BAB 2
TINJAUAN PUSTAKA

2.1 Gelombang bunyi

Gelombang bunyi atau lebih khusus dikenal sebagai gelombang akustik adalah gelombang longitudinal yang berada dalam sebuah medium,biasanya udara.Gelombang suara berasal dari benda yang bergetar.Bentuk gelombang suara periodik akan menimbulkan sensasi yang menyenangkan (jika intensitasnya tepat) seperti bunyi musik.Sebaliknya,bunyi yang tidak *periodic* akan menimbulkan suara derau (*noise*) [9].

2.2 Nada

Sebuah nada adalah suatu bunyi yang memiliki nilai tertentu,yaitu *pitch,*yang diukur untuk menggambarkan jenis nada tersebut. Serangkaian nada yang tersusun teratur berdasarkan ketukan dan pola disebut musik [10]. Dalam teori musik, setiap nada memiliki nilai tertentu menurut frekuensinya ataupun jarak relatif nada tersebut terhadap nada patokan. Nada dapat diatur dalam tangga nada yang berdeda-beda. Nada dasar menentukan frekuensi tiap-tiap nada pada suatu karya music [11].

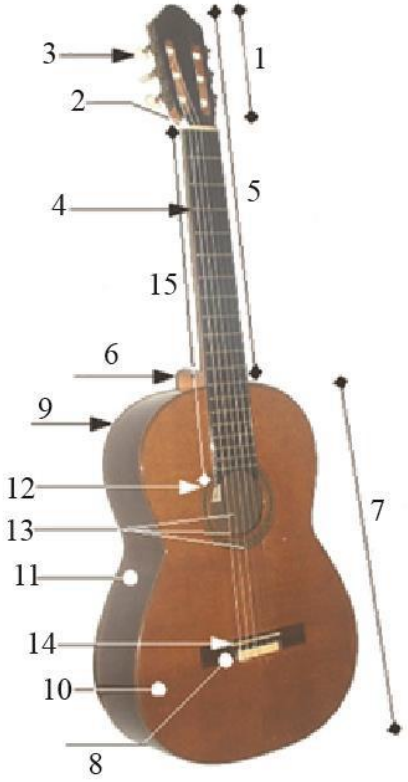
Nada merupakan jenis supra segmental yang ditandai oleh tinggi-rendahnya arus ujaran yang terjadi karena frekuensi getaran yang berbeda antar segmen [6]. Misalnya pada saat seseorang berada dalam kesedihan maka ia akan berbicara dengan nada yang rendah.Sebaliknya bila ia berada dalam keadaan marah atau gembira, ia berbicara dengan nada yang tinggi.

2.2 Tangga Nada

Tangga nada merupakan kumpulan nada-nada yang harmonis.Kumpulan dari semua nada pada music disebut tangga nada kromatik. Tangga nada kromatik dapat diartikan sebagai “nada setiap warna”. Istilah “tangga nada kromatik” dipakai untuk kedua belas nada dari tiap oktaf [5].

2.3 Alat Musik Gitar

Masuknya alat musik gitar ke Indonesia salah satunya adalah dibawa oleh tawanan asal portugis sekitar abad ke-7. [6]. Secara umum gitar dibagi menjadi 2 jenis, gitar akustik dan gitar listrik (*electric guitar*). Gitar akustik merupakan gitar yang terbuat dari kayu, dan terdapat lubang suara atau tabung resonansi (*sound hole*). Bagian-bagian pada gitar akustik ditunjuk kan pada gambar 2.1 berikut.



1. Gagang (head stock)
2. Ganjal senar atas (nut)
3. Penyetem senar (tuning key/ tuning machines/ tuners)
4. Dudukan senar yang ditekan (frets)
5. Leher (neck)
6. Tumit (heel)
7. Badan gitar (body)
8. Ganjal senar bawah (bridge)
9. Badan belakang (back body)
10. Papan bunyi (sound board(top))
11. Badan samping (bodysides(ribs))
12. Lubang bunyi (sound hole)
13. Senar (strings)
14. Dudukan senar bawah (saddle)
15. Papan jari (fretboard/ fingerboard)

Gambar 2. 1 Bagian-bagian gitar akustik[9].

Senar gitar yang dipetik akan menghasilkan bunyi yang memiliki frekuensi dan setelan nada yang berbeda tergantung kepada penggunaan dan keinginan dari pengguna gitar tersebut. Frekuensi suara senar gitar yang umum digunakan dapat dilihat pada tabel 2.1.

Tabel 2. 1 Frekuensi Nada Gitar [6].

