkam dengan rincian sebagai berikut:
 - a. 1 nilai Z minimum jarak antara *joint* telapak tangan kanan dan *joint* kepala.
 - b. 1 nilai Z minimum jarak antara *joint* telapak tangan kanan dan *joint* pinggul.
 - c. 1 nilai Z minimum jarak antara *joint* telapak tangan kiri dan *joint* kepala.
 - d. 1 nilai Z minimum jarak antara *joint* telapak tangan kiri dan *joint* pinggul.

Dari hasil perhitungan empat kelompok di atas, maka didapatkan fitur data dengan total sebanyak 45 data. Fitur tersebut yang kemudian dijadikan data *training* pada saat pembuatan data dan data uji ketika melakukan pengujian data.

3.2.6.2 Rancangan Proses Implementasi DTW untuk Menentukan Pengenalan Gerakan Tangan

Proses implementasi algoritma DTW dilakukan dengan cara membandingkan fitur data *training* dengan data uji, hasilnya yaitu berupa data jarak antara data uji terhadap semua data *training* yang telah disimpan. Kemudian dilakukan pemilihan nilai jarak yang paling kecil yang dianggap sebagai gerakan yang paling mirip dengan gerakan masukan. Rancangan proses implementasi DTW untuk menentukan pengenalan gerakan tangan dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Alir Implementasi DTW

1. Perekaman gerakan oleh Kinect 2.0

Dalam perekaman gerakan yang dilakukan oleh Kinect 2.0 gerakan yang terekam kemudian diolah menjadi fitur data. Perangkat lunak mengambil total 39 data/frame untuk dikalkulasi dan dilakukan implementasi DTW untuk mengidentifikasi posisi gerakan yang dilakukan.

2. Penentuan *skeleton joints* yang digunakan

Terdapat 12 *skeleton joints* yang diolah dalam Tugas Akhir ini. Masing-masing *skeleton joints* diinterpretasikan dalam bentuk koordinat X, Y, dan Z. Dua belas *skeleton joints* tersebut adalah:

1. Telapak tangan kanan (HR)
2. Telapak tangan kiri (HL)
3. Leher
4. Bagian tengah tulang belakang
5. Pergelangan tangan kanan
6. Pergelangan tangan kiri
7. Siku kanan
8. Siku kiri
9. Bahu kanan
10. Bahu kiri
11. Kepala
12. Pinggul Tengah

Seluruh *skeleton joints* digunakan untuk mendapatkan fitur data. Data koordinat *skeleton joints* HR dan HL digunakan sebagai parameter untuk mendapatkan data pada saat proses ekstraksi fitur.

3. Implementasi DTW dan pengenalan gerakan tangan

Dalam proses pengenalan gerakan tangan, perangkat lunak terlebih dahulu mencocokkan data uji dengan data *training* yang sudah ada dengan metode DTW. Ada 20 gerakan yang dilakukan pada saat pembuatan data, dengan masing-masing kata dilakukan 15 kali gerakan. Sehingga, total gerakan yang disimpan pada saat *training* data ada sebanyak 300 gerakan.

Pada Tugas Akhir ini, masukan gerakan yang dikenali sebagai gerakan pada referensi merupakan gerakan yang memiliki *warping path* keseluruhan yang paling kecil dari gerakan lainnya.

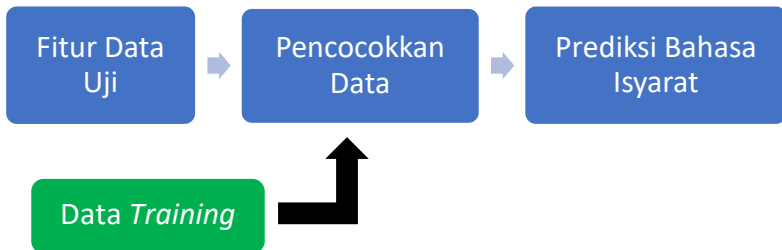
3.2.6.3 Rancangan Proses *Training* Data



Gambar 3.7 Rancangan Proses *Training* Data

Seperti yang dapat dilihat pada Gambar 3.7, proses *training* data dilakukan dengan menyimpan fitur data yang didapatkan pada Subbab 3.2.6.1 ke dalam sebuah berkas berekstensi .csv. Ada 20 gerakan yang dilakukan pada saat pembuatan data, dengan masing-masing kata dilakukan 15 kali gerakan. Sehingga, total gerakan yang disimpan pada saat *training* data ada sebanyak 300 gerakan.

3.2.6.4 Rancangan Proses *Testing* Data



Gambar 3.8 Rancangan Proses *Testing* Data

Seperti yang dapat dilihat pada Gambar 3.8, proses *testing* data dilakukan dengan mendapatkan fitur data uji yang dilakukan secara *real-time*. Kemudian dilakukan pencocokkan antara hasil dari fitur data tersebut dengan fitur data yang didapatkan pada proses *training* data di Subbab 3.2.6.3 menggunakan metode DTW. Gerakan yang dikenali sebagai gerakan pada referensi merupakan

gerakan yang memiliki nilai *warping path* paling kecil. Keluaran hasil klasifikasi yang dilakukan berupa prediksi gerakan bahasa isyarat yang dimaksud oleh pengguna dalam bentuk gambar dan juga tulisan bahasa isyarat.

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

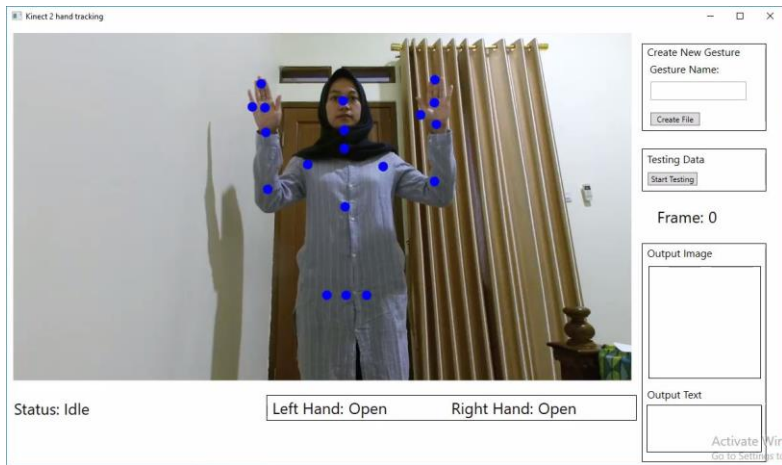
4.1.2 Lingkungan Pembangunan Perangkat Lunak

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

- Microsoft Visual Studio 2012
- 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	<code><Window x:Class="Kinect2DTW.MainWindow"</code>
2	<code>xmlns="http://schemas.microsoft.com/winfx/2006/xaml</code>
3	<code>/presentation"</code>
4	<code>xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"</code>
5	<code>Title="Kinect 2 hand tracking" Height="710"</code>
6	<code>Width="1200" Loaded="Window_Loaded"</code>
7	<code>Closed="Window_Closed"></code>
8	
9	<code><Grid Margin="0,0,2,0"></code>
10	<code><Border BorderBrush="Black" BorderThickness="1"</code>
11	<code>HorizontalAlignment="Left" Height="39"</code>
12	<code>Margin="395,561,0,0" VerticalAlignment="Top"</code>
13	<code>Width="563"/></code>
14	

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

```

64         Margin="973,49,0,0" VerticalAlignment="Top"
65         Width="115" FontSize="16"/>
66     <TextBox Name="fileName" HorizontalAlignment="Left"
67         Height="28" Margin="979,84,0,0"
68         TextWrapping="Wrap" VerticalAlignment="Top"
69         Width="146" FontSize="16"/>
70     <Button Name="createButton" Content="Create File"
71         HorizontalAlignment="Left"
72         Margin="979,131,0,0" VerticalAlignment="Top"
73         Width="75" Click="DtwCreateClick"/>
74
75     <Label Name="outputimage" Content="Output Image"
76         HorizontalAlignment="Left"
77         Margin="969,329,0,0" VerticalAlignment="Top"
78         Width="152" FontSize="16"/>
79     <Border HorizontalAlignment="Left" Height="171"
80         Margin="976,366,0,0" VerticalAlignment="Top"
81         Width="171" >
82         <Image x:Name="outputImage"
83             Stretch="UniformToFill" Margin="0,-
84             3,0.2,2.2"/>
85     </Border>
86
87     <Label Name="label8" Content="Output Text"
88         HorizontalAlignment="Left"
89         Margin="969,544,0,0" VerticalAlignment="Top"
90         Width="152" FontSize="16"/>
91     <Label Name="outputText" Content=""
92         HorizontalAlignment="Left"
93         Margin="973,585,0,0" VerticalAlignment="Top"
94         Height="48" Width="174" FontSize="10"
95         RenderTransformOrigin="0.981,2.1"/>
96     <Label Name="status" Content="Status:"
97         HorizontalAlignment="Left" Margin="6,560,0,0"
98         VerticalAlignment="Top" Width="100"
99         FontSize="24" Height="47"/>
100    <Label Name="statusDetail" Content="Kinect Not
101        Connected" HorizontalAlignment="Left"
102        Height="47" Margin="82,560,0,0"
103        VerticalAlignment="Top" Width="341"
104        FontSize="24"/>
105    <Label Name="lefthand" Content="Left Hand:"
106        HorizontalAlignment="Left"
107        Margin="399,559,0,0" VerticalAlignment="Top"
108        Width="119" FontSize="24" Height="47"/>
109    <Label Name="tblLeftHandState" Content="Not
110        Detected" HorizontalAlignment="Left"
111        Height="47" Margin="515,559,0,0"
112

