BAB 2

TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence* atau *AI*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu *entitas* buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.[1]

2.1.1 Sejarah Kecerdasan Buatan

Artificial Intelligence (AI) atau kecerdasan buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris pertama kali mengusulkan adanya pengujian untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil pengujian tersebut kemudian dikenal dengan *Turing Test*, di mana mesin tersebut menyamar seolah-olah sebagai seseorang di dalam suatu permainan yang mampu memberikan respon terhadap serangkaian pertanyaan yang diajukan. Turing beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia).

Kecerdasan buatan atau *Artificial Intelligence* (AI) itu sendiri dimunculkan oleh seorang professor dari *Massachusetts Institute of Technology* yang bernama

John McCarthy pada tahun 1956 pada *Dartmouth Conference* yang dihadiri oleh para peneliti *Artificial Intelligence* (AI). Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

2.1.2 Definisi Kecerdasan Buatan

Kecerdasan buatan merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peranan komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian

pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu pada *Artificial Intelligence* (AI) akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin yang pintar.

Pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain :

1. Sudut pandang kecerdasan

   Kecerdasan buatan akan membuat mesin menjadi ‘cerdas’ (mampu berbuat seperti apa yang dilakukan oleh manusia).

2. Sudut pandang penelitian

   Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

3. Sudut pandang bisnis

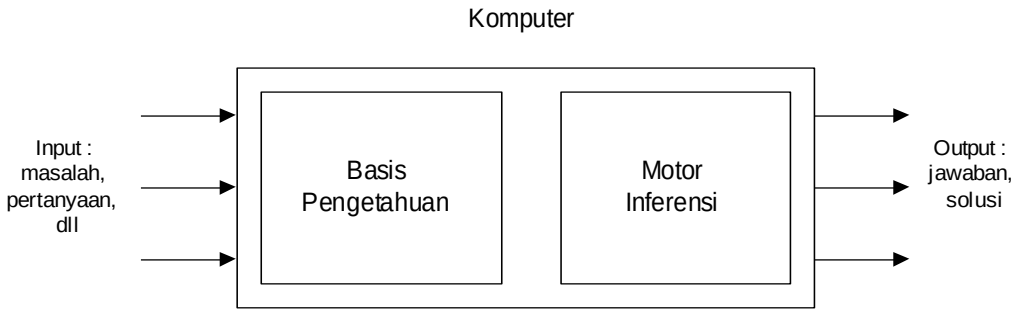
   Kecerdasan buatan adalah kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalah-masalah bisnis.

4. Sudut pandang pemrograman

   Kecerdasan buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

Untuk membuat aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu :

a. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
b. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 2.1 Penerapan Konsep Kecerdasan Buatan di Komputer

2.1.3 Kecerdasan Buatan dan Kecerdasan Ilmiah

Kecerdasan alamiah adalah kecerdasan yang dimiliki oleh manusia. Jika dibandingkan dengan kecerdasan buatan, ada beberapa keuntungan kecerdasan buatan dibanding kecerdasan alamiah, yaitu:

1. Lebih permanen.

   Kecerdasan alamiah akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak diubah.

2. Memberikan kemudahan dalam duplikasi dan penyebaran.

   Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama, dan keahlian itu juga tidak akan

pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

3. Relatif lebih murah dan kecerdasan alamiah.

   Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

4. Konsisten dan teliti.

   Hal ini disebabkan karena kecerdasan buatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.

5. Dapat didokumentasi.

   Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.

6. Dapat mengerjakan beberapa *task* dengan lebih cepat dan lebih baik dibanding manusia.

Sedangkah, keuntungan kecerdasan alamiah dibanding kecerdasan buatan:

1. Bersifat lebih kreatif.

   Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.

2. Dapat melakukan proses pembelajaran secara langsung, sementara AI harus mendapatkan masukan berupa simbol dan representasi.
3. Fokus yang luas sebagai referensi untuk pengambilan keputusan, sebaiknya AI menggunakan fokus yang sempit.

Komputer dapat digunakan untuk mengumpulkan informasi tentang obyek, kegiatan (*events*), proses dan dapat memproses sejumlah besar informasi dengan lebih efisien dari yang dapat dikerjakan manusia. Namun di sisi lain, manusia dengan menggunakan insting dapat melakukan hal yang sulit diprogram pada komputer, yaitu kemampuan mengenali (*recognize*) hubungan antara hal-hal tersebut, menilai kualitas dan menemukan pola yang menjelaskan hubungan tersebut.

2.1.4 Lingkup Kecerdasan Buatan pada Aplikasi Komersial

Dewasa ini, kecerdasan buatan juga memberikan konstribusi yang cukup besar di bidang manajemen. Adanya sistem pendukung keputusan dan sistem informasi manajemen juga tidak terlepas dari andil *Artificial Intelligence* (AI). Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran *Artificial Intelligence* (AI). Karakteristik ‘cerdas’ sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. *Artificial Intelligence* (AI) tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain. Irisan antara psikologi dan *Artificial Intelligence* (AI) melahirkan sebuah area yang dikenal dengan nama *cognition* & *psycolinguistics*. Irisan antara

teknik elektro dengan *Artificial Intelligence* (AI) melahirkan berbagai ilmu, seperti: pengolahan citra, teori kendali, pengenalan pola dan robotika.

Adanya irisan penggunaan *Artificial Intelligence* (AI) di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan *Artificial Intelligence* (AI) menurut disiplin ilmu yang menggunakannya. Untuk memudahkan hal tersebut, maka pengklasifikasian lingkup *Artificial Intelligence* (AI) didasarkan pada *output* yang diberikan, yaitu pada aplikasi komersial (meskipun sebenarnya *Artificial Intelligence* itu sendiri bukan merupakan medan komersial).

Lingkup utama dalam kecerdasan buatan adalah:

1. Sistem Pakar (*Expert System*)

   Disini, komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian, komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.

2. Pengolahan Bahasa Alami (*Natural Language Processing*).

   Dengan pengolahan bahasa alami ini diharapkan *user* dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.

3. Pengenalan Ucapan (*Speech Recognition*).

   Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer menggunakan suara.

4. Robotika & Sistem Sensor (*Robotics & Sensory Systems*).

5. *Computer Vision*, mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer-aided Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game Playing*.

Seiring dengan perkembangan teknologi, muncul beberapa teknologi yang juga bertujuan untuk membuat agar komputer menjadi cerdas sehingga dapat menirukan kerja manusia sehari-hari. Teknologi ini juga mampu mengakomodasi adanya ketidakpastian dan ketidaktepatan data *input*. Dengan didasari pada teori himpunan, maka pada tahun 1965 muncul Logika Fuzzy. Kemudian pada tahun 1975, John Holland mengatakan bahwa setiap *problem* berbentuk adaptasi (alami maupun buatan) secara umum dapat diformulasikan dalam terminologi genetika. Algoritma genetika ini merupakan simulasi proses evolusi Darwin dan operasi genetika atas kromosom.

2.2 Pengenalan Suara pada komputer

Pengenalan suara (Speech Recognition) merupakan salah satu kajian utama dalam kecerdasan Buatan. Pengenalan suara ini bertujuan agar komputer dapat mengeti apa yang diucapkan oleh manusia. Salah satu aplikasinya adalah *voice command* , dengan aplikasi ini diharapkan komputer dapat mengerti perintah yang diberikan manusia padanya. Secara garis besar, tugasnya adalah bagaimana komputer dapat mengerti apa yang diperintahkan oleh manusia melalui ucapan.

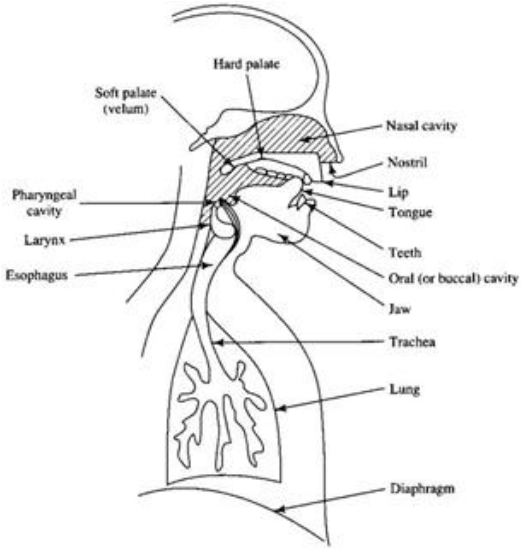
2.3 Pengenalan Suara pada Manusia

Pada sistem pengenalan suara oleh manusia terdapat tiga organ penting yang saling berhubungan yaitu : telinga yang berperan sebagai *transduser* dengan menerima sinyal masukan suara dan mengubahnya menjadi sinyal syaraf, jaringan syaraf yang berfungsi mentransmisikan sinyal ke otak, dan otak yang akan mengklasifikasi dan mengidentifikasi informasi yang terkandung dalam sinyal masukan.

2.3.1 Proses Produksi Suara pada Manusia

Proses produksi suara pada manusia dapat dibagi menjadi tiga buah proses fisiologis, yaitu : pembentukan aliran udara dari paru-paru, perubahan aliran udara dari paru-paru menjadi suara, baik *voiced*, maupun *unvoiced* yang dikenal dengan istilah *phonation*, dan artikulasi yaitu proses modulasi atau pengaturan suara menjadi bunyi yang spesifik.

Organ tubuh yang terlibat pada proses produksi suara adalah : paru-paru, tenggorokan (*trachea*), laring (*larynx*), faring (*pharynx*), pita suara (*vocal cord*), rongga mulut (*oral cavity*), rongga hidung (*nasal cavity*), lidah (*tongue*), dan bibir (*lips*), seperti dapat dilihat pada gambar 2.2 dibawah ini



Gambar 2.2 Organ Tubuh manusia

Organ tubuh ini dapat dikelompokkan menjadi tiga bagian utama, yaitu : *vocal tract* (berawal di awal bukaan pita suara atau glottis, dan berakhir di bibir), *nasal tract* (dari velum sampai nostril), dan *source generator* (terdiri dari paru-paru, tenggorokan, dan *larynx*). Ukuran *vocal tract* bervariasi untuk setiap individu, namun untuk laki-laki dewasa rata-rata panjangnya sekitar 17 cm. Luas dari vocal tract juga bervariasi antara 0 (ketika seluruhnya tertutup) hingga sekitar 20 cm^2. Ketika *velum*, organ yang memiliki fungsi sebagai pintu penghubung antara *vocal tract* dengan *nasal tract*, terbuka, maka secara akustik *nasal tract* akan bergandengan dengan vocal tract untuk menghasilkan suara *nasal*.