Nada senar gitar	Nomor Kunci	Frekuensi
E(senar 6)	E2	82.4
A(senar 5)	A2	110.0
G(senar 4)	D3	146.8
D(senar 3)	G3	196.0
B(senar 2)	B3	246.9
E(senar 1)	E4	329.6

2.3 Akor Gitar

Chord atau akor secara umum dapat diartikan sebagai suatu rangkaian nada-nada yang tersusun secara teratur dari sebuah tangga nada dan bias mewakili tangga nada tersebut [11].Jika diartikan dalam gitar, akor adalah dua atau lebih nada yang dibunyikan bersamaan dalam satu petikan dan dalam satu waktu yang sama. Dalam sebuah lagu, akor berfungsi sebagai musik pengiring atau *rhythm*.

Secara umum, akor dibagi kedalam 3 kelompok yaitu : akor mayor, akor minor dan akor substitusi [10].

1. Akor Mayor

   Akor mayor adalah beberapa nada diatonic mayor yang dibunyikan secara bersamaan, dalam satu waktu yang sama. Nada diatonis mayor adalah nada-nada dengan urutan jarak antar nada 1-1-1/2-1-1-1-1/2.

2. Akor Minor

   Akor minor adalah beberapa nada diatonic minor yang dibunyikan secara bersamaan, dalam satu waktu yang sama. Nada diatonis mimor adalah nada-nada dengan urutan jarak antar nada 1-1/2-1-1-1/2-1-1.

3. Akor Substitusi

   Akor substitusi tidak berasal dari tangga nada diatonis mayor maupun minor, tetapi lebih ke interval atau jarak nada yang menyusunnya.

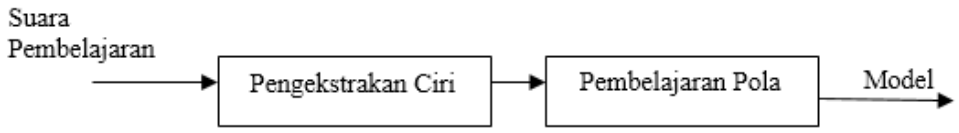
Akor dasar gitar dapat dilihat pada tabel 2.2 berikut.

Tabel 2. 2 Akor Dasar Gitar

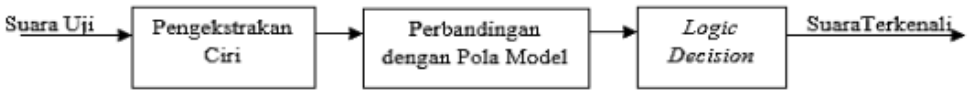
Chord dasar	Mayor	Minor
C	CM	Cm
C#	C#M	C#m
D	DM	Dm
D#	D#M	D#m
E	EM	Em
F	FM	Fm
F#	F#M	F#m
G	GM	Gm
G#	G#M	G#m
A	AM	Am
A#	A#M	A#m
B	BM	Bm

2.6 Pengenalan Suara

Pengenalan suara merupakan salah satu upaya agar suara dapat dikenali atau diidentifikasi sehingga dapat dimanfaatkan. Pengenalan suara dapat dibedakan ke dalam tiga bentuk pendekatan, yaitu pendekatan akustik-fonetik (*the acoustic-phonetic approach*), pendekatan kecerdasan buatan (*the artificial intelligence approach*), dan pendekatan pengenalan-pola (*the pattern recognition approach*). Blok diagram pengenalan pola pada pengenalan suara ditunjukkan pada Gambar 2.2.



(a) Blok Diagram Pembelajaran Pola



(b) Blok Diagram Pengenalan Suara

Gambar 2. 2 Blok diagram pembelajaran pola dan pengenalan suara[7].

Berikut ini merupakan penjelasan dari masing-masing blok:

1. Pengekstraksi ciri.

   Merupakan proses mendapatkan sederetan besaran pada bagian sinyal masukan untuk menetapkan pola pembelajaran atau pola uji. Pada sinyal suara, ciri-ciri besaran biasanya merupakan keluaran dari beberapa bentuk teknik analisis spektrum seperti *filter bank analyzer,* LPC atau DFT (*Discrete Fourier Transform*).

2. Pembelajaran Pola

   Satu atau lebih pola pembelajaran yang berhubungan dengan bunyi suara dari kelas yang sama, digunakan untuk membuat pola representatif dari ciri-ciri kelas tersebut. Hasilnya yang biasa disebut dengan pola referensi, dapat menjadi sebuah model yang mempunyai karakteristik bentuk statistik dari ciri-ciri pola referensi.

3. Perbandingan dengan Pola Model

   Pola uji yang akan dikenali, dibandingkan dengan setiap kelas pola referensi. Kesamaan besaran antara pola uji dengan setiap pola referensi akan dihitung.

4. Logic Decision

   Menentukan kelas pola referensi mana yang paling cocok untuk pola uji berdasarkan klasifikasi pola.

