BAB II
LANDASAN TEORI

2.1. Al-Qur’an

Al-Qur’an dari segi Bahasa arab merupakan “yang dibaca” atau “bacaan”. Sedangkan menurut istilah pengertian Al-Qur’an adalah kitab suci umat islam yang berisi firman-firman allah SWT, yang diwahyukan kepada nabi muhammad dan membaca nya merupakan ibadah. Al-Qur'an berfungsi sebagai petunjuk/pedoman bagi umat manusia dalam mencapai kebahagiaan hidup di dunia dan akhirat. Sebagai pedoman hidup, isi/kandungan Al-Qur'an terbagi menjadi tiga pembahasan pokok yaitu akidah, ibadah, dan prinsip-prinsip syariat. Di indonesia terdapat beberapa metode pembelajaran Al-Qur’an. Salah satunya dengan melatih membaca huruf-huruf alquran yaitu huruf hijaiyah.

2.1.1. Huruf Hijaiyah

Huruf hijaiyah merupakan huruf yang digunakan di dalam alquran, terdiri dari dua puluh Sembilan huruf, banyak metode pembelajar membaca huruf hijaiyah, salah satunya iqra. Beberapa huruf hijaiyah, yaitu ;

Tabel 2.1 Huruf Hijaiyah

No	Huruf Arab	Nama Huruf
1	ا	Alif
2	ب	Ba
3	ت	Ta
4	ث	Tsa
5	ج	Jim
6	ح	Kha

7	خ	Kho
8	د	Dal
9	ذ	Dzal
10	ر	Ro
11	ز	Za
12	س	Sin
13	ش	Syin
14	ص	Shod
15	ض	Dhod
16	ط	Tho
17	ظ	Dhlo
18	ع	Ain
19	غ	Ghoin
20	ف	Fa
21	ق	Qof
22	ك	Kaf
23	ل	Lam
24	م	Mim

25	ن	Nun
26	و	Wawu
27	ء	Hamzah
28	ه	Ha
29	ي	Ya

2.1.2. Tanda Baca Huruf Hijaiyah

Adapun beberapa Tanda baca huruf hijaiyah yaitu ;

a) Tanda Baca *Fathah* sering disebut baris atas. Artinya letak barisnya di atas huruf hijaiyah.

Misalnya:

ketika tanda baca fathah di letakkan di atas huruf “alif” maka dibaca A.

ketika tanda baca fathah diletakkan di atas huruf “ba” maka dibaca BA. begitu seterusnya.

HURUF HIJAIYYAH	تَ	بَ	اَ
DIBACA	*TA*	*BA*	*A*

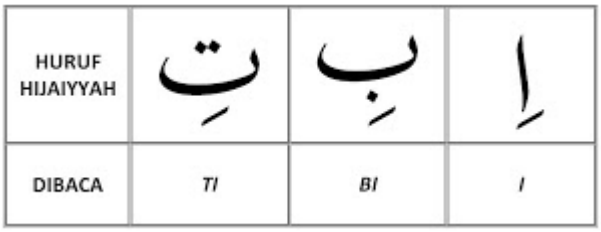
Gambar 2.1 *Fathah*

b) Tanda baca *Kasrah*

sering disebut baris bawah . artinya letak barisnya dibawah huruf hijaiyah.

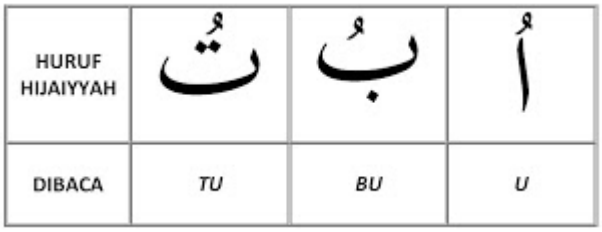
Misalnya :

ketika tanda baca kasrah di letakkan di bawah huruf “alif” maka dibaca I.
ketika tanda baca kasrah diletakkan di bawah huruf “ba” maka dibaca BI. begitu seterusnya.



Gambar 2.2 *Kasrah*

c) Tanda baca *dhammah* sering disebut baris depan. Artinya letak barisnya di depan huruf hijjaiyah.
Misalnya:
Ketika tanda baca *dhammah* diletakan di depan huruf “*alif*” maka dibaca U.
Ketika tanda baca *dhammah* di letakan di depan huruf “ba” maka dibaca u dan begitu seterusnya.



Gambar 2.3 *Dhammah*

2.2. Multimedia

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi, dan video dengan alat bantu dan koneksi sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi. Multimedia sering digunakan dalam dunia hiburan. Selain dari

dunia hiburan, Multimedia juga diadopsi oleh dunia Game.

Multimedia juga dapat diartikan sebagai penggunaan beberapa media yang berbeda dalam menyampaikan informasi berbentuk teks, audio, grafik, animasi, dan video.

2.2.1. Multimedia Interaktif

Multimedia interaktif adalah media yang menggabungkan teks, grafik, video, animasi dan suara. Untuk menyampaikan suatu pesan dan informasi, melalui media elektronik seperti komputer dan perangkat elektronik lainnya.

Pengertian Multimedia Interaktif menurut beberapa ahli dijelaskan sebagai berikut:

Menurut Robin dan Linda (seperti dikutip Benardo, 2011) Multimedia interaktif adalah alat yang dapat menciptakan peresentasi yang dinamis dan interaktif, yang mengkombinasikan teks, grafik, animasi, audio dan gambar video.

Menurut Hofstetter (seperti dikutip Benardo, 2011) Multimedia interaktif adalah pemanfaatan komputer untuk membuat dan menggunakan teks, grafik, audio, gambar bergerak dengan menggunakan link dan tool yang memungkinkan pemakai melakukan navigasi, berinteraksi dan berkomunikasi.

2.2.2. Jenis Multimedia Interaktif

Menurut suyanto (seperti dikutip Bernardo,2011) jenis multimedia interaktif
terbagi menjadi dua bagian, yaitu:

a) Multimedia Interaktif Online

   Multimedia Interaktif Online adalah media interaktif yang cara penyampaiannya melalui jalur/kawat/saluran/jaringan. Contohnya situs Web, Yahoo Messengers, dan lain sebagainya. Jenis media ini termasuk media lini atas, yang komunitas sasarannya luas, dan mencakup masyarakat luas.

b) Multimedia Interaktif Offline

   Multimedia Interaktif Offline adalah media interaktif yang cara penyampaiannya melalui CD interaktif. Media ini termasuk media lini bawah

klarena sasarannya, tidak terlalu luas dan hanya mencakup masyarakat daerah tertentu saja.

