# BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai dasar teori yang menjadi dasar pembuatan Tugas Akhir ini. Beberapa hal yang dibahas adalah penjelasan tentang tunarungu, bahasa isyarat secara umum, Kinect 2.0 dan SDK-nya, serta decicion tree.



## 2.1 Tunarungu

Tuna rungu dapat diartikan sebagai keterbatasan yang dimiliki seseorang dalam mendengar sesuatu karena tidak berfungsinya organ pendengaran yang dimilikinya. Ketunarunguan dapat dibedakan menjadi dua kategori yaitu tuli (*deaf*) dan kurang dapat mendengat (*low hearing*) [2].Tuli adalah keadaan dimana organ pendengaran telah mengalami kerusakan yang sangat parah dan mengakibatkan tidak berfungsinya pendengaran. Sedangkan kurang dapat mendengar adalah keadaan dimana organ pendengaran mengalami kerusakan tetapi masih dapat berfungsi untuk mendengar.

## 2.2 Bahasa Isyarat

Bahasa isyarat adalah bahasa yang digunakan untuk berkomunikasi dengan penyandang tunarungu. Bahasa isyarat yang sering digunakan di Indonesia berdasarkan pada Sistem Isyarat Bahasa Indonesia (SIBI). Ada 4 jenis bahasa isyarat dalam SIBI [3], yaitu:

1. Isyarat Pokok: melambangkan sebuah kata atau konsep;

2. Isyarat Tambahan: melambangkan awalan, akhiran, dan

partikel (imbuhan);

3. Isyarat Bentukan: dibentuk dengan menggabungkan

isyarat pokok dan isyarat tambahan;

4. Abjad Jari: dibentuk dengan jari-jari untuk mengeja huruf / angka.

Pada Tugas Akhir ini penulis menggunakan 20 bahasa isyarat pokok yang terdiri dari 10 bahasa isyarat statis dan 10 bahasa isyarat dinamis. Seluruh 20 gerakan bahasa isyarat ini dapat dilihat pada Gambar ()()().

Pada Tugas Akhir ini terdapat 20 bahasa isyarat pokok yang digunakan seperti yang dapat dilihat pada Gambar 2.1. Bahasa isyarat tersebut kemudian dikelompokkan berdasarkan 10 gerakan dasar, yaitu:

(1) gerakan satu tangan : membentuk garis lurus

(terhadap sumbu X)

(2) gerakan satu tangan : membentuk garis lurus dua arah

(terhadap sumbu X)

(3) gerakan satu tangan : membentuk gelombang

(4) gerakan satu tangan : membentuk lengkungan

(5) gerakan satu tangan : membentuk lingkaran

(6) gerakan dua tangan : membentuk garis lurus

(terhadap sumbu Y)

(7) gerakan dua tangan : membentuk garis lurus dan

kedua tangan saling menyatu

(terhadap sumbu X)

(8) gerakan dua tangan : membentuk garis lurus dan

kedua tangan saling memisah

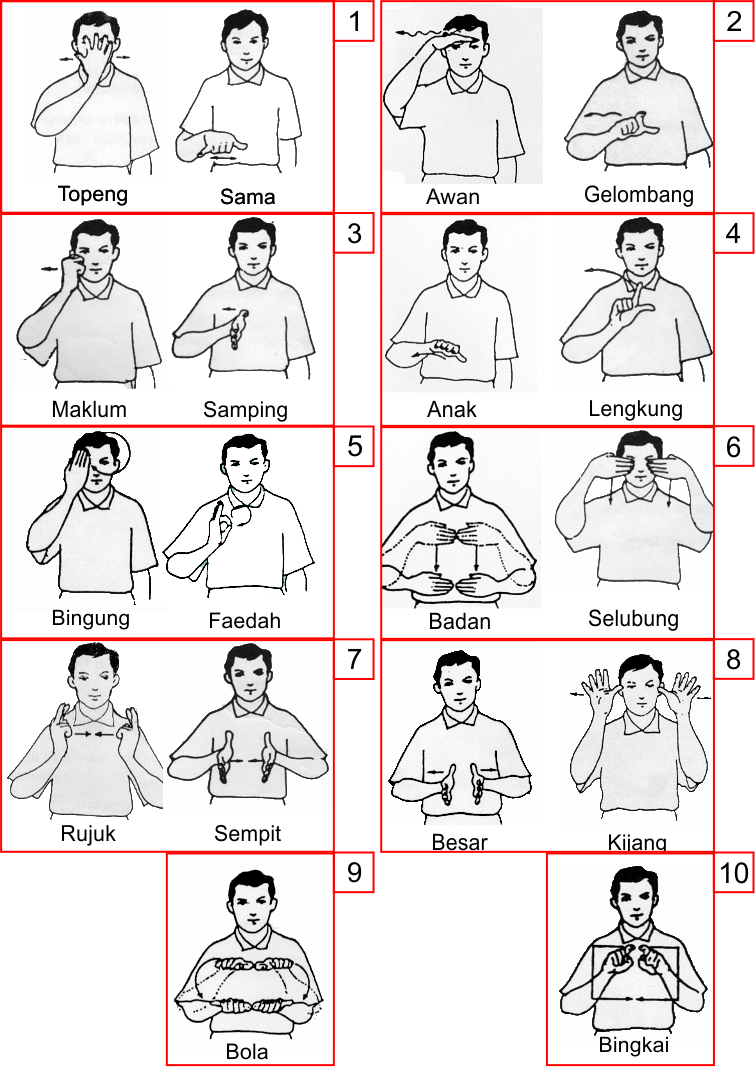
(terhadap sumbu X)

