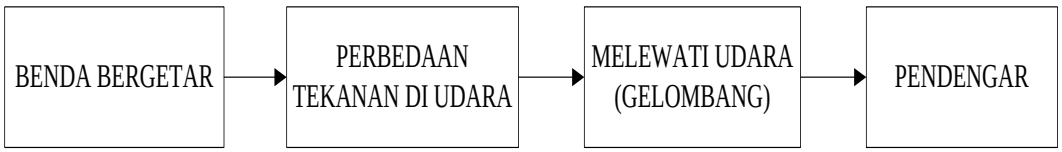
BAB 2
LANDASAN TEORI

2.1 Suara

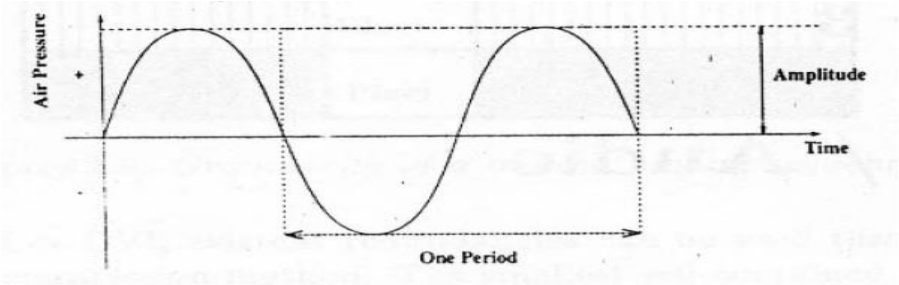
Suara merupakan fenomena fisik yang dihasilkan oleh getaran benda. Getaran benda tersebut berupa sinyal analog dan amplitudo yang berubah secara kontinyu terhadap waktu. Suara atau bunyi biasanya merambat melalui udara dan tidak bisa merambat melalui ruang hampa [6]. Berikut pada Gambar 2.1. merupakan diagram blok suara.



Gambar 2.1 Diagram Blok Suara

2.2 Konsep Dasar Sinyal Suara

Suara dihasilkan oleh getaran suatu benda. Selama bergetar, perbedaan tekanan terjadi di udara dan sekitarnya. Pola osilasi yang terjadi dinamakan sebagai gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu yang disebut dengan periode [6].

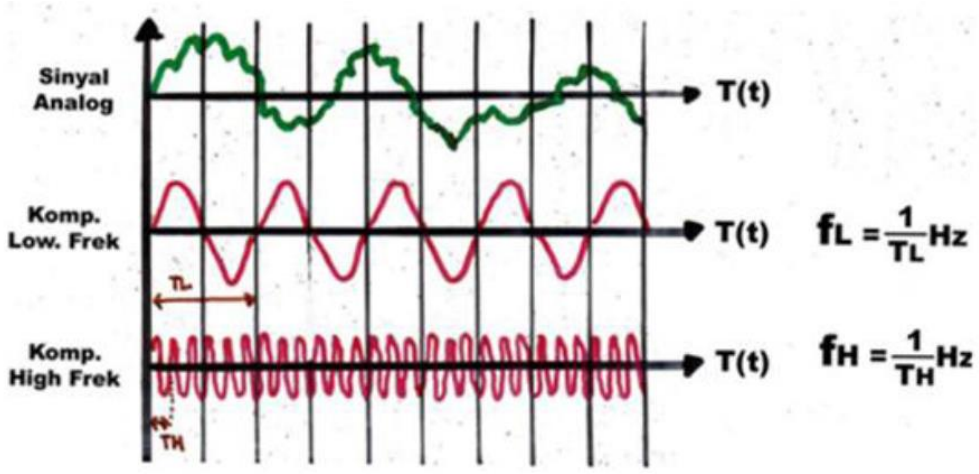


Gambar 2.2 Gelombang Dalam Periode Tertentu

2.3 Frekuensi

Frekuensi adalah jumlah gelombang yang terjadi setiap detik. Manusia menghasilkan suara dengan frekuensi 300Hz – 3400Hz, sehingga rentangnya 3100Hz. Suara yang berada dalam range pendengaran manusia disebut sebagai audio dan gelombangnya sebagai acoustic signals. Suara diluar pendengaran

manusia dapat dikatakan sebagai noise (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi dan tidak dapat didengar manusia [6].



Gambar 2. 3 Representasi Sinyal Analog

2.4 Amplitudo

Amplitudo adalah keras lemahnya bunyi atau tinggi rendahnya gelombang. Satuan amplitudo adalah desibel (db), bunyi dapat merusak telinga jika tingkat volumenya lebih besar dari 85 db dan pada ukuran 130db akan mampu membuat hancur gendang telinga [6].