2.2.3. Fungsi Multimedia Interaktif

Dalam kegiatan pembelajaran di kelas, multimedia dapat berfungsi sebagai:

a) *Suplemen* (Tambahan)

Multimedia dikatakan *suplemen* (tambahan), apabila guru atau sisa mempunyai kebebasan memilih, apakah akan memanfaatkan multimedia atau tidak untuk materi pembelajaran tertentu.

b) *Komplement* (pelengkap)

Multimedia dikatakan sebagai *komplemen* apabila multimedia tersebut di programkan untuk melengkapi atau menunjang materi pembelajaran yang diterima siswa di kelas.

c) *Subtitusi* (pengganti)

Multimedia dikatakan sebagai *subtitusi* (pengganti) apabila multimedia dapat menggantikan sebagian besar peran guru. Ini dapat menjadi alternatif sebagai sebuah model pembelajaran.

2.2.4. Elemen-elemen Multimedia

Multimedia dapat menyampaikan dan menyebarkan informasi dengan cara baru yang informatif dan efisien. Menurut James A Senn. Multimedia terdiri dari beberapa unsur program yang masing-masing unsur menyediakan fasilitas bagi penggunanya, yaitu:

a) Teks bagian multimedia ini menayangkan tulisan di layar. Teks merupakan cara yang cepat untuk menyampaikan dan menyebarkan informasi, sedangkan teks akan selalu digunakan dalam setiap program.
b) Gambar yang digunakan dalam produksi, produksi multimedia berkisar dari klip art sederhana sampai fotografi. Dengan melihat gambar, akan menghasilkan pengaruh yang kuat dibandingkan hanya sekedar membacanya saja.

c) Film Dengan suatu bagian atau program multimedia, komputer dapat berubah menjadi seperangkat televisi, memungkinkan untuk menghadirkan atau menyaksikan suatu film yang diputar berulang-ulang.

d) Animasi Pada dasarnya, animasi adalah menayangkan gambar-gambar secara bargantian, hingga mata kita menangkap pergantian gambar-gambar sebagai sebuah pergerakan. Kadang-kadang animasi dapat mengekspresikan suatu unsur yang lebih mengena karakteristiknya dibandingkan film.

e) Suara, sebuah animasi akan terasa lebih indah karena suara akan menciptakan suasana yang lebih hidup, menghilangkan rasa jenuh dan menjadi sebuah daya tarik tersendiri bagi pemakai aplikasi tersebut.

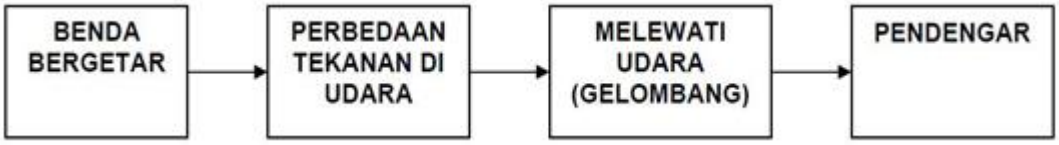
2.2.5. Manfaat Multimedia Interaktif

Penggunaan media dalam pembelajaran dapat membantu memberikan pengalaman yang bermakna kepada siswa, karena penggunaan media dapat mempermudah siswa dalam mengamati pengalaman nyata kemudian diimplementasikan kedalam media pembelajaran interaktif untuk membantu siswa untuk lebih mudah mengingat pelajarannya.

Penggunaan multimedia interaktif pada penelitian ini digunakan untuk membuat tampilan yang menarik yang disukai oleh user.

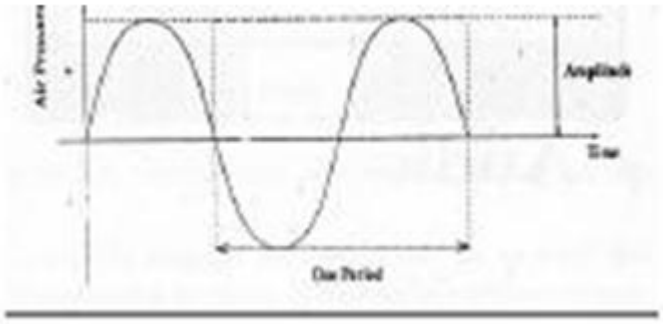
2.2.6. Suara

Suara adalah sesuatu yang dihasilkan oleh getaran yang berasal dari benda bergerak, sesuatu yang menghasilkan bunyi, dan suara adalah sebuah tekanan gelombang udara, maka memiliki nilai kontinu terhadap waktu (*analog*).



Gambar 2.4 Skema Suara

Suara berhubungan erat dengan rasa “mendengar”. Suara/bunyi biasanya merambat melalui udara. Suara/bunyi tidak bisa merambat melalui ruang hampa. Gelombang suara bervariasi dalam tingkatan tekanan suara (*amplitudo*) dan dalam frekuensi. Jumlah waktu yang diperlukan untuk terjadinya suatu getaran atau gelombang dinamakan perioda (T). Sedangkan jumlah gelombang yang terjadi setiap detik dinamakan frekuensi (f) dengan satuan m/dt (Hz).



Gambar 2.5 Gelombang Suara

Suara berkaitan erat dengan :

a. Frekuensi

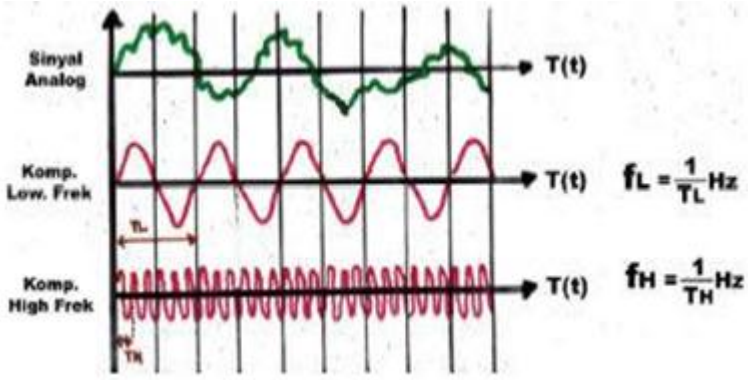
Frekuensi adalah jumlah gelombang yang terjadi setiap detik. Jenis-jenis

frekuensi ada 4, sebagai berikut:

1) *Infra sound*	0 - 20 Hz
2) Pendengaran manusia	20 Hz – 20 KHz
3) *Ultra sound*	20 KHz – 1 GHz
4) *Hyper sound*	1 GHz – 10 THz

Manusia bersuara dengan frekuensi 50 Hz – 10 KHz, Sinyal suara musik memiliki frekuensi 20 Hz – 20 KHz, Sistem multimedia menggunakan suara yang berada dalam jarak pendengaran manusia. Suara yang berada pada jarak pendengaran manusia disebut “*AUDIO*” dan gelombangnya sebagai “*ACCOUSTIC SIGNALS*”. Suara di luar jarak pendengaran manusia dapat

dikatakan sebagai "*NOISE*" (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat didengar manusia).



