

SKRIPSI
SISTEM PREDIKSI PENUMPANG BUS

Disusun dan diajukan oleh:

ISMAYANTI

D421 16 016



DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

LEMBAR PENGESAHAN SKRIPSI

SISTEM PREDIKSI PENUMPANG BUS

Disusun dan diajukan oleh

ISMAYANTI

D421 16 016

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka penyelesaian studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin

Pada tanggal 7 Juli 2021
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui

Pembimbing Utama,

Pembimbing Pendamping,

Dr. Ir. Ingrid Nurtanio, M.T.
NIP. 19610813 198811 2 001

Dr. Indrabayu, S.T., M.T., M.Bus.Svs.
NIP. 19750716 200212 1 004



Ketua Program Studi,

Dr. Khalil Ahmad Ilham, S.T., M.IT.
NIP. 19731018 199802 1 001

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : ISMAYANTI

NIM : D421 16 016

Program Studi : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul:

SISTEM PREDIKSI PENUMPANG BUS

Adalah karya tulisan saya sendiri, bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 7 Juli 2021

Yang Menyatakan,



ISMAYANTI

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat dan rahmat-Nya maka skripsi ini dapat diselesaikan dengan baik.

Skripsi yang berjudul “*Sistem Prediksi Penumpang Bus*” disusun untuk memenuhi persyaratan kurikulum sarjana strata-1 (S-1) pada Program Studi Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis mengucapkan rasa terima kasih atas semua bantuan yang telah diberikan, baik secara langsung maupun tidak langsung selama penyusunan tugas akhir ini hingga selesai. Secara khusus rasa terima kasih tersebut penulis sampaikan kepada:

1. Tuhan Yang Maha Esa yang melalui berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir ini.
2. Orang tua dan keluarga yang senantiasa memberikan doa, dukungan dan motivasi selama pembuatan, perancangan hingga penyelesaian tugas akhir.
3. Ibu **Dr. Ir. Ingrid Nurtanio, M.T.** selaku pembimbing I dan Bapak **Dr. Indrabayu, ST., M.T., M.Bus,Sys** selaku pembimbing II yang telah memberikan bimbingan dan dorongan dalam penyusunan tugas akhir ini.
4. Bapak **Amil Ahmad Ilham, ST., M.IT., Ph.D** selaku ketua Prodi Teknik Informatika Fakultas Teknik Universitas Hasanuddin.
5. Bapak **Robert** dan **Zainuddin** selaku staf Kantor Prodi Informatika Universitas Hasanuddin.
6. **Teman-teman Prodi Teknik Informatika** terkhusus angkatan 2016 atas segala dukungan dan bantuannya.
7. **Teman-teman Research Group AIMP** atas segala dukungan dan bantuannya dalam penyelesaian tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini belum sempurna, baik dari segi materi maupun dalam penyajiannya, oleh karena itu saran dan kritik yang membangun sangat diharapkan dalam penyempurnaan tugas akhir ini.

Akhirnya, penulis berharap semoga tugas akhir ini dapat memberikan hal yang bermanfaat dan menambah wawasan bagi pembaca dan khususnya bagi penulis juga.

Makassar, Juli 2021

Penulis,

(Ismayanti)

ABSTRAK

Jumlah penumpang Bus sangat berfluktuasi dari waktu ke waktu, saat akhir pekan terutama hari libur nasional, jumlah penumpang dapat membludak sedangkan ketersediaan kursi tidak mencukupi. Di sisi lain, dapat terjadi kelebihan kursi jika penyediaan kursi penumpang dibuat tetap dari waktu ke waktu. Hal tersebut merupakan salah satu faktor yang memengaruhi kualitas pelayanan dalam ketersediaan kursi penumpang, sehingga perusahaan dapat mengefisienkan penyediaan kursi penumpang sesuai permintaan.

Algoritma Jaringan Saraf Tiruan *Backpropagation* digunakan dalam memprediksi jumlah penumpang melalui data historis jumlah penumpang dengan melatih suatu jaringan dengan menggunakan data penumpang sebelumnya. Disini digunakan fungsi pelatihan *Gradient descent with Momentum & Adaptive LR* (traingdx), *Levenberg-Marquardt* (trainlm) dan *Gradient descent* (traingd). Dari hasil penelitian dengan tiga fungsi pelatihan ini diperoleh nilai *MSE* 0.0042332 untuk traingdx, 0.25193 untuk trainlm dan 0.0074848 untuk traingd. Didapatkan fungsi pelatihan *Gradient descent with Momentum & Adaptive LR* (traingdx) menghasilkan prediksi yang lebih baik dilihat pada nilai *MSE* terkecil yang didapatkan dibandingkan fungsi pelatihan lainnya. Hasil prediksi yang didapatkan pada salah satu trayek memiliki nilai MAPE sebesar 5,313255297%, kurang dari 10% maka kemampuan prediksi sangat baik.

Kata Kunci: Penumpang Bus, prediksi, jst *bacpropagation*, *traingdx*, *trainlm*, *traingd*

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	vii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	3
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Defenisi Jaringan Saraf Tiruan.....	7
2.2. Sejarah Jaringan Saraf Tiruan	8
2.3. Perbandingan Jaringan Saraf Tiruan dengan konvensional.....	9
2.4. Komponen Jaringan Saraf Tiruan.....	9
2.5. Arsitektur Jaringan	10
2.6. Fungsi Aktivasi.....	12
2.7. Algoritma <i>Backpropagation</i>	13
2.8. Aplikasi <i>Backpropagation</i> dalam Peramalan	19
2.9. Perhitungan Error	21
2.10. Peramalan (<i>forecasting</i>)	24
2.11. Karakteristik Peramalan	26
BAB III METODOLOGI PENELITIAN.....	28
3.1. Tahapan Penelitian	28
3.2. Waktu dan Lokasi Penelitian.....	29
3.3. Instrumen Penelitian.....	29
3.4. Akuisisi Data	30
3.5. Perancangan dan Implementasi Sistem	31
3.5.1. Tahapan <i>Training</i> / Pelatihan.....	33

