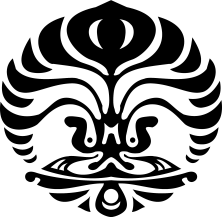
****

**UNIVERSITAS INDONESIA**

**SISTEM PENGENALAN GERAKAN TANGAN MANUSIA MENGGUNAKAN SENSOR *INERTIAL MEASUREMENT UNIT* DENGAN ALGORITMA *HIDDEN MARKOV MODEL***

**SKRIPSI**

**HANDISON JAYA**

**1106068913**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

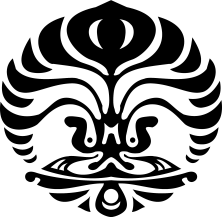
**DEPARTEMEN TEKNIK ELETKRO**

**TEKNIK KOMPUTER**

**DEPOK**

**DESEMBER**

**2014**

****

**UNIVERSITAS INDONESIA**

**SISTEM PENGENALAN GERAKAN TANGAN MANUSIA MENGGUNAKAN SENSOR *INERTIAL MEASUREMENT UNIT* DENGAN ALGORITMA *HIDDEN MARKOV MODEL***

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**HANDISON JAYA**

**1106068913**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

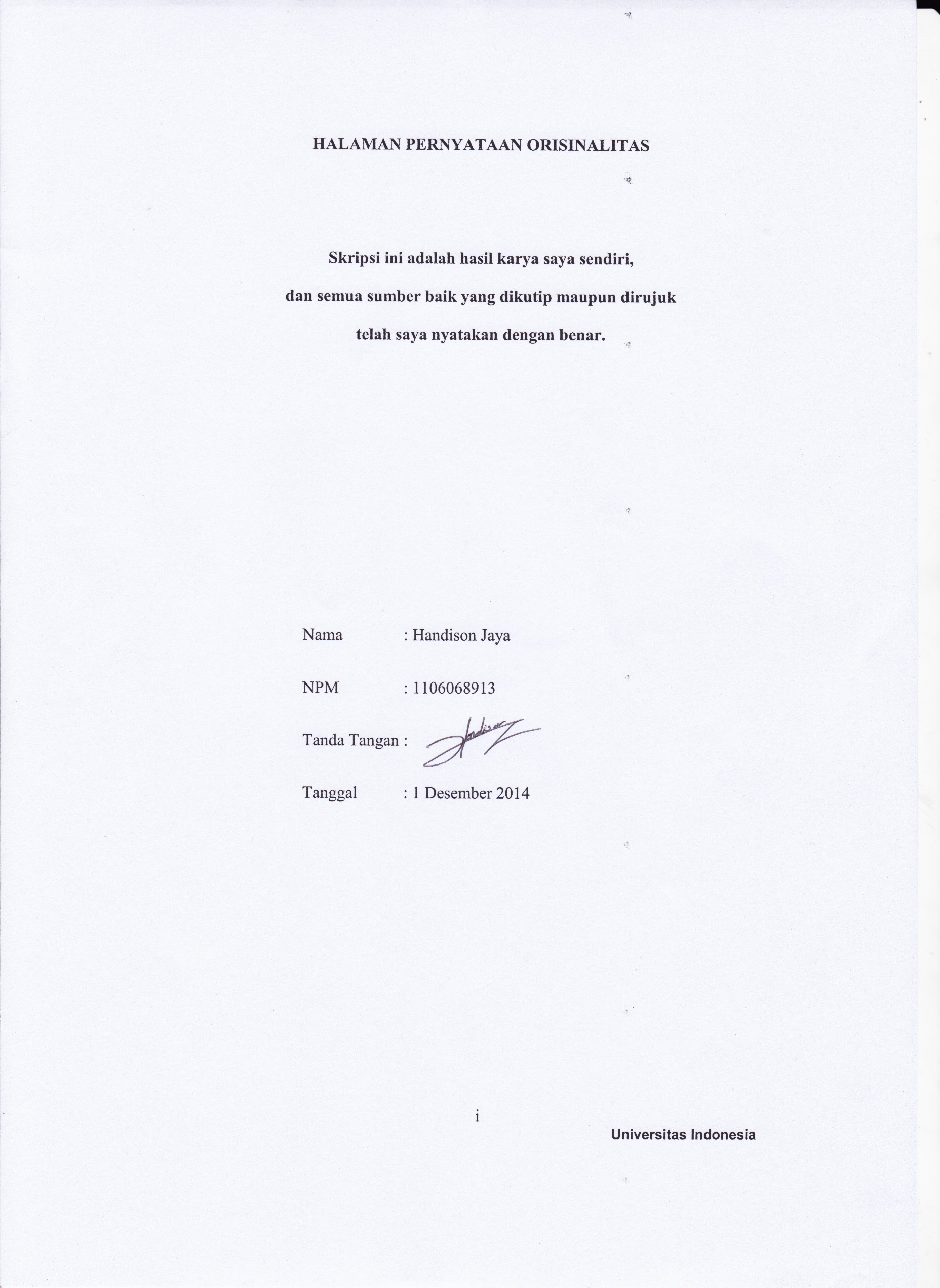
**DEPARTEMEN TEKNIK ELETKRO**

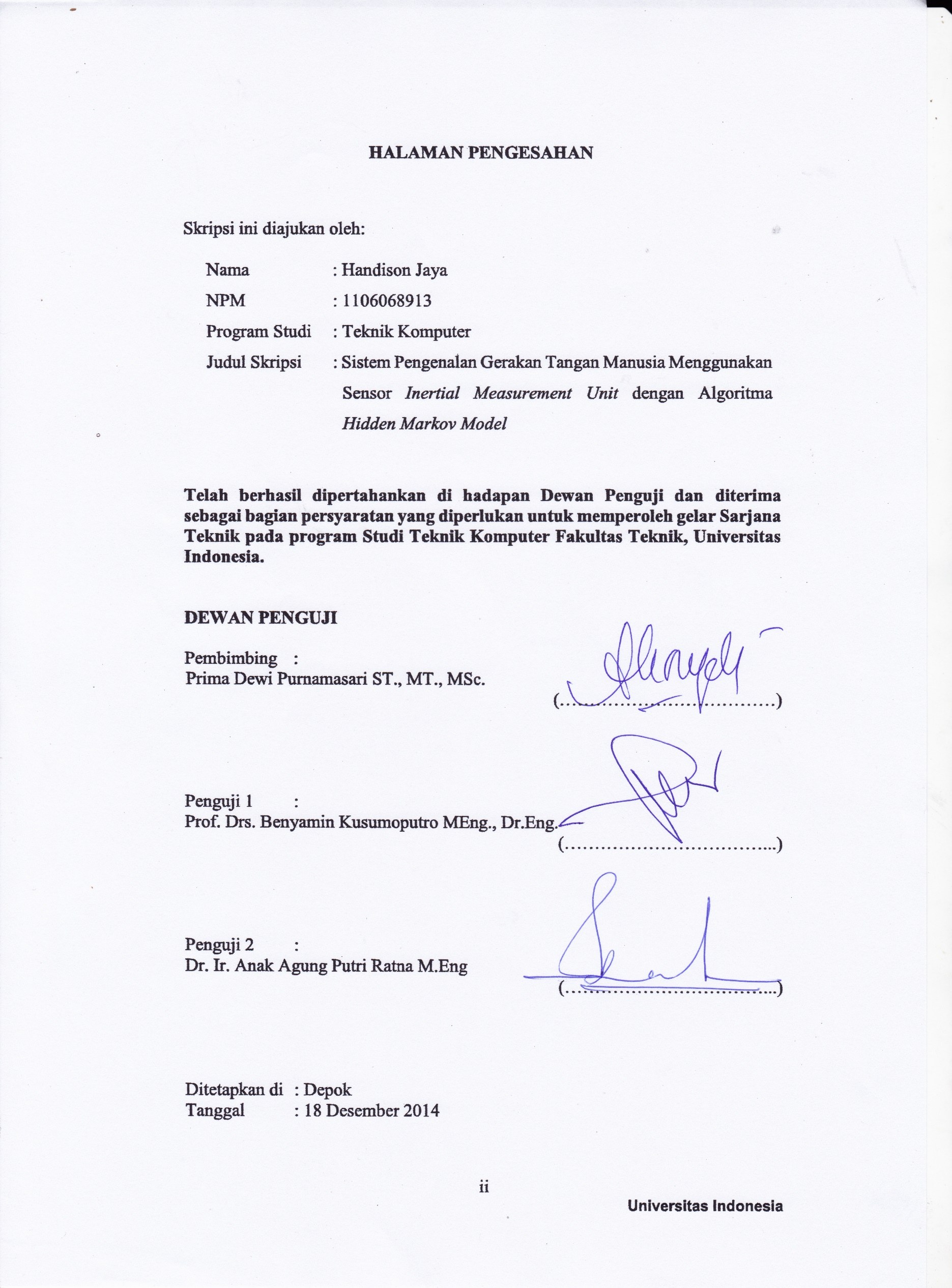
**TEKNIK KOMPUTER**

**DEPOK**

**DESEMBER**

**2014**





KATA PENGANTAR

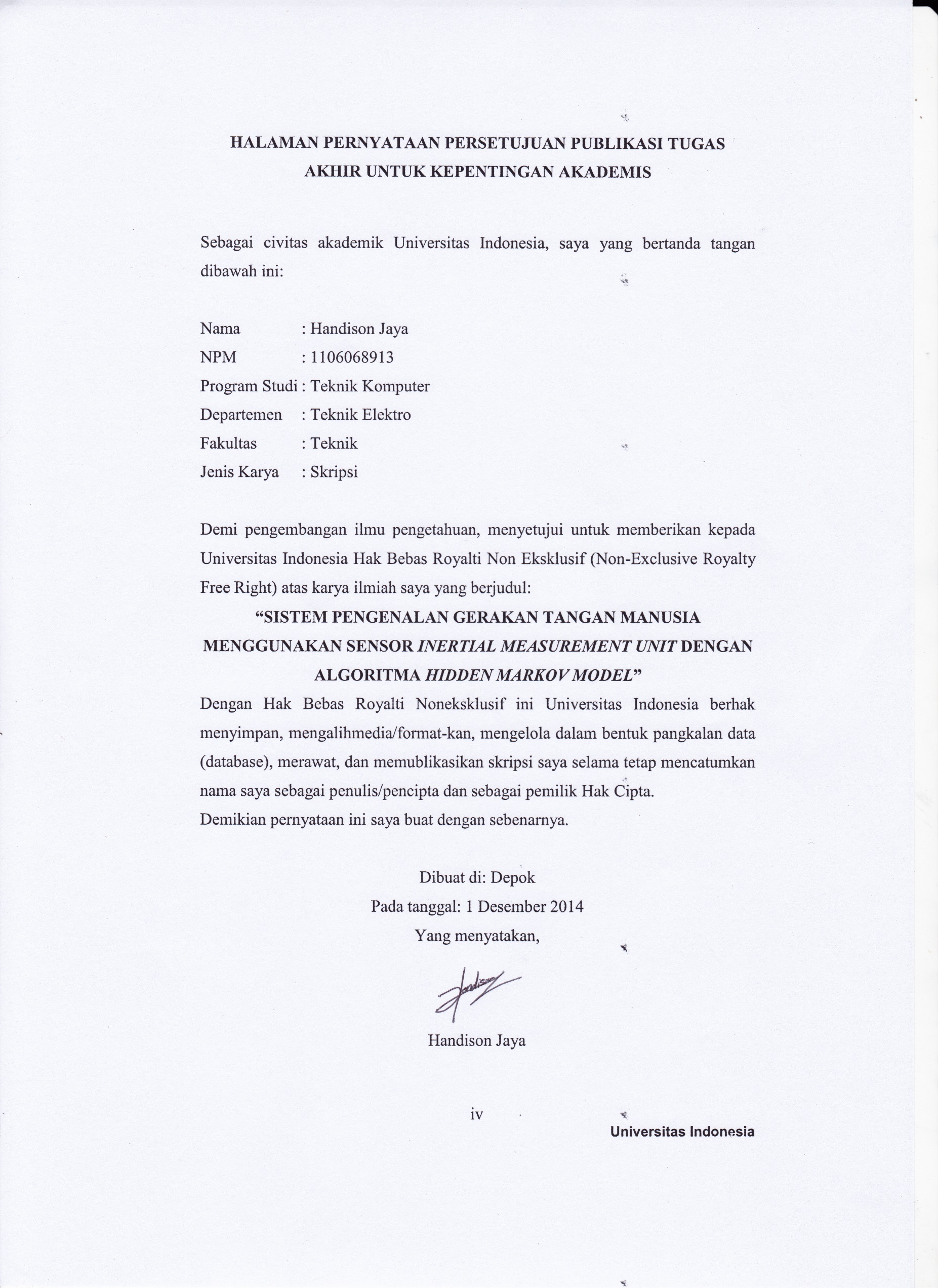
Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa sebab atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan baik. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mendapatkan gelar Sarjana Teknik Program Studi Teknik Komputer Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Penulis menyadari bahwa skripsi ini tidak dapat diselesaikan tanpa bantuan dari banyak pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Ibu Prima Dewi Purnamasari ST., MT., MSc. selaku pembimbing skripsi yang telah memberikan pengarahan, koreksi, dukungan, waktu, dan akomodasi alat selama penulis mengerjakan penelitian ini.
2. Orang tua dan keluarga penulis yang telah memberikan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan penelitian ini.
3. Teman riset seperjuangan Tryan Aditya Putra dalam membangun sistem pengenalan gerakan, yang banyak memberikan bantuan dan masukkan dalam penelitian ini.
4. Tim Robotika UI Annisaa Primadini, Ismi, Alif, Lintang, Zaka atas dukungan dan bantuannya saat pengerjaan penelitian ini.
5. Rekan-rekan dari laboratorium Digital Edo, Dwi, Antoni, Puput, Ibam.
6. Teman-teman Teknik Komputer 2011 Emily, Yessy, Zhafir, Kautsar, Yudha, Dinar, Sean, dan Keluarga Departemen Teknik Elektro lain.

Akhir kata, semoga Tuhan berkenan membalas kebaikan dari semua pihak yang telah berbaik hati membantu penulis dan semoga penelitian ini dapat bermanfaat bagi pengembangan teknologi dan ilmu pengetahuan.