Gambar 2.6 Signal analog tersdiri dari sebuahg frekuensi

b. Amplitudo

Amplitudo adalah keras lemahnya bunyi atau tinggi rendahnya gelombang. Satuan *amplitudo* adalah *desibel* (db), bunyi dapat merusak telinga jika tingkat volumenya lebih besar dari 85 db dan pada ukuran 130 db akan mampu membuat hancur gendang telinga.

c. Velocity

Velocity adalah kecepatan perambatan gelombang bunyi sampai ke telinga pendengar. Satuan yang digunakan : m/s pada udara kering dengan suhu 20˚ C (68˚ F) m kecepatan rambat suara sekitar 343 m/s.

2.2.7. Speech Recognition

Biometrik, termasuk di dalamnya *speech recognition*, secara umum digunakan untuk identifikasi dan verifikasi. Identifikasi ialah mengenali identitas subyek, dilakukan perbandingan kecocokan antara data *biometric* subyek dalam database berisi record karakter subyek. Sedangkan verifikasi adalah menentukan apakah subyek sesuai dengan apa yang dikatakan terhadap dirinya.

Biometric merupakan suatu metoda untuk mengenali manusia berdasarkan pada satu atau lebih ciri-ciri fisik atau tingkah laku yang unik. *Bionemic recognition* atau biasa disebut dengan Sistem pengenalan *biometric* mengacu pada

identifikasi secara otomatis terhadap manusia berdasarkan psikological atau karakteristik tingkah laku manusia. Ada beberapa jenis teknologi *biometric* antara lain suara (*speech recognition*).

2.2.8. Skema Utama Speech Recognition

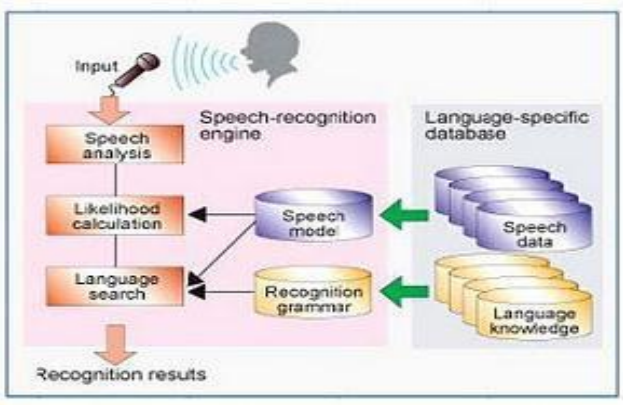
Terdapat 4 langkah utama dalam pengenalan suara :

a) Penerimaan data input

b) Ekstrasi, yaitu penyimpanan data masukan sekaligus pembuatan database untuk template.

c) Pembandingan/pencocokan, yaitu tahap pencocokan data baru dengan data suara (pencocokan tata bahasa) pada template.

d) Validasi identitas pengguna.



Gambar 2.7 Skema *Speech Recognition*

Secara umum, speech recognizer memproses sinyal suara yang masuk dan menyimpannya dalam bentuk digital. Hasil proses digitalisasi tersebut kemudian dikonversi dalam bentuk spektrum suara yang akan dianalisa dengan membandingkannya dengan template suara pada database sistem.

2.2.9. Implementasi Speech Recognition

Hardware yang dibutuhkan dalam implementasi *Speech Recognition*

a) Sound card

Merupakan perangkat yang ditambahkan dalam suatu Komputer yang

fungsinya sebagai perangkat input dan output suara untuk mengubah sinyal elektrik, menjadi analog maupun menjadi digital.

b) Microphone

perangkat input suara yang berfungsi untuk mengubah suara yang melewati udara, air dari benda orang menjadi sinyal elektrik.

c) Komputer atau Komputer Server

Dalam proses suara digital menterjemahkan gelombang suara menjadi suatu simbol biasanya menjadi suatu nomor biner yang dapat diproses lagi kemudian diidentifikasikan dan dicocokan dengan database yang berisi berkas suara agar dapat dikenali.

Metode *spech recognition* pada penelitian ini digunakan untuk mendeteksi suara dan menampilkan hasil apa yang diucapkan user.

2.3. MFCC (Mel Frequency Cepstrum Coefficient)

MFCC (*Mel Frequency Cepstrum Coefficients*) merupakan salah satu metode yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognation* maupun *speech recognation*. Metode ini digunakan untuk melakukan *feature extraction*, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter. Keunggulan dari metode ini adalah:

a) Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara. Atau dengan kata lain, mampu menangkap informasi-informasi penting yang terkandung dalam sinyal suara.

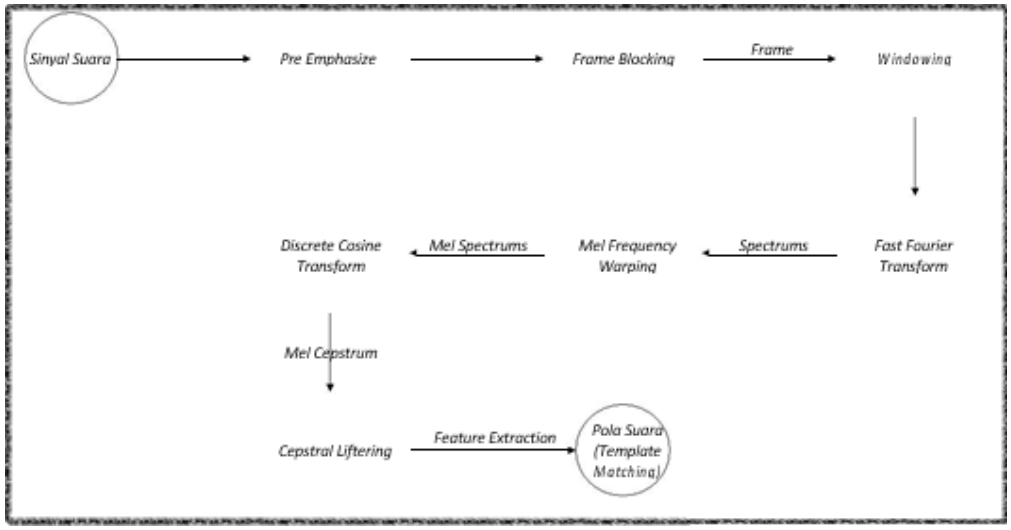
b) Menghasilkan data seminimal mungkin tanpa menghilangkan informasi-informasi penting yang ada.

c) Mengadaptasi organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara.