3.5.2. Tahapan <i>Testing</i> / Pengujian	39
3.5.2. Tahapan Prediksi.....	41
3.6. Data Masukan.....	42
3.7. Analisa Data	43
3.8. Analisa Kinerja Sistem	44
BAB IV HASIL DAN PEMBAHASAN	45
4.1. Hasil Penelitian.....	45
4.2. Pembahasan	57
BAB V PENUTUP.....	58
5.1. Kesimpulan.....	58
5.2. Saran.....	59
DAFTAR PUSTAKA	60
LAMPIRAN.....	62

DAFTAR TABEL

Tabel 2. 1 <i>Range Nilai Mape</i>	30
Tabel 3. 1 Data Penumpang Harian	30
Tabel 3. 2 Analisis Variasi Arsitektur Jaringan menggunakan <i>trainlm</i>	31
Tabel 3. 3 Analisis Variasi Arsitektur Jaringan menggunakan <i>traindx</i>	32
Tabel 3. 4 Analisis Variasi Arsitektur Jaringan menggunakan <i>traind</i>	32
Tabel 4. 1 Inisialisasi Parameter <i>Neural Network Backpropagation</i>	45
Tabel 4. 2 Hasil Pelatihan Model Jaringan	46
Tabel 4. 3 Hasil Pengujian Model Jaringan	48
Tabel 4. 4 Perbandingan Data Actual dengan Hasil Prediksi Bulan Januari	50
Tabel 4. 5 Hasil Pelatihan Model Jaringan Fungsi Baru	51
Tabel 4. 6 Hasil Pengujian Model Jaringan Fungsi Baru	51
Tabel 4. 7 Perbandingan Data Actual dengan Hasil Prediksi Bulan Januari.....	55
Tabel 4. 8 Perbandingan Data Actual dengan Hasil Prediksi Bulan Februari	56

DAFTAR GAMBAR

Gambar 2. 1 Jaringan Neuron Single Layer	10
Gambar 2.2 Arsitektur Layar Tunggal	11
Gambar 2. 3 Multi <i>Layer</i>	11
Gambar 2. 4 Grafik Fungsi <i>Sigmoid</i> Biner	11
Gambar 3. 1 Tahapan Penelitian	28
Gambar 3. 2 Flowchart Pelatihan	33
Gambar 3. 3 Flowchart Pengujian	39
Gambar 3. 4 Flowchart Prediksi	41
Gambar 4. 1 Plot Hasil Pelatihan.....	52
Gambar 4. 2 Plot Hasil Pengujian	54

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring berjalannya waktu, jumlah penumpang Bus Perum Damri Makassar mengalami naik turun menurut data dalam 4 (empat) tahun terakhir. Adanya perayaan hari-hari besar ataupun libur nasional menjadi faktor naik turunnya jumlah penumpang. Hal tersebut dapat memengaruhi kualitas pelayanan dalam ketersediaan kursi penumpang jika tidak mampu melayani ketersediaan kursi penumpang dengan baik disaat jumlah penumpang meningkat melebihi batas dalam 1 bus. Melakukan langkah yang kongkret dalam mengatasi naik turunnya penumpang susah untuk dilakukan jika suatu perusahaan bus tidak dapat melakukan prediksi jumlah penumpang. Apalagi jika suatu perusahaan memiliki jumlah bus yang tersedia terbatas dan untuk menyiapkan keberangkatan sebuah bus juga dibutuhkan waktu dan persiapan. Oleh karena itu, untuk menyelesaikan kasus ini diperlukan sebuah sistem yang mampu meramalkan atau memprediksi jumlah penumpang di masa yang akan datang. Sehingga Perum Bus Damri Makassar dapat menyiapkan bus tambahan ataupun melakukan pengalihan bus dengan tepat dan baik disaat jumlah penumpang bus meningkat dengan menyesuaikan ketersediaan dari jumlah bus yang ada.

Jaringan syaraf tiruan (Artificial Neural Network) atau disingkat dengan JST adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. Jaringan syaraf tiruan merupakan sistem adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan

informasi eksternal maupun internal yang mengalir melalui jaringan syaraf tiruan. Secara sederhana, JST adalah sebuah alat pemodelan data statistik non-linear. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara *input* dan *output* untuk menemukan pola-pola data. Memprediksi jumlah penumpang Perum Bus Damri Makassar sangat diperlukan hal ini dikarenakan lonjakan penumpang yang mungkin terjadi, agar mencegah banyaknya jumlah penumpang yang tidak terangkut oleh armada dengan cara menyediakan jumlah penumpang yang telah diketahui sebelumnya melalui sistem peramalan yang telah dirancang menggunakan metode *backpropagation*.

Menurut Atur Wulan Hutajulu dalam penelitian berjudul *Jaringan Syaraf Tiruan untuk Memprediksi Jumlah Penumpang Menggunakan Metode Backpropagation* pada tahun 2018 mengatakan bahwa *backpropagation* merupakan salah satu metode statistik yang dipergunakan dalam produksi untuk melakukan peramalan atau prediksi tentang karakteristik kualitas atau kuantitas. Dalam penelitian tersebut menghasilkan kesimpulan bahwa Jaringan Syaraf Tiruan dengan metode *Backpropagation* mampu memprediksi jumlah penumpang pada C.V."MAKMUR" Medan. Sehingga dalam penelitian ini menggunakan metode *backpropagation* yang dapat digunakan untuk memprediksi jumlah penumpang bus pada Perum Damri Makassar kedepannya dengan melihat data-data yang ada sebelumnya. Prediksi merupakan sumber informasi yang dapat digunakan oleh perusahaan untuk mempersiapkan diri dalam menentukan strategi kedepan yang lebih baik.

