****

**UNIVERSITAS INDONESIA**

**SISTEM PENGENALAN GERAKAN MANUSIA MENGGUNAKAN *HIDDEN MARKOV MODEL***

**SEMINAR**

**HANDISON JAYA**

**1106068913**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

**DEPARTEMEN TEKNIK ELETKRO**

**TEKNIK KOMPUTER**

**DEPOK**

**JUNI**

**2014**

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa sebab atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan seminar ini dengan baik. Penulis menyadari bahwa seminar 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 seminar yang telah memberikan pengarahan, koreksi, dukungan, waktu, dan akomodasi alat selama penulis mengerjakan seminar ini.
2. Orang tua dan keluarga penulis yang telah memberikan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan seminar ini.
3. Teman riset seperjuangan Tryan Aditya Putra dalam membangun sistem pengenalan gerakan, yang banyak memberikan bantuan dan masukkan dalam seminar ini.
4. Tim Robotika UI Annisaa Primadini, Lintang, Ismi, Alif, Zaka atas dukungan dan bantuannya saat pengerjaan seminar ini.
5. Rekan-rekan dari laboratorium Digital Edo, Dwi, Antoni, Puput, Ibam.
6. Teman-teman Teknik Komputer 2011 Kautsar, Yudha, Dinar, Sean, Emily, Yessy, Zhafir dan Keluarga Departemen Teknik Elektro lain.

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

Depok, Juni 2014

Penulis

ABSTRAK

|  |  |  |
| --- | --- | --- |
| Nama | : | Handison Jaya |
| Program Studi | : | Teknik Komputer |
| Judul | : | Sistem Pengenalan Gerakan Manusia Menggunakan *Hidden Markov Model* |

Sistem pengenalan gerakan manusia terdiri dari dua fase, yaitu fase pelatihan dan fase pengenalan. Fase pelatihan merupakan fase sistem dilatih dengan data dari sensor agar sistem belajar dan tidak melakukan banyak kesalahan. Fase pengenalan merupakan fase sistem digunakan untuk mengenali gerakan manusia dengan hasil pembelajaran ketika fase pelatihan. Adapun gerakan yang bisa dikenali adalah gerakan manusia berjalan, gerakan manusia berlari, gerakan pergelangan tangan memutar kekanan dan kekiri, serta gerakan tangan ditarik mendekati tubuh dan gerakan tangan mendorong menjauhi tubuh. Algoritma yang digunakan antara lain *Artificial Neural Network* (ANN) untuk pembelajaran*, Hidden Markov Model* (HMM) untuk pengenalan*,* dan *Support Vector Machine* (SVM) untuk pengklasifikasian. Akan tetapi pada penelitian ini fokus akan dititik-beratkan pada HMM sebagai algoritma dalam pengenalan gerakan manusia. HMM sendiri merupakan suatu permodelan statistika yang dimana sistem yang dimodelkan diasumsikan merupakan proses Markov yang memiliki *state/*keadaan yang tersembunyi (*hidden*).

Kata kunci: *Artificial Neural Network, Hidden Markov Model*, *Support Vector Machine*, Proses Markov

ABSTRACT

|  |  |  |
| --- | --- | --- |
| Name | : | Handison Jaya |
| Major | : | Teknik Komputer |
| Title | : | Human Motion Recognition Using *Hidden Markov Model* |

Human motion recognition consist of two phase, which is training phase and recognition phase. Training phase is phase where system is trained with set of data from sensors so that system can learn and update its system. Whereas recognition phase is phase where system is ready to use to recognize human motion with learning result from training phase. As for motion, sytem can recognize six motion, which is walking, running, hand twisting left, hand twisting right, hand push forward, and hand pull inward. As for algorithm, system will be using *Artificial Neural Network* (ANN) for learning, *Hidden Markov Model* (HMM) for recognition, and *Support Vector Machine* (SVM) for classification. This research will focusing on Hidden Markov Model as algorithm for recognizing pattern of human motion. HMM itself is a statistical model where the system which being modelled was assumed as Markov process that has *hidden* state.

