



TUGAS AKHIR - TE 141599

**INTERAKSI INTERAKTIF ROBOT NAO MENGGUNAKAN
DETEKSI WAJAH DAN SUARA BAHASA INDONESIA**

Achmad Rizal Muttaqin
NRP 2210100177

Dosen Pembimbing
Dr. Surya Sumpeno ST., M.Sc.
Muhtadin, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

**INTERACTIVE NAO ROBOT INTERACTION USING FACE
DETECTION AND INDONESIAN VOICE**

Achmad Rizal Muttaqin
NRP 2210100177

Advisors
Dr. Surya Sumpeno ST., M.Sc.
Muhtadin, ST., MT.

Department of Electrical Engineering
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2015

**INTERAKSI INTERAKTIF ROBOT NAO MENGGUNAKAN
DETEKSI WAJAH DAN SUARA BAHASA INDONESIA**

TUGAS AKHIR

**Diajukan guna Memenuhi Sebagian Persyaratan
untuk Memperoleh Gelar Sarjana Teknik
pada**

**Bidang Studi Teknik Komputer dan Telematika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Surya Sumpeno, ST., M.Sc.

Muhtadin, ST., MT.

NIP. 196906131997021003

NIP. 198106092009121003



**SURABAYA
JANUARI, 2015**

ABSTRAK

Robot *NAO* sebagai salah satu robot humanoid memiliki banyak fitur yang masih dapat dikembangkan. Salah satunya adalah bagaimana nantinya robot *NAO* dapat berkomunikasi dengan manusia secara interaktif. Untuk membuat robot *NAO* yang dapat berkomunikasi secara interaktif, dapat dilakukan dengan mendeteksi objek di sekitarnya atau pembangkitan wicara menggunakan *Speech Engine System*. Metode deteksi objek bertujuan agar robot *NAO* mampu mendeteksi objek yang ada di sekitarnya, dalam hal ini objek yang dimaksud adalah wajah manusia. Adapun *Speech Engine System* yang dimaksud sebelumnya adalah gabungan dari metode *Speech to Text*, mesin dialog, dan *Text to Speech* yang membuat robot *NAO* mampu mengucapkan dan mengerti ujaran-ujaran sederhana dalam Bahasa Indonesia. Dari sistem yang dibuat, didapatkan hasil bahwa sistem mampu melakukan deteksi wajah secara kontinu pada *range* sudut 0° sampai 30° , ujaran manusia yang mampu dideteksi dengan tepat adalah sebesar 80.45%, ujaran manusia yang mampu direspon dengan tepat adalah sebesar 47.58%.

Kata kunci : Interaksi manusia-robot, kecerdasan buatan, deteksi wajah, *Speech to Text*, *Text to Speech*

(Halaman ini sengaja dikosongkan)

ABSTRACT

NAO Robot is a humanoid robot which still have developable features. One of these developable features is that how NAO can communicate with human interactively. Making interactively communicative NAO Robot can be achieved by detecting its surrounding object visually or generating speech using Speech Engine System. Object detection method is intended to make NAO Robot can detect its surrounding object, which in this case is human face. And about Speech Engine System mentioned before, it is a system consisted of Speech to Text, dialog machine, and Text to Speech which make NAO Robot can speak and understand simple human speech in Indonesian Language. The resulted system is able to perform face detection continously on angles of range between 0° and 30° , detect human speech accurately by 80.45%, and give response to human speech appropriately by 47.58%.

Keyword : Human-Robot Interaction, Artificial Intelligence, Face Detection, Speech to Text, Text to Speech

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul : **Interaksi Interaktif Robot NAO menggunakan Deteksi Wajah dan Suara Bahasa Indonesia.**

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Elektro ITS, Bidang Studi Teknik Komputer dan Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu dan Bapak tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Dr. Surya Sumpeno, ST., M.Sc. dan Bapak Muhtadin, ST., MT. atas bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Bidang Studi Teknik Komputer dan Telematika, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman angkatan e-50 serta teman-teman *b201-crew* Laboratorium Bidang Studi Teknik Komputer dan Telematika.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Januari 2015

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	2
1.6 Relevansi	3
BAB 2 TEORI PENUNJANG	5
2.1 Robot <i>Aldebaran NAO</i>	5
2.2 Perangkat Lunak Robot <i>Aldebaran NAO</i>	7
2.2.1 <i>NAOqi</i>	8
2.2.2 <i>Choregraphe</i>	8
2.2.3 <i>NAOsim</i>	9
2.2.4 <i>NAOqi Framework</i>	10
2.3 <i>Natural Language Processing</i>	12
2.3.1 <i>Artificial Intelligence Markup Language</i>	13
2.4 <i>Object Detection</i>	14
2.4.1 <i>Face Detection</i>	15
2.5 <i>Speech to Text</i>	17
2.5.1 <i>Teori Sampling</i>	18
2.5.2 <i>Hamming Window</i>	20
2.5.3 <i>Short-Time Fourier Transform</i>	20
2.6 <i>Text to Speech</i>	21
2.7 <i>BEAST (Bodily Expressive Action Stimulus Test)</i>	24
BAB 3 DESAIN SISTEM DAN IMPLEMENTASI	27
3.1 Konsep Sistem	27
3.2 Alur Kerja Sistem	28
3.3 Implementasi Sistem	29

3.3.1	Proses Penangkapan Informasi	30
3.3.2	Pengolahan Data	31
3.3.3	Eksekusi Perintah / Aktuator	32
3.4	Desain Kecerdasan Buatan	33
3.5	Desain Perilaku Robot	34
3.6	Desain Antarmuka	39
BAB 4	PENGUJIAN DAN ANALISA	41
4.1	Pengujian <i>Speech Recognition</i>	41
4.2	Pengujian Kemampuan Kecerdasan Sistem	42
4.3	Pengujian Kemampuan Deteksi Wajah	45
4.4	Pengujian Kemampuan <i>Tracking</i> Wajah	49
4.5	Kuesioner Keinteraktifan Sistem	56
BAB 5	PENUTUP	59
5.1	Kesimpulan	59
5.2	Saran	59
DAFTAR PUSTAKA		61
RIWAYAT HIDUP PENULIS		63

DAFTAR GAMBAR

2.1	Struktur <i>joint</i> robot <i>NAO</i>	7
2.2	Tampilan <i>NAOqi</i>	8
2.3	Tampilan <i>Choregraphe</i>	9
2.4	Tampilan <i>NAOsim</i>	10
2.5	Fitur <i>cross-language</i> pada <i>NAOqi Framework</i>	11
2.6	Arsitektur komunikasi melalui <i>NAOqi Framework</i>	12
2.7	Fitur yang digunakan pada <i>Viola-Jones Object Detection Framework</i>	15
2.8	Organ Tubuh Penghasil Suara pada Manusia.	18
2.9	Ilustrasi <i>Aliasing</i>	19
2.10	Proses <i>Speech to Text</i>	21
2.11	Mekanisme Produksi Wicara Manusia	22
2.12	Proses <i>Text to Speech</i>	23
2.13	Pose <i>BEAST</i>	25
3.1	Konsep Sistem	27
3.2	Alur Kerja Algoritma <i>Template Matching</i>	29
3.3	Diagram Alir Proses Penangkapan Informasi	30
3.4	Diagram Alir Proses Pengolahan Data	32
3.5	Diagram Alir Proses Eksekusi Perintah oleh Aktuator.	33
3.6	Pembuatan <i>Behavior</i> pada <i>Choregraphe</i>	35
3.7	Desain antarmuka <i>NANA</i>	39
4.1	<i>NANA</i> tidak mendeteksi adanya akses internet	41
4.2	Performansi <i>Speech to Text</i>	42
4.3	Performansi Mesin Dialog.	44
4.4	Persebaran <i>Confidence</i> dari Metode <i>Template Matching</i>	49
4.5	Persebaran Data Hasil Kuesioner	57

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

2.1	Spesifikasi Robot.	5
2.2	Nilai <i>Joint Robot Aldebaran NAO H25</i> versi 33.	6
2.3	<i>Haar feature</i> pada <i>Viola-Jones Object Detection Framework</i> .	16
3.1	Daftar Pose <i>BEAST</i> yang Digunakan dalam Penelitian	35
4.1	Daftar Percakapan	43
4.2	Performansi <i>NANA</i> dalam Deteksi Wajah.	45
4.3	Performansi <i>NANA</i> dalam <i>Tracking</i> Wajah	49

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Sistematika Penulisan	2
1.6 Relevansi	3
BAB 2 TEORI PENUNJANG	5
2.1 Robot <i>Aldebaran NAO</i>	5
2.2 Perangkat Lunak Robot <i>Aldebaran NAO</i>	7
2.2.1 <i>NAOqi</i>	8
2.2.2 <i>Choregraphe</i>	8
2.2.3 <i>NAOsim</i>	9
2.2.4 <i>NAOqi Framework</i>	10
2.3 <i>Natural Language Processing</i>	12
2.3.1 <i>Artificial Intelligence Markup Language</i>	13
2.4 <i>Object Detection</i>	14
2.4.1 <i>Face Detection</i>	15
2.5 <i>Speech to Text</i>	17
2.5.1 <i>Teori Sampling</i>	18
2.5.2 <i>Hamming Window</i>	20
2.5.3 <i>Short-Time Fourier Transform</i>	20
2.6 <i>Text to Speech</i>	21
2.7 <i>BEAST (Bodily Expressive Action Stimulus Test)</i>	24
BAB 3 DESAIN SISTEM DAN IMPLEMENTASI	27
3.1 Konsep Sistem	27
3.2 Alur Kerja Sistem	28
3.3 Implementasi Sistem	29

3.3.1	Proses Penangkapan Informasi	30
3.3.2	Pengolahan Data	31
3.3.3	Eksekusi Perintah / Aktuator	32
3.4	Desain Kecerdasan Buatan	33
3.5	Desain Perilaku Robot	34
3.6	Desain Antarmuka	39
BAB 4	PENGUJIAN DAN ANALISA	41
4.1	Pengujian <i>Speech Recognition</i>	41
4.2	Pengujian Kemampuan Kecerdasan Sistem	42
4.3	Pengujian Kemampuan Deteksi Wajah	45
4.4	Pengujian Kemampuan <i>Tracking</i> Wajah	49
4.5	Kuesioner Keinteraktifan Sistem	56
BAB 5	PENUTUP	59
5.1	Kesimpulan	59
5.2	Saran	59
DAFTAR PUSTAKA		61
RIWAYAT HIDUP PENULIS		63

DAFTAR GAMBAR

2.1	Struktur <i>joint</i> robot <i>NAO</i>	7
2.2	Tampilan <i>NAOqi</i>	8
2.3	Tampilan <i>Choregraphe</i>	9
2.4	Tampilan <i>NAOsim</i>	10
2.5	Fitur <i>cross-language</i> pada <i>NAOqi Framework</i>	11
2.6	Arsitektur komunikasi melalui <i>NAOqi Framework</i>	12
2.7	Fitur yang digunakan pada <i>Viola-Jones Object Detection Framework</i>	15
2.8	Organ Tubuh Penghasil Suara pada Manusia.	18
2.9	Ilustrasi <i>Aliasing</i>	19
2.10	Proses <i>Speech to Text</i>	21
2.11	Mekanisme Produksi Wicara Manusia	22
2.12	Proses <i>Text to Speech</i>	23
2.13	Pose <i>BEAST</i>	25
3.1	Konsep Sistem	27
3.2	Alur Kerja Algoritma <i>Template Matching</i>	29
3.3	Diagram Alir Proses Penangkapan Informasi	30
3.4	Diagram Alir Proses Pengolahan Data	32
3.5	Diagram Alir Proses Eksekusi Perintah oleh Aktuator.	33
3.6	Pembuatan <i>Behavior</i> pada <i>Choregraphe</i>	35
3.7	Desain antarmuka <i>NANA</i>	39
4.1	<i>NANA</i> tidak mendeteksi adanya akses internet	41
4.2	Performansi <i>Speech to Text</i>	42
4.3	Performansi Mesin Dialog.	44
4.4	Persebaran <i>Confidence</i> dari Metode <i>Template Matching</i>	49
4.5	Persebaran Data Hasil Kuesioner	57

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

2.1	Spesifikasi Robot.	5
2.2	Nilai <i>Joint Robot Aldebaran NAO H25</i> versi 33.	6
2.3	<i>Haar feature</i> pada <i>Viola-Jones Object Detection Framework</i> .	16
3.1	Daftar Pose <i>BEAST</i> yang Digunakan dalam Penelitian	35
4.1	Daftar Percakapan	43
4.2	Performansi <i>NANA</i> dalam Deteksi Wajah.	45
4.3	Performansi <i>NANA</i> dalam <i>Tracking</i> Wajah	49

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penelitian mengenai *Human-Robot Interaction* merupakan topik yang relatif baru dan masih dapat dikembangkan ke arah yang lebih lanjut. Pelayanan berbasis robot semisal penyajian makanan pasien di rumah sakit, atau pekerjaan-pekerjaan rumah telah membawa robot *autonomous* ke dalam kehidupan manusia[1]. Namun, fungsi interaksi manusia-robot dalam mengerjakan tugas-tugas tersebut berada pada tingkat yang masih minimum. Interaksi sosial manusia-robot akan memainkan peran penting dalam hal penggunaan robot dalam keseharian manusia. Interaksi sosial manusia-robot yang dimaksud dapat berupa penanganan pekerjaan-pekerjaan rumah, pemberian perawatan medis bagi lansia, menjadi asisten bagi orang dengan cacat fisik, hiburan yang edukatif, menjadi penunjuk arah di pusat-pusat informasi umum, dan sebagainya[2].

Penelitian mengenai *Human-Robot Interaction* seringkali diimplementasikan pada robot humanoid. Robot humanoid merupakan robot yang dibuat dengan bentuk yang menyerupai manusia. Secara umum, robot humanoid memiliki tubuh, kepala, dua lengan, dan dua kaki, meskipun beberapa robot humanoid hanya dibuat sebagian, misalnya, hanya dari pinggang sampai ke atas. Beberapa robot humanoid mungkin juga memiliki kepala yang dirancang untuk meniru fitur-fitur pada wajah manusia seperti mata dan mulut.

Robot *NAO* sebagai salah satu robot humanoid memiliki banyak fitur yang masih dapat dikembangkan. Salah satunya adalah bagaimana nantinya robot *NAO* dapat berkomunikasi dengan manusia secara interaktif. Untuk membuat robot *NAO* yang dapat berkomunikasi secara interaktif, dapat dilakukan dengan menggunakan metode deteksi objek, *Speech to Text* dan suatu *Speech Engine System*. Metode deteksi objek bertujuan agar robot *NAO* mampu mendeteksi objek yang ada di sekitarnya, dalam hal ini objek yang dimaksud adalah manusia. Adapun *Speech Engine System* dan *Speech to Text* yang dimaksud sebelumnya adalah sistem yang membuat robot *NAO* mampu mengucapkan dan mengerti ujaran-ujaran sederhana dalam Bahasa Indonesia.

1.2 Permasalahan

Kemampuan robot dalam berkomunikasi dengan manusia memiliki peranan penting pada penelitian mengenai *Human-Robot Interaction*. Oleh

karena itu diperlukan pengembangan lebih lanjut mengenai desain algoritma agar robot humanoid *NAO* mampu berinteraksi secara interaktif dengan manusia.