```

113	VerticalAlignment="Top" Width="187"
114	FontSize="24"/>
115	<Label Name="righthand" Content="Right Hand:"
116	HorizontalAlignment="Left"
117	Margin="671,559,0,0" VerticalAlignment="Top"
118	Width="144" FontSize="24" Height="47"/>
119	<Label Name="tblRightHandState" Content="Not
120	Detected" HorizontalAlignment="Left"
121	Height="47" Margin="803,559,0,0"
122	VerticalAlignment="Top" Width="187"
123	FontSize="24"/>
124	<Label Name="label12" Content="Frame:"
125	HorizontalAlignment="Left"
126	Margin="984,268,0,0" VerticalAlignment="Top"
127	Width="144" FontSize="24" Height="47"/>
128	<Label Name="ambilData" Content="Done"
129	HorizontalAlignment="Left" Height="47"
130	Margin="1062,268,-57.4,0"
131	VerticalAlignment="Top" Width="187"
132	FontSize="24"/>
133	</Grid>
134	</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 2012 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.

Untuk melakukan pengenalan gerakan tangan, perangkat lunak mendeteksi tubuh pengguna terlebih dahulu. Ketika tubuh


```

33     Joint elbowRight =
body.Joints[JointType.ElbowLeft];
34     Joint elbowLeft =
body.Joints[JointType.ElbowRight];
35     Joint wristRight =
body.Joints[JointType.WristLeft];
36     Joint wristLeft =
body.Joints[JointType.WristRight];
37     Joint handRight =
body.Joints[JointType.HandRight];
38     Joint handLeft =
body.Joints[JointType.HandLeft];
39     Joint head =
body.Joints[JointType.Head];

40
41     foreach (Joint joint in
body.Joints.Values)
42     {
43         if (joint.TrackingState ==
TrackingState.Tracked)
44         {
45             // 3D space point
46             CameraSpacePoint
jointPosition =
joint.Position;

47
48             // 2D space point
49             Point point = new Point();
50             ColorSpacePoint colorPoint =
_sensor.CoordinateMapper
.MapCameraPointToColorSpace
(jointPosition);
51             point.X = float
.IsInfinity(colorPoint.X)
? 0 : colorPoint.X;
52             point.Y = float
.IsInfinity(colorPoint.Y)
53             ? 0 : colorPoint.Y;
54
55             Ellipse ellipse = new Ellipse
56             {
57                 Fill = Brushes.Blue,
58                 Width = 30,
                    Height = 30
59             };
60             Canvas.SetLeft(ellipse,
point.X
- ellipse.Width / 2);
61             Canvas.SetTop(ellipse,

```

62	point.Y
63	- ellipse.Height / 2);
64	canvas.Children.Add(ellipse);
65	}
66	}
67	}
68	}
69	}
70	}

Kode Sumber 4.2 Kode Sumber Deteksi *Skeleton* Pengguna

4.3.2 Implementasi Proses Ekstraksi Fitur

Pada Kode Sumber 4.3 dijelaskan mengenai proses menentukan fitur pada setiap lintasan gerakan tangan yang dilakukan oleh pengguna sesuai dengan kelompok 1 dan 2 berdasarkan rancangan pada Subbab 3.2.6.1. Penjelasan kode sumber tersebut adalah sebagai berikut:

1. Variable *i* menunjukkan *data/frame* ke-*i* yang sedang diolah
2. Variable *j* menunjukkan baris *skeleton* pengguna
3. Variable *a* menunjukkan perhitungan jarak Euclidean antara *joint* telapak tangan dan *joint* lain
4. Variable *listResultBaris* menunjukkan hasil perhitungan akar dari variable *a*
5. Variable *listResult* menunjukkan hasil perhitungan dari *listResultBaris*
6. Variable *feature2* menunjukkan data yang ada dalam setiap baris *j*
7. Variable *minimum* menunjukkan data yang paling minimum yang ada dalam setiap baris *j*
8. Variable *avg* menunjukkan perhitungan rata-rata pada setiap baris *j*
9. Variable *sum* menunjukkan penjumlahan pada data di setiap baris *j* dikurangi dengan variable *avg*
10. Variable *variance* menunjukkan perhitungan varians


```

50         listResultBaris.Add(Math.Sqrt(resultkiri));
51         double resultheadkanan =
           Math.Pow((Convert.ToDouble(column[32])
           - Convert.ToDouble(column[17])), 2);
52         listResultBaris.Add(Math.Sqrt(resultheadkanan));
53         double resultheadkiri =
           Math.Pow((Convert.ToDouble(column[35])
           - Convert.ToDouble(column[17])), 2);
54         listResultBaris.Add(Math.Sqrt(resultheadkiri));
55     }
56 }
57 listResult.Add(listResultBaris);
58 }
59 }
60 string Euclidean = "E:\\Kuliah\\SEMESTER
           7\\TA\\Dataset\\EuclideanKanan.csv";
61 using (StreamWriter writer = new
           StreamWriter(Euclidean))
62 {
63     StringBuilder line = new StringBuilder();
64     foreach (List<double> subList in listResult)
65     {
66         foreach (double item in subList)
67         {
68             line.Append(string.Format("{0:0.000}", item)
           + ",");
69         }
70         line.AppendLine();
71     }
72     writer.Write(line);
73 }
74 return Euclidean;
75 }
76 }

```

Kode Sumber 4.3 Kode Sumber Ekstraksi Fitur

```

1 public int calc_feature(String resultPath)
2 {
3     Dictionary<int, List<double>> feature2 = new
           Dictionary<int, List<double>>();
4     Dictionary<int, List<double>> minimum = new
           Dictionary<int, List<double>>();
5     using (var reader = new StreamReader(@resultPath))
6     {
7         List<string> listA = new List<string>();

```

```

8         while (!reader.EndOfStream)
9         {
10            listA.Add(reader.ReadLine());
11        }
12        for (int i = 0; i < listA.Count; i++)
13        {
14            var column = listA[i].Split(',');
15            for (int j = 0; j < column.Count(); j++)
16            {
17                if (j < column.Count() - 1)
18                {
19                    if (j == 3 || j == 5 || j == 13 || j ==
20                        15)
21                    {
22                        if (!feature2.ContainsKey(j))
23                        {
24                            List<double> list = new
25                                List<double>();
26                            feature2.Add(j, list);
27                        }
28                        feature2[j].Add(Convert.ToDouble(column[j]));
29                    }
30                    if (!minimum.ContainsKey(j))
31                    {
32                        List<double> list = new
33                            List<double>();
34                        minimum.Add(j, list);
35                    }
36                    minimum[j].Add(Convert.ToDouble(column[j]));
37                }
38            }
39        }
40        List<string> listB = new List<string>();
41        string resultMean = "E:\\Kuliah\\SEMESTER
42            7\\TA\\Dataset\\FeatureVector.csv";
43        using (StreamWriter writer =
44            File.AppendText(resultMean))
45        {
46            StringBuilder line = new StringBuilder();
47            StringBuilder lineSD = new StringBuilder();
48
49            //euclidean, depth
50            foreach (List<double> item in minimum.Values)
51            {
52                line.Append(string.Format("{0:0.000}",
53                    item.Min()) + ",");

```

```

50     }
51     //mean
52     foreach (List<double> item in feature2.Values)
53     {
54         line.Append(string.Format("{0:0.000}",
55             item.Average() + ","));
56     }
57     //variance, standard deviation
58     foreach (List<double> item in feature2.Values)
59     {
60         List<double> itemItung = item;
61         double avg = itemItung.Average();
62         double sum = itemItung.Select(val => (val - avg)
63             * (val - avg)).Sum();
64         double variance = sum /
65             Convert.ToDouble(itemItung.Count);
66         double sd = Math.Sqrt(variance);
67         line.Append(string.Format("{0:0.000}", variance)
68             + ","));
69         lineSD.Append(string.Format("{0:0.000}", sd) +
70             ",");
71     }
72     lineSD.Append(string.Format("{0}",
73         MainWindow.namaGerakan));
74     lineSD.AppendLine();
75     //mode
76     foreach (List<double> item in feature2.Values)
77     {
78         double most = item.GroupBy(y => y).Select(x =>
79             new { x.Key, Count = x.Count()
80             }).OrderByDescending(a =>
81             a.Count).First().Key;
82         line.Append(most + ",");
83     }
84     //median
85     foreach (List<double> item in feature2.Values)
86     {
87         List<double> itembaru = new List<double>();
88         itembaru = item.OrderBy(n => n).ToList();
89         line.Append(itembaru[19] + ",");
90     }
91     writer.Write(line);
92     writer.Write(lineSD);
93 }
94 return 1;
95 }