Depok, Desember 2014

Handison Jaya

ABSTRAK

|  |  |  |
| --- | --- | --- |
| Nama | : | Handison Jaya |
| Program Studi | : | Teknik Komputer |
| Judul | : | Sistem Pengenalan Gerakan Tangan Manusia Menggunakan Sensor *Inertial Measurement Unit* Dengan Algoritma *Hidden Markov Model* |

Sistem pengenalan gerakan manusia merupakan teknologi yang penting karena dapat mempermudah pekerjaan manusia dalam berbagai aspek dan membantu manusia yang memiliki keterbatasan. Adapun gerakan yang bisa dikenali adalah gerakan manusia dimana tangan lurus dan berayun, gerakan manusia dimana tangan ditekuk dan berayun, gerakan pergelangan tangan memutar kekanan dan kekiri, serta gerakan tangan ditarik mendekati tubuh dan gerakan tangan mendorong menjauhi tubuh. Salah satu algoritma dalam bidang *Artificial Intelligence* yang bisa digunakan adalah *Hidden Markov Model* (HMM). HMM sendiri merupakan suatu permodelan statistika yang dimana sistem yang dimodelkan diasumsikan merupakan proses Markov yang memiliki *state/*keadaan yang tersembunyi (*hidden*). Pada penelitian ini digunakan sensor *Inertial Measurement Unit* sebagai pendeteksi gerakan manusia. Komunikasi antara sensor dengan komputer dilakukan secara nirkabel menggunakan XBee. Sistem yang dibuat dapat mengenali enam gerakan manusia tadi secara *real time*. Hasil pengujian menunjukkan bahwa HMM dapat mengenali gerakan manusia dengan tingkat akurasi sebesar 88% dalam waktu 0.004 detik.

Kata kunci: *Hidden Markov Model*, Proses Markov, *Inertial Measurement Unit*

ABSTRACT

|  |  |  |
| --- | --- | --- |
| Name | : | Handison Jaya |
| Major | : | Computer Engineering |
| Title | : | Hand Motion Recognition Using Inertial Measurement Unit Sensor with *Hidden* Markov Model Algorithm |

Human motion recognition is an important technology to be developed, as it can facilitate human work and also help people with disabilities. As for motion, sytem could recognize six motion, which is human arms straight and swinging, human hand bent and swinging, hand twisting left, hand twisting right, hand push *forward*, and hand pull inward. One of Artificial Intelligence algorithm that can be used is *Hidden* Markov Model (HMM). HMM itself is a statistical model where the system which being modelled was assumed as Markov process that has *hidden* *state*. This research used Inertial Measurement Unit sensor as human motion detector. Communication between sensor and computer was conducted wirelessly with XBee. This system can recognize six motion real time. From the results show that *Hidden* Markov Model can recognize human motion with an accuracy rate of 88% within 0.004 seconds.

Keywords: *Hidden* Markov Model, Markov Process, Inertial Measurement Unit.

DAFTAR ISI

[HALAMAN PERNYATAAN ORISINALITAS Error! Bookmark not defined.](#_Toc407710883)

[HALAMAN PENGESAHAN Error! Bookmark not defined.](#_Toc407710884)

[KATA PENGANTAR iii](#_Toc407710885)

[HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS Error! Bookmark not defined.](#_Toc407710886)

[ABSTRAK iv](#_Toc407710887)

[ABSTRACT vi](#_Toc407710888)

[DAFTAR ISI vii](#_Toc407710889)

[DAFTAR GAMBAR ix](#_Toc407710890)

[DAFTAR TABEL xi](#_Toc407710891)

[BAB 1 PENDAHULUAN 1](#_Toc407710892)

[1.1. Latar Belakang 1](#_Toc407710893)

[1.2. Tujuan Penelitian 2](#_Toc407710894)

[1.3. Batasan Masalah 2](#_Toc407710895)

[1.4. Metode Penelitian 3](#_Toc407710896)

[1.5. Sistematika Penulisan 4](#_Toc407710897)

[BAB 2 *HIDDEN* MARKOV MODEL DALAM SISTEM PENGENALAN GERAKAN MANUSIA 5](#_Toc407710898)

[2.1. Pengenalan dengan *Artificial Neural Network* 5](#_Toc407710899)

[2.1.1. Pengertian *Artificial Neural Network* 5](#_Toc407710900)

[2.1.2. Jenis *Artificial Neural Network* 7](#_Toc407710901)

[2.2. Pengenalan dengan *Hidden* Markov Model 9](#_Toc407710902)

[2.2.1. Markov Model 9](#_Toc407710903)

[2.2.2. *Hidden* Markov Model 11](#_Toc407710904)

[2.3. Hardware 20](#_Toc407710905)

[2.3.1. IMU 20](#_Toc407710906)

[2.3.2. Zigbee dan Xbee *Wireless* 23](#_Toc407710907)

[2.4. MATLAB [11] 26](#_Toc407710908)

[2.5. Perbandingan dengan Penelitian Lainnya 27](#_Toc407710909)

[BAB 3 PERANCANGAN SISTEM UNTUK PENGENALAN GERAKAN MANUSIA 29](#_Toc407710910)

[3.1. System Requirement 29](#_Toc407710911)

[3.2. System and *Software* Design 31](#_Toc407710912)

[3.2.1. Desain Perangkat Keras (*Hardware*) 33](#_Toc407710913)

[3.2.2. Teknik Pengambilan Data Sensor 34](#_Toc407710914)

[3.2.3. Algoritma Perangkat Lunak (*Software*) 35](#_Toc407710915)

[BAB 4 IMPLEMENTASI DAN PENGUJIAN 45](#_Toc407710916)

[4.1. Implementasi Sistem 45](#_Toc407710917)

[4.2. Pengujian Akurasi Pengenalan Gerakan Manusia dengan Menggunakan *Hidden* Markov Model 48](#_Toc407710918)

[4.3. Pengujian Akurasi Pengenalan Gerakan Manusia dengan Menggunakan *Artificial Neural Network* dan *Hidden* Markov Model 49](#_Toc407710919)

[4.4. Perbandingan Waktu Pelatihan dan Waktu Pengujian 50](#_Toc407710920)

[BAB 5 KESIMPULAN 52](#_Toc407710921)

[DAFTAR REFERENSI 53](#_Toc407710922)

DAFTAR GAMBAR

[Gambar 1.1 Gerakan tangan berayun ketika (a) ditekuk (b) lurus 3](#_Toc407710923)

[Gambar 2.1 Struktur *Artificial Neural Network* 6](#_Toc407710924)

[Gambar 2.2 Fungsi Aktivasi 7](#_Toc407710925)

[Gambar 2.3 *Markov Chain* yang Menggambarkan Perubahan Cuaca [1] 10](#_Toc407710926)

[Gambar 2.4 Representasi Parameter HMM [1] 12](#_Toc407710927)

[Gambar 2.5 Contoh HMM [3] 14](#_Toc407710928)

[Gambar 2.6 Sensor Razor IMU [6] 20](#_Toc407710929)

[Gambar 2.7 Arah Orientasi dan Polaritas Rotasi [7] 21](#_Toc407710930)

[Gambar 2.8 Blok Diagram Fungsional ITG-3200 [7] 21](#_Toc407710931)

[Gambar 2.9 Orientasi Sensitivitas ADXL345 [8] 22](#_Toc407710932)

[Gambar 2.10 Blok Diagram Fungsional ADXL345 [8] 22](#_Toc407710933)

[Gambar 2.11 Periode Pengambilan Data Magnetometer [9] 23](#_Toc407710934)

[Gambar 2.12 XBee Chip Antenna Series 2 [10] 24](#_Toc407710935)

[Gambar 2.13 *Interface* X-CTU [10] 25](#_Toc407710936)

[Gambar 3.1 *Use Case* Diagram 30](#_Toc407710937)

[Gambar 3.2 Blok Diagram Sistem Pengenalan Gerakan Manusia 31](#_Toc407710938)

[Gambar 3.3 *Sequence* Diagram 32](#_Toc407710939)

[Gambar 3.4 Blok *Hardware* Sistem 33](#_Toc407710940)

[Gambar 3.5 Teknik *Sliding Window* dalam Pengambilan Data dari Sensor 34](#_Toc407710941)

[Gambar 3.6 *Activity* Diagram Ketika Pelatihan dengan HMM 36](#_Toc407710942)

[Gambar 3.7 Fase Pelatihan dengan HMM 37](#_Toc407710943)

[Gambar 3.8 Pseudocode Pelatihan dengan HMM 38](#_Toc407710944)

[Gambar 3.9 *Activity* Diagram Ketika Pengenalan dengan HMM 39](#_Toc407710945)

[Gambar 3.10 Fase Pengenalan dengan HMM 40](#_Toc407710946)

[Gambar 3.11 *Activity* Diagram Ketika Pelatihan dengan ANN dan HMM 41](#_Toc407710947)

[Gambar 3.12 Fase Pelatihan dengan ANN dan HMM 42](#_Toc407710948)

[Gambar 3.13 Pseudocode Pelatihan dengan ANN dan HMM 43](#_Toc407710949)

[Gambar 3.14 *Activity* Diagram Ketika Pengenalan dengan ANN dan HMM 44](#_Toc407710950)

[Gambar 4.1 Alat Pengenalan Gerakan 46](#_Toc407710951)

[Gambar 4.2 Pengambilan Data 47](#_Toc407710952)

[Gambar 4.3 Tampilan Hasil pada *Command Window* MATLAB 47](#_Toc407710953)

DAFTAR TABEL

[Tabel 3.1 Perbandingan dengan Penelitian Lain 28](#_Toc407710954)

[Tabel 4.1 Akurasi Sistem Menggunakan HMM 48](#_Toc407710955)

[Tabel 4.2 Akurasi Sistem Menggunakan ANN dan HMM 49](#_Toc407710956)

[Tabel 4.3 Waktu Pelatihan dan Waktu Pengujian Sistem 50](#_Toc407710957)

1. PENDAHULUrrrrrAN

Pada bab ini akan dijabarkan mengenai latar belakang, tujuan, batasan masalah, metode penulisan, serta sistematika penulisan dari penelitian ini.

1. Latar Belakang

Pengenalan terhadap gerakan manusia merupakan salah satu bidang riset yang sedang banyak dikembangkan. Hal ini karena pengenalan terhadap gerakan ini dapat dimanfaatkan pada berbagai bidang untuk memudahkan pekerjaan manusia, serta praktis dalam penggunaannya. Salah satunya adalah teknologi *smart home* dimana pengguna dapat mengendalikan rumahnya hanya dengan gerakan tangan. Hal ini sangat berguna ketika pengguna mengharapkan suatu rumah yang praktis, ataupun untuk pengguna yang memiliki keterbatasan dalam beberapa hal. Dimana dengan adanya *smart home* ini akan sangat membantu pengguna dalam melaksanakan kegiatan sehari-hari.

Ada beberapa teknik yang bisa digunakan untuk mengenali gerakan manusia, diantaranya dengan menggunakan suatu sensor yang dapat mengukur percepatan sesaat dari gerakan manusia atau dengan mengunakan teknik *image processing,* seperti teknik videografi atau optoelektronik*.* Untuk teknik *image processing*, banyak digunakan alat berupa kamera CCTV atau kamera *webcam* serta *processor* yang memiliki kemampuan komputasi yang kuat. Sedangkan jika meggunakan sensor percepatan sesaat bisa menggunakan sensor IMU (*Inertial Measurement Unit*).

Kelebihan menggunakan sensor percepatan untuk mengenali gerakan manusia adalah biaya yang cukup murah jika dibandingkan dengan teknik lain. Selain itu, proses komputasi yang dibutuhkan juga relatif lebih ringan, sehingga waktu komputasi akan lebih cepat dibanding dengan teknik *image processing*. Akan tetapi mempunyai kelemahan yakni rentan akan *noise* terhadap data yang diterima dari sensor.

Algoritma dalam mengenali gerakan manusiajuga menjadi hal penting dalam membangun suatu sistem yang bisa mengenali gerakan manusia. Adapun masalah yang dihadapi dalam memilih algoritma adalah pada proses pembacaaan data dari sensor yang akan mengenali gerakan rentan terhadap *noise,* sehingga diperlukan suatu algoritma yang adaptif, yang bisa “belajar” dengan sendirinya dan mentoleransi error yang dihasilkan. Salah satu algoritma yang memenuhi kriteria tersebut dan sering digunakan adalah *Artificial Neural Network* (ANN) yang kemudian dikenali dan diklasifikasi lagi dengan menggunakan algoritma *Hidden Markov Model* (HMM) dan *Support Vector Machine* (SVM).

1. Tujuan Penelitian

* Membuat alat/sistem yang dapat mengenali gerakan manusia.
* Menggunakan HMM untuk mengenali gerakan manusia.
* Menganalisis tingkat akurasi alat/sistem yang telah dibuat dalam mengenali gerakan manusia.

1. Batasan Masalah

Sistem dirancang untuk mengenali 6 jenis gerakan, antara lain gerakan tangan manusia lurus dan berayun, gerakan tangan manusia ditekuk dan berayun, seperti Gambar 1.1, pergelangan tangan diputar ke arah kiri dan ke arah kanan, gerakan tangan mendorong ke depan menjauhi badan serta gerakan tangan menarik ke dalam mendekati tubuh. Gerakan-gerakan dapat dikenali dengan menggunakan sensor IMU (*Inertial Measurement Unit*) yang memiliki 9DOF (3-*axis accelerrometer,* 3-*axis gyroscope*, 3-*axis magnetometer*) untuk mengenali vektor gerakan manusia yang berbeda dalam suatu ruang tiga dimensi. Akan tetapi pada penelitian ini DOF yang digunakan hanya 3, yakni 3-*axis gyroscope,* hal ini karena pengaruh *magnetometer* dan *accelerometer* masih belum menjadi pertimbangan utama dalam mengenali gerakan manusia. Pada penelitian ini, CPU utama yang digunakan untuk melakukan pemrosesan dan pengenalan gerakan manusia adalahMATLAB. Adapun proses komunikasi antara sensor IMU dengan *user* dalam melakukan proses *debugging* dilakukan secara nirkabel dengan menggunakan Xbee. Dimana terdapat 2 Xbee yang akan digunakan. Xbee pada sensor akan berperan sebagai *transmitter* data, sedangkan Xbee satunya lagi akan sebagai *receiver* yang terhubung pada komputer.

a b



Gambar 1.1 Gerakan tangan berayun ketika (a) ditekuk (b) lurus

1. Metode Penelitian

Metode penelitian mengadopsi *software engineering cycle*, yaitu sebagai berikut:

1. Studi Literatur: mencari dan membaca buku, jurnal, *paper,* ataupun *web* yang berkaitan dengan topik penelitian.
2. *System Requirement*: menentukan spesifikasi sistem dengan melakukan studi literatur dan pengamatan pada aplikasi yang sudah ada.
3. *System and Software Design*: merancang simulasi algoritma *Hidden Markov Model*.
4. *Implementation*: mengimplementasikan rancangan simulasi, yaitu dengan membuat alat yang diintegrasikan dengan sensor.
5. *Testing*: melakukan pengujian terhadap performa alat.
6. Analisis: analisis hasil pengujian sistem.
7. Sistematika Penulisan

*Bab 1 Pendahuluan*

Pada bab ini akan dijabarkan mengenai latar belakang, tujuan, batasan masalah, metode penelitian, serta sistematika penulisan.

