BAB 2
TINJAUAN PUSTAKA

1.1 Tinjauan Non Statistik

1.1.1 Pengertian Ekspor

Pengertian Ekspor barang pada umumnya adalah kegiatan mengeluarkan atau mengirim barang ke luar negeri, biasanya dalam jumlah besar untuk tujuan perdagangan. Dimana ekspor ini merupakan kegiatan penjualan komoditi yang dimiliki sebuah negara asing dengan mengharap pembayaran dalam valuta asing, serta melakukan komunikasi dengan memakai bahasa asing (Amir,2000: 1). Adapun tujuan ekspor (Sarjiyanto, 2012 :1) yaitu sebagai berikut :

a. Untuk meningkatkan keuntungan perusahaan atau Negara
b. Untuk membuka pasar baru di luar negeri
c. Agar dapat terbiasa dalam bersaing di dalam paar internasional

1.1.2 Ekspor Migas dan Non-Migas

Kegiatan ekspor di Indonesia terbagi dalam dua sektor. Dimana sektor tersebut adalah sektor minyak, gas (Migas) dan Non Migas. Dalam sektor migas, Indonesia memiliki komoditi yaitu minyak mentah dan gas. Minyak dan gas merupakan sumber energi dan bahan bakar utama di dunia. Adapun komoditi yang termasuk dalam sektor migas adalah minyak tanah, hasil minyak, dan gas.

Sektor Non Migas adalah sektor yang meliputi komoditi selain dari minyak dan gas. Adapun komoditi yang termasuk pada sektor non migas adalah sebagai berikut: (Amalina, 2016)

Tabel 0.1 Komoditi Non-Migas

Kelompok Komoditi Sektor Non-Migas	**Contoh Komoditi**
Tambang Non-Migas	Batubara, Emas, Perak, Tembaga, Nikel, Bauksit
Hasil Perkebunan dan Pertanian	Karet, Kopi, Kelapa, Sawit, Cengkeh, The, Lada, Kina, Tembakau, Cokelat, Kacang-kacangan, Buah-buahan
Hasil Hutan	Kayu, Rotan, Mangrove
Hasil Peternakan	Daging Sapi, Daging Ayam
Hasil Perikanan	Ikan tuna, Cakalang, Udang, Bandeng
Hasil Pertambangan	Timah, Alumunium, Batu Bara, Tembaga, Emas
Hasil Industri	Semen, Pupuk, Tekstil, Pakaian Jadi, Besi dan Baja

1.1.3 Provinsi Jawa Timur

Jawa Timur (Hanacaraka (Jawa: Jåwå Wétan)) adalah sebuah provinsi di bagian timur Pulau Jawa, Indonesia. Ibu kotanya terletak di Surabaya. Luas wilayahnya 47.922 km², dan jumlah penduduknya 42.030.633 jiwa (sensus 2015). Jawa Timur memiliki wilayah terluas di antara 6 provinsi di Pulau Jawa, dan memiliki jumlah penduduk terbanyak kedua di Indonesia setelah Jawa Barat. Jawa Timur berbatasan dengan Laut Jawa di utara, Selat Bali di timur, Samudra Hindia di selatan, serta Provinsi Jawa Tengah di barat. Wilayah Jawa Timur juga meliputi Pulau Madura, Pulau Bawean, Pulau Kangean serta sejumlah pulau-pulau kecil di

Laut Jawa (Kepulauan Masalembu), dan Samudera Hindia (Pulau Sempu, dan Nusa Barung).

1.2 Tinjauan Statistik

1.2.1 Analisis Deskriptif

Menurut Hasan (2001) statistik deskriptif atau statistik deduktif adalah bagian dari statistik mempelajari cara pengumpulan data dan penyajian data sehingga mudah dipahami. Statistik deskriptif hanya berhubungan dengan hal menguraikan atau memberikan keterangan-keterangan mengenai suatu data atau keadaan atau fenomena. Dengan kata lain statistik deskriptif berfungsi menerangkan keadaan, gejala, atau persoalan (Nasution, 2017).

1.2.2 Pengertian Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013). Prediksi sama dengan peramalan atau perkiraan. Peramalan bertujuan mendapatkan peramalan *(forecast)* yang bisa meminimumkan kesalahan meramal *(forecast error)* yang biasanya diukur dengan MSE *(Mean Squared Error), MAE (Mean Absolute Error),* dan sebagainya (Subagyo, 2000).

