BAB 2
LANDASAN TEORI

2.1 Bahasa Isyarat

Bahasa isyarat adalah salah satu media komunikasi utama bagi para penderita tuna-rungu di seluruh dunia. Pengguna bahasa isyarat di seluruh dunia cukup banyak. Setiap negara bahkan setiap daerah mempunyai bahasa isyarat masing-masing yang berbeda. Bahasa isyarat pada dasarnya mempunyai sifat dinamis, karena menggunakan gerakan atau perubahan gestur tubuh sebagai ganti suara tutur untuk berkomunikasi [1].

Dalam ranah penelitian sistem pengenalan bahasa isyarat dengan menggunakan komputer, para peneliti mengelompokkan bahasa isyarat menjadi dua kelompok berdasarkan sifat masukan sistem, yaitu bahasa isyarat statis dan bahasa isyarat dinamis. Bahasa isyarat statis digunakan pada sistem pengenalan bahasa isyarat yang masukannya bersifat statis atau diam, yang berupa citra atau foto. Karena masukannya adalah citra atau foto bersifat statis maka, maka bahasa isyarat statis hanya dapat digunakan untuk mengenal isyarat abjad saja. Jadi untuk membentuk suatu kata atau kalimat masih belum bisa.

Sedangkan yang dimaksud dengan bahasa isyarat dinamis adalah sistem pengenal bahasa isyarat yang menggunakan masukan yang bersifat dinamis yang berupa gerakan gestur peraga bahasa isyarat dan umumnya dilakukan secara waktu nyata.

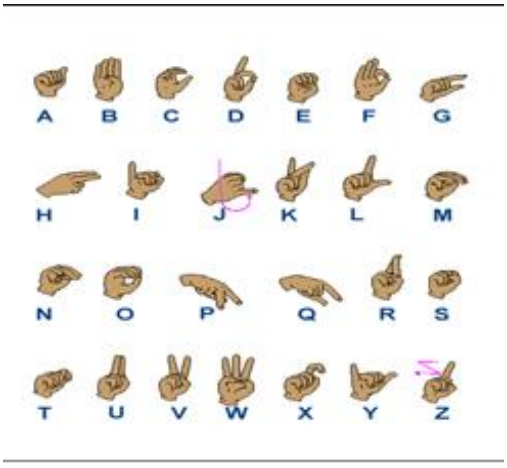
Gestur adalah suatu bentuk bahasa tubuh atau komunikasi non-verbal. Gestur yang umum digunakan merupakan kombinasi atas bentuk/pola tangan, orientasi dan gerakan tangan, ekspresi muka, dan pola bibir. Gestur tangan sendiri dapat diklasifikasikan menjadi beberapa kategori, seperti gestur untuk percakapan, gestur kontrol, gestur manipulatif, dan gestur untuk komunikasi. Salah satu bagian dari bahasa isyarat adalah bahasa isyarat yang diperagakan hanya dengan menggunakan perubahan *pose gesture* tangan terutama pose dari jari-jari tangan yang disebut dengan bahasa isyarat *finger spelling* [1].

2.1.1 Finger Spelling

Bahasa isyarat *finger spelling* menggunakan variasi pose jari-jari dan telapak tangan untuk merepresentasikan abjad demi abjad sehingga membentuk suatu kata, atau dengan kata lain digunakan untuk mengeja suatu kata. Bahasa isyarat *finger spelling* terutama digunakan untuk mengkomunikasikan informasi tentang nama orang, tempat dan objek lain yang tidak tercakup atau belum dikenal dalam bahasa isyarat. Jadi isyarat *finger spelling* ini juga memegang peran yang penting dalam bahasa isyarat. Secara umum bahasa isyarat *finger spelling* ini pada dasarnya sudah dapat digunakan untuk berkomunikasi namun dengan cara mengeja kata atau kalimat abjad demi abjad [1].

2.1.2 Sistem Isyarat Bahasa Indonesia (SIBI)

Sistem Isyarat Bahasa Indonesia (SIBI) yang dibakukan ini merupakan salah satu media yang membantu berkomunikasi sesama kaum tuna rungu dan dengan masyarakat luas. Wujudnya adalah tataan yang sistematis tentang seperangkat isyarat jari, tangan, dan berbagai gerak yang mlambangkan kosa kata bahasa Indonesia. Sistem ini sama dengan bahasa isyarat yang diterapkan di Amerika (*ASL - American Sign Language*) [4].