2.5 Velocity

Velocity adalah kecepatan perambatan gelombang bunyi sampai ke telinga pendengar. Satuan yang digunakan adalah m/s pada udara kering dengan suhu 20º C (68º F) dengan kecepatan rambat suara sekitar 343 m/s [6].

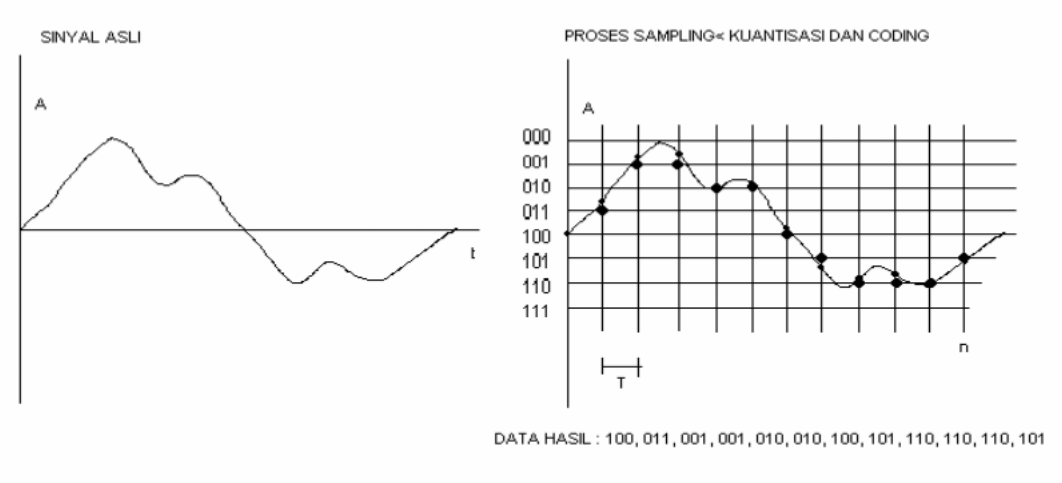
2.6 Representasi Sinyal Suara

Gelombang sinyal suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan sample. Sampling adalah proses mencuplik sinyal analog menjadi beberapa potongan waktu, kemudian sample tersebut diberi nilai eksak dan nilainya diberikan dalam bentuk digital [7].

2.7 Konversi Analog Menjadi Digital

Signal – signal yang natural pada umumnya seperti signal suara merupakan signal continue dimana memiliki nilai yang tidak terbatas. Sedangkan pada komputer, semua signal yang dapat diproses oleh komputer hanyalah signal discrete atau sering dikenal sebagai istilah *digital signal*. Agar signal natural dapat diproses oleh komputer, maka harus diubah terlebih dahulu dari data signal *continue* menjadi *discrete* [7]. Hal itu dapat dilakukan melalui 3 proses, diantaranya sebagai berikut:

1. Proses sampling adalah suatu proses untuk mengambil data *signal continue* untuk setiap periode tertentu. Dalam melakukan proses sampling data, berlaku aturan Nyquist, yaitu bahwa frekuensi sampling (sampling *rate*) minimal harus 2 kali lebih tinggi dari frekuensi maksimum yang akan di sampling. Jika signal sampling kurang dari 2 kali frekuensi maksimum signal yang akan di sampling, maka akan timbul efek *aliasing*. *Aliasing* adalah suatu efek dimana signal yang dihasilkan memiliki frekuensi yang berbeda dengan signal aslinya.
2. Proses kuantisasi adalah proses untuk membulatkan nilai data ke dalam bilangan-bilangan tertentu yang telah ditentukan terlebih dahulu. Semakin banyak *level* yang dipakai maka semakin akurat pula data *signal* yang disimpan tetapi akan menghasilkan ukuran data besar dan proses yang lama.
3. Proses pengkodean adalah proses pemberian kode untuk tiap-tiap data *signal* yang telah terkuantisasi berdasarkan *level* yang ditempati.