```

Kode Sumber 4.4 Kode Sumber Ekstraksi Fitur dan Penyimpanan Data Hasil Ekstraksi Fitur *Skeleton* Pegguna

4.3.3 Implementasi Proses *Testing Data*

Hasil dari *training* data yang disimpan kemudian dibandingkan dengan data uji yang dilakukan secara *real-time* oleh pengguna. Kedua data tersebut dibandingkan dengan menggunakan algoritma DTW seperti pada rancangan pada Subbab 3.2.6.3. Kode sumber implementasi DTW dapat dilihat pada Kode Sumber 4.5. Penjelasan kode sumber berikut adalah seperti berikut:

1. Variable x menunjukkan elemen pada data *training*
2. Variable y menunjukkan elemen pada data *testing*
3. Variable *jarak* menunjukkan hasil pengurangan masing-masing elemen dari variable x dan variable y
4. Variable f menunjukkan elemen pada matriks $x \times y$
5. Variable *sum* menunjukkan hasil penjumlahan jalur pada matriks $x \times y$

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  double[] x;
2  double[] y;
3  double[,] jarak;
4  double[,] f;
5  double sum;
6  public CalculateDTW(double[] _x, double[] _y, double
   threshold)
7  {
8      x = _x;
9      y = _y;
10     threshold = posThreshold;
11     jarak = new double[x.Length, y.Length];
12     f = new double[x.Length+1, y.Length+1];
13     for (int i = 0; i < x.Length; ++i)
14     {
15         for (int j = 0; j < y.Length; ++j)
16         {
17             jarak[i, j] = Math.Abs(x[i] - y[j]);
18         }
19     }
20
21     for (int i = 0; i <= x.Length; ++i)

```

```

22     {
23         for (int j = 0; j <= y.Length; ++j)
24         {
25             f[i, j] = -1.0;
26         }
27     }
28
29     for (int i = 1; i <= x.Length; ++i)
30     {
31         f[i,0] = double.PositiveInfinity;
32     }
33     for (int j = 1; j <= y.Length; ++j)
34     {
35         f[0, j] = double.PositiveInfinity;
36     }
37     f[0, 0] = 0.0;
38     sum = 0.0;
39 }
40
41 public double getSum(){
42     return sum;
43 }
44 public double[,] getFMatrix() {
45     return f;
46 }
47 public void computeDTW() {
48     sum = computeFBackward(x.Length, y.Length);
49 }
50 public double computeFForward()
51 {
52     for (int i = 1; i <= x.Length; ++i)
53     {
54         for (int j = 1; j <= y.Length; ++j)
55         {
56             if (f[i - 1, j] <= f[i - 1, j - 1] && f[i - 1,
57                 j] <= f[i, j - 1] && f[i - 1, j])
58             {
59                 f[i, j] = jarak[i - 1, j - 1] + f[i - 1, j];
60             }
61             else if (f[i, j - 1] <= f[i - 1, j - 1] && f[i,
62                 j - 1] <= f[i - 1, j] && f[i - 1, j])
63             {
64                 f[i, j] = jarak[i - 1, j - 1] + f[i, j - 1];
65             }
66             else if (f[i - 1, j - 1] <= f[i, j - 1] && f[i -
67                 1, j - 1] <= f[i - 1, j] && f[i - 1, j])
68             {
69                 f[i, j] = jarak[i - 1, j - 1] + f[i - 1, j -
70                 1];
71             }
72         }
73     }
74 }

```