Perhitungan yang dilakukan MFCC (*Mel Frequency Cepstrum Coefficients*) menggunakan dasar dari perhitungan *short-term analysis*. Hal ini dilakukan mengingat sinyal suara yang bersifat *quasistationary*. Pengujian yang dilakukan untuk periode waktu yang cukup pendek (sekitar 0 sampai 10 milidetik) akan menunjukan karakteristik sinyal suara yang *stationary*. Tetapi bila dilakukan dalam periode waktu yang lebih panjang karakteristik sinyal suara akan terus

berubah sesuai dengan kata yang diucapkan.

MFCC (*Mel Frequency Cepstrum Coefficients*) *feature extraction* sebenarnya merupakan adaptasi dari sistem pendengaran manusia dimana sinyal suara akan di-filter secara linear untuk frekuensi rendah (dibawah 1000 Hz) dan secara logaritmik untuk frekuensi tinggi [3]. Berikut ini blok diagram untuk proses yang dilakukan pada metode MFCC (*Mel Frequency Cepstrum Coefficients*):



Gambar 2.8 Block Diagram Alur Proses MFCC

2.3.1. Konversi Analog Menjadi Digital

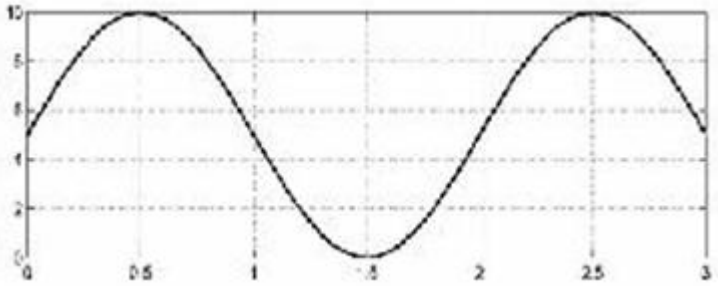
Sinyal–sinyal yang natural pada umumnya, seperti sinyal suara merupakan sinyal continue dimana memiliki nilai yang tidak terbatas. Sedangkan pada komputer, semua sinyal yang dapat diproses oleh komputer hanyalah sinyal *discrete* atau sering dikenal dengan istilah digital signal. Agar sinyal natural dapat diproses oleh komputer kita harus dapat mengubah data sinyal *continue* menjadi *discrete*. Hal itu dapat melalui tiga proses, diantaranya adalah proses sampling data, proses kuantisasi, dan proses pengkodean.

Proses sampling adalah suatu proses untuk mengambil data sinyal continue untuk setiap periode tertentu. Dalam melakukan proses sampling data, berlaku aturan *Nquist*, yaitu bahwa frekuensi sampling (*sampling rate*) minimal harus dua kali lebih tinggi dari frekuensi maksimum yang akan disamplingkan. Jika sinyal sampling kurang dari dua kali frekuensi maksimum sinyal yang akan

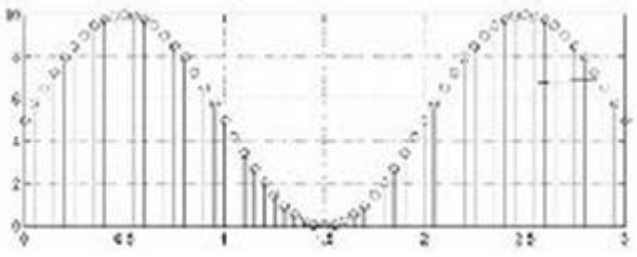
disampling, maka akan timbul efek *aliasing*. *Aliasing* adalah suatu efek dimana sinyal yang dihasilkan memiliki frekuensi yang berbeda dengan sinyal aslinya,

Proses kuantisasi adalah proses untuk membulatkan nilai data kedalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu. Semakin banyak level yang dipakai maka semakin akurat pula data sinyal yang disimpan tetapi akan menghasilkan ukuran data yang besar dan proses lama.

Proses pengkodean adalah proses pemberian kode untuk tiap tiap data sinyal yang telah terkuantisasi berdasarkan level yang ditempati.



Gambar 2.9 Sinyal Sinus



Gambar 2.10 Sinyal Sinus Setelah Tersampling

2.3.2. Pre-Empasis

Pre-emphasis dilakukan untuk memperbaiki signal dari gangguan *noise*, sehingga dapat meningkatkan tingkat akurasi dari proses *feature extraction*. Default dari nilai alpha yang digunakan dalam proses *pre-emphasis filtering* adalah 0,97.

Bentuk yang paling umum digunakan dalam *pre-emphasis filter* adalah sebagai berikut:

$$y[n] = s[n] - \alpha s[n-1]$$

Rumus 2.1 ***Pre-emphasis***

Keterangan:

$y[n]$ = signal hasil *pre-emphasis filter*

$s[n]$ = signal sebelum *pre-emphasis filter*

2.3.3. Frame Blocking

Hasil perekaman suara merupakan sinyal analog yang berada dalam domain waktu yang bersifat variant time, yaitu suatu fungsi yang bergantung waktu. Oleh karena itu sinyal tersebut harus dipotong-potong dalam slot-slot waktu tertentu agar dapat dianggap invariant. Setiap potongan sinyal suara tersebut disebut frame.