Gambar 2. 4 Proses Pembentukan Sinyal Digital

2.8 Wav Audio

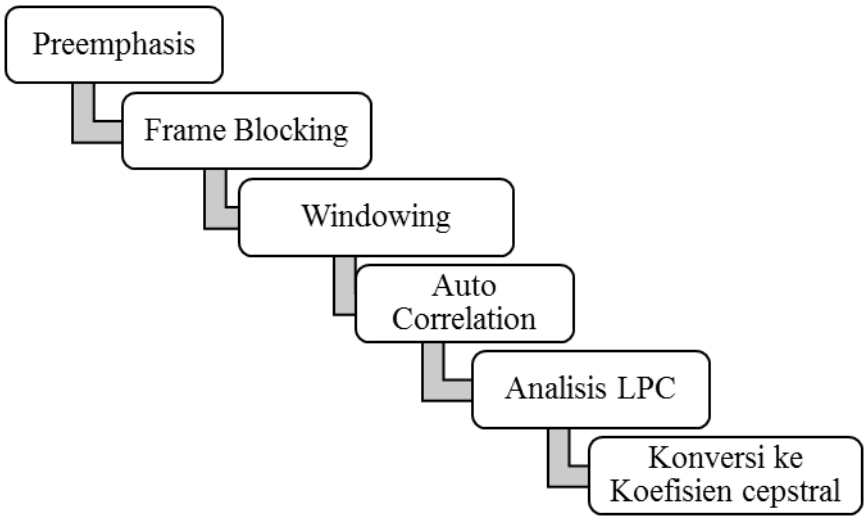
WAV adalah singkatan dari *waveform audio* format yang merupakan standar format *audio* untuk Microsoft dan IBM. WAV biasanya menggunakan *coding* PCM (*pulse code modulation*) dan jarang sekali digunakan di internet karena ukuranya relatif besar dengan maksimal ukuran file WAV adalah 2GB. File WAV terdiri dari 3 bagian, yaitu *main chunk*, *format chunk* dan *data chunk*. Sinyal suara yang direpresentasikan file WAV dalam bentuk *discrete*, berupa deret bilangan yang merepresentasikan amplitude dalam domain waktu. Pada bagian file header terdapat informasi tentang file wav tersebut, diantaranya menyatakan nilai *sample rate*, jumlah *channel*, dan bit per *sample*. Dan pada file *header* juga dapat diketahui jumlah *sample* yang dicuplik dari sinyal analog tiap detik [7]

2.9 Linear Predictive Coding

Linear Predictive Coding (LPC) adalah metode digital untuk melakukan proses *encoding* sinyal analog, dimana sebuah nilai tertentu di prediksi oleh sebuah fungsi *linear* nilai sebelumnya dari sinyal tersebut [2]. Ada dua alasan utama penggunaan metode LPC, yaitu:

a. LPC dapat digunakan untuk memprediksi trayek dari sebuah sinyal. Dalam domain frekuensi, prediksi trayek *ekuivalen* dengan permodelan *spectrum* sinyal [2]
b. LPC mampu menghilangkan bagian yang dapat terprediksi dalam upaya untuk menghindari bagian redudansi pada saat *re-transmitting* yang dapat diprediksi oleh *receiver* dan sekaligus dapat menghemat penyimpanan, *bandwith*, dan waktu [2]

Langkah-langkah dari pengolahan sinyal dengan LPC untuk memperoleh koefisien *cepstral* yaitu:



Gambar 2. 5 Blok Diagram Untuk LPC

Pada Gambar 2.5 dapat dilihat proses LPC, proses preemphasis adalah proses dimana sinyal suara masukan dilakukak perbaikan signal dari gangguan noise, lalu proses frame blocking yang bertujuan untuk membagi sampel sinyal menjadi beberapa frame, lalu proses windowing yaitu proses untuk mengurangi efek diskontinuitas pada ujung ujung frame, lalu proses auto correlation [2].

2.9.1 Pre-Emphasis

Pada tahap ini digunakan untuk mendatarkan *spectral* sinyal dan meningkatkan keaslian sinyal pada pemrosesan sinyal yang selanjutnya. Persamaan untuk perhitungan pada proses *pre-emphasis* dapat dilihat pada persamaan 2.1 berikut:

$$s'(n) = s(n) - 0.9375\ x\ s(n-1) \qquad (2.1)$$

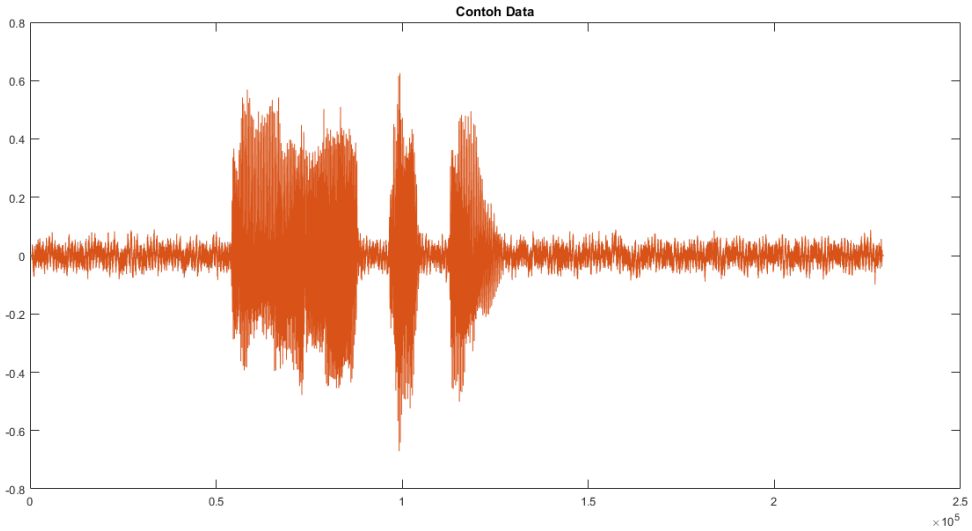
Dimana:

$s'(n)$: Fungsi dari pre-emphasis

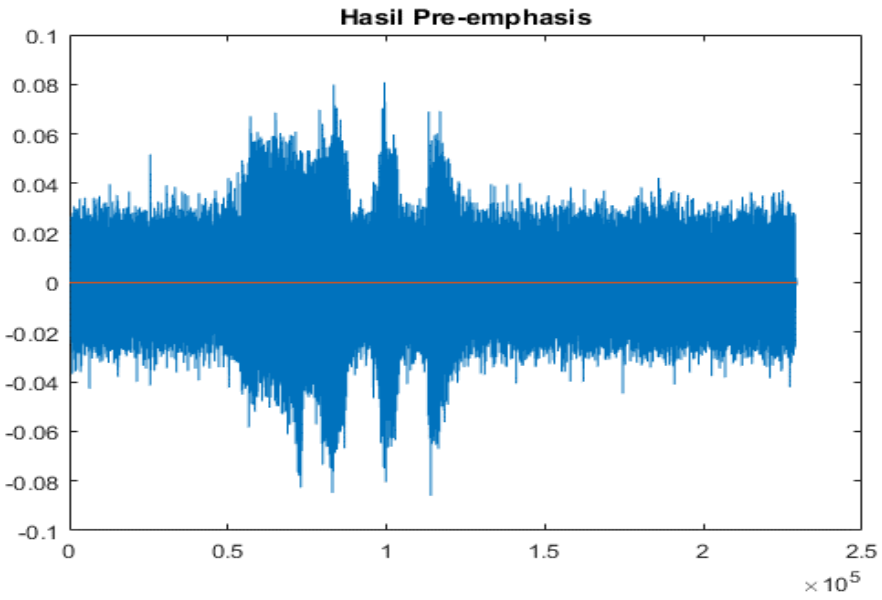
$s(n)$: Data sinyal suara ke n

$s(n-1)$: Data sinyal suara ke $n-1$

Hasil dari proses *pre-emphasis* berbeda dengan data aslinya namun tidak merubah data yang terkandung di dalam data aslinya. Pada Gambar 2.4 akan menampilkan data awal atau data masukan sebelum dilakukan proses *pre-emphasis*, lalu pada Gambar 2.5 merupakan contoh hasil dari proses *pre-emphasis*.



