BAB II

LANDASAN TEORI

2.1 Tangis Bayi

Menangis merupakan jalan utama bayi untuk berkomunikasi. Dalam beberapa hari pertama dari hidupnya, bayi yang baru lahir menangis dikarenakan reaksi dari kedua internal dan eksternal stimuli dengan tujuan untuk menguatkan fungsi jantung dan paru-paru (Brazelton, 1962). Setelah bayi dilahirkan, tangisannya akan menjadi respons terhadap kebutuhan atas perubahan temperatur, rasa lapar, dan rasa sakit atau tidak nyaman (Brazelton, 1962).

Bayi pada umur 0-3 bulan masih dalam masa periode transisi dimana bayi bergerak dari refleks-refleks untuk bertahan hidup ke cara yang lebih terorganisir untuk memproses informasi (Perry, 1997). Refleks dari seorang bayi dan gerak-gerik pertamanya diatur oleh batang otak, yang berlokasi di pangkal otak dimana saraf tulang belakang terhubung dengan otak (Perry, 1997). Batang otak dapat mengalami kerusakan selama periode manangis yang intens, dikarenakan batang otak membanjiri tubuh dengan adrenalin dan hormon stres lainnya pada frekuensi waktu yang tidak tepat (Perry, 1997). Perhatian ketika bayi menangis diperlukan untuk menjaga kesehatan bayi selama masa pertumbuhan.

Dunstan Baby Language (DBL) merupakan salah satu teknik klasifikasi tangis bayi yang menyatakan tentang refleks vokal yang berhubungan dengan bayi sebagai sinyal suara (Dunstan, 2006b). DBL menyatakan bahwa ada lima klasifikasi tangis bayi di seluruh budaya dan kelompok linguistik, yang digunakan oleh bayi sebelum akuisisi bahasa (Dunstan, 2006b).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

2.1.1 Dunstan Baby Language (DBL)

DBL merupakan klasifikasi tangis bayi berdasarkan rasa ketidaknyamanan yang dirasakan pada bayi yang ditemukan oleh Priscilla Dunstan (Dunstan, 2012). Setiap kata yang diidentifikasi dalam DBL berdasarkan refleks dari aksi natural dikarenakan kebutuhan fisik bayi (Dunstan, 2012). Ketika bayi merasakan ada ketidaknyamanan dengan fisiknya, tubuh bayi akan menghasilkan refleks dan akan menambahkan suara dalam bentuk tangisan (Dunstan, 2012). Kombinasi dari refleks dan suara membuat sebuah *signature* suara yang spesifik sesuai dengan kebutuhan bayi, dan *signature* suara ini dapat diartikan sebagai sebuah kata (Dunstan, 2012). Ketika sebuah kata tersebut dapat dimengerti, maka penanganan yang berasosiasi dengan refleks dapat ditentukan, seperti ketika bayi lapar akan mengisap tangannya dan kemudian dia akan mulai menangis, maka sebuah kata lapar dapat diartikan dari tangisan bayi tersebut (Dunstan, 2012).

Priscilla Dunstan memulai penelitian ini dengan mencatat pola tangis bayi yang berbeda pada waktu-waktu tertentu dan menamakan tiap suara berdasarkan pola dan fonetik suara (Dunstan, 2012). Dari tiap nama fonetik suara yang muncul ketika bayi menangis, ditemukan solusi yang berbeda-beda untuk menenangkan bayi yang sedang menangis (Dunstan, 2012). Setelah berhasil melakukan percobaan pada satu bayi, Priscilla Dunstan menerapkan hasil penelitiannya kepada bayi yang lain, dan hasilnya bayi tersebut mengeluarkan pola suara tangisan yang sama dengan bayi sebelumnya (Dunstan, 2012). Pada setiap tangis bayi tersebut, dideteksi fonetik *signature* suara yang akhirnya menjadi DBL. Ketika setiap pola tangisan ditanggapi dengan solusi yang telah diterapkan pada bayi sebelumnya, tiap bayi akan berhenti

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

menangis (Dunstan, 2012). Klasifikasi suara tangis bayi DBL terbagi ke dalam lima bagian sebagai berikut (Dunstan, 2006b).

1. “Neh”

   Suara “Neh” adalah respon dari refleks mengisap, yang menyatakan bayi sedang kelaparan. Memberikan Air Susu Ibu (ASI) kepada bayi ketika mendengar suara “Neh” dari bayi sangatlah penting, meskipun bayi telah diberikan ASI secara teratur setiap tiga sampai empat jam. Ketika menyusui pada masa ini, bayi akan menyapih lebih baik dan menyusu lebih lama. Pada hari yang bersuhu panas, bayi lebih sering mengeluarkan kata “Neh” karena bayi lebih haus dari biasanya. Dengan memberikan asupan ASI ketika bayi mengeluarkan suara “Neh”, bayi akan mendapatkan asupan makanan yang tepat.

2. “Owh”

   Suara “Owh” didasarkan pada efek menguap, yang mengindikasikan bayi siap untuk tidur. Sama seperti orang dewasa ketika menguap, mulut bayi akan berbentuk oval ketika mengeluarkan suara “Owh”. Jika meletakkan bayi pada posisi tidur di ranjang setelah mendengar suara “Owh”, maka bayi akan lebih mudah dan cepat tertidur. Suara “Owh” biasanya keluar sebelum bayi melakukan gerakan tersentak-sentak dan mengusap-ngusap matanya, yang menandakan kalau bayi sedang kelelahan. Dengan menidurkan bayi pada waktu yang tepat, bayi tidak akan merasa lelah dan frustasi.

3. “Eh”

   Tangisan “Eh” dihasilkan dari dada yang berkontriksi dalam upaya untuk mengeluarkan angin. Hal ini biasa didengar sebagai rangkaian singkat dari tangisan “Eh, Eh, Eh” sebagaimana bayi mencoba untuk bersendawa. Ketika

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

bayi terlalu lama dalam posisi berbaring, maka bayi perlu bersendawa sewaktu siang dan malam, tidak hanya pada waktu sehabis makan saja. Tangisan "Eh" merupakan yang paling penting untuk menghindari tangis bayi yang berkepanjangan. Membantu bayi bersendawa ketika bayi mengeluarkan suara "Eh" akan membantu untuk mencegah udara tertekan pada bagian perut bawah yang menyebabkan rasa sakit pada bayi.