Pengenalan suara secara umum dapat dibagi menjadi tiga tahap, yaitu : ekstraksi ciri, pemodelan, dan pengenalan. Ekstraksi ciri adalah upaya untuk memperoleh ciri dari sinyal suara.

2.4 Audio Digital

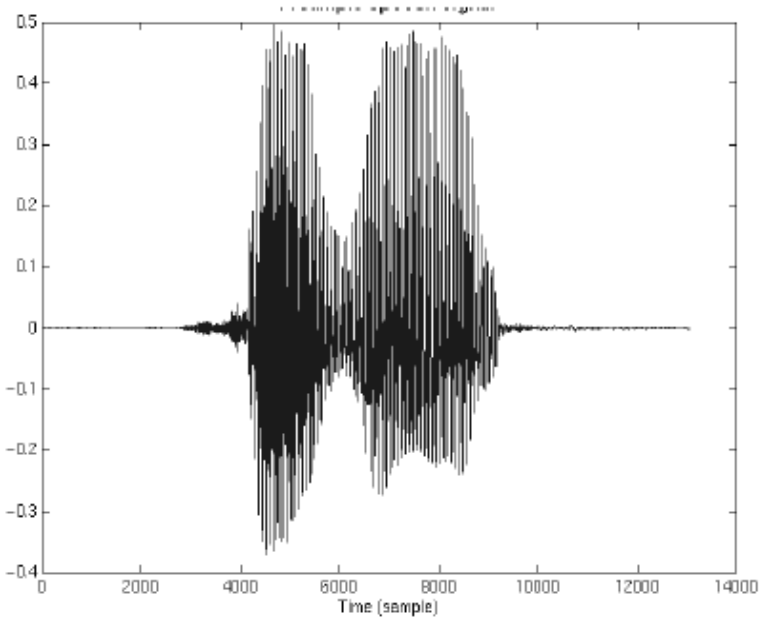
Audio digital merupakan versi digital dari suara analog. Pengubahan suara analog menjadi suara digital membutuhkan suatu alat yang disebut Analog To Digital Converter (ADC). ADC akan mengubah amplitudo sebuah gelombang analog ke dalam waktu interval (sampling) sehingga menghasilkan representasi digital dari suara. Sampling adalah melakukan pencuplikan amplitudo gelombang suara pada tiap satu satuan waktu [7].

Berlawanan dengan ADC, Digital to Analog Converter (DAC) akan mengubah suara digital ke alat suara analog (speaker). Audio digital merupakan representasi dari suara asli (original sound). Dengan kata lain, audio digital merupakan sampel suara. Kualitas perekaman digital tergantung pada seberapa sering sampel diambil (angka sampling atau frekuensi dihitung dalam KiloHertz atau seribu sampel per detik). Tiga frekuensi sampling yang paling sering digunakan dalam multimedia adalah kualitas CD 44.1 kHz, 22.05 kHz, 11.25 kHz dengan ukuran sampel 8 bit dan 16 bit. Ukuran sampel 8 bit menyediakan 256 unit deskripsi jarak dinamis atau amplitudo (level suara dalam satu waktu) [8].

Ketika pengambilan sampling gelombang dengan ADC mempunyai kendali terhadap dua variabel, yaitu:

1. *Sampling Rate*

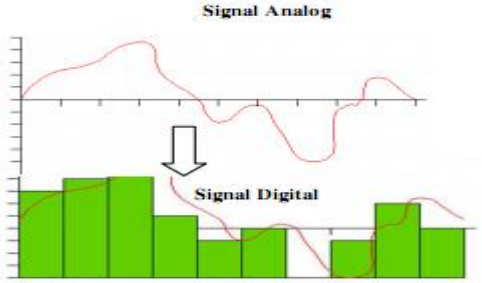
   Sampling rate adalah mengendalikan berapa banyak sampel yang akan diambil per-detiknya. Contoh input suara dapat dilihat pada gambar 2.2 berikut.



Gambar 2. 3 Input Suara

2. *Sampling Precision Sampling*

   precision adalah mngendalikan berapa banyak gradasi (tingkat kuantisasi) yang dimungkinkan ketika mengambil sampel. Dengan gambar sebagai berikut:



Gambar 2. 4 Suara Analog menjadi Suara Digital

2.5 *Wave form audio* (WAV)

WAV adalah format audio standar Microsoft dan IBM untuk PC. WAV biasanya menggunakan coding PCM (*Pulse Code Modulation*). WAV adalah data tidak terkompres sehingga seluruh sampel audio disimpan semuanya di harddisk. Sofware yang dapat menciptakan WAV dari analog sound misalnya windows sound recorder. WAV jarang digunakan di internet karena ukurannya yang relative besar. Maksimum ukuran file WAV adalah 2GB.[8]

2.6 Kecerdasan Buatan

Kecerdasan buatan (*Artificial intelligence*) adalah salah satu cabang ilmu pengetahuan yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal ini biasanya dilakukan dengan mengikuti karakteristik dan analogi berpikir dari kecerdasan manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer.