```

67     }
68     }
69     }
70     return f[x.Length, y.Length];
71 }
72
73 public double computeFBackward(int i, int j)
74 {
75     if (!(f[i, j] < 0.0) ){
76         return f[i, j];
77     }
78     else {
79         if (computeFBackward(i - 1, j) <=
80             computeFBackward(i, j - 1) &&
81             computeFBackward(i - 1, j) <=
82             computeFBackward(i - 1, j - 1)
83             && computeFBackward(i - 1, j) <
84             double.PositiveInfinity)
85         {
86             f[i, j] = jarak[i - 1, j - 1] +
87                 computeFBackward(i - 1, j);
88         }
89         else if (computeFBackward(i, j - 1) <=
90             computeFBackward(i - 1, j) &&
91             computeFBackward(i, j - 1) <=
92             computeFBackward(i - 1, j - 1)
93             && computeFBackward(i, j - 1) <
94             double.PositiveInfinity)
95         {
96             f[i, j] = jarak[i - 1, j - 1] +
97                 computeFBackward(i - 1, j - 1);
98         }
99     }
100     return f[i, j];
101 }

```

Kode Sumber 4.5 Implementasi DTW untuk Pengenalan Gerakan Tangan

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) i5-4200M CPU @ 2.50GHz
- Memori (RAM) 4.00 GB
- Kinect Sensor

5.2 Skenario Pengujian

Pengujian dilakukan terhadap 20 bahasa isyarat yang dipilih oleh penulis seperti yang dapat dilihat pada Gambar 2.1. Pada awalnya penulis memilih bahasa isyarat yang mengacu pada *paper* referensi. Namun, karena ada beberapa kata yang tidak terdapat pada kamus Sistem Isyarat Bahasa Indonesia, maka penulis mengubah kata yang dipilih sesuai dengan kelasnya. Bahasa isyarat tersebut adalah sebagai berikut:

1. Halo
2. Salam
3. Bagaimana

4. Apa
5. Anak
6. Ibu
7. Saudara
8. Laki-laki
9. Keluarga
10. Saya
11. Anda
12. Rumah
13. Sekolah
14. Jahat
15. Baik
16. Masalah
17. Kerja
18. Pagi
19. Jumpa
20. Marah

Skenario pengujian yang dilakukan dibagi menjadi tiga bagian, yaitu skenario A, skenario B, dan skenario C. Pengujian bahasa isyarat pada ketiga skenario dilakukan menggunakan 20 gerakan yang terdiri dari 3 bahasa isyarat statis dan 17 bahasa isyarat dinamis seperti yang sudah dijelaskan pada Subbab 2.2. Penjelasan ketiga skenario tersebut adalah sebagai berikut:

1. Pengujian skenario A merupakan pengujian akurasi yang dilakukan oleh penulis dengan menggunakan data *training* yang sudah ada sebelumnya.
2. Pengujian skenario B merupakan pengujian akurasi yang dilakukan oleh pengguna lain menggunakan data *training* yang sudah ada sebelumnya.
3. Pengujian skenario C merupakan pengujian yang dilakukan oleh pengguna lain dengan data *training* baru yang ditambahkan oleh pengguna tersebut.

5.2.1 Pengujian Skenario A dan Analisis

Pada pengujian skenario A, uji coba dilakukan sendiri oleh penulis yang mempunyai karakteristik tinggi badan 163 cm dengan menggunakan data gerakan sebanyak 300 data/frame hasil implementasi yang dilakukan oleh penulis. Skenario dapat dilihat pada Tabel 5.1.

Tabel 5.1 Skenario Pengujian A

Nama Skenario Pengujian	Pengujian Akurasi A
Kode	SP-A
Algoritma	<i>Dynamic Time Warping</i>
Jumlah Data/Frame	300 data/frame
Penguji	Penulis memiliki tinggi 163cm
Prosedur Pengujian	Penulis melakukan uji coba 20 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 94%

Hasil yang didapatkan pada skenario A cukup baik, seperti yang dapat dilihat pada Tabel 5.1. Kesalahan pengenalan terjadi pada beberapa gerakan. Menurut pengamatan penulis berdasarkan skenario A yang dilakukan, kesalahan pengenalan tersebut terjadi karena posisi dan gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan dan ada beberapa gerakan yang memiliki kemiripan posicisi tangan.

5.2.2 Pengujian Skenario B dan Analisis

Pada pengujian skenario B, uji coba dilakukan oleh pengguna yang mempunyai karakteristik tinggi badan 173 cm dengan menggunakan data gerakan sebanyak 300 data/frame hasil

implementasi yang dilakukan oleh penulis. Skenario dapat dilihat pada Tabel 5.3.

Tabel 5.3 Skenario Pengujian B

Nama Skenario Pengujian	Pengujian Akurasi B
Kode	SP-B
Algoritma	<i>Dynamic Time Warping</i>
Jumlah Data/Frame	300 data/frame
Penguji	Pengguna memiliki tinggi 173 cm
Prosedur Pengujian	Pengguna melakukan uji coba 20 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 91%

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

Berdasarkan skenario B 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 pengenalan gerakan tangan.

Hal tersebut disebabkan karena tinggi badan pengguna turut andil dalam menentukan posisi gerakan. Selain itu, perbedaan karakteristik tinggi badan juga mempunyai efek terhadap identifikasi *skeleton joints* oleh Kinect 2.0.

5.2.3 Pengujian Skenario C dan Analisis

Pada pengujian skenario C, uji coba dilakukan oleh pengguna yang mempunyai karakteristik tinggi badan 175 cm dengan menggunakan data gerakan sebanyak 300 data/frame hasil implementasi DTW dilakukan oleh pengguna lain. Skenario dapat dilihat pada Tabel 5.5.

Tabel 5.5 Skenario Pengujian C

Nama Skenario Pengujian	Pengujian Akurasi C
Kode	SP-C
Algoritma	<i>Dynamic Time Warping</i>
Jumlah Data/Frame	300 data/frame
Penguji	Pengguna memiliki tinggi 175 cm
Prosedur Pengujian	Pengguna melakukan uji coba 20 gerakan bahasa isyarat pokok dimana masing-masing gerakan dilakukan uji coba sebanyak 5 kali
Hasil yang Diperoleh	Akurasi 94%

Hasil yang didapatkan pada skenario C cukup baik. Akurasi yang didapatkan sama seperti skenario A, seperti yang dapat dilihat pada Tabel 5.6. Kesalahan pengenalan terjadi pada beberapa gerakan. Menurut pengamatan penulis, kesalahan pengenalan terjadi karena posisi dan gerakan yang dilakukan kurang sesuai dengan gerakan bahasa isyarat yang diinginkan dan terdapat beberapa gerakan yang memiliki kemiripan dengan gerakan lain.

Tabel 5.2 Terjemahan Isyarat Kata Skenario Pengujian A

		Hasil Uji																			
		Halo	Salam	Bagaimana	Apa	Anak	Ibu	Saudara	Laki-laki	Keluarga	Saya	Anda	Rumah	Sekolah	Jahat	Baik	Masalah	Kerja	Pagi	Jumpa	Marah
Target Keabs	Halo	5																			
	Salam		4																		
	Bagaimana			4									1								
	Apa				5																
	Anak					5															
	Ibu						4														
	Saudara							5													
	Laki-laki						1		5												
	Keluarga		1							4											
	Saya										5										
	Anda											5									
	Rumah			1										4							
	Sekolah														5				1		
	Jahat															5					
	Baik																5				
	Masalah																	5			
	Kerja																		4		
Pagi																			5		
Jumpa																				5	
Marah										1											5
Total (%)		100%	80%	80%	100%	100%	80%	100%	100%	80%	100%	100%	80%	100%	100%	100%	100%	80%	100%	100%	100%

Tabel 5.4 Terjemahan Isyarat Kata Skenario Pengujian B

		Hasil Uji																				
		Halo	Salam	Bagaimana	Apa	Anak	Ibu	Saudara	Laki-laki	Keluarga	Saya	Anda	Rumah	Sekolah	Jahat	Baik	Masalah	Kerja	Pagi	Jumpa	Marah	
Target Kelas	Halo	5																				
	Salam		4																			
	Bagaimana			3						2												
	Apa				4																	
	Anak				1	5																
	Ibu						4															
	Saudara							5														
	Laki-laki								1	5												
	Keluarga										3							1				1
	Saya											5										
	Anda												5									
	Rumah			2										5								
	Sekolah														5							
	Jahat															5						
	Baik																5					
	Masalah																	4				
	Kerja																		5			
	Pagi																			5		
Jumpa																					5	
Marah			1																			4
Total (%)		100%	80%	60%	80%	100%	80%	100%	100%	60%	100%	100%	100%	100%	100%	100%	80%	100%	100%	100%	100%	80%

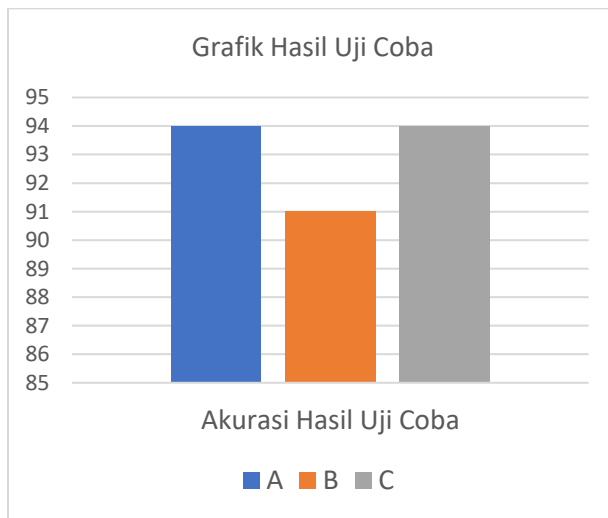
Tabel 5.6 Terjemahan Isyarat Kata Skenario Pengujian C

		Hasil Uji																				
		Halo	Salam	Bagaimana	Apa	Anak	Ibu	Saudara	Laki-laki	Keluarga	Saya	Anda	Rumah	Sekolah	Jahat	Baik	Masalah	Kerja	Pagi	Jumpa	Marah	
Target Kelas	Halo	5																				
	Salam		5																			
	Bagaimana			3																		