1.2. Rumusan Masalah

Dari paparan latar belakang yang telah dijelaskan diatas, maka bisa diperoleh suatu rumusan masalah yang terbagi menjadi beberapa hal sebagai berikut :

1. Bagaimana menciptakan suatu sistem yang dapat memprediksi jumlah penumpang bus menggunakan model jaringan syaraf tiruan *backpropagation*?
2. Bagaimana tingkat akurasi sistem dalam memprediksi jumlah penumpang berdasarkan data-data yang ada?
3. Bagaimana menentukan arsitektur jaringan yang terbaik untuk jaringan syaraf tiruan dalam memprediksi jumlah penumpang bus?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini antara lain:

1. Merancang suatu sistem yang dapat memprediksi jumlah penumpang bus perharinya dengan menggunakan jaringan syaraf tiruan *backpropagation*.
2. Mengetahui tingkat akurasi sistem dalam memprediksi jumlah penumpang bus.
3. Menentukan arsitektur jaringan yang optimal untuk jaringan syaraf tiruan dalam memprediksi jumlah penumpang bus.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberi manfaat sebagai berikut:

- a. Bagi peneliti

1. Peneliti dapat membuat suatu sistem yang dapat memprediksi jumlah penumpang perharinya dengan jaringan saraf tiruan *backpropagation*.
2. Peneliti dapat menentukan dan mengetahui arsitektur jaringan syaraf tiruan yang paling optimal untuk memprediksi jumlah penumpang bus.
3. Peneliti dapat mengetahui tingkat keakuratan sistem yang digunakan dalam memprediksi jumlah penumpang bus.

b. Bagi Mahasiswa

Penelitian ini dapat memberikan inspirasi dan gagasan baru bagi mahasiswa untuk mengembangkan dan merancang suatu sistem menggunakan jaringan saraf tiruan yang nantinya dapat dimanfaatkan di berbagai bidang ilmu pengetahuan sehingga berguna dalam kehidupan sehari-hari.

c. Bagi instansi terkait

Hasil penelitian ini dapat bermanfaat bagi Perum Damri Makassar dalam memprediksi jumlah penumpang sehingga dapat melakukan langkah-langkah konkret kedepannya.

1.5 Batasan Masalah

Batasan masalah yang ditentukan dalam sistem ini adalah:

1. Data diperoleh dari kantor Perum Bus Damri Makassar berupa file excel dengan format xls.
2. Data yang akan digunakan adalah data penumpang bus selama 4 (empat) tahun, yaitu tahun 2016, 2017, 2018, dan 2019.

3. Menggunakan algoritma JST *Backpropagation* dalam memprediksi jumlah penumpang
4. Menggunakan software Matlab 2017a untuk membangun sistem prediksi

2 Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan ini, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

BAB I PENDAHULUAN

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori mendasar dan berhubungan dengan penelitian dan metode-metode yang digunakan dalam penelitian.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang perencanaan dan proses penerapan metode-metode dalam pengolahan data dari preprocessing hingga menghasilkan prediksi.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil penelitian dan pembahasan terkait pengolahan data yang telah dilakukan disertai dengan tabel hasil penelitian.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran untuk pengembangan sistem yang lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Defenisi Jaringan Saraf Tiruan

Jaringan syaraf tiruan adalah sistem pemroses informasi dengan karakteristik dan performa yang mendekati syaraf biologis. Selain memproses, jaringan syaraf tiruan juga memiliki kemampuan menyimpan informasi seperti definisi oleh Haykin bahwa jaringan syaraf tiruan merupakan pemroses sederhana yang berjumlah banyak dan bekerja secara paralel dan terdistribusi, yang memiliki kemampuan menyimpan pengetahuan dan memberikan saat dibutuhkan yang terdiri dari pengetahuan yang dimiliki sebagai hasil proses pembelajaran dan koneksi antar neuron yang berfungsi menyimpan pengetahuan itu.

Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan artinya adalah digunakan karena jaringan syaraf diimplementasikan menggunakan program komputer yang dapat menyelesaikan proses perhitungan selama proses pembelajaran. Pada dasarnya jaringan syaraf tiruan mempunyai banyak tipe, tetapi semua tipe dari jaringan syaraf tiruan memiliki komponen-komponen yang sama.

Pada dasarnya struktur jaringan syaraf tiruan ditentukan oleh tiga hal dasar sebagai berikut (Monika, dkk. 2016):

1. Arsitekur, dimana terdapat pola neuron yang saling berhubungan yang terdiri dari: lapisan *input*, lapisan *hidden* dan lapisan *output*.

2. Penentuan bobot penghubung pada masing-masing lapisan *input*, *hidden* dan *output* yang disebut sebagai (*training/learning/algorithm*).
3. Fungsi aktivasi, dimana fungsi yang paling umum digunakan adalah *sigmoid*.

2.2 `Sejarah Jaringan Saraf Tiruan

Jaringan syaraf tiruan sederhana pertama kali diperkenalkan oleh McCulloch dan Pitts pada Tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi threshold. Tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan yang disebut Perceptron. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya. Widrow dan Hoff (1960) mengembangkan perceptron dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan delta (atau sering disebut kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot perceptron apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan. Selain itu, beberapa model jaringan syaraf tiruan lain juga dikembangkan oleh Kohonen (1972), Hopfield (1982), dan lain-lain. Pengembangan yang ramai dibicarakan sejak tahun 1990an adalah aplikasi model-model jaringan syaraf tiruan untuk menyelesaikan berbagai masalah di dunia nyata (Alvian. 2017).