Aliran udara yang dihasilkan dorongan otot paru-paru bersifat konstan. Ketika pita suara dalam keadaan berkontraksi, aliran udara yang lewat membuatnya bergetar. Aliran udara tersebut dipotong-potong oleh gerakan pita suara menjadi sinyal pulsa yang bersifat *quasi-periodik*. Sinyal pulsa tersebut kemudian mengalami modulasi frekuensi ketika melewati *pharynx*, rongga mulut

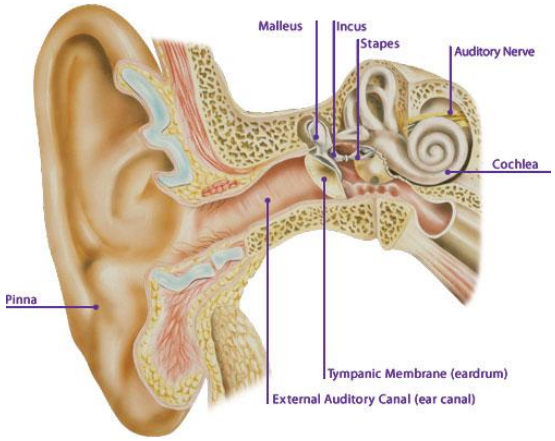
ataupun pada rongga hidung. Sinyal suara yang dihasilkan pada proses ini dinamakan sinyal *voiced*. Namun, apabila pita suara dalam keadaan relaksasi, maka aliran udara akan berusaha melewati celah sempit pada permulaan *vocal tract* sehingga alirannya menjadi turbulen, proses ini akan menghasilkan sinyal *unvoiced*. Ketika sumber suara melalui *vocal tract*, kandungan frekuensinya mengalami modulasi sehingga terjadi resonansi pada *vocal tract* yang disebut *formants*. Apabila sinyal suara yang dihasilkan adalah sinyal *voiced*, terutama vokal, maka pada selang waktu yang singkat bentuk *vocal tract* relative konstan (berubah secara lambat) sehingga bentuk *vocal tract* dapat diperkirakan dari bentuk spektral sinyal *voiced*.

Aliran udara yang melewati pita suara dapat dibedakan menjadi *phonation*, bisikan, *frication*, kompresi, vibrasi ataupun kombinasi diantaranya. *Phonated excitation* terjadi bila aliran udara dimodulasi oleh pita suara. *Whispered excitation* dihasilkan oleh aliran udara yang bergerak cepat masuk ke dalam lorong bukaan segitiga kecil antara *arytenoids cartilage* di belakang pita suara yang hampir tertutup. *Frication excitation* dihasilkan oleh desakan di vocal tract. *Compression excitation* dihasilkan akibat pelepasan udara melalui vocal tract yang tertutup dengan tekanan tinggi. *Vibration excitation* disebabkan oleh udara yang dipaksa memasuki rusang selain pita suara, khususnya lidah. Suara yang dihasilkan oleh *Phonated excitation* disebut *voiced.* Suara yang dihasilkan oleh *Phonated excitation* ditambah *frication* disebut *mixed voiced*, sedangkan yang dihasilkan oleh selain itu disebut *unvoiced*. Karakteristik suara tiap individu

bersifat unik karena terdapat perbedaan dalam hal panjang maupun bentuk *vocal tract*.

2.3.2 Karakteristik Telinga

Telinga terbagi menjadi tiga bagian, yaitu bagian luar, tengah, dan dalam.



Gambar 2.3 Organ Telinga

Pinna, sebagai bagian luar telinga, berfungsi sebagai corong, untuk mengumpulkan sinyal suara menuju *auditory canal* sehingga dapat memberikan kesan arah sinyal suara yang diterima.

Auditory canal adalah struktur berbentuk pipa lurus sepanjang 2,7 cm, dengan diameter sekitar 0,7 cm, yang pada bagian ujungnya terdapat selaput membrane, yaitu gendang telinga. Membran ini merupakan pintu masuk telinga bagian tengah, yaitu ruangan berisi udara dengan volume sebesar 2 cm^3, yang terdiri dari tiga buah tulang, yaitu *malleus* (martil), *incus* (landasan), dan *stapes* (sanggurdi). Bagian ini terhubung dengan tenggorokan melalui *Eustachian tube*. Getaran pada gendang telinga ditransmisikan ke *malleus* melalui *incus*, dan *stapes*, yaitu membentuk *oval window*.

Telinga bagian dalam (*labyrinth*) memiliki tiga bagian, yaitu *vestibule* (ruang pintu masuk), *semicular canal*, dan *cochlea*. *Vestibule* terhubung dengan telinga bagian tengah melalui dua jalur, yaitu *oval window*, dan *round window*. Keduanya tertutup untuk mencegah keluarnya cairan yang mengisi telinga telinga bagian dalam. Pada *cochlea*, yang berstruktur seperti rumah siput, terdapat syaraf pendengaran. Syaraf ini memanjang sampai ke *basilar membrane*. Pada bagian atas *basilar membrane* terdapat *organ of corty* yang memiliki empat baris sel rambut (sekitar 3×10^4 sel seluruhnya).

2.3.3 Proses Pendengaran

Proses pendengaran pada telinga manusia dijelaskan sebagai berikut :

1. Sinyal suara memasuki saluran telinga dan variasi tekanan yang dihasilkannya menekan gendang telinga. Karena sisi bagian dalam dari gendang telinga mempunyai tekanan yang nilainya dijaga konstan maka gendang telinga akan bergetar.
2. Getaran dari gendang telinga disalurkan pada tiga rangkaian tulang yaitu; *martil*, *incus* dan *stapes*. Mekanisme ini dirancang untuk mengkopel variasi suara dari udara luar ke telinga bagian dalam. Karena luas permukaan penampang yang ditekan *stapes* lebih kecil dari luas penampang gendang telinga maka tekanan suara yang sampai ke telinga bagaian dalam bertambah besar.
3. Cairan pada *cochlea* bergetar dengan frekuensi yang sama dengan gelombang yang datang. *Basilar membrane* kemudian memisahkan sinyal berdasarkan frekuensinya. *Basilar membrane* berstruktur kuat dan panjang di daerah

sekitar *oval window* namun bersifat lentur pada bagian ujungnya. Frekuensi resonansi yang dihasilkan membrane tersebut berbeda sepanjang dimensi *basilar membrane*. Dimana resonansi frekuensi tinggi terjadi pada bagian bagian *basilar membrane* yang berada dekat dengan *oval window*, sedangkan resonansi frekuensi rendah terjadi pada daerah ujung lainnya. Syaraf yang berada pada mambran kemudian mendeteksi posisi terjadinya resonansi yang juga akan menentukan frekuensi suara yang datang. Ukuran dari *basilar membrane* rata-rata sekitar 35 mm. Dari ukuran panjang tersebut dapat dihasilkan 10 resolusi frekuensi, sehingga pada setiap 3.5 mm panjang membran terdapat 1 oktaf frekuensi resonansi.

2.3.4 Sinyal Suara Ucapan

Sinyal suara ucapan manusia dapat dipandang sebagai sinyal yang berubah lambat terhadap waktu (*slowly time varying* sinyal), jika diamati pada selang waktu yang singkat yaitu 5-100 ms. Pada selang waktu tersebut, katakteristik sinyal suara ucapan dapat dianggap stasioner. Untuk selang waktu yang lebih panjang (dengan orde 0.2 detik atau lebih), karakteristik sinyal berubah untuk merefleksikan suara berbeda yang diucapkan.

2.3.5 Klasifikasi berdasarkan sinyal eksitasi

Berdasarkan sinyal eksitasi yang dihasilkan pada proses produksi suara, sinyal suara ucapan dapat dibagi menjadi tiga bagian yaitu *silence*, *unvoiced*, dan *voiced*. Berikut ini adalah penjelasannya :

1. Sinyal *silence* : sinyal pada saat tidak terjadi proses produksi suara ucapan, dan sinyal yang diterima oleh pendengar dianggap sebagai bising latar belakang.

2. Sinyal *unvoiced* : terjadi pada saat pita suara tidak bergetar, dimana sinyal eksitasi berupa sinyal random.

3. Sinyal *voiced* : terjadi jika pita suara bergetar, yaitu pada saat sinyal eksitasi berupa sinyal pulsa kuasi-periodik. Selama terjadinya sinyal voiced ini, pita suara bergetar pada frekuensi fundamental – inilah yang dikenal sebagai *pitch* dari suara tersebut.

2.3.6 Analisis Sinyal Ucapan

Informasi yang terdapat di dalam sebuah sinyal ucapan dapat dianalisis dengan berbagi cara. Beberapa peneliti telah membagi beberapa level pendekatan untuk menggambarkan informasi tersebut, yaitu level akustik, fonetik, fonologi, morfologi, sintatik, dan semantik.

1. Level Akustik

Sinyal ucapan merupakan variasi tekanan udara yang dihasilkan oleh sistem artikulasi. Untuk menganalisa aspek-aspek akustik dari sebuah sinyal ucapan, dapat dilakukan dengan transformasi dari bentuk sinyal ucapan menjadi sinyal listrik dengan menggunakan tranduser seperti *microphone*, telepon, dan sebagainya. Setelah melalui berbagai pengolahan sinyal digital, maka akan di peroleh informasi yang menunjukkan sifat-sifat akustik dari sinyal ucapan tersebut yang meliputi frekuensi fundamental (F_0), intensitas, dan distribusi energi spektral.

2. Level Fonetik

Level ini menggambarkan bagaimana suatu sinyal suara diproduksi oleh organ-organ di dalam tubuh manusia.

3. Level Fonologi

Di dalam level ini, dikenal istilah fonem yang merupakan unit terkecil yang membentuk sebuah kalimat atau ucapan. Deskripsi ini memuat informasi durasi, intensitas, dan *pitch* dari fonem-fonem yang membangun kalimat tersebut.

2.3.7 Intonasi Sebagai Aspek Akustik Sinyal Ucapan

Intonasi (prosodi) sebagai aspek akustik sinyal suara sangat membantu di dalam mengidentifikasi setiap segmen akustik dengan fonem. Setiap fonem dihasilkan terutama oleh sistem vokal selama artikulasi yang selanjutnya mempengaruhi dinamika spektrum spektral suara (dalam hal ini formant). Pengucapan suatu kata dapat secara substansial bervariasi di dalam intonasinya mempengaruhi idetitas kata. Fonem dapat menjadi panjang atau pendek, keras atau lemah, dan memiliki pola *pitch* (nada) yang bervariasi.