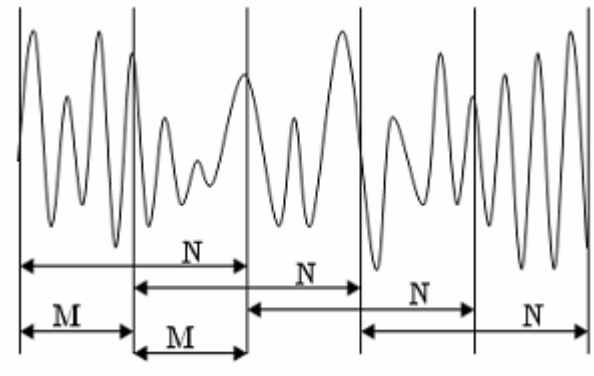
Gambar 2. 6 Contoh Data Awal



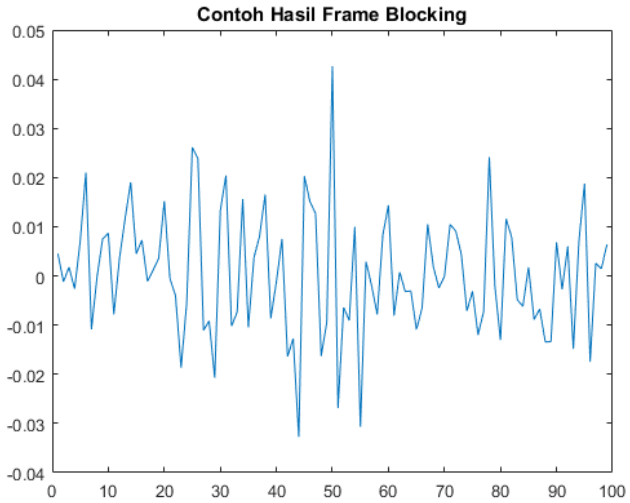
Gambar 2. 7 Contoh Hasil *Pre-Emphasis*

2.9.2 Frame Blocking

Tahapan ini mengelompokkan sinyal hasil proses *pre-emphasis* ke dalam *frame-frame* dengan ukuran sebesar N data yang dimasukkan. *Frame* ini berurutan dengan pemisahan antara kedua bingkai sebesar M data, sehingga $M = \frac{1}{3}N$, dimana N melambangkan jumlah dari blok sinyal. Adapun contoh dari *frame blocking* dapat dilihat pada Gambar 2.8 berikut [4]:



Gambar 2. 8 Contoh Frame Blocking



Gambar 2. 9 Contoh hasil Frame Blocking

2.9.3 Windowing

Setelah proses *frame blocking* dilakukan, maka proses selanjutnya yaitu proses *windowing*. Proses ini menggunakan persamaan *Hamming*, dimana setiap data dari *frame blocking* dimasukan kedalam proses *Hamming* [8]. Persamaan untuk perhitungan pada proses *Hamming* dapat dilihat pada persamaan 2.2 berikut:

$$W(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N} - 1\right) \quad (2.2)$$

Dimana :

$W(n)$: Fungsi Hamming

N : Jumlah *sample* dari blok sinyal

Tujuan *windowing* adalah untuk meruncingkan atau memperhalus sinyal sehingga dapat memperkecil sinyal kesalahan pada batas daerah tersebut.

2.9.4 Auto Korelasi

Korelasi adalah metode untuk mengetahui tingkat keeratan hubungan dua peubah ataulebih yang digambarkan oleh besarnya koefisien korelasi [4].

Korelasi antara dua bentuk gelombang adalah suatu ukuran kesamaan keduanya. Bentuk gelombang dibandingkan pada selang waktu berbeda, dan "kesamaan" dihitung pada selang waktu masing-masing. Hasil suatu korelasi

adalah ukuran kesamaan sebagai fungsi penyimpangan waktu antara permulaan kedua bentuk gelombang. Fungsi autokorelasi adalah korelasi suatu bentuk gelombang dengan dirinya sendiri. Kesamaan tepat pada selang waktu nol, sedangkan ketidaksamaan terus meningkat seiring bertambahnya selang waktu. Definisi matematika untuk menggambarkan fungsi autokorelasi ditunjukkan pada persamaan 1 untuk suatu fungsi diskrit tanpa batas x(n) sedangkan definisi matematika autokorelasi suatu fungsi diskrit terbatas x(n) untuk ukuran N [9]

Setiap himpunan *windowing* di autokorelasi-kan sehingga didapatkan sebuah himpunan *koefisien* $p-1$, dengan p adalah orde LPC yg diharapkan [4].

Berikut adalah persamaan dari fungsi auto korelasi:

$$rl(m) = \sum_{n=0}^{N-1-m} xi(n) * xi(n+m) \quad (2.3)$$

2.9.5 Analisis LPC

Pada tahap ini setiap *frame* dengan autokorelasi ke-$(p+1)$ akan dikonversi ke bentuk parameter LPC yang dapat berupa koefisien LPC, koefisien refleksi, koefisien *cepstral* atau transformasi yang lainnya sesuai kebutuhan [2].

$$E_{(o)}=r(0) \quad (2.4)$$

$$k_m=\{r(m)-\Sigma_{m-1j=1}\alpha_{j(m-1)}r(|m-j|)/E_{(m-1)},\ 1 \le m \le p \quad (2.5)$$

$$\alpha_{m(m)}= k_m \quad (2.6)$$

$$\alpha_{j(m)} = \alpha_{j(m-1)}- k_m\, \alpha_{m-j(m-1)},\ 1 \le j \le m-1 \quad (2.7)$$

$$E(m)=(1- km2)\ E(m-1) \quad (2.8)$$

Dimana:

E = nilai error

r = nilai auto correlation

m = posisi autocorrelation

p = orde LPC

k = nilai Analisis LPC

2.9.6 Konversi Parameter LPC Menjadi Koefisien *Cepstral*

Rangkaian parameter yang sangat penting, yang dapat diturunkan secara langsung dari rangkaian koefisien LPC adalah koefisien cepstral C_m[1]. Persamaan untuk perhitungan koefisien *cepstral* dapat dilihat pada persamaan 2.3 berikut:

$$C_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) . C_k . a_{m-k} \quad 1 \leq m \leq p \tag{2.9}$$

$$C_m = \sum_{k=m-p}^{m-1} \left(\frac{k}{m}\right) . C_k . a_{m-k} \quad m > p \tag{2.10}$$

2.10 Jaringan Syaraf Tiruan

JST (Jaringan Syaraf Tiruan) adalah merupakan salah satu representasi buatan dari otak manusa yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia [10]. Ada beberapa tipe jaringan syaraf tiruan, namun hampir semuanya memiliki komponen yang sama. Jaringan syaraf tiruan terdiri dari beberapa neuron, dimana neuron tersebut akan mentransformasikan informasi. Informasi (disebut dengan: input) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya [10].

Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya. Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan input sampai ke lapisan output melalui lapisan lainnya yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). JST menyerupai otak manusia dalam dua hal [10]. yaitu:

a. Pengetahuan diperoleh jaringan melalui proses belajar
b. Kekuatan hubungan antar sel syaraf (neuron) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan

JST merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut neuron, sehingga mampu melaksanankan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi [10].

2.10.1 Supervised Learning

Supervised atau *active learning* adalah proses belajar membutuhkan guru. Yang dimaksud guru disini adalah sesuatu yang memiliki pengetahuan tentang lingkungan. Guru bisa direpresentasikan sebagai sekumpulan sampel *input-output*. Pembangunan pengetahuan dilakukan oleh guru dengan memberikan respon yang diinginkan kepada JST. Parameter-parameter jaringan berubah-ubah berdasarkan vektor latih dan sinyal kesalahan (sinyal kesalahan adalah perbedaan antara keluaran JST dan respon yang diinginkan). Proses perubahan ini dilakukan secara berulang-ulang, selangkah demi selangkah, dengan tujuan agar JST bisa memiliki kemampuan yang mirip dengan gurunya. Dengan kata lain, JST dilatih untuk dapat memetakan sekumpulan sampel *input-output* dengan akurasi yang tinggi. Pada proses belajar ini, output yang diharapkan telah diketahui sebelumnya [10].

2.10.2 Pembelajaran Dengan Pengawasan

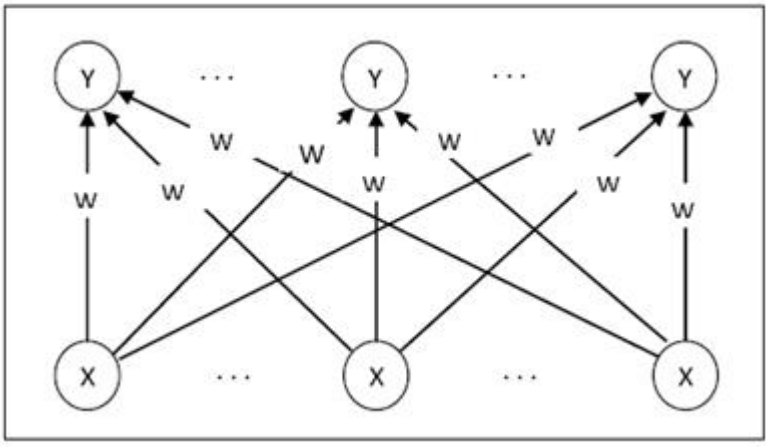
Ada banyak metode pembelajaran yang termasuk dalam *supervised learning* dan sudah pernah diimplementasikan dalam beberapa kasus. Namun dalam penelitian ini, metode pembelajaran yang digunakan yaitu *Learning Vector Quantization* (LVQ). Di bawah ini dijelaskan beberapa metode pembelajaran yang termasuk ke dalam *supervised learning* [10].

2.11 Learning Vector Quantization (LVQ)

Learning Vector Quantization adalah suatu metode klasifikasi pola yang masing-masing unit output mewakili kategori atau kelas tertentu (beberapa unit output seharusnya digunakan untuk masing-masing kelas). Vektor bobot untuk sebuah unit output sering dinyatakan sebagai sebuah vektor referensi. Diasumsikan bahwa serangkaian pola pelatihan dengan klasifikasi yang tersedia bersama dengan sistribusi awal dari vektor referensi [10]. Sesudah pelatihan, sebuah jaringan LVQ mengklasifikasikan vektor input dengan menugaskan ke kelas yang sama sebagai unit output, sedangkan yang mempunyai vektor referensi diklasifikasikan sebaga vektor input.

2.11.1 Arsitektur Learning Vector Quantization

Arsitektur dari sebuah jaringan syaraf LVQ dapat dilihat pada Gambar 2.8. berikut ini [5]:



Gambar 2. 10 Arsitekur Jaringan Syaraf LVQ

2.11.2 Algoritma LVQ

Motivasi untuk sebuah algoritma yang diterapkan pada jaringan syaraf LVQ adalah untuk menghasilkan unit output yang terdekat dengan vektor input. Hal tersebut akan berakhir jika 1 dan w berada pada kelas yang sama, maka bobot

dipindahkan ke vektor input yang baru dan jika x dan w berada pada kelas yang berbeda, maka bobot akan dipindahkan dari vektor input.