1.2.3 Analisis Time Series

Time series merupakan serangkaian observasi terhadap suatu variabel yang diambil secara berurutan berdasarkan interval waktu yang tetap (Wei, 2006). Data *time series* merupakan jenis data yang terdiri dari satu objek tetapi meliputi beberapa periode waktu misalnya harian, mingguan, bulanan, tahunan, dan lain-lain. Data *time series* dianggap sangat berguna untuk memprediksi kejadian di masa depan. Hal ini diyakini pola yang ada pada masa lalu akan terulang kembali di masa mendatang (Huda, Ridok, & Dewi, 2014).

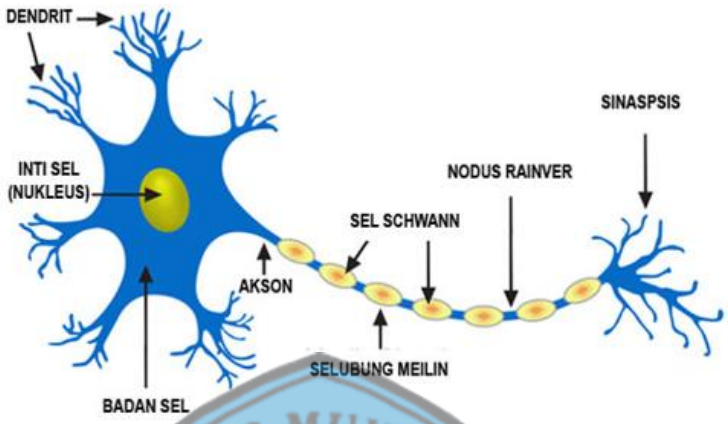
1.2.4 Jaringan Syaraf Tiruan *(Artificial Neural Network)*

Jaringan Saraf Tiruan (JST) atau sering disebut dengan *Artificial Neural Network* (ANN), adalah sistem pemrosesan informasi yang memiliki karakteristik kinerja tertentu yang sama dengan jaringan syaraf biologis. Jaringan syaraf tiruan memiliki karakteristik-karakteristik sebagai berikut: (Fausett, 2017)

1. Pola hubungan antar *neuron* yang disebut arsitektur.
2. Metode penentuan bobot pada hubungan yang disebut pelatihan *(training)* atau pembelajaran *(learning)*.
3. Fungsi aktivasi yang dijalankan masing-masing *neuron* pada input jaringan untuk menentukan output.

Dalam syaraf biologis, setiap sel saraf *(neuron)* akan memiliki satu inti sel yang bertugas untuk melakukan pemrosesan informasi yang akan diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini menjadi masukan bagi

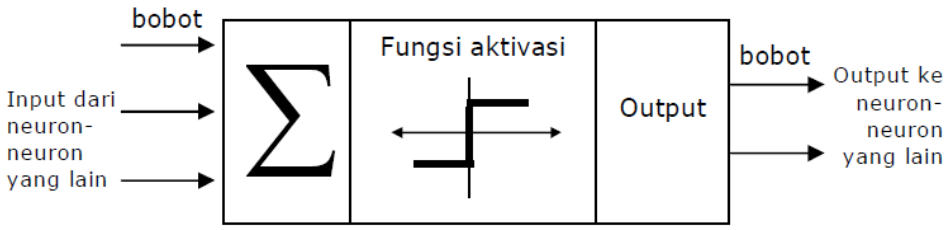
neuron lain (Hadjratie, 2011). Berikut adalah bentuk sederhana dari sebuah *neuron* yang oleh para ahli dianggap sebagai satuan unit pemroses:



Gambar 0.1 Jaringan Syaraf Manusia

Seperti halnya otak manusia, *Neural Network* juga terdiri dari beberapa neuron yang berhubungan untuk mentransformasikan informasi yang di terima melalui sambungan keluarnya. Hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Informasi *(input)* akan dikirim ke neuron dengan bobot kedatangan. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan niai suatu ambang *(threshold)* tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, sehingga neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Neuron-neuron akan dikumpulkan dalam lapisan-lapisan *(layer)* yang disebut dengan lapisan neuron *(neuron layer)* yang saling berhubungan. Infomasi akan dirambatkan mulai dari lapisan input sampai ke lapisan output melalui lapisan

lainnya yang sering dikenal dengan lapisan tersembunyi *(hidden layer)*, dan perambatannya tergantung algoritma pembelajarannya yang dijelaskan melalui gambar dibawah ini: (Kusumadewi dalam Hadjaratie, 2011)