4. "Eairh"

   Ketika bayi menangis terus menerus, ada kemungkinan besar suara "Eairh" dapat terdengar dari tangisan bayi. Suara "Eairh" dihasilkan dari perut bagian bawah dan menandakan bahwa bayi mengalami kembung yang menyebabkan rasa sakit pada bayi. Ketika bayi menangis dan mengeluarkan suara "Eairh", posisikan bayi tengkurap dan usap bagian belakang bayi, atau dengan lemah lembut pijat bagian perut bayi untuk mengeluarkan udara untuk mengurangi rasa sakit pada bayi. Jika penanganan ketika suara tangis "Eh" pada bayi telah dilakukan, kemungkinan suara tangis "Eairh" untuk muncul akan kecil, karena udara yang akan menyebabkan rasa kembung pada bayi sudah dikeluarkan terlebih dahulu.

5. "Heh"

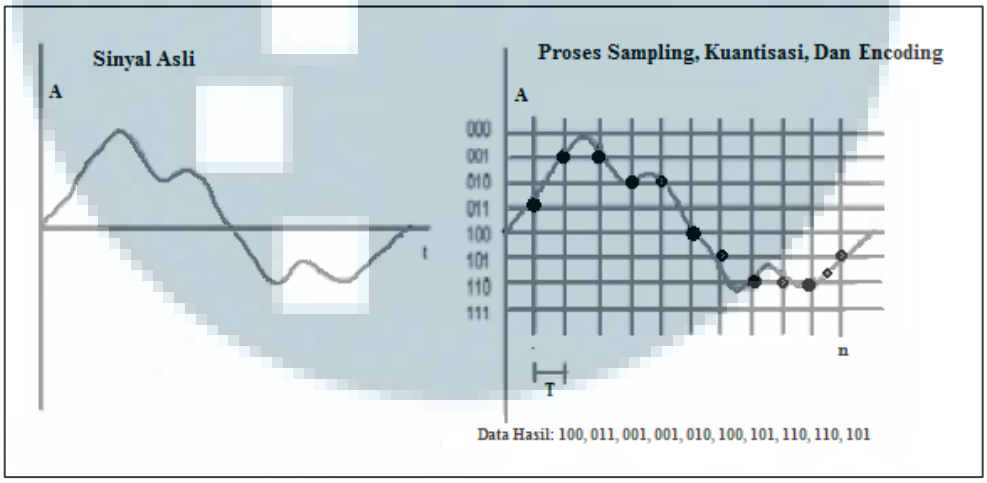
   Bayi akan sering rewel atau cerewet ketika dia merasa tidak nyaman karena popok yang dipakainya basah dan kotor, atau ketika bayi merasa kepanasan atau kedinginan. Suara "Heh" berdasarkan refleks yang disebabkan oleh ketidaknyamanan pada kulit bayi. Bayi yang baru lahir tidak dapat mengatur suhu tubuhnya secara efektif, sehingga bayi akan cepat kepanasan atau kedinginan, tetapi dengan mengenali tangisan "Heh" akan mudah merasakan

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

rasa tidak nyaman yang dirasakan bayi. Ada beberapa alasan lain bayi mengeluarkan suara "Heh", seperti tali pengaman di kursi mobilnya terlalu ketat, atau pakaian bayi membuat kulitnya mengalami iritasi.

2.2 Audio Digital

Audio digital merupakan sebuah versi digital dari suara analog (Binanto, 2010). Pengubahan suara analog menjadi suara digital membutuhkan suatu alat yang dinamakan *Analog to Digital Converter* (ADC). ADC mengubah amplitudo sebuah gelombang analog ke dalam waktu interval (*sampling*), sehingga menghasilkan representasi digital dari suara (Binanto, 2010).



Gambar 2.1 Proses Pembentukan Sinyal Suara Digital
(Sumber: Putra dan Resmawan, 2011)

Sampling adalah proses untuk mengambil data dari sinyal kontinu untuk setiap periode tertentu. Ketika melakukan *sampling* data, berlaku aturan *Nyquist*, yaitu bahwa frekuensi *sampling* (*sampling rate*) minimal harus dua kali lebih tinggi dari frekuensi maksimum yang akan di-*sampling* (Putra dan Resmawan, 2011). Jika sinyal *sampling* kurang dari dua kali frekuensi maksimum sinyal, maka akan menyebabkan efek *aliasing*. *Aliasing* adalah suatu efek dimana sinyal yang

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

dihasilkan memiliki frekuensi yang berbeda dengan sinyal aslinya (Putra dan Resmawan, 2011).

Kuantisasi adalah proses untuk membulatkan nilai ke dalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu (Putra dan Resmawan, 2011). Semakin tinggi tingkat kuantisasi yang dipakai maka semakin akurat data sinyal yang disimpan, tetapi akan menghasilkan ukuran yang lebih besar dan proses yang lebih lama (Putra dan Resmawan, 2011). *Encoding* adalah proses pemberian kode untuk tiap-tiap data sinyal yang telah terkuantisasi berdasarkan level yang ditempati (Putra dan Resmawan, 2011).

Digital to Analog Converter (DAC) merupakan kebalikan dari ADC, yaitu mengubah suara digital ke alat suara analog (*speaker*). Audio digital merupakan representasi dari suara asli (*original sound*) (Binanto, 2010). Kualitas perekaman digital tergantung pada berapa sering sampel diambil (angka *sampling* atau frekuensi dalam kiloHertz (kHz) atau seribu sampel per detik) (Binanto, 2010). Empat frekuensi *sampling* yang paling sering digunakan dalam multimedia adalah kualitas *Compact Disk* (CD) 44.1 kHz, 22.05kHz, 11.25 kHz, dan 8kHz dengan ukuran sampel 8-bit dan 16-bit. Ukuran sampel 8-bit menyediakan 256 unit deskripsi jarak dinamis atau amplitudo, sedangkan 16-bit menyediakan 65.546 unit (Binanto, 2010). Pengambilan *sampling* gelombang dengan ADC mempunyai kendali terhadap dua variabel berikut (Binanto, 2010).

1. *Sampling Rate*

   Sampling rate adalah mengendalikan berapa banyak sampel yang akan diambil per-detiknya.