1.3 Tujuan

Penelitian ini bertujuan untuk membuat desain algoritma agar robot humanoid *NAO* dapat berinteraksi secara interaktif dengan manusia dengan mendeteksi objek tertentu di sekitarnya dan membangkitkan wicara menggunakan *Speech Engine System*. Robot *NAO* yang sudah dapat berinteraksi dengan manusia dapat dimanfaatkan dalam penelitian lain yang lebih mendalam di bidang *Human-Robot Interaction*.

1.4 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah :

1. Robot yang digunakan adalah *NAO H25*
2. Interaktifitas antara robot *NAO* dengan manusia hanya mencakup hal-hal yang berkaitan dengan salam perkenalan (*greeting*)

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun secara sistematis dan terstruktur sehingga lebih mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang hendak melanjutkan penelitian ini. Sistematika laporan penelitian ini diatur sedemikian rupa sehingga mengikuti alur sebagai berikut.

BAB I PENDAHULUAN

Bab ini berisi uraian tentang latar belakang, permasalahan, penegasan dan alasan pemilihan judul, tujuan penelitian, metodologi penelitian, dan sistematika laporan.

BAB II TEORI PENUNJANG

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini.

BAB III DESAIN SISTEM DAN IMPLEMENTASI

Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram agar sistem yang akan dibuat mudah dibaca dan mudah untuk diimplementasikan pada proses pembuatan aplikasi perangkat lunak.

BAB IV PENGUJIAN DAN ANALISA

Bab ini menjelaskan tentang pengujian dan analisa yang dilakukan terhadap sistem dalam penelitian ini. Spesifikasi perangkat keras dan pe-

rangkat lunak yang digunakan juga disebutkan dalam bab ini. Sehingga ketika akan dikembangkan lebih jauh, spesifikasi perlengkapannya bisa dipenuhi dengan mudah tanpa harus melakukan ujicoba perangkat lunak maupun perangkat keras lagi.

BAB V PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

Penelitian mengenai interaksi manusia dengan robot humanoid sudah sering dibahas. Pada tugas akhir ini, pengujian dilakukan pada robot *NAO* dengan *NAO SDK* 1.8.11.

(Halaman ini sengaja dikosongkan)

BAB 2 TEORI PENUNJANG

2.1 Robot *Aldebaran NAO*

Robot humanoid *NAO* dikembangkan oleh *Aldebaran Robotics* sejak 2005, robot ini dikembangkan dengan 25 sendi atau biasa disebut *degree of freedom* (DOF). Robot humanoid ini memiliki tinggi 57 cm dengan bobot 4.5 kg. Bobot ini sudah termasuk baterai yang digunakan pada robot. Lama daya tahan baterai bila kondisi *autonomous* dapat digunakan hingga 90 menit, hal ini tergantung dari perilaku yang dilakukan oleh robot. Spesifikasi dari robot humanoid *NAO* dapat dilihat pada tabel 2.1.

Tabel 2.1 Spesifikasi Robot[3]

<i>Operating system</i>	<i>Linux platform</i>
<i>Processor</i>	AMD Geode
<i>Speed Clock</i>	500 MHz
RAM	256 MB SDRAM
<i>Flash memory</i> ¹	1 GB

Ada dua cara komunikasi yang digunakan antara modul *remote* (robot humanoid *NAO*) dan modul lokal (aplikasi) yaitu dengan *Ethernet* atau *Wi-Fi*. *Aldebaran Robotics* telah mengeluarkan beberapa versi untuk robot humanoid *NAO* antara lain :

1. Edisi *Robocup* dirilis dengan 21 DOF pada Maret 2008.
2. Edisi *Academics* dirilis dengan 25 DOF 2008.
3. *NAO H25* Versi 33 dirilis dengan 25 DOF, memiliki lengan yang lebih panjang, dan memiliki beberapa versi kepala pada tahun 2010.

Adapun robot *NAO* yang digunakan dalam penelitian ini adalah *NAO H25* versi 33. Robot humanoid *NAO H25* versi 3.3 memiliki beberapa sensor yang dapat digunakan, sensor itu antara lain:

1. Dua *gyrometer* dan sebuah *accelerometer* yang digunakan untuk melakukan estimasi posisi dan orientasi dari robot. Sensor ini terletak di dalam tubuh robot.
2. Dua kamera yang terletak di kepala.
3. Empat *microphone* yang terletak di sekitar kepala.

¹*Flash memory* adalah salah satu jenis dari *Electrically Erasable Programmable Read-Only Memory* (*EEPROM*). *Flash memory* merupakan perangkat keras umumnya berupa kuni- ci yang digunakan untuk mengaktifkan perangkat lunak tertentu.

4. Dua *ultrasonic transmitter* dan *receiver* terletak di depan bagian dada.
5. Empat *force sensitive resistor* yang terletak pada masing-masing pergelangan kaki.
6. Sensor sentuh yang terletak pada kepala dan lengan.

Akan tetapi dalam penelitian ini, tidak semua sensor digunakan. Sensor yang digunakan dalam penelitian ini adalah kamera dan *microphone*. Selain itu, penelitian ini juga melibatkan komponen mekanik pada robot (*joint robot*). Untuk itu diperlukan gambaran mengenai nilai maksimum dan minimum untuk beberapa *joint* yang digunakan dalam penelitian ini. Tabel 2.2 menunjukkan informasi mengenai nilai maksimum dan minimum untuk *joint* pada robot *Aldebaran NAO H25* versi 33.

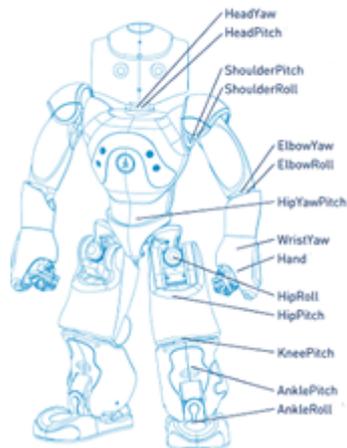
Tabel 2.2 Nilai *Joint Robot Aldebaran NAO H25* versi 33[3]

<i>Joint</i>	<i>Angle Name</i>	<i>Min Value</i>	<i>Max Value</i>
<i>Head</i>	<i>Yaw</i>	-119.5°	119.5°
	<i>Pitch</i>	-38.5°	39.5°
<i>RShoulder</i>	<i>Pitch</i>	-119.5°	119.5°
	<i>Roll</i>	-94.5°	-0.5°
<i>RElbow</i>	<i>Yaw</i>	-119.5°	119.5°
	<i>Roll</i>	0.5°	89.5°
<i>RWrist</i>	<i>Yaw</i>	-104.5°	104.5°
<i>RHand</i>	<i>Open and Close</i>		
<i>LShoulder</i>	<i>Pitch</i>	-119.5°	119.5°
	<i>Roll</i>	0.5°	94.5°
<i>LElbow</i>	<i>Yaw</i>	-119.5°	119.5°
	<i>Roll</i>	-89.5°	-0.5°
<i>LWrist</i>	<i>Yaw</i>	-104.5°	104.5°
<i>LHand</i>	<i>Open and Close</i>		
<i>Hip</i> ²	<i>YawPitch</i>	-65.62°	42.44°
<i>RHip</i>	<i>Pitch</i>	-101.54°	27.82°
	<i>Roll</i>	-42.3°	23.76°

²Orientasi kanan dan kiri dari *HipYawPitch* terdiri dari satu motor, sehingga tidak dapat dikendalikan secara terpisah.

<i>Joint</i>	<i>Angle Name</i>	<i>Min Value</i>	<i>Max Value</i>
<i>RKnee</i>	<i>Pitch</i>	-5.9°	121.47°
<i>RAnkle</i>	<i>Pitch</i>	-67.97°	53.4°
	<i>Roll</i>	-22.27°	45.03°
<i>LHip</i>	<i>Pitch</i>	-101.63°	27.73°
	<i>Roll</i>	-21.74°	45.29°
<i>LKnee</i>	<i>Pitch</i>	-5.29°	121.04°
<i>LAnkle</i>	<i>Pitch</i>	-68.15°	53.86°
	<i>Roll</i>	-44.06°	22.79°

Adapun untuk pengilustrasian *joint* robot *NAO H25* versi 33 yang disebutkan pada Tabel 2.2 disajikan pada Gambar 2.1.



Gambar 2.1 Struktur *joint* robot *NAO*[4]

2.2 Perangkat Lunak Robot *Aldebaran NAO*

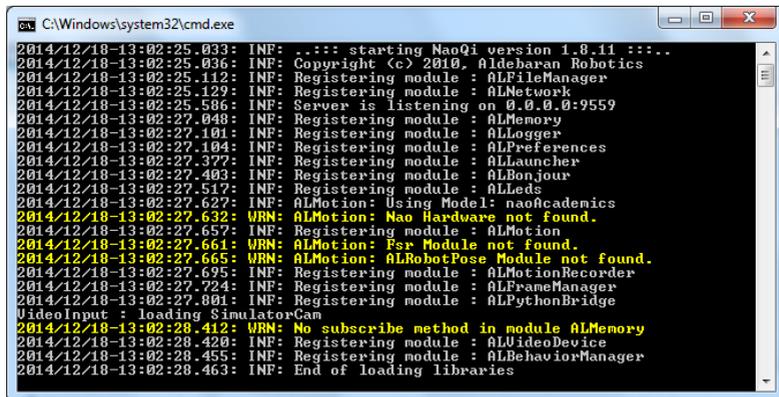
Selain perangkat keras berupa robot humanoid, *Aldebaran NAO* juga dilengkapi dengan perangkat lunak pendukung sebagai antarmuka yang menghubungkan antara perangkat keras robot *Aldebaran NAO* dengan pengguna atau *developer*. Selain itu, perangkat lunak yang dimaksud dibuat dengan tujuan untuk memudahkan pengguna atau *developer* dari

robot *Aldebaran NAO* untuk mengeksplorasi fitur-fitur yang dimiliki oleh robot *Aldebaran NAO*.

2.2.1 *NAOqi*

NAOqi merupakan perangkat lunak pendukung yang dapat dipergunakan pada beberapa *Operating System* sebagai virtual robot *Aldebaran NAO*. Karena merupakan virtual robot, maka tidak semua fitur yang terdapat pada robot *Aldebaran NAO* dapat diimplementasikan oleh *NAOqi*.

Pada penelitian tugas akhir ini, *NAOqi* digunakan pada kondisi-kondisi yang memerlukan virtualisasi dari robot *Aldebaran NAO*. Gambar 2.2 mengilustrasikan *NAOqi* saat sedang dijalankan.



```
C:\Windows\system32\cmd.exe
2014/12/18-13:02:25.033: INF: ..:: starting Naoqi version 1.8.11 ::...
2014/12/18-13:02:25.036: INF: Copyright (c) 2010. Aldebaran Robotics
2014/12/18-13:02:25.112: INF: Registering module : ALFileManager
2014/12/18-13:02:25.129: INF: Registering module : ALNetwork
2014/12/18-13:02:25.586: INF: Server is listening on 0.0.0.0:9559
2014/12/18-13:02:27.040: INF: Registering module : ALMemory
2014/12/18-13:02:27.101: INF: Registering module : ALLogger
2014/12/18-13:02:27.104: INF: Registering module : ALPreferences
2014/12/18-13:02:27.377: INF: Registering module : ALLauncher
2014/12/18-13:02:27.403: INF: Registering module : ALBonjour
2014/12/18-13:02:27.517: INF: Registering module : ALLEDs
2014/12/18-13:02:27.627: INF: ALMotion: Using Model: naoAcademics
2014/12/18-13:02:27.632: WRN: ALMotion: Nao Hardware not found.
2014/12/18-13:02:27.657: INF: Registering module : ALMotion
2014/12/18-13:02:27.661: WRN: ALMotion: Fr Module not found.
2014/12/18-13:02:27.665: WRN: ALMotion: ALRobotPose Module not found.
2014/12/18-13:02:27.695: INF: Registering module : ALMotionRecorder
2014/12/18-13:02:27.724: INF: Registering module : ALFrameManager
2014/12/18-13:02:27.801: INF: Registering module : ALPythonBridge
VideoInput : loading SimulatorCam
2014/12/18-13:02:28.412: WRN: No subscribe method in module ALMemory
2014/12/18-13:02:28.420: INF: Registering module : ALVideoDevice
2014/12/18-13:02:28.456: INF: Registering module : ALBehaviorManager
2014/12/18-13:02:28.463: INF: End of loading libraries
```

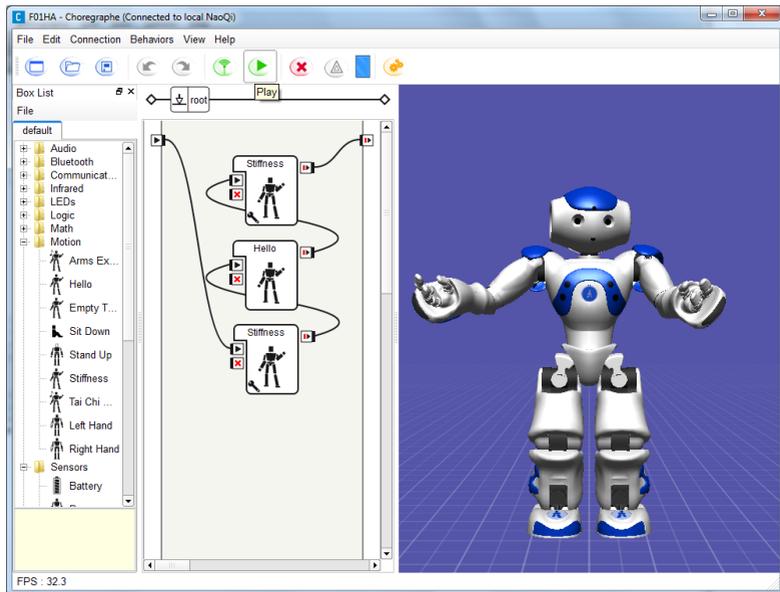
Gambar 2.2 Tampilan *NAOqi*

2.2.2 *Choregraphe*

Choregraphe merupakan perangkat lunak yang dipergunakan untuk merancang perilaku pada robot *Aldebaran NAO*. Perilaku robot adalah perangkat lunak berupa file yang berisi posisi-posisi *joint* robot *Aldebaran NAO* pada satu *range* waktu tertentu. Selain itu juga terdapat beberapa *script* yang mendukung pergerakan atau perubahan posisi-posisi *joint* dari robot. Pada *Choregraphe* terdapat beberapa *box* yang merepresentasikan satu perilaku tertentu, satu area kerja, serta area yang berisi visualisasi dari robot *Aldebaran NAO*. Dikarenakan *Choregraphe* merupakan perangkat lunak yang memiliki fitur “*drag and drop*”, untuk membuat perilaku sederhana dari robot *Aldebaran NAO*, pengguna atau *developer* dapat memilih salah satu *box* standar dari *Choregraphe*, meletakkannya pada area kerja, dan menghubungkannya sesuai dengan *sequence* yang diharapkan.

Adapun selanjutnya, *Choregraphe* dapat dihubungkan pada robot yang sesungguhnya maupun virtual, sehingga hasil yang telah dibuat pada area kerja *Choregraphe* dapat langsung dikirim ke robot yang sesungguhnya atau dapat dicoba terlebih dahulu ke robot virtual.