Fenomena intonasi dapat direpresentasikan ke dalam beberapa level antara lain adalah sebagai berikut :

1. Level Akustik

   Terdiri atas beberapa komponen penting yaitu Frekuensi Fundamental (F_0), amplitudo, dan durasi sinyal.

2. Level Perseptual

   Merepresentasikan fenomena intonasi sebagaimana yang didengar oleh pendengarnya. Beberapa komponennya antara lain *pitch* (nada), keras atau lemahnya suara, dan panjang atau pendeknya suara.

3. Level Bahasa (Linguistik)

   Merepresentasikan fenomena prosodi ke dalam bentuk simbol atau tanda. Beberapa komponennya antara lain bunyi (tone), intonasi, dan aspek tekanan.

Menonjolkan suku kata yang mendapat tekanan terhadap suku kata yang lain yang tidak mendapat tekanan adalah fungsi utama sebuah intonasi (prosodi). Suku kata yang mendapat tekanan menjadi lebih panjang, lebih intens, dan memiliki pola F0 yang menyebabkan mereka lebih menonjol dibanding suku kata lainnya.

Parameter-parameter yang diperlukan dalam pengidentifikasian suara manusia antara lain adalah :

1. Pitch

Pitch digunakan sebagai standar tinggi-rendah dari sebuah *tone* atau suara. Sinyal suara umumnya merupakan proses secara fisis yang terdiri dari dua bagian: yaitu sebagai hasil dari sumber suara (pita suara) dan sebagai hasil dari penyaringan (oleh lidah, bibir, dan gigi). menganalisa *pitch* berarti mencoba untuk menangkap frekuensi dasar sumber bunyi dari keseluruhan proses pengucapan suara. Frekuensi dasar sendiri merupakan frekuensi yang dominan yang dikeluarkan oleh sumber bunyi. Frekuensi dasar merupakan parameter paling kuat

untuk mengetahui korelasi bagaimana suatu suara diterima oleh pendengar ditinjau dari segi intonasi dan tekanan suaranya.

2. Formant

Frekuensi fundamental dikenal juga dengan F0 yang koheren dalam bentuk transisi formant F1, F2, dan sebagainya. Komponen frekuensi dominan yang mengkarakterisasi fonem-fonem yang berhubungan dengan komponen frekuensi resonansi dari sistem vokal didefinisikan sebagai formant. Suara yang terucapkan, secara khusus adalah vokal, biasanya memiliki 3 buah formant dan seringkali disebut sebagai formant kesatu, kedua, dan ketiga, dimulai dengan komponen frekuensi terendah. Ketiganya selalu dituliskan sebagai F1, F2, dan F3. Formant 4 dan formant 5 dbutuhkan untuk mendapatkan nilai parameter formant yang lebih detail karena bila sinyal suara yang kita olah hanya memiliki formant yang kurang dari 3 buah, maka dapat dipastikan analisa terhadap data tersebut akan gagal.

2.3.8 Durasi Fonem

Salah satu komponen terpenting di dalam intonasi adalah durasi sinyal. Setiap fonem yang memberikan kontribusi dalam menentukan pola intonasi suatu kalimat. Durasi fonem ini sangat dipengaruhi oleh tekanan dan kecepatan bicara. Durasi sebuah fonem vokal sangat dipengaruhi oleh tekanan, sementara durasi sebuah konsonan umumnya memiliki variasi tekanan yang lebih kecil.

Menurut Douglas O'Shugnessy (1.200) suatu ucapan dalam percakapan melibatkan 150-250 kata permenit, termasuk jeda yang masing-masing rata-rata sepanjang 6-50 ms. Durasi fonem bervariasi karena faktor seperti gaya bicara

(membaca atau bercakap-cakap). Durasi suku kata umumnya sekitar 200ms dengan vokal yang mendapat tekanan sekitar 130 ms dan fonem lain sekitar 70ms. Durasi fonem bermacam-macam untuk fonem yang berbeda karakteristiknya.

2.3.9 Durasi dan Kekerasan Suara

Bagaimana kekerasan suara dari sebuah suara yang bersifat impulsif menyamai kekerasan suara dari suara yang diberikan secara kontinyu pada tingkatan yang sama. Beberapa eksperimen telah menetapkan bahwa telinga merata-ratakan energi suara sekitar lebih dari 200ms, maka kekerasan suara yang bersifat impulsif akan bertambah dengan durasi hingga mencapai nilai tersebut. Dengan kata lain, tingkat kekerasan suara akan bertambah 10 dB ketika durasi bertambah dengan faktor 10. Dari sini dapat diketahui bahwa berapa lamanya durasi yang dilakukan membantu dalam adaptasi pendengaran terhadap kekerasan suara, terutama untuk suara yang sifatnya impulsif atau muncul tidak kontinyu.

2.3.10 Durasi dan Pitch

Lamanya durasi dapat mempengaruhi persepsi *pitch*. Kebergantungan *pitch* terhadap durasi mengikuti prinsip ketikpastian akustik! Berdasarkan pengamatan yang dilakukan Rossing dan Houtsma pada tahun 1986, ketika durasi *pitch* jatuh hingga di bawah 25 ms, *pitch* dirasakan berubah, walaupun batasan ini berbeda untuk beberapa pengamat.

2.3.11 Durasi dan Timbre

Durasi dari sinyal suara membedakan panjang pendeknya sinyal suara dengan domain waktu. Dalam timbre musikal, lamanya durasi dapat membagi nada ke dalam dua jenis yaitu : nada kontinyu dan nada transien. Persepsi timbre

dalam suatu permainan musik yang melibatkan banyak alat musik dipengaruhi oleh durasinya. Seorang pendengar yang diminta untuk menebak jenis alat musik akan menebak dengan benar untuk alat musik yang dimainkan dengan durasi yang lebih lama dibandingkan dengan alat musik yang dimainkan hanya sesaat (transien).

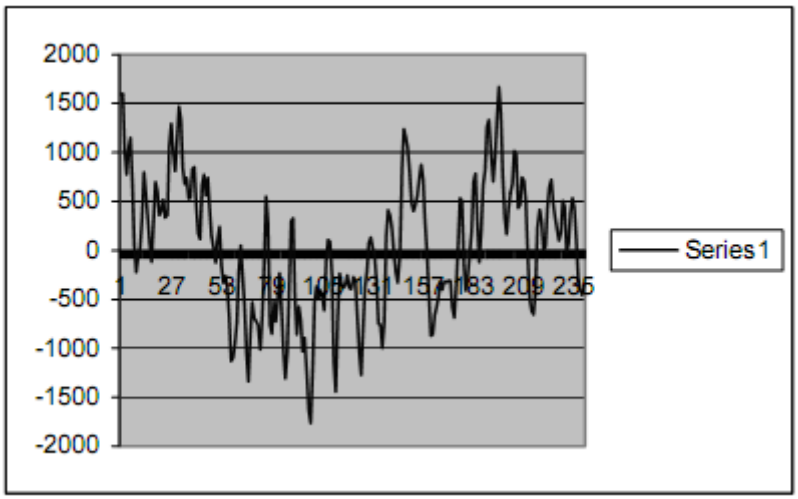
2.3.12 Intensitas Suara

Intensitas bunyi menentukan keras lemahnya suara pada bagian tertentu dari suatu kalimat. Telinga kita sangat peka dan dapat mendeteksi intensitas-intensitas suara dalam orde 10-13 W/m^2. Ini setara dengan gerakan selaput telinga sebesar 10-12 m. Intensitas suara minimum yang masih dapat didengar dinamakan ambang pendengaran (*threshold of hearing*). Intensitas suara biasanya dinyatakan dalam desibel di atas ambang pendengaran karena kekerasan suara (*loudness*) kira-kira adalah sebanding dengan logaritma dari intensitas. Pedoman nol desibel untuk intensitas suara sudah ditentukan standarnya yaitu pada 10-12W/m2 pada 1000 Hz (yaitu ambang pendengaran pada 1000Hz).

2.4 MFCC (*Mel Frequency Cepstrum Coefficients*)

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[2]. Keunggulan dari metode ini adalah :

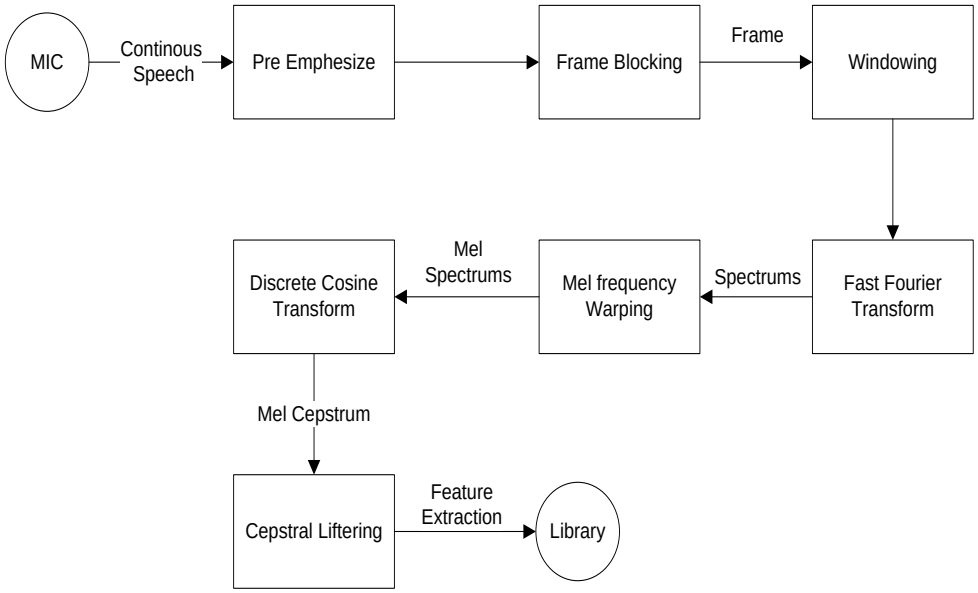
1. Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara. Atau dengan kata lain, mampu menangkap informasi-informasi penting yang terkandung dalam sinyal suara.
2. Menghasilkan data seminimal mungkin tanpa menghilangkan informasi-informasi penting yang ada.
3. Mengadaptasi organ pendengaran manusia dalam melakukan persepsi terhadap sinyal suara.