Untuk menghitung jumlah frame digunakan rumus:

$$Jumlah\ frame = \left(\frac{I - N}{M}\right) + 1$$

Rumus 2.2 ***Frame Blocking***

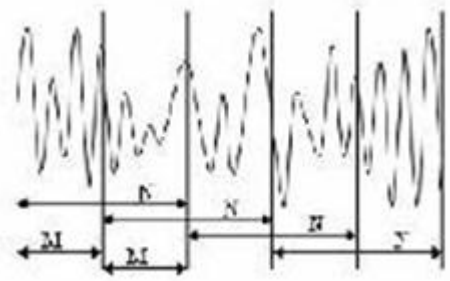
Keterangan:

I = *Sample rate*

N = *Sample point* (*Sample rate* * waktu framing (s))

M = N/2

Potongan *frame* digambarkan seperti gambar dibawah ini:



Gambar 2.11 ***Framing***

2.3.4. Windowing

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

Window Hamming:

$$w(n) = 0{,}54 - 0.46\cos(\frac{2\pi n}{N-1})$$

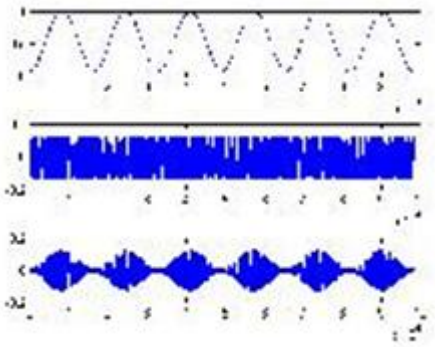
Rumus 2.3 *Windowing*

Keterangan:

$W_{ham}(n)$ = *Window Hamming*

n = Sinyal ke [1 ... n]

N = *Sample Point*



Gambar 2.12 Window Haming

2.3.5. Fast Fourier Transform

FFT (*Fast Fourier Transform*) adalah teknik perhitungan cepat dari DFT (*Discrete Fourier Transfrom*). FFT (*Fast Fourier Transform*) adalah DFT (*Discrete Fourier Transfrom*) dengan teknik perhitungan yang cepat dengan memanfaatkan sifat periodikal dari *transformasi fourier*. Perhatikan definisi dari FFT (*Fast Fourier Transform*):

$$S[k] = \sum_{n=0}^{N-1} s[n]\, e^{-\frac{j2\pi nk}{N}}, 0 \leq k \leq N-1$$

Rumus 2.4 *Fast Fourier Transform*

Keterangan :

N = jumlah sampel yang akan diproses (*N* N)

S(n) = nilai sampel *signal*

K = variable frekuensi *discrete*

2.3.6. Mel Frequency Warping (FilterBank)

Mel Frequency Warping umumnya dilakukan dengan menggunakan f*ilterbank*. Filterbank menggunakan representasi konvolusi dalam melakukan filter terhadap sinyal. Konvolusi dapat dilakukan dengan melakukan multiplikasi antara spektrum sinyal dengan koeffisien *filterbank*.

Berikut ini adalah rumus yang digunakan dalam perhitungan *filterbank*.

$$Y[i] = \sum_{j=1}^{n} S[j] H_i[j]$$

Rumus 2.5 *Filterbank (1)*

Keterangan:

N = jumlah magnitude spectrum

S[j] = magnitude spectrum pada frekuensi j

$H_i[j]$ = koefisien *filterbank* pada frekuensi j ($1 \leq i \leq M$)

M = jumlah channel dalam *filterbank*

Untuk mendapatkan Hi digunakan rumus :

$$H_i = \frac{2529 * \log\left(1 + \frac{f}{700}\right)}{\frac{Si}{2}}$$

Rumus 2.6 Koefisien *Filterbank*

2.3.7. Discrete Cosine Transform

Proses ini merupakan langkah akhir dari *feature extraction*. Hasil dari DCT (*Discrete Cosine Transform*) ini adalah fitur-fitur yang dibutuhkan untuk melakukan proses analisa terhadap pengenalan suara tersebut. Menggunakan rumus:

$$\tau_n = \sum_{k=0}^{k} (logS_k) \cos[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}]$$

Rumus 2.7 *Discrete Cosine Transform*

Keterangan:

S_k = keluaran dari proses *filterbank* pada indeks *k*

K = jumlah koefisien yang diharapkan

Koefisien ke-nol dari DCT(*Discrete Cosine Transform*) pada umumnya akan dihilangkan, walaupun sebenarnya mengindikasikan energi dari *frame* sinyal tersebut. Hal ini dilakukan karena, berdasarkan penelitian-penelitian yang pernah dilakukan, koefisien ke nol ini tidak *reliable* terhadap *speaker recognation*. Tetapi koefisien ke nol dari DCT(*Discrete Cosine Transform*) sangatlah berguna dalam menentukan *end point* dari suatu suku kata maupun kata. Hal ini disebabkan karena pada *end point* dari suatu suku kata maupun kata mempunyai energi yang lebih rendah daripada *point-point*.

2.3.8. Cepstral Liftering

Cepstral, hasil dari proses utama MFCC (*Mel Frequency Cepstrum Coefficients*) *feature extraction*, memiliki beberapa kelemahan. *Low-order* dari *cepstral coefficients* sangat sensitif terhadap *spectral slope*, sedangkan bagian *high order*-nya sangat sensitif terhadap *noise*. Oleh karena itu, maka *cepstral liftering* menjadi salah satu standar teknik yang diterapkan untuk meminimalisasi sensitifitas tersebut.

Cepstral liftering dapat dilakukan dengan mengimplementasikan fungsi window terhadap *cepstral features*.

$$w[n] = \left\{1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right)\right\}$$

Rumus 2.8 *Cepstral Liftering*

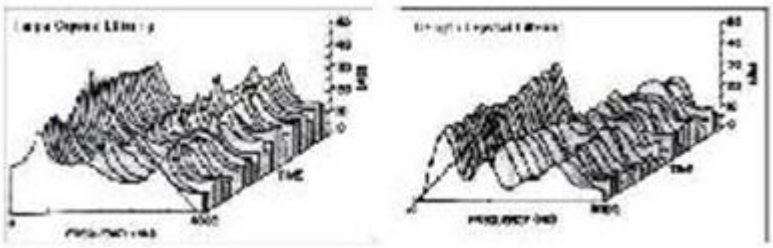
Keterangan:

L = jumlah *cepstral coefficients*

n= index dari *cepstral coefficients*

Cepstral liftering menghaluskan spektrum hasil dari main processor sehingga dapat

digunakan lebih baik untuk *pattern matching*.