Pada penelitian tugas akhir ini, *Choregraphe* digunakan untuk merancang perilaku robot *Aldebaran NAO* yang akan digunakan saat berinteraksi dengan manusia. Gambar 2.3 mengilustrasikan tampilan *Choregraphe* saat sedang dijalankan.



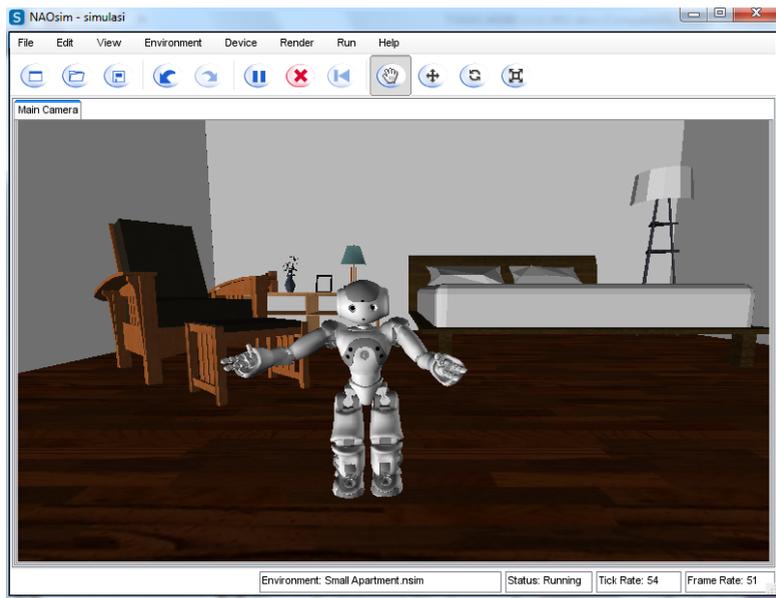
Gambar 2.3 Tampilan *Choregraphe*

2.2.3 *NAOsim*

NAOsim merupakan perangkat lunak yang yang difungsikan untuk memvisualisasikan perangkat lunak *NAOqi* yang bertindak sebagai robot virtual. Pada *NAOsim* ini, terdapat beberapa fitur yang diberikan, antara lain jenis robot yang dapat divisualisasikan oleh *NAOsim* bervariasi mulai dari *NAO* versi 32 hingga *NAO* versi 40, lingkungan yang juga bervariasi dan dapat diatur, serta pencahayaan yang dapat diatur sesuai dengan kebutuhan.

Pada penelitian tugas akhir ini, *NAOsim* digunakan untuk memvisua-

lisasikan rancangan perilaku robot *Aldebaran NAO* yang akan digunakan saat berinteraksi dengan manusia. Gambar 2.4 mengilustrasikan tampilan *NAOsim* saat sedang dijalankan.



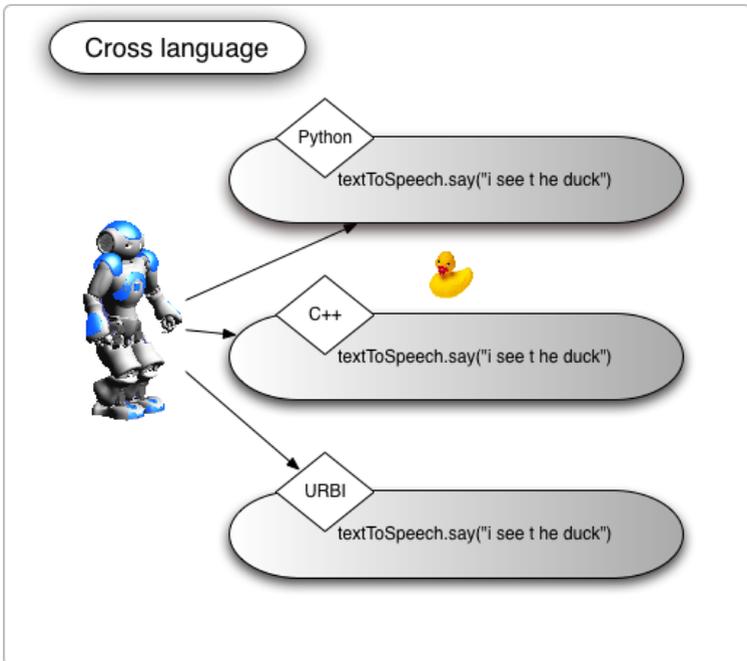
Gambar 2.4 Tampilan *NAOsim*

2.2.4 *NAOqi Framework*

Telah diketahui bahwa untuk melakukan eksplorasi atau perancangan perilaku robot *Aldebaran NAO* dapat dilakukan dengan menggunakan perangkat lunak *Choregraphe*. Akan tetapi, pilihan yang diberikan oleh perangkat lunak *Choregraphe* adalah terbatas, sehingga diperlukan metode lain bagi *developer* yang ingin melakukan eksplorasi lebih jauh terhadap robot *Aldebaran NAO*. *NAOqi Framework* merupakan perangkat lunak yang difungsikan untuk menjembatani para *developer* dengan robot *Aldebaran NAO* pada level pemrograman yang belum atau tidak terdapat pada perangkat lunak *Choregraphe*. Para *developer* dapat merancang sendiri perilaku atau modul robot *Aldebaran NAO* sesuai dengan keinginan. Sebagai perangkat lunak yang berupa *framework*, *NAOqi Framework* menawarkan beberapa pilihan bahasa pemrograman kepada para *developer*, antara lain :

1. Python
2. C++
3. Urbi

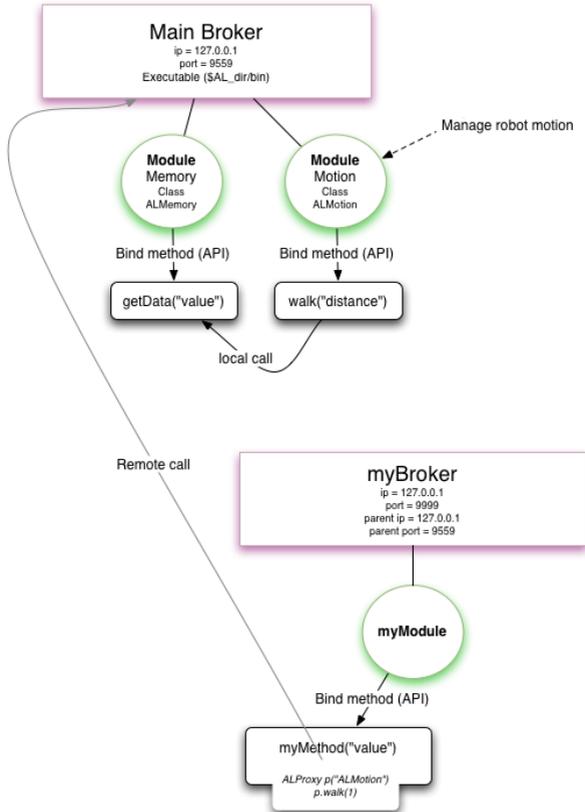
Dengan *NAOqi Framework* ini apapun pilihan para *developer* terkait bahasa pemrograman yang digunakan, semua fungsi pada *NAOqi Framework* mempunyai nama yang sama. Hal ini memberikan kemudahan bagi para *developer* untuk membuat fungsi aplikasi yang sama dengan menggunakan bahasa pemrograman yang berbeda-beda. Gambar 2.5 mengilustrasikan perbandingan antara beberapa bahasa pemrograman yang didukung oleh *NAOqi Framework* dalam hal pemanggilan fungsi atau metode dari *NAOqi Framework*.



Gambar 2.5 Fitur *cross-language* pada *NAOqi Framework*[3]

Dalam pengaplikasiannya, *NAOqi Framework* menggunakan arsitektur *client-server*, yakni ada komponen yang berperan sebagai sebuah *server* yang menunggu untuk melakukan perintah dan komponen yang lain

berperan sebagai *client* yang mengirimkan perintah ke *server*. Gambar 2.6 mengilustrasikan bagaimana komunikasi yang terbangun antara *client* (aplikasi / *local module*) dengan *server* (robot / *remote module*).



Gambar 2.6 Arsitektur komunikasi melalui *NAOqi Framework*[3]

2.3 *Natural Language Processing*

Dalam dunia psikologi neural dan linguistik, *Natural Language* adalah segala macam bahasa yang berasal dari otak manusia dan bersifat spontan (tak sadar). Secara sederhana, *Natural Language* adalah bahasa yang digunakan oleh manusia untuk saling berkomunikasi dengan sesamanya, baik melalui pembicaraan, isyarat, sentuhan, atau tulisan[5]. Selain itu,

Natural Language juga sering dikaitkan dengan bahasa yang dapat dipelajari oleh semua bayi manusia yang memiliki kemampuan kognitif yang normal.

Adapun *Natural Language Processing* adalah cabang dari ilmu komputer, kecerdasan buatan, dan linguistik yang berkaitan dengan interaksi antara komputer dan bahasa manusia (*Natural Language*). Dalam implementasinya, *Natural Language Processing* seringkali diimplementasikan sebagai *chatterbot*³. Pada penelitian tugas akhir ini, *Natural Language Processing* juga diimplementasikan sebagai *chatterbot* yang mengadopsi salah satu *chatterbot* yang sudah ada sebelumnya, yaitu A.L.I.C.E (*Artificial Linguistic Internet Computer Entity*). A.L.I.C.E dalam pengimplementasian *Natural Language Processing*, menggunakan semacam *Markup Language*⁴ yang diberi nama *AIML* (*Artificial Intelligence Markup Language*). Dan pada penelitian tugas akhir ini, *chatterbot* dirancang menggunakan bahasa *AIML* dan difungsikan sebagai komponen kecerdasan buatan dari sistem secara keseluruhan.

2.3.1 *Artificial Intelligence Markup Language*

Artificial Intelligence Markup Language atau disingkat *AIML* pada dasarnya merupakan perluasan dari bahasa XML. Sebagai *Markup Language*, *AIML* memiliki beberapa elemen, antara lain sebagai berikut :

1. *Categories*

Elemen *categories* ditandai dengan *tag* `<category>`. Elemen ini merupakan unit pengetahuan dasar pada *AIML*. Elemen ini setidaknya terdiri dari 2 elemen lain, yakni `<pattern>` dan `<template>`. Jika diberikan contoh sederhana dari sebuah *category* dalam bahasa *AIML*,

```
<category>
  <pattern>SIAPA NAMAMU</pattern>
  <template>Saya robot NAO.</template>
</category>
```

³*Chatterbot* merupakan program komputer yang didesain untuk melakukan percakapan melalui metode tekstual ataupun auditorial. Seringkali *chatterbot* didesain hanya untuk melakukan percakapan ringan sebatas agar lawan percakapannya berpikir bahwa program adalah manusia.

⁴*Markup Language* merupakan sistem penambahan keterangan (anotasi) pada dokumen sehingga isi dokumen dapat dibedakan / dikelompokkan. Contoh *Markup Language* antara lain adalah HTML dan XML.

maka *category* tersebut akan memberikan respon pada masukan “SIAPA NAMAMU” dengan “Saya robot NAO”.

2. *Patterns*

Elemen *patterns* ditandai dengan *tag* <pattern>. Elemen ini digunakan untuk mencocokkan satu atau lebih masukan. Misalkan terdapat *pattern* dengan isi SIAPA NAMAMU, maka masukan yang cocok hanyalah “SIAPA NAMAMU” tanpa memperhatikan apakah masukan terdiri dari huruf besar, kecil, atau keduanya (*case-insensitive*).

3. *Templates*

Elemen *templates* ditandai dengan *tag* <template>. Elemen ini digunakan sebagai respon yang akan dikembalikan berdasarkan *pattern* yang cocok.

2.4 *Object Detection*

Manusia mampu mengenali berbagai macam objek yang diberikan pada suatu citra dengan relatif mudah / tanpa tenaga. Manusia bahkan masih mampu mengenali objek yang memiliki kemiripan atau bervariasi, baik dari segi *view point*, ukuran, skala, jarak, dan transformasi (rotasi atau translasi). Manusia juga masih mampu mengenali objek meskipun beberapa bagian dari objek yang dimaksud tidak tampak dari *view point* manusia.

Visi komputer merupakan cabang dari ilmu komputer yang berkaitan dengan pengambilan, pemrosesan, analisa, dan pemahaman terhadap *image* (citra). Citra secara umum merupakan data yang didapatkan dari kehidupan sehari-hari. Hasil yang didapatkan merupakan informasi yang bersifat numerik atau simbolis, misalnya dalam bentuk pemilihan keputusan.[6]

Salah satu hal yang dikaitkan dengan visi komputer adalah *object detection*. *Object detection* merupakan teknologi komputer yang berkaitan dengan pendeteksian benda dalam citra atau video digital. Adapun benda yang dimaksud dapat dikategorikan ke dalam *class-class* tertentu semisal *human*, *buildings*, atau *cars*.

Class dibedakan berdasarkan pada fitur (*feature*)⁵ yang unik yang dimiliki masing-masing *class*. Sebagai contoh, lingkaran dapat ditemukan dengan mencari objek dengan fitur memiliki jarak yang sama dari satu

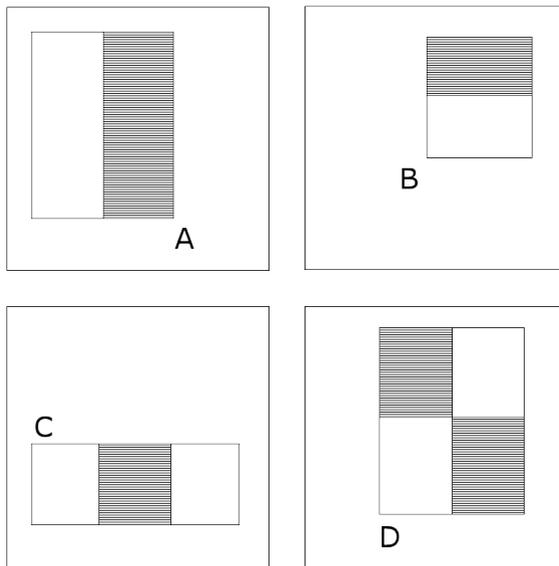
⁵Fitur dalam visi komputer merupakan bagian kecil atau properti dari citra yang mampu dibedakan dari bagian yang lain. Contoh fitur antara lain adalah tepi (*edge*) dan sudut (*corner*).

titik tertentu (pusat lingkaran). Begitu juga persegi dapat ditemukan dengan mencari objek dengan fitur memiliki sudut siku-siku pada sudutnya dan memiliki sisi yang sama panjang.

2.4.1 Face Detection

Pada penelitian tugas akhir ini, *Object Detection* diimplementasikan untuk mendeteksi wajah manusia. Pada penelitian ini, deteksi wajah diimplementasikan dengan menggunakan *Viola-Jones Object Detection Framework*[7] yang diimplementasikan pada OpenCV sebagai sebuah fungsi yang diberi nama `cvHaarDetectObjects()`.

Fitur dari citra yang digunakan pada *Viola-Jones Object Detection Framework* secara umum merupakan jumlah dari pixel citra dalam satu area segiempat tertentu. Fitur ini merupakan fitur umum yang digunakan dalam deteksi objek berbasis citra[8]. Akan tetapi fitur yang digunakan oleh Viola dan Jones adalah lebih dari satu area segiempat. Gambar 2.7 mengilustrasikan fitur yang digunakan oleh Viola dan Jones.