	Apa				5																	
	Anak					5																
	Ibu						5		1													
	Saudara							5														
	Laki-laki								4													
	Keluarga									5												
	Saya										4											
	Anda										1	5										
	Rumah				2									5								
	Sekolah														5			2				
	Jahat															5						
	Baik																5					
	Masalah																	3				
	Kerja																		5			
	Pagi																			5		
	Jumpa																				5	
	Marah																					5
Total (%)	100%	100%	60%	100%	100%	100%	100%	100%	80%	100%	80%	100%	100%	100%	100%	100%	60%	100%	100%	100%	100%	

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. Implementasi DTW di dalam perangkat lunak yang dibuat mempunyai akurasi rata-rata yang cukup baik yaitu 93%. Grafik hasil akurasi dari masing-masing percobaan dapat dilihat pada Gambar 5.1.



Gambar 5.1 Grafik Tingkat Akurasi Hasil Uji Coba

2. Hasil yang didapatkan pada skenario A cukup baik, yaitu 94%. Sedangkan hasil akurasi yang didapatkan pada skenario B lebih rendah daripada skenario A, namun masih cukup baik, yaitu 91%. Hal ini membuktikan bahwa perbedaan tinggi pengguna pada *range* 163 cm – 173 cm

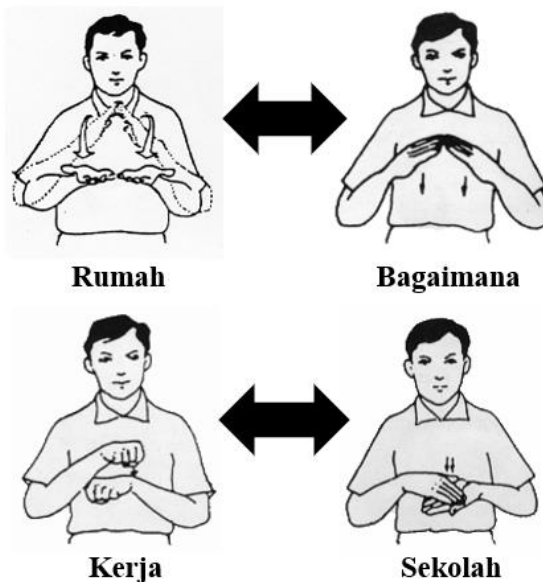
memiliki pengaruh pada akurasi pengenalan gerakan. Untuk skenario C, hasil akurasi yang didapatkan sama dengan skenario A, yaitu 94%.

3. Adanya beberapa gerakan bahasa isyarat yang hampir sama antara satu dengan yang lain menyebabkan kesalahan pada proses pengenalan. Analisis dapat dijelaskan sebagai berikut:
 - a. Pada skenario A, gerakan Ibu terdapat beberapa kesalahan pengenalan, yaitu mendapatkan hasil gerakan Laki-laki. Pada skenario B, gerakan Laki-laki juga terdapat kesalahan pengenalan, yaitu mendapatkan hasil gerakan Ibu, dan gerakan Apa mendapatkan hasil gerakan Anak. Pada skenario C, gerakan Saya memiliki kesalahan pengenalan, yaitu mendapatkan hasil gerakan Baik. Hal ini disebabkan karena gerakan-gerakan tersebut mempunyai gerakan yang serupa dan posisinya yang berdekatan. Gerakan yang memiliki kemiripan dapat dilihat pada Gambar 5.2.



Gambar 5.2 Kemiripan Gerakan dari Beberapa Gerakan Isyarat

- b. Pada skenario A dan B, gerakan Salam terdapat kesalahan pengenalan yaitu mendapatkan hasil gerakan Keluarga, dan gerakan Keluarga dikenali sebagai gerakan Marah dan Bagaimana. Pada skenario A, B, dan C, gerakan Bagaimana dikenali sebagai gerakan Rumah. Pada Skenario B dan C, gerakan Masalah terdapat kesalahan yaitu mendapatkan gerakan Keluarga dan gerakan Sekolah. Pada skenario B, gerakan Marah terdapat satu kesalahan pengenalan, yaitu mendapatkan hasil Keluarga. Pada skenario A gerakan Kerja terdapat satu kesalahan pengenalan, yaitu mendapatkan gerakan Sekolah. Kesalahan pengenalan pada gerakan-gerakan tersebut disebabkan karena masing-masing gerakan memiliki posisi di awal yang serupa. Gerakan yang memiliki posisi awal serupa dapat dilihat pada Gambar 5.3.



Gambar 5.3 Gerakan yang Memiliki Posisi Awal Serupa

4. Selain adanya beberapa gerakan bahasa isyarat yang hampir sama antara satu dengan yang lain, sensitifitas Kinect 2.0 dalam merekam koordinat masing-masing *skeleton joints* juga mempengaruhi tingkat akurasi pengenalan gerakan bahasa isyarat.

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. Untuk menentukan titik yang merepresentasikan gerakan tangan, dilakukan dengan cara ekstraksi fitur data. Fitur-fitur tersebut antara lain mencari jarak Euclidean antara *joint* telapak tangan terhadap *joint* lain (kecuali *joint* dirinya sendiri), kemudian menghitung nilai rata-rata, median, modus, varians, standar deviasi, dan kedalaman minimum (sumbu Z) antara *joint* telapak tangan terhadap *absolute joints*, yaitu kepala dan pinggul. Dari perhitungan tersebut, maka akan didapatkan 45 titik yang merepresentasikan masing-masing gerakan tangan.
2. Metode *Dynamic Time Warping* (DTW) yang digunakan untuk menerjemahkan bahasa isyarat berjalan dengan baik secara *real-time* dengan akurasi rata-rata 93%.
3. Identifikasi posisi gerakan yang dilakukan ketika melakukan *training* dan *testing* data sangat berpengaruh terhadap akurasi pengenalan gerakan.
4. Perbedaan karakteristik tinggi badan yaitu pada *range* antara 163 cm – 175 cm berefek terhadap penentuan koordinat *skeleton joints* dimana perbedaan tersebut mempengaruhi identifikasi posisi gerakan yang dilakukan oleh perangkat lunak, namun tidak mempengaruhi akurasi secara signifikan.
5. Perbedaan posisi Kinect dengan pengguna, baik pada saat pengambilan data maupun uji coba data, mempengaruhi

akurasi hasil pengenalan gerakan yang dilakukan oleh pengguna.

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 yang berbeda-beda.
2. Hasil yang didapatkan sudah cukup akurat, namun masih lambat ketika mengenali gerakan tangan. Untuk itu, menggabungkan metode DTW dengan teknik bisa menjadi solusi untuk meningkatkan kecepatan pada saat pengenalan gerakan tangan.
3. Identifikasi *skeleton joints* tidak hanya di pergelangan tangan saja. Karena pada beberapa gerakan bahasa isyarat, terdapat kesamaan dalam gerakan yang dihasilkan namun berbeda jika dilihat dari bentuk tangan.

DAFTAR PUSTAKA

- [1] A. Basuki, M. Zikky, J. Akhmad, N. Hasim, and N. I. Ramadhan, "Sensor Gerak Dengan Leap Motion Untuk Membantu Komunikasi Tuna Rungu / Wicara A-317 A-318," vol. 8, no. 1994, pp. 317–321, 2016. Retrieved from <http://sentia.polinema.ac.id/index.php/SENTIA2016/article/viewFile/61/56>
- [2] I. Hidayat, A. Hasibuan, A. Mulyana, and A. B. O, "Menjadi Suara Berbasis Kinect Menggunakan Metode Dynamic Time Warping Design And Implementation Of Sign Language To Speech Application," pp. 1–7. Retrieved from <https://openlibrary.telkomuniversity.ac.id/pustaka/files/100124/cover/perancangan-dan-implementasi-aplikasi-penerjemah-bahasa-isyarat-menjadi-suara-berbasis-kinect-menggunakan-metode-dynamic-time-warping.pdf>
- [3] A. A. S. Gunawan, "Pembelajaran Bahasa Isyarat Dengan Kinect Dan Metode Dynamic Time Warping," *Univ. Binus*, no. 9, pp. 1–3, 2013. Retrieved from http://library.binus.ac.id/eColls/eJournal/01_MatStat_Alexander%20Ashadi.pdf
- [4] W. N. Khotimah, Y. A. Susanto, and N. Suciati, "Combining Decision Tree And Back Propagation Genetic Algorithm Neural Network For Recognizing Word Gestures In Indonesian Sign Language Using Kinect," vol. 95, no. 2, pp. 292–298, 2017. Retrieved From <http://www.jatit.org/Volumes/Vol95No2/6Vol95No2.Pdf>
- [5] Nugyasa, Yahya Eka (2017), "Ekstraksi Fitur Dinamis pada Gerakan Tangan Menggunakan Kinect 2.0 untuk mengenali Bahasa Isyarat Indonesia," *Undergraduate thesis*, Institut Teknologi Sepuluh Nopember. Retrieved from <http://repository.its.ac.id/42827/>
- [6] G. García-bautista, F. Trujillo-romero, and S. O. Caballero-morales, "Mexican Sign Language Recognition Using Kinect and Data Time Warping Algorithm," 2010. Retrieved from <http://ieeexplore.ieee.org/document/7891832/?reload=true>

- [7] A. W. Yanuardi, S. Prasetyo and P. P. Johannes Adi, "Indonesian Sign Language Computer Application for the Deaf," in International Conference on Education Technology and Computer, Shanghai, 2010. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5529450>
- [8] M. Iqbal, "Pengenalan Bahasa Isyarat Indonesia Berbasis Sensor Flex dan Accelerometer Menggunakan Dynamic Time Warping," Institut Teknologi Sepuluh Nopember, Surabaya, 2011. Retrieved from <http://digilib.its.ac.id/ITS-Master-3100012045709/18061>
- [9] Y. Chen, B. Luo, Y.-L. Chen, G. Liang and X. Wu, "A Real-time Dynamic Hand Gesture Recognition System Using Kinect Sensor," in *IEEE Conference on Robotics and Biomimetics*, Zhuhai, 2015. Retrieved from <http://ieeexplore.ieee.org/document/7419071/>
- [10] "Kinect," Wikipedia, [Online]. Available: <http://en.wikipedia.org/wiki/Kinect>. [Accessed 7 June 2016].
- [11] "Microsoft Visual Studio," Wikipedia, [Online]. Available: http://id.wikipedia.org/wiki/Microsoft_Visual_Studio.

LAMPIRAN A KODE SUMBER