Gambar 2.4 Contoh Sinyal suara huruf vokal o pada satu frame pria

Perhitungan yang dilakukan MFCC 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 *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. Berikut ini blok diagram untuk MFCC :



Gambar 2.5 Blok Diagram MFCC

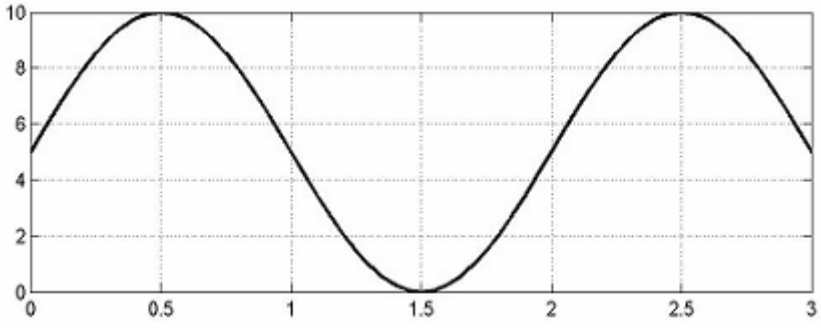
2.4.1 Konversi Analog Menjadi Digital

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.

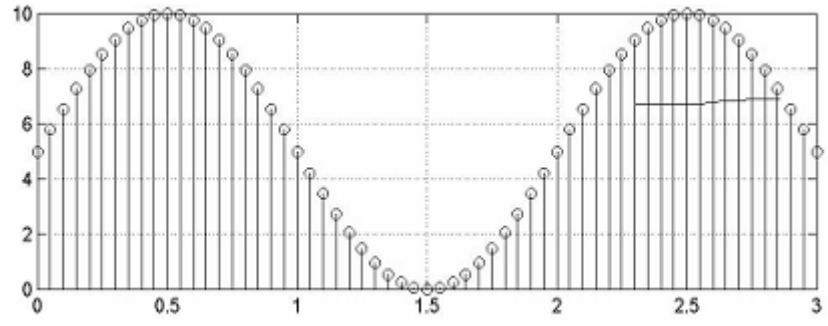
Proses sampling adalah suatu proses untuk mengambil data sinyal *continue* untuk setiap periode tertentu. Dalam melakukan proses samplingdata, berlaku aturan Nquist, yaitu bahwa frekuensi sampling (sampling rate) minimal harus dua kali lebih tinggi dari frekuensi maksimum yang akan disamplingkan. Jika 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 2.6 Sinyal Sinus



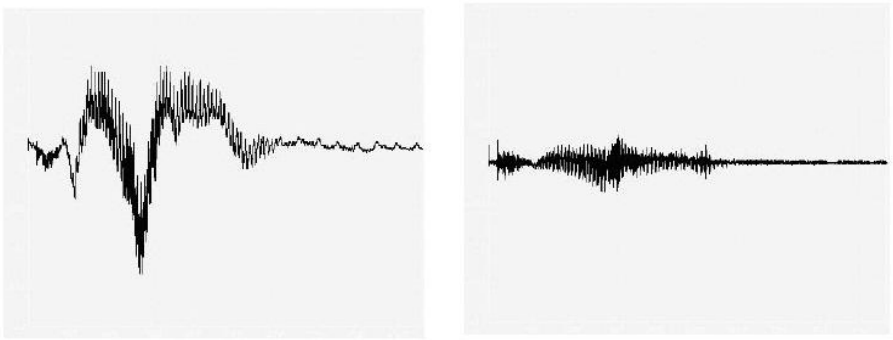
Gambar 2.7 Sinyal sinus Setelah tersampling

2.4.2 *Pre-emphasis Filtering*

Pre-emphasis filtering merupakan salah satu jenis filter yang sering digunakan dalam sebuah sinyal diproses lanjut. Filter ini mempertahankan frekuensi tinggi pada sebuah spektrum, yang umumnya tereliminasi pada saat proses produksi suara.

Tujuan dari *pre-emphasis filter* ini adalah

1. Mengurangi *noise ratio* pada sinyal, sehingga dapat meningkatkan kualitas sinyal.
2. Menyeimbangkan spektrum dari *voice sound* pada saat memproduksi *voiced sound*, glotis manusia menghasilkan sekitar -12dB octave slope**.** namun ketika energi akustik tersebut dikeluarkan melalui bibir, terjadi peningkatan sebesar +6 dB. Sehingga sinyal yang terekam oleh *microphone* adalah sekitar -6dB octave slope.



Gambar 2.8 Perbandingan Sinyal Tanpa dan Sinyal dengan Pre-emphasis

Perhatikan perbedaan pada frekuensi domain akibat diimplementasikannya *pre-emphasis filter*. Pada gambar 2.8 di atas tampak bahwa distribusi energi pada setiap frekuensi menjadi lebih seimbang setelah diimplementasikan *pre-emphasis filter*.

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

$$H\,(z) = 1 - \alpha z^{-1} \tag{1}$$

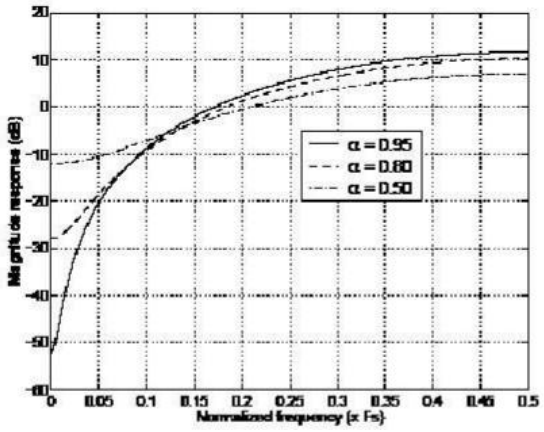
Dimana $0.9 \leq \alpha \leq 1.0$ dan $\alpha \in R$. formula diatas dapat diimplementasikan sebagai *filter order differentiator*, sebagai berikut

$$y\,[n] = s\,[n] - \alpha\, s[n-1] \tag{2}$$

$$y[n] = signal\ hasil\ pre-emphasize\ filter$$

$$s[n] = signal\ sebelum\ pre-emphasize\ filter$$

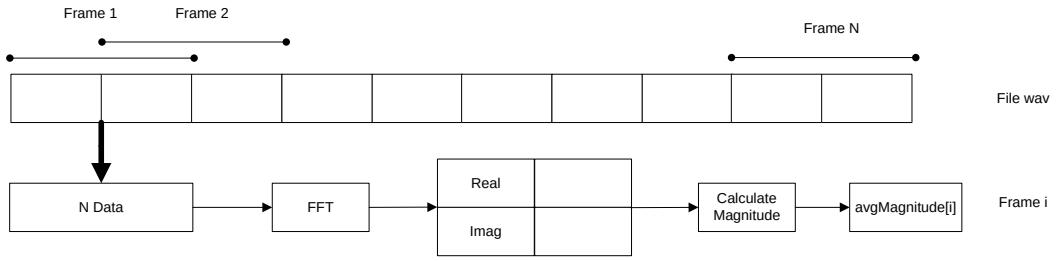
Pada umumnya nilai α paling sering digunakan adalah antara 0.9 sampai dengan 1.0 . Magnitude response (dB scale) untuk nilai α yang berbeda dapat dilihat pada gambar berikut ini



Gambar 2.9 Magnitude Response dari pre-emphasis

2.4.3 *Frame Blocking*

Karena sinyal suara terus mengalami perubahan akibat adanya pergeseran artikulasi dari organ produksi vokal, sinyal harus diproses secara *short segments* (*short frame*). Panjang frame yang biasanya digunakan untuk pemrosesan sinyal adalah antara 10-30 milidetik. Panjang frame yang digunakan sangat mempengaruhi keberhasilan dalam analisa spektral. Di satu sisi, ukuran dari frame harus sepanjang mungkin untuk dapat menunjukan resolusi frekuensi yang baik. Tetapi di sisi lain, ukuran frame juga harus cukup pendek untuk dapat menunjukan resolusi waktu yang baik.



Gambar 2.10 Diagram Blok Proses Penentuan Frame

Proses *framing* ini dilakukan terus sampai seluruh sinyal dapat terproses. Selain itu, proses ini umumnya dilakukan secara *overlapping* untuk setiap framenya. Panjang daerah overlapp yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang frame.

Untuk menghitung jumlah frame digunakan rumus :

$$jumlah\ frame = \left(\frac{I-N}{M}\right) + 1) \qquad (3)$$

Dimana :

$$N = frame\ size = sample\ rate * Ts(detik)$$

$$M = N/2$$

2.4.4 Windowing

Proses *framing* dapat menyebabkan terjadinya kebocoran spektral (*spectral leakage*) atau aliasing. Efek ini dapat terjadi karena rendahnya jumlah *sampling rate*, ataupun karena proses *frame blocking* dimana menyebabkan sinyal menjadi *discontinue*. Untuk mengurangi kemungkinan terjadinya kebocoran spektral, maka hasil dari proses framing harus melewati proses windowing.

Ada banyak fungsi window, w(n). Sebuah fungsi window yang baik harus menyempit pada bagian main lobe, dan melebar pada bagian side lobe-nya.