*Bab 2 Hidden Markov Model dalam Sistem Pengenalan Gerakan Manusia*

Pada bab ini akan dijelaskan dasar teori dan algoritma yang berkaitan dengan *Hidden Markov Model* dalam fase pembelajaran dan pelatihan sistem untuk mengenali gerakan manusia. Selain itu juga akan dijelaskan dasar teori mengenai perangkat keras yang digunakan oleh sistem.

*Bab 3 Perancangan Sistem untuk Pengenalan Gerakan Manusia*

Pada bab ini akan dijelaskan mengenai perancangan sistem dan algoritma dalam menunjang pengenalan gerakan manusia. Perancangan sistem mencakup *system requirement* dan *system software* desain.

*Bab 4 Implementasi dan Pengujian*

Pada bab ini akan dijelaskan mengenai implementasi sistem pengenalan gerakan manusia, serta pengujian dan analisis fungsionalitas sistem.

*Bab 5 Kesimpulan*

Pada bab ini akan disampaikan kesimpulan dari implementasi, pengujian, serta analisis sistem pengenalan gerak manusia.

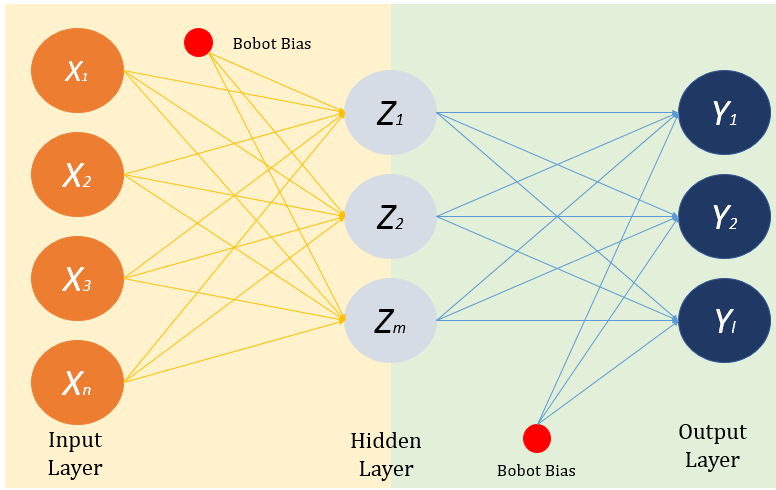
1. *HIDDEN* MARKOV MODEL DALAM SISTEM PENGENALAN GERAKAN MANUSIA

Pada bab ini akan dibahas mengenai teori-teori yang berkaitan dengan topik penelitian. Adapun yang akan dibahas yaitu mengenai pengenalan dengan menggunakan *Hidden Markov Model* dan *hardware-hardware* yang akan digunakan, seperti IMU dan Xbee *wireless*, serta *software* yang digunakan, yakni MATLAB.

1. Pengenalan dengan *Artificial Neural Network*
2. Pengertian *Artificial Neural Network*

*Artificial Neural Network* merupakan metode komputasi yang dimodelkan berdasarkan proses pengolahan informasi biologis pada unit sistem saraf manusia yang disebut dengan neuron. Sebelum informasi yang didapatkan dari neuron reseptor di kirimkan ke neuron lainnya, informasi akan dikalikan dengan suatu nilai bobot. Hal ini akan terjadi pada setiap proses pengiriman informasi pada neuron yang terdapat pada sistem saraf manusia. Proses penyampaian informasi dari reseptor hingga menuju pusat saraf manusia terdiri atas neuron-neuron yang terlewati dimana masing-masing informasi tersebut akan dipengaruhi oleh nilai bobot yang berbeda hingga jenis informasi yang diterima di pusat saraf dapat diartikan sebagai jenis rangsangan tertentu. Bobot dari proses penyampaian informasi ini disebut dengan pola pembelajaran.

Pembelajaran dengan *Artificial Neural Network* termasuk pembelajaran melalui contoh. Hal ini dikarenakan pembelajaran dengan *Artificial Neural Network* akan berusaha mengenali pola yang merepresentasikan sistem atau hubungan antara rangsangan yang diterima dan informasi mengenai respons terhadap rangsangan. Respons yang dihasilkan belum tentu sesuai dengan yang diinginkan, sehingga menimbulkan nilai kesalahan. Lalu sistem pusat saraf akan menyesuaikan pola pembelajarannya sehingga akan menghasilkan respons yang tepat untuk rangsangan tersebut. Sistem juga harus dapat menghasilkan respons yang berbeda dan untuk setiap rangsangan yang berbeda. Setiap kali sistem menghasilkan respons disebut dengan siklus pembelajaran. Pembelajaran akan berhenti jika pada siklus pembelajaran terakhir menghasilkan respons yang sesuai, dimana siklus ini akan mempunyai nilai kesalahan yang kecil.

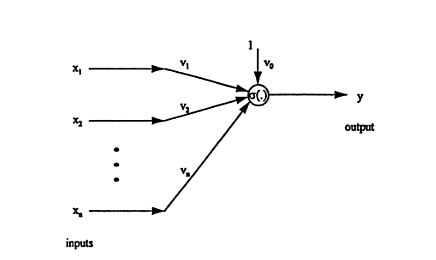


Gambar 2.1 Struktur *Artificial Neural Network*

Dari proses penyampaian informasi pada sistem saraf manusia dapat diturunkan suatu model matematis yang dapat digunakan dalam komputasi data, seperti pada Gambar 2.1. Dengan memisalkan informasi data input yang mempunyai dimensi sebanyak *n* pada waktu *t* sebagai *x(t*), nilai akan dikalikan dengan nilai bobot dan bias untuk mendapatkan nilai keluaran. Nilai bobot pada neuron yang berkorespondensi untuk setiap n dimensi data input direpresentasikan dengan dan bias . Sehingga, output *y(t)* dapat dirumuskan sebagai berikut:

|  |  |
| --- | --- |
|  | (2.1) |

Dengan merupakan fungsi penjumlahan yang disebut dengan fungsi aktivasi, yang merupakan representasi sifat neuron. Fungsi aktivasi akan berbeda untuk setiap metode yang dipakai dalam *Artificial Neural Network* sesuai dengan karakteristik masing-masing metode yang digunakan. Fungsi aktivasi, seperti yang terlihat pada Gambar 2.2, yang sering digunakan adalah fungsi aktivasi sigmoid dan fungsi radial.



Gambar 2.2 Fungsi Aktivasi

1. Jenis *Artificial Neural Network*

*Artificial Neural Network* dapat dibedakan ke dalam 2 kategori menurut mekanisme pembelajarannya, yaitu:

* *Supervised* *Learning*

Metode pembelajaran ini meggunakan nilai target sebagai pengontrol pembelajaran. Tujuan pembelajaran ini adalah untuk mencari bobot optimal yang akan diperoleh ketika nilai output dari suatu nilai input sistem akan mendekati nilai target yang diinginkan. Dengan menggunakan metode pembelajaran ini didapatkan pola pengenalan untuk setiap pasangan data input dan data output. Pola pengenalan diperoleh dengan menyesuaikan nilai bobot pembelajaran dan menghitung persen kesalahan dari output pembelajaran dari target yang diinginkan untuk setiap data input. Bobot akan terus disesuaikan hingga persen kesalahan akan konvergen. Salah satu metode pembelajaran yang menerapkan prinsip ini adalah metode *Backpropagation*.

Metode *Backpropagation* adalah metode pembelajaran yang menggunakan perbedaan persen kesalahan dengan masing-masing bobot pada lapisan pembelajaran sebagai nilai perbaikan bobot. *Backpropagation* terdiri atas dua lapisan pembelajaran, dan dua tahapan pembelajaran, tahapan *feedforward* dan *backward*. Pada tahapan *feedforward* terjadi perhitungan keluaran dari masing masing lapisan pembelajaran, dengan keluaran sistem pembelajaran terdapat pada output layer.

Keluaran pada *hidden* layer dapat dirumuskan dengan:

|  |  |
| --- | --- |
|  | (2.2) |

Keluaran pada output layer dapat dirumuskan dengan:

|  |  |
| --- | --- |
|  | (2.3) |

Sebelum melakukan tahapan *backward*, nilai persen kesalahan akan dihitung berdasarkan perbedaan nilai target dan nilai keluaran pembelajaran, didefinisikan dengan:

|  |  |
| --- | --- |
|  | (2.4) |

Jika tahapan *feedforward* bertujuan untuk mencari nilai keluaran sistem dan persen kesalahan, pembelajaran tahapan *backward* bertujuan untuk menyesuaikan bobot lapisan pembelajaran berdasarkan persen kesalahan pembelajaran terhadap bobot pembelajaran. Perbaikan bobot dapat dirumuskan sebagai turunan nilai persen kesalahan (error) terhadap bobotnya, sehingga:

|  |  |
| --- | --- |
|  | (2.5) |

Dengan adalah nilai momentum dan adalah nilai konstanta pembelajaran, yang akan mempengaruhi waktu yang diperlukan untuk mencapai kondisi konvergen. Semakin besar nilai alpha dan miu, maka waktu pembelajaran akan lebih singkat. Akan tetapi ketelitian yang dihasilkan berkurang.

* *Unsupervised* *Learning*

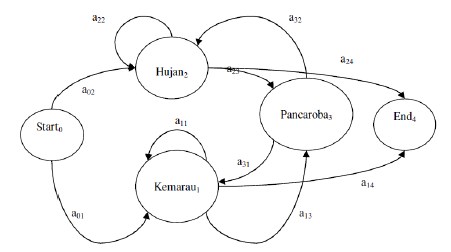
Metode pembelajaran *unsupervised* tidak menggunakan target sebagai sinyal pembelajaran, sehingga nilai bobot yang merepresentasikan pola pembelajaran berdasarkan aturan-aturan tertentu pada pembelajaran tersebut. Setiap data input akan menyesuaikan diri dengan setiap representasi kelompok pola atau setiap data input tersebut. *Unsupervised* *learning* biasanya digunakan dalam mengelompokkan informasi data input berdasarkan kemiripan karakteristik setiap data input, ke dalam kelompok tertentu. Terdapat berbagai mekanisme pembelajaran *unsupervised*, salah satunya adalah HMM, yang akan dijelaskan pada subbab berikutnya.

1. Pengenalan dengan *Hidden* Markov Model
2. Markov Model

*Markov Model* biasa disebut sebagai *Markov Chain* atau *Markov Process,* merupakan sub-ilmu dari *data mining* dan *soft computing*. Model ini merupakan bagian dari proses stokastik yang memiliki properti Markov. Suatu proses stokastik dikatakan memiliki properti Markov apabila pada proses tersebut diberikan suatu inputan yang berupa keadaan saat ini, maka keadaan pada saat yang akan datang dapat diprediksi dengan hanya melihat keadaan pada masa sekarang ataupun masa lampau. Adapun keadaan saat ini dan keadaan yang akan datang bersifat independen terhadap keadaan pada masa lampau. Artinya, deskripsi kondisi saat ini menangkap semua informasi yang mempengaruhi evolusi dari suatu sistem dimasa yang akan datang. Dengan kata lain, kondisi masa depan dicapai dengan menggunakan probabilitas bukan dengan determinitas yang ada [1].

*Markov Model* cocok digunakan untuk data yang bersifat temporal sekuensial. Seperti pada sinyal suara ataupun sinyal digital, suatu urutan data pada DNA, ataupun urutan unsur-unsur kimia yang membangun suatu molekul yang kompleks. Oleh karena itu, *Markov Model* banyak dimanfaatkan pada bidang genomik dan bioinformatika [1].

Model ini merupakan bagian dari *finite state. Finite state* sendiri adalah kumpulan *state* yang transisi antar *state*-nya dilakukan berdasarkan masukkan observasi. *Markov Model* dapat digambarkan sebagai suatu graf yang memiliki arah dan bobot, dengan setiap *node-*nya (simpul) merepresentasikan suatu *state* (keadaan) dan setiap *edge* (busur) merupakan nilai probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua busur yang keluar dari sebuah simpul adalah satu. Gambar dibawah memperlihatkan contoh *Markov Chain* yang menggambarkan kemungkinan kondisi cuaca [1].



Gambar 2.3 *Markov Chain* yang Menggambarkan Perubahan Cuaca [1]

Pada Gambar 2.3, aij menyatakan probabilitas transisi dari *state* i ke *state* j. Misalkan, dari simpul start0 keluar dua kemungkinan, yakni a02 dan a01, sehingga jumlah probabilistik a01+a02 adalah satu. Hal ini juga berlaku bagi simpul-simpul lainnya. *Markov Chain* bermanfaat untuk menghitung probabilitas suatu kejadian yang teramati (suatu urutan kejadiann). Secara matematis, dapat dirumuskan sebagai berikut:

|  |  |
| --- | --- |
|  | (2.6) |

Dimana adalah kondisi saat ini, dan adalah kondisi pada waktu tertentu yang berhubungan dengan . Sedangkan adalah kondisi sebelum . Kemudian dapat diasumsikan bahwa persamaan sebelah kanan bersifat invariant, yaitu dihipotesiskan dalam keseluruhan sistem, transisi diantara keadaan tertentu tetap sama dalam hubungan probabilistiknya.