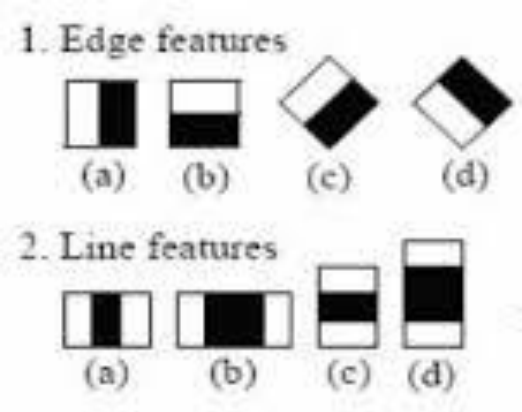
Gambar 2. 1 Sistem Isyarat Bahasa Indonesia

2.2 Metode Viola Jones

Metode *Viola-Jones* merupakan salah satu metode pendeteksian objek yang cukup populer, yang dapat memberikan hasil dengan tingkat keakuratan yang cukup tinggi dan dengan kecepatan yang sangat tinggi. Metode ini, diusulkan oleh *Paul Viola* dan *Michael Jones* pada tahun 2001, dengan menggabungkan empat kunci utama untuk mendeteksi suatu objek, yaitu [3]:

a. Fitur persegi sederhana, disebut fitur Haar
b. Integral image untuk pendeteksian fitur dengan cepat
c. Metode AdaBoost machine-learning
d. Cascade classifier untuk mengkombinasikan banyak fitur

Fitur yang digunakan dalam metode *Viola-Jones* adalah *fitur Haar* yang didasari oleh Haar *wavelets*. Haar *wavelets* adalah satu gelombang panjang berupa gelombang persegi (satu interval tinggi satu dan satu interval rendah). Dalam dua dimensi, gelombang persegi ini digambarkan dengan sepasang persegi yang berdekatan – satu terang dan satu gelap.



Gambar 2. 2 Fitur *Haar*

Ada atau tidaknya fitur Haar dalam suatu citra ditentukan dengan cara *mengurangi nilai rata-rata piksel daerah gelap dengan nilai rata-rata piksel daerah terang*. Jika hasilnya diatas ambang maka fitur tersebut dikatakan ada.

Untuk menentukan ada atau tidaknya ratusan fitur Haar pada suatu citra secara efektif, *Viola-Jones* menggunakan teknik yang disebut *integral image*.

Integral image merupakan teknik yang dapat mempercepat proses pendeteksian suatu objek dengan menggabungkan unit terkecil dari citra yaitu nilai-nilai piksel menjadi suatu representasi citra baru, caranya dengan menjumlahkan piksel yang ada di sebelah kiri dan atas titik tersebut.

Nilai fitur *haar* = | (total piksel hitam) – (total piksel putih) | (2.1)

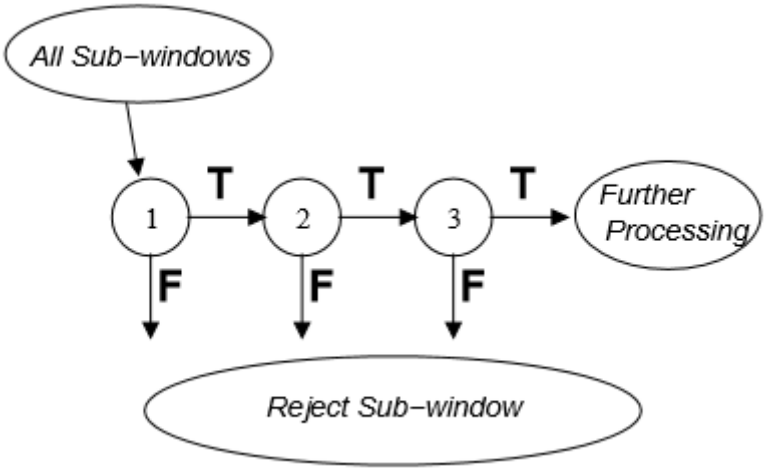
Untuk menghitung nilai area dengan menggunakan integral image, terdapat rumusan sebagai berikut:

SUM = D + A - (B + C) (2.2)

SUM adalah: hasil mengurangi pada bagian kanan bawah area (D) dengan bagian luar pojok kiri atas area (A), dikurangi hasil tambah bagian pojok luar kiri bawah area (C) dengan luar pojok kanan atas area (B) [3].

Proses pencarian nilai fitur ini dilakukan secara iteratif mulai dari ujung kiri atas hingga ujung kanan bawah dengan pergeseran sebesar Δx dan Δy. Semakin kecil nilai Δx dan Δy, maka semakin akurat deteksi citra tersebut. Nilai Δx dan Δy yang sering digunakan adalah 1.