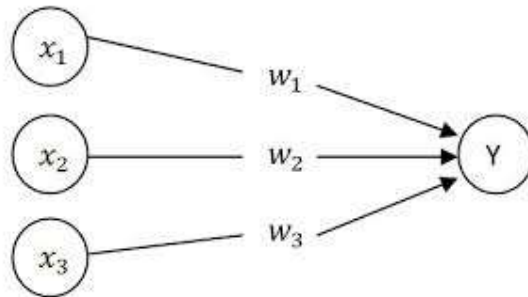
2.3 Perbandingan Jaringan Saraf Tiruan dengan Konvensional

Jaringan Syaraf Tiruan memiliki pendekatan yang berbeda untuk memecahkan masalah bila dibandingkan dengan sebuah komputer konvensional. Umumnya komputer konvensional menggunakan pendekatan algoritma (komputer konvensional menjalankan sekumpulan perintah untuk memecahkan masalah). Jika suatu perintah tidak diketahui oleh komputer konvensional maka komputer konvensional tidak dapat memecahkan masalah yang ada. Sangat penting mengetahui bagaimana memecahkan suatu masalah pada komputer konvensional dimana komputer konvensional akan sangat bermanfaat jika dapat melakukan sesuatu dimana pengguna belum mengetahui bagaimana melakukannya. Jaringan syaraf tiruan dan suatu algoritma komputer konvensional tidak saling bersaing namun saling melengkapi satu sama lain. Pada suatu kegiatan yang besar, sistem yang diperlukan biasanya menggunakan kombinasi antara keduanya (biasanya sebuah komputer konvensional digunakan untuk mengontrol Jaringan Syaraf Tiruan untuk menghasilkan efisiensi yang maksimal. Jaringan Syaraf Tiruan tidak memberikan suatu keajaiban tetapi jika digunakan secara tepat akan menghasilkan sesuatu hasil yang luarbiasa (Herri dan Rivai. 2008)

2.4 Komponen Jaringan Saraf

Ada beberapa tipe jaringan saraf tiruan, namun demikian hampir semuanya memiliki komponen-komponen yang sama, seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan menuju ke neuron-neuron yang lain. Pada jaringan syaraf,

hubungan ini disebut dengan bobot, informasi tersebut disimpan pada satu nilai tertentu pada bobot tersebut (Alvian. 2017).



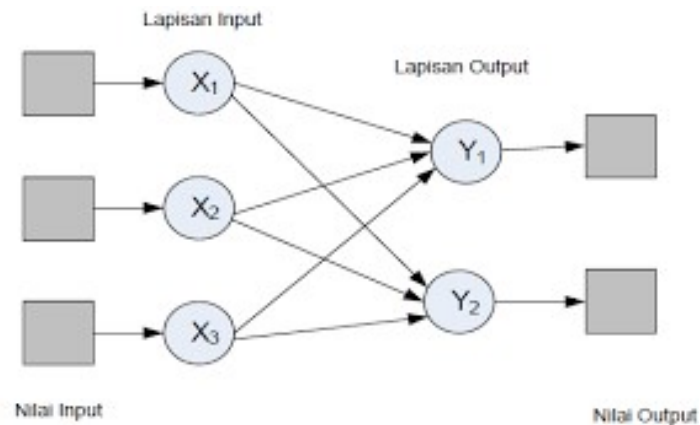
Gambar 2.1 Jaringan *Neuron Single Layer* (Alvian. 2017).

2.5 Arsitektur Jaringan

Jaringan syaraf tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur jaringan syaraf tiruan tersebut, antara lain :

a. Jaringan Layar Tunggal (*single layer network*)

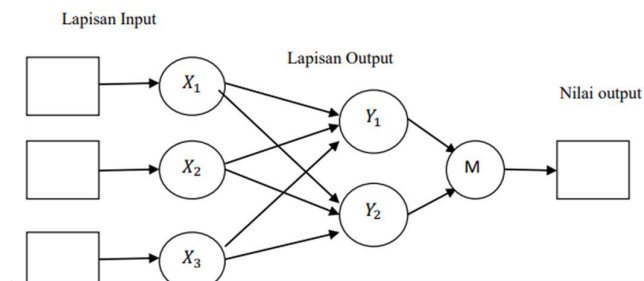
Pada jaringan ini, sekumpulan masukan neuron dihubungkan langsung dengan sekumpulan keluarannya. Sinyal mengalir searah dari layar (lapisan) masukan sampai layar (lapisan) keluaran. Setiap simpul dihubungkan dengan simpul lainnya yang berada di atasnya dan dibawahnya (Maria dan Toni. 2012).



Gambar 2.2 Arsitektur Layar Tunggal (Maria dan Toni. 2012)

b. Jaringan layar jamak (*Multi Layer Network*)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yakni *layer input*, *layer output*, dan *layer* tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini yaitu : *MADALINE*, *backpropagation*, *neocognitron* (Maria dan Toni. 2012).



Gambar 2.3 Multi Layer (Maria dan Toni. 2012).

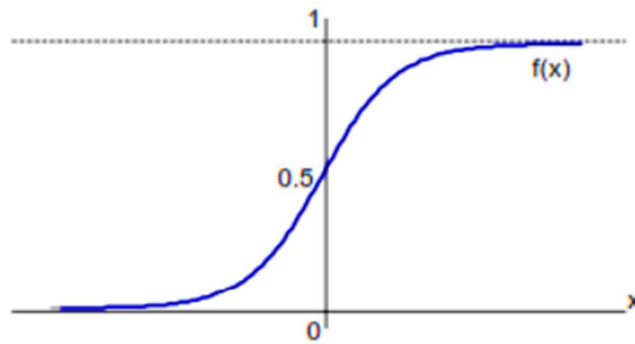
2.6 Fungsi Aktivasi

Fungsi aktivasi *sigmoid* biner ini digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi *sigmoid* memiliki nilai pada *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan saraf yang nilai *output*nya 0 atau 1. Fungsi *sigmoid* biner dirumuskan sebagai berikut (Dahriani. 2014) :

$$y = f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

Fungsi Step dirumuskan sebagai:

$$y'(x) = f(x)(1 - f(x)) \quad (2.2)$$



Gambar 2.4 Grafik Fungsi *Sigmoid* Biner (Alvian. 2017).