Gambar 2.13 Perbandingan Spectrum Dan Tanpa Cepstral Liftering

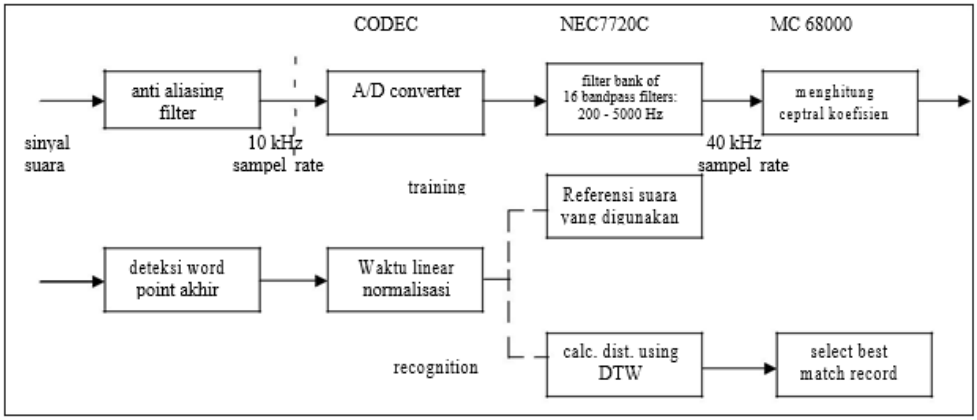
Berdasarkan gambar diatas dapat dilihat bahwa pola spektrum setelah dilakukannya *cepstral liftering* lebih halus daripada spektrum yang tidak melalui tahap *cepstral liftering*. Dalam beberapa jurnal dikatakan bahwa *cepstral liftering* dapat meningkatkan akurasi dari aplikasi pengenalan suara, baik *speaker recognition*, maupun *speech recognition.*

2.4. Dynamic Time Warping (DTW)

Satu masalah yang cukup rumit dalam pengenalan suara (*voice recognition*) adalah proses perekaman yang terjadi seringkali berbeda durasinya, biarpun kata atau kalimat yang diucapkan sama. Bahkan untuk satu suku kata yang sama atau vocal yang sama seringkali proses perekaman terjadi dalam durasi yang berbeda. Sebagai akibatnya proses pencocokan antara sinyal uji dengan sinyal *record* suara di *database* seringkali tidak menghasilkan nilai yang optimal.

Sebuah teknik yang cukup popular di awal perkembangan teknologi pengolahan sinyal pengenalan suara (*voice recognition*) adalah dengan memanfaatkan sebuah teknik *Dynamic Time Warping* (DTW) yang juga lebih

dikenal sebagai *dynamic programming*. Teknik ini ditujukan untuk mengakomodasi perbedaan waktu antara proses perekaman saat pengujian dengan suara yang tersedia pada *record* suara di *database*. Prinsip dasarnya adalah dengan memberikan sebuah rentang '*steps*' dalam ruang (dalam hal ini sebuah frame-frame waktu dalam sampel, frame-frame waktu dalam *database*) dan digunakan untuk mempertemukan lintasan yang menunjukkan *local match* terbesar (kemiripan) antara time frame yang lurus. Total `*similarity cost*' yang diperoleh dengan algoritma ini merupakan sebuah indikasi seberapa bagus *sample* suara dan *record* suara di *database* ini memiliki kesamaan, yang selanjutnya akan dipilih *best-matching record* suara di *database*.



Gambar 2.14 Diagram *Dynamic time warping*

2.5. Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar (*Smartphone*) dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti begerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile* dan *Nvidia.*

Bebrapa pengertian lain dari *Android,* yaitu :

Merupakan *platform* terbuka (*Open Source*) bagi para programmer untuk

membuat atau mengembangkan aplikasi. Merupakan sistem operasi yang dibeli *Google Inc* dari *Android Inc*.

Bukan bahasa pemrograman, akan tetapi hanya menyediakan lingkungan hidup atau *runtime environment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk *device* dengan sistem memori yang kecil.

2.5.1. Versi Android

Banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versi, fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah HTC Dream yang dirilis pada tanggal 22 Oktober 2008. Beberapa uraian versi *android* seperti dibawah ini :

a) ***Android* versi 1.1**

Pada 9 Maret 2009, *Google* merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan Email.

b) ***Android* versi 1.5 (*Cupcake*)**

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan *Android* dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c) ***Android* versi 1.6 (*Donut*)**

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang

memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang diintegrasikan pada CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-spech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

d) ***Android* versi 2.0/2.1 (*Eclair*)**

Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah mengoptimalkan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML 5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, *digital zoom*, dan *Bluetooth* 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps*-aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset Android*, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam sistem operasi *Android* adalah Shazam, *Backgrounds* dan *WeatherBug*. Sistem operasi *Android* dalam situs internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh MySpace dan Facebook.

e) ***Android* versi 2.2 (*Froyo* : *Frozen Yoghurt*).**

Pada 20 Mei 2010, *Android* versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi dengan V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel dan kemampuan *auto update* dalam aplikasi Android Market.

f) ***Android* versi 2.3 (*Gingerbread*)**

Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb, equalization, headphone virtualization,* dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC) dan dukungan jumlah kamera yang lebih dari satu.

g) ***Android* versi 3.1 (*Honeycomb*)**

Android Honeycomb dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet peratam yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

h) ***Android* versi 4.0 (*Ice Cream Sandwich*)**

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline* dan berbagi informasi dengan menggunakan NFC.

i) ***Android* versi 4.1 (*Jelly Bean*)**

Pada tanggal 27 Juni 2012 Google mengumumkan *Android* versi 4.1 (*Jelly Bean*) dalam konferensi Google I/O berdasarkan kernel Linux 3.0.31. *Jelly Bean* adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuak pengguna (*User Interface*). Pembaruan ini diwujudkan dalam proyek *Butter*, perbaikan ini termasuk antisipasi sentuh, *triple buffering*, perpanjangan waktu vsync dan peningkatan *frame rate* hingga 60fps untuk menciptakan UI yang lebih halus. Perangkat pertama yang menggunakan sistem operasi ini adalah

tablet Nexus 7, yang dirilis pada 13 Juli 2012.

j) ***Android* versi 4.4 (*KitKat*)**

Android versi 4.4 (KitKat) direncanakan akan dirilis pada bulan Oktober 2013. Setelah sebelumnya beredar rumor bahwa Android versi berikutnya setelah *Jelly Bean* diperkirakan akan diberi nomor 5.0 dan dinamai Key Lime Pie.