Gambar 2.7 Fitur yang digunakan pada *Viola-Jones Object Detection Framework*[9]

Secara umum algoritma yang digunakan pada *Viola-Jones Object De-*

tection Framework terdiri dari beberapa tahap utama, antara lain adalah sebagai berikut :

1. Pemilihan *Haar Features*

Haar Features merupakan properti yang dimiliki oleh semua citra wajah manusia. Beberapa properti wajah yang memiliki kemiripan antara satu wajah dengan yang lain tersebut disajikan pada tabel 2.3.

Tabel 2.3 *Haar feature* pada *Viola-Jones Object Detection Framework*[9]

Fitur	Ilustrasi	
Daerah hidung lebih terang jika dibanding daerah mata		
Daerah mata lebih gelap jika dibanding daerah pipi		

2. Membuat *Integral Image*

Integral Image merupakan penjumlahan dari beberapa nilai pixel citra pada satu titik (x, y) tertentu. *Integral image* pada satu titik (x, y) adalah jumlah dari nilai pixel citra di atas sampai kiri dari titik (x, y) secara inklusif[10].

$$I(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$

3. *Training Algorithm*

Viola-Jones Object Detection Framework menggunakan *AdaBoost*⁶ da-

⁶*AdaBoost* (“*Adaptive Boosting*”) merupakan algoritma *machine-learning* yang diformulasikan oleh Yoav Freund dan Robert Schapire pada tahun 2003 dan dapat digunakan bersama algoritma *machine-learning* lain untuk meningkatkan performansi.

lam proses *training*-nya. Algoritma ini membangun “*strong*” classifier dari kombinasi linier “*weak*” classifier sederhana yang diberikan bobot.

$$F(x) = a_1f_1(x) + a_2f_2(x) + a_3f_3(x) + \dots$$

dimana F adalah “*strong*” classifier, f adalah “*weak*” classifier, a adalah bobot, dan x adalah objek atau citra. Setiap fitur segiempat adalah “*weak*” classifier dan diinisialisasi mempunyai bobot yang sama. *Ada-Boost* merupakan algoritma yang iteratif, pada setiap iterasinya memilih “*weak*” classifier dan memberikan bobot sesuai dengan akurasi.

2.5 *Speech to Text*

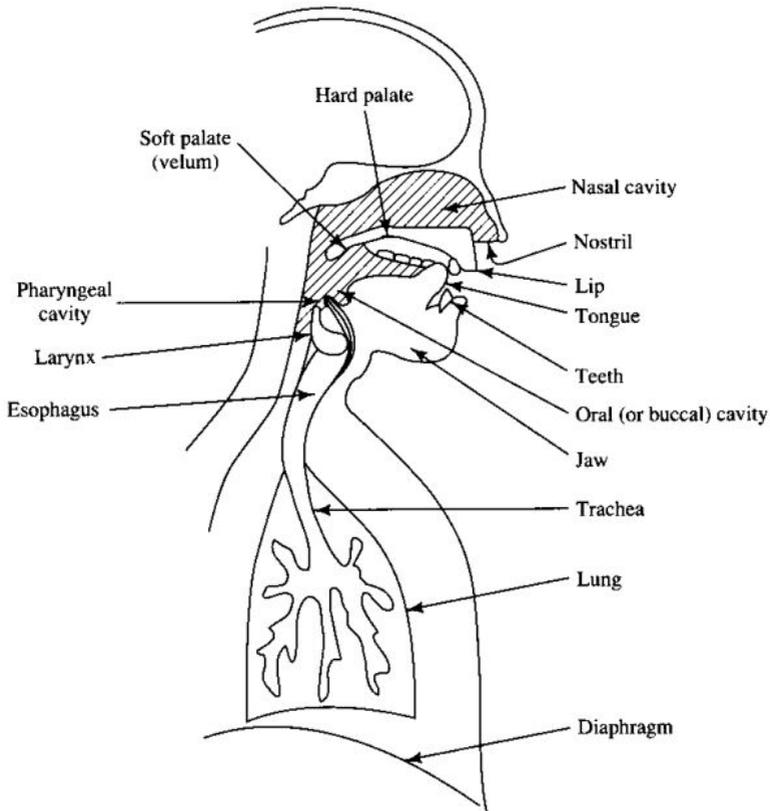
Suara manusia merupakan sesuatu yang unik, hal ini dikarenakan manusia mampu menirukan suara apapun dengan membangkitkan sejumlah frekuensi tertentu sehingga mampu membentuk berbagai macam suara. Daerah frekuensi suara yang bisa didengar oleh telinga manusia berkisar antara 20Hz-20kHz. Namun secara khusus untuk suara manusia dapat diekstraksi pada daerah frekuensi 300-3400Hz yang merupakan kategori suara untuk kualitas sinyal telepon.

Ujaran manusia merupakan gabungan dari *fonem*⁷ yang berbeda-beda untuk setiap bahasa di dunia. Selain itu, suara manusia mampu membentuk berbagai macam jenis suara, antara lain :

1. *Vowel* atau huruf vokal yang dapat dikategorikan menjadi 3 tipe, yakni *front vowel* (misal : iy, ih, eh, ae), *mid vowel* (misal : aa, er, ah, ax, ao), dan *back vowel* (misal : uw, uh, ow).
2. Konsonan atau huruf mati (misal : b, c, d, f, g, dsb.)
3. *Diphthong* atau huruf rangkap, yang merupakan transisi suara dari satu vokal ke vokal yang lain (misal : ai, ay, ae, oy, dsb.)
4. *Semivowel* atau konsonan yang dapat disuarakan layaknya vokal (misal : y, r, w, dsb.)
5. Nasal, yang terbentuk dikarenakan adanya suara yang juga keluar melalui rongga hidung (misal : m, n, ng).
6. *Fricative*, yang merupakan suara yang jika disuarakan memiliki kemiripan dengan *noise* serta memiliki penekanan (misal : v, th, z, f, zh, dsb.)

⁷Satuan terkecil dalam sebuah bahasa yang masih bisa menunjukkan perbedaan makna. Fonem berbentuk bunyi. Contoh dalam bahasa Indonesia bunyi [k] dan [g] adalah dua fonem yang berbeda, misalkan dalam kata “*cakar*” dan “*cahar*”.

Suara manusia yang telah disebutkan dihasilkan oleh organ-organ yang mampu membentuk suara. Gambar 2.8 mengilustrasikan bagian organ tubuh manusia yang mampu menghasilkan suara. Proses *Speech to Text*



Gambar 2.8 Organ Tubuh Penghasil Suara pada Manusia[11]

merupakan proses yang berkaitan dengan proses pengolahan sinyal digital. Dalam proses pengolahan sinyal digital dikenal istilah-istilah yang berkaitan dengan proses *Speech to Text* secara umum antara lain sebagai berikut.

2.5.1 Teori *Sampling*

Proses awal dari pengolahan sinyal digital adalah proses yang melakukan *sampling* terhadap sinyal analog menjadi sinyal diskrit. Jika diberikan

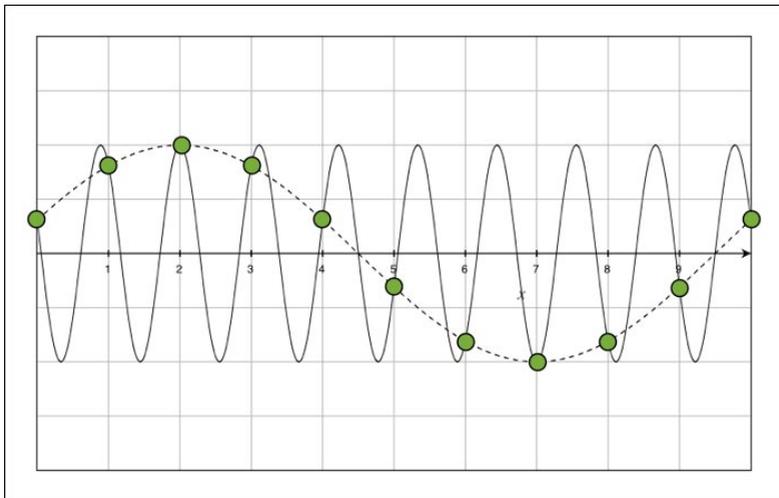
sinyal analog $x_a(n)$ dengan periode *sampling* T , maka sinyal diskrit dapat diperoleh dengan mengikuti persamaan

$$x(n) = x_a(nT)$$

Jika x_a merupakan sinyal dengan frekuensi tunggal ataupun banyak yang memiliki frekuensi maksimum F_{max} , maka agar hasil *sampling* tidak mengalami *aliasing* perbandingan antar frekuensi *sampling* dengan frekuensi maksimum F_{max} sinyal haruslah mengikuti syarat *Nyquist*, yakni

$$F_s \geq F_{max}$$

Adapun yang dimaksud dengan *aliasing* adalah keadaan yang terjadi saat melakukan *sampling* terhadap suatu sinyal dengan frekuensi yang rendah dibandingkan dengan sinyal yang di-*sampling*. Keadaan ini mengakibatkan kesalahan pengukuran frekuensi atau amplitudo. Gambar 2.9 mengilustrasikan kondisi *aliasing*. Pada gambar 2.9, sinyal asli berupa sinyal



Gambar 2.9 Ilustrasi *Aliasing*[12]

sinusoidal diilustrasikan dengan garis (*solid line*) mempunyai frekuensi 0.9Hz dan di-*sampling* pada frekuensi 1.0Hz. Hal ini dapat dilihat dari data hasil *sampling* yang diilustrasikan sebagai titik hijau. Dari data hasil *sampling*, didapatkan sinyal dengan frekuensi 0.1Hz yang diilustrasikan

sebagai garis putus-putus (*dashed line*). Dari ilustrasi ini, didapatkan bahwa sinyal hasil *sampling* yang memiliki frekuensi 0.1Hz tidak mampu merepresentasikan sinyal asli yang memiliki frekuensi 0.9Hz.

2.5.2 Hamming Window

Di dalam desain filter digital dan estimasi *power* spektrum pemilihan *window* akan sangat berpengaruh pada kualitas keseluruhan hasil. Peran utama *window* adalah untuk menghilangkan fenomena Gibbs yang berakibat pada terpotongnya sinyal yang infinitif.

Formula untuk *Hamming Window* dapat dihitung dari persamaan berikut

$$w[k + 1] = 0.54 - 0.46 \cos \left(2\pi \frac{k}{n-1} \right) , k = 0, \dots, n-1$$

2.5.3 Short-Time Fourier Transform

Transformasi *Fourier* merupakan suatu cara untuk menganalisa sinyal multi-frekuensi. Sinyal asal dalam bentuk fungsi waktu, $h(t)$ ditransformasi ke dalam bentuk fungsi frekuensi $h(f)$. Spektrum frekuensi dari suatu sinyal, memiliki daerah waktu tertentu dengan periode tiap *frame* yang tidak terlalu besar. Adapun *Short-Time Fourier Transform* merupakan salah satu jenis metode spektrumisasi sinyal waktu pendek dengan cara melakukan *windowing* terlebih dahulu pada sinyal *Short-Finite* tersebut. Hal ini dilakukan untuk mengurangi fenomena Gibbs. Selanjutnya dilakukan *Fast Fourier Transform* dengan membuat nol daerah sinyal diluar sinyal *Short-Finite* tersebut.

$$S_n (e^{j\omega_i}) = \sum_m s(m)w(n-m)e^{-j\omega_i m}$$

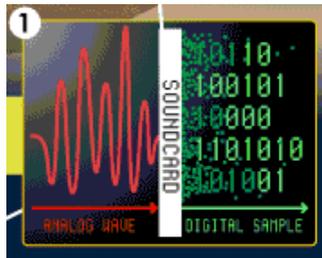
dengan $n = n_0$ maka

$$S_n (e^{j\omega_i}) = FT[s(n)w(n_0 - m)]|_{\omega=\omega_i}$$

Dimana $FT[]$ adalah *Fourier Transform*, $s(m)$ adalah sampel sekuensial, yakni *window* pada titik tengah *frame*, nilai m didapatkan melalui *overlap* dengan *frame* sebelum dan sesudahnya.

Proses *Speech to Text* akan memecah data audio menjadi *fonem-fonem*, mengubahnya menjadi sinyal digital, dan mencocokkannya dengan *data-base* teks yang bersesuaian dengan *fonem*. Gambar 2.10 mengilustrasikan

proses *Speech to Text* secara sederhana.



(a) ADC dan *Sampling* Sinyal Suara



(b) Pemecahan *Fonem* Sinyal Suara



(c) Pencocokkan Sinyal Suara dengan *Database*

Gambar 2.10 Proses *Speech to Text*[13]

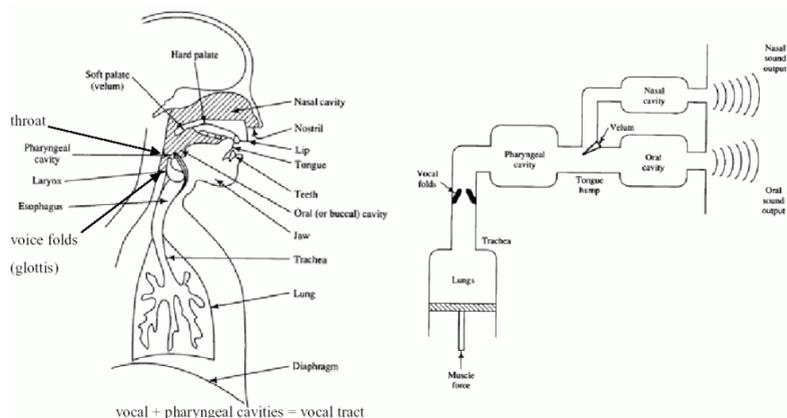
2.6 *Text to Speech*

Pada manusia, proses produksi wicara adalah suatu proses untuk menghasilkan sinyal suara manusia dimana sinyal suara pada awalnya berasal dari otak yang memformulasikan pesan yang ingin disampaikan melalui mulut. Pesan tersebut sebenarnya adalah seperti halnya teks / tulisan yang selanjutnya diubah menjadi kode bahasa (*language code*). Kode bahasa

ini sebenarnya adalah *database* (memori) dalam otak yang mencocokkan pesa dengan kode-kode tersebut. Keluaran dari proses tersebut akan diteruskan oleh proses *neuro-muscular actions* dimana pada proses ini melakukan kontrol terhadap semua aspek gerak artikulasi seperti kontrol bibir, rahang, dan lidah. Secara umum, proses produksi wicara manusia mengikuti beberapa langkah, antara lain sebagai berikut :

1. Terbuka tertutupnya pita suara atau *glottis* (anak tekak) untuk membangkitkan getaran pada udara yang mengalir.
2. Terjadi resonansi rongga *pharyngeal*, rongga hidung, dan rongga mulut.
3. Energi getaran diteruskan oleh udara sekitar.

Gambar 2.11 mengilustrasikan proses produksi wicara pada manusia.