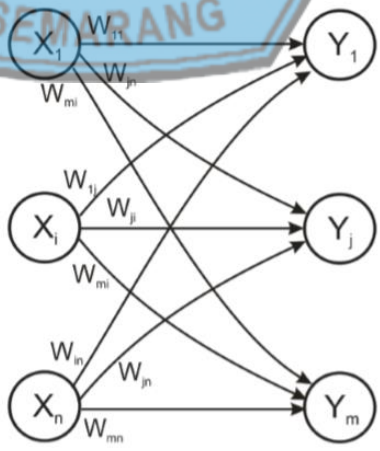
Gambar 0.2 Struktur Syaraf Tiruan

1.2.5 Arsitektur Jaringan

Beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan antara lain:(Siang, 2005)

1.2.5.1 Jaringan Layar Tunggal *(Single Layer Network)*

Dalam jaringan ini, sekumpulan input neuron dihubungkan langsung dengan sekumpulan outputnya.



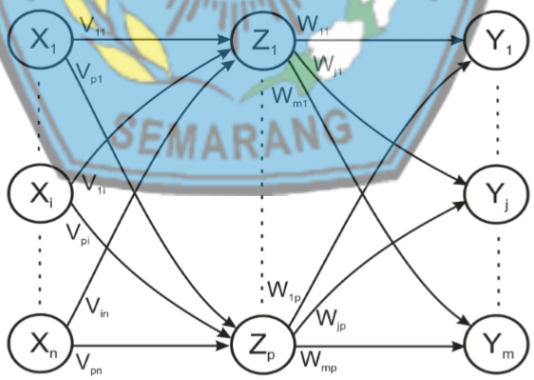
Gambar 0.3 Jaringan Layar Tunggal

Gambar 2.3 menunjukkan arsitektur jaringan dengan n input (X_1, X_i, …, X_n) dan m buah unit ouput (Y_1, Y_j, …, Y_m). Perhatikan bahwa dalam jaringan ini,

semua unit input dihubungkan dengan semua unit ouput, meskipun dengan bobot yang berbeda-beda. Tidak ada unit input yang dihubungkan dengan unit input lainnya. Demikian pula dengan unit output. Besarnya W_{ij} menyatakan bobot hubungan antara unit ke-i dalam input dengan unit ke-j dalam output. Bobot-bobot ini saling independen. Selama proses pelatihan, bobot-bobot tersebut akan dimodifikasi untuk meningkatkan keakuratan hasil. Model semacam ini tepat digunakan untuk pengenalan pola karena kesederhanaanya.

1.2.5.2 Jaringan Layar Jamak *(Multi Layer Network)*

Jaringan layar jamak merupakan perluasan dari layar tunggal. Dalam jaringan ini, selain unit input dan output, ada unit-unit lain (sering disebut layar tersembunyi). Dimungkinkan pula ada beberapa layar tersembunyi. Sama seperti pada unit input dan output, unit-unit dalam satu layar tidak saling berhubungan.

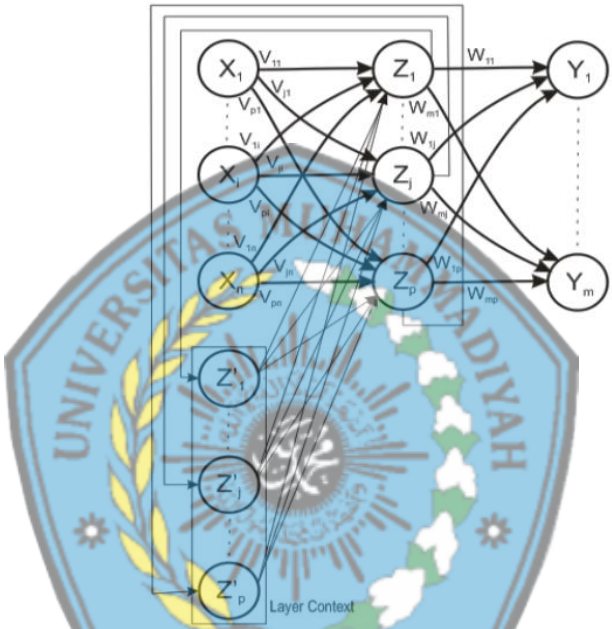


Gambar 0.4 Jaringan Layar Jamak

Gambar 2.4 adalah jaringan dengan n buah unit input (X_1, X_i, …, X_n), sebuah layar tersembunyi yang terdiri dari p buah unit (Z_1, …, Z_p) dan m buah unit output (Y_1, Y_j, …, Y_m). Jaringan layar jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layar tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.