Berdasarkan asumsi tersebut, dapat dibuat suatu set keadaan probabilistik aij diantara dua keadaan Si dan Sj:

|  |  |
| --- | --- |
|  | (2.7) |

Selain itu juga berlaku batasan, yakni:

|  |  |
| --- | --- |
|  | (2.8) |

Dari penjelasan diatas, dapat disimpulkan bahwa *Markov Chain* bermanfaat untuk menghitung probabilitas urutan kejadian yang dapat diamati. Akan tetapi, terkadang ada urutan kejadian yang ingin diketahui tetapi tidak dapat diamati. Untuk menyelesaikan kasus tersebut, dikembangkanlah model baru yang memodelkan kejadian yang tersembunyi, *Hidden Markov Model*.

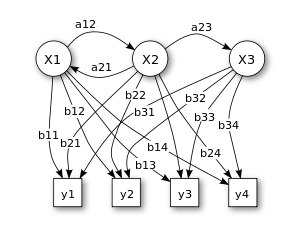
1. *Hidden* Markov Model

*Hidden Markov Model* atau yang sering disingkat HMM, merupakan suatu permodelan statistika yang dimana sistem yang dimodelkan diasumsikan merupakan proses Markov yang memiliki *state/*keadaan yang tersembunyi (*hidden*). Model ini dikembangkan pertama kali oleh Andreyevich Markov, seorang ilmuwan Rusia pada awal abad 20. Pada awalnya, model ini merupakan murni model teoritis namun mulai dikembangkan oleh kalangan akademisi dan engineer dalam paper-paper internasional. Aplikasi awal HMM adalah untuk proses pengenalan suara (*speech recognition*) yang dimulai pada pertengahan 1970. Pada pertengahan 1980, HMM mulai digunakan untuk berbagai macam hal lain, seperti untuk analisis rangkaian biologis terutama pada DNA, selain itu juga menjadi salah satu model yang penting dalam bidang bioinformatika [2].

Suatu HMM dapat dianggap sebagai jaringan Bayesian dinamis yang sederhana, perbedaannya dengan Markov Model biasa adalah adanya keadaan yang tidak dapat terlihat secara langsung tetapi output yang bergantung pada keadaan tersebut terlihat. Setiap kondisi memiliki distribusi kemungkinan disetiap output yang mungkin. Oleh karena itu, urutan langkah yang dibuat oleh HMM memberikan suatu informasi tentang urutan dari keadaan. Kata “*hidden*” pada HMM merujuk kepada kondisi langkah yang dilewati oleh model, bukan kepada parameter dari model tersebut. Walaupun parameter model diketahui, model tersebut tetap bisa diasumsikan tersembunyi/*hidden* [3].

HMM adalah variasi dari *finite state machine* yang memiliki kondisi tersembunyi Q, suatu nilai output O (observasi), kemungkinan/probabilitas transisi A, kemungkinan/probabilitas output B, sebuah kondisi awal Π. Kondisi saat ini tidak dapat dilihat (tidak terobservasi). Tetapi, setiap keadaan menghasilkan output kemungkinan B, sehingga HMM disebut triple (A,B, Π).

* Himpunan *observed state*: O = o1,o2,...,oN.
* Himpunan *hidden* *state*: Q = q1,q2,...,qN.
* Probabilitas transisi: A = ao1, ao2, ..., anm; dimana aij adalah probabilitas untuk berpindah dari *state* i ke *state* j.
* Probabilitas emisi atau *observation likelihood*: B = bi(Ot), yang merupakan probabilitas observasi Ot dibangkitkan oleh *state* i.
* *State* awal dan akhir: q0, qend, yang tidak terkait dengan observasi.



Gambar 2.4 Representasi Parameter HMM [1]

Dari Gambar 2.4, dapat dilihat bahwa:

* x = kondisi
* y = observasi yang mungkin
* a = kemungkinan/probabilitas keadaan transisi
* b = kemungkinan/probabilitas output

Ada tiga permasalahan khusus yang dapat diselesaikan oleh metode *Hidden Markov Model*, yakni:

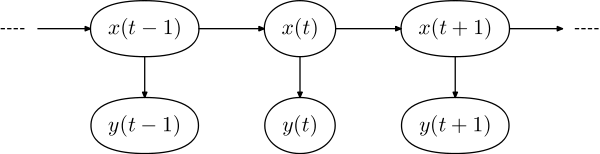
1. Evaluasi

Pengertian dari operasi evaluasi dalam *Hidden Markov Model* adalah perhitungan probabilitas dari urutan nilai observasi yang diberikan oleh *Hidden Markov Model*. Dimana dalam hal ini parameter dari HMM sudah diketahui dan yang dihitung adalah probabilitas dari suatu output tertentu, tanpa memperhatikan urutan output yang akan dihasilkan. Masalah ini dapat diselesaikan dengan algoritma *Forward* dan *Backward*. Adapun perbedaan algoritma *Forward* dan *Backward* adalah pada urutan observasi dari nilai probabilitasnya. Algoritma *Forward* seperti namanya, bergerak maju, dengan observasi sesuai dengan urutan, dan sebaliknya algoritma *Backward* bergerak secara mundur. Nilai probabilitas pada setiap langkah observasi digunakan dalam perhitungan selanjutnya.

#### Algoritma Forward [4]

Merupakan algoritma yang digunakan untuk menghitung ‘tingkat kepercayaan’ dari suatu *state* pada HMM. Proses ini sering dikenal dengan *filtering.* Algoritma *Forward* memiliki hubungan yang dekat, namun berbeda dengan Algoritma *Viterbi* yang akan dijelaskan nantinya.

Asumsikan terdapat HMM seperti pada Gambar 2.5:



Gambar 2.5 Contoh HMM [3]

Dimana suatu probabilitas dari suatu *state* dapat dituliskan sebagai , dengan adalah *hidden state,* adalah observasi pada saat 1 hingga t. Suatu ‘*state* yang dipercaya’ dapat dihitung untuk setiap time step, akan tetapi tidak akan bisa menghitung urutan *state* yang akan terjadi.

Tujuan dari algoritma *Forward* adalah untuk menghitung *Joint Probability*/Probabilitas Gabungan . Jika proses komputasi dilakukan secara langsung, akan diperlukan proses untuk menggeser semua *state* yang mungkin, sehingga komplesitas algoritma akan tumbuh secara eksponensial sehingga tidak efektif. Oleh karena itu algoritma *Forward* menggunakan kelebihan dari HMM yakni, kondisi *state* yang saling bebas bersyarat (*conditional independence)* untuk melakukan proses perhitungan secara rekursif.

|  |  |
| --- | --- |
|  | (2.9) |

Dengan menggunakan aturan rantai, didapatkan bahwa:

|  |  |
| --- | --- |
|  | (2.10) |

Karena saling bebas bersyarat dengan *state* lainnya kecuali dan juga saling bebas bersyarat dengan semuanya kecuali , maka persamaan diatas dapat disederhanakan menjadi:

|  |  |
| --- | --- |
|  | (2.11) |

Dengan demikian, proses perhitungan dari dapat dihitung dengan cepat menggunakan cara rekursif tanpa harus menimbulkan waktu komputasi yang eksponensial.

#### Algoritma Backward [4]

Sama seperti algoritma *Forward*, algoritma *Backward* ini juga digunakan untuk menghitung ‘tingkat kepercayaan’ dari suatu *state* di HMM. Akan tetapi proses yang dilakukan berkebalikan antara *state* yang diobservasi dengan *state* yang tersembunyi.

Dengan cara yang sama dengan algoritma *Forward*, variabel probabilitas bersyarat dari suatu urutan observasi mulai dari y(t+1) hingga selesai yang dihasilkan dari urutan *state* yang dimulai dari *state* ke- () dapat dituliskan bahwa:

|  |  |
| --- | --- |
|  | (2.12) |

Sama halnya dengan algoritma *Forward,* algoritma *Backward* juga bisa dihitung secara rekursif. Akan tetapi rekursif yang terjadi memiliki urutan yang terbalik dari urutan observasi yang dilakukan. Meskipun algoritma *Backward* ini lebih jarang digunakan untuk menghitung probabilitas suatu urutan yang diobservasi, tetapi algoritma ini akan sangat berguna ketika hendak dicari suatu urutan *state* yang optimal dan ketika memperkirakan parameter-parameter dari HMM.

|  |  |
| --- | --- |
|  | (2.13) |

1. Penarikan kesimpulan

Pengertian penarikan kesimpulan (inferensi) dalam *Hidden Markov Model* adalah penarikan kesimpulan berdasarkan asumsi yang diperoleh dari nilai probabilitas observasi yang didapat sebelumnya pada operasi evaluasi. Kesimpulan dalam hal ini merupakan bagaimana cara untuk menentukan urutan *hidden states* paling mungkin yang bisa terjadi yang akan menghasilkan output tertentu. Operasi ini juga sering kali digunakan untuk mencari nilai optimum. Masalah ini dapat diselesaikan dengan algoritma *Viterbi* dan *Posterior Decoding*.

#### Algoritma Viterbi [4]

Merupakan sebuah algoritma pemrograman dinamis yang berfungsi untuk menemukan urutan *state*ment yang disembunyikan, yang sering disebut dengan jalur Viterbi (*Viterbi path*). Pada awalnya diasumsikan bahwa terdapat sebuah *Hidden Markov Model* (HMM) dengan *state* space S, nilai probabilitas awal πi pada *state* i dan probabilitas transisi aij ketika transisi dari *state* i ke *state* j. Kemudian diasumsikan juga bahwa yang diobservasi adalah output y1, ..., yT. Akan didapatkan bahwa urutan *state* x1, ..., xT yang bisa menghasilkan observasi yang dikehendaki akan memenuhi relasi:

|  |  |
| --- | --- |
|  | (2.14) |

Dimana merupakan probabilitas dari urutan *state* yang paling mungkin yang bertanggung jawab atas observasi pertama dengan sebagai *state* terakhir yang dihasilkan. Jalur Viterbi bisa didapatkan dengan menyimpan data *state* (*pointers*) yang digunakan untuk persamaan kedua, dan persamaan berikutnya. Asumsikan merupakan persamaan yang mengembalikan nilai dari yang digunakan dalam proses komputasi dengan atau jika .

|  |  |
| --- | --- |
|  | (2.15) |
|  | (2.16) |

Kompleksitas dari algoritma ini adalah .

1. *Posterior Decoding* [4]

Sering disebut juga sebagai *posterior probability* yang dimana didefinisikan sebagai probabilitas dari parameter jika terdapat output . Hal ini merupakan kebalikan dari fungsi *likelihood* yang mencari nilai probabilitas dari output jika terdapat parameter: . Asumsikan terdapat fungsi distribusi probabilitas dan observasi/ouput dengan *likelihood* , dapat dituliskan persamaan matematis dari probabilitas *posterior*-nya adalah:

|  |  |
| --- | --- |
|  | (2.17) |

*Posterior decoding* bisa juga dituliskan dalam bentuk yang lebih mudah diingat secara konseptual, yakni:

1. Pembelajaran

Pengertian dari operasi *learning* dalam *Hidden Markov Model* adalah melatih parameter HMM jika diberikan dataset barisan-barisan tertentu agar dapat menemukan himpunan *transition state* yang paling mungkin beserta probabilitas outputnya.

Untuk menyelesaikan permasalahan *learning*, digunakan algoritma *Baum-Welch*. Algoritma ini secara umum berfungsi untuk menentukan nilai harapan dan maksimalisasi.

Algoritma ini mempunyai dua langkah dalam penyelesaian masalah, yaitu :

* 1. Menghitung nilai probabilitas *Forward* dan *Backward* untuk setiap *state*ment.
  2. Menentukan frekuensi dari pasangan transisi emisi dan membaginya dengan nilai probabilitas semua observasi.

#### Algoritma Baum-Welch [4]

Algoritma Baum-Welch digunakan untuk mencari parameter yang tidak diketahui pada HMM. Algoritma ini memanfaatkan algoritma *Forward*-*Backward*. Algoritma ini dinamai berdasarkan nama penemunya, yakni Leonard E. Baum dan Lloyd R. Welch [5].

HMM mendeskripsikan suatu probabilitas gabungan sebagai sekumpulan variabel diskrit yang acak dari *state* yang tersembunyi maupun *state* yang dapat diobservasi. Dengan asumsi bahwa variabel tersembunyi ke-i dan variabel tersembunyi ke-(i-1) saling bebas satu sama lain, maupun dengan variabel tersembunyi sebelumnya. Selain itu variabel yang diobservasi sekarang hanya bergantung pada *state* tersembunyi sekarang.

Pada algoritma *Baum-Welch* ini digunakan algoritma yang dikenal dengan nama EM (*Expectation-Maximization*) untuk mencari nilai perkiraan maksimum *likelihood* dari suatu set parameter yang diobservasi pada HMM. Dalam hal ini algoritma *Baum-Welch* dapat dituliskan dalam persamaan matematis, yakni:

|  |  |
| --- | --- |
|  | (2.18) |

Dimana menyatakan maksimum lokal dari parameter HMM () yang diobservasi.

Terdapat 3 tahapan pada algoritma *Baum-Welch*, yaitu:

1. *Forward*

Asumsikan terdapat , yang merupakan probabilitas untuk mendapatkan hasil observasi ketika sedang berada pada *state* dan waktu . Untuk menghitung nilai , dapat dilakukan dengan persamaan matematis, yakni:

|  |  |
| --- | --- |
|  | (2.19) |

1. *Backward*

Asumsikan terdapat , yang merupakan probabilitas dari berakhirnya sebagian urutan dengan diberikan awal *state* pada waktu . Maka nilai dapat dihitung dengan:

|  |  |
| --- | --- |
|  | (2.20) |

1. *Update*

Setelah melalui tahap *Forward* dan *Backward*, maka dapat dihitung suatu variabel sementara yang merupakan probabilitas sedang berada pada *state* pada waktu dengan diberikan urutan observasi dan parameter :

|  |  |
| --- | --- |
|  | (2.21) |

Selain itu juga dapat dihitung suatu variabel lainnya, yakni yang merupakan probabilitas dimana sedang berada di *state* dan pada saat dan dengan diberikan urutan observasi dan parameter :

|  |  |
| --- | --- |
|  | (2.22) |

Setelah menghitung kedua variabel diatas, parameter-parameter dapat diperbaharui menjadi:

Yang merupakan ekspektasi frenkuensi yang akan dihabiskan di *state* pada waktu .