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

2. *Sampling Precision*

   Sampling precision adalah mengendalikan berapa banyak tingkat kuantisasi yang dimungkinkan ketika mengambil sampel.

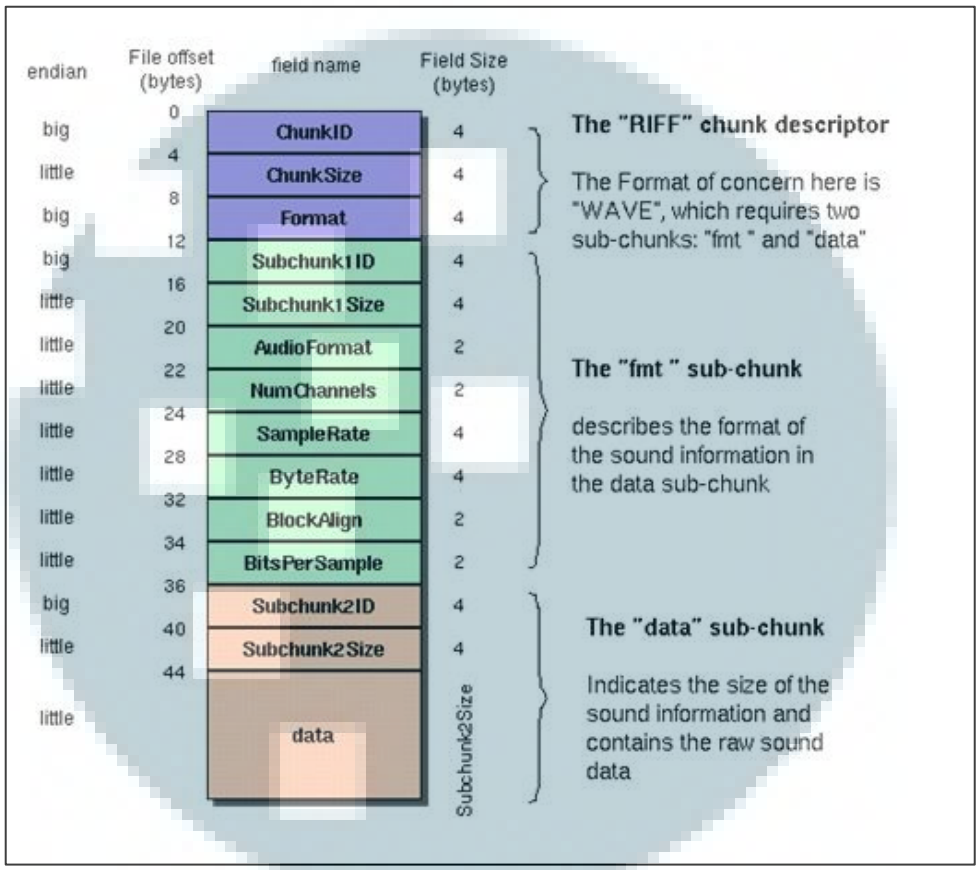
2.2.1 File Waveform Audio File Format (WAV)

File WAV adalah format *file* audio digital yang diciptakan oleh Microsoft untuk *Personal Computer* (PC) (Gunawan dan Gunadi, 2005). Suara yang disimpan dalam *file* WAV berupa audio digital dalam bentuk gelombang sehingga *file* ini memiliki ekstensi .wav (Wave) (Gunawan dan Gunadi, 2005). *File* WAV dibuat dengan menggunakan berbagai program *wave editor* maupun *wave recorder* (Gunawan dan Gunadi, 2005). Secara umum kualitas suatu *file* WAV dapat diukur dengan karakteristik yang dinyatakan dengan parameter-parameter berikut (Gunawan dan Gunadi, 2005).

1. *Sample Rate*, menyatakan banyaknya jumlah sampel yang dimainkan setiap detiknya. *Sample rate* yang umum digunakan adalah 8000 Hz, 11025 Hz, 22050 Hz, dan 44100 Hz.
2. Bit *Rate*, merupakan ukuran bit tiap sampelnya, yaitu 8-bit, 16-bit, atau 32-bit. Pada 8-bit semua sampel hanya mengonsumsi satu *byte*, dan memiliki nilai -128 sampai 127. Sedangkan untuk 16-bit mengonsumsi dua *byte*, dan memiliki nilai antara -32768 sampai 32767, sehingga menghasilkan suara yang lebih baik karena datanya lebih akurat.
3. Jumlah *channel*, merupakan banyaknya *channel* yang dipakai, yang menentukan suara yang dihasilkan apakah mono atau stereo. Mono hanya memiliki satu *channel*, sedangkan stereo hanya memiliki dua *channel* dan memakan tempat

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

dua kali lebih banyak dari pada mono. Untuk merekam suara manusia, mono sudah cukup memberikan kualitas yang baik, sedangkan untuk kualitas CD menggunakan stereo.



Gambar 2.2 Struktur *File* WAV
(Sumber: Gandhi dan Garg, 2015)

Struktur *file* WAV terbagi ke dalam 3 bagian utama sebagai berikut (Gunawan dan Gunadi, 2005).

1. *Resource Interchange File Format* (RIFF) *descriptor*

   File WAV menggunakan struktur standar dari *Resource Interchange File Format* (RIFF). RIFF merupakan struktur yang biasa digunakan untuk data multimedia dalam Windows. Struktur RIFF mengelompokkan data ke dalam bagian-bagian yang masing-masing memiliki *header* dan memiliki ukuran sendiri, yang disebut sebagai *chunk.* Sesuai dengan struktur RIFF, *file* WAV di

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

awali dengan empat *byte* yang berisi RIFF kemudian diikuti dengan empat *byte* yang merupakan ukuran *file* tersebut, dan empat *byte* lagi yang berisi 'WAVE', yang merupakan format *file* tersebut (.wav).

2. Format audio

   Pada bagian format audio terdapat informasi-informasi mengenai bagaimana memainkan data dan menyimpannya. Bagian ini dimulai dengan 'Subchunk ID', lalu diikuti dengan empat *byte* yang merupakan panjang dari informasi dan bernilai 16 untuk *Pulse-Code Manipulation* (PCM). Format audio menempati dua *byte* berikutnya dengan nilai satu untuk PCM. Jumlah *channel* yang digunakan pada *file* WAV menggunakan dua *byte*, lalu empat *byte* menyatakan *sample rate* dan empat *byte* lagi menyatakan rata-rata *byte* tiap detiknya. *Block Align* menyatakan ukuran data untuk satu sampel yang mewakili nilai dari sampel pada semua *channel* pada satu waktu. Dua *byte* terakhir dari bagian format audio ini menyatakan bit *rate* dari data yang disimpan, bernilai 8, 16, 24, atau 32.