Keywords: *Artificial Neural Network, Hidden Markov Model*, *Support Vector Machine*, Markov Process

DAFTAR ISI

[KATA PENGANTAR ii](#_Toc390979499)

[ABSTRAK iii](#_Toc390979500)

[ABSTRACT iv](#_Toc390979501)

[DAFTAR ISI v](#_Toc390979502)

[DAFTAR GAMBAR vii](#_Toc390979503)

[BAB I PENDAHULUAN 1](#_Toc390979504)

[1.1 Latar Belakang 1](#_Toc390979505)

[1.2 Tujuan Penelitian 2](#_Toc390979506)

[1.3 Batasan Masalah 2](#_Toc390979507)

[1.4 Metode Penelitian 3](#_Toc390979508)

[1.5 Sistematika Penulisan 3](#_Toc390979509)

[BAB II HIDDEN MARKOV MODEL DALAM SISTEM PENGENALAN GERAKAN MANUSIA 5](#_Toc390979510)

[2.1 Pengenalan dengan Hidden Markov Model 5](#_Toc390979511)

[2.1.1 Markov Model 5](#_Toc390979512)

[2.1.2 Hidden Markov Model 7](#_Toc390979513)

[2.2 Hardware 16](#_Toc390979514)

[2.2.1 Arduino 16](#_Toc390979515)

[2.2.2 IMU 17](#_Toc390979516)

[2.2.3 Zigbee dan Xbee Wireless 20](#_Toc390979517)

[2.3 Perbandingan dengan Penelitian Lainnya 23](#_Toc390979518)

[BAB III PERANCANGAN SISTEM UNTUK PENGENALAN GERAKAN MANUSIA 25](#_Toc390979520)

[3.1 System Requirement 25](#_Toc390979521)

[3.2 System and Software Design 27](#_Toc390979522)

[3.2.1 Desain Perangkat Keras (*Hardware*) 29](#_Toc390979523)

[3.2.2 Algoritma Perangkat Lunak (*Software*) 30](#_Toc390979524)

[BAB IV KESIMPULAN 36](#_Toc390979525)

[DAFTAR PUSTAKA 37](#_Toc390979526)

DAFTAR GAMBAR

[Gambar 2.1 *Markov Chain* yang menggambarkan perubahan cuaca [**1**] 6](#_Toc390909967)

[Gambar 2.2 Representasi Parameter HMM [**1**] 8](#_Toc390909968)

[Gambar 2.3 Contoh HMM [**3**] 9](#_Toc390909969)

[Gambar 2.4 Modul Mikrokontroler Arduino UNO [**6**] 17](#_Toc390909970)

[Gambar 2.5 Sensor Razor IMU [**7**] 17](#_Toc390909971)

[Gambar 2.6 Arah Orientasi dan Polaritas Rotasi [**8**] 18](#_Toc390909972)

[Gambar 2.7 Blok Diagram Fungsional ITG-3200 [**8**] 18](#_Toc390909973)

[Gambar 2.8 Orientasi Sensitivitas ADXL345 [**9**] 19](#_Toc390909974)

[Gambar 2.9 Blok Diagram Fungsional ADXL345 [**9**] 19](#_Toc390909975)

[Gambar 2.10 Periode Pengambilan Data Magnetometer [**10**] 20](#_Toc390909976)

[Gambar 2.11 Xbee Chip Antenna Series 2 [**11**] 21](#_Toc390909977)

[Gambar 2.12 *Interface* X-CTU [**11**] 22](#_Toc390909978)

Gambar 3.1 *Use Case* Diagram 26

Gambar 3.2 Blok Diagram Sistem Pengenalan Gerakan Manusia 27

Gambar 3.3 *Sequence* Diagram 28

Gambar 3.4 Blok *Hardware* Sistem 29

Gambar 3.5 Activity Diagram Ketika Pelatihan 31

Gambar 3.6 Fase Pelatihan 32

Gambar 3.7 Activity Diagram Ketika Pengenalan 33

Gambar 3.8 Fase Pengenalan 34

# PENDAHULUAN

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

## 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. 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 webcam serta processor yang memiliki kemampuan komputasi yang kuat. Sedangkan jika meggunakan sensor percepatan 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).

## Tujuan Penelitian

* Mampu membuat alat/sistem yang mampu mengenali gerakan manusia.
* Menganalisis tingkat reliabilitas dan tingkat keakuratan alat/sistem yang telah dibuat dalam mengenali gerakan manusia.
* Menganalisis algoritma-algoritma yang digunakan pada Hidden Markov Model dalam mengambil kesimpulan.

## Batasan Masalah

Sistem dirancang untuk mengenali 6 jenis gerakan, antara lain gerakan jalan dan lari manusia, 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 6, yakni 3-*axis accelerrometer* dan3-*axis gyroscope,* hal ini karena pengaruh *magnetometer* masih belum menjadi pertimbangan utama dalam mengenali gerakan manusia. Pada penelitian ini, CPU utama yang digunakan untuk melakukan pemrosesan dan pengenalan gerakan manusia adalah *Arduino*. 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.