Fungsi *sigmoid* memiliki nilai maksimum = 1. Maka untuk pola yang targetnya > 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti fungsi *sigmoid* yang dipakai. Alternatif lain adalah menggunakan fungsi aktivasi *sigmoid* hanya pada

layer yang bukan layer keluaran . Pada layer keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas : $f(x) = x$ (Alvian. 2017).

2.7 Algoritma Backpropagation

Backpropagation adalah algoritma pembelajaran untuk memperkecil tingkat *error* dengan cara menyesuaikan bobotnya berdasarkan perbedaan *output* dan target yang diinginkan.

Pelatihan *backpropagation* meliputi tiga fase yaitu sebagai berikut :

Fase I : Propagasi Maju

Selama propagasi maju, sinyal masukan ($= x_i$) dipropagasikan ke lapis tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit lapis tersembunyi ($= z_j$) tersebut selanjutnya dipropagasikan maju lagi ke lapis tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ($= y_k$).

Berikutnya, keluaran jaringan ($= y_k$) dibandingkan dengan target yang harus dicapai ($= t_k$). Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap garis dalam jaringan akan dimodifikasikan untuk mengurangi kesalahan yang terjadi (Jumarwanto. 2009).

Fase II : Propagasi Mundur

Berdasarkan kesalahan $t_k - y_k$, dihitung faktor δ_k ($k=1, 2, \dots, m$) yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k . δ_k juga dipakai untuk mengubah bobot garis yang menghubungkan langsung dengan unit keluaran. Dengan cara yang sama, dihitung di setiap unit di lapis tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di lapis di bawahnya. Demikian seterusnya hingga faktor δ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung (Jumarwanto. 2009).

Fase III : Perubahan Bobot

Setelah semua faktor δ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor δ neuron di lapis atasnya. Sebagai contoh, perubahan bobot garis yang menuju ke lapis keluaran didasarkan atas dasar δ_k yang ada di unit keluaran. Ketiga fase tersebut diulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum literasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Algoritma pelatihan *backpropagation* atau yang sering disebut dengan propagasi balik termasuk metode pelatihan supervisi dimana terdapat pasangan *input-target* serta didesain untuk operasi jaringan *feed forward* multi-lapis. Dalam

penelitian ini, notasi yang akan digunakan dalam algoritma pelatihan *backpropagation* adalah sebagai berikut.

n : Banyak variabel prediktor.

m : Banyak kategori variabel respon.

x : Vektor *input*, $x = (x_1, \dots, x_i, \dots, x_n)$.

t : Vektor target, $t = (t_1, \dots, t_k, \dots, t_m)$.

δ : Informasi galat

δ_k : Informasi galat untuk unit *output* Y_k .

δ_j : Informasi galat untuk unit tersembunyi Z_j ,

a : *Learning rate* (tingkat pembelajaran).

X_i : Unit *input* ke- i .

v_{0j} : Bias untuk unit tersembunyi ke- j .

v_{ij} : Bobot dari unit *input* ke- i ke unit tersembunyi ke- j .

Z_j : Unit tersembunyi ke- j .

w_{0k} : Bias untuk unit *output* ke- k .

w_{jk} : Bobot dari unit tersembunyi ke- j ke unit *output* ke- k .

Y_k : Unit *output* ke- k .

$t_k - y_k$: Selisih kesalahan yang terjadi

Berikut merupakan algoritma pelatihan *backpropagation*.

0. Inisialisasi bobot dan parameter

1. Ketika kondisi berhenti tidak terpenuhi, dilakukan langkah 2 sampai langkah

9.

2. Untuk setiap pasang data, lakukan langkah 3 sampai langkah 8.

3. Setiap unit *input* ($X_i, i = 1, \dots, n$) menerima sinyal *input* x_i .

4. Setiap unit tersembunyi ($Z_j, j = 1, \dots, p$) menghitung nilai

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \text{ dan } z_j = f(z_{in_j}). \quad (2.3)$$

5. Setiap unit *output* ($Y_k, k = 1, \dots, m$) menghitung nilai

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \text{ dan } y_k = f(y_{in_k}). \quad (2.4)$$

6. Setiap unit *output* ($Y_k, k = 1, \dots, m$) menghitung nilai

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \text{ kemudian}$$

$$\Delta w_{jk} = \alpha \delta_k z_j \text{ dan } \Delta w_{0k} = \alpha \delta_k \quad (2.5)$$

7. Setiap unit tersembunyi ($Z_j, j = 1, \dots, p$) menghitung nilai

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \text{ dan } \delta_j = \delta_{in_j} f'(z_{in_j})$$

kemudian menghitung

$$\Delta v_{ij} = \alpha \delta_j x_i \text{ dan } \Delta v_{0j} = \alpha \delta_j \quad (2.6)$$

8. Setiap unit *output* ($y_k, k = 1, \dots, m$) mengupdate bias bobot dengan cara

$$w_{0k}^{(l+1)} = w_{0k}^l + \Delta w_{0k} \text{ dan } v_{ij}^{(l+1)} = v_{ij}^l + \Delta v_{ij} \quad (2.7)$$

9. Uji kondisi berhenti.

Dalam penelitian ini, fungsi aktivasi yang digunakan dalam unit tersembunyi dan *output* adalah fungsi aktivasi *sigmoid* biner. Berikut persamaan untuk fungsi *sigmoid* biner dan turunannya.

$$Z_j = f(z_{in_j}) = \frac{1}{1 + e^{-z_{in_j}}} \quad (2.8)$$

Kriteria yang digunakan untuk menghentikan suatu pelatihan terdapat dua macam cara yaitu menggunakan tingkat galat yang diinginkan dan menentukan jumlah maksimum *epoch* yang ingin dilakukan. Tinjauan pustaka mengenai variasi-variasi *backpropagation* yang akan digunakan adalah sebagai berikut.