3. Data audio

   Bagian berikutnya adalah bagian data audio. Pada bagian ini sampel digital audio disimpan. Bagian ini dimulai dengan ID dari data (Subchunk2ID) dan diikuti dengan empat *byte* yang menyatakan besarnya data dalam *byte* (Subchunk2Size), lalu selebihnya adalah data dari audio digital.

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

2.3 Pengenalan Suara

Pengenalan suara atau istilahnya dalam Bahasa Inggris adalah *Automatic Speech Recognition* (ASR) merupakan suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan kata yang diucapkan (Lestary, 2009). Teknologi pengenalan suara memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital dengan suatu pola tertentu yang disimpan dalam suatu perangkat (Lestary, 2009).

Gambar 2.3 Tahapan dalam Pengenalan Suara
(Sumber: Rachman, 2006)

Secara umum, proses pengenalan suara dimulai dengan memasukkan sinyal suara ke dalam sistem (Rachman, 2006). Sinyal suara yang menjadi *input* bersifat kontinu, untuk itu diperlukan pemrosesan awal (*pre-processing*) untuk mengubah sinyal tersebut menjadi *discrete* agar dapat diproses oleh komputer (Rachman, 2006). Setelah itu sinyal tersebut akan melalui proses ekstraksi ciri untuk mendapatkan parameter khusus yang menjadi bahan pembanding dalam proses pencocokan pola. Pada proses pencocokan pola, sistem akan membandingkan sinyal suara masukan dengan sinyal pembanding lalu sistem akan menentukan keputusan (Rachman, 2006).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

2.4 Mel-Frequency Cepstral Coefficients (MFCC)

Ekstraksi ciri merupakan proses dimana setiap sampel sinyal akan diubah menjadi vektor-vektor data. Terdapat dua buah proses ekstraksi ciri yang biasa digunakan, antara lain *Linear Prediction Coding* (LPC) dan *Mel-Frequency Cepstral Coefficients* (MFCC) (Putri, 2012). Metode ekstraksi ciri yang dipakai dalam penelitian ini adalah MFCC.

Metode ekstraksi ciri MFCC merupakan metode ekstraksi ciri yang lebih baik dibandingkan dengan metode ekstraksi ciri lainnya (Manunggal, 2005). Berikut keunggulan yang dimiliki metode ekstraksi ciri MFCC dibandingkan dengan metode ekstraksi ciri lainnya (Manunggal, 2005).

1. 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.

Perhitungan yang dilakukan dalam MFCC menggunakan dasar-dasar perhitungan *short-term analysis*, karena sinyal suara bersifat *quasistationary* (Manunggal, 2005). Pengujian yang dilakukan untuk periode waktu yang cukup pendek (sekitar 10 sampai 30 milidetik) akan menunjukkan karakteristik sinyal suara yang *stationary*, tetapi bila dilakukan dalam periode waktu yang lebih panjang, karakteristik sinyal suara akan berubah sesuai dengan kata yang diucapkan (Manunggal, 2005).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

Penelitian yang dilakukan oleh Patel dan Prasad (2013) menunjukkan bahwa persepsi manusia terhadap konten dari frekuensi suara untuk sinyal suara tidak mengikuti skala linear. Demikian pula untuk tiap *pitch* dengan frekuensi aktual, *f*, diukur dalam satuan Hz dan suatu *pitch* subjektif diukur dalam skala yang dinamakan skala *mel*.

2.4.1 End Point Detection (EPD)

Tahap pemrosesan awal dalam pengenalan suara sangatlah penting dalam pengaplikasiannya dimana *silent* atau *background noise* tidaklah diinginkan (Saha dkk., 2005). Sinyal suara dapat mengandung *silent* pada posisi yang berbeda-beda seperti di awal atau di akhir sinyal. Bagian *silent* dapat mempengaruhi proses identifikasi suara yang dimana terdapat bagian yang seharusnya tidak berkontribusi dalam proses identifikasi suara (Saha dkk., 2005). Proses penghilangan *silent* dapat dilakukan dengan mendeteksi bagian *voiced* atau *unvoiced* dari sebuah sinyal suara dengan menggunakan fungsi *one-dimensional Mahalanobis distance,* yang merupakan *linear pattern classifier* (Saha dkk., 2005). *Mahalanobis distance* adalah pengukuran antara dua titik data dengan fitur yang relevan dalam suatu ruang untuk menyesuaikan distribusi geometris dari data sehingga jarak antara data yang mirip menjadi kecil (Xiang dkk., 2008).

Algoritma untuk *End Point Detection* (EPD) terbagi menjadi dua bagian. Bagian pertama memberikan label untuk sampel menggunakan sifat statistik dari *background noise* sedangkan bagian kedua menyederhanakan label dengan menggunakan aspek psikologikal dari proses penghasilan suara (Saha dkk., 2005). Berikut algoritma dari EPD (Saha dkk., 2005).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

1. Hitung nilai rata-rata (μ) dan standar deviasi (σ) dari 200 milidetik pertama data sampel.
2. Dari sampel pertama sampai sampel terakhir, cek apakah fungsi *one-dimensional Mahalanobis distance* lebih besar dari 3 atau tidak. Jika lebih besar dari 3 merupakan *voiced* sampel, jika tidak merupakan *unvoiced/silent* sampel.

$$\frac{|x-\mu|}{\sigma} > 3 \qquad ...(2.1)$$

   Keterangan:

   x = Data sampel pertama.

   μ = Nilai rata-rata.