Berikut ini adalah representasi fungsi window terhadap sinyal suara yang diinputkan.

$$x(n) = x_1(n)\, w\,(n) \qquad (4)$$

$$n = 0,1, \dots, \; N-1$$

X(n) = nilai sampel sinyal

$x_i(n)$ = nilai sampel dari frame sinyal ke i

$w(n)$ = fungsi window

N= frame size, merupakan kelipatan 2

Setiap fungsi window mempunyai karakteristik masing-masing. Diantara berbagai fungsi window tersebut , Blackman window menghasilkan sidelobe level yang paling tinggi (kurang lebih -58dB), tetapi fungsi ini juga menghasilkan noise paling besar (kurang lebih 1.73 BINS). Oleh karena itu fungsi ini jarang sekali digunakan baik untuk *speaker recognation*, maupun *speech recognation.*

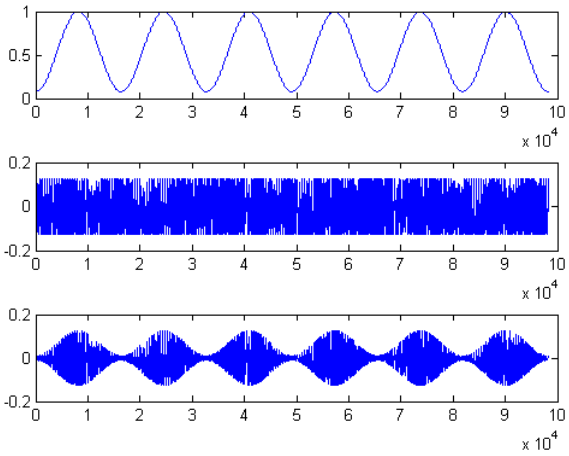
Fungsi window rectangle adalah fungsi window yang paling mudah untuk diaplikasikan. Fungsi ini menghasilkan noise yang paling rendah yaitu sekitar

1.00 BINS. Tetapi sayangnya fungsi ini memberikan sidelobe level yang paling rendah. Sidelobe level yang rendah tersebut menyebabkan besarnya kebocoran spektral yang terjadi dalam proses *feature extraction.*

Fungsi window yang paling sering digunakan dalam aplikasi speaker recognation 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 :

$$W_{ham}(n) = 0.52 - 0.46\cos\left[\frac{2\pi n}{(N-1)}\right] \; 0 \leq n \; \leq N-1 \qquad (5)$$

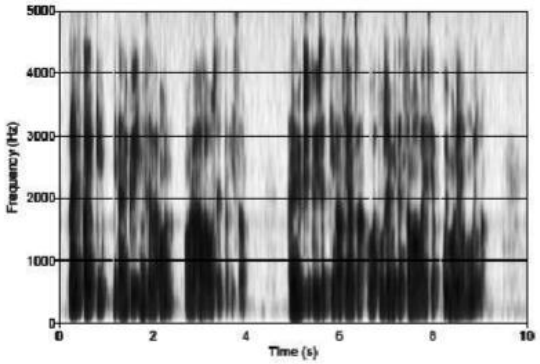


Gambar 2.11 Window Hamming

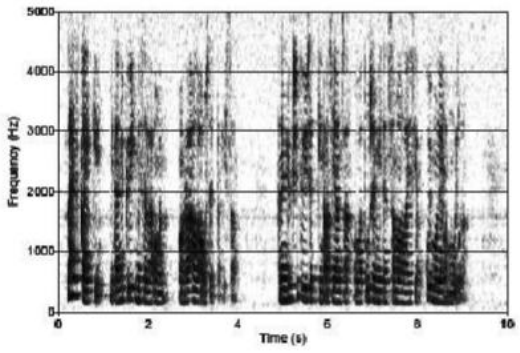
2.4.5 Analisis Fourier

Analisis fourier adalah sebuah metode yang memungkinkan untuk melakukan analisa terhadap *spectral properties* dari sinyal yang diinputkan. Representasi dari *spectral properties* sering disebut sebagai spectogram.

Dalam spectogram terdapat hubungan yang erat antara waktu dan frekuensi. Hubungan antara frekuensi dan waktu adalah hubungan berbanding terbalik. Bila resolusi waktu yang digunakan tinggi, maka resolusi frekuensi yang dihasilkan akan semakin rendah. Kondisi seperti ini akan menghasilkan narrowband spectogram. Sedangkan wideband spectogram adalah kebalikan dari narrowband spectogram



Gambar 2.12 Wideband Spectogram



Gambar 2.13 Narrowband Spectogram

Inti dari transformasi fourier adalah menguraikan sinyal kedalam komponen-komponen bentuk sinus yang berbeda-beda frekuensinya. Dalam gambar diatas menunjukan ada tiga gelombang sinus dan superposisisnya. Sinyal

semula yang periodik dapat diuraikan menjadi beberapa komponen bentuk sinus dengan frekuensi yang berbeda. Jika sinyal semula tidak periodik maka transformasi fouriernya merupakan fungsi frekuensi yang kontinue, artinya merupakan penjumlahan bentuk sinus dari segala frekuensi. Jadi dapat disimpulkan bahwa tansformasi fourier merupakan representasi frekuensi domain dari suatu sinyal. Representasi ini mengandung informasi yang tepat sama dengan kandungan informasi dari sinyal semula.

2.4.5.1 Discrete Fourier Transform (DFT)

DFT merupakan perluasan dari transformasi fourier yang berlaku untuk sinyal-sinyal diskrit dengan panjang yang terhinggga. Semua sinyal periodik terbentuk dari gabungan sinyal-sinyal sinusoidal yang menjadi satu yang dalam perumusannya dapat ditulis

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-\frac{j2\pi nk}{N}},\ 0 \leq k \leq N-1 \qquad (6)$$

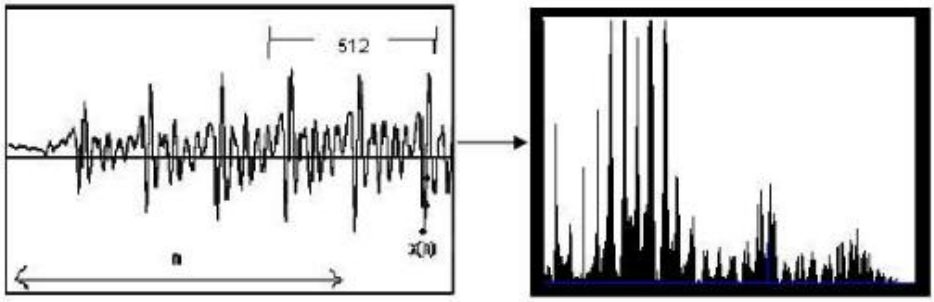
N = jumlah sampel yang akan diproses

S(n) = nilai sampel sinyal

K = variable frekuensi discrete, dimana akan bernilai (k=N/2, k ϵ N)

Dengan rumusan diatas, suatu sinyal suara dalam domain waktu dapat kita cari frekuensi pembentuknya. Hal ini lah tujuan dari penggunaan analisa Fourier pada data suara, yaitu untuk mengubah data dari domain waktu menjadi data spektrum di domain frekuensi. Untuk pemrosesan sinyal suara, hal ini sangatlah menguntungkan karena data pada domain frekuensi dapat diproses dengan mudah

dibandingkan data pada domain waktu, karena pada domain frekuensi, keras lemahnya suara tidak seberapa berpengaruh.



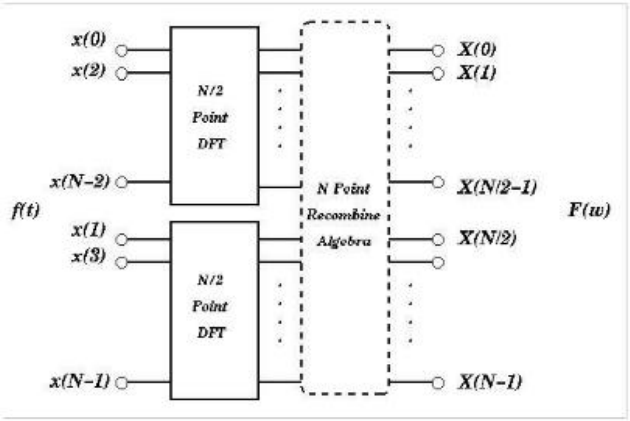
Gambar 2.14 Domain Waktu menjadi Domain Frekuensi

Untuk mendapatkan spektrum dari sebuah sinyal dengan DFT diperlukan N buah sample data berurutan pada domain waktu, yaitu data x[m] sampai dengan x[m+N-1]. Data tersebut dimasukan dalam fungsi DFT maka akan dihasilkan N buah data. Namun karena hasil dari DFT adalah simetris, maka hanya N/2 data yang diambil sebagai spektrum.

2.4.5.2 Fast Fourier Transform (FFT)

Perhitungan DFT secara langsung dalam komputerisasi dapat menyebabkan proses perhitungan yang sangat lama. Hal itu disebabkan karena dengan DFT, dibutuhkan N^2 perkalian bilanngan kompleks. Karena itu dibutuhkan cara lain untuk menghitung DFT dengan cepat. Hal itu dapat dilakukan dengan menggunakan algoritma Fast Fourier Transform (FFT) dimana FFT menghilangkan proses perhitungan yang kembar dalam DFT. Algoritma DFT hanya membutuhkan $N\ log_2 N$ perkalian kompleks.

Jumlah Sample sinyal yang akan diinputkan ke dalam algoritma ini harus merupakan kelipatan dua (2^M). Algoritma Fast Fourier Transform dimulai dengan membagi sinyal menjadi dua bagian, dimana bagian pertama berisi nilai sinyal suara pada indeks waktu genap, dan sebagian yang lain berisi nilai sinyal suara pada indeks waktu ganjil. Visualisasi dari proses ini dapat dilihat pada gambar dibawah. Setelah itu akan dilakukan analisis Fourier (*recombine algebra*) untuk setiap bagiannya. Proses pembagian sinyal suara tersebut terus dilakukan sampai didapatkan dua seri nilai sinyal suara.



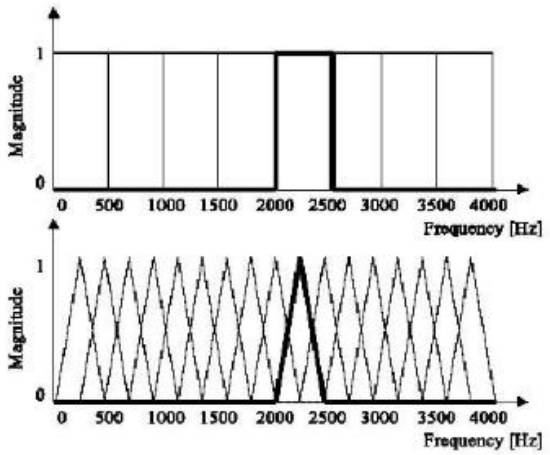
Gambar 2.15 Pembagian Sinyal Suara Menjadi Dua Kelompok

Algoritma recombine (DFT) melakukan N perkalian kompleks, dan dengan metode pembagian seperti ini, maka terdapat $log_2(N)$ langkah perkalian kompleks. Hal ini berarti jumlah perkalian kompleks berkurang dari N^2 (pada DFT) menjadi $N\ log_2(N)$.

Hasil dari proses FFT ini adalah simetris antara indeks 0- (N/2-1) dan (N/2)-(N-1). Oleh karena itu, umumnya hanya blok pertama saja yang akan digunakan dalam proses-proses selanjutnya.