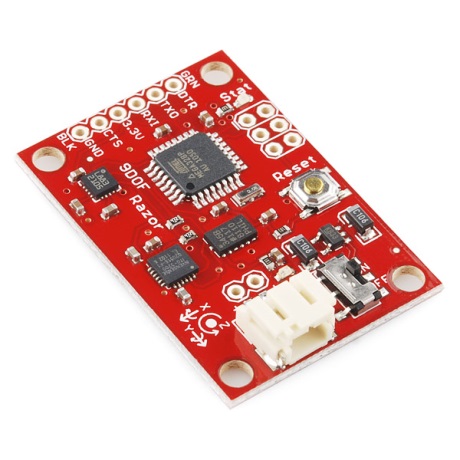
Yang merupakan nilai yang diharapkan ketika transisi dari *state* ke *state* jika dibandingkan dengan total nilai yang diharapkan ketika transisi dari *state* ke *state* lain yang belum tentu merupakan *state .*

Dimana yang merupakan fungsi indikator dan yang merupakan nilai/banyaknya output observasi yang sama dengan ketika berada di *state* dibandingkan dengan output lain pada *state* yang sama.

Pembaharuan terhadap nilai-nilai ini dilakukan secara berulang-ulang hingga didapatkan suatu kondisi yang konvergen.

1. Hardware
2. IMU

IMU (*Inertial Measurement Unit*) merupakan suatu sensor yang digunakan untuk mengukur percepatan, orientasi dan medan magnet dengan menggunakan kombinasi dari sensor *accelerometer, gyroscope,* dan *magnetometer*. IMU biasanya digunakan untuk pergerakan pesawat terbang ataupun satelit. Pada penelitian ini, IMU yang digunakan adalah *Razor IMU 9-Degrees of Freedom*.

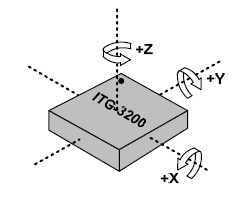


Gambar 2.6 Sensor Razor IMU [6]

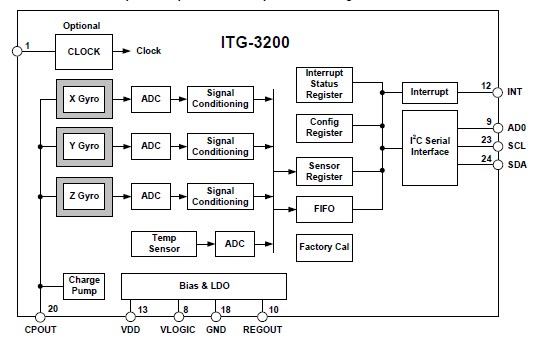
Adapun spesifikasi teknis yang dimiliki *Razor IMU 9-Degrees of Freedom* antara lain [6]:

* Terdiri dari 3 sensor, yakni ITG-3200 (*MEMS triple-axis gyroscope*), ADXL345 (*triple-axis accelerometer*), dan HMC5883L (*triple-axis magnetometer*).
* ITG-3200

Merupakan sensor *gyroscope* 3-axis yang dapat mengambil data percepatan sudut/*gyro* (*yaw* sumbu-z*, pitch* sumbu-y*, roll* sumbu-x) hingga ±2000º/detik seperti yang terlihat pada Gambar 2.7. Data juga bisa diukur secara digital 16-bit dengan menggunakan ADC yang sudah tersedia, dan proses transmisi data dapat dilakukan melalui *interface* I2C. Blok diagram fungsional ITG-3200 juga dapat dilihat pada Gambar 2.8.



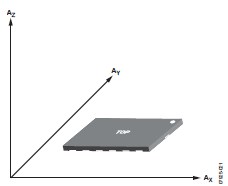
Gambar 2.7 Arah Orientasi dan Polaritas Rotasi [7]



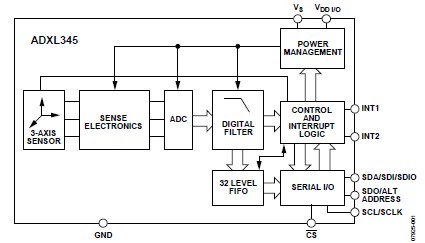
Gambar 2.8 Blok Diagram Fungsional ITG-3200 [7]

* ADXL345

Merupakan sensor percepatan linier (*accelerometer*) 3-axis dengan resolusi data hingga 13-bit dan dapat mengukur hingga ±16 kali percepatan gravitasi. Output data digital memiliki format data 2’s *complement* 16-bit dan dapat diakses melalui SPI atau *interface* I2C. Orienstasi dan blok diagram fungsional ADXL345 dapat dilihat pada Gambar 2.9 dan Gambar 2.10.

**

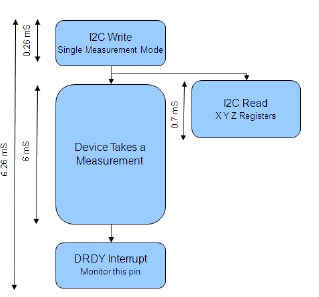
Gambar 2.9 Orientasi Sensitivitas ADXL345 [8]



Gambar 2.10 Blok Diagram Fungsional ADXL345 [8]

* HMC5883L

Merupakan sensor medan magnet yang memiliki ketelitian kompas 1º hingga 2º dengan 12-bit ADC. Komunikasi juga dapat dilakukan dengan *interface* I2C. Sensor ini dapat digunakan untuk mengukur arah orientasi serta besarnya medan magnet bumi, yang berkisar antara milli-gauss hingga 8 gauss. Periode proses pengambilan data pada HMC5883L dapat dilihat pada Gambar 2.11.



Gambar 2.11 Periode Pengambilan Data Magnetometer [9]

* Memiliki board Atmega328 sebagai processor utama untuk mengolah data dari sensor.
* Terdapat fitur *autorun*.
* Pin output yang tersedia dapat dengan mudah di gunakan dan dikoneksikan dengan FTDI Basic Breakout, Bluetooth Mate, atau Xbee Explorer.
* 3.5-16 VDC input.

1. Zigbee dan Xbee *Wireless*

Zigbee merupakan suatu protokol komunikasi tingkat tinggi yang menggunakan radio digital dengan ukuran yang kecil dan daya yang relatif rendah, serta berbasis pada standar IEEE 802.15.4-2003 untuk jaringan personal nirkabel tingkat rendah. Sedangkan Xbee *wireless* merupakan sebuah modul *RF (Radio Frequency)* yang dimana sering digunakan programmer sistem *embedded* yang ingin menggunakan sistem komunikasi *wireless* *end-point connectivity* menuju suatu devais. Modul Xbee *wireless* ini juga menggunakan jaringan protokol IEEE 802.15.4 untuk *fast point-to-multipoint* atau *peer-to-peer* yang sesuai dengan protokol jaringan Zigbee. Komunikasi *wireless* ini dirancang untuk aplikasi yang memiliki *high-throughput* serta memerlukan *latency* yang rendah, dan waktu komunikasi yang dapat diprediksi. Xbee *wireless* juga merupakan alat yang ideal untuk digunakan karena konsumsi daya yang rendah dan biaya yang tidak terlalu mahal. Modul Xbee *wireless* bekerja pada band ISM (*industrial, scientific, medical*) 2,4GHz dan sudah dilengkapi dengan chip antena. Pada penelitian ini, Xbee *wireless* yang digunakan dalah Xbee Chip Antenna Series 2, seperti yang terlihat pada Gambar 2.12 [10].

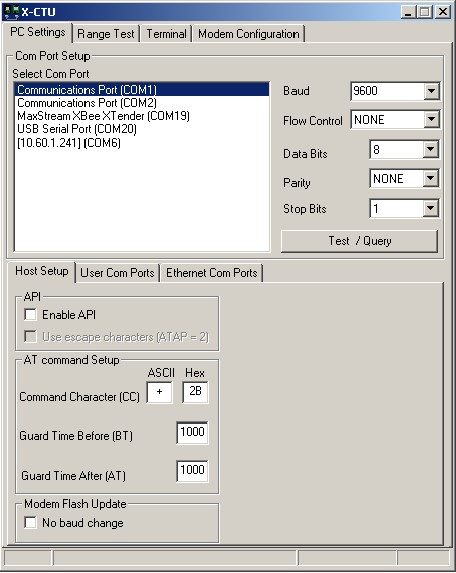


Gambar 2.12 XBee Chip Antenna Series 2 [10]

Adapun spesifikasi teknis yang dimiliki oleh Xbee Chip Antenna Series 2 antara lain [10]:

* Memiliki jarak komunikasi sejauh 30 meter untuk lingkungan *indoor*.
* Memiliki jarak komunikasi sejauh 90 meter untuk lingkungan *outdoor* jika terjadi *line-of-sight.*
* *RF Data Rate* sebesar 250kbps.
* *Serial Interface Data Rate* sebesar 1200bps – 1Mbps.
* *Data Throughput* hingga 35kbps.
* Suplai tegangan 2,8 hingga 3,6 Volt.
* ISM dengan operasi frekuensi 2,4GHz.
* Mendukung jika digunakan pada jaringan *mesh.*

Dari segi *software*, Xbee dapat diubah *firmware*/fungsi yang akan dijalankannya dengan menggunakan *software* yang disediakan, yakni X-CTU.



Gambar 2.13 *Interface* X-CTU [10]

Dari Gambar 2.13, dapat dilihat bahwa melalui X-CTU ini dapat dilakukan berbagai pengaturan terhadap PORT komunikasi yang sedang digunakan melalui *PC Settings*, *Range Test* untuk melakukan pengujian terhadap jarak maksimum komunikasi, *Terminal* untuk melihat hasil komunikasi Serial, dan *Modem Configuration* untuk melakukan pembaharuan/instalasi baru terhadap *firmware* yang akan digunakan oleh Xbee *Wireless*. Beberapa contoh *firmware* yang bisa digunakan oleh Xbee Chip Antenna Series 2 ini antara lain ZNET 2.5 Coordinator API, ZNET 2.5 Router/End Device AT, ZNET 2.5 Router/End Device Analog IO. [10]

1. MATLAB [11]

MATLAB merupakan bahasa pemrograman tingkat tinggi yang menyediakan lingkungan yang interaktif untuk proses komputasi numerik, visualisasi, dan programming. MATLAB sendiri memiliki fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic maupun C/C++. Oleh karena itu, MATLAB banyak digunakan dalam bidang riset-riset yang memerlukan komputasi numerik yang kompleks.

Penggunaan MATLAB meliputi bidang-bidang:

* Matematika dan komputasi.
* Pembentukan algorithma.
* Akusisi data.
* Pemodelan, simulasi, dan pembuatan prototipe.
* Analisa data, eksplorasi, dan visualisasi.
* Grafik keilmuan dan rekayasa.

MATLAB sendiri merupakan kepanjangan dari *Matrix Laboratory*. Sesuai dengan namanya, struktur data yang terdapat dalam MATLAB menggunakan matriks atau array berdimensi dua. Oleh karena itu, penguasaan teori matriks sangat penting bagi pengguna MATLAB agar mudah dalam mempelajari dan memahami operasi-operasi yang ada di MATLAB. MATLAB hadir dalam nuansa warna yang berbeda, hal ini karena membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan visualisasi. MATLAB dikembangkan oleh MathWorks, yang pada awalnya dibuat untuk memberikan kemudahan mengakses data matriks pada proyek LINPACK dan EISPACK. Saat ini MATLAB memiliki ratusan fungsi yang dapat digunakan sebagai *problem solver* mulai dari yang sederhana sampai masalah-masalah yang kompleks dari berbagai disiplin ilmu.

1. Perbandingan dengan Penelitian Lainnya

* *Human Motion Recognition Using A Wireless Sensor-Based Wearable System* (2011) [12]

Pada penelitian ini, dibahas mengenai pengenalan gerakan manusia, antara lain: gerakan manusia ketika berdiri, berjalan, berlari, mendongkrak, menulis, dan merokok. Adapun untuk sensor yang digunakan dalam mengenali gerakan manusia adalah sensor *accelerometer* dan *gyroscope* (*Shimmer mote*), sensor ini nantinya akan dipakai pada pergelangan tangan kanan dan pergelangan kaki kanan manusia. Untuk algoritma pengenalan digunakan metode klasifikasi *Support Vector Machine* (SVM).

* *Automated Video Surveillance System for Human Motion Detection* (2013) [13]

Pada penelitian ini, dibahas mengenai pengenalan gerakan manusia, akan tetapi menggunakan teknik *image processing*, yakni melalui kamera CCTV. Metode yang digunakan hingga bisa mengenali gerakan manusia: diawali dengan proses pengambilan gambar dari kamera, kemudian diproses untuk dibedakan bagian mana yang merupakan latar belakang gambar dan bagian mana yang menjadi objek yang akan dianalisis, tahap berikutnya adalah pengurangan gambar dengan gambar latar belakangnya sehingga hanya tersisa objek, proses pengurangan terhadap bagian yang didepan objek namun bukan objek juga dilakukan. Tahap terakhir yang dilakukan adalah pendeteksian gerakan manusia dan kemudian diklasifikasikan gerakan tersebut dengan *Support Vector Machine* (SVM).

* Perbandingan

Tabel 3.1 Perbandingan dengan Penelitian Lain

|  |  |  |  |
| --- | --- | --- | --- |
| **Hal yang dibandingkan** | ***Wireless* Sensor-Based** | **Video Surveillance-Based** | **Sistem yang akan dibuat** |
| **Sensor** | *Shimmer mote (accelerometer* dan *gyroscope)* | Kamera CCTV | *Razor* IMU 9-DOF |
| **Algoritma** | SVM | *Image Processing,* SVM | ANN, HMM, SVM |
| **Komunikasi** | *Wireless* melalui *Shimmer mote* | Tidak ada proses komunikasi | *Wireless Xbee* |
| **Letak sensor di tubuh manusia** | Pergelangan tangan kanan dan pergelangan kaki kanan | - | Pergelangan tangan |
| **Jenis gerakan yang dikenali** | Berdiri, berjalan, berlari, mendongkrak, menulis, dan merokok | - | Tangan lurus dan berayun, tangan ditekuk dan berayun, pergelangan tangan memutar kekiri dan kekanan, tangan mendorong dan menarik |
| **Kelebihan** | Tingkat akurasi yang cukup tinggi karena banyaknya jumlah sensor | Pendeteksian dapat dilakukan terhadap banyak objek pada satu waktu yang bersamaan | Biaya yang cukup murah untuk bisa mengenali gerakan manusia |
| **Kelemahan** | Harga alat dan sensor yang mahal | Proses pengolahan data yang cukup rumit dengan proses yang lebih panjang | Jarak komunikasi yang cukup terbatas (±20m) |

1. PERANCANGAN SISTEM UNTUK PENGENALAN GERAKAN MANUSIA

Pada bab ini akan dibahas mengenai perancangan sistem berkaitan dengan sistem yang akan dibangun untuk mengenali gerakan manusia. Berdasarkan *Software Development Life Cycle* (SLDC), setelah menentukan tema dan batasan masalah serta tujuan, maka tahapan berikutnya dalam pembuatan sistem pengenalan gerakan manusia adalah perencanaan. Tahapan perencanaan ini meliputi *system requirement*, desain, dan implementasi. Pengujian sistem dan analisa tidak akan dibahas pada bab ini. Dalam mendokumentasikan setiap tahapan SLDC, *Unified Modeling Language* (UML) akan digunakan sebagai metode standar. Dengan UML, rancangan perangkat keras serta alur kerja dapat direpresentasikan ke dalam diagram-diagram yang memiliki fungsi masing-masing.