2.5.2. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan Android bukanlah linux, karena dalam Android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada Android hanya terdapat beberapa pelayanan yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel linux menyediakan *driver* layar, kamera, *keypad*, *WiFi*, *Flash Memory*, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2.5.3. Library

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

a) *Media Library* untuk memutar dan merekam berbagai macam format audio dan video.
b) *Surface Manager* untuk mengatur hak akses layer dari berbagai aplikasi.
c) *Graphic Library* termasuk didalamnya *SGL* dan *OpenGL*, untuk tampilan 2D dan 3D.
d) SQLite untuk mengatur relasi database yang digunakan pada aplikasi.
e) *SSl* dan *WebKit* untuk browser dan keamanan *internet*.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit* (NDK).

2.5.4. Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi Android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *User Interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi Android adalah sebagai berikut:

a) *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan ”*Backstack*“ untuk navigasi penggunaan.
b) *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.
c) *Resuource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan *file layout*.
d) *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat Android berada.
e) *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *status bar*.

2.5.5. Application Layer

Puncak dari diagram arsitektur Android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam Android *runtime* dengan menggunakan kelas dan service yang tersedia pada *Framework* aplikasi.

Lapisan aplikasi Android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada Android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.5.6. Kelebihan Android

Sudah banyak platform untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada *platform* lain, Android adalah yang pertama menggabungkan hal seperti berikut :

a) Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar royality. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh *vendor* manapun.

b) Arsitektur komponen dasar Android terinspirasi dari teknologi *internet Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.

c) Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database* SQL, *browser* dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.

d) Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.

e) Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh Flash menyatu dalam 3D menggunakan OpenGL memungkinkan membuat aplikasi maupun *game* yang berbeda.

Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin *virtual Dalvik,* sehingga kode program *portable* antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard,* layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.6. Perancangan Berorientasi Objek

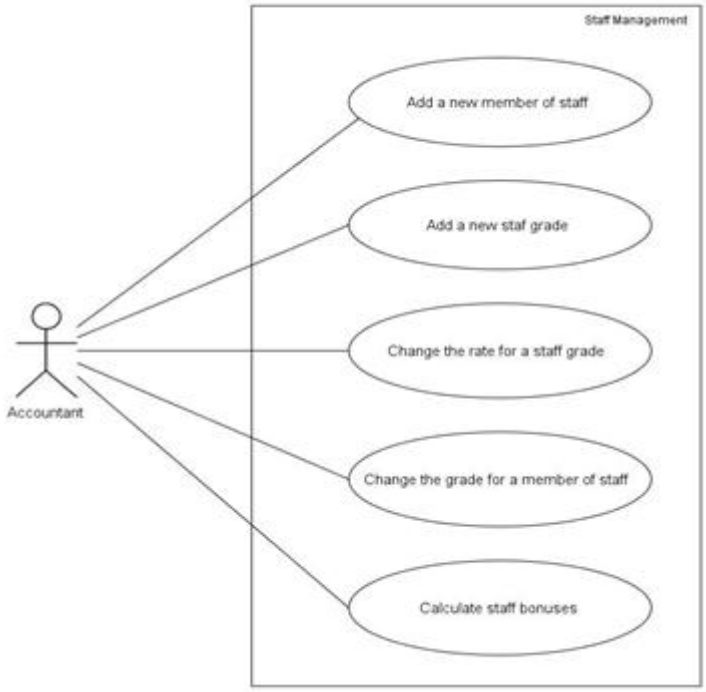
Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan *software* yaitu *Rational Software,* telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysist Design*(OOAD).

2.6.1. Unified Modelling Language (UML)

UML dalam sebuah bahasa untuk menentukan visualisasi, konstruksi, dan mendokumentasikan artifacts dari sistem *software*, untuk memodelkan bisnis, dan sistem *non-software* lainnya. UML merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, konstruksi dan mendokumentasikan *artifact* yang terdapat dalam sistem. *Artifact* adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. *Artifact* dapat berupa model, deskripsi atau *software*.

2.6.1.1. Use Case Diagram

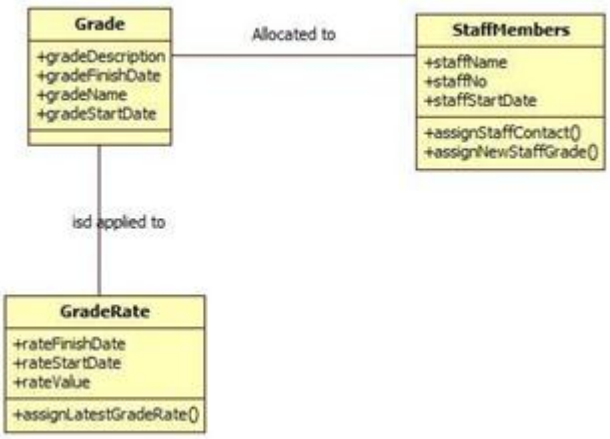
Use Case Diagram menjelakan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (Aktor). Diagram ini menunjukan fungsionalitas suatu sistem yang berinteraksi dengan dunia luar. *Use Case Diagram* dapat digunakan selama proses analisis untuk menagkap *requitment* sistem dan untuk memahami bagaimana sistem bekerja.



Gambar 2.15 Diagram *Use Case*

2.6.1.2. Class Diagram

Class Diagram menjelaskan dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan prilaku sistem. Selama tahap desain, *class diagram* berperan dalam menagkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.



Gambar 2.16 *Class Diagram*

2.6.1.3. Behavior Diagram

Behavior diagram dapat dikelompokan menjadi tiga diagram, yaitu :

a) *Statechart Diagram*

Statechart Diagram berfungsi untuk memodelkan prilaku dinamis satu kelas satu objek.