## Metode Penelitian

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

1. *System Requirement*: menentukan spesifikasi sistem dengan melakukan studi literatur dan pengamatan pada aplikasi yang sudah ada.
2. *System and Software Design*: merancang simulasi algoritma *Hidden Markov Model*.
3. *Implementation*: mengimplementasikan rancangan simulasi, yaitu dengan membuat alat yang diintegrasikan dengan sensor.
4. *Testing*: melakukan pengujian terhadap performa alat.
5. Analisis: analisis hasil pengujian sistem.

## 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.

# 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 Arduino, IMU, serta Xbee wireless.

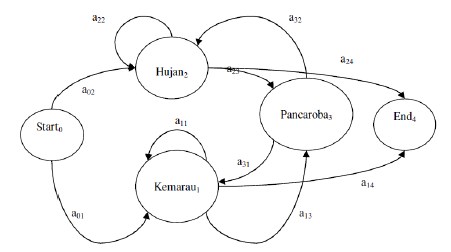
## Pengenalan dengan Hidden Markov Model

### 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](#Muh14)]

*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](#Muh14)]

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](#Muh14)]



Gambar 2.1 *Markov Chain* yang menggambarkan perubahan cuaca [[1](#Muh14)]

Pada Gambar 2.1, 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:

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

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:

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

Selain itu juga berlaku batasan, yakni:

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

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*.

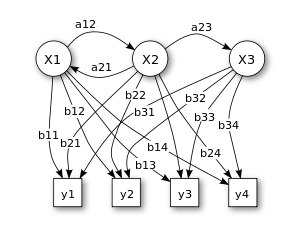
### Hidden Markov Model

*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](#LEB67)].

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/HMM. [[3](#HMM14)]

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 2.2 Representasi Parameter HMM [[1](#Muh14)]

Dari Gambar 2.2, 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:

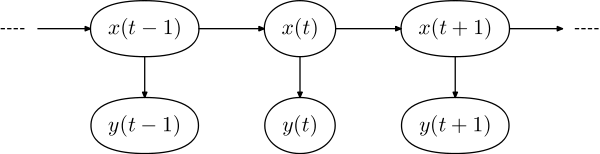
1. Evaluasi

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](#Alg)]

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 sudah dijelaskan sebelumnya.

Asumsikan terdapat HMM seperti berikut:



Gambar 2.3 Contoh HMM [[3](#HMM14)]

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.

Dengan menggunakan aturan rantai, didapatkan bahwa:

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

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

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

#### Algoritma Backward [[4](#Alg)]

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:

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.

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

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](#Alg)]

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:

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 .

|  |  |
| --- | --- |
|  | (2.6) |
|  | (2.7) |

Kompleksitas dari algoritma ini adalah .

1. *Posterior Decoding* [[4](#Alg)]

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:

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

*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](#Alg)]

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](#Law)].

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:

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

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:

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

1. Backward

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

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

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 :

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

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 :

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

Yang merupakan ekspektasi frekuensiyang 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.

## Hardware

### Arduino

Arduino merupakan suatu mikrokontroler yang sudah terintegrasi (*single-board microcontroller*), sehingga bisa langsung digunakan untuk aplikasi mikrokontroler. Selain itu, Arduino juga merupakan perangkat keras yang mempunyai *platform* yang *open-*source. Dari segi hardware, Arduino terdiri dari mikrokontroler 8-bit Atmel AVR atau mikrokontroler 32-bit Atmel ARM.

Dari sisi software, Arduino memiliki IDE (*Integrated Development Environment*) sendiri. Cara pemrogramannya sendiri merupakan implementasi dari sebuah *wiring* (*framework* pemrograman *open-source* untuk mikrokontroler) yang mempunyai kesamaan dengan pemrograman *platform* perangkat fisik komputer yang didasari dengan *environment* pemrosesan pemrograman multimedia [[6](#Ard141)].