2.4.6 *Mel Frequency Warping*

Mel Frequency Warping umunya dilakukan dengan menggunakan *Filterbank*. *Filterbank* adalah salah satu bentuk dari filter yang akan dilakukan dengan tujuan untuk mengetahui enegi dari *frequency band* tertentu dalam sinyal suara. *Filterbank* dapat diterapkan baik pada domain waktu maupun domain frekuensi, tetapi untuk keperluan MFCC, *filterbank* harus diterapkan dalam domain frekuensi. Gambar dibawah menunjukan dua jenis *filterbank magnitude*. Dalam kasus dibawah, filter yang dilakukan adalah secara linear terhadap frekuensi 0-4 kHz.



Gambar 2.16 Magnitude Dari Rectangular Dan Triangular *Filterbank*

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 digunaka dalam perhitungan *filterbank*s.

$$Y[i] = \sum_{j=1}^{N} S[j]H_i[j] \quad (7)$$

N = jumlah magnitude spectrum

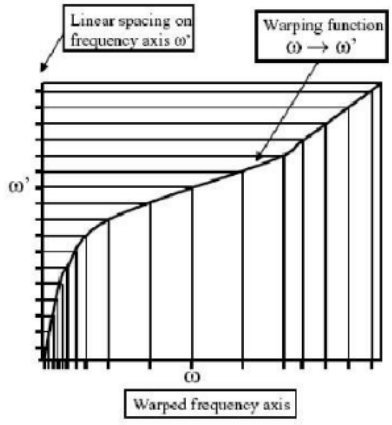
S[j] = magnitude spectrum pada frekuensi j

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

M= jumlah Channel dalam *filterbank*

Dimana $H_i = \frac{mel\ f}{X_i/_2}$ (8)

Persepsi manusia terhadap frekuensi dari sinyal suara tidak mengikuti linear scale. Frekuensi yang sebenarnya (dalam Hz) dalam sebuah sinyal akan diukur manusia secara subyektif dengan menggunakan *mel scale*. *Mel Frequency scale* adalah linear frekuensi *scale* pada frekuensi dibawah 1000 Hz, dan merupakan logarithmic scale pada frekuensi diatas 1000 Hz.



Gambar 2.17 Prinsip Frekuensi Warping

Berikut ini adalah formula untuk menghitung mel scale :

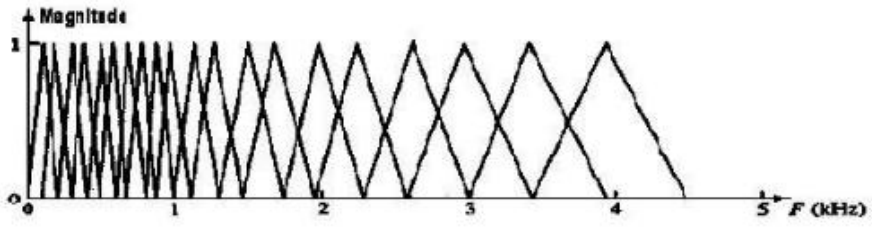
$$mel(f) = 2595\ X\ log_{10}(1 + \frac{f}{700}) \quad (9)$$

$Mel\ (f)$= Fungsi Mel Scale

f= frekuuensi

Gambar 2.18 dibawah menunjukan triangular *filterbank* dengan menggunakan *mel scale*. Bila diperhatikan lebih jauh, tampak bahwa untuk

frekuensi 1 kHz kebawah pola filternya terdistribusikan secara linear, dan diatas 1 kHz akan terdistribusi secara logarithmic.



Gambar 2.18 Triangular *Filterbank* Dengan Mel Scale

Dalam aplikasi *speaker recognation* dan *speech recognation*, jumlah *channel filterbank* yang digunakan akan sangat berpengaruh terhadap keberhasilan aplikasi tersebut. Semakin besar jumlah *channel* yang digunakan akan semakin meningkatkan keberhasilan aplikasi, tetapi akan membutuhkan waktu lebih besar untuk melakukan proses tersebut, begitu pula sebaliknya.

2.4.7 Discrete Cosine Transform (DCT)

DCT merupakan langkah terakhir dari proses utama MFCC *feature extraction*. Konsep dasar dari DCT akan mendekolerasikan mel Spektrum sehingga menghasilkan representasi yang baik dari properti spektral lokal. Pada dasarnya konsep dari DCT sama dengan *inverse fourier transform*. Namun hasil dari DCT mendekati PCA (*Principle Component Analisys*) . PCA adalah metode statistik klasik yang digunakan secara luas dalam analisa data dan kompresi. Hal inilah yang menyebabkan seringkali DCT menggantikan *inverse fourier trransform* dalam proses MFCC *feature extraction.*

Berikut ini adalah formula yang digunakan untuk menghitung DCT.

$$\tau_n = \sum_{k=1}^{K} (\log S_k) \cos[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}] \quad (10)$$

S_k = keluaran dari proses *filterbank* pada indeks k

K = jumlah koefisien yang diharapkan

Koefisien ke-nol dari DCT pada umumnya akan dihilangkan, walaupun sebenarnya mengindikasikan energi dari *frame* sinyal tersebut. Hal ini dilakukan karena, berdasarkan penelitian-penelitian yangn pernah dilakukan, koefisien ke nol ini tidak *reliable* terhadap *speaker recognation*. Tetapi koefisien ke nol dari DCT 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*.

2.4.8 Cepstral Liftering

Cepstral, hasil dari proses utama MFCC *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.

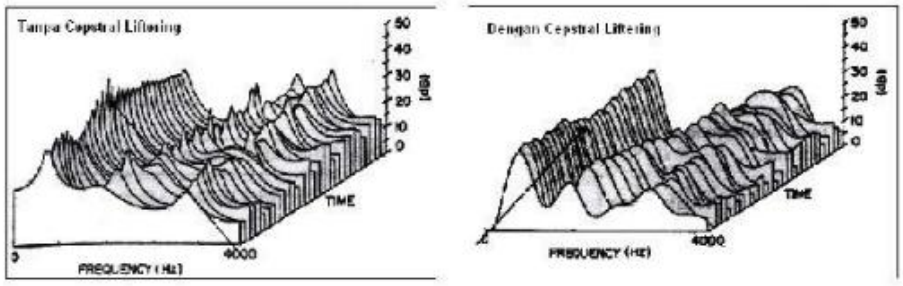
Cepstral liftering dapat dilakukan dengan mengimplementasikan fungsi *window* terhadap *cepstral features*.

$$w[n] = \left\{1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right)\right\} \quad (11)$$

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 2.20 berikut ini menunjukan perbandingan spektrum dengan dan tanpa *cepstral liftering*.



Gambar 2.19 Perbandingan Spectrum dengan dan tanpa Cepstral Liftering

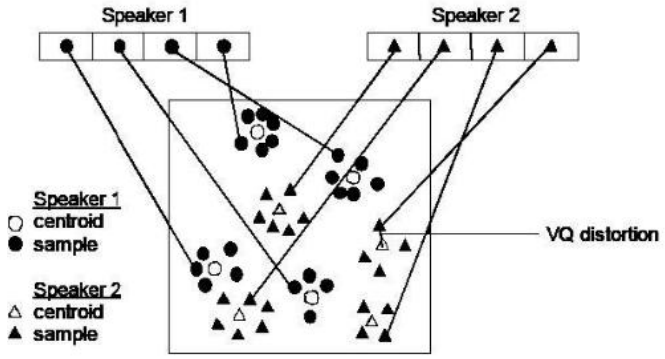
Berdasarkan gambar 2.20 diatas dapat dilihat bahwa pola spektrum setelah dilakukannya *cepstral liftering* lebih halus daripada spektrum yang tidak melalui tahap *cepstral liftering*. Dalam beberapa jurnal dikatakan bahwa *cepstral liftering* dapat meningkatkan akurasi dari aplikasi pengenalan suara, baik *speaker recognation*, maupun *speech recognation.*

2.5 K-Means Clustering

Clustering merupakan faktor yanng paling fundamental dalam *pattern recognation*. Masalah utama dari *clustering* adalah mendapatkan beberapa nilai vektor pusat yang dapat mewakili keseluruhan vektor dari hasil *feature extraction*. K-*means Clustering* adalah salah satu metode yang digunakan untuk mempartisi vektor hasil *feature extraction* ke dalam k vektor pusat[3].

K-Means Clustering adalah prroses memetakan vektor-vektor yang berada pada lingkup wilayah yang luas besar menjadi sejumlah tertentu (k) vektor. Wilayah yang terwakili oleh vektor pusat hasil dari proses kuantisasi disebut

sebagai *cluster*. Sebuah vektor pusat hasil dari proses kuantisasi dikenal sebagai *codewords*. Sedangkan kumpulan dari vektor pusat dikenal sebagai *codebooks*. Gambar berikut menunjukan hasil ilustrasi *K-Means Clustering*.



Gambar 2.20 K-Means Clustering

merepresentasikan *speech spectral vectors* adalah :

1. Mengurangi *storage memory* yang digunakan untuk analisis informasi spektral.
2. Mengurangi perhitungan yang digunakan untuk menentukan kemiripan dari vektor spektral.

Kelemahan dari penggunaan *K-Means Clustering codebooks* dalam merepresentasikan *speech spectral vectors* adalah :

1. Timbulnya spektral distorsi. Hal ini terjadi karena vektor yang dianalisa bukanlah vektor asli, tetapi sudah mengalami proses kuantisasi
2. *Storage* yang digunakan untuk menyimpan *codebooks* vektor sering kali menjadi masalah. Untuk jumlah *codebooks* yang besar, membutuhkan storage yang cukup besar juga.

2.6 Euclidean Distance

*Euclidean dista*nce adalah sebuah metode yang digunakan untuk mengukur jarak (*distance*). *Euclidean distance* sebenarnya merupakan generalisasi dari *teorema phytagoras*.

Berikut ini adalah contoh perhitungan dengan menggunakan *Euclidean Di*stance. Jika terdapat dua buah titik pada sebuah bidang dua dimensi (R^2), $u = (x_{1,}y_{1)}$ dan $v = (x_2, y_2)$ maka untuk mengukur jarak dari kedua buah titik tersebut dapat digunakan persamaan *phytagoras*

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (12)$$

x_1, x_2 = koordinat sumbu x dari sebuah titik

y_1, y_2 = koordinat sumbu y dari sebuah titik

Jarak tersebut menyebabkan sebuah metric pada R^2, yang disebut sebagai *Euclidean metric* pada R^2.