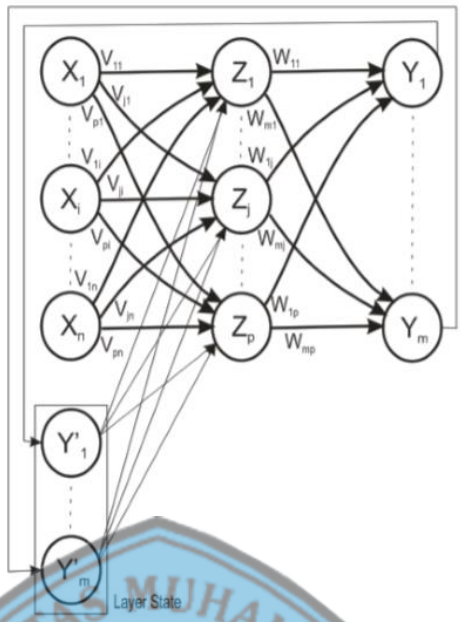
1.2.5.3 Jaringan Berulang *(Recurrent Network)*

Model jaringan *recurrent* mirip dengan jaringan layar tunggal ataupun ganda. Hanya saja, ada neuron output yang memberikan sinyal pada unit input (sering disebut *feedback loop*). Terdapat 2 bentuk jaringan recurrent sederhana yaitu:



Gambar 0.5 Jaringan *Recurrent* Model Elman

Gambar 2.5 adalah jaringan dengan n buah unit input (X_1, X_i, …, X_n), sebuah layar tersembunyi yang terdiri dari p buah unit (Z_1, Z_i, …, Z_p), m buah unit output (Y_1, …, Y_m), dan ‘p buah *layer context* (Z'_1, Z'_j, …, Z'_p) yang merupakan proses *learning* dengan membuat salinan neuron *layer hidden* pada *layer input*.



Gambar 0.6 Jaringan *Recurrent* Model Jordan

Gambar 2.6 adalah jaringan dengan n buah unit input (X_1, X_i, …, X_n), sebuah layar tersembunyi yang terdiri dari p buah unit (Z_1, Z_i, …, Z_p), m buah unit output (Y_1, …, Y_m), dan ‘m buah *layer state* (Y'_1, …, Y'_m) yang merupakan proses *learning* dengan membuat salinan neuron *layer output* pada *layer input*.

1.2.6 Algoritma Pembelajaran

Salah satu bagian terpenting dari konsep JST adalah terjadinya proses pembelajaran. Tujuan utama dari proses pembelajaran adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan syaraf, sehingga diperoleh bobot akhir yang tepat dan sesuai dengan pola data yang dilatih. Cara berlangsungnya pembelajaran atau pelatihan JST dikelompokkan menjadi 3 (tiga), yaitu (Puspitaningrum, 2006):

1. Pembelajaran terawasi *(Supervised Learning)*

   Pada metode ini setiap pola yang diberikan ke dalam JST telah diketahui target outputnya. Selisih antara pola output yang dihasilkan dengan ouput yang dikehendaki (output target) yang disebut *error* digunakan untuk mengoreksi bobot JST sehingga JST mampu menghasilkan output sedekat mungkin dengan pola target yang telah diketahui oleh JST.

2. Pembelajaran yang tak terawasi *(Unsupervised Learning)*

   Pada metode pembelajaran yang tak terawasi tidak memerlukan target output. Pada metode ini tidak ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk pengelompokkan (klasifikasi) pola.

3. Pembelajaran Hibrida *(Hybrid Learning)*

   Merupakan kombinasi dari metode pembelajaran supervised learning dan unsupervised learning. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi.

1.2.7 Fungsi Aktivasi

Setiap neuron memiliki keadaan internal, yang disebut tingkat pengaktifan atau aktivasi, yang merupakan fungsi dari input yang telah diterimanya. Neuron mengirimkan pengaktifannya sebagai sinyal ke beberapa neuron lain. Namun

neuron dapat mengirim hanya satu sinyal pada satu waktu, meskipun sinyal itu disiarkan ke beberapa neuron lain (Fausett, 2017).

Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya). Jika net = $\sum x_i w_i$, maka fungsi aktivasinya adalah f (net) = f ($\sum x_i w_i$). Beberapa fungsi aktivasi yang sering dipakai adalah sebagai berikut: (Siang, 2005)

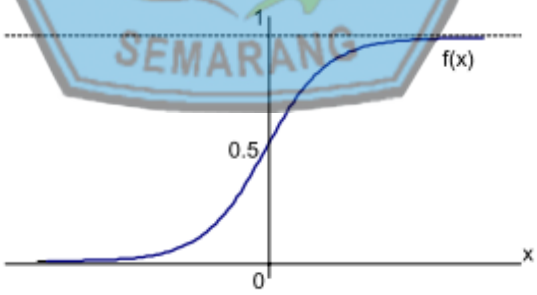
1.2.7.1 Fungsi sigmoid biner

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1.

$$f(x) = \frac{1}{(1+e^{-x})} \quad 0.1$$

Dengan turunannya:

$$f'(x) = f(x)\,(1 - f(x)) \quad 0.2$$



Gambar 0.7 Grafik fungsi sigmoid biner

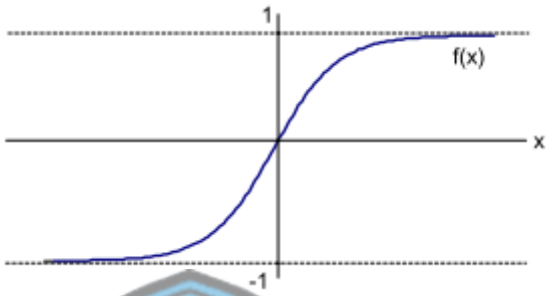
1.2.7.2 Fungsi sigmoid bipolar

Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar yang bentuk fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range (-1, 1).

$$f(x) = \frac{2}{(1 + e^{-x})} - 1$$ 0.3

Dengan turunannya:

$$f'(x) = \frac{(1+f(x))\,(1-f(x))}{2}$$ 0.4

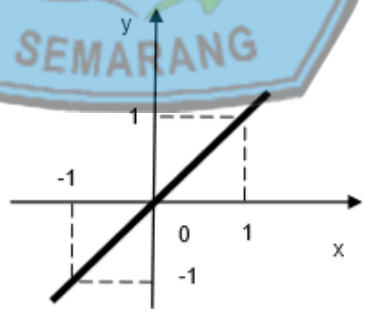


Gambar 0.8 Grafik fungsi sigmoid bipolar

1.2.7.3 Fungsi identitas

Fungsi identitas sering dipakai apabila diinginkan keluaran jaringan berupa sembarang bilangan riil (bukan hanya pada range [0,1] atau [-1,1]).

$$f(x) = x$$ 0.5



Gambar 0.9 Grafik Fungsi Identitas

1.2.8 *Backpropagation Neural Network* (BPNN)

Metode *Backpropagation* adalah sebuah metode sistematik jaringan saraf tiruan yang menggunakan algoritma pembelajaran terawasi *(Supervised Learning)*

dan biasanya digunakan oleh perceptron dengan banyak layar lapisan untuk mengubah bobot-bobot yang ada pada lapisan tersembunyinya (Amalina, 2016).

Backpropagation melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukkan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan (Siang, 2005). Terdapat tiga fase pelatihan dari *backpropagation*, yaitu sebagai berikut: (Amalina, 2016)

a. Fase I: Propagasi Maju *(Feedforward)*

   Di fase ini pola masukan dihitung maju mulai dari input layer hingga output layer menggunakan fungsi aktivasi yang ditentukan.

b. Fase II: Propagasi Mundur

   Selisih antar target yang akan dicapai dengan hasil keluaran merupakan error. Error yang terjadi itu dipropagasi mundur, dimulai dari jaringan yang berhubungan di unit-unit output layer.

c. Fase III : Perubahan Bobot

   Melakukan perubahan pada bobot untuk menurunkan error yang terjadi. Fase ini diulang-ulang terus hingga kondisi penghentian dipenuhi.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Terdapat 2 (dua) alternatif untuk memperbaiki pelatihan backpropagation dengan algoritma pelatihan yang lebih cepat yaitu menggunakan teknik heuristik dan optimasi numeris (Kusumadewi, 2004 dalam Hadjratie, 2011).