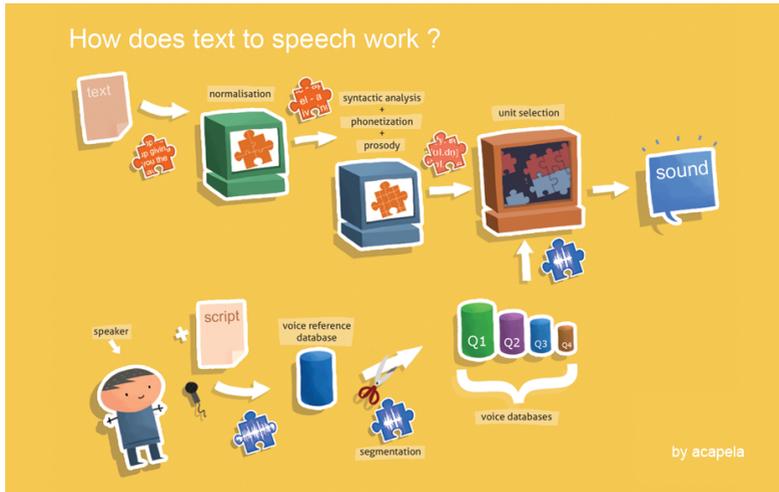


Gambar 2.11 Mekanisme Produksi Wicara Manusia[14]

Dikarenakan tekanan dari *glottis* dan udara yang didorong dari paru-paru, pita suara mampu melakukan gerakan membuka dan menutup dengan sangat cepat, sehingga menimbulkan getaran di udara sekitar. Getaran yang terjadi dimodulasi oleh resonansi dari rongga *pharyngeal*, rongga hidung, atau rongga mulut sehingga menghasilkan *timbre* dari suara.

1. Frekuensi getaran dari pita suara menentukan *pitch* dari suara.
2. Posisi atau bentuk dari bibir, lidah, dan hidung menentukan *timbre* dari suara.
3. Tekanan dari paru-paru menentukan keras lemahnya suara.

Proses produksi wicara ini seringkali diistilahkan sebagai *speech synthesizing* (sintesa suara)[15].



Gambar 2.12 Proses *Text to Speech*[16]

Pada ilmu komputer, proses *Text to Speech* merupakan proses pembangkitan wicara manusia dari data masukan yang berupa teks. Gambar 2.12 mengilustrasikan bagaimana proses *Text to Speech* berlangsung.

Secara umum, proses *Text to Speech* terdiri dari 2 proses, yakni proses pembuatan *database* suara dan proses pencocokan sinyal suara terhadap *database* yang telah dibuat sebelumnya.

Proses pembuatan *database* dilakukan dengan cara merekam wicara manusia. Adapun wicara yang direkam adalah wicara yang mengandung kombinasi kemungkinan suara yang bisa muncul dari satu bahasa tertentu. Wicara ini dapat diambil dari puisi, berita politik, hasil pertandingan olahraga, dan lain sebagainya. Hasil rekaman selanjutnya disegmentasi ke dalam satu atau lebih kriteria berikut, yakni *diphone*, *syllable*, *morpheme*, kata, ungkapan, atau kalimat.

Adapun untuk memproduksi kembali wicara manusia dari data masukan berupa teks, sistem *Text to Speech* akan melakukan analisa linguistik dari data masukan yang berupa teks. Analisa ini akan mengubah data masukan yang berupa teks tertulis menjadi teks fonetik. Dari hasil ini, sistem *Text to Speech* akan mampu mendefinisikan bagaimana kata atau

kalimat dibunyikan. Definisi teks fonetik selanjutnya akan dicocokkan dengan unit-unit suara yang telah disimpan sebelumnya sebagai sebuah *database*. Unit-unit suara yang memiliki tingkat kecocokan paling tinggi nantinya digabungkan menjadi satu suara yang lengkap membentuk suatu kata atau kalimat.

Adapun dalam penelitian tugas akhir ini, sistem *Text to Speech* diimplementasikan menggunakan perangkat lunak *eSpeak* dan *mBrola* dengan *database* suara bahasa Indonesia milik Arry Akhmad Arman⁸.

2.7 BEAST (Bodily Expressive Action Stimulus Test)

Manusia dipandang sebagai makhluk yang paling memiliki kemampuan sosial jika dibandingkan dengan makhluk yang lain. Bahkan dalam kehidupannya, kebanyakan waktunya dihabiskan bersama manusia yang lain, baik dari keluarga, kerabat, teman, atau bahkan orang asing sekalipun. Dikarenakan dalam kesehariannya manusia secara berkelanjutan terlibat dalam interaksi sosial yang *heterogen*, satu manusia mampu menginterpretasikan kondisi dan emosi dari manusia yang lain secara adaptif dan intuitif.

Sejauh ini, penelitian mengenai *Emotion Perception* terfokus pada bagaimana manusia memandang, memproses, dan mengenali ekspresi wajah. Akan tetapi, ada salah satu teori mengenai emosi yang menyatakan bahwa emosi adalah suatu mekanisme yang berkaitan dengan aksi. Dan emosi adalah sesuatu yang mengikuti penilaian manusia terhadap apa yang dialaminya, dan dapat mendorong manusia untuk melakukan aksi secara sengaja maupun tidak sengaja (impulsif), dan reaktif terhadap apa yang sedang dan akan terjadi[17].

Ekspresi tubuh adalah kategori stimulus yang secara alami berkaitan dengan wajah. Salah satu hasil dari penelitian mengenai ekspresi tubuh adalah sebuah *database* yang berisi gambar dari beberapa ekspresi tubuh yang merepresentasikan kondisi dan emosi seseorang dan diberi nama *BEAST (Bodily Expressive Action Stimulus Test)*[18]. Pose *BEAST* dibuat dengan bantuan dari 46 non-profesional aktor (31 wanita). Semua aktor diinstruksikan untuk mengekspresikan 4 macam ekspresi, yakni marah, takut, senang, dan sedih dengan ekspresi tubuh. Keempat ekspresi diambil melalui kejadian sehari-hari yang berkaitan dengan masing-masing ekspresi. Sebagai contoh, ekspresi marah dapat terjadi saat seseorang berada dalam suatu pertengkaran, ekspresi takut dapat terjadi saat seseorang di-

⁸<http://indotts.melsa.net.id/>

kejar oleh penjahat, ekspresi senang dapat terjadi saat seseorang bertemu dengan teman lama yang kebetulan merupakan teman akrab dan sangat senang untuk bisa bertemu kembali, dan ekspresi sedih dapat terjadi saat seseorang mendapat kabar tentang kematian dari seseorang yang berarti baginya. Keseluruhan *database* pose *BEAST* adalah 254 dengan rincian 64 ekspresi marah, 67 ekspresi takut, 61 ekspresi senang, dan 62 ekspresi sedih. Gambar 2.13 menunjukkan beberapa contoh dari pose *BEAST* untuk masing-masing ekspresi, yakni marah, takut, senang, dan sedih.



(a) Angry



(b) Fear



(c) Happy



(d) Sad

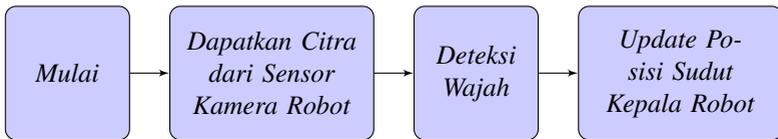
Gambar 2.13 Pose *BEAST*[18]

(Halaman ini sengaja dikosongkan)

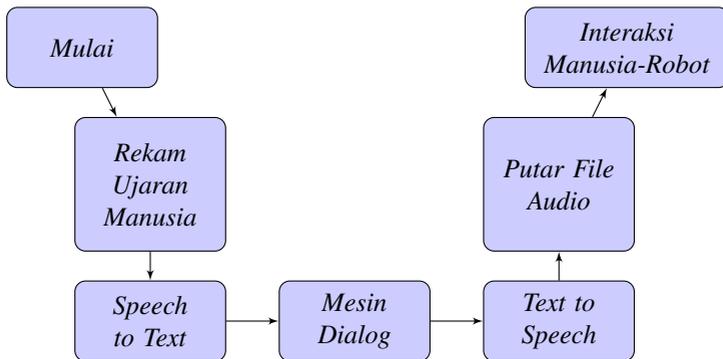
BAB 3 DESAIN SISTEM DAN IMPLEMENTASI

3.1 Konsep Sistem

Pada penelitian ini, sistem dirancang untuk menerapkan komunikasi interaktif antara manusia dengan robot *Aldebaran NAO* menggunakan deteksi wajah dan ujaran bahasa Indonesia. Gambar 3.1 mengilustrasikan sistem secara keseluruhan.



(a) Interaksi berbasis Deteksi Wajah



(b) Interaksi berbasis Suara Bahasa Indonesia

Gambar 3.1 Konsep Sistem

Pada sistem ini, suara dari operator berfungsi sebagai masukan sistem. Saat sistem mendeteksi wajah manusia, maka sistem akan mulai mendeteksi suara yang ada di sekitarnya. Jika sistem mendeteksi adanya suara di sekitarnya, maka sistem akan melakukan *tracking* pada wajah manusia yang terdeteksi sebelumnya. Suara yang terdeteksi akan dianalisa untuk kemudian didapatkan fitur-fitur yang merepresentasikan informasi yang terdapat dalam suara yang dimaksud. Informasi yang didapatkan akan dijadikan sebagai masukan komponen kecerdasan buatan

pada sistem. Komponen kecerdasan buatan pada sistem merupakan suatu subsistem berupa gabungan dari unit-unit pengetahuan sederhana. Komponen kecerdasan buatan akan melakukan analisa terhadap unit-unit pada informasi yang diberikan untuk kemudian didapatkan keluaran yang bersesuaian dengan masukan yang diberikan. Keluaran dari komponen kecerdasan buatan akan dijadikan sebagai masukan komponen aktuator⁹ pada sistem. Komponen aktuator pada sistem akan mengeksekusi perintah berupa gerakan maupun ujaran yang bersesuaian dengan masukan yang diberikan.

3.2 Alur Kerja Sistem

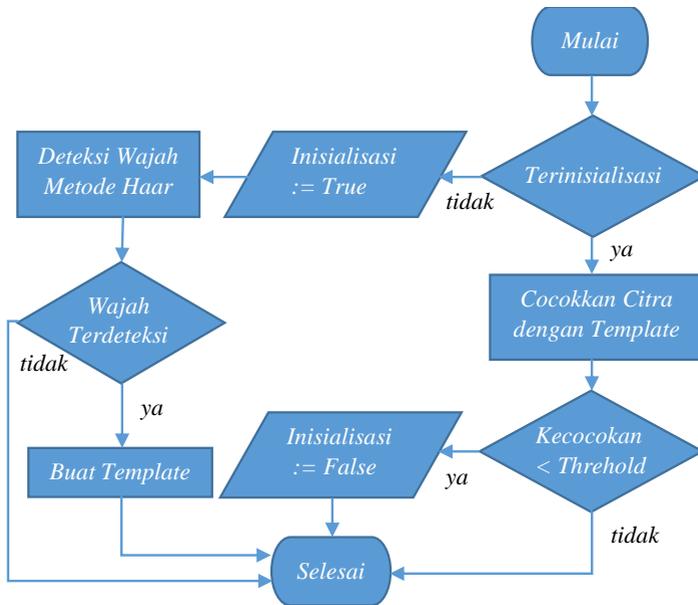
Berdasarkan konsep sistem yang telah dijelaskan, alur kerja sistem mulai dari awal hingga akhir secara keseluruhan dapat dibagi menjadi dua bagian, yaitu : bagian yang berkaitan dengan interaksi berbasis deteksi wajah dan bagian yang berkaitan dengan interaksi berbasis suara bahasa Indonesia.

Gambar 3.1a mengilustrasikan bagaimana alur kerja sistem dalam melakukan interaksi berbasis deteksi wajah. Secara garis besar, pada mulanya sistem mendapatkan citra dari sensor kamera robot. Selanjutnya sistem melakukan proses deteksi wajah pada citra yang telah didapatkan. Hasil dari proses deteksi wajah adalah koordinat posisi dari wajah yang terdeteksi. Adapun selanjutnya, koordinat yang didapatkan akan digunakan sebagai masukan untuk meng-*update* posisi sudut dari kepala robot. Adapun selanjutnya, sistem akan kembali mendapatkan citra dari sensor kamera robot dan melakukan proses yang sama secara terus menerus.

Proses deteksi wajah pada sistem menggunakan algoritma *Template Matching* yang diilustrasikan pada gambar 3.2. Blok “Mulai” pada ilustrasi merupakan proses sebelumnya, yakni proses untuk mendapatkan citra. Dan blok “Selesai” pada ilustrasi merupakan proses selanjutnya, yakni proses untuk melakukan *update* posisi sudut kepala robot.

Adapun mengenai bagaimana alur kerja sistem dalam melakukan interaksi berbasis suara bahasa Indonesia diilustrasikan pada gambar 3.1b. Secara garis besar, pada mulanya sistem mendapatkan ujaran manusia dengan cara melakukan perekaman ujaran manusia ke dalam sebuah *file*. Selanjutnya *file* yang berisi ujaran manusia dianalisa sehingga didapatkan

⁹Aktuator adalah komponen mekanik untuk menggerakkan atau meengontrol sistem. Aktuator yang dimaksud dalam penelitian tugas akhir ini dapat berupa *joint* atau *speaker* dari robot.



Gambar 3.2 Alur Kerja Algoritma *Template Matching*

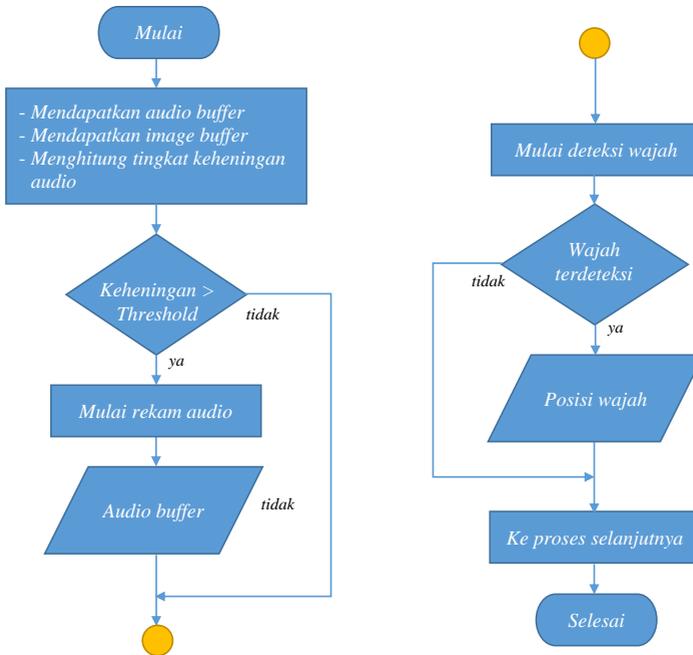
informasinya dalam bentuk teks. Adapun teks yang didapatkan dijadikan sebagai masukan dari komponen kecerdasan buatan. Selanjutnya komponen kecerdasan buatan memberikan respon yang juga berupa teks. Dan akhirnya informasi dalam teks yang diberikan oleh komponen kecerdasan buatan dirupakan ke dalam *file* audio yang kemudian dikirimkan ke robot *Aldebaran NAO* sebagai ujaran yang dibunyikan melalui *speaker* robot. Ujaran yang dibunyikan melalui *speaker* robot akan membuat robot *Aldebaran NAO* seolah-olah mampu berinteraksi dengan manusia.