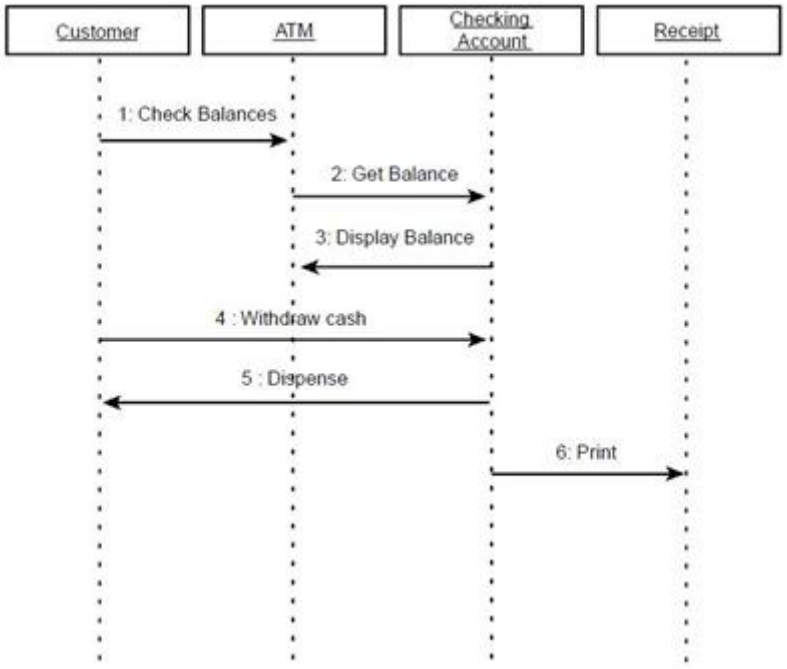
b) *Activity Diagram*

Activity Diagram memodelkan alur kerja (work flow) sebuah proses bisnis dan urutan aktifitas dalam suatu proses.

c) *Interaction Diagram*

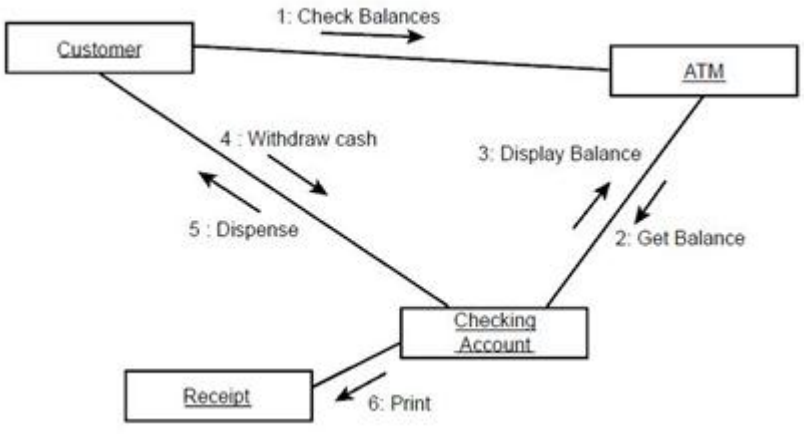
Interaction Diagram dibagi menjadi dua model diagram yaitu :

1. *Sequence Diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus bersosialisasi dengan *use case*. Sequence diagram, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu dalam *use case*.



Gambar 2.17 *Sequence Diagram*

2) *Collaboration Diagram* melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan (*relationship*) antar objek, sedangkan *sequence diagram* menekankan pada urutan kejadian. Dalam *collaboration diagram* terdapat beberapa objek, *link*, dan *message*.



Gambar 2.18 *Collaboration Diagram*

2.6.1.4. Implementation Diagram

I*mplementation* diagram dibagi menjadi dua diagram, yaitu :

a) *Component Diagram* menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-komponen *software*, seperti *source code*, *binary code*, dan komponen tereksekusi (*execute components*).
b) *Deployment Diagram* memperlihatkan pemetaan *software* kepada *hardware*. Dimana akan berjalan (di *server/multitier*, *standalone* atau lainnya), dan menggambarkan model koneksi dan kemampuan jaringan dan hal lainya yang bersifat fisik.

2.7. Tool yang akan digunakan

Adapun beberapa *Tools* yang digunakan dalam penelitian ini sebagai berikut ;

2.7.1. Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

a) *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
b) *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
c) *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya popular

adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.7.2. Audacity

Audacity adalah aplikasi editor sinyal suara yang dapat digunakan untuk berbagai macam kebutuhan, seperti merekam sinyal suara, menghitung data sinyal suara, dan lain – lain.

2.8. Metode Pengujian Sistem

Metode pengujian sistem untuk mengetahui efektifitas dari *software* yang digunakan selain memberikan kesempatan kepada pengguna untuk mengoperasikan dan melakukan pengecekan terhadap laporan yang dihasilkan melalui *software*. Metode pengujian sistem terdiri dari Pengujian alpha yang menerapkan pengujian *Whitebox,* Pengujian *Blackbox* dan pengujian beta [10].

2.8.1. Skenario pengujian alpha

Skenario pengujian alpha menjelaskan pengujian terhadap sistem yang dibangun, dimana dilakukan pengujian *white box* dan pengujian *black box.* Metode yang digunakan dalam pengujian *white box* ini adalah metode *Basis Path*. Metode

Basis Path mengijinkan pendesain kasus uji untuk membuat perkiraan lojik yang kompleks dari desain prosedural dan menggunakan perkiraan ini untuk mendefinisikan aliran eksekusi.

2.8.1.1. Pengujian *Blackbox*

Black Box Testing adalah tipe pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerjanya pada bagian *internal*. Para *tester* memandang perangkat lunak sebagai “kotak hitam” yang tidak penting di lihat isinya, tapi cukup dikenali proses *testing* di bagian luar, dimana *testing* ini banyak berkonsentrasi pada bagaimana proses berjalannya perangkat lunak. Pada *black box testing* atau pengujian fungsional, kondisi pengujian dikembangkan berdasarkan fungsionalitas dari program atau sistem yang akan diuji, oleh karena itu, penguji membutuhkan informasi mengenai data input dan output yang diamati, tetapi tidak mengetahui bagaimana program atau sistem tersebut bekerja atau tidak perlu mengetahui bagaimana struktur internal dari program tersebut melakukan eksekusi [14].

Penguji berfokus pada pengujian fungsionalitas dari program terhadap spesifikasi, juga memfokuskan pada *requirement* fungsi dari perangkat lunak, pengujian ini merupakan komplenetari dari pengujian *whitebox*. Pengujian *whitebox* dilakukan terlebih dahulu pada proses pengujian, sedangkan pengujian *blackbox* dilakukan pada tahap akhir dari pengujian perangkat lunak.

Proses yang terdapat dalam proses pengujian *blackbox* antara lain sebagai berikut:

a) Pembagian kelas data untuk pengujian setiap kasus yang muncul pada pengujian *whitebox*.
b) Analisis batasan nilai yang berlaku untuk setiap data.