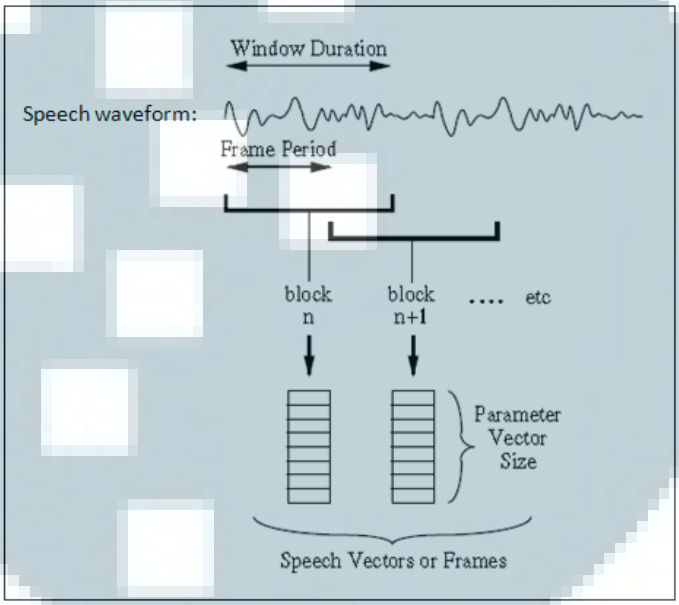
   σ = Standar deviasi.
3. Tandai sampel *voiced* sebagai '1' dan *unvoiced* sebagai '0'. Bagi seluruh sinyal suara ke dalam 10 milidetik *non-overlapping windows*.
4. Jika jumlah sampel yang ditandai '0' ≥ jumlah sampel yang ditandai '1', ubah setiap sampel '1' menjadi '0', dan begitu pula sebaliknya. Metode ini digunakan karena sistem penghasil sinyal suara tidak dapat berubah secara tiba-tiba dalam durasi *window* 10 milidetik.
5. Kumpulkan hanya bagian *voiced* sesuai dengan sampel yang telah ditandai '1' dari *array* yang telah di-*window* dan buang ke dalam *array* yang baru. Ambil bagian *voiced* dari sinyal asli berdasarkan data sampel yang telah ditandai '1'.

2.4.2 Frame Blocking

Karena sinyal suara terus mengalami perubahan akibat adanya pergeseran artikulasi dari organ reproduksi vokal, sinyal suara harus diproses secara *short segments* (*short frame*) (Putra dan Resmawan, 2011). Proses *frame blocking*

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

digunakan untuk membagi sampel sinyal suara yang didapatkan dari hasil konversi sinyal suara analog menjadi digital ke dalam *frame-frame* dengan rentang waktu 20-40 milidetik, disebut juga sebagai *frame blocking* (Razak dkk., 2008). Ukuran dari suatu *frame* harus sepanjang mungkin untuk dapat menunjukkan resolusi frekuensi yang baik, tetapi ukuran *frame* juga harus cukup pendek untuk mendapatkan resolusi waktu yang baik (Putra dan Resmawan, 2011).



Gambar 2.4 *Frame Blocking*
(Sumber: Wahib, 2015)

Proses *frame blocking* dilakukan secara terus-menerus sampai semua sinyal dapat diproses. Selain itu, proses *frame blocking* ini umumnya dilakukan secara *overlapping* untuk setiap *frame* (Putra dan Resmawan, 2011). Panjang daerah *overlap* yang umum digunakan adalah kurang lebih 30% sampai 50% dari panjang *frame*. *Overlapping* dilakukan untuk menghindari hilangnya ciri atau karakteristik suara pada perbatasan perpotongan setiap *frame* (Putra dan Resmawan, 2011).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

2.4.3 Windowing

Aliasing atau kebocoran spektral dapat terjadi setelah melakukan proses *frame blocking*. *Aliasing* merupakan sinyal baru yang memiliki frekuensi yang berbeda dengan sinyal aslinya (Putra dan Resmawan, 2011). *Sampling rate* yang rendah dapat menyebabkan terjadinya efek ini, ataupun karena proses *frame blocking* dapat menyebabkan sinyal menjadi *discontinue* (Putra dan Resmawan, 2011). *Windowing* merupakan suatu proses untuk mengatasi masalah ini (Putra dan Resmawan, 2011).

Windowing bekerja untuk me-*window* masing-masing *frame* untuk meminimalisir sinyal menjadi *discontinue* pada awal dan akhir dari sebuah *frame*. Jika suatu *window* didefinisikan sebagai w(n), $0 \leq n \leq N-1$, dimana N merupakan jumlah dari *sample* pada tiap *frame*. Maka, hasil dari *windowing* dapat direpresentasikan dalam Rumus 2.2 (Putra dan Resmawan, 2011).

$$Y(n) = W(n) \times X(n), \quad 0 \leq n \leq N - 1 \qquad ...(2.2)$$

Keterangan:

Y(n) = Nilai sampel sinyal hasil *windowing*.

W(n) = Fungsi *window*.

X(n) = Nilai sampel sinyal hasil *frame blocking*.

N = Ukuran *frame*, merupakan kelipatan 2.

Hamming window paling banyak digunakan sebagai *window shape* dalam teknologi pemrosesan sinyal suara, dengan mempertimbangkan blok berikutnya dalam proses rantai ekstraksi ciri, mengintegrasikan semua lini frekuensi (Razak dkk., 2008). *Hamming window* meminimalkan nilai dari sinyal menuju nol pada batas *window* dan menghindari *discontinue* (Razak dkk., 2008). *Hamming window* dapat direpresentasikan dalam Rumus 2.3 (Putra dan Resmawan, 2011).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

$$w(n) = 0{,}54 - 0{,}46 \cos\left(\frac{2\pi n}{N-1}\right) \quad ...(2.3)$$

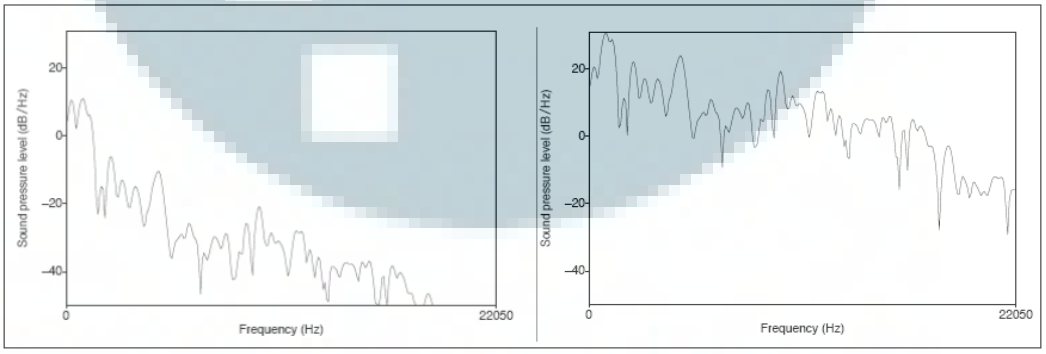
Keterangan:

n = $0 \leq n \leq N\text{-}1$.