Semakin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Karakteristik cerdas sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Berikut merupakan lingkup utama kecerdasan buatan: [4]

a. Sistem pakar (*expert system*)

b. Pengolahan bahasa alami (*natural language processing*)

c. Pengolahan suara (*speech recognition*)

d. Robotika dan sistem sensor (*robotic and sensory system)*

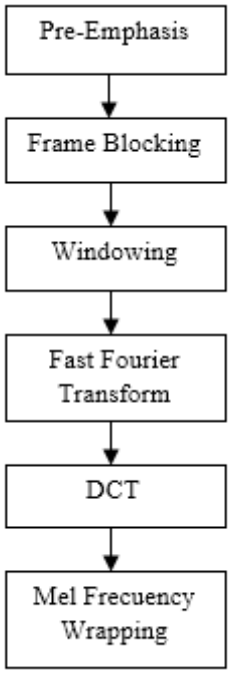
e. Visi Komputer (*computer vision*)

f. Tutor Berbasis Komputer Cerdas (*Intelligent computer-aided instruction*)

g. *Game Playing*

2.7 Metode MFCC (*Mel Frequency Cepstral Coefficients*)

MFCC (*Mel Frequency Cepstral Coefficients*) merupakan salah satu metode yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition*. Metode ini digunakan untuk melakukan *feature extraction*, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter [2].



Gambar 2. 5 Diagram Blok MFCC

Ada pun proses pengolahan sinyal pada ekstraksi ciri menggunakan MFCC:

1. *Pre-emphasis*

   Pre-emphasis merupakan salah satu jenis filter yang sering digunakan sebelum sebuah signal diproses lebih lanjut. Filter ini mempertahankan frekuensi-frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara. Tujuan dari preemphasis filter ini adalah:

   a. Mengurangi noise ratio pada sinyal, sehingga dapat meningkatkan kulitas sinyal.
   b. Untuk mendapatkan bentuk spektral frekuensi sinyal wicara yang lebih halus.
   c. Menyeimbangkan spektrum dari sinyal suara.

Pada saat memproduksi sinyal suara, glotis manusia menghasilkan sekitar -12 dB *octave slope*. Namun ketika energi akustik tersebut dikeluarkan melalui bibir, terjadi peningkatan sebesar +6 dB. Sehingga sinyal yang terekam oleh *microphone* adalah sekitar -6 dB *octave slope*. Dimana bentuk spektral yang relatif bernilai tinggi untuk daerah rendah dan cenderung turun secara tajam untuk daerah fekuensi diatas 2000 Hz. *Filter preemphasis* didasari oleh hubungan *input/output* dalam domain waktu yang dinyatakan dalam persamaan seperti berikut:

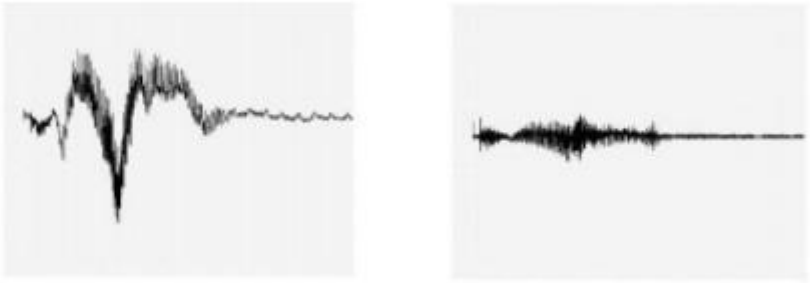
$$y(n) = s(n)-\alpha s(n-1) \qquad (2.1)$$

dimana:

y(n) = Sinyal hasil *preemphasis*

s(n) = Sinyal sebelum *preemphasis*

α merupakan konstanta *filter preemphasis*, biasanya bernilai 0.97. Dalam bentuk dasar operator s sebagai unit filter, persamaan diatas akan memberikan sebuah *transfer function filter preemphasis* seperti berikut:



a. Sebelum preemphasis b. Sesudah preemphasis

Gambar 2. 6 Sebelum dan sesudah Preemphasis

2. *Frame blocking*

Frame Blocking adalah pembagian suara menjadi beberapa frame yang nantinya dapat memudahkan dalam perhitungan dan analisa suara, satu frame terdiri dari beberapa sampel tergantung tiap berapa detik suara akan disampel dan berapa besar frekuensi samplingnya [7]. Panjang frame yang digunakan, sangat mempengaruhi keberhasilan dalam analisa spektral. Disatu sisi, ukuran dari frame harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik. Tetapi di sisi lain, ukuran frame juga harus cukup pendek untuk dapat menunjukkan resolusi waktu yang baik. Refresentasi fungsi *frame blocking* sebagai berikut:

$$x(n) = y(M+n) \quad (2.2)$$

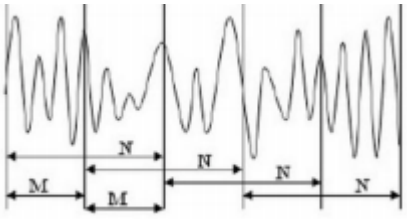
dimana,

x(n) =sinyal sesudah di frame blocking

y =sinyal hasil preemphasis

M = overlapping frame

n= 1,2,3,…



Gambar 2. 7 Bentuk sinyal yang di frame blocking

3. *Windowing*

Suara yang di potong-potong menjadi beberapa frame membuat data suara menjadi discontinue, hal ini mengakibatkan kesalahan data proses *fourier transform.* Agar tidak terjadi kesalahan data pada proses fourier transform maka sampel suara yang telah dibagi menjadi beberapa *frame* perlu dijadikan suara kontinu dengan cara mengalikan tiap frame dengan *window* tertentu[3]. Jenis *window* yang dipakai pada proses ini adalah *window hamming*. Berikut ini adalah representasi fungsi *window* terhadap signal suara yang diinputkan.

$$(n) = x\,(n)w(n) \qquad (2.3)$$

rumus window hamming sebagai berikut:

$$w[n]= 0.54-0.46*\cos(2*\pi*n/(N-1)) \qquad (2.4)$$

dimana:

n = 1,2,3,…

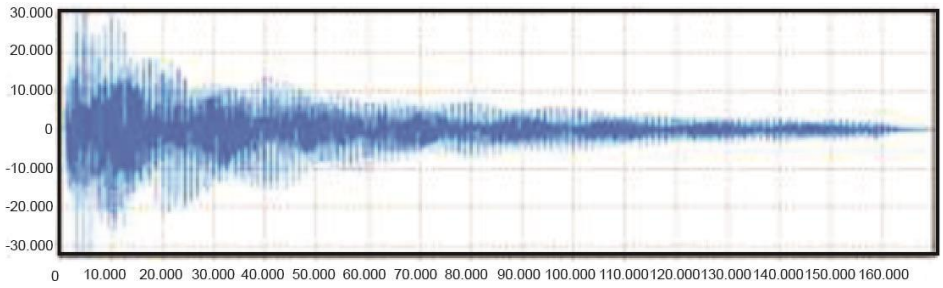
π = 3.14

N = panjang f*rame*

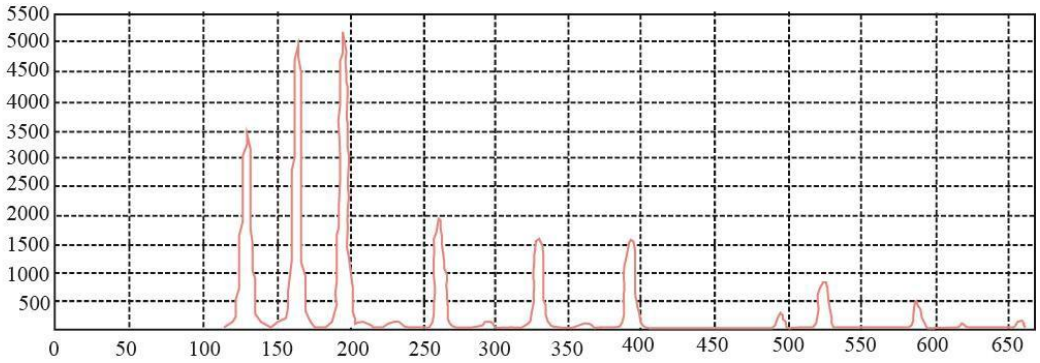
Fungsi window yang paling sering digunakan dalam aplikasi *voice recognition* adalah *Window Hamming*. Fungsi *window hamming* ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu noise yang dihasilkan pun tidak terlalu besar (kurang lebih 1.36 BINS).

4. Fast Fourier Transform (FFT)

   Fast Fourier Transform merupakan DFT dengan algoritma yang lebih optimal, sehingga menghasilkan perhitungan yang lebih cepat. Dengan DFT, memerlukan waktu O(n2) untuk mengolah sampel data sebanyak nbuah. Hal ini tentunya akan memakan waktu lama bila sampel data makin banyak. Proses pada metode *fast Fourier Transform* (FFT) adalah mengkonversi setiap *frame* N sampel dari domain waktu menjadi domain frekuensi, transformasi fourier merupakan metode yang efisien untuk transformasi fourier diskrit yang banyak digunakan pada pemrosesan sinyal digital [5]. Contoh sinyal dalam domain waktu pada gambar 2.3 dan contoh sinyal dalam domain frekuensi pada gambar 2.8 berikut [7].