Mikrokontroler Arduino memiliki beberapa kelebihan dibandingkan dengan mikrokontroler yang lain, diantaranya:

* Relatif murah. Arduino relatif lebih muarh dibandingkan dengan *platform* mikrokontroler lain.
* Lintas *platform*. Software mikrokontroler Arduino dapat bekerja pada sistem operasi *Windows, Macintosh OSX,* dan *Linux*. Sedangkan kebanyakan sistem mikrokontroler lain hanya dapat bekerja pada sistem *Windows* atau *Linux* saja.
* Sederhana. Pemrograman Arduino sangat mudah untuk digunakan, bahkan untuk pemula dan cukup fleksibel untuk para profesional.
* Perangkat lunak *open-source* dan *extensible.* Bahasa pemrograman yang digunakan adalah bahasa C yang sudah disesuaikan. Sehingga dapat dengan mudah diadaptasi oleh orang yang sudah terbiasa dalam melakukan pemrograman mikrokontroler.
* Perangkat keras *open-source* dan *extensible*. Dasar dari mikrokontroler Arduino adalah ATmega8 dan ATmega168. Sehingga para desainer sirkuit dapat membuat modul mikrokontroler Arduino sendiri berdasarkan skematik yang disediakan.



Gambar 2.4 Modul Mikrokontroler Arduino UNO [[7](#Ard14)]

### 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*.

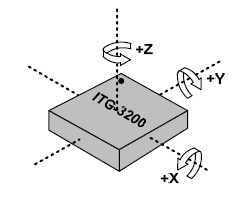


Gambar 2.5 Sensor Razor IMU [[8](#Spa14)]

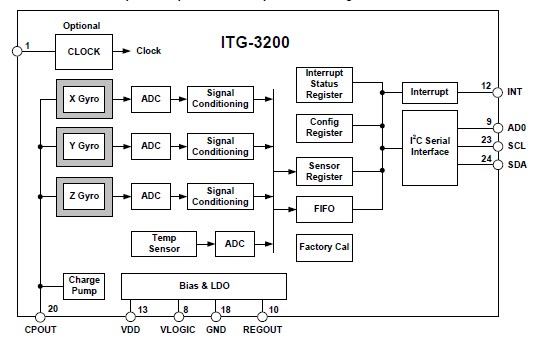
Adapun spesifikasi teknis yang dimiliki *Razor IMU 9-Degrees of Freedom* antara lain [[8](#Spa14)]:

* 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, pitch, roll*) hingga ±2000º/detik. Data juga bisa diukur secara digital 16-bit dengan menggunakan ADC yang sudah tersedia, dan proses transmisi data dapat dilakukan melalui *interface* I2C.



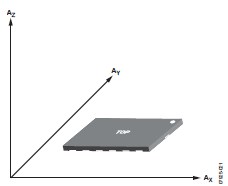
Gambar 2.6 Arah Orientasi dan Polaritas Rotasi [[9](#Spa141)]



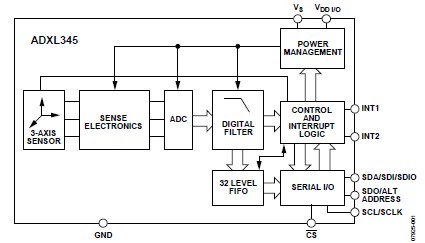
Gambar 2.7 Blok Diagram Fungsional ITG-3200 [[9](#Spa141)]

* 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.

**

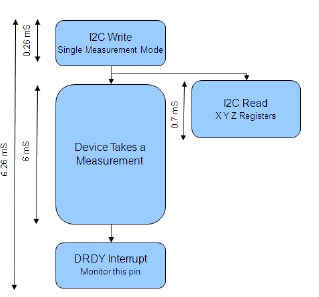
Gambar 2.8 Orientasi Sensitivitas ADXL345 [[10](#Spa142)]



Gambar 2.9 Blok Diagram Fungsional ADXL345 [[10](#Spa142)]

* 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.



Gambar 2.10 Periode Pengambilan Data Magnetometer [[11](#Spa143)]

* 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.

### 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 [[12](#Dig14)].

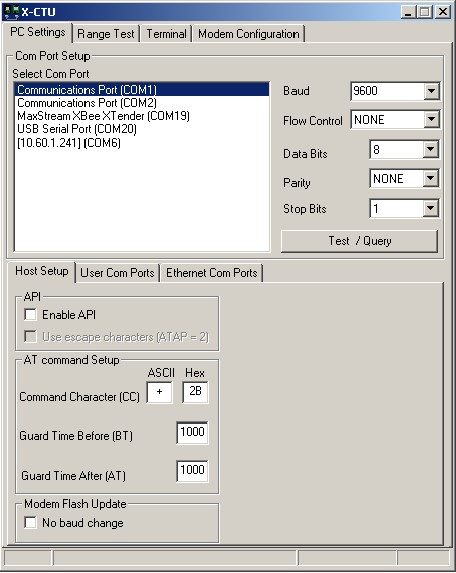


Gambar 2.11 Xbee Chip Antenna Series 2 [[12](#Dig14)]