```

1 private void DtwReadClick(object sender, RoutedEventArgs e)
2 {
3     statusDetail.Content = "Testing Data";
4     statusAmbil = 2;
5     Button button = sender as Button;
6     test = button;
7     if (_recorder.IsRecording == false)
8     {
9         statusStop = 0;
10        startRecord();
11        _recorder.Start();
12        counter = 0;
13        countergenap = 0;
14        test.Content = "Stop";
15    }
16    else
17    {
18        statusStop = 1;
19        _recorder.Stop();
20        test.Content = "Start Testing";
21    }
22 }

```

Kode Sumber A.1 Fungsi *Start Testing*

```

1 private void DtwCreateClick(object sender, RoutedEventArgs
2 e)
3 {
4     Button button = sender as Button;
5     test = button;
6     namaGerakan = fileName.Text;
7     statusAmbil = 1;
8     statusDetail.Content = "Create Dataset";
9     if (_recorder.IsRecording)
10    {
11        statusStop = 1;
12        _recorder.Stop();
13        test.Content = "Create File";
14    }
15    else
16    {
17        statusStop = 0;
18        startRecord();
19        recorder.Start();

```

```

19         counter = 0;
20         countergenap = 0;
21         test.Content = "Stop";
22     }
23 }

```

Kode Sumber A.2 Fungsi *Create File*