N = Panjang *frame*.

2.4.4 Pre-Emphasis

Spektrum untuk segmen suara mempunyai energi yang lebih besar pada frekuensi yang lebih rendah dibandingkan dengan frekuensi yang lebih tinggi, dan ini dinamakan *spectral tilt* (Wahib, 2015). Penyeimbangan spektrum suara dapat dilakukan dengan menggunakan *filter pre-emphasis*. *Pre-emphasis* meningkatkan energi pada frekuensi tinggi untuk mengurangi *noise ratio* pada sinyal, sehingga dapat meningkatkan kualitas sinyal (Wahib, 2015).



Gambar 2.5 Spektral Sebelum dan Sesudah *Pre-Emphasis*
(Sumber: Wahib, 2015).

Pada Gambar 2.5 terlihat bahwa distribusi energi pada setiap frekuensi terlihat lebih seimbang setelah diimplementasikan *pre-emphasis.* Rumus 2.4 merupakan bentuk persamaan yang paling umum digunakan dalam *pre-emphasis* (Putra dan Resmawan, 2011).

$$Y[n] = S[n] - \alpha\, S[n\text{-}1] \quad ...(2.4)$$

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

Keterangan:

Y[n] = Sinyal hasil *pre-emphasis.*

S[n] = Sinyal sebelum *pre-emphasis.*

α = Koefisien *pre-emphasis,* $0.9 \leq \alpha \leq 1$.

2.4.5 Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) mengubah sinyal suara dari *time domain* menjadi *frequency domain.* Dengan menggunakan *window* sinyal sebagai *input,* maka dapat digunakan untuk menghitung FFT dan *magnitude* untuk tiap *frame* (Wahib, 2015). FFT merupakan versi lebih cepat dari *discrete fourier transform* (DFT). FFT menggunakan beberapa algoritma yang lebih efisien untuk melakukan fungsionalitas yang sama seperti DFT (Hartaman, 2009). DFT mengambil *discrete* sinyal dalam domain waktu dan mengubahnya ke dalam domain frekuensi. Dalam pemrosesan sinyal suara, hal ini sangatlah menguntungkan karena data pada domain frekuensi dapat diproses dengan lebih mudah dibandingkan data pada domain waktu, karena pada domain frekuensi keras lemahnya suara tidak seberapa berpengaruh (Hartaman, 2009). FFT dapat mereduksi jumlah perhitungan untuk setiap N data yang sama pada perhitungan DFT sehingga perhitungan menjadi lebih cepat, terutama jika nilai N yang digunakan cukup besar (Hartaman, 2009). FFT dapat direpresentasikan dalam Rumus 2.5 (Hartaman, 2009).

$$X_n = \sum_{n=0}^{N-1} X_n e^{-j2\pi nk/N} \quad ...(2.5)$$

Keterangan:

k = $0 \leq n \leq N - 1$.

Faktor dari $e^{-j2\pi nk/N}$ dapat dituliskan sebagai W_n.

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

$W_n = e^{-j2\pi/N}$.

Persamaan FFT dapat disederhanakan menjadi seperti Rumus 2.6.

$$X_n = \sum_{n=0}^{N-1} X_n W_n^{kn} \quad ...(2.6)$$

Keterangan:

$k = 0 \leq n \leq N-1$.

X_n = Sinyal masukan.

W_n^{kn} = *Twiddle factors.*

2.4.6 Mel-Frequency Wrapping

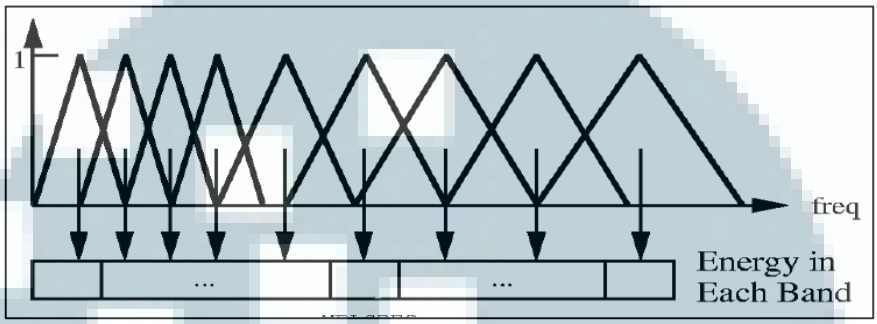
Informasi yang dibawa oleh frekuensi yang rendah dari sinyal suara lebih penting dibandingkan dengan frekeunsi yang tinggi (Razak dkk., 2008). Skala *mel* diaplikasikan untuk lebih menekankan pada komponen frekuensi yang rendah. Skala *mel* merupakan satuan ukuran khusus atau skala dari *pitch* dari nada yang diterima (Razak dkk., 2008).

Persepsi manusia terhadap frekuensi dari sinyal suara tidak mengikuti skala linear. Frekuensi yang sebenarnya (dalam Hz) pada sebuah sinyal diukur oleh manusia secara subjektif dengan menggunakan skala *mel* (Putra dan Resmawan, 2011). Skala *mel* merupakan pemetaan frekuensi secara linear pada frekuensi di bawah 1 kHz dan merupakan skala logaritmik pada frekuensi di atas 1 kHz (Putra dan Resmawan, 2011). Sebagai contoh, *pitch* dari 1 kHz suara, 40 dB di atas batas ambang pendengaran, didefinisikan sebagai 1000 *mel* (Patel dan Prasad, 2013). Skala *mel* menentukan ruang dari *filter* dan menentukan seberapa lebar *filter* yang akan terbentuk, untuk menghitung skala *mel* dengan diberikan frekuensi (f) dalam Hz dapat menggunakan Rumus 2.7 (Muda dkk., 2010).

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

$$f_{mel}(f) = 2595 \times \log_{10}(1+\frac{f}{700}) \quad ...(2.7)$$

Jarak frekuensi dalam *spectrum* FFT sangat lebar dan sinyal suara tidak mengikuti skala linear (Muda dkk., 2010). *Bank* dari *filter-filter* menurut skala *mel* dapat dilihat pada Gambar 2.6.