(9) gerakan dua tangan : membentuk lingkaran

(10) gerakan dua tangan : membentuk persegi

## 2.3 Kinect 2.0

Generasi kedua dari Kinect yang dirilis oleh Microsoft pada tahun 2014 adalah versi terbaru Kinect dari yang pertama kali dikeluarkan pada tahun 2010. Perangkat Kinect 2.0 seperti yang terlihat pada Gambar 2.2, terdapat tiga lensa yaitu kamera RGB yang digunakan untuk menangkap spektrum warna, *infrared emitters* yang memproyeksikan spektrum inframerah dan sensor kedalaman yang menghasilkan gambar mendalam dari seseorang atau objek dengan menganalisis informasi inframerah. Dan sebuah *microphone* array yang dapat menemukan lokasi timbulnya suara. Alhasil, ada enam sumber data yang dihasilkan, termasuk warna, inframerah, kedalaman, indeks tubuh, tubuh, dan suara .



Gambar 2.1 Bahasa Isyarat yang Digunakan



Gambar 2.2 Kinect 2.0

## 2.4 Kinect SDK

Kinect SDK adalah pustaka yang dibuat oleh Microsoft untuk pengembangan aplikasi perangkat lunak yang menggunakan Kinect sebagai alat input utama. Kinect SDK dapat diimplementasikan dengan bahasa pemrograman C#, C++, dan JavaScript. Pustaka ini memiliki beberapa fitur diantaranya *skeleton tracking*, *thumb tracking*, *end of hand tracking*, *open/close hand gesture* dan lainnya .

## 2.5 Microsoft Visual Studio

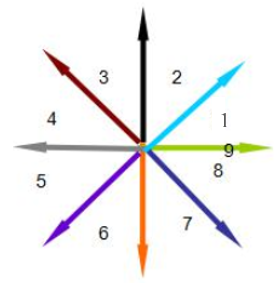
Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan perangkat lunak, baik itu perangkat lunak bisnis, perangkat lunak pribadi, ataupun komponen perangkat lunak nya dalam bentuk perangkat lunak berbasis *console*, Windows, ataupun berbasis *website*.

## 2.6 Ekstraksi Fitur Dinamis dari *Hand Gesture*

Fitur dinamis merupakan fitur gerak atau fitur yang bersifat dinamis yang didapatkan dari deteksi Kinect 2.0 terhadap tangan. Fitur dinamis yang digunakan pada metode ini adalah fitur untuk mengolah gerakan tangan (*hand gesture*).

Sebuah fitur yang baik memainkan peran penting dalam pengenalan. Pada Tugas Akhir yang dibuat ini, terdapat 40 data/*frame* yang akan diproses. Setiap data/*frame* akan menghasilkan koordinat gerak dinamis yang diproyeksikan ke dalam bidang X0Y yang merupakan prinsip dalam bidang. Kemudian dari hasil setiap data/*frame* tersebut, akan dicatat orientasi sudut mutlak  yang dapat dilihat pada persamaan (1.0) (1.1) dan (1.2) . Orientasi sudut mutlak tersebut kemudian diterjemahkan menjadi data kuantisasi hasil ekstraksi fitur dinamis seperti yang dapat dilihat pada Gambar 2.3.

|  |  |
| --- | --- |
|  | (1.0) |
|  | (1.1) |
|  |  |
|  | (1.2) |



(A) (B)

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

Tabel 2.1 Data Hasil Ekstraksi Fitur Dinamis

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

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

## 2.7 Decision Tree

Pohon Keputusan atau dikenal dengan Decision Tree adalah salah satu metode klasifikasi yang menggunakan representasi suatu struktur pohon yang berisi alternatif-alternatif untuk pemecahan suatu masalah. Pohon ini juga menunjukkan faktor-faktor yang mempengaruhi hasil alternatif dari keputusan tersebut disertai dengan estimasi hasil akhir bila kita mengambil keputusan tersebut. Peranan pohon keputusan ini adalah sebagai *Decision Support Tool* untuk membantu manusia dalam mengambil suatu keputusan. Manfaat dari Decision Tree adalah melakukan *break down* proses pengambilan keputusan yang kompleks menjadi lebih mudah sehingga orang yang mengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan. Konsep yang digunakan oleh Decision Tree adalah mengubah data menjadi suatu keputusan pohon dan aturan-aturan keputusan (*rule*).

Decision Tree menggunakan struktur hierarki untuk pembelajaran *supervised*. Proses dari Decision Tree dimulai dari *root node* hingga *leaf node* yang dilakukan secara rekursif. Di mana setiap percabangan menyatakan suatu kondisi yang harus dipenuhi dan pada setiap ujung pohon menyatakan kelas dari suatu data. Pada Decision Tree terdiri dari tiga bagian yaitu:

a. Root Node

*Node* ini merupakan *node* yang terletak paling atas dari suatu pohon.

b. Internal Node

*Node* ini merupakan *node* percabangan, hanya terdapat satu input serta mempunyai minimal dua output.

c. Leaf Node

*Node* ini merupakan *node* akhir, hanya memiliki satu masukan, dan tidak mempunyai keluaran.

1). Algoritma Decision Tree C4.5

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

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

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (1.3) |

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

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

|  |  |  |
| --- | --- | --- |
|  |  | (1.4) |
|  |  |  |

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

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

|  |  |  |
| --- | --- | --- |
|  |  | (1.5) |

*S1* sampai *Sc* adalah *c* subset yang dihasilkan dari pemecahan *S* dengan menggunakan atribut *A* yang mempunyai sebanyak *c* nilai. Selanjutnya *gain ratio* dihitung dengan cara:

|  |  |  |
| --- | --- | --- |
|  |  | (1.6) |

*[Halaman ini sengaja dikosongkan]*