3.3 Implementasi Sistem

Secara keseluruhan sistem diimplementasikan ke dalam sebuah program aplikasi *Desktop* yang diberi nama *NANA (NANA Aldebaran NAO App)* yang berperan sebagai modul lokal (*client*) yang mengirim permintaan kepada modul *remote (server)*. Adapun yang berperan sebagai *server* adalah robot *Aldebaran NAO*. Komunikasi antara *server* dan *client* adalah menggunakan media kabel UTP (*Unshielded Twisted Pair*).

3.3.1 Proses Penangkapan Informasi

Proses penangkapan informasi terdiri dari 2 bagian, yakni pendeteksian wajah oleh sensor kamera dari robot *Aldebaran NAO* dan perekaman suara oleh sensor *microphone* dari robot *Aldebaran NAO*. Proses pendeteksian wajah diimplementasikan sebagai sebuah *routine*¹⁰ yang menerima data berupa *image buffer* dari sensor kamera robot *Aldebaran NAO* melalui *NAOqi Framework* dan melakukan ekstraksi fitur-fitur wajah dengan menggunakan bantuan dari *library OpenCV*¹¹. Adapun proses perekaman suara juga diimplementasikan sebagai sebuah *routine* yang menerima data berupa *audio buffer* dari *microphone*. Proses penangkapan informasi mengikuti diagram alir yang diilustrasikan pada Gambar 3.3.



Gambar 3.3 Diagram Alir Proses Penangkapan Informasi

¹⁰Satu blok bagian dari aplikasi atau program, umumnya memiliki satu fungsi tertentu dan nama yang bersesuaian dengan fungsinya.

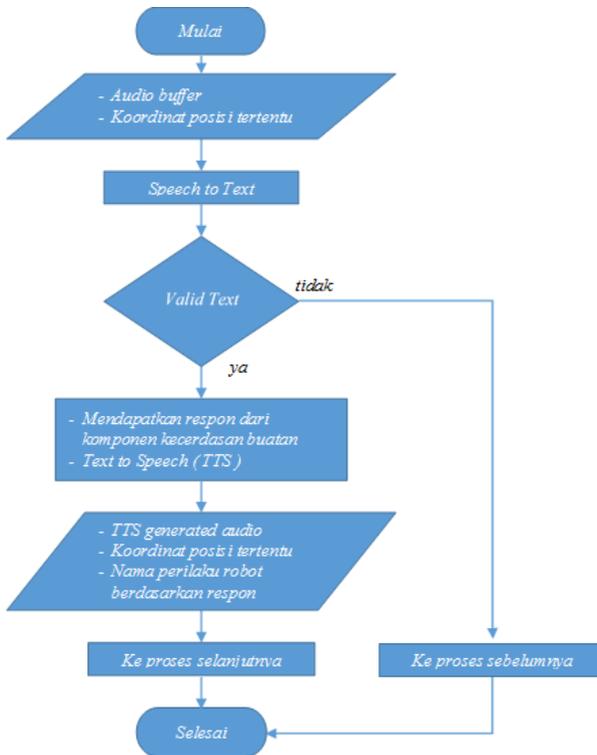
¹¹*OpenCV (Open Source Computer Vision)* merupakan perangkat lunak pengolah citra dan *machine learning*

Secara detail, proses penangkapan informasi dimulai dengan mendapatkan *image buffer* dan *audio buffer* dari sensor kamera robot *Aldebaran NAO* dan sebuah *microphone*. Tingkat keheningan *audio buffer* kemudian dianalisa dan dibandingkan terhadap nilai *threshold* tertentu untuk penentuan ada atau tidaknya “suara” dalam *audio buffer*. Jika sistem mendeteksi adanya “suara” dalam *audio buffer*, maka sistem akan mulai melakukan perekaman suara. Selanjutnya sistem akan mulai melakukan deteksi wajah dalam *image buffer* yang diterima dan wajah yang terdeteksi oleh sistem akan disimpan posisinya. Adapun jika sistem tidak mendeteksi adanya wajah pada *image buffer* ataupun tidak mendeteksi adanya “suara” dalam *audio buffer*, maka selanjutnya akan meneruskan ke proses pengolahan data dengan data posisi wajah dan *audio buffer*.

3.3.2 Pengolahan Data

Proses pengolahan data terdiri dari 3 bagian, yakni proses *Speech to Text* pada *audio buffer* yang diterima dari proses penangkapan informasi, proses mendapatkan respon komponen kecerdasan buatan, dan proses *Text to Speech* pada respon komponen kecerdasan buatan. Proses *Speech to Text* diimplementasikan sebagai sebuah *routine* yang menerima *audio buffer* dari proses penangkapan informasi dan melakukan ekstraksi *fonem* dengan menggunakan bantuan dari *Google Speech API*. Komponen kecerdasan buatan diimplementasikan dengan menggunakan bantuan *library AIML*. Adapun proses *Speech to Text* diimplementasikan dengan menggunakan bantuan perangkat lunak *eSpeak* dan *MBrola*. Proses penangkapan informasi mengikuti diagram alir yang diilustrasikan pada Gambar 3.4.

Secara detail, proses pengolahan data dimulai dengan proses *Speech to Text* menggunakan bantuan *Google Speech API*. Proses dimulai dengan sistem mengirimkan *HTTP Request* kepada *Google* dengan data *audio buffer* yang didapatkan pada proses penangkapan informasi. Sebelum melakukan pengiriman, untuk dapat menggunakan fitur *Google Speech API*, sistem terlebih dahulu melakukan *formatting* pada *audio buffer* sesuai dengan spesifikasi yang diberikan oleh *Google*. Selanjutnya *Google* akan mengirimkan respon kepada sistem berupa teks yang dikodekan ke dalam format *JSON (JavaScript Object Notation)*. Jika respon yang dikirim oleh *Google* adalah kode *JSON* yang valid, maka teks akan diteruskan ke komponen kecerdasan buatan. Komponen kecerdasan buatan akan mencocokkan teks hasil *Speech to Text* dengan masukan unit-unit pengetahuan yang dimilikinya dan mengembalikan keluaran berupa teks yang



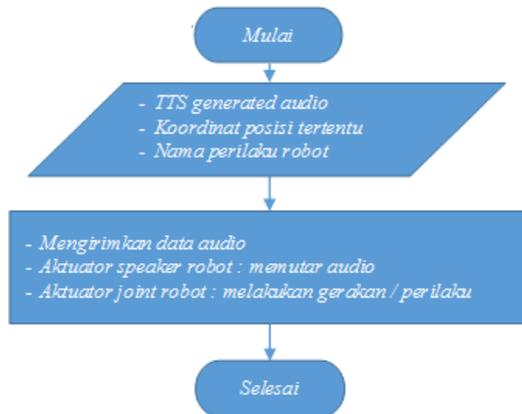
Gambar 3.4 Diagram Alir Proses Pengolahan Data

bersesuaian dengan masukan yang diberikan. Respon teks yang diberikan oleh komponen kecerdasan buatan selanjutnya akan disintesa menjadi *audio*. Respon teks selanjutnya dicocokkan dengan *database* perilaku robot yang dimiliki oleh komponen kecerdasan buatan. Sistem selanjutnya akan meneruskan ke proses eksekusi perintah dengan data *audio* yang disintesa pada proses *Speech to Text*, nama perilaku robot, dan koordinat posisi yang didapat dari proses penangkapan informasi. Adapun jika respon yang dikirim bukan merupakan kode *JSON* yang valid, maka sistem akan kembali melakukan proses penangkapan informasi.

3.3.3 Eksekusi Perintah / Aktuator

Proses eksekusi perintah oleh aktuator terdiri dari 2 bagian, yakni proses pengiriman data *audio* hasil sintesa dan eksekusi perilaku sesuai de-

ngan nama yang diberikan pada proses pengolahan data. Proses pengiriman data *audio* diimplementasikan sebagai sebuah *routine* yang mengimplementasikan protokol *SSH2* sebagai metode komunikasinya. Adapun eksekusi perilaku dilakukan oleh sistem melalui *NAOqi Framework*. Proses eksekusi perintah mengikuti diagram alir yang diilustrasikan pada Gambar 3.5.



Gambar 3.5 Diagram Alir Proses Eksekusi Perintah oleh Aktuator

Secara detail, proses dimulai dengan sistem melakukan pengiriman data *audio* hasil sintesa ke robot *Aldebaran NAO*. Selanjutnya sistem melalui *NAOqi Framework* mengirimkan permintaan ke robot *Aldebaran NAO* untuk melakukan *playback* atau memutar *audio* ke *loudspeaker*-nya. Secara bersamaan sistem juga mengirimkan permintaan ke robot *Aldebaran NAO* untuk melakukan gerakan (*gesture*) sesuai dengan perilaku dan menghadap ke arah koordinat posisi tertentu yang diberikan.

3.4 Desain Kecerdasan Buatan

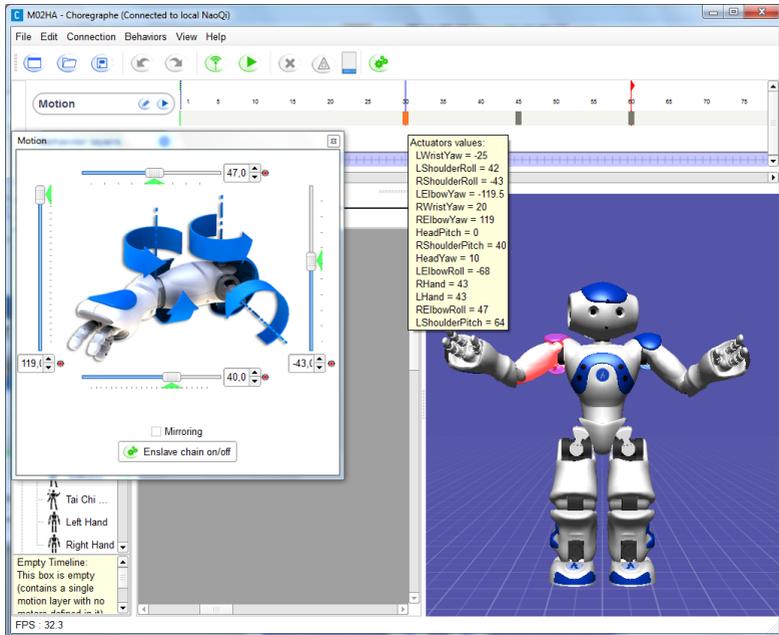
Pada penelitian tugas akhir ini, komponen kecerdasan buatan sistem dibuat dengan menggunakan perangkat lunak *AIML (Artificial Intelligence Markup Language)* yang diimplementasikan pada bahasa pemrograman *Python*. Komponen kecerdasan buatan dari sistem memiliki unit pengetahuan yang terbatas. Unit-unit pengetahuan yang dimiliki oleh sistem hanya terbatas pada pengetahuan mengenai salam perkenalan (*greeting*) dan sedikit basa-basi sederhana. Secara rinci, unit pengetahuan yang dimiliki oleh komponen kecerdasan buatan sistem antara lain meliputi :

1. Pengetahuan sederhana mengenai waktu, yakni pagi, siang, sore, dan malam.
2. Pengetahuan sederhana mengenai diri sendiri, yakni nama dan keadaan diri.
3. Pengetahuan sederhana mengenai cara berinteraksi dengan manusia, yakni berkenalan.

Dalam implementasinya, komponen kecerdasan buatan yang terdiri dari unit-unit pengetahuan dikelompokkan ke dalam beberapa kategori tertentu berdasarkan topik atau keterkaitannya dan disimpan dalam bentuk *file*. *File-file* ini selanjutnya akan dimuat saat sistem mulai dijalankan.

3.5 Desain Perilaku Robot

Untuk membuat robot *Aldebaran NAO* yang interaktif, selain pembicaraan yang interaktif, diperlukan juga pendukung yang meningkatkan keinteraktifan dari robot, yakni gerakan-gerakan komunikatif yang mampu merepresentasikan suasana atau keadaan pada satu pembicaraan tertentu. Pada robot *Aldebaran NAO*, gerakan-gerakan yang dilakukan oleh robot dikenal dengan istilah perilaku (*behavior*). Perilaku robot adalah perangkat lunak berupa *file* yang berisi posisi-posisi *joint* robot *Aldebaran NAO* pada satu *range* waktu tertentu. Selain itu juga terdapat beberapa *script* yang mendukung pergerakan atau perubahan posisi-posisi *joint* dari robot. Pada penelitian tugas akhir ini, perilaku robot dibuat menggunakan perangkat lunak *Choregraphe*. Dengan menggunakan perangkat lunak *Choregraphe*, perilaku robot dapat dibuat dengan relatif mudah. Dengan sebuah *Toolbox Empty Timeline* pada *Choregraphe*, nilai-nilai *joint* dari robot *Aldebaran NAO* dapat disimpan pada satu waktu tertentu. Gambar 3.6 mengilustrasikan proses pembuatan perilaku robot. Pada gambar 3.6 perilaku robot adalah sepanjang 60 satuan waktu. Hal ini dapat dilihat pada adanya tanda bendera merah pada satuan waktu ke 60. Posisi *joint* pada satu waktu tertentu dari robot dapat diatur dengan memilih waktu saat *joint* yang diinginkan bergerak dan memilih *joint* yang dimaksud pada ilustrasi robot, maka akan muncul kotak dialog yang mengilustrasikan nilai-nilai *joint* robot yang saling berdekatan, dan memasukkan nilai yang diinginkan pada *textbox* yang disediakan. Pada ilustrasi yang diberikan pada gambar 3.6, waktu saat *joint* yang diinginkan bergerak adalah pada satuan waktu ke 45 dan kumpulan *joint* yang dipilih adalah lengan kanan. Pada kumpulan *joint* ini, terdapat *joint RShoulderRoll*, *RShoulderPitch*, *RElbowRoll*, dan *RElbowYaw*. Setelah memasukkan nilai yang diinginkan



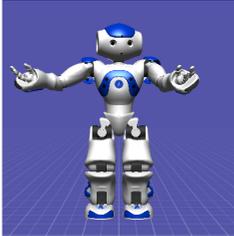
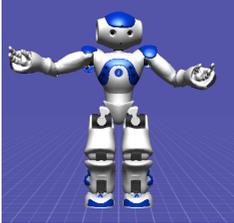
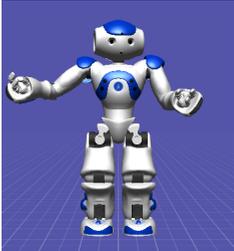
Gambar 3.6 Pembuatan *Behavior* pada *Choregraphe*

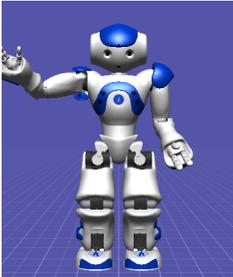
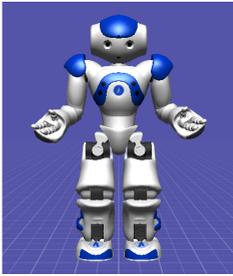
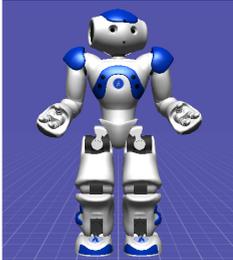
an, ilustrasi robot pun akan segera diperbarui mengikuti nilai yang diberikan pada kotak dialog.