Gambar 2. 8 Sinyal dalam domain waktu



Gambar 2. 9 Sinyal dalam domain frekuensi

Metode FFT dapat dilakukan dalam domain waktu dan frekuensi yang disebut sebagai desimasi – dalam – waktu *(decimation-in-time)* dan desimasi-dalam frekuensi *(decimation-in-frequency).* Pada prinsipnya algoritma ini adalah memecah N-titik menjadi dua (N/2)–titik, kemudian di pecah lagi pada tiap (N/2)– titik menjadi dua (N/4) –titik, begitu seterusnya sampai hanya terdapat 1 titik.

5. *Discrete Cosine Transform* (DCT)

   Tahap ini melakukan konversi hasil *mel-frequency* ke domain waktu dengan menggunakan *Discrete Cosine Transform* (DCT). Berikut adalah persamaan yang digunakan dalam DCT.

$$C_j = \sum_{i=0}^{M} X_i \cos\left(j(i-1)2\left(\frac{\pi}{M}\right)\right) \qquad \textbf{(2.5)}$$

Keterangan :

Cj = nilai koefisien C ke-j

j = jumlah koefisien cepstral

i = jumlah wrapping

Xi = hasil mel-frequency pada frekuensi ke-i

2.8 Metode *Naïve Bayesian*

*Naïve bayes*ian adalah teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes [2].

1. Asumsi independensi (ketidaktergantungan) yang kuat (naif).
2. Model yang digunakan adalah “model fitur independen”.

Independensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak ada kaitannya dengan adanya atau tidak adanya fitur yang lain dalam data yang sama. Contoh: kasus klasifikasi hewan dengan fitur: penutup kulit, melahirkan, berat, dan menyusui.

1. Dalam dunia nyata, hewan yang berkembang biak dengan cara melahirkan dipastikan hewan tersebut menyusui juga, disini ada ketergantungan pada fitur menyusui karena hewan yang menyusui biasanya melahirkan, atau hewan yang bertelur biasanya tidak menyusui.
2. Dalam Bayes, hal tersebut tidak dipandang, sehingga masing-masing fitur seolah tidak ada hubungan apa-apa.

Kasus lain: prediksi hujan

1. Hujan tergantung angin, cuaca kemarin, kelembaban udara (tidak ada kaitan satu sama lain).
2. Tapi juga tidak boleh memasukkan fitur lain yang tidak ada hubungannya dengan hujan, seperti: gempa bumi, kebakaran, dsb.

Naïve bayes adalah suatu metode pengklasifikasian paling sederhana dengan menggunakan peluang yang ada, dimana diasumsikan bahwa setiap variable X bersifat bebas (*independence*). Karena asumsi variabel tidak saling terikat, maka didapatkan :

$$P(X \mid Y = y) = \prod_{i=1}^{q} P(X_i \mid Y = y) \qquad (2.6)$$

Maka dari itu untuk data *continue* dapat diselesaikan dengan menggunakan langkah-langkah berikut.

Training :

1. Hitung rata-rata (*mean*) tiap fitur dalam dataset training dengan.

$$\mu = \frac{\sum x_i}{n} \qquad (2.7)$$

Dimana:

μ = mean

n = banyaknya data

$\sum x_i$ = jumlah nilai data

2. Kemudian hitung nilai varian dari diat dataset training tersebut seperti pada.

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2 \quad (2.8)$$

Dimana:

σ^2= varians

μ= mean

x_i = nilai data

n = banyaknya data

Testing :

1. Hitung probabilitas (*Prior*) tiap kelas yang ada dengan cara menghitung jumlah data tiap kelas dibagi jumlah total data secara keseluruhan.

2. Selanjutnya menghitung densitas probabilitasnya. Fungsi densitas mengekspresikan probabilitas relatif. Data dengan mean μ dan standar deviasi σ, fungsi densitas probabilitasnya adalah :

$$\varphi_{\mu\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.9)$$

Dimana :

x = data masukan

π = 3,14

σ = standar deviasi

μ = mean

3. Setelah didapatkan nilai densitas probabilitasnya, selanjutnya menghitung posterior masing-masing kelas dengan menggunakan persamaan 2.10.

$$posterior = \frac{prior\ x\ likehood}{evidnce} \quad (2.10)$$

Atau bisa ditulis

$$Posterior = \frac{P(C_i)x\ P(X1 \dots \dots \dots \dots \dots \dots \dots Xn|C_i)}{P(X1 \dots \dots \dots \dots \dots \dots \dots Xn|C_i)} \qquad (2.11)$$

Setelah didapat nilai posterior, kelas yang sesuai adalah nilai posterior terbesar.

2.8 Object Oriented Programming (OOP)

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapann prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktifitas analisis beorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [9].