Adapun spesifikasi teknis yang dimiliki oleh Xbee Chip Antenna Series 2 antara lain [[12](#Dig14)]:

* 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 2.12 *Interface* X-CTU [[12](#Dig14)]

Dari Gambar 2.12 diatas, 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. [[12](#Dig14)]

## Perbandingan dengan Penelitian Lainnya

* *Human Motion Recognition Using A Wireless Sensor-Based Wearable System*(2011)[[13](#Joh11)]

*Author*: John Paul Varkey, Dario Pompilii, Theodore A.Walls

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) [[14](#Har13)]

*Author*: Hardik H. Joshi, N.N. Shah, P.D. Raval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | - | Berjalan, berlari, 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) |

# 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.

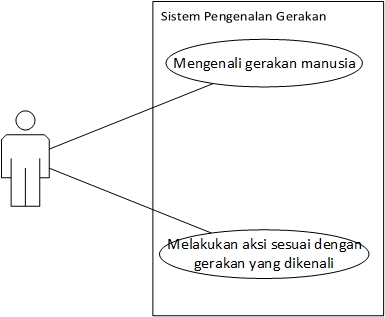
## System Requirement

*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 (indikator berupa LED) yang sesuai untuk setiap gerakan yang dikenali. Adapun gerakan-gerakan yang akan dikenali antara lain:

1. Gerakan jalan manusia. Gerakan jalan dikenali ketika dari keadaan diam terjadi 2 langkah tangan berayun, sebagaimana normalnya manusia berjalan.
2. Gerakan lari manusia. Gerakan lari dikenali ketika dari keadaan diam terjadi 3 langkah tangan berayun, sebagaimana normalnya manusia berlari.
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.



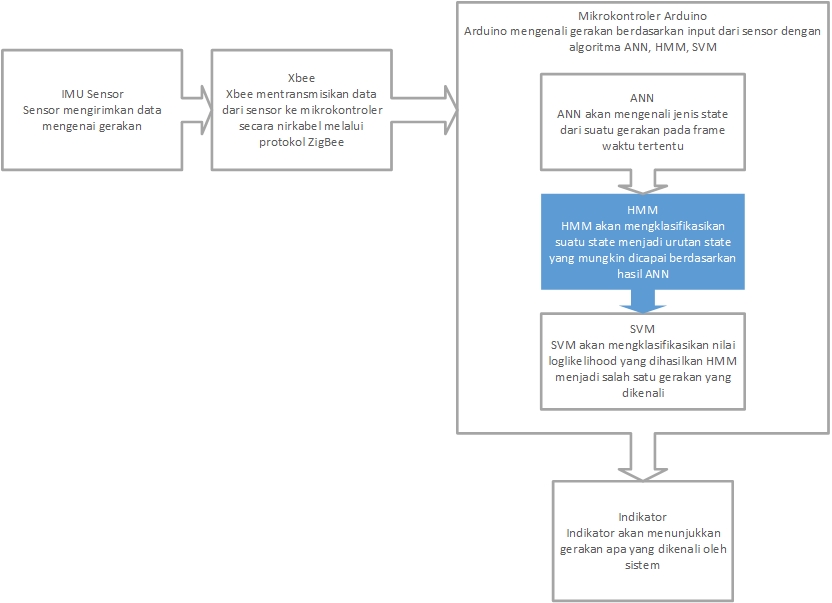
Gambar 3.1 *Use Case* Diagram

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.