Selanjutnya, setelah nilai fitur didapatkan Viola-Jones menggunakan metode *AdaBoost machine-learning* untuk mengetahui suatu fitur apakah merepresentasikan ada tidaknya *object* dalam suatu citra masukan. *Metode AdaBoost* menggabungkan banyak *classifier* lemah menjadi satu *classifier* kuat. *Classifier* adalah informasi ciri yang menandakan adanya suatu objek pada suatu citra. Sedangkan *classifier* lemah adalah suatu jawaban benar namun memiliki tingkat kebenaran yang kurang akurat, jika digabung maka *classifier* lemah tersebut akan menghasilkan suatu *classifier* kuat. Berikut ini adalah skema deteksi *cascade classifier*:



Gambar 2. 3 Skema *Cascade Classifier*

Gambar 2.3, menjelaskan bagaimana metode *AdaBoost* ini akan menyeleksi nilai fitur dari citra masukan, jika pada fitur tersebut tidak terdeteksi adanya tangan maka akan disimpulkan tidak ada tangan, jika ada akan diteruskan ke tingkat selanjutnya sampai ke tingkat terakhir, lalu akan disimpulkan pada citra masukan terdeteksi adanya tangan. Proses penyeleksian fitur tersebut dengan cara membandingkan nilai fitur dengan nilai ambang yang telah ditentukan pada tiap tingkat, jika nilai fitur sama atau diatas nilai ambang maka terdeteksi adanya tangan dalam fitur tersebut.

Tingkatan proses penyeleksian fitur diurutkan berdasarkan bobot terberat sampai teringan, sehingga dapat secepat mungkin untuk mengklasifikasikan bahwa tidak terdeteksi tangan dalam citra masukan, pengurutan proses penyeleksian fitur ini disebut *cascade classifier*.

2.3 Computer Vision

Computer Vision adalah suatu proses transformasi atau perubahan dari data yang berasal dari kamera video maupun foto atau gambar kedalam sebuah hasil keputusan ataupun sebuah presentasi yang baru, dimana hasil dari kegiatan transformasi tersebut memiliki kepentingan untuk mencapai suatu tujuan. Data yang dimasukan kedalam kegiatan transformasi tersebut memungkinkan untuk memiliki beberapa informasi yang terkonstektual seperti halnya sebuah

foto/gambar yang didalamnya terdapat berbagai objek. Dengan demikian akan didapatkan keputusan-keputusan yang akan diambil pada gambar, misalnya berupa “adakah telapak tangan seseorang pada gambar tersebut?” atau “siapa sajakah orang yang terdapat pada foto tersebut?”. Adapun perubahan kedalam presentasi yang baru seperti perubahan gambar menjadi *grayscale* atau juga pemotongan objek pada gambar.

Berbeda halnya dengan manusia yang memiliki pemikiran untuk meneliti, memahami, dan membandingkan informasi pada objek secara langsung dengan informasi dari pengalaman-pengalaman yang didapatkan selama bertahun-tahun hidup didunia. Dalam sistem penglihatan mesin (*machine vision*) komputer hanya mampu mendapatkan informasi berupa kumpulan angka-angka dari media input data seperti kamera atau sensor. [5]

2.4 Pengolahan Citra Digital

Pengolahan citra digital adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Setiap foto dengan bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Prinsip-prinsip dasar dalam pengolahan citra: [6]

1. Peningkatan kecerahan dan kontras

   Peningkatan kecerahan dan kontras adalah sebuah perbaikan dalam citra yang merubah citra yang memiliki nilai kecerahan dan nilai kontras kurang menjadi nilai kecerahan dan nilai kontras yang lebih baik.

2. Penghilang derau

   Penghilang derau adalah proses penghilangan derau dari sebuah citra dengan keadaan distorsi atau memiliki derau.

3. Pencarian bentuk objek

Pencarian bentuk objek adalah proses untuk mengenali suatu objek dalam sebuah citra dengan menggunakan pendekatan-pendekatan tertentu untuk mendapatkan nilai dari objek tersebut.

Beberapa jenis operasi dari pengolahan citra adalah sebagai berikut: [6]

1. Modifikasi Kecemerlangan (Brightness Modifikasi)

   Mengubah nilai keabuan atau warna dari gelap menjadi terang ataupun sebaliknya mengubah citra yang terlalu terang/pucat menjadi gelap.

2. Peningkatan Kontras (Contrast Enhancement)

   Dengan meningkatkan kontras dari sebuah citra maka titik yang cenderung gelap menjadi lebih gelap dan yang cenderung terang menjadi lebih terang.

3. Negasi

   Operasi untuk mendapatkan citra negatif (negative image)

4. Pengabuan (*Grayscale*)

   Merupakan proses konversi citra dengan warna sebenarnya (*true color*) menjadi citra keabuan (*grayscale*). Pengubahan gambar ke dalam bentuk grayscale ini dilakukan dengan mengambil nilai pixel dari satu gambar masukan kemudian dihitung menggunakan persamaan sebagai berikut:

   Grayscale = 0.2989R + 0.5870G + 0.1140G (2.3)

5. Pengambangan (*Thresholding*)

   Operasi ini digunakan untuk mengubah citra dengan format skala keabuan yang mempunyai nilai lebih dari dua menjadi citra biner yang hanya memiliki dua nilai yaitu 0 dan 1.