Pada saat ini, metode berorientais objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentranformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan saat ini sangat beragam (aplikasi bisnis, *real-time, utility* dan sebagainya) dengan platform yang berbeda-beda,sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

a. Meningkatkan Produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reuseable*).

b. Kecepatan Pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan mennyababkan berkurangnya kesalahan pada saat pengkodean

c. Kemudahan Pemeliharaan

   Karena dengan model objek, pola-pola yang cendrung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering diubah-ubah.

d. Adanya Konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

e. Meningkatkan Kualitas Perangkat Lunak

   Karena adanya pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengambangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Berikut beberapa contoh bahasa pemograman yang mendukung pemrograman berorientasi objek adalah:

a. Smalltalk

   Smalltalk adalah salah satu bahasa pemograman yang diekmbangkan untuk mendukung pemrograman beroirentasi objek.

b. Bahasa Pemrograman Eiffel

   Eiffel merupakan bahsa pemrograman yang kembangkan untuk mendukung pemrograman berorientasi objek oleh Bertrand Meyer dan *compiler*.

c. Bahasa Pemrograman Web (PHP)

   Php dibuat pertama kali oleh seorang perekayasa perangkat (software engineering) yang bernama Rasmus Lerdoff.

d. Bahasa Pemrograman C++

   C++ merupakan pengembangan lebih lanjut dari bahasa pemrograman C untuk mendukung pemrograman berorientasi objek.

e. Bahasa Pemrograman Java

Java dikembangkan oleh perusahaan Sun Microsystem. Java menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan.

2.9 Basis Data

Basis data (database), atau sering pula dieja basis data, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil query basis data disebut sistem manajemen basis data (database management system, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut.

Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data, ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah lama mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

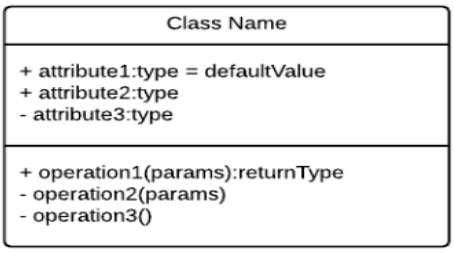
Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (database management system/DBMS).

2.10 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu : *Diagram Interchange Spesification*, UML *Infrastrukture*, UML *Superstrukture, dan Object Constraint Language* (OCL). Pada UML 2.3 terdiri 13 macam diagram yang dikelompokan pada 3 kategori, yaitu:

1. *Structure Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
   a. Diagram Kelas

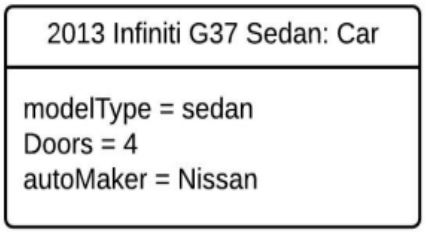
      Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut attribut dan metode atau operasi.Diagram kelas dapat dilihat pada gambar 2.10 berikut.



Gambar 2. 10 Diagram Kelas

   b. Diagram Objek

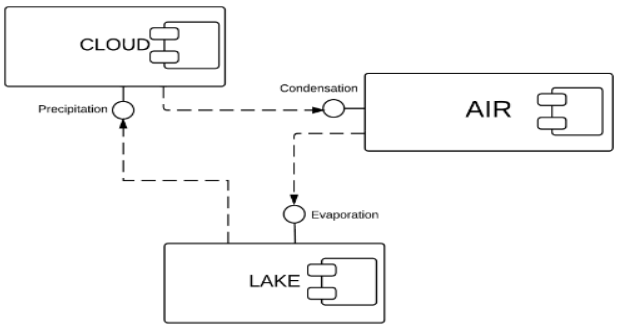
      Digram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.



Gambar 2. 11 Diagram Objek

c. Diagram Komponen

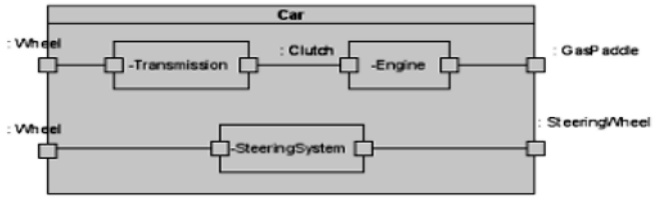
Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.



Gambar 2. 12 Diagram Komponen

d. *Composite Structure Diagram*

Composite Structure Diagram baru mulai ada pada UML versi 2.0. diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*).