Ada beberapa notasi yang akan digunakan, yaitu:

x : vektor pelatihan $((x1, \dots, xj, \dots xn)$

T : kategori atau kelas yang benar untuk vektor pelatihan

wj : vektor bobot untuk unit output $(w1j, \dots, wij, \dots, wn)$

cj : kategori atau kelas yang direpresentasikan oleh unit output j

$\|x - wj\|$: jarak *Euclidean* antara vektor input (vektor bobot) dan unit output j

Adapun algoritma dari LVQ adalah:

1. Inisialisasikan vektor referensi dan *learning rate*, $\alpha(0)$
2. Selama kondisi berhenti bernilai salah, kerjakan:
   a. Untuk masing-masing pelatihan vektor input x
   b. Temukan j sehingga $\|x - wj\|$ bernilai minimum
   c. Perbaiki wj dengan :
      i. Jika $T = cj$ maka :

$$wj\ (baru) = wj\ (lama) + \ \alpha * (x - wj\ (lama)) \quad (2.11)$$

      ii. Jika $T \neq cj$ maka :

$$wj\ (baru) = wj\ (lama) - \ \alpha * \ (x - wj(lama))\ ) \quad (2.12)$$

   d. Kurangi learning rate
   e. Tes kondisi berhenti

2.12 Unified Modelling Language (UML)

Unified modelling language merupakan bahasa standar untuk merancang dan mendokumentasikan perangkat lunak dengan cara berorientasi objek. Ada beberapa diagram yang digunakan pada proses pembuatan perangkat lunak berorientasi objek, diantaranya *use case diagram*, *activity diagram*, *class diagram* dan *sequence diagram* [11].

2.12.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk tingkah laku (*behavior*) pada sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang terdapat pada sistem. Terdapat dua hal utama yang diperlukan dalam pembentukan suatu *use case diagram*, yaitu aktor dan *use case* [11].

1. Aktor merupakan orang, benda maupun sistem lain yang berinteraksi dengan sistem yang akan dibangun
2. *Use case* merupakan fungsionalitas atau layanan yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor [3].

2.12.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem, proses bisnis atau menu yang ada pada perangkat lunak. Setiap *use case* yang telah dibentuk digambarkan aktivitasnya dalam *activity diagram*, mulai dari peran aktor, peran sistem, dan *decision* [11]. *Activity diagram* juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis, dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem.
2. Urutan atau pengelompokkan tampilan dari sistem /*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan tampilan antarmuka.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

2.12.3 Class Diagram

Class diagram menggambarkan interaksi dan relasi antar kelas yang ada di dalam suatu sistem. Kelas memiliki atribut dan metode. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. *Class diagram* mempunyai 3 macam hubungan, yaitu [11]:

1. Assosiation

Association adalah hubungan antara bagian dari dua kelas. *Association* terjadi antara dua kelas akan terjadi jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam *class diagram*, sebuah *association* adalah penghubung antara dua kelas.

2. Aggregation

Relasi yang membuat *class* yang saling terikat satu sama lain namun tidak terlalu berkegantungan.

3. Composition

Relasi agregasi dengan mengikat satu sama lain dengan ikatan yang sangat kuat dan saling berkegantungan.

4. Dependency

Hubungan antar *class* dimana *class* yang memiliki relasi *dependency* menggunakan *class* lain sebagai atribut pada *method*.

5. Realization

Hubungan antar *class* dimana sebuah *class* memiliki keharusan untuk mengikuti aturan yang ditetapkan *class* lainnya.

2.12.4 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu [11].

Banyaknya *sequence diagram* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak. Penomoran pesan berdasarkan urutan iteraksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu [11].

2.13 Pengujian Sistem

Pengujian adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikan atau mengidentifikasi perbedaan-

perbedaan antara hasil yang diharapkan dengan hasil yang terjadi [10]. Pengujian seharusnya meliputi tiga konsep berikut.

1. Demonstrasi validitas perangkat lunak pada masing-masing tahap di siklus pengembangan sistem.
2. Penentuan validitas sistem akhir dikaitkan dengan kebutuhan pemakai.
3. Pemeriksaan perilaku sistem dengan mengeksekusi sistem pada data sampel pengujian.