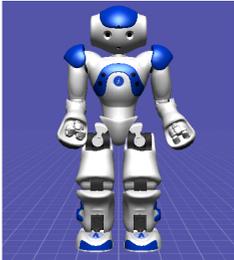
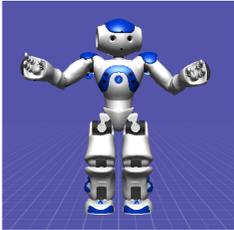
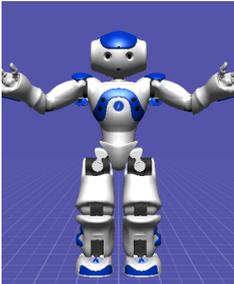
Untuk membuat perilaku robot yang interaktif, perilaku robot dibuat dengan menirukan pose *BEAST* (*Bodily Expressive Action Stimulus Test*). Pose *BEAST* yang digunakan dalam penelitian tugas akhir ini adalah pose-pose yang menunjukkan ekspresi bahagia. Pose *BEAST* yang digunakan dalam penelitian tugas akhir ini diberikan pada Tabel 3.1.

Tabel 3.1 Daftar Pose *BEAST* yang Digunakan dalam Penelitian

Kode Pose	Pose <i>BEAST</i>	Implementasi pada robot <i>Aldebaran NAO</i>
-----------	-------------------	--

Kode Pose	Pose <i>BEAST</i>	Implementasi pada robot <i>Aldebaran NAO</i>
F01HA		
F06HA		
F13HA		

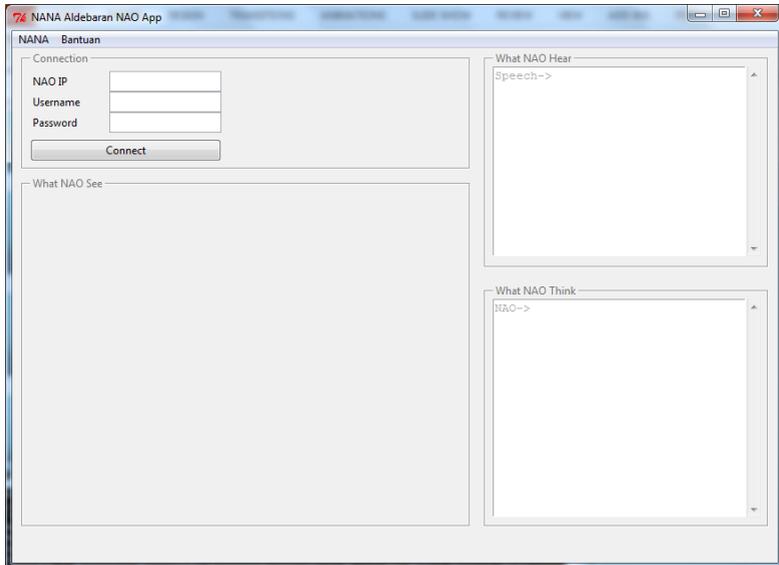
Kode Pose	Pose <i>BEAST</i>	Implementasi pada robot <i>Aldebaran NAO</i>
F22HA2		
F29HA2		
F29HA3		

Kode Pose	Pose <i>BEAST</i>	Implementasi pada robot <i>Aldebaran NAO</i>
F33HA		
M02HA		
M10HA		

3.6 Desain Antarmuka

Dalam pengerjaan tugas akhir ini, antarmuka sistem dibuat dengan menggunakan *library Tcl/Tk* yang diimplementasikan pada bahasa pemrograman *Python*.

Gambar 3.7 memberikan ilustrasi mengenai desain antarmuka dari sistem. Desain antarmuka sistem dibuat sedemikian rupa sehingga terbagi



Gambar 3.7 Desain antarmuka *NANA*

menjadi beberapa bagian sesuai dengan fungsi dan alur yang akan dijalankan dari program. Bagian-bagian dari antarmuka tersebut diberi nama sebagai berikut :

1. Bagian *Connection*

Bagian ini digunakan untuk mengatur koneksi antara sistem dengan robot *Aldebaran NAO*. Pada bagian ini terdapat 3 buah *textbox* yang disusun secara vertikal dan 2 buah *button*. *Textbox* pertama terletak pada baris paling atas dan terdapat label IP, diisi dengan alamat IP dari robot *Aldebaran NAO*. *Textbox* kedua dan ketiga yang masing-masing terdapat label *Username* dan *Password*, diisi dengan *username* dan *password* yang digunakan untuk *login* ke alamat IP robot *Aldebaran NAO*. *Button* yang memiliki label *Connect* digunakan untuk melakukan ko-

neksi ke robot *Aldebaran NAO* sesuai dengan informasi yang diberikan oleh 3 *textbox* di atasnya dan memulai keseluruhan proses dari sistem.

2. Bagian *What NAO See*

Bagian ini digunakan untuk menampilkan hasil dari *image* yang ditangkap oleh kamera robot *Aldebaran NAO* dan diproses oleh sistem.

3. Bagian *What NAO Hear*

Pada bagian ini terdapat sebuah *textbox* yang berisi informasi mengenai hasil *Speech to Text* yang diperoleh pada proses pengolahan data.

4. Bagian *What NAO Say*

Pada bagian ini terdapat sebuah *textbox* yang berisi informasi mengenai respon teks yang diberikan oleh komponen kecerdasan buatan sistem pada proses pengolahan data.

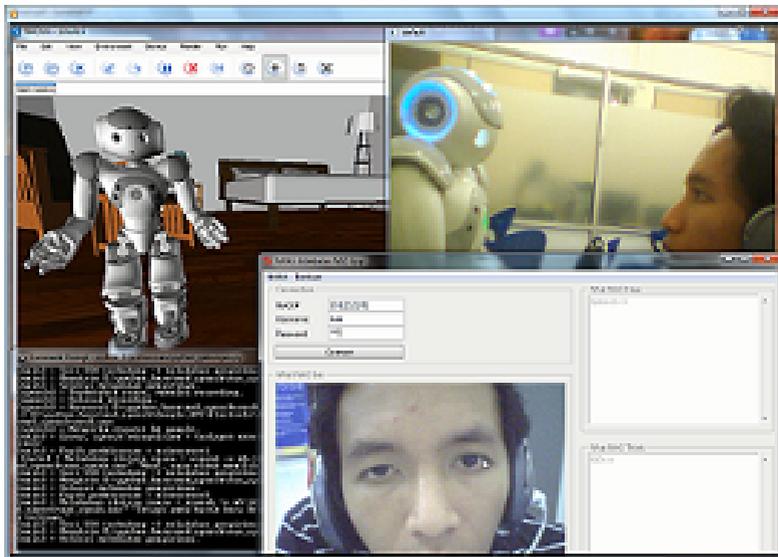
BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini akan dipaparkan hasil pengujian dan analisa dari desain sistem dan implementasi yang telah dibahas pada bab III. Bentuk pengujian yang dilakukan untuk menguji *NANA* (*NANA Aldebaran NAO App*) adalah sebagai berikut.

4.1 Pengujian *Speech Recognition*

Pada pengujian *Speech Recognition*, hal yang diamati adalah bagaimana *NANA* merespon masukan suara yang diberikan oleh pengguna. Pada penelitian tugas akhir ini, metode *Speech Recognition* yang digunakan adalah menggunakan *Google Speech API*, sehingga *NANA* membutuhkan akses internet untuk melakukan proses *Speech Recognition*. Adapun jika akses internet tidak tersedia, maka *NANA* akan memberikan respon yang menyatakan bahwa akses internet tidak tersedia dan memberitahukan bahwa pengguna tetap dapat berkomunikasi dengan *NANA* melalui media tulisan (*chat*). Gambar 4.1 mengilustrasikan kasus saat *NANA* mendeteksi tidak tersedianya akses internet.



Gambar 4.1 *NANA* tidak mendeteksi adanya akses internet

Adapun jika *NANA* tidak mendeteksi adanya masalah pada akses in-

ternet, maka *NANA* akan mampu mendeteksi teks pada data audio yang diterimanya. Dari data audio yang diterima, hampir semua kata mampu dideteksi dengan baik. Gambar 4.2a menunjukkan beberapa *sample* kata dari data audio yang diterima.

```
What NAO Hear
Speech->
Speech-> halo
Speech-> nama saya rizal
Speech-> siapa namamu
Speech-> apa kabar now
Speech-> apa kabar now
```

(a)



(b)

Gambar 4.2 Performansi *Speech to Text*

Dari total 312 kata yang diperoleh dari 13 orang partisipan, 251 kata berhasil terdeteksi dengan tepat. Adapun sisanya tidak dapat terdeteksi dengan tepat. Gambar 4.2b mengilustrasikan perbandingan total kata yang berhasil terdeteksi dengan tepat dengan total kata yang tidak terdeteksi dengan tepat. Ketidaktepatan dapat disebabkan oleh beberapa faktor, antara lain *noise* yang ada disekitar sumber suara, *volume* sumber suara yang terlalu rendah, atau *NANA* yang belum siap untuk menerima masukan suara.

4.2 Pengujian Kemampuan Kecerdasan Sistem

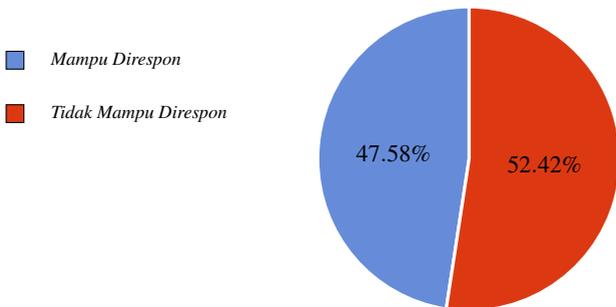
Pada pengujian kemampuan kecerdasan sistem, hal yang diamati adalah sejauh mana *NANA* mampu merespon terhadap masukan dari peng-

guna yang memiliki kemungkinan yang banyak. Masukan dari pengguna dicocokkan dengan *pattern-pattern* tertentu, untuk mengatasi banyaknya kemungkinan masukan. Tabel 4.1 menyajikan daftar contoh masukan percakapan beserta respon keluaran yang diberikan oleh *NANA* berdasarkan variasi *pattern* yang dimilikinya.

Tabel 4.1 Daftar Percakapan

<i>Pattern</i>	Masukan Cocok	Respon
1. HELLO 2. HALO 3. HALO * 4. HAI 5. HAI *	1. halo 2. hai	1. Ya 2. Halo 3. Hai
1. SELAMAT *	1. Selamat pagi 2. Selamat siang 3. Selamat sore 4. Selamat malam	1. Selamat pagi 2. Selamat siang 3. Selamat sore 4. Selamat malam
1. HOW ARE YOU 2. APA KABAR 3. APA KABAR * 4. _APA KABAR	1. Apa kabar 2. Apa kabar NAO 3. Halo apa kabar	1. Saya baik-baik saja, terima kasih
1. WHO ARE YOU 2. SIAPA NAMAMU 3. SIAPA KAMU *	1. Siapa namamu?	1. Saya NAO 2. Saya robot NAO

<i>Pattern</i>	Masukan Cocok	Respon
1. NAMA SAYA *	1. Nama saya rizal	<ol style="list-style-type: none"> 1. Senang bisa berkenalan dengan Anda, rizal 2. Salam kenal, rizal
1. *	1. Berapa jumlah roda mobil?	<ol style="list-style-type: none"> 1. Maaf, saya tidak mengerti maksud Anda 2. Maaf, apa yang sebenarnya sedang Anda bicarakan? 3. Maaf?



Gambar 4.3 Performansi Mesin Dialog

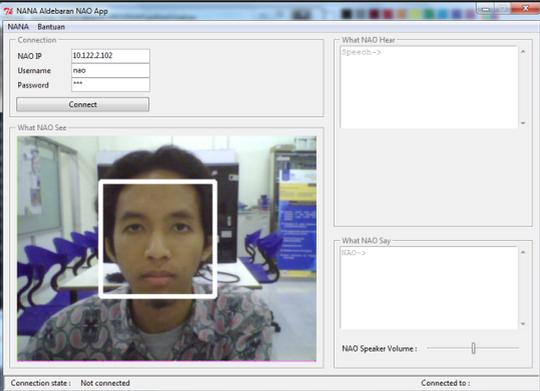
Masukan yang tidak memiliki kecocokkan dengan *pattern* yang ada, akan dicocokkan ke *pattern* * yang berarti *NANA* “tidak mampu” memberikan respon yang tepat kepada pengguna dan memberikan respon bahwa *NANA* tidak mengerti apa yang dimaksud oleh pengguna. Dari total

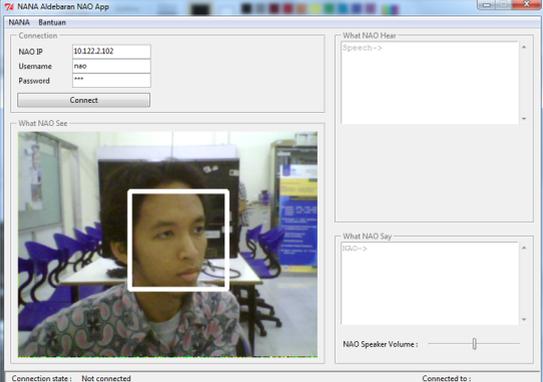
124 masukan yang diperoleh dari 13 partisipan, 59 masukan mampu direspon oleh *NANA*. Adapun sisanya tidak mampu direspon. Gambar 4.3 mengilustrasikan perbandingan total masukan yang mampu direspon oleh *NANA* dengan masukan yang tidak mampu direspon oleh *NANA*. Ketidakmampuan *NANA* untuk merespon masukan dari partisipan yang paling utama disebabkan oleh keterbatasan total variasi pasangan masukan keluaran yang didefinisikan pada mesin dialog. Selain itu, kesalahan pada proses *Speech to Text* juga dapat memberikan masukan yang tidak memiliki kecocokan dengan variasi-variasi pasangan masukan keluaran pada mesin dialog, terkecuali *pattern* *.