1. *Gradient descent*

Algoritma *gradient descent* mencari derivasi dari E terhadap vektor bobot w untuk melakukan update bobot, dimana E merupakan energi dari perbedaan antara *output* dengan target yang diinginkan pada titik data ke- l dan nilai E bergantung pada nilai bobot yang diperoleh. Setelah turunan parsial dari E terhadap w_{jk} dijabarkan dan disederhanakan, maka diperoleh turunan parsial dari E .

$$\frac{\partial E}{\partial w_{jk}} = -(t_k - y_k) f'(y_{in_k}) z_j = -\delta_k z_j \quad (2.9)$$

sehingga nilai koreksi bobot w_{jk} dapat diringkas menjadi seperti berikut.

$$\begin{aligned} \Delta w_{jk} &= -\alpha \frac{\partial E}{\partial w_{jk}}, \\ \Delta w_{jk} &= \alpha \delta_k z_j. \end{aligned} \quad (2.10)$$

2. *Gradient descent* dengan Momentum dan *Adaptive Learning Rate*

Variasi *backpropagation* ini merupakan kombinasi dari dua variasi sebelumnya. Selain terdapat penambahan momentum, dalam pelatihan ini juga menerapkan *adaptive learning rate*. Oleh karena itu, perubahan bobot dan bias dengan adanya momentum dan *adaptive learning rate* adalah sebagai berikut.

$$w_{jk}^{l+1} = w_{jk}^l + a^{(l+1)}\delta_k z_j + \rho[w_{jk}^{(l)} - w_{jk}^{(l-1)}],$$

Dan

$$v_{ij}^{l+1} = v_{ij}^{(l)} + a^{(l+1)}\delta_j x_i + \rho[v_{ij}^{(l)} - v_{ij}^{(l-1)}] \quad (2.11)$$

3. Levenberg-Marquardt

Metode ini merupakan kombinasi dari dua metode yaitu metode Newton dan *gradient descent* dimana metode Newton cepat konvergen namun kurang stabil sedangkan *gradient descent* pasti konvergen namun sangat lambat. Algoritma pelatihan ini menggunakan dasar metode Newton namun tidak menggunakan matriks Hessian untuk orde kedua. Matriks Hessian didekati dengan $\mathbf{H} = \mathbf{J}^T \mathbf{J}$ dan vektor gradient dengan $\mathbf{g} = \mathbf{J}^T \mathbf{e}$. Matriks \mathbf{J} adalah matriks Jacobian yang terdiri dari turunan pertama galat berdasarkan masing-masing bobot. Aturan *update* bobot untuk pelatihan Levenberg-Marquardt *backpropagation* adalah sebagai berikut.

$$\mathbf{w}^{(l+1)} = \mathbf{w}^{(l)} - [\mathbf{H} + \lambda \mathbf{I}]^{-1} \mathbf{g} \quad (2.12)$$

Ket:

\mathbf{H} : Matriks Hessian

\mathbf{J} : Matriks Jacobian

λ : Nilai Parameter

g : Vektor gradient

Matriks I adalah matriks identitas yang ukurannya bersesuaian dengan matriks H . Nilai parameter λ sangat penting dalam pelatihan Levenberg-Marquardt. Apabila nilai λ bernilai nol, algoritma pelatihan ini akan menjadi seperti algoritma dalam metode Newton. Sedangkan apabila λ bernilai besar, pelatihan ini akan menjadi seperti algoritma *gradient descent* (Jumarwanto. 2009).

2.8 Aplikasi *Backpropagation* dalam Peramalan

Salah satu bidang dimana *backpropagation* dapat diaplikasikan dengan baik adalah bidang peramalan (*forecasting*). Peramalan yang sering kita dengar adalah peramalan besarnya penjualan, nilai tukar valuta asing, prediksi besarnya aliran air sungai, dll. Sebagai contoh, dalam penjualan barang, diketahui *record* data penjualan suatu produk pada beberapa bulan/tahun terakhir. Masalahnya adalah memperkirakan berapa perkiraan produk yang terjual dalam bulan/tahun yang akan datang.

Secara umum, masalah peramalan dapat dinyatakan sebagai berikut :
Diketahui sejumlah data runtun waktu (*time series*) x_1 Masalahnya adalah memperkirakan berapa harga x_{n+1}, x_2, \dots, x_n berdasarkan x_1, x_2, \dots, x_n .

Dengan *backpropagation*, *record* data dipakai sebagai data pelatihan untuk mencari bobot yang optimal. Untuk itu kita perlu menetapkan besarnya periode dimana data berfluktuasi. Periode ini kita tentukan secara intuitif. Misalkan pada data besarnya debit air sungai dengan data bulanan, periode data dapat diambil selama satu tahun karena pergantian musim terjadi selama satu.

Jumlah data dalam satu periode ini dipakai sebagai jumlah masukan dalam *backpropagation*. Sebagai targetnya diambil data bulan pertama setelah periode berakhir. Pada data bulanan dengan periode satu tahun, maka masukan *backpropagation* yang dipakai terdiri dari 12 masukan, keluaran adalah 1 unit.