```

1     if (statusAmbil == 2)
2     {
3         await Task.Delay(1000);
4         if (statusStop != 1)
5         {
6             if (_recorder.IsRecording == true)
7             {
8                 resultPath = _recorder.Stop();
9             }
10            calc_feature_testing(EuclidianDistanceTesting
11                (resultPath));
12            test.Content = "Start Testing";
13            var dataHasil = new List<Hasil>();
14            Debug.Write("\n");
15            foreach (KeyValuePair<string, List<List<double>>>
16                gerakan in dataSet)
17            {
18                foreach (List<double> item in gerakan.Value)
19                {
20                    _dtw = new CalculateDTW(item.ToArray(),
21                        dataTesting.ToArray());
22                    _dtw.computeDTW();
23                    dataHasil.Add(new Hasil { nama =
24                        gerakan.Key, hasil = _dtw.getSum() });
25                }
26            }
27            IEnumerable<Hasil> query =
28                dataHasil.OrderByDescending(x => x.hasil);
29            foreach (Hasil x in query)
30            {
31                Debug.Write(x.nama + " : " + x.hasil + "\n");
32                outputText.Content = x.nama;
33            }
34            startRecord();
35            string imageFullPath = imagePath +
36                outputText.Content + ".bmp";
37            string imageFullPath2 = imagePath +
38                outputText.Content + ".jpg";

```

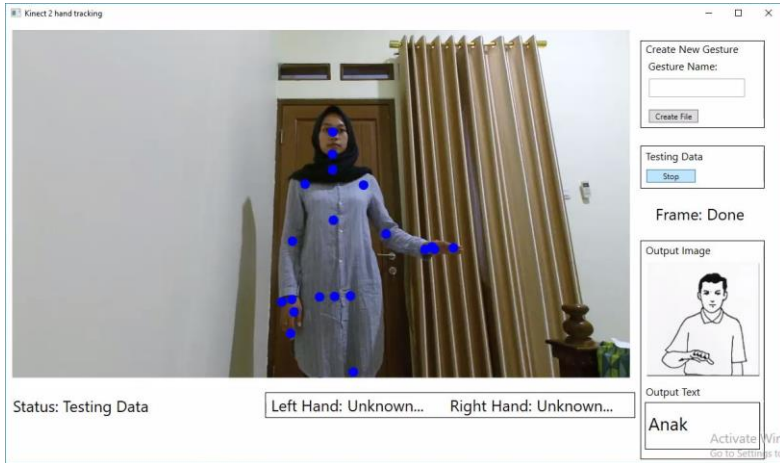

32	if (File.Exists(imageFullPath))
33	outputImage.Source = (ImageSource)new
	ImageSourceConverter().ConvertFrom
	(imageFullPath);
34	if (File.Exists(imageFullPath2))
35	outputImage.Source = (ImageSource)new
	ImageSourceConverter().ConvertFrom
	(imageFullPath2);
36	}
37	}

Kode Sumber A.3 Testing Data

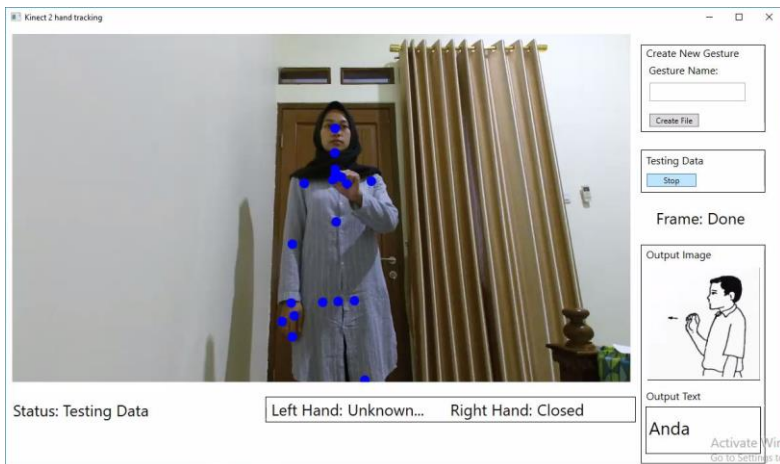
[Halaman ini sengaja dikosongkan]

LAMPIRAN B

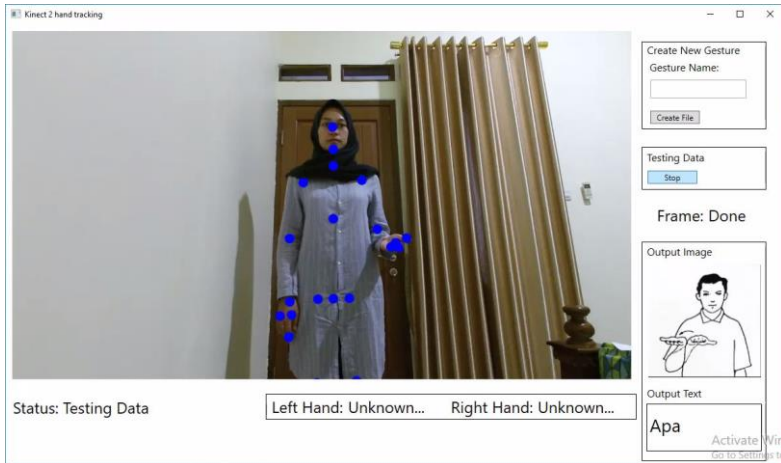
SCREENSHOT PERANGKAT LUNAK



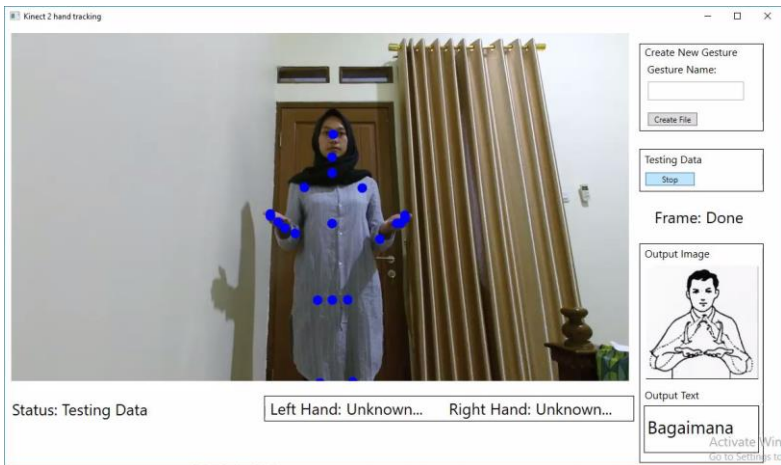
Gambar B.1 Penulis Mempraktekkan Bahasa Isyarat Anak



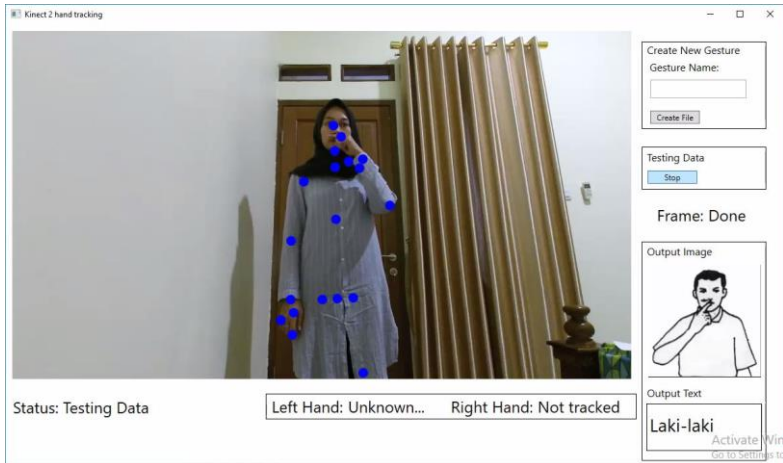
Gambar B.2 Penulis Mempraktekkan Bahasa Isyarat Anda



Gambar B.3 Penulis Mempraktekkan Bahasa Isyarat Apa



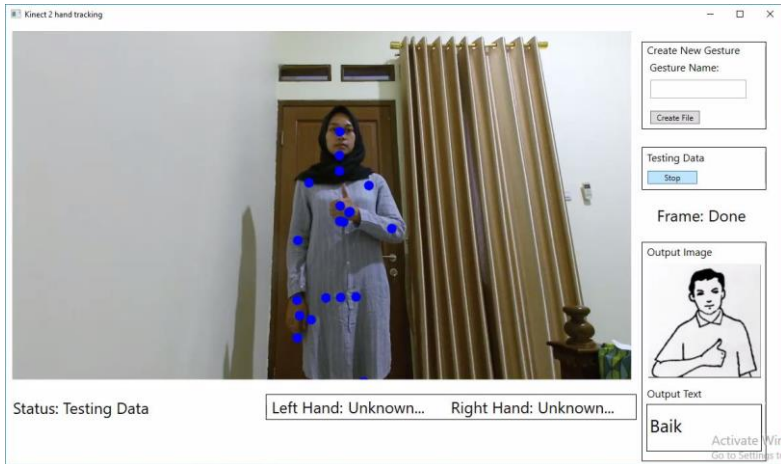
Gambar B.4 Penulis Mempraktekkan Bahasa Isyarat Bagaimana



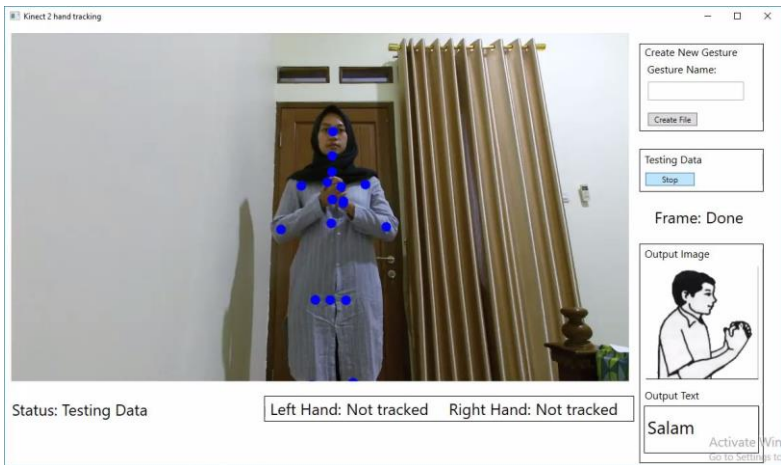
Gambar B.5 Penulis Mempraktekkan Bahasa Isyarat Laki-laki



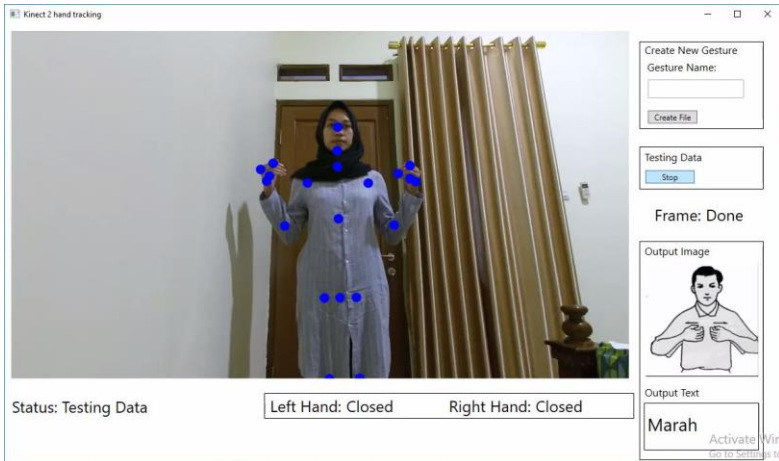