## System and Software Design

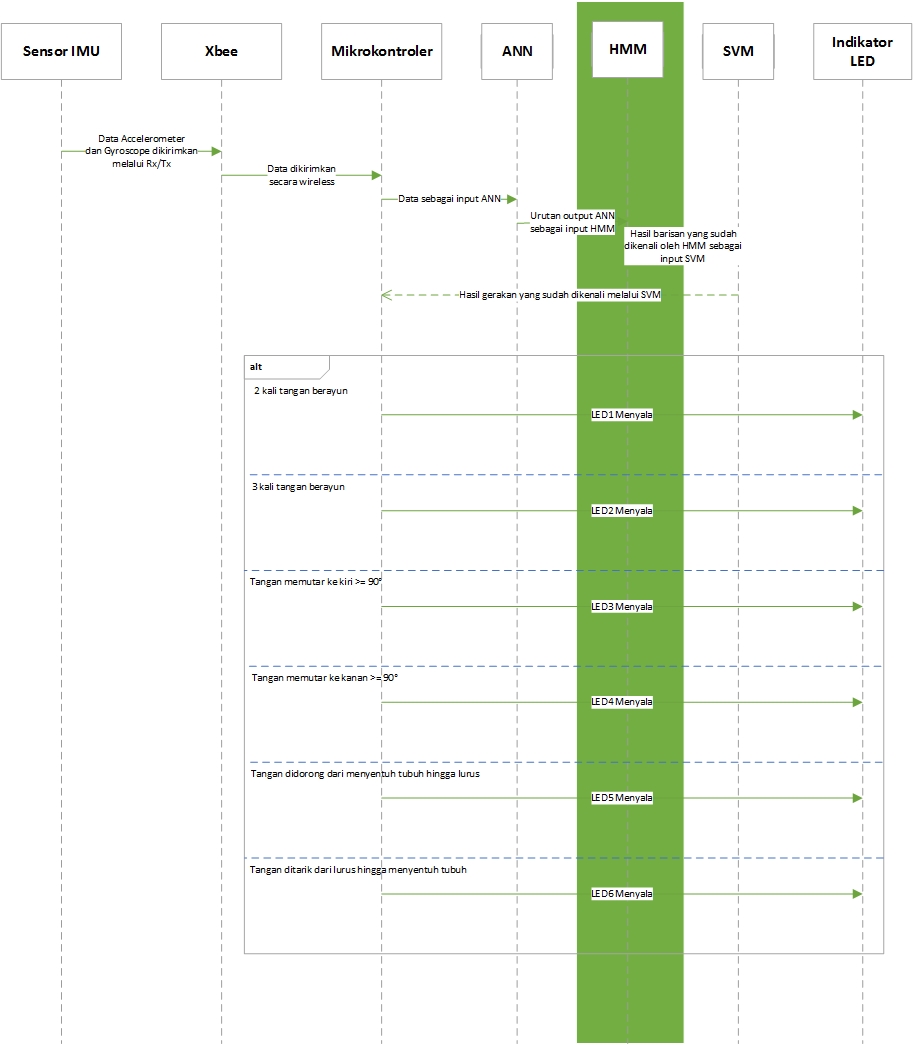
Sistem pengenalan gerakan manusia yang dibuat dalam penelitian ini dirancang untuk memberikan solusi yang cepat dan akurat dalam mengenali gerakan pengguna yang spesifik yang nantinya dapat berguna dalam berbagai bidang lainnya.Sistem ini akan dikendalikan oleh mikrokontroler Arduino. Sedangkan untuk 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



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. Pada Gambar 3.2, cakupan penelitian yang dilakukan ditunjukkan oleh bagian yang berwarna biru.



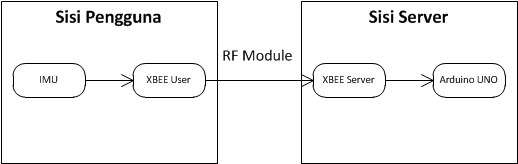
Gambar 3.3 *Sequence* Diagram

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 mikrokontroler untuk diproses. Adapun proses pengolahan data yang terjadi di dalam mikrokontroler adalah pemrosesan data dari sensor sebagai input bagi *Artificial Neural Network* (ANN), kemudian hasil pengenalan state 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 yang berbeda.

### 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 titik 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

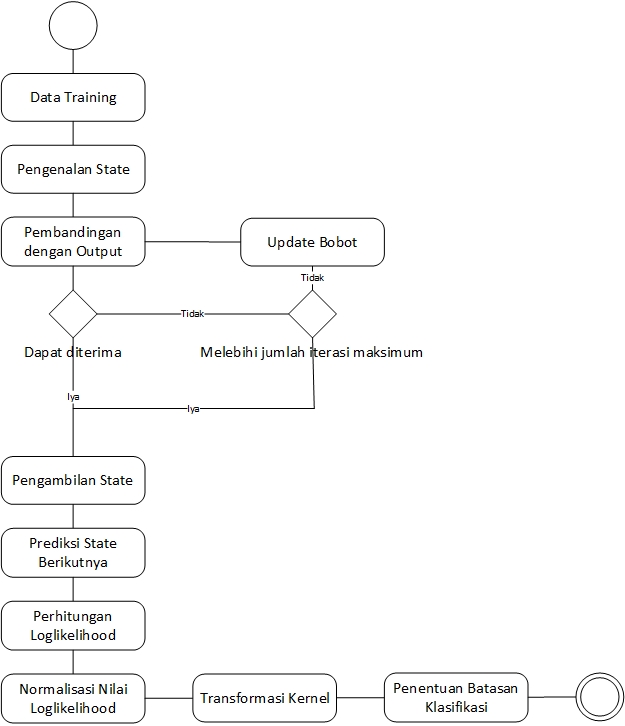
### Algoritma Perangkat Lunak (*Software*)

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 jalan dan lari manusia, 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.