Gambar 2.6 *Mel Scale Filter Bank*
(Sumber: Muda dkk., 2010)

Gambar 2.6 menunjukkan kumpulan *filter* berbentuk segitiga yang digunakan untuk mengetahui ukuran energi dari *frequency band* tertentu dalam sinyal suara. Setiap ukuran dari *filter* frekuensi bergantung tehadap ukuran *filter* segitiga yang terbentuk, *filter* pertama sangat sempit, dan mengindikasikan berapa banyak energi yang ada mendekati 0 hertz (Muda dkk., 2010). Semakin frekuensi meningkat bentuk *filter* akan bertambah lebar, karena sudah semakin mengabaikan variasi dari frekuensi (Muda dkk., 2010).

Sebuah *filterbank* dengan i sebagai banyaknya *filterbank*, akan dilalui oleh *magnitude spectrum* hasil dari proses *Fast Fourier Transform* (FFT) dengan menggunakan Rumus 2.8 (Ganchev dkk., 2005).

$$H_i(k) = \begin{cases} 0 & \text{untuk } K < f_{b_{i-1}} \\ \frac{(k - f_{b_{i-1}})}{(f_{b_i} - f_{b_{i-1}})} & \text{untuk } f_{b_{i-1}} \le k \le f_{b_i} \\ \frac{(f_{b_{i+1}} - k)}{(f_{b_{i+1}} - f_{b_i})} & \text{untuk } f_{b_i} \le k \le f_{b_{i+1}} \\ 0 & \text{untuk } K > f_{b_{i+1}} \end{cases} \quad ...(2.8)$$

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

Keterangan:

H_i = Sinyal hasil *filterbank.*

i = *Filter* ke-1, 2, …, jumlah *filter*.

f_{b_i} = Frekuensi hasil FFT.

K = 1, 2, …, jumlah data hasil FFT.

2.4.7 Discrete Cosine Transform (DCT)

Proses *Discrete Cosine Transform* (DCT) merupakan proses terakhir dalam proses MFCC yang menghasilkan *mel-cepstrum* (Putra dan Resmawan, 2011). Karena *filterbank* bersifat *overlapping*, hasil dari *filterbank* yang berupa *mel-spectrum* akan didekorelasikan, sehingga menghasilkan representasi yang baik dari *property spectral* lokal (Putra dan Resmawan, 2011). Konsep dasar dari DCT sama dengan *inverse fourier transform*, tetapi hasil dari DCT mendekati *Principle Component Analysis* (PCA), sehingga DCT lebih banyak digunakan dalam proses MFCC (Putra dan Resmawan, 2011). Perhitungan untuk mendapatkan *mel-cepstrum* dapat direpresentasikan dengan Rumus 2.9 (Putra dan Resmawan, 2011).

$$\tau_n = \sum_{k=1}^{K} (\log S_k) \cos[n(k - \tfrac{1}{2})\tfrac{\pi}{k}] \quad ...(2.9)$$

Keterangan:

τ_n = *Mel-cepstrum.*

S_k = Keluaran dari proses *filterbank* pada indeks k.

k = Jumlah koefisien yang diharapkan, 1,2,...,k.

n = Nilai sampel sinyal hasil *filterbank.*

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

2.5 Learning Vector Quantization (LVQ)

Metode klasifikasi *Learning Vector Quantization* (LVQ) merupakan jaringan saraf tiruan dengan tipe arsitektur lapisan jamak (*multi layer network*) dimana lapisan masukan terhubung dengan setiap *neuron* pada keluaran dengan satu *hidden layer* di antaranya (Hidayati dan Warsito, 2010). Metode ini merupakan metode pembelajaran dengan model *supervised* (Hidayati dan Warsito, 2010).

Metode LVQ menggunakan vektor acuan dari satu unit keluaran yang menjadi acuan bagi kelas atau kategori yang mewakili keluaran tersebut. Langkah yang dituju adalah mengelompokkan vektor masukan berdasarkan kedekatan jarak vektor masukan terhadap bobot (Kusumadewi, 2003). Bobot merupakan nilai matematis dari koneksi yang memindahkan data dari satu lapisan ke lapisan lainnya, yang berfungsi untuk mengatur jaringan sehingga dapat menghasilkan *output* yang diinginkan (Kusumadewi, 2003). Metode LVQ merupakan salah satu metode jaringan saraf tiruan yang berbasis kompetisi dengan mekanisme *squared Euclidean distance* (Kusumadewi, 2003). Metode LVQ menggunakan *squared Euclidean distance* dalam memilih vektor pewakil pemenang untuk menentukan kategori vektor masukan. *Euclidean distance* merupakan jarak antara dua data poin yang akan diukur sehingga membentuk sebuah *metric space* (Singla dan Karambir, 2012). Proses pembelajaran LVQ merupakan proses pembelajaran *supervised* atau dengan kata lain menggunakan pengarahan, dengan tujuan mendapatkan vektor-vektor pewakil yang akan melakukan kuantisasi terhadap vektor masukan (Kusumadewi, 2003).

Setelah menentukan vektor-vektor pewakil untuk digunakan pada saat pelatihan, maka dengan pengarahan vektor pewakil tersebut akan mengenali target

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

yang telah diberikan bersamaan dengan masukan (Kusumadewi, 2003). Dalam proses pelatihan data, unit-unit keluaran diarahkan kepada suatu *decision surface* teori dengan memperbarui bobot pada proses pelatihan data (Kusumadewi, 2003). LVQ diarahkan untuk menentukan unit keluaran yang paling sesuai dengan target dari vektor masukannya melalui cara penggeseran posisi vektor pewakil (Kusumadewi, 2003). Apabila vektor data latih dikelompokkan sama dengan vektor pewakil pemenang, maka vektor pewakil digeser mendekati vektor data latih, tetapi jika vektor data latih dikelompokkan tidak sama dengan vektor pewakil pemenang, maka vektor pewakil digeser menjauhi vektor data latih (Kusumadewi, 2003). Apabila terdapat sebuah vektor masukan yang ingin dicocokkan, maka vektor masukan akan ditentukan masuk dalam kelompok mana dengan mencari nilai *Euclidean distance* terendah dengan vektor pewakil (Kusumadewi, 2003).