Gambar B.6 Penulis Mempraktekkan Bahasa Isyarat Jumpa



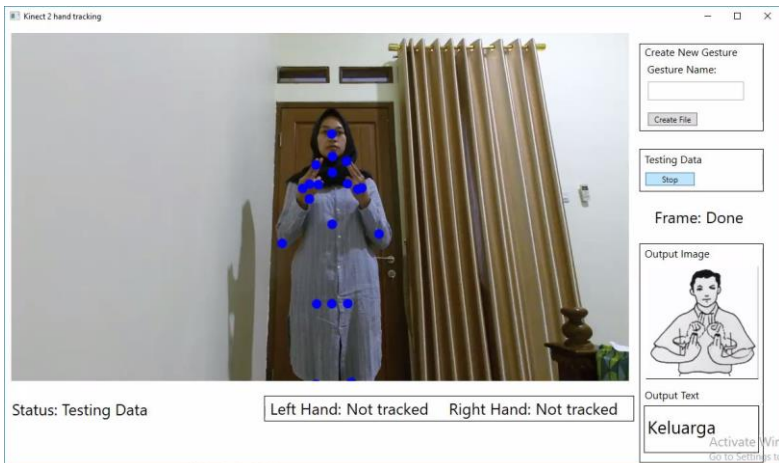
Gambar B.7 Penulis Mempraktekkan Bahasa Isyarat Baik



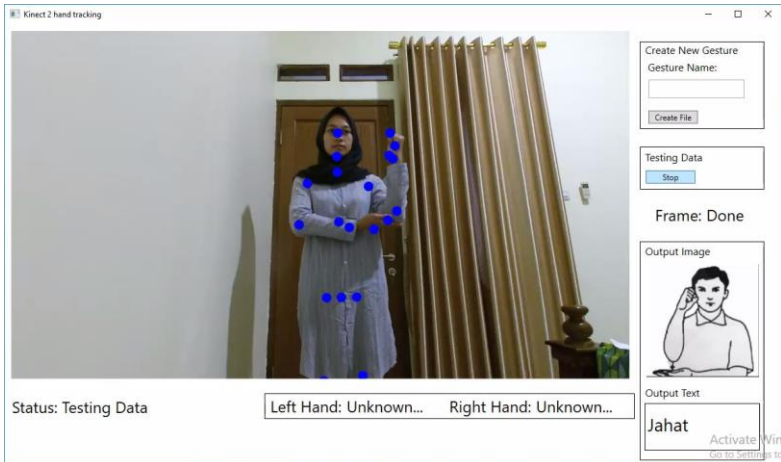
Gambar B.8 Penulis Mempraktekkan Bahasa Isyarat Salam



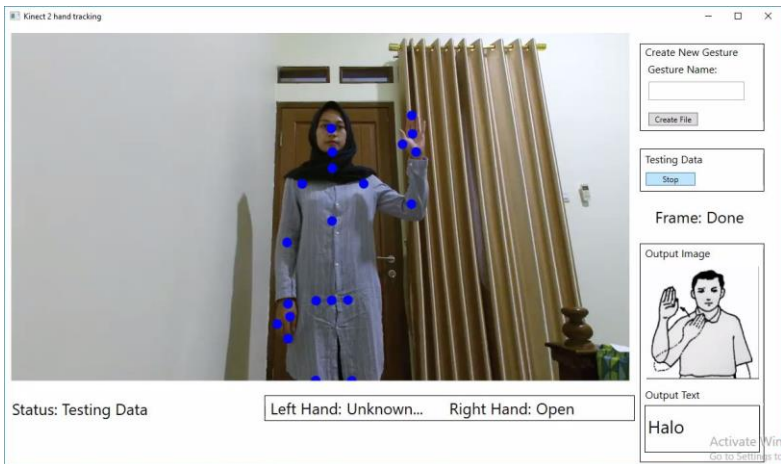
Gambar B.9 Penulis Mempraktekkan Bahasa Isyarat Marah



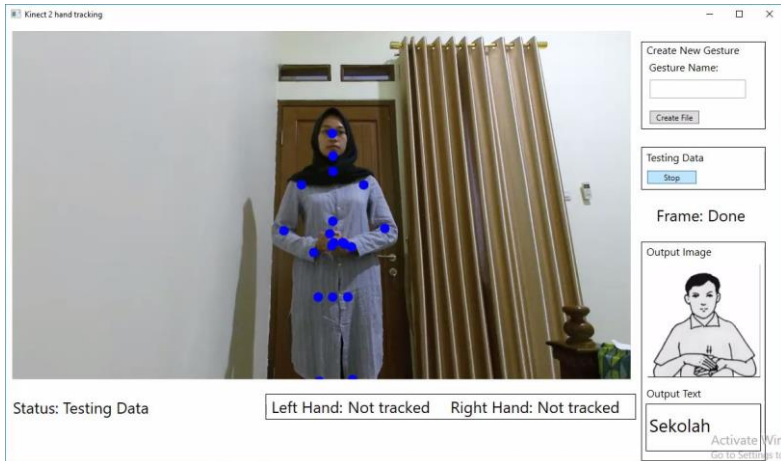
Gambar B.10 Penulis Mempraktekkan Bahasa Isyarat Keluarga



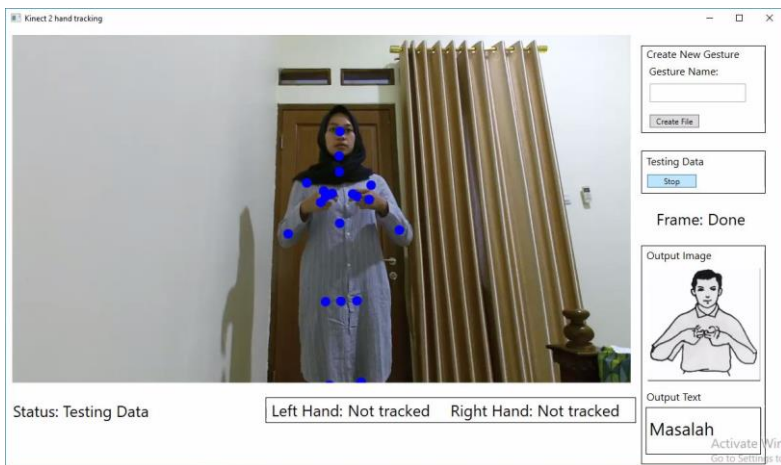
Gambar B.11 Penulis Mempraktekkan Bahasa Isyarat Jahat



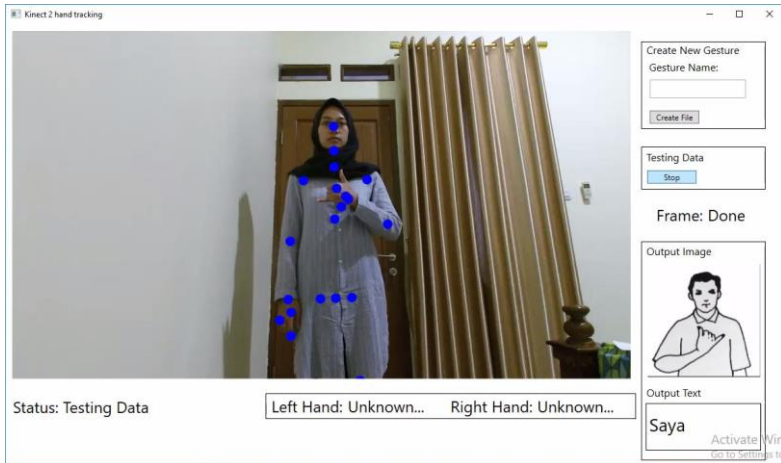
Gambar B.12 Penulis Mempraktekkan Bahasa Isyarat Halo



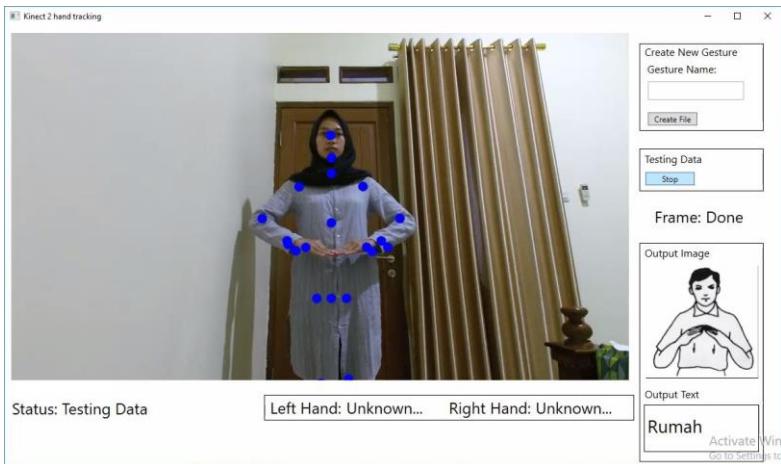
Gambar B.13 Penulis Mempraktekkan Bahasa Isyarat Sekolah



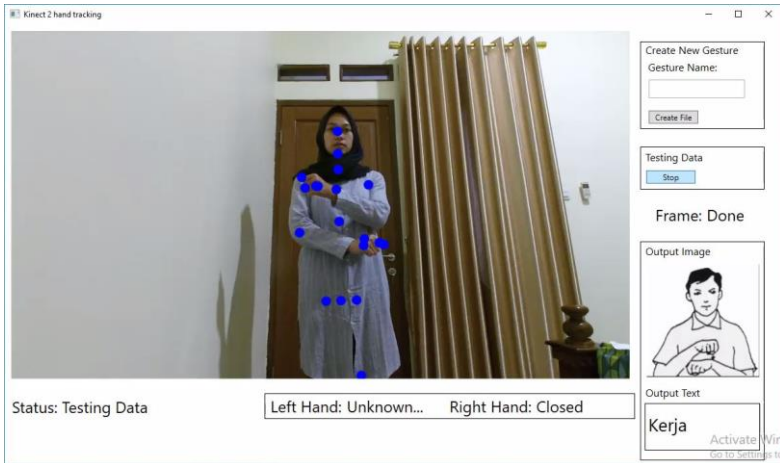
Gambar B.14 Penulis Mempraktekkan Bahasa Isyarat Masalah



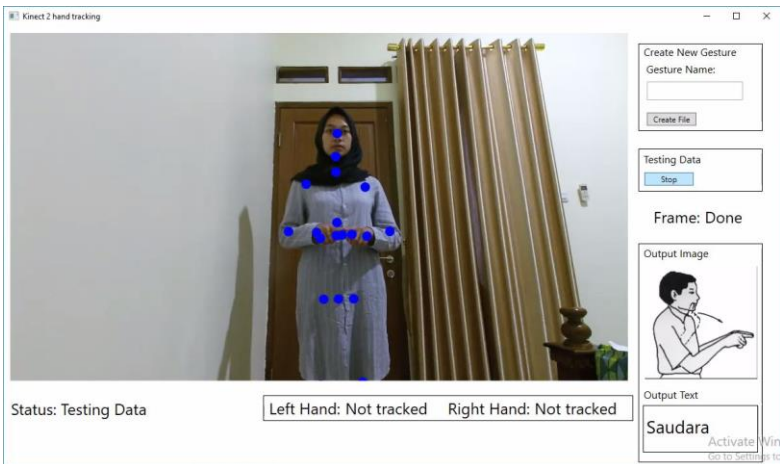
Gambar B.15 Penulis Mempraktekkan Bahasa Isyarat Saya



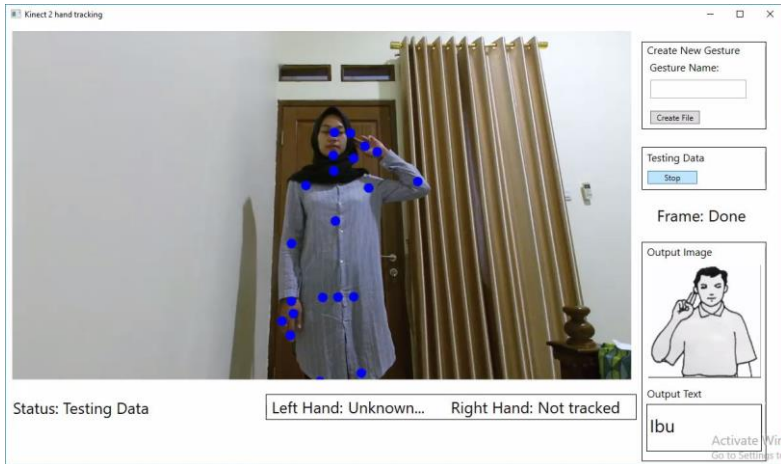
Gambar B.16 Penulis Mempraktekkan Bahasa Isyarat Rumah



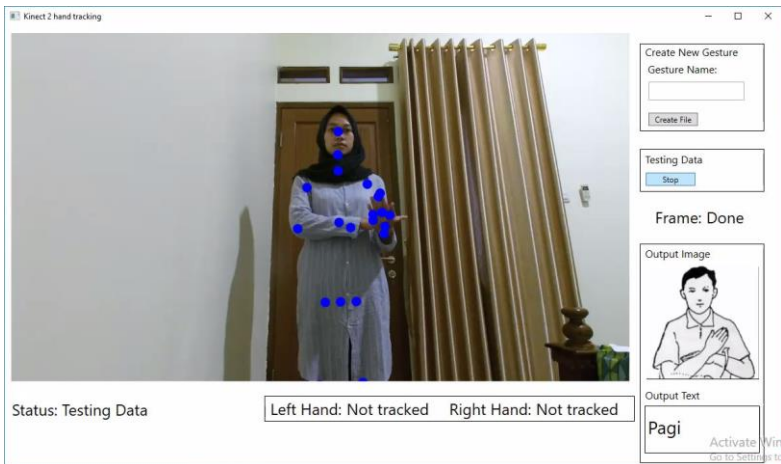
Gambar B.17 Penulis Mempraktekkan Bahasa Isyarat Kerja



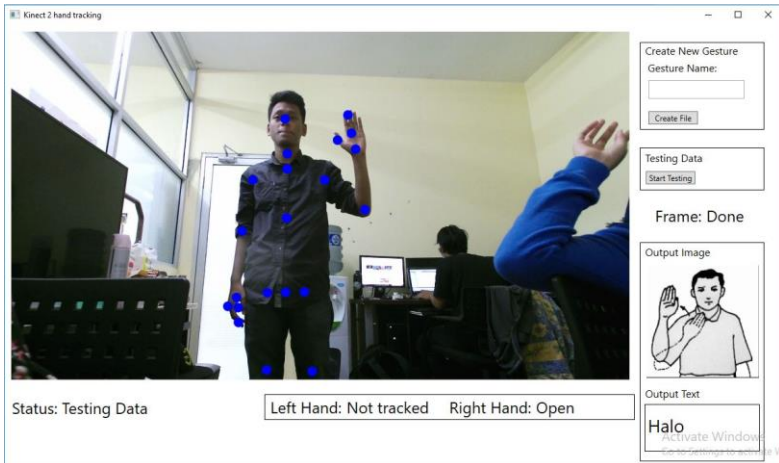
Gambar B.18 Penulis Mempraktekkan Bahasa Isyarat Saudara



Gambar B.19 Penulis Mempraktekkan Bahasa Isyarat Ibu



Gambar B.20 Penulis Mempraktekkan Bahasa Isyarat Pagi



Gambar B.21 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Halo



Gambar B.22 Salah Satu Pengguna Mempraktekkan Bahasa Isyarat Halo

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis lahir di kota Tangerang pada 11 Desember 1996. Merupakan anak dari dua bersaudara yang mempunyai hobi traveling dan menggambar. Penulis telah menempuh pendidikan formal yaitu TK Nurul Islam (2000 – 2002), SD Nurul Islam (2002 – 2007), SDN Karawaci Baru 2 (2007 – 2008), SMPN 1 Tangerang (2008 – 2011), SMAN 1 Tangerang (2011 – 2014), dan mahasiswa S1 Departemen

Informatika Fakultas Teknologi Informasi dan Komunikasi 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 Minat dan Bakat (2015 – 2016) serta Kepala Departemen Kewirausahaan (2015 – 2016) dan UKM Paduan Suara Mahasiswa (PSM) ITS (2014 - 2015). Selain itu, penulis juga mengikuti beberapa kepanitiaan di ITS seperti SCHEMATICS ITS sebagai staf (2015 - 2016), dan ICTS sebagai *organizing committee* (2015). Selanjutnya, penulis juga pernah mengikuti program pertukaran pelajar ke Thailand yang diselenggarakan oleh Suranaree University (2016). Penulis dapat dihubungi melalui surel tiaraanggita23@gmail.com.