6. Pencerminan (*Flipping*)

   Pencerminan merupakan proses menggambar citra ke bentuk kebalikan dari citra tersebut, seperti ketika orang sedang bercermin.

7. Rotasi (*Rotating*)

Rotasi adalah proses memutar koordinat citra sesuai dengan derajat yang ditentukan.

8. Pemotongan (*Cropping*)

   Memotong bagian yang diinginkan dari sebuah citra.

9. Pengskalaan (*Scaling*)

   Mengubah ukuran citra menjadi lebih besar atau lebih kecil.

10. Deteksi Tepi (Edge *Detection*)

    Deteksi tepi adalah suatu proses yang menghasilkan tepi-tepi dari obyek dalam sebuah citra.

2.5 Object Oriented Analysis and Design

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika *requirement* (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [7].

2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal. Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka [7]. UML merupakan bahasa pemodelan yang terdiri banyak model diantaranya adalah:

1. Use Case Diagram

Use case adalah teknik untuk menggambarkan fungsional sebuah sistem. *Use case* diagram memperlihatkan hubungan diantara *actor* dan *use case*. Aktor

mempresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang menggambarkan interaksi antara user dengan sistem. Fungsionalitas sistem didefinisikan ke dalam *use case* dari sudut eksternal sistem yang berguna untuk diuji kelayakan sistem.

2. *Sequence Diagram*

Sequence diagram menggambarkan interaksi diagram yang menunjukan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior, Sebuah *Sequence diagram* secara khusus menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam *use case*. *Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan *output* tertentu.

3. *Class Diagram*

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

2.7 Microsoft Visual Studio 2012

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. *Visual Studio*

mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun managed code (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan diatas *.NET Compact Framework*).

2.8 Pengujian

Pengujian adalah suatu proses mengevaluasi suatu sistem atau komponen dengan maksud untuk menemukan apakah itu memenuhi persyaratan yang ditentukan atau tidak. Kegiatan ini menghasilkan persyaratan yang sebenarnya, dan perbedaan antara hasil mereka. Dengan kata lain pengujian ialah mengeksekusi sistem untuk mengidentifikasi kesenjangan, kesalahan atau persyaratan yang hilang bertentangan dengan keinginan atau persyaratan yang sebenarnya. [8] Menurut *ANSI* atau *IEEE 1059* standar, Pengujian dapat didefinisikan sebagai "Suatu proses menganalisis item perangkat lunak untuk mendeteksi perbedaan antara kondisi yang ada dan diperlukan (yaitu cacat / kesalahan / bug) dan mengevaluasi fitur item

perangkat lunak".

2.8.1 Pengujian *Black Box*

Pengujian yang mengabaikan mekanisme internal sistem atau komponen dan fokus semata-mata pada output yang dihasilkan yang merespon input yang dipilih dan kondisi eksekusi. Pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu. [8]

Tujuan *Black Box* adalah menemukan:

1. Fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Error pada struktur data atau akses database external
4. Error pada kinerja
5. Error pada saat inisialisasi dan terminasi
6. Kesensitifan sistem terhadap nilai input tertentu
7. Batasan dari suatu data

2.8.2 *Confusion Matrix*

Confusion Matrix (Kohavi dan Provost, 1998) berisi informasi tentang klasifikasi aktual dan yang telah diprediksi yang dilakukan oleh sistem klasifikasi. Kinerja sistem tersebut umumnya dievaluasi dengan menggunakan data dalam matriks. Tabel berikut menunjukkan *confusion matrix* untuk klasifikasi dua kelas.

Tabel 2. 1 *Confusion Matrix*

		Predicted Class	
		Positif	**Negatif**
Actual Class	**Positif**	True Positives	False Positive
	Negatif	False Negative	True Negative

True positives adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *false negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *true negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif. Setiap kolom dari *confusion matrix* merupakan contoh di kelas yang telah diprediksi, sedangkan setiap baris mewakili contoh di kelas yang sebenarnya. Setelah didapat *true positives, false positives, true negatives* dan *false negatives*, selanjutnya hitung nilai *precision* dan akurasinya. *Precision* adalah ukuran terhadap suatu kelas yang telah diprediksi. Berikut persamaan dari *precision* dan akurasi.

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad \text{…(2.4)}$$

$$Precision = \frac{TP}{TP + FP} \quad \text{…(2.5)}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{…(2.6)}$$

Dimana:

: Jumlah *True Positive*

: Jumlah *False Positive*

: Jumlah *True Negative*

: Jumlah *False Negative*