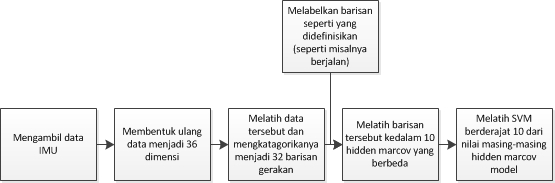
Sedangkan pada fase pengenalan merupakan fase dimana sistem sudah melewati fase pelatihan. Sehingga sistem dapat dianggap telah dapat digunakan untuk mengenali gerakan-gerakan berbeda yang dilakukan oleh pengguna.

1. Fase Pelatihan



Gambar 3.5 Activity Diagram Ketika Pelatihan

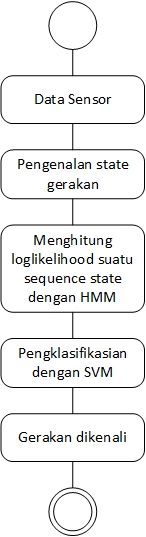
Dari sisi pengguna, diagram aktifitas ketika pelatihan dilakukan terhadap sistem dapat dilihat seperti pada Gambar 3.5 diatas. Fase pelatihan merupakan fase dimana sistem masih akan belajar untuk mengenali gerakan tertentu secara akurat. Oleh karena itu pengguna akan melakukan suatu gerakan yang sama untuk beberapa kali yang nantinya akan digunakan sebagai set data *training* untuk dikenali *state*-nya lalu 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. Jika dapat diterima maka sistem akan dilanjutkan ke tahap berikutnya yakni pengenalan dan prediksi dari *state* berikutnya untuk membentuk suatu urutan *state*. Berdasarkan nilai *loglikelihood* yang sudah ternormalisasi maka akan dilakukan transformasi kernel sebelum terakhir untuk ditentukan batasan klasifikasi dari sistem.



Gambar 3.6 Fase Pelatihan

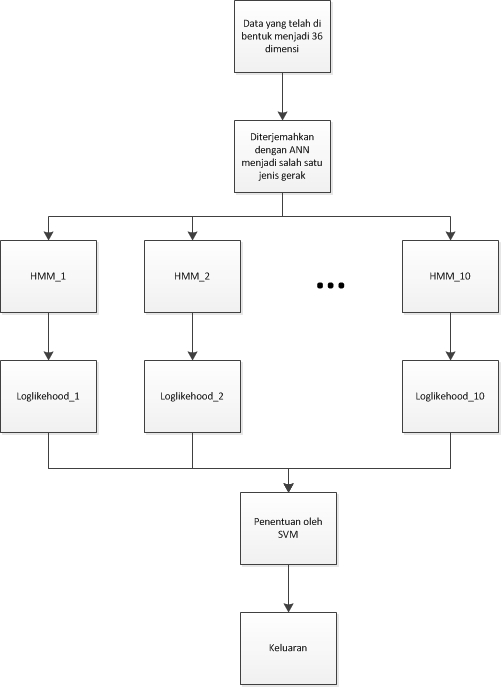
Sedangkan dari sisi yang lebih mendetail, pada fase pelatihan, seperti pada Gambar 3.6, data akan diambil oleh sensor IMU untuk setiap satuan waktu. Data dari sensor IMU yang aka dipakai adalah data percepatan sudut (*gyro*) 3-axis dan data percepatan linier (*accelero*) 3-axis, sehingga untuk setiap satuan waktunya terdapat 6 data yang akan diambil. Untuk masing-masing gerakan digunakan 6 *frame* waktu yang masing-masing *frame* waktunya terdiri dari 6 data yang dipakai, dengan demikian akan terbentuk data dengan 36 dimensi yang akan digunakan sebagai input ANN, dengan output yang akan dihasilkan berdimensi 5 (5-bit). Bagian berikutnya adalah dari hasil pengolahan ANN ini akan dibentuk/dikategorikan menjadi 32 barisan gerakan yang mungkin. Lalu dari barisan suku-suku yang sebelumnya telah ditentukan oleh ANN dilakukan pelatihan terhadap *Hidden Markov Model* yang telah dibuat. Dalam hal ini akan dibuat 10 jenis HMM berbeda yang akan digunakan untuk melatih barisan yang sama. Kemudian masing-masing HMM tersebut akan memiliki nilai loglikelihood yang berbeda yang dimana nilai loglikelihood ini akan dijadikan sebagai dimensi dari SVM. Kemudian dari 10 dimensi HMM tersebut program SVM dilatih.