Bagian tersulit adalah menentukan jumlah *layer* (dan unitnya). Tidak ada teori yang dengan pasti dapat dipakai. Tapi secara praktis dicoba jaringan yang kecil terlebih dahulu (misal terdiri dari 1 *layer* tersembunyi dengan beberapa unit saja). Jika gagal (kesalahan tidak turun dalam *epoch* yang besar), maka jaringan diperbesar dengan menambahkan unit tersembunyi atau bahkan menambah *layer* tersembunyi.

Jika ingin menggunakan fungsi aktivasi *sigmoid* (biner), data harus ditransformasikan dulu karena *range* keluaran fungsi aktivasi *sigmoid* adalah [0,1]. Data bisa ditransformasikan ke interval [0,1]. Tapi akan lebih baik jika ditransformasikan ke interval yang lebih kecil, misal pada interval [0.1, 0.9]. Ini mengingat fungsi *sigmoid* merupakan fungsi asimtotik yang nilainya tidak pernah mencapai 0 ataupun 1.

Jika a adalah data minimum dan b adalah data maksimum transformasi linier yang dipakai untuk mentransformasikan data ke interval [0.1 , 0.9] adalah :

$$x' = \frac{0.8(x-a)}{b-a} + 0.1, \quad (2.13)$$

keterangan :

x' = Data hasil normalisasi

x = Data asli/data awal

a = Nilai maksimum data asli

b = Nilai minimum data asli

Untuk mendapatkan hasil *real condition* maka data *output* harus melalui proses denormalisasi . Adapun Persamaan denormalisasi untuk mengembalikan data hasil menjadi data *real* adalah sebagai berikut (Alvian. 2017) :

$$x = \frac{(x' - 0.1)(a - b)}{0.8} \quad (2.14)$$

2.9 Perhitungan *Error*

2.9.1 Nilai MSE

Nilai *Mean Square Error* (MSE) pada satu siklus pelatihan adalah nilai kesalahan (*error* (e) = nilai keluaran – nilai masukan) rata-rata dari seluruh *record* (*tupple*) yang dipresentasikan ke jaringan syaraf tiruan dan dirumuskan sebagai :

$$MSE = \frac{\sum e^2}{\text{juml record}} \quad (2.15)$$

Semakin kecil MSE, jaringan syaraf tiruan semakin kecil kesalahannya dalam memprediksi kelas dari *record* yang baru. Maka, pelatihan jaringan syaraf tiruan ditujukan untuk memperkecil MSE dari satu siklus ke siklus berikutnya sampai selisish nilai MSE pada siklus ini dengan siklus sebelumnya lebih kecil atau sama dengan batas minimal yang diberikan (Giri dan Veronica. 2004).

Kondisi penghentian *training* adalah tercapainya nilai MSE di akhir *training*. *Training* dihentikan apabila nilai MSE di akhir *training* lebih kecil atau sama

dengan nilai MSE yang ditetapkan pada awal *training*. Tetapi apabila nilai MSE di akhir *training* lebih besar dari nilai MSE yang ditetapkan di awal *training* maka proses *training* akan dilanjutkan ke tahap propagasi balik (Arman, dkk. 2013).

Proses penelitian secara umum dilakukan dengan bantuan *tool Matlab*. *Tool* ini merupakan sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh The Mathworks, Matlab memungkinkan manipulasi matriks, melakukan fungsi plot dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan penantarmukaan dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah *toolbox* yang menggunakan mesin simbol MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Banyak model jaringan syaraf tiruan yang menggunakan manipulasi matriks atau vektor dalam iterasinya. Matlab menyediakan fungsi-fungsi khusus untuk menyelesaikan model jaringan syaraf tiruan. Penggunaan hanya memasukkan vektor masukan, target, model, dan parameter yang diinginkan (laju pemahaman, *threshold*, bias, dll).

Di dalam Matlab *script* di simpan dalam bentuk M-File. Pemrograman M-File memberikan kontrol lebih banyak dibandingkan *command line*. Dengan M-File dapat dilakukan percabangan, perulangan, dll. Struktur M-File pada Matlab hampir sama seperti bahasa C yang membagi program dalam blok program berupa fungsi-fungsi. Setiap fungsi dapat memanggil fungsi yang lainnya (Monica, dkk. 2016).

2.9.2 Nilai Mape (*Means Absolute Percentage Error*)

Prediksi tidak selalu tepat karena teknik yang digunakan belum tentu sesuai dengan sifat datanya. Oleh karena itu perlu diadakan presentase kebenaran prediksi sehingga dapat diketahui sesuai atau tidaknya teknik prediksi yang digunakan. Pada prinsipnya, perhitungan presentase kebenaran prediksi dilakukan dengan membandingkan hasil prediksi dengan kenyataan yang terjadi. Perhitungan besarnya *error* peramalan dihitung dengan menggunakan MAPE yang merupakan nilai tengah presentase kesalahan absolut dari suatu prediksi dengan persamaan (Mustaziri. 2012).

Perhitungan yang digunakan untuk menghitung rata-rata persentase kesalahan mutlak, dengan rumus :

$$MAPE = \sum (| \text{Aktual} - \text{Forecast} | / \text{Aktual}) * 100 / n \quad (2.16)$$

Dari rumus, dapat diartikan bahwa $\sum (| \text{Aktual} - \text{Forecast} | / \text{Aktual})$ merupakan hasil pengurangan antara nilai aktual dan *forecast* yang telah di absolut-kan, kemudian dibagi dengan nilai aktual per periode masing-masing, kemudian dilakukan penjumlahan terhadap hasil-hasil tersebut. Dan *n* merupakan jumlah periode yang digunakan untuk perhitungan. Semakin rendah nilai MAPE, kemampuan dari model peramalan yang digunakan dapat dikatakan baik, dan untuk MAPE terdapat *range* nilai yang dapat dijadikan bahan pengukuran mengenai kemampuan dari suatu model peramalan, *range* nilai tersebut dapat dilihat pada tabel 2.1 (M. Azman, 2019).