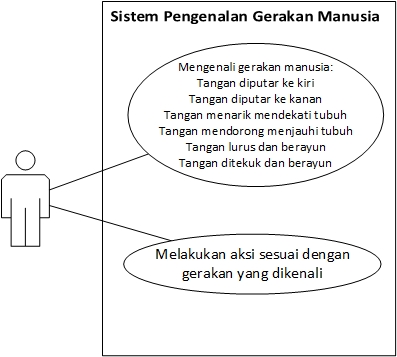
1. System Requirement

*System Requrement* merupakan tahapan yang mendefinisikan sistem dan fitur yang dibutuhkan. Tahapan ini memegang peranan penting dalam perancangan fungsionalitas sistem yang akan dibuat. Terdapat satu pengguna yang akan dikenali gerakannya oleh sistem.

Di bawah ini adalah hasil pengumpulan *requirement* yang didapat melalui tukar wawasan dengan beberapa pengguna dan studi literatur:

* Pengguna memasang sensor dari sistem pada pergelangan tangan pengguna dan kemudian bergerak seperti biasa.
* Gerakan pengguna dapat dikenali oleh sistem dan sistem akan memberikan aksi (keluran berupa tampilan pada GUI MATLAB) yang sesuai untuk setiap gerakan yang dikenali. Adapun gerakan-gerakan yang akan dikenali antara lain:

1. Gerakan tangan lurus dan berayun. Gerakan ini dikenali ketika dari keadaan diam/tangan lurus terjadi langkah tangan berayun, seperti manusia berjalan. Sebagaimana yang terlihat pada Gambar 1.1b.
2. Gerakan tangan ditekuk dan berayun. Gerakan ini dikenali ketika dari keadaan tangan ditekuk, terjadi langkah tangan berayun, seperti manusia berlari. Sebagaimana yang terlihat pada Gambar 1.1a.
3. Gerakan pergelangan tangan diputar ke kiri. Pergelangan tangan dianggap sudah memutar ke kiri jika sensor sudah mendeteksi arah putaran ke kiri lebih besar atau sama dengan 90º.
4. Gerakan pergelangan tangan diputar ke kanan. Pergelangan tangan dianggap sudah memutar ke kanan jika sensor sudah mendeteksi arah putaran ke kanan lebih besar atau sama dengan 90º.
5. Gerakan tangan mendorong ke depan. Tangan dianggap sudah mendorong ke depan ketika tangan yang awalnya menyentuh dada didorong hingga lurus.
6. Gerakan tangan menarik mendekati tubuh. Tangan dianggap sudah menarik mendekati tubuh ketika tangan yang awalnya lurus ditarik hingga menyentuh dada.



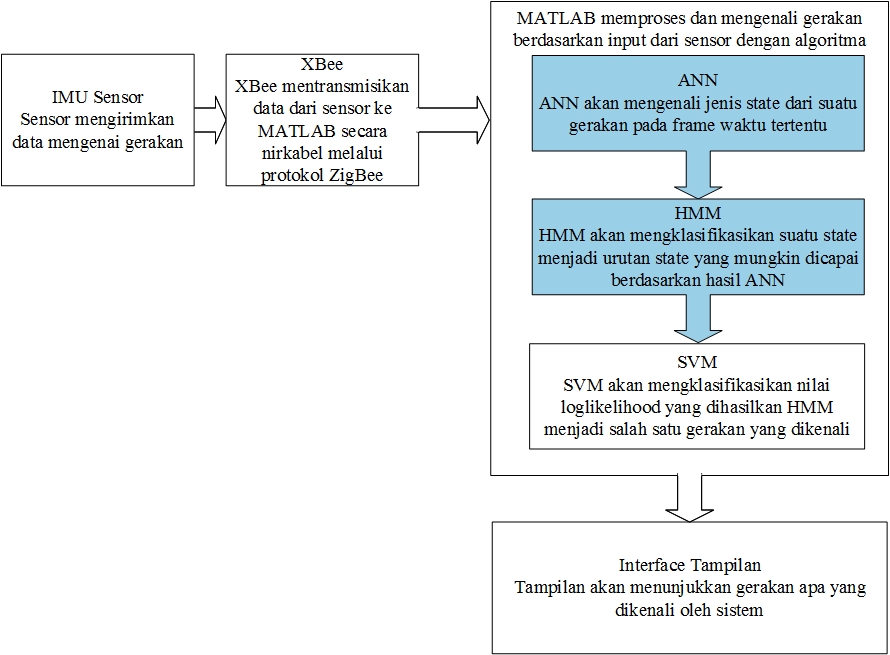
Gambar 3.1 *Use Case* Diagram

Fungsi-fungsi di atas merupakan fungsi-fungsi yang harus ada pada sistem ini guna memenuhi kebutuhan pengguna yang akan menggunakannya. *Requirement* di atas merupakan *requirement* dasar yang masih dapat dikembangkan lagi sehingga menjadi sebuah sistem yang lebih baik. Gambar 3.1 menggambarkan fungsi yang sudah diolah dalam *use case* diagram.

1. System and *Software* Design

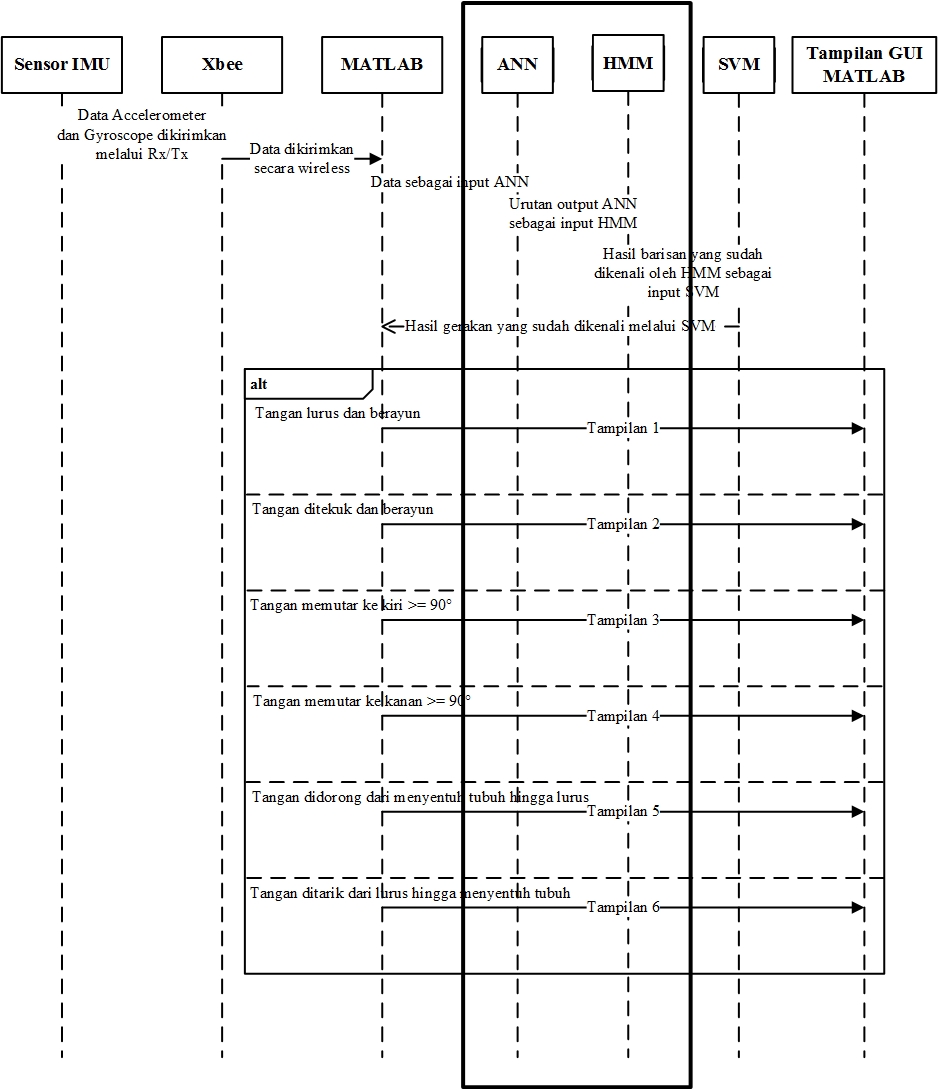
Sistem pengenalan gerakan manusia yang dibuat dalam penelitian ini dirancang untuk memberikan solusi yang cepat dan akurat dalam mengenali gerakan pengguna yang spesifik yang nantinya dapat berguna dalam berbagai bidang lainnya. Sistem ini akan dikendalikan oleh program MATLAB pada komputer. Sedangkan untuk pendeteksi gerakan digunakan sensor IMU (*Inertial Measurement Unit*). Untuk proses pengiriman data secara nirkabel digunakan Xbee *Wireless.*

Blok diagram dari sistem yang digunakan secara keseluruhan ditunjukkan pada Gambar 3.2. Bagian yang dikerjakan pada penelitian ini diberi warna biru.



Gambar 3.2 Blok Diagram Sistem Pengenalan Gerakan Manusia

Sesuai dengan Gambar 3.2 sistem ini dibagi menjadi dua bagian yang memiliki peranan yang berbeda dalam mengenali gerakan manusia. Bagian-bagian tersebut antara lain blok pengenalan *state* gerakan manusia dan pengambilan kesimpulan atas gerakan apa yang dikenali oleh sistem serta blok pengenalan suatu urutan *state* gerakan yang dilakukan oleh pengguna. Lingkup pada penelitian ini adalah pengenalan atas urutan *state* gerakan yang dilakukan oleh pengguna.



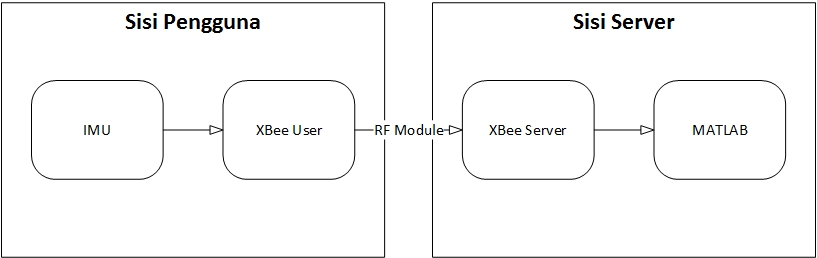
Gambar 3.3 *Sequence* Diagram

Dari Gambar 3.3, dapat dilihat bahwa urutan berjalannya sistem akan dimulai dari sensor IMU yang mengirimkan data *accelerometer* dan data *gyroscope* secara serial melalui Xbee *Wireless* ke MATLAB di komputer untuk diproses. Adapun proses pengolahan data yang terjadi di dalam MATLAB adalah pemrosesan data dari sensor sebagai input bagi *Artificial Neural Network* (ANN), kemudian hasil pengenalan sensor dari ANN ini akan digunakan sebagai input bagi *Hidden Markov Model* (HMM) dalam mengenali *sequence/*urutan dari *state* yang akan dikenali sebagai sebuah gerakan. Lalu output *loglikelihood* dari HMM akan digunakan sebagai input bagi *Support Vector Machine* (SVM) dalam mengklasifikasikan gerakan yang dilakukan oleh pengguna. Ketika sudah diklasifikasikan, output jenis gerakan yang dikenali sistem akan menghasilkan indikator keluaran GUI yang berbeda.

1. Desain Perangkat Keras (*Hardware*)

Sistem yang digunakan akan berbentuk seperti gelang dan diletakkan pada pergelangan tangandari pengguna. Pada sistem akan terdapat sensor IMU yang ditempatkan berdampingan dengan Xbee *Wireless* dan rangkaian yang disertai baterai.

Sistem pada dasarnya dibagi menjadi 2 bagian, yaitu bagian pada sisi pengguna dan pada sisi server. Pada sisi pengguna terdapat sensor IMU yang digunakan untuk merekam data gerak pengguna. Pada sisi server merupakan bagian dimana proses komputasi dilakukan. Seluruh pemilihan keputusan dan proses pembelajaran dilakukan pada sisi ini. Diantara pengguna dan server, digunakan komunikasi secara nikabel yaitu dengan menggunakan Xbee *Wireless*. Xbee yang digunakan ada dua, yaitu Xbee pengirim (*transmitter*), yang akan mengirimkan data dari sisi pengguna ke server dan Xbee penerima (*receiver*) yang akan menunggu data yang dikirimkan dari sensor ke sisi server.