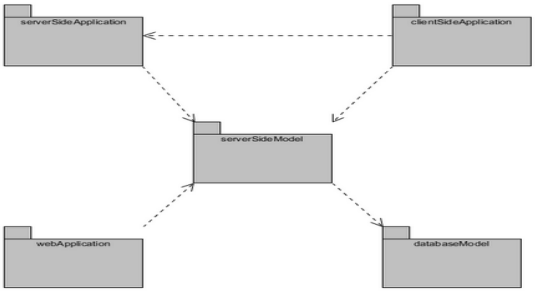


Gambar 2. 13 Composite Structure Diagram

e. *Package Diagram*

Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua

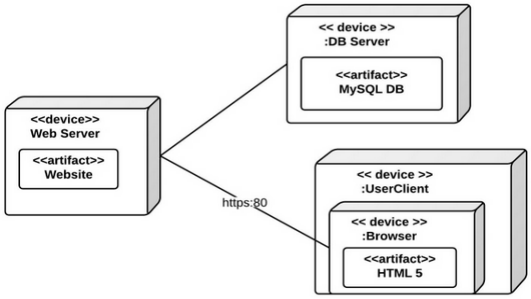
diagram dalam UML dapat dikelompokan menggunakan *Package Diagram*.



Gambar 2. 14 Package Diagram

f. *Deployment Diagram*

Deployment diagram menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.

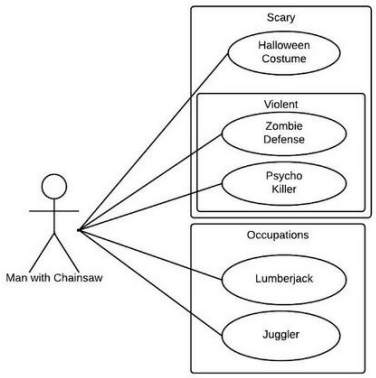


Gambar 2. 15 Deployment Diagram

2. *Behavior Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem [5]. Pada *Behavior Diagram* dibagi menjadi 3 bagian

   a. Use Case Diagram

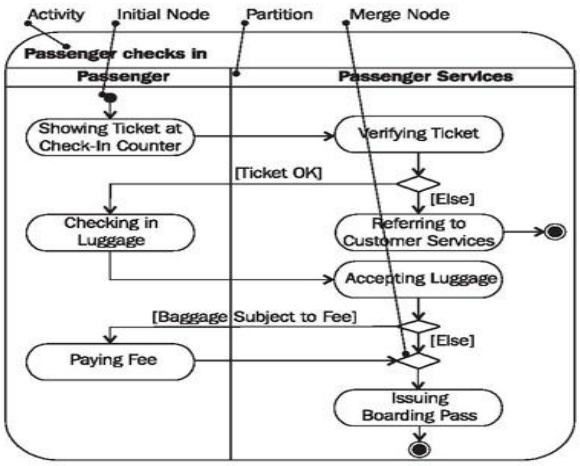
   Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat,. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.



Gambar 2. 16 Use Case Diagram

b. Activity Diagram

Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.



Gambar 2. 17 *Activity* Diagram

c. State Machine Diagram

State machine diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

2.11 Bahasa Pemograman C#

C# (tanda '#' dibaca "Sharp") merupakan bahasa pemograman baru yang diciptakan Microsoft secara khusus sebagai salah satu bahasa pemrograman dalam teknologi .Net sebagai bahasa baru, C# tidak berevolusi dari bahasa C# versi bukan teknologi .Net. dengan demikian C# dapat memaksimalkan kemampuannya tanpa khawatir dengan masalah kompatibilitas dengan versi-versi sebelumnya. Keharusan sebuah perangkat lunak untuk tetap dapat kompatibel dengan versi-versi sebelumnya sebagaimana yang terjadi pada Visual Basic (VB) maupun C++biasanya menghambat optimalitas kemampuan dari perangkat lunak tersebut.

Sejak diluncurkan pada tahun 2000, C# dengan cepat merebut hati progammer C++ bahkan VB. Dengan tata cara penulisan yang mirip C++ dan *interface* mirip VB 6.0 menurut wikipedia, sebuah ensiklopedia gratis di internet pengguna C# .Net pada saat ini sudah melebihi pengguna VB.Net. sementara itu jumlah pengguna bahasa pemrograman lain masih berada dibawah jumlah pengguna VB .Net. masih menurut wikipedia, jumlah buku C# yang terjual pun berada dikisaran 2 hingga 3 kali lebih banyak dari jumlah buku VB yang terjual.
Dari informasi ini dapat disimpulkan bahwa C# merupakan bahasa pemrogrman baru yang sedang berkembang dan dapat diterima dengan baik oleh kebanyakan progammer dan kalangan industri. Di Microsoft sendiri, C# merupakan bahasa pemrograman yang digunakan untuk membuat perangkat lunak yang berteknologi .Net dengan demikian dapat diperkirakan bahwa C# akan menjadi bahasa pemrograman yang akan banyak digunakan di masa yang akan datang [12].