Parameter-parameter pembelajaran pada LVQ antara lain sebagai berikut (Hidayati dan Warsito, 2010).

1. Inisialisasi vektor pewakil

   Inisialisasi pada LVQ dapat dilakukan dengan beberapa cara, antara lain memilih salah satu vektor data latih pada tiap kelas, memilih vektor data latih secara acak, dan inisialisasi awal dengan ‘0’.

2. Alfa (*Learning rate*)

   Laju pembelajaran dalam metode klasifikasi LVQ sangat berpengaruh pada saat pergeseran vektor pewakil. Jika nilai alfa terlalu besar, maka algoritma akan menjadi tidak stabil, sebaliknya jika nilai alfa terlalu kecil, maka prosesnya akan terlalu lama. Nilai alfa adalah $0 < \alpha < 1$.

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

3. DecAlfa (Penurunan *Learning Rate*)

   Nilai penurunan tingkat pembelajaran yang digunakan untuk mengurangkan nilai alfa pada proses pembelajaran.

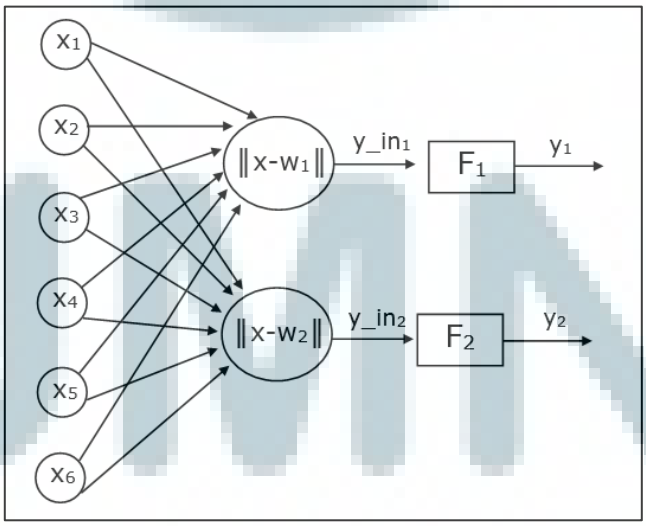
4. MinAlfa (Minimum *Learning Rate*)

   Nilai minimal tingkat pembelajaran yang diperbolehkan selama proses pembelajaran.

5. MaxEpoch (Maksimum *epoch*)

   Nilai *epoch* atau iterasi maksimum yang boleh dilakukan selama proses pelatihan. Semakin tinggi nilai maksismum *epoch* yang ditentukan maka proses pelatihan yang dilakukan akan semakin banyak. Jika nilai maksimum *epoch* yang ditentukan semakin rendah maka proses pelatihan akan semakin sedikit. Iterasi proses pelatihan akan dihentikan jika nilai *epoch* melebihi nilai *epoch* maksimum.

2.5.1 Arsitektur Jaringan dan Algoritma LVQ



Gambar 2.7 Arsitektur Jaringan LVQ dengan Enam Unit *Input* dan Dua Vektor Bobot
(Sumber: Kusumadewi, 2003)

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

Gambar 2.7 menggambarkan jaringan LVQ yang memiliki enam *input layer* dengan dua unit *neuron* pada *output layers*. W1 dan W2 adalah bobot yang menghubungkan *input layers* ke *output layers* (Putri, 2012). Fungsi aktivasi (F) yang digunakan dalam jaringan LVQ merupakan fungsi aktivasi identitas (Hidayati dan Warsito, 2010). Fungsi aktivasi identitas akan membuat hasil keluaran sama dengan masukan, sesuai dengan rumus fungsi identitas yaitu y=x (Hidayati dan Warsito, 2010). Setiap fungsi aktivitas F akan melakukan pemetaan setiap y_In ke klasifikasi y1 atau y2. Pada F1, jika $||x - w_1|| < ||x - w_2||$ maka y_In_1 dipetakan ke y1=1 dan dipetakan ke y1=0 jika $||x - w_1|| > ||x - w_2||$ (Putri, 2012). Kondisi ini juga berlaku untuk F2 dengan kondisi yang sama dengan F1.

Pada intinya, metode LVQ akan mencari unit keluaran yang paling mirip dengan vektor masukan. Jika vektor data latih bukan bagian dari kelas yang sama, maka vektor bobot digeser menjauhi vektor masukan tersebut. Sebaliknya jika vektor data latih merupakan bagian kelas yang sama, maka vektor bobot digeser mendekati vektor masukan tersebut (Putri, 2012).

Berikut algoritma metode pembelajaran LVQ (Kusumadewi, 2003).

1. Tetapkan: Vektor-vektor acuan, Alfa (α), MinAlfa, MaxEpoch, dan *epoch* = 0.
2. Masukan vektor *input* dan target.
3. Selama (*epoch* < MaxEpoch) atau (α > MinAlfa), lakukan langkah nomor 4-7.
4. Untuk setiap vektor masukan x, lakukan langkah nomor 5 sampai 6.
5. Tentukan j hingga $|| x - w_j ||$ bernilai minimum

$$D_i = \sqrt{\sum_{j=1}^{m}(Wji - Xi)^2} \quad ...(2.10)$$

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016

6. Perbarui nilai bobot Wj.

   a. Jika T = Cj

$$Wj(baru) = Wj(lama) + \alpha\ (x\text{-}Wj(lama)) \quad ...(2.11)$$

   b. Jika T ≠ Cj

$$Wj(baru) = Wj(lama) - \alpha\ (x\text{-}Wj(lama)) \quad ...(2.12)$$

7. Kalikan nilai *learning rate* dengan MinAlfa.

$$\alpha = \alpha \times MinAlfa \quad ...(2.13)$$

Keterangan:

X = Vektor data latih (x1,...,xi,...,xn).

m = Jumlah data *input*.

T = Kategori vektor data latih yang benar untuk pelatihan.

Wj = Vektor bobot untuk unit keluaran ke-k (W1j,...,Wij,...,Wim).

Cj = Kategori atau kelas yang diwakili oleh nilai urut keluaran ke-j (hasil pelatihan).

$\| x - w_j \|$ = *Euclidean distances* antara vektor-vektor masukan dan vektor bobot dari unit keluaran ke-j.

Rancang bangun..., Muhammad Idham Choudry, FTI UMN, 2016