1. Fase Pengenalan



Gambar 3.7 Activity Diagram Ketika Pengenalan

Dari sisi pengguna, diagram aktifitas ketika sistem melakukan proses pengenalan dapat dilihat seperti pada Gambar 3.7 diatas. 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 ANN sebagai suatu *state* gerakan tertentu, kemudian menghitung nilai *loglikelihood* suatu urutan *state* yang ada dengan HMM. Setelah itu akan diklasifikasikan dengan menggunakan SVM sehingga akan menghasilkan output salah satu dari gerakan yang sudah dikenali.



Gambar 3.8 Fase Pengenalan

Dari sisi yang lebih mendetail, fase pengenalan seperti pada Gambar 3.8, 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* dan 3-axis *accelerometer* selama 6 potongan waktu. Sehingga dari 36 data tersebut, akan dijadikan sebagai dimensi bagi input ANN. Kemudian ouput dari ANN akan menjadi suku-suku yang dimana suku-suku tersebut dinamakan kondisi/*state* pada HMM. Proses berikutnya adalah proses kalkulasi nilai loglikelihood dari setiap kemungkinan urutan *state* yang ada dengan menggunakan 10 arsitektur HMM yang berbeda. Berdasarkan hasil dari ke-10 nilai loglikelihood tersebut, sistem akan mengklasifikasikannya dengan menggunakan SVM yang dari hasil SVM ini kemudian ditentukan gerakan apa yang dikenali oleh sistem dari input gerakan oleh pengguna.

# KESIMPULAN

1. Alat/sistem yang mampu mengenali gerakan manusia dapat diimplementasikan.
2. Secara teoritis, tingkat reliabilitas dan tingkat keakuratan alat/sistem yang dibuat dalam mengenali gerakan manusia cukup baik.
3. Algoritma *Hidden Markov Model* yang digunakan pada sistem mencakup tahapan evaluasi (algoritma *Forward* dan *Backward*), penarikan kesimpulan (algoritma *Viterbi* dan *Posterior decoding*), dan pembelajaran (algoritma *Baum-Welch*).

DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | M. Eko. (2014) Teori Dasar Hidden Markov Model. [Online]. <http://informatika.stei.itb.ac.id/~rinaldi.munir/Probstat/2010-2011/Makalah2010/MakalahProbstat2010-025.pdf> |
| [2] | J. A. E. L.E. Baum, "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. (2014) Hidden Markov Model Example. [Online]. <https://controls.engin.umich.edu/wiki/index.php/File:MarkovM.JPG> |
| [4] | Algorithms: Hidden Markov Model. [Online]. <http://www.shokhirev.com/nikolai/abc/alg/hmm/hmm.html> |
| [5] | L. Rabiner, "First Hand: The Hidden Markov Model," *IEEE Global History Network*. |
| [6] | A. IDE. (2014) Arduino Development Environment. [Online]. <http://www.arduino.cc/en/Guide/Environment> |
| [7] | A. UNO. (2014) Arduino UNO: Overview, Power, Memory, Input, Output, Communication, and Programming. [Online]. <http://arduino.cc/en/Main/ArduinoBoardUno> |
| [8] | S. 9-DOF\_IMU Electronics. (2014) SparkFun. [Online]. <https://www.sparkfun.com/products/10736> |
| [9] | S. Gyroscope-ITG3200 Electronics. (2014) SparkFun. [Online]. <https//www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf> |
| [10] | S. Accelerometer-ADXL345 Electronics. (2014) SparkFun. [Online]. <https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf> |
| [11] | S. Magnetometer-HMC5883L Electronics. (2014) SparkFun. [Online]. <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Magneto/HMC5883L-FDS.pdf> |
| [12] | X. D. International. (2014) Xbee/Xbee-PRO RF Modules. [Online]. <http://ftp1.digi.com/support/documentation/90000982_H.pdf> |
| [13] | John Paul Varkey, "Human Motion Recognition Using A Wireless Sensor-Based Wearable System," *Pers Ubiquit Comput*, Sep. 2011. |
| [14] | Hardik H Joshi, "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, Oct. 2013. |