Tabel 2.1 *Range* nilai MAPE

Range MAPE	Arti
< 10 %	Kemampuan Model Peramalan Sangat Baik
10 - 20 %	Kemampuan Model Peramalan Baik
20 - 50 %	Kemampuan Model Peramalan Layak
> 50 %	Kemampuan Model Peramalan Buruk

2.10 Peramalan (Forecasting)

Peramalan (*forecasting*) merupakan alat bantu yang penting dalam perencanaan yang efektif dan efisien khususnya dalam bidang ekonomi. Dalam organisasi modern mengetahui keadaan yang akan datang tidak saja penting untuk melihat yang baik atau buruk tetapi juga bertujuan untuk melakukan persiapan peramalan. Peramalan adalah prediksi, proyeksi atau estimasi tingkat kejadian yang tidak pasti di masa yang akan datang. Ketepatan secara mutlak dalam memprediksi peristiwa dan tingkat kegiatan yang akan datang adalah tidak mungkin dicapai, oleh karena itu ketika perusahaan tidak dapat melihat kejadian yang akan datang secara pasti, diperlukan waktu dan tenaga yang besar agar mereka dapat memiliki kekuatan untuk menarik kesimpulan terhadap kejadian yang akan datang. Situasi peramalan sangat beragam dalam horizon waktu peramalan, faktor yang menentukan hasil sebenarnya, tipe pola dan berbagai aspek lainnya. Untuk menghadapi penggunaan yang luas seperti itu, beberapa teknik telah dikembangkan. Peramalan pada umumnya dapat dibedakan dari berbagai segi tergantung dalam cara melihatnya. Dilihat dari jangka waktu ramalan yang disusun, peramalan dapat dibedakan atas dua macam, yaitu:

1. Peramalan jangka panjang, yaitu peramalan yang dilakukan untuk penyusunan hasil ramalan yang jangka waktunya lebih dari satu setengah tahun atau tiga semester. Lebih tegasnya peramalan jangka panjang ini berorientasi pada dasar atau perencanaan.
2. Peramalan jangka pendek, yaitu peramalan yang dilakukan untuk penyusunan hasil ramalan yang dilakukan kurang dari satu setengah tahun atau tiga semester.

Apabila dilihat dari sifat penyusunannya, maka peramalan dapat dibedakan menjadi dua macam, yaitu:

1. Peramalan subjektif, yaitu peramalan yang didasarkan atas perasaan atau intuisi dari orang yang menyusunnya. Dalam hal ini pandangan atau ketajaman pikiran orang yang menyusunnya sangat menentukan baik tidaknya hasil peramalan.
2. Peramalan objektif, yaitu peramalan yang didasarkan atas data yang relevan pada masa lalu dengan menggunakan teknik-teknik dan metode-metode dalam penganalisaan data tersebut.

Dilihat dari sifat ramalan yang telah disusun, maka peramalan dapat dibedakan atas dua macam, yaitu:

1. Peramalan kualitatif atau teknologis, yaitu peramalan yang didasarkan atas data kualitatif masa lalu. Hasil peramalan yang ada tergantung pada orang yang menyusunnya, karena peramalan tersebut sangat ditentukan oleh

pemikiran yang bersifat intuisi, *judgement* (pendapat) dan pengetahuan serta pengalaman dari penyusunnya.

2. Peramalan kuantitatif, yaitu peramalan yang didasarkan atas data kuantitatif pada masa lalu. Hasil peramalan yang dibuat tergantung pada metode yang digunakan dalam peramalan tersebut. Metode yang baik adalah metode yang memberikan nilai-nilai perbedaan atau penyimpangan yang mungkin.

Peramalan kuantitatif hanya dapat digunakan apabila terdapat tiga kondisi sebagai berikut:

1. Informasi tentang keadaan masa lalu.
2. Informasi tersebut dapat dikuantifikasikan dalam bentuk data numerik.
3. Dapat diasumsikan bahwa beberapa aspek pola masa lalu akan terus berkelanjutan pada masa yang akan datang.

2.11 Karakteristik Peramalan

Yang baik karakteristik dari peramalan yang baik harus memenuhi beberapa kriteria yaitu dari hal-hal sebagai berikut:

1. Ketelitian/ Keakuratan

Tujuan utama peramalan adalah menghasilkan prediksi yang akurat. Peramalan yang terlalu rendah mengakibatkan kekurangan persediaan (*inventory*). Peramalan yang terlalu tinggi akan menyebabkan *inventory* yang berlebihan dan biaya operasi tambahan.

2. Biaya

Biaya untuk mengembangkan model peramalan dan melakukan peramalanakan menjadi signifikan jika jumlah produk dan data lainnya semakin besar. Mengusahakan melakukan peramalan jangan sampai menimbulkan ongkos yang terlalu besar ataupun terlalu kecil. Keakuratan peramalan dapat ditingkatkan dengan mengembangkan model lebih kompleks dengan konsekuensi biaya menjadi lebih mahal. Jadi ada nilai tukar antara biaya dan keakuratan.

3. Responsif

Ramalan harus stabil dan tidak terpengaruhi oleh fluktuasi demand.

4. Sederhana

Keuntungan utama menggunakan peramalan yang sederhana yaitu kemudahan untuk melakukan peramalan. Jika kesulitan terjadi pada metode sederhana, diagnosa dilakukan lebih mudah. Secara umum, lebih baik menggunakan metode paling sederhana yang sesuai dengan kebutuhan peramalan (Makridakis, dkk. 1999).