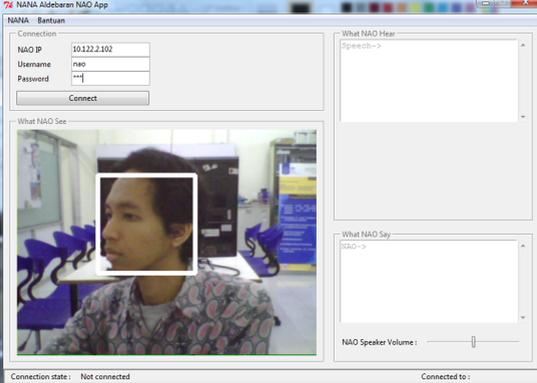
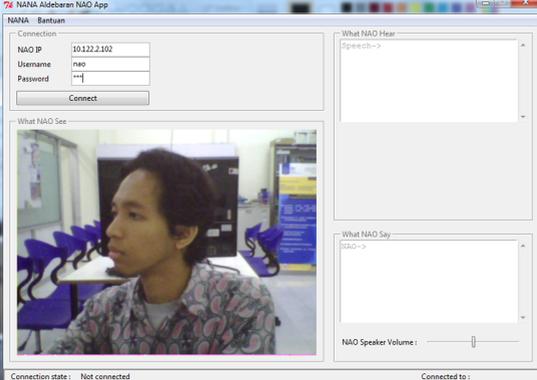
4.3 Pengujian Kemampuan Deteksi Wajah

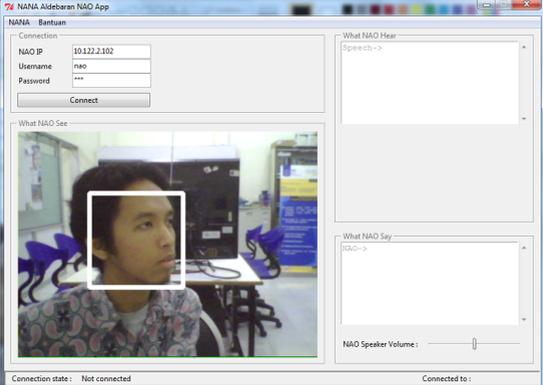
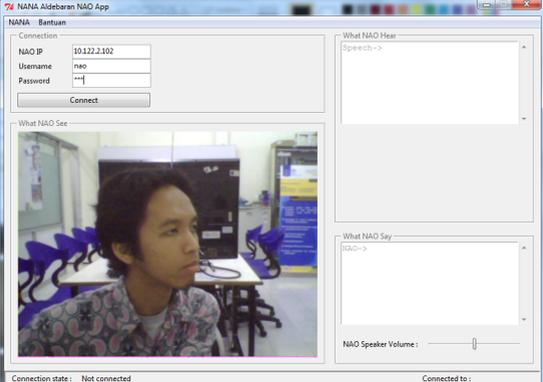
Pada pengujian kemampuan deteksi wajah, hal yang diamati adalah berapa *range* yang sudut yang harus dipenuhi agar *NANA* mampu mendeteksi wajah dari pengguna. Percobaan dilakukan dengan melakukan sudut 0° , 30° *clockwise* dan *counter clockwise*, 45° *clockwise* dan *counter clockwise*. Percobaan performansi deteksi wajah untuk sudut 30° dan 45° dilakukan dengan melakukan gerakan menoleh dari sudut 0° ke arah *clockwise* maupun *counter clockwise* masing-masing sebesar 30° dan 45° .

Tabel 4.2 Performansi *NANA* dalam Deteksi Wajah

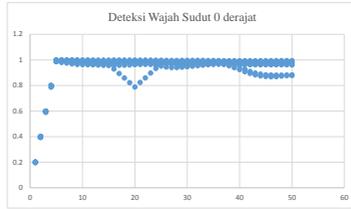
Sudut	Wajah yang Terdeteksi oleh <i>NANA</i>	Keterangan
0°		<i>NANA</i> mampu mendeteksi wajah dengan baik

Sudut	Wajah yang Terdeteksi oleh NANA	Keterangan
<p>30° (clockwise)</p>		<p>NANA mampu mendeteksi wajah dengan baik</p>
<p>30° (counterclockwise)</p>		<p>NANA mampu mendeteksi wajah dengan baik</p>

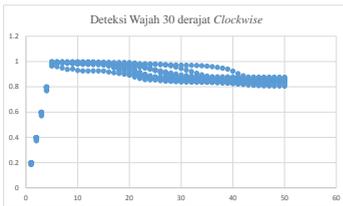
Sudut	Wajah yang Terdeteksi oleh NANA	Keterangan
<p>45° (clockwise)</p>		<p>2 sampai 3 detik pertama NANA mampu mendeteksi wajah dengan baik</p>
<p>45° (clockwise)</p>		<p>Setelah lebih dari 3 detik, NANA tidak mampu mendeteksi adanya wajah</p>

Sudut	Wajah yang Terdeteksi oleh NANA	Keterangan
<p>45° (<i>counter clockwise</i>)</p>		<p>2 sampai 3 detik pertama NANA mampu mendeteksi wajah dengan baik</p>
<p>45° (<i>counter clockwise</i>)</p>		<p>Setelah lebih dari 3 detik, NANA tidak mampu mendeteksi adanya wajah</p>

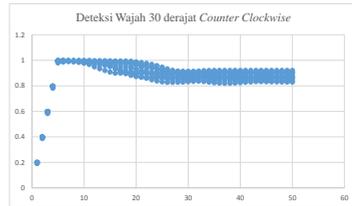
Dari tabel 4.2 dapat diperhatikan bahwa **NANA** pada sudut 45° baik *clockwise* maupun *counter clockwise* mampu mendeteksi wajah dengan baik pada beberapa detik awal. Akan tetapi selanjutnya, **NANA** sudah tidak mampu lagi mendeteksi wajah. Hal ini dikarenakan **NANA** menerapkan sistem *Template Matching* yang membuat **NANA** mampu untuk mendeteksi wajah menggunakan *template* yang telah didapat pada proses sebelumnya. Gambar 4.4 menunjukkan persebaran *confidence* dari metode *Template Matching* untuk sudut 0°, 30°, 45° *clockwise* maupun *counter clockwise*.



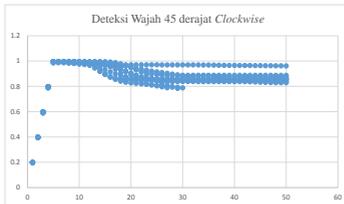
(a)



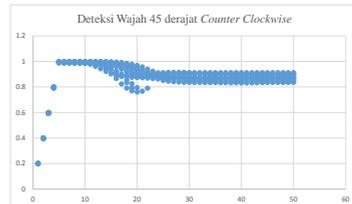
(b)



(c)



(d)



(e)

Gambar 4.4 Persebaran *Confidence* dari Metode *Template Matching*

4.4 Pengujian Kemampuan *Tracking* Wajah

Pada pengujian kemampuan *tracking* wajah, hal yang diamati adalah berapa *range* yang sudut yang harus dipenuhi agar *NANA* mampu mendeteksi wajah dari pengguna.

Tabel 4.3 Performansi *NANA* dalam *Tracking* Wajah

Sudut	Robot <i>Aldebaran NAO</i> melakukan <i>tracking</i>	Keterangan
-------	--	------------

Sudut	Robot <i>Aldebaran NAO</i> melakukan <i>tracking</i>	Keterangan
0°		<p><i>NANA</i> mampu melakukan <i>tracking</i> wajah dengan baik</p>

Sudut	Robot Aldebaran <i>NAO</i> melakukan <i>tracking</i>	Keterangan
<p>30° (<i>clockwise</i>)</p>		<p><i>NAO</i> mampu melakukan <i>tracking</i> wajah dengan baik</p>

Sudut	Robot <i>Aldebaran NAO</i> melakukan <i>tracking</i>	Keterangan
<p>30° (<i>co- unter cloc- kwi- se</i>)</p>		<p><i>NANA</i> mampu melakukan <i>tracking</i> wajah dengan baik</p>

Sudut	Robot Aldebaran <i>NAO</i> melakukan <i>tracking</i>	Keterangan
<p>45° (<i>clockwise</i>)</p>		<p><i>NAO</i> mampu melakukan <i>tracking</i> wajah dengan baik</p>

Sudut	Robot <i>Aldebaran NAO</i> melakukan <i>tracking</i>	Keterangan
<p>45° (<i>co- unter cloc- kwi- se</i>)</p>		<p><i>NANA</i> mampu melakukan <i>tracking</i> wajah dengan baik</p>

Sudut	Robot Aldebaran <i>NAO</i> melakukan <i>tracking</i>	Keterangan
<p>85° (clockwise)</p>		<p><i>NAO</i> mampu melakukan <i>tracking</i> wajah dengan baik</p>

Sudut	Robot <i>Aldebaran NAO</i> melakukan <i>tracking</i>	Keterangan
85° (<i>counter clockwise</i>)		<i>NANA</i> mampu melakukan <i>tracking</i> wajah dengan baik

Pada tabel 4.3 dapat diperhatikan bahwa *NANA* mampu melakukan *tracking* wajah manusia untuk sudut 0°, 30°, 45°, dan 85° *clockwise* maupun *counter clockwise*. Batas kemampuan *NANA* untuk melakukan *tracking* bergantung pada derajat kebebasan *joint HeadYaw* dari robot *Aldebaran NAO*, yakni 119.5° baik *clockwise* dan *counter clockwise*.

4.5 Kuesioner Keinteraktifan Sistem

Pada pengujian ini, dilakukan kuesioner mengenai desain sistem *NANA* untuk mengetahui tingkat keinteraktifan dari sistem. Kuesioner terdiri dari empat *point* utama, yaitu :

1. Saat berinteraksi, robot *NAO* memandang ke arah lawan bicara.
2. Gerak-gerik yang dilakukan robot *NAO* pada saat berinteraksi meru-

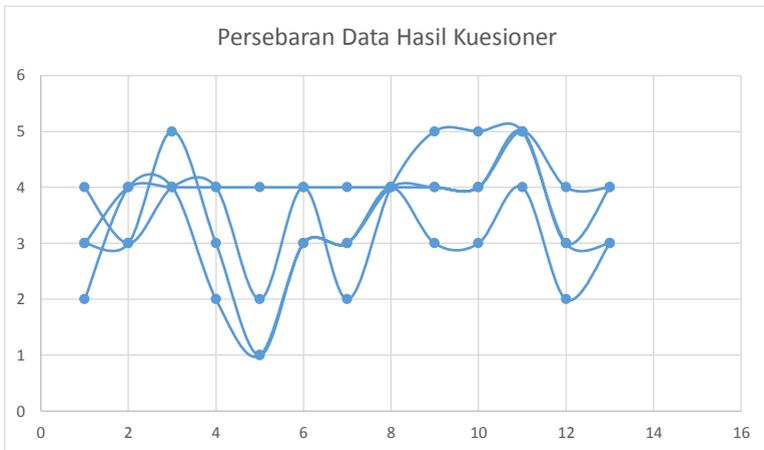
pakan gerakan yang wajar.

3. Saat berinteraksi, robot *NAO* mampu memberikan respon yang tepat, baik berupa suara maupun gerakan.
4. Waktu yang diperlukan robot *NAO* untuk memberikan respon masih berada dalam taraf kewajaran interaksi secara umum (interaksi sesama manusia).

Pada pengujian ini, partisipan diminta untuk memberikan tanggapan terkait empat *point* utama dari kuesioner dengan skala penilaian sebagai berikut :

- 1 : Sangat tidak setuju
- 2 : Tidak setuju
- 3 : Netral
- 4 : Setuju
- 5 : Sangat setuju

Dari 13 partisipan yang mengikuti kuesioner ini, didapatkan hasil berupa persebaran data tanggapan partisipan yang disajikan pada gambar 4.5.



Gambar 4.5 Persebaran Data Hasil Kuesioner

(Halaman ini sengaja dikosongkan)

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil perancangan dan pengujian seluruh sistem dalam tugas akhir ini, dapat diambil beberapa kesimpulan, antara lain sebagai berikut :

1. Penerapan deteksi wajah menggunakan *library OpenCV* dapat dikatakan cukup dengan kondisi wajah yang ingin dideteksi adalah menghadap ke arah sudut dalam *range* 0-30 derajat.
2. Penerapan *Speech to Text* menggunakan *Google Speech API* terbukti cukup baik dengan akurasi sebesar 80.45%.
3. Penerapan mesin dialog menggunakan *AIML* membuat sistem mampu merespon 47.58% ujaran manusia dengan tepat.

5.2 Saran

Dikarenakan *Speech to Text* bersifat *cloud*, artinya proses terjadi di *server* sehingga keberhasilan sistem sangat bergantung pada akses internet. Hal ini sebenarnya dapat diatasi dengan menggunakan sistem *Speech to Text* yang bersifat *offline*. Akan tetapi untuk membangun sistem *Speech to Text* yang bersifat *offline* diperlukan waktu dan *database* yang akan dibangun juga harus terdiri dari banyak *sample* suara (*speaker*) agar hasil yang diperoleh tingkat akurasi yang tinggi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] D. M. Wilkes, A. Alford, R. T. Pack, T. Rogers, R. A. P. Li, and K. Kawamura, "Toward socially intelligent service robots," *Applied Artificial Intelligence Journal*, pp. 729–766, 1998.
- [2] M. Salichs, R. Barber, A. Khamis, M. Malfaz, J. Gorostiza, R. Pacheco, R. Rivas, A. Corrales, E. Delgado, and D. Garcia, "Maggie: A robotic platform for human-robot social interaction," in *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, pp. 1–7, June 2006.
- [3] A. Robotics, *User Guide version 1.8.11*. francis: NAO Aldebaran Robotics, 2008.
- [4] "Lumen specification." [Online]. Available at: <http://lumen.lskk.ee.itb.ac.id/Products/LumenSpecification.aspx>. [Accessed: 2015-01-16].
- [5] J. Lyons, *Natural language and universal grammar*. Cambridge England New York: Cambridge University Press, 1991.
- [6] L. Shapiro, *Computer vision*. Upper Saddle River, NJ: Prentice Hall, 2001.
- [7] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.
- [8] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Narosa Publishing House, 1998.
- [9] "Viola-jones object detection framework - wikipedia, the free encyclopedia." [Online]. Available at: http://en.wikipedia.org/wiki/Viola-Jones_object_detection_framework. [Accessed: 2015-01-16].
- [10] H. Christiansen, ed., *SIGGRAPH '84: Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, (New York, NY, USA), ACM, 1984.

- [11] “Rosistem barcode - biometric education.” [Online]. Available at: <http://www.barcode.ro/tutorials/biometrics/voice.html>. [Accessed: 2015-01-16].
- [12] “Signal processing: Aliasing.” [Online]. Available at: http://www.xyobalancer.com/xyo-balancer-blog/signal_processing_aliasing. [Accessed: 2015-01-16].
- [13] “Speech to data - HowStuffWorks.” [Online]. Available at: <http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/speech-recognition1.htm>. [Accessed: 2015-01-16].
- [14] “3-3 human voice production (äžžèAšçŽĎçŤćçŤš).” [Online]. Available at: <http://mirlab.org/jang/books/audiosignalprocessing/humanVoiceProduction.asp>. [Accessed: 2015-01-16].
- [15] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [16] “how does it work? | acapela group - voice synthesis - text to speech.” [Online]. Available at: <http://www.acapela-group.com/voices/how-does-it-work/>. [Accessed: 2015-01-16].
- [17] N. Frijda, *The emotions*. Cambridge New York Paris: Cambridge University Press Editions de la Maison des Sciences de l’homme, 1986.
- [18] B. de Gelder and J. V. den Stock, “The bodily expressive action stimulus test (BEAST). construction and validation of a stimulus basis for measuring perception of whole body expression of emotions,” *Front. Psychology*, vol. 2, 2011.

RIWAYAT HIDUP PENULIS



Achmad Rizal Muttaqin lahir di Mojokerto pada tanggal 17 Juli 1992. Penulis menyelesaikan pendidikan dasarnya di SD Negeri Miji IV Mojokerto pada tahun 2004, kemudian menyelesaikan pendidikan menengah pertama di SMP Negeri 2 Kota Mojokerto pada tahun 2007, lalu menyelesaikan pendidikan tingkat menengah atas di SMA Negeri 1 Sooko Mojokerto pada tahun 2010. Pada jenjang pendidikan tinggi, penulis memilih untuk melanjutkan di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya (ITS) pada tahun 2010. Dalam perjalanan akademik di kampus ITS, penulis memilih untuk mengambil konsentrasi di bidang studi Teknik Komputer dan Telematika.

(Halaman ini sengaja dikosongkan)