Awalnya pengujian diartikan sebagai aktivitas yang dapat atau hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian seharusnya dilakukan dalam skala lebih luas. Pengujian dapat dilakukan begitu spesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik di pengujian. Kategori pengujian dapat dikategorikan menjadi dua [16], yaitu:

1. Berdasarkan ketersediaan logik sistem, terdiri dari *Black box testing* dan *White box testing*.
2. Berdasarkan arah pengujian, terdiri dari Pengujian *top down* dan Pengujian *bottom up*.

2.13.1 Pengujian Akurasi K-Fold Cross Validation

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (true value atau reference value). Dalam teknik ini dataset dibagi menjadi sejumlah K-buah partisi secara acak. Kemudian dilakukan sejumlah K-kali eksperimen, dimana masing-masing eksperimen menggunakan data partisi ke-K sebagai data testing dan memanfaatkan sisa partisi lainnya sebagai data training.

2.14 Bahasa Pemograman

Bahasa pemrograman diperlukan untuk menjalankan instruksi-instruksi apa yang harus dilakukan komputer. Komputer tidak bisa memahami bahasa manusia, sehingga diperlukan penggunaan bahasa komputer di dalam program komputer. Penelitian ini menggunakan bahasa pemrograman C# untuk

mengembangkan aplikasi dan habasa pemrograman SQL untuk mengelola *database* [4].

2.14.1 C#

C# adalah salah satu bahasa pemrogrman intermediate yang digunakan oleh programmer untuk membuat executable programs / program yang dapat dieksekusi. C# dapat memisahkan jarak antara aplikasi yang powerfull seperti C++ dan sangat mudah sekali untuk digunakan, C# memiliki ekstension **.CS**. Sebelum menjalankan apliaksi yang dibuat dengan C#, user memerlukan Common Language Runtime (CLR) untuk mengeksekusi program yang degenerate dengan C#. berikut merupakan kelebihan C# [12]:

a. *Flexible* : C# program dapat dieksekusi di mesin komputer sendiri atau ditransmisikan melalui web dan dieksekusi di komputer lainnya.
b. *Powerfull* : C# memiliki sekumpulan perintah yang sama dengan C++ yang kaya akan fitur yang lengkap tetapi dengan gaya bahasa yang lebih diperhalus sehingga memudahkan penggunanya.
c. *Easier to use* : C# memodifikasi perintah yang sepenuhnya sama dengan C++ dan memberitahu dimana letak kesalahan kita bila ada kesalahan dalam aplikasi, hal ini dapat mengurangi waktu kita dalam mencari error.
d. *Visually Oriented* : The .**NET** library code yang digunakan oleh C# menyediakan bantuan yang dibutuhkan untuk membuat tampilan yang *complicated* dengan *frames, dropdown, tabbed windows, group button, scroll bar, background image,* dan lainnya.
e. *Secure* : semua bahasa pemrograman yang digunakan untuk kebutuhan internet mesti memiliki *security* yang benar-benar aman untuk menghindari aksi kejahatan dari pihak lain seperti hacker, C# memiliki segudang fitur untuk menanganinya.

2.14.2 Structured Query Language (SQL)

SQL (Structured Query Language) merupakan sebuah bahasa permintaan database yang tersetruktur yang dibuat sebagai bahasa yang dapat merelasikan

beberapa tabel dalam database maupun merelasikan antar database [10]. Bahasa SQL ditulis langsung dalam sebuah program database sehingga pengguna dapat melihat langsung permintaan yang diinginkan dan sekaligus melihat hasilnya. Ada beberapa program database server yang menggunakan standar query berupa SQL yaitu: Oracle, PostgreSQL, MySQL, MsQL, SQL Server, dan lain-lain. Dalam penelitian ini SQL diimplementasikan pada database server MySQL [10].

2.15 Aplikasi

Aplikasi merupakan perangkat lunak siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari para pengguna aplikasi tersebut. Dalam penelitian ini digunakan aplikasi untuk mendukung penelitian yang dilakukan meliputi Xampp dan Microsoft Visual Studio 2015 [4].

2.15.1 Microsoft Visual Studio 2015

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup compiler, SDK, Integrated Development Environment (IDE), dan dokumentasi yang umumnya berupa MSDN Library. Compiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe [12].

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

2.15.2 Xampp

Xampp merupakan software web server apache yang di dalamnya sudah tersedia database server mysql dan support php programming. Xampp juga mudah digunakan, gratis dan mendukung instalasi Linux dan Windows. Salah satu keuntungan xampp yaitu instalasi yang dilakukan satu kali dan sudah tersedia Apache Web Server, MySql Database Server, PHP Support, FileZilla, dan modul lainnya. Versi Xampp yang digunakna dalam penelitian ini adalah versi 5.6.19-0. Kemudian fitur yang digunakan adalah fitur MySQL dan Apache Web Server [10].