Bila terdapat dua buah vektor dengan n dimensi , $a = (a_1, a_{2\,,\ldots\ldots,}\, a_n)$ dan $b = (b_1, b_2, \ldots., b_n)$ maka formula *phytagoras* dapat digeneralisasikan ke dalam n dimensi (R^2) menjadi :

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}$$

Perhatikan kemiripan dari dua buah formula diatas. Formula *euclidean distamce* juga merupakan metric pada R^n, dikenal sebagai *Euclidean Space.* Berikut ini adalah bentuk umum dari *euclidean distance*

$$d = (\sum_{n=1}^{N}(x_n - y_n)^p)^{1/p} \quad (13)$$

N= jumlah dimensi vektor

X,y = vektor

$$P = \text{norm} (p \epsilon R)$$

Nilai p yang paling sering digunakan adalah p=1 dan p=2, atau yang sering disebut sebagai 1-norm dan 2-norm *distance* kecepatan yang berbeda. Terdapat dua istilah yang digunakan :

1. *features*

   informasi di dalam suatu sinyal *speech* harus diepresentasikan dengan suatu cara.

2. *Distance*

   Suatu ukuran yang dipakai untuk menentukan kemiripan. Ada dua macam distance :

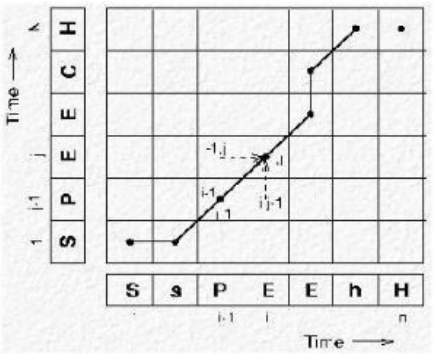
   a. Local :

      Perhitungan perbedaan antara suatu fitur dari suatu sinyal dengan fitur dari sinyal lain.

   b. Global :

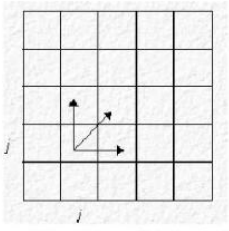
      Perhitungan secara keseluruhan perbedaan antara suatu sinyal secara keseluruhan dengan sinyal lain yang mungkin berbeda panjangnya.

Untuk menghitung *global distance* dari dua buah *speech pattern* maka dapat kita gambarkan dalam grafik waktu sebagai berikut :

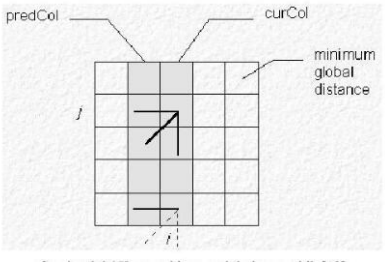


Gambar 2.21 Ilustrasi Pencocokan Input Dengan Template

Ilustarasi diatas adalah pencocokan antara input "Speech" dalam kondisi yang ramai dengan template.



Gambar 2.22 Tiga kemungkinan arah untuk titik selanjutnya



Gambar 2.23 Kemungkinan Asal dari Suatu Titik

Perangkat lunak ini menggunakan *Symetric Dynamic Time Warping*. Pada metode ini, suatu titik diperkirakan dapat berasal dari tiga kemungkinan arah oleh karena itu global *distance* pada suatu titik diperoleh dengan menjumlahkan *local distance* pada tititk itu dengan *global distance* paling rendah dari kemungkinan

tiga titik yang dapat menjadi titik sebelum titik tersebut. Untuk menghitung *Global distance* dari suatu titik dapat dicari dengan rumus berikut :

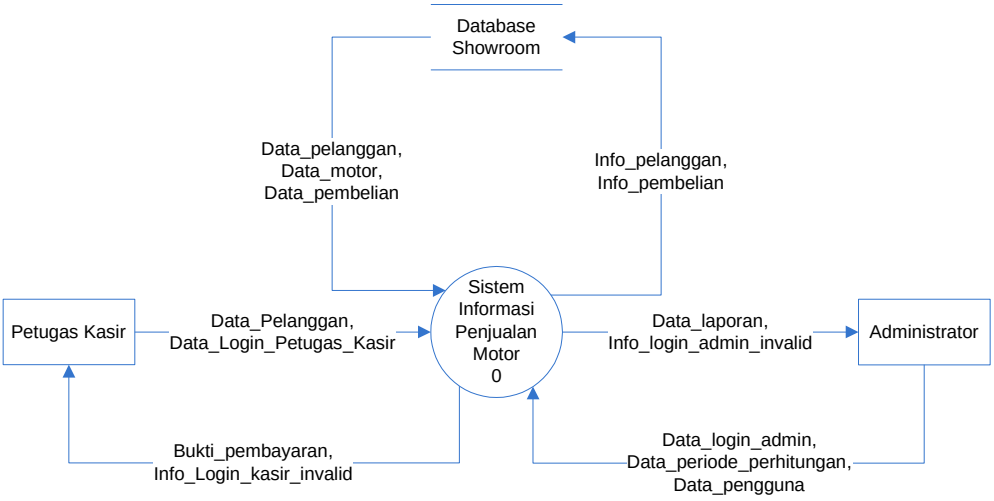
$$D(i,j) = \min[\, D(i-1,j-1), D(i-1) \qquad (14)$$

2.7 Pemodelan Sistem

Dalam suatu proses pengembangan *software*, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegosiasikan, dapat dikatakan bahwa kita berada pada tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu *tool* atau model untuk merancang pengembangan *software* yang berbasis terstruktur adalah DFD dan Diagram Konteks.

2.7.1 Diagram Konteks

Yaitu diagram tingkat atas, merupakan diagram dari sebuah sistem yang menggambarkan aliran-aliran data yang masuk dan keluar dari sistem dan dari entitas luar.



Gambar 2.24 Diagram Konteks

2.7.2 Data Flow Diagram (DFD)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang akan dikenakan pada data tersebut.

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan sacara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telepon, surat dan sebagainya). Atau lingkungan fisik dimana data tersebut akan disimpan (misalnya file kartu, *hard disk*, *tape*, disket dan sebagainya). DFD merupakan alat yang cukup populer saat ini, karena dapat menggambarkan arus data didalam sistem dengan terstruktur dan jelas. Lebih lanjut DFD merupakan dokumentasi dari sistem yang baik. Beberapa simbol yang akan digunakan di dalam DFD antara lain adalah sebagai berikut :

1. Kesatuan luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan dilingkungan luar sistem dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem. Kesatuan luar ini kebanyakan adalah salah satu dari berikut ini:

a. Suatu kantor, departemen atau divisi dalam perusahaan tetapi di luar sistem yang sedang dikembangkan.
b. Orang atau sekelompok orang di organisasi tetapi di luar sistem yang sedang dikembangkan.
c. Suatu organisasi atau orang di luar organisasi.
d. Sistem informasi yang lain di luar sistem yang sedang dikembangkan.
e. Sumber asli dari suatu transaksi.
f. Penerimaan akhir dari suatu laporan yang dihasilkan oleh sistem.

2. Aliran Data *(Data Flow)*

Aliran data di DFD diberi simbol suatu panah. Aliran data ini mengalir diantara proses (*process*), simpan data (*data store*) dan kesatuan luar (*external entity*). Aliran data ini menunjukkan aliran dari data yang dapat berupa masukkan untuk sistem atau hasil dari proses sistem.

3. Proses

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses yang digambarkan secara umum. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut – sudutnya tumpul.

4. Berkas atau Simpanan Data (*Data Store*)

Berkas atau simpanan data merupakan simpanan dari data yang dapat berupa :

a) Suatu file atau database di sistem komputer.

b) Suatu arsip atau catatan manual.

2.8 Metode Pengembangan Perangkat Lunak

Merupakan sebuah model *chaos* yang menggambarkan "perkembangan P/L sebagai sebuah kesatuan dari pemakai ke pengembang dan ke teknologi" disebut dengan "Prescriptive" karena Menentukan sekumpulan elemen proses (Aktivitas, Aksi, tugas, produk kerja, jaminan kualitas, dll untuk setiap proyek), setiap model proses juga menentukan alur kerjanya.

Pada saat kerja bergerak maju menuju sebuah sistem yang lengkap, keadaan yang digambarkan secara rekursif diaplikasikan kepada kebutuhan pemakai dan spesifikasi teknis P/L pengembang Saat ini, prescriptive memberikan jawaban secara definitive untuk masalah pengembangan P/L dalam setiap perubahan lingkungan komputasi

2.8.1 Model Skuensial Linear

Pada Tugas akhir ini menggunakan metode penggunaan perangkat lunak siklus kehidupan klasik atau model air terjun. Model ini mengusulkan sebuah pendekatan kepada perkembangan software yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode,

pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas–aktivitas sebagai berikut :

1. Rekayasa dan pemodelan sistem/informasi

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke software tersebut. Pandangan sistem ini penting ketika software harus berhubungan dengan elemen-elemen yang lain seperti software, manusia, dan database. Rekayasa dan anasisis system menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mancakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan Software

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khusunya pada software. Untuk memahami sifat program yang dibangun, analis harus memahami domain informasi, tingkah laku, unjuk kerja, dan interface yang diperlukan. Kebutuhan baik untuk sistem maupun software didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain software sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi software yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode.

Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi software.

4. Generasi Kode

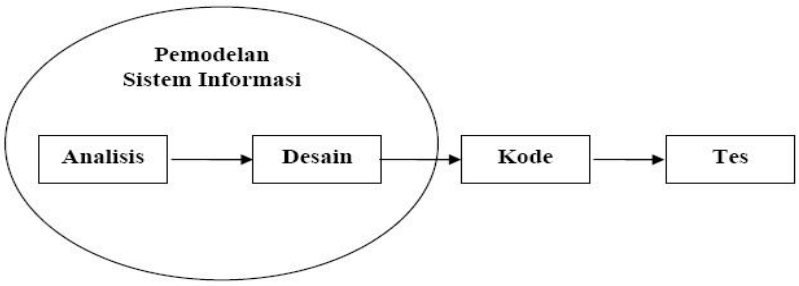
Desain harus diterjemahkan kedalam bentuk mesin yang bias dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal software, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan–kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

Software akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah software yang dilekatkan). Perubahan akan terjadi karena kesalahan–kesalahan ditentukan, karena software harus disesuaikan untuk mengakomodasi perubahan–perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan software mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.