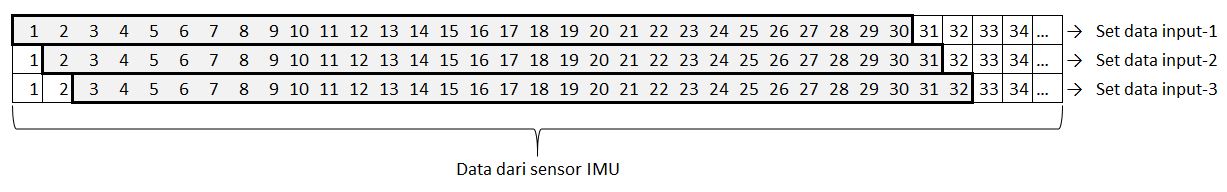
Gambar 3.4 Blok *Hardware* Sistem

1. Teknik Pengambilan Data Sensor

Sistem pengenalan gerakan yang dibuat akan mengolah data langsung dari sensor IMU Razor 9-DOF. Oleh karena itu perlu ada teknis/ketetapan dalam mengambil data dari sensor yang akan digunakan oleh sistem. Dari kesembilan dimensi yang bisa dihasilkan oleh sensor Razor IMU, sistem yang dibuat hanya menggunakan tiga dimensi keluaran dari sensor, yakni 3-*axis gyroscope*: *yaw, pitch,* dan *roll*. Alasan menggunakan hanya tiga dimensi dari *gyroscope* adalah dalam pengenalan gerakan manusia, terutama gerakan pergelangan tangan, yang paling mempengaruhi dan paling terlihat perubahannya adalah percepatan sudut dari tangan manusia. Sedangkan untuk percepatan linear dan medan magnet yang dihasilkan akibat gerakan merupakan data yang kurang berpengaruh dalam mengenali gerakan pergelangan tangan manusia.

Untuk pengambilan data pelatihan, semua gerakan dilakukan oleh satu orang subjek saja. Sedangkan untuk data pengujian setiap gerakan diulang sebanyak 5 kali untuk satu orang dengan 4 orang sebagai subjek pengambilan data.

Data dari sensor ini akan dihasilkan/diperbaharui untuk setiap selang waktu 50 ms. Untuk *input* bagi sistem, data yang dikirimkan akan terdiri dari 10 set data yang dimana 10 set data ini terdiri dari 10 data *yaw,* 10 data *pitch*, dan 10 data *roll*, sehingga satu kali input bagi sistem adalah 30 data dari sensor. Cara pengambilan 30 data ini menggunakan teknik *sliding window* terhadap data dari sensor menurut urutan selang waktunya.



Gambar 3.5 Teknik *Sliding Window* dalam Pengambilan Data dari Sensor

1. Algoritma Perangkat Lunak (*Software*)

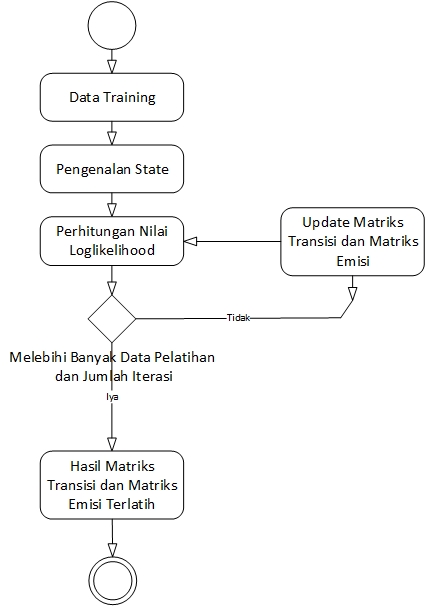
Secara garis besar akan terdapat dua fase pada sistem dalam mengenali gerakan manusia, yakni fase pelatihan sistem dan fase pengenalan. Pada setiap fase ini akan terdapat algoritma berbeda yang akan digunakan sistem dalam melakukan pengolahan data.

Fase pelatihan adalah fase dimana algoritma dilatih agar dapat mengenali gerakan yang dimaksud. Fase pelatihan ini pada umumnya adalah fase pengumpulan data *training* untuk menjadi dasar bagi algoritma agar mampu mengenali gerakan yang berbeda. Adapun gerakan-gerakan yang akan dikenali antara lain, gerakan tangan lurus dan berayun, gerakan tangan ditekuk dan berayun, pergelangan tangan diputar ke kiri dan ke kanan, gerakan tangan mendorong ke depan serta gerakan tangan menarik mendekati tubuh. Fase pelatihan ini bertujuan agar sistem tidak melakukan banyak kesalahan.

Sedangkan pada fase pengenalan merupakan fase dimana sistem sudah melewati fase pelatihan. Sehingga sistem dapat dianggap telah dapat digunakan untuk mengenali gerakan-gerakan berbeda yang dilakukan oleh pengguna dengan menggunakan data pembanding yang telah dilatih dan disimpan sebelumnya oleh pengguna.

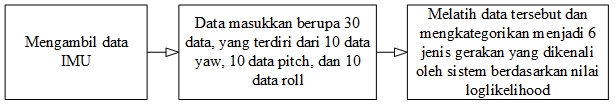
Pada penelitian ini, untuk setiap fase, akan terdapat dua variasi dalam algoritma yang digunakan, yakni fase pengenalan dan fase pelatihan yang dengan menggunakan algoritma *Hidden Markov Model* saja, serta fase pengenalan dan fase pelatihan yang menggunakan algoritma *Artificial Neural Network* terlebih dahulu dan kemudian menggunakan algoritma *Hidden Markov Model*.

1. Fase pelatihan dengan algoritma *Hidden Markov Model*



Gambar 3.6 *Activity* Diagram Ketika Pelatihan dengan HMM

Dari sisi pengguna, diagram aktifitas ketika pelatihan dilakukan terhadap sistem dapat dilihat seperti pada Gambar 3.6. Fase pelatihan merupakan fase dimana sistem masih akan belajar untuk mengenali gerakan tertentu secara akurat. Oleh karena itu pengguna akan melakukan suatu gerakan yang sama untuk beberapa kali yang nantinya akan digunakan sebagai set data *training* untuk dikenali *state*-nya lalu dilakukan perhitungan terhadap nilai *loglikelihood* dari sistem. Pada fase pelatihan ini, nilai dari matriks emisi dan matriks transisi akan terus dilatih dan diperbaharui nilainya selama masih ada data yang perlu dilatih serta masih dalam iterasi yang ditetapkan.



Gambar 3.7 Fase Pelatihan dengan HMM

Sedangkan dari sisi yang lebih mendetail, pada fase pelatihan, seperti pada Gambar 3.7, data akan diambil oleh sensor IMU untuk setiap satuan waktu. Data dari sensor IMU yang aka dipakai adalah data percepatan sudut (*gyro*) 3-axis, sehingga untuk setiap satuan waktunya terdapat 3 data yang akan diambil. Untuk masing-masing gerakan digunakan 10 *frame* waktu yang masing-masing *frame* waktunya terdiri dari 3 data yang dipakai, dengan demikian akan terbentuk data dengan 30 dimensi yang akan digunakan sebagai input bagi HMM, dengan output yang akan dihasilkan berdimensi 6, yang kemudian masing-masing HMM tersebut akan memiliki nilai *loglikelihood* yang berbeda yang dimana nilai *loglikelihood* ini akan dihitung nilai maksimumnya dan kemudian dikategorikan berdasarkan HMM yang memiliki nilai *loglikelihood* paling bagus.

Pseudocode pada proses pelatihan dengan HMM seperti pada Gambar 3.8, diawali dengan inisiasi model yang digunakan, antara lain probabilitas awal dari *state*, probabilitas transisi, dan probabilitas emisi. Kemudian diasumsikan S sebagai semua data pelatihan yang ada bagi sistem, lalu proses dilanjutkan dengan iterasi oleh sistem terhadap data input yang ada. Di dalam setiap iterasi, dilakukan proses *forward* dan *backward*, kemudian dari kedua proses ini dilakukan *update* terhadap matriks transisi dan matriks emisi dari sistem. Keluaran yang dihasilkan oleh sistem adalah L, suatu nilai *likelihood* dari data pelatihan S.

/\*\*Sebuah Model HMM adalah triple dimana

1. adalah awal probabilitas.
2. adalah probabilitas dari menjadi
3. adalah probabilitas muncul dari \*\*/

Require: semua data pelatihan

Require:

FOR DO

FOR DO

FOR DO

FOR DO

FOR DO

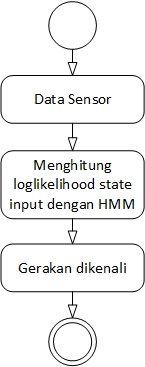
FOR DO

FOR DO

RETURN L, likelihood dari *training* data S

Gambar 3.8 Pseudocode Pelatihan dengan HMM

1. Fase pengenalan dengan algoritma *Hidden Markov Model*



Gambar 3.9 *Activity* Diagram Ketika Pengenalan dengan HMM

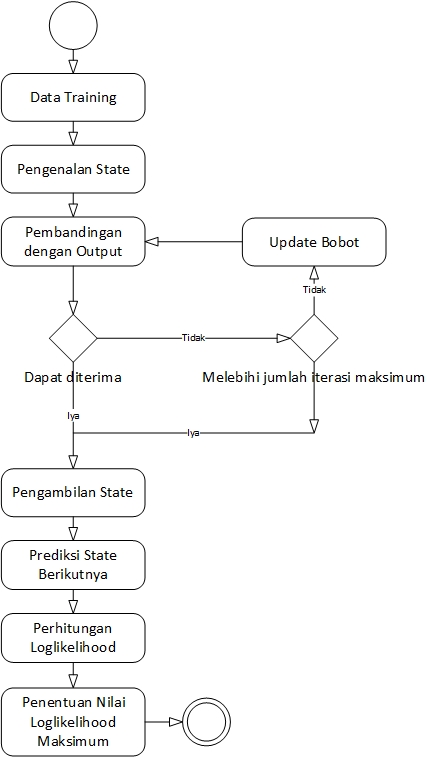
Dari sisi pengguna, diagram aktifitas ketika sistem melakukan proses pengenalan dapat dilihat seperti pada Gambar 3.9. Fase pengenalan merupakan fase dimana sistem sudah selesai mempelajari gerakan yang ada dan siap untuk digunakan oleh pengguna dalam melakukan pengenalan gerakan. Sehingga pada fase ini, pengguna cukup melakukan gerakan yang kemudian akan menjadi data sensor sebagai input sistem, lalu dari input ini akan dikenali oleh HMM sebagai suatu *state* gerakan tertentu, kemudian menghitung nilai *loglikelihood* suatu urutan *state* yang ada dengan HMM. Setelah itu akan diklasifikasikan dengan melihat nilai *loglikelihood* yang paling bagus dan menerjemahkan menjadi salah satu dari keenam gerakan yang mungkin.



Gambar 3.10 Fase Pengenalan dengan HMM

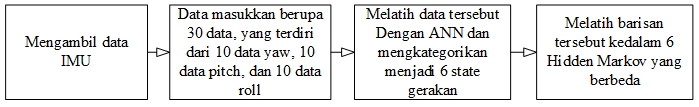
Dari sisi yang lebih mendetail, fase pengenalan seperti pada Gambar 3.10, seluruh algoritma diasumsikan telah dilatih saat fase pelatihan sehingga telah mampu mengenali gerakan yang telah ditentukan sebelumnya. Pada fase pengenalan, data diambil dari IMU berupa 3-axis *gyroscope* selama 10 potongan waktu. Sehingga dari 30 data tersebut, akan dijadikan sebagai dimensi bagi input HMM. Kemudian suku-suku tersebut dinamakan kondisi/*state* pada HMM. Proses berikutnya adalah proses kalkulasi nilai *loglikelihood* dari setiap kemungkinan urutan *state* yang ada dengan menggunakan 6 arsitektur HMM yang berbeda. Berdasarkan hasil dari ke-6 nilai *loglikelihood* tersebut, sistem akan mengklasifikasikannya berdasarkan nilai yang paling bagus dari *loglikelihood* sistem dan kemudian ditentukan gerakan apa yang dikenali oleh sistem dari input gerakan oleh pengguna.

1. Fase pelatihan dengan algoritma *Artificial Neural Network* dan *Hidden Markov Model*



Gambar 3.11 *Activity* Diagram Ketika Pelatihan dengan ANN dan HMM

Dari sisi pengguna, diagram aktifitas ketika pelatihan dilakukan terhadap sistem dapat dilihat seperti pada Gambar 3.11. Fase pelatihan ini menggunakan data *training* untuk dikenali *state*-nya dengan ANN lalu dibandingkan dengan output yang seharusnya. Jika terdapat perbedaan yang tidak dapat ditoleransi dan masih belum melebihi iterasi maksimum dari sistem maka akan terjadi pembaharuan terhadap bobot dari sistem ANN ini. Jika dapat diterima maka sistem akan dilanjutkan ke tahap berikutnya yakni pengenalan dan prediksi dari *state* berikutnya untuk membentuk suatu urutan *state*. Kemudian dilakukan perhitungan terhadap nilai *loglikelihood* dari sistem. Nilai *loglikelihood* ini akan dicari nilai maksimumnya untuk menentukan jenis gerakan yang dikenali oleh sistem secara keseluruhan.



Gambar 3.12 Fase Pelatihan dengan ANN dan HMM

Sedangkan dari sisi yang lebih mendetail, pada fase pelatihan, seperti pada Gambar 3.12, data yang diambil oleh sensor IMU akan digunakan sebagai input bagi ANN, dengan output yang akan dihasilkan berupa 6 *state* gerakan yang dikenali, yang kemudian akan diolah oleh masing-masing HMM untuk mencari nilai *loglikelihood* maksimumnya dan kemudian dikategorikan berdasarkan HMM yang memiliki nilai *loglikelihood* paling bagus.

Psedocode pada proses pelatihan dengan algoritma ANN dan HMM seperti pada Gambar 3.13, diawali dengan proses pengolahan data input dengan menggunakan algoritma ANN. Kemudian keluaran dari algoritma ANN ini akan digunakan oleh sistem untuk menjalankan algoritma HMM, sama seperti pada fase pelatihan sebelumnya, akan tetapi dengan input yang berbeda.

Require: semua data pelatihan

Require:

FOR DO

FOR DO

FOR DO

FOR DO

FOR DO

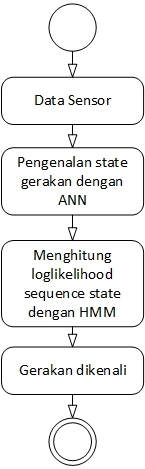
FOR DO

FOR DO

RETURN L, likelihood dari *training* data S

Gambar 3.13 Pseudocode Pelatihan dengan ANN dan HMM

1. Fase pengenalan dengan algoritma *Artificial Neural Network* dan *Hidden Markov Model*



Gambar 3.14 *Activity* Diagram Ketika Pengenalan dengan ANN dan HMM

Dari sisi pengguna, diagram aktifitas ketika sistem melakukan proses pengenalan dapat dilihat seperti pada Gambar 3.14. Pada fase ini, input dari sensor akan dikenali oleh ANN sebagai suatu *state* gerakan tertentu, kemudian dihitung nilai *loglikelihood* suatu urutan *state* yang ada dengan HMM. Setelah itu akan diklasifikasikan dengan menggunakan nilai *loglikelihood* yang paling sesuai, yang dimana akan menghasilkan output salah satu dari gerakan yang dikenali oleh sistem.

1. IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dibahas mengenai implementasi dan pengujian sistem pengenalan gerakan manusia serta analisis perbandingan penggunaan algoritma *Hidden Markov Model* dengan gabungan dari algoritma *Artificial Neural Network* dan *Hidden Markov Model*. Pengujian dilakukan untuk mengukur akurasi kebenaran dari sistem dalam mendeteksi dan mengenali gerakan manusia.

1. Implementasi Sistem

Sistem pengenalan gerakan manusia yang dibuat menggunakan sensor percepatan sesaat. Perangkat keras yang digunakan antara lain: sensor IMU Razor 9DOF dan XBee *Wireless*. Perangkat lunak yang digunakan dalam pembuatan sistem yaitu: MATLAB.

Untuk mengetahui performa dari sistem pengenalan gerakan manusia ini, maka dibutuhkan pengujian dengan cara pengambilan beberapa data. Pengujian dilakukan dengan menggunakan data berupa gerakan pergelangan tangan manusia yang dipasangi sensor dengan spesifikasi:

1. Durasi tiap gerakan 50ms.
2. Pengulangan sebanyak 20 kali untuk gerakan tangan memutar, tarik, dan dorong.
3. Untuk gerakan tangan lurus serta ditekuk lalu berayun dilakukan sejauh 30 meter.

Sedangkan mesin pengolah yang digunakan adalah sebuah komputer laptop dengan spesifikasi:

* Processor Intel® Core™ i5-4200U CPU @ 1.60GHz × 4
* Memori 3.7 GB
* MATLAB 2014a

Untuk *hardware* sistem yang dibuat, dapat dilihat seperti pada Gambar 4.1. *Hardware* sistem terdiri dari:

1. Baterai 9 volt.
2. Sensor IMU dan rangkaian *transmitter*.
3. Kabel USB.
4. XBee *receiver*.
5. Gelang.



E

C

B

A

D

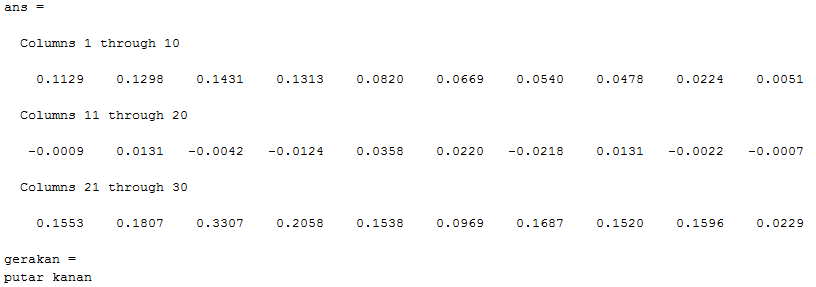
Gambar 4.1 Alat Pengenalan Gerakan

Untuk cara pengambilan data dapat dilihat seperti pada Gambar 4.2.



Gambar 4.2 Pengambilan Data

Untuk keluaran dari sistem, berupa suatu tampilan pada *Command Window* MATLAB, seperti pada Gambar 4.3, dapat dilihat bahwa terdapat keluaran *“ans”* yang terdiri dari kolom 1 hingga 30, dimana ini merupakan data yang menjadi inputan dari sensor. Sedangkan untuk hasil gerakan yang dikenali oleh sistem dapat dilihat pada bagian “gerakan = putar kanan” yang menyatakan gerakan yang dikenali pada Gambar 4.3 adalah gerakan tangan putar kanan.



Gambar 4.3 Tampilan Hasil pada *Command Window* MATLAB

1. Pengujian Akurasi Pengenalan Gerakan Manusia dengan Menggunakan *Hidden* Markov Model

Tabel 4.1 Akurasi Sistem Menggunakan HMM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Gerakan** | **Jumlah Data Testing** | **Jumlah Data yang Dikenali** | ***Recognition rate* (%)** |
| 1 | Putar kanan | 127 | 112 | 88.19 |
| 2 | Putar kiri | 80 | 71 | 88.75 |
| 3 | Dorong | 79 | 79 | 100 |
| 4 | Tarik | 77 | 77 | 100 |
| 5 | Tangan lurus dan berayun | 744 | 421 | 56.59 |
| 6 | Tangan ditekuk dan berayun | 599 | 589 | 98.33 |
|  |  |  |  | **88.64** |

Untuk data pengujian ini, setiap gerakan diulang sebanyak 5 kali untuk satu orang dengan 4 orang sebagai subjek pengambilan data, 2 orang diantaranya wanita dan 2 orang sisanya pria dengan umur 21 tahun. Untuk algoritma *Hidden Markov Model*, data input berupa data langsung dari sensor IMU, yang dimana satu set data masukkan terdiri dari 30 data, yang dapat dibagi menjadi: 10 data *yaw*, 10 data *pitch*, dan 10 data *roll*. Keluran dari sistem adalah nilai *loglikelihood* dari setiap set data yang kemudian diterjemahkan menjadi jenis gerakan hasil pengenalan oleh sistem.

Dari segi jumlah data pengujian, dapat dilihat bahwa jumlah data untuk gerakan tangan berayun jauh lebih banyak dibandingkan dengan gerakan lainnya, hal ini disebabkan untuk gerakan tangan berayun, parameter dalam pengambilan data yang digunakan berbeda jika dibandingkan dengan gerakan lainnya. Jika pada gerakan lain parameter pengambilan data adalah sebanyak 20 kali pengulangan untuk satu gerakan yang sama, maka untuk gerakan tangan berayun parameter pengambilan data yang digunakan adalah jarak yang sudah ditempuh oleh pengguna, yakni sejauh 30 meter.

Dari segi hasil, seperti yang dapat dilihat pada Tabel 4.1, secara keseluruhan setiap jenis gerakan dikenali dengan *recognition rate* diatas 85%, kecuali untuk gerakan tangan lurus dan berayun yang memiliki *recognition rate* yang kecil, yakni hanya sebesar 56.59%. Hal ini mungkin disebabkan gerakan tangan lurus dan berayun yang memiliki bentuk data yang cukup kompleks yang cukup susah untuk dikenali oleh sistem. Sedangkan untuk gerakan lainnya memiliki struktur data yang cukup memiliki pola yang lebih gampang dikenali oleh sistem.

1. Pengujian Akurasi Pengenalan Gerakan Manusia dengan Menggunakan *Artificial Neural Network* dan *Hidden* Markov Model

Tabel 4.2 Akurasi Sistem Menggunakan ANN dan HMM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nama Gerakan** | **Jumlah Data Testing** | **Jumlah Data yang Dikenali** | ***Recognition rate* (%)** |
| 1 | Putar kanan | 127 | 127 | 100 |
| 2 | Putar kiri | 80 | 80 | 100 |
| 3 | Dorong | 79 | 79 | 100 |
| 4 | Tarik | 77 | 77 | 100 |
| 5 | Tangan lurus dan berayun | 744 | 744 | 100 |
| 6 | Tangan ditekuk dan berayun | 599 | 599 | 100 |
|  |  |  |  | **100** |

Untuk metode gabungan, sistem berupa gabungan dari dua algoritma: ANN dan HMM. Pada awalnya, data dari sensor yang berupa 10 data *yaw*, 10 data *pitch*, dan 10 data *roll* dimasukkan kedalam sistem dengan algoritma ANN untuk diproses, dengan keluaran dari sistem ANN ini berupa *state* kondisi yang mungkin. Lalu dari *state* tersebut, menjadi masukkan bagi sistem HMM untuk diproses *loglikelihood* dari masing-masing *state* dengan menghasilkan keluaran berupa nilai-nilai *loglikelihood* yang nantinya akan diproses oleh sistem untuk dicari nilai maksimumnya dengan keluaran akhir dari sistem ini berupa jenis gerakan yang dikenali. Dari segi hasil, seperti yang dapat dilihat pada Tabel 4.2, *recognition rate* dari sistem gabungan ini sangat memuaskan, yakni 100%, benar untuk semua data pengujian.

Dari segi hasil pengujian, dapat dilihat bahwa pengenalan yang hanya menggunakan satu algoritma saja, misalnya HMM saja, hasil yang didapatkan sudah cukup baik, hampir semua gerakan dapat dikenali dengan dengan baik, dengan *recognition rate* rata-rata mencapai 88%. Akan tetapi, untuk gerakan tangan lurus dan berayun, hasil yang didapatkan kurang memuaskan, hanya sekitar 56% data yang berhasil dikenali. Sedangkan untuk hasil yang lebih baik, didapatkan ketika diaplikasikan dua algoritma secara bersamaan, ANN dan HMM, hasil pengenalan terhadap keenam gerakan yang dilakukan oleh pengguna mencapai 100%, yakni tidak ada gerakan yang salah dikenali oleh sistem.

1. Perbandingan Waktu Pelatihan dan Waktu Pengujian

Tabel 4.3 Waktu Pelatihan dan Waktu Pengujian Sistem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Waktu Pelatihan (menit)** | | **Waktu Pengujian (second)** | |
| **HMM** | **ANN+HMM** | **HMM** | **ANN+HMM** |
| **Rata-rata** | 39.4000 | 86.1000 | 0.0047 | 0.0056 |
| **Standar Deviasi** | 4.0792 | 7.2173 | 0.0005 | 0.0005 |

Dari segi waktu pelatihan, seperti yang dapat dilihat pada Tabel 4.3, program dengan algoritma HMM saja memiliki waktu yang lebih cepat dibandingkan dengan pelatihan pada sistem dengan algoritma ANN dan HMM. Perbedaannya adalah antara sekitar 39 menit dengan 86 menit, dengan demikiaan, dapat dilihat bahwa sistem dengan HMM saja memiliki waktu pelatihan 118% lebih cepat dibanding sistem ANN dan HMM dalam hal waktu pelatihan. Adanya perbedaan waktu pelatihan ini disebabkan proses yang dijalani berbeda, pada program dengan ANN dan HMM proses pelatihan yang dilakukan lebih banyak dibandingkan program yang hanya menggunakan HMM.

Sedangkan dari segi waktu pengujian sistem, program yang menggunakan HMM saja maupun program dengan algoritma ANN dan HMM memiliki waktu pengujian yang tidak jauh berbeda, yakni 0.0047 detik untuk program dengan algoritma HMM saja dan 0.0056 detik untuk program dengan algoritma ANN dan HMM. Dari waktu tersebut, dapat disimpulkan bahwa waktu pengujian sistem HMM lebih cepat 19% dibanding pengujian sistem ANN digabung dengan HMM. Tidak adanya perbedaan yang terlalu besar ini karena pada fase pengenalan proses yang dijalani oleh sistem tidak memiliki banyak iterasi, sehingga perbedaan waktu pengenalan antara kedua program sangat kecil.

1. KESIMPULAN
2. Alat/sistem yang mampu mengenali enam gerakan manusia (tangan memutar kanan, tangan memutar kiri, tangan mendorong, tangan ditarik, tangan lurus dan berayun, serta tangan ditekuk dan berayun) berhasil diimplementasikan.
3. Algoritma HMM berhasil digunakan untuk mengenali gerakan manusia.
4. Tingkat akurasi dari sistem dengan algoritma HMM adalah sebesar 88.64%.
5. Tingkat akurasi dari sistem dengan algoritma ANN dan HMM adalah sebesar 100%.
6. Waktu pelatihan untuk sistem dengan algoritma HMM adalah sebesar 39.4 menit, sedangkan untuk sistem dengan algoritma ANN dan HMM sebesar 86.1 menit (waktu pelatihan HMM lebih cepat 118% dibandingkan waktu pelatihan sistem dengan ANN dan HMM).
7. Waktu pengujian untuk sistem dengan algoritma HMM adalah sebesar 0.0047 detik, sedangkan untuk sistem dengan algoritma ANN dan HMM sebesar 0.0056 detik (waktu pengujian HMM lebih cepat 19% dibandingkan waktu pengujian sistem dengan ANN dan HMM).

DAFTAR REFERENSI

|  |  |
| --- | --- |
| [1] | M. Eko, Mei 2014. [Online]. Available: http://informatika.stei.itb.ac.id /~rinaldi.munir/Probstat/2010-2011/Makalah2010/MakalahProbstat2010-025.pdf. |
| [2] | L. Baum and J. Egon, "An Inequality with Application to Statistic Estimation for Probabilistic Function of Markov Process and To a Model for Ecology," *Bull. Amer. Meteorol,* vol. 73, pp. 360-363, 1967. |
| [3] | HMM, Mei 2014. [Online]. Available: https://controls.engin.umich.edu /wiki/index.php/File:MarkovM.JPG. |
| [4] | [Online]. Available: http://www.shokhirev.com/nikolai/abc/alg/hmm /hmm.html. |
| [5] | L. Rabiner, "First Hand: The Hidden Markov Model," *IEEE Global History Network.* |
| [6] | S. 9-DOF\_IMU Electronics, Mei 2014. [Online]. Available: https://www.sparkfun.com/products/10736. |
| [7] | S. Gyroscope-ITG3200 Electronics, Mei 2014. [Online]. Available: https//www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf. |
| [8] | S. Accelerometer-ADXL345 Electronics, Mei 2014. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf. |
| [9] | S. Magnetometer-HMC5883L Electronics, Mei 2014. [Online]. Available: http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Magneto/HMC5883L-FDS.pdf. |
| [10] | X. D. International, Mei 2014. [Online]. Available: http://ftp1.digi.com /support/documentation/90000982\_H.pdf. |
| [11] | Mathworks, "Mathworks," [Online]. Available: http://www.mathworks.com /products/matlab/. [Accessed December 2014]. |
| [12] | J. P. Varkey, D. Pompili and T. A. Walls, "Human Motion Recognition Using A Wireless Sensor-Based Wearable System," *Pers Ubiquit Comput,* September 2011. |
| [13] | H. H. Joshi, N. Shah and P. Raval, "Automated Video Surveillance System For Human Motion Detection," *Journal Of Information, Knowledge And Research In Electronics And Communication Engineering,* vol. 02, no. 02, pp. 675-678, October 2013. |
|  |  |