Gambar 2.25 ***Sequential Linear Model***

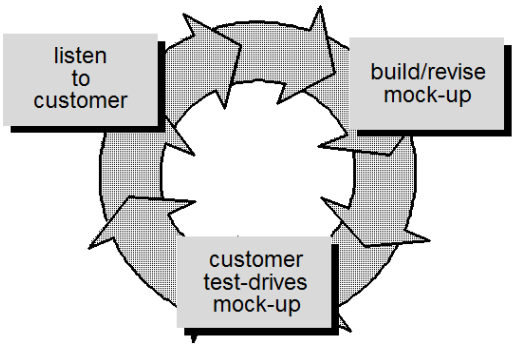
Masalah yang kadang terjadi ketika model sekuensial linier diaplikasikan adalah :

1. Jarang sekali proyek nyata mengikuti aliran sekuensial yang dianjurkan oleh model. Meskipun model linier bisa mengakomodasi iterasi, model ini melakukannya dengan cara tidak langsung. Sebagai hasilnya, perubahan–perubahan dapat menyebabkan keraguan pada saat tim proyek berjalan.
2. Kadang–kadang sulit bagi pelanggan untuk menyatakan semua kebutuhannya secara eksplisit. Model linier sekuensial memerlukan hal ini dan mengalami kesulitan untuk mengakomodasi ketidakpastian natural yang ada pada bagian awal beberapa proyek. Pemodelan Sistem Informasi
3. Pelanggan harus bersifat sabar. Sebuah versi kerja dari program – program kerja itu tidak akan diperoleh sampai akhir waktu proyek dilalui. Sebuah kesalahan besar, jika tidak terdeteksi sampai program yang bekerja tersebut dikaji ulang, bisa menjadi petaka.
4. Pengembang sering melakukan penundan yang tidak perlu. Sifat alami dari siklus kehidupan klasik membawa kepada *blocking state* di

mana banyak anggota tim proyek harus menunggu tim yang lain untuk melengkapi tugas yang saling memiliki ketergantungan. Blocking state cenderung menjadi lebih lazim pada awal dan akhir sebuah proses sekuensial linier.

2.8.2 Model Prototype

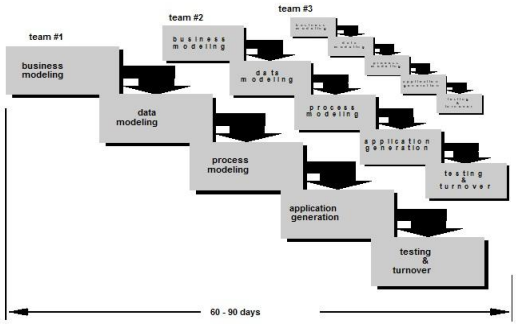
Model prototype dibangun dari mengumpulkan berbagai kebutuhan, kemudian tim pengembang akan nertemu dengan pelanggan untuk menentukan tujuan dari perangkat lunak, dan mengidentifikasi kebutuha-kebutuhan yang telah diketahui oleh pelanggan, dan batasan-batasan apa saja yang dapat dikategorikan sebagai tugas utama. Hasilnya akan dibangun rancangan sementara yang mewakili berbagai aspek dari perangkat lunak yang kelak akan digunakan oleh pelanggan/pengguna (seperti bentuk pendekatan input yang digunakan dan bentuk output). Idealnya model prototype melayani sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Dimana jika nantinya sebuah model prototype berhasil dibuat, seorang developer harus berusaha mendayagunakan tools yang ada (semisal, report generator, windows manager) dapat bekerja dengan baik (cepat).



Gambar 2.26 ***Model Prototype***

2.8.3 Model RAD

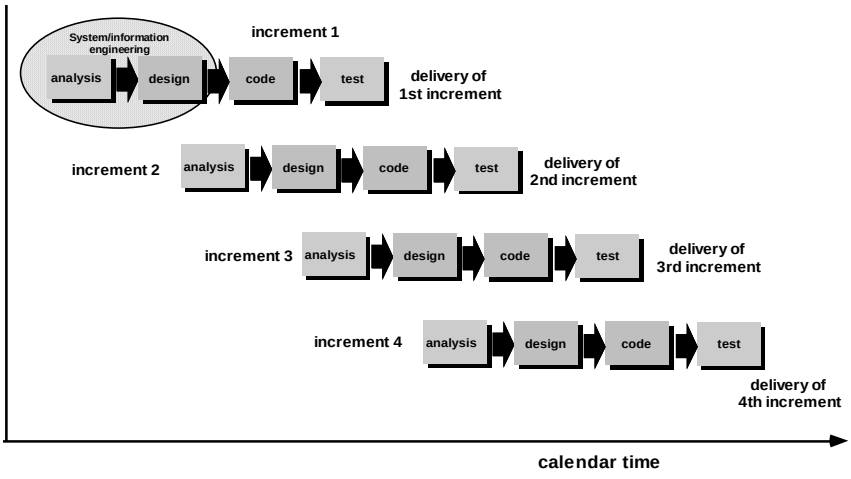
Rapid Aplication Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi "kecepatan tinggi" dari model sequensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.



Gambar 2.27 ***Model RAD***

2.8.4 Model Evolusioner

Model Revolusioner adalah model iterative. Model itu ditandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat yang lebih lengkap sedikit demi sedikit.



Gambar 2.28 ***Model Revolusioner***

2.9 Software Pendukung

2.9.1 Microsoft Visual Basic

Microsoft Visual Basic merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM).

Visual Basic merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.

Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic Program-program yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi luar tambahan.

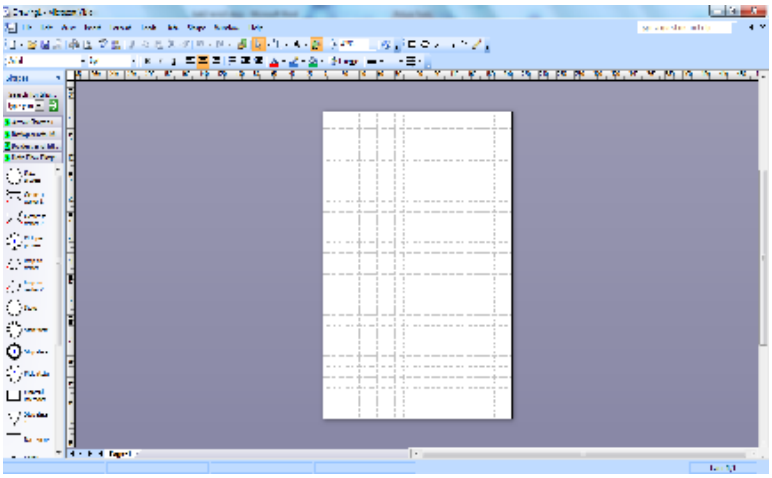
Dalam pemrograman untuk bisnis, Visual Basic memiliki pangsa pasar yang sangat luas. Sebuah survey yang dilakukan pada tahun 2005 menunjukkan bahwa 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk Visual Basic, yang diikuti oleh C++, JavaScript, C#, dan Java.

2.9.2 *Microsoft Office Visio*

Rational Rose adalah *software* yang memiliki perangkat-perangkat pemodelan secara visual untuk membangun suatu solusi dalam rekayasa *software*

dan pemodelan bisnis. Microsoft Visio adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *brainstorm*, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.

Visio aslinya bukanlah buatan Microsoft Corporation, melainkan buatan Visio Corporation, yang diakusisisi oleh Microsoft pada tahun 2000. Versi yang telah menggunakan nama Microsoft Visio adalah Visio 2002, Visio 2003, dan Visio 2007 yang merupakan versi terbaru. Visio 2007 Standard dan Professional menawarkan antarmuka pengguna yang sama, tapi seri Professional menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (layout). Selain itu, edisi Professional juga memudahkan pengguna untuk mengoneksikan diagram-diagram buatan mereka terhadap beberapa sumber data dan juga menampilkan informasi secara visual dengan menggunakan grafik.



Gambar 2.29 Tampilan Utama *Microsoft Office Visio 2007*

2.9.3 Mikrofon

Mikrofon (*michrophone*) adalah suatu jenis tranduser yang mengubah energi-energi akustik (gelombang suara) menjadi sinyal listrik. Mikrofon merupakan salah satu alat untuk membantu komunikasi manusia. Mikrofon dipakai pada banyak alat seperti telepon, alat perekam, alat bantu dengar, dan pengudaraan radio serta televisi.

Istilah *mikrofon* berasal dari bahasa Yunani *mikros* yang berarti kecil dan *fon* yang berarti suara atau bunyi. Istilah ini awalnya mengacu kepada alat bantu dengar untuk suara berintensitas rendah. Penemuan mikrofon sangat penting pada masa awal perkembangan telepon. Pada awal penemuannya, mikrofon digunakan pada telepon, kemudian seiring berkembangnya waktu, mikrofon digunakan dalam pemancar radio hingga ke berbagai penggunaan lainnya. Penemuan mikrofon praktis sangat penting pada masa awal perkembangan telepon. Beberapa penemu telah membuat mikrofon primitif sebelum Alexander Graham Bell.

Pada tahun 1827, Sir Charles Wheatstone telah mengembangkan mikrofon. Ia merupakan orang pertama yang membuat “mikrofon frase". Selanjutnya, pada tahun 1876, Emile Berliner menciptakan mikrofon pertama yang digunakan sebagai pemancar suara telepon. Mikrofon praktis komersial pertama adalah mikrofon karbon yang ditemukan pada bulan Oktober 1876 oleh Thomas Alfa Edison. Pada tahun 1878, David Edward Hughes juga mengambil andil dalam perkembangan mikrofon karbon. Mikrofon karbon tersebut mengalami perkembangan hingga tahun 1920-an.

James West and Gerhard Sessler juga memainkan peranan yang besar dalam perkembangan mikrofon. Mereka mempatenkan temuan mereka yaitu mikrofon elektrik pada tahun 1964. Pada waktu itu, mikrofon tersebut menawarkan sesuatu yang tidak dimiliki oleh mikrofon sebelumnya, yaitu harga rendah, sehingga dapat dijangkau oleh seluruh konsumen. Bagian lain dalam sejarah perkembangan mikrofon ialah revolusionalisasi mikrofon dalam industri dimana memungkinkan masyarakat umum untuk mendapatkannya. Hampir satu juta mikrofon diproduksi tiap tahunnya. Lalu pada tahun 1970-an, mikrofon dinamik dan mikrofon kondenser mulai dikembangkan.Mikrofon ini memiliki tingkat kesensitifan yang tinggi. Oleh karena itu, hingga saat ini mikrofon tersebut digunakan dalam dunia penyiaran.