1. Perbaikan dengan Teknik Heuristik

   Teknik ini merupakan pengembangan dari suatu analisis kinerja pada algoritma *steepest (gradient) descent standard*. Ada 3 (tiga) algoritma dengan teknik ini, yakni:

   a. *Gradient descent* dengan *Adaptive Learning Rate* (traingda)

      Pada fungsi ini, selama proses pembelajaran *learning rate* akan terus bernilai konstan karena apabila *learning rate* terlalu tinggi maka algoritma menjadi tidak stabil dan jika terlalu rendah algoritma akan sangat lama dalam mencapai kekonvergenan.

   b. *Gradient descent* dengan Momentun dan *Adaptive Learning Rate* (traingdx)

      Fungsi ini akan memperbaiki bobot-bobot berdasarkan *gradient descent* dengan *learning rate* yang bersifat adaptive seperti traingda tapi juga dengan menggunakan momentum. Menurut (Demuth & Beale, 2000) menyatakan bahwa momentum membuat perubahan *weight* pada proses sebelumnya sehingga hal ini dapat menghindarkan jaringan dari tren penurunan error yang salah *(local gradient)*. Semakin besar nilai momentum, membuat jaringan kurang sensitive terhadap keberadaan *local gradient*, yang artinya jaringan tidak mengalami pembelajaran yang baik.

c. *Resilent Backpropagation* (trainrp/Rprop)

Algoritma ini dikembangkan oleh Martin Riedmiller dan Heinrich Braun pada tahun 1992. Algoritma pelatihan ini menggunakan fungsi aktivasi sigmoid yang membawa input dari range yang tak terbatas ke nilai *output* pada range yang terbatas yaitu antara 0 sampai 1. Algoritma ini berusaha mengeliminasi besarnya efek dari turunan parsial dengan cara hanya menggunakan turunannya saja dan mengabaikan besarnya nilai turunan serta untuk menghindari perubahan gradient yang terlalu kecil selama proses *update* yang dapat menyebabkan pembentukan jaringan menjadi lambat. Dalam proses update *weight*, Rprop memiliki faktor delta, dimana nilai delta akan mengikuti arah perubahan *weight*. Jika perubahan *weight* kecil maka nilai delta akan membesar, dan sebaliknya ketika perubahan weight aktif maka nilai delta akan mengecil (Demuth & Beale, 2000).

2. Perbaikan dengan Teknik Optimasi Numeris

Teknik ini terbagi menjadi 2 macam, yaitu:

a. Algoritma *Conjugate Gradient* (CGA)

Pada algoritma ini pengaturan bobot tidak selalu dilakukan dalam arah turun seperti pada metode *gradient descent*, tapi menggunakan *conjugate gradient* dimana pengaturan bobot tidak selalu dengan arah menurun tapi disesuaikan dengan arah konjugasinya. Dan yang menjadi perbedaan utama algoritma ini dari yang lain adalah pencarian nilai negatif dari gradient dalam jaringan sejak iterasi pertama. Algoritma ini memanfaatkan fungsi *line search*

untuk mendapatkan sebuah titik minimum. Terdapat beberapa variasi perhitungan nilai β dalam algoritma *Conjugate Gradient*, yaitu : *Fletcher-Reeves Update* (traincgf), *Polak-Ribiere Updates* (traincgp), *Powell-Beale Restarts* (traincgb) dan *Scaled Conjugate Gradient* (trainscg). Pada umumnya, algoritma ini bekerja lebih cepat dibandingkan dengan Rprop (Demuth & Beale, 2000).

b. Algortima *Quasi Newton*

Metode *Newton* merupakan salah satu alternatif *conjugate gradient* yang bisa mendapatkan nilai optimum lebih cepat. Metode *Newton* ini memang berjalan lebih cepat, namun metode ini sangat kompleks, memerlukan waktu dan memori yang cukup besar karena pada setiap iterasinya harus menghitung turunan kedua, perbaikan dari metode ini dikenal dengan nama metode *Quasi-Newton* atau metode *Secant*. Terdapat 2 (dua) alternatif algoritma dalam metode ini, yaitu:

1) Algortima *one step secant* (trainoss) yang menjembatani antara metode *Quasi-Newton* dengan *Gradient Conjugate*, dimana algoritma ini tidak menyimpan matriks Hessian secara lengkap dengan asumsi bahwa pada setiap iterasi matriks Hessian sebelumnya merupakan matriks identitas sehingga pencarian arah baru dapat dihitung tanpa harus menghitung invers matriks.
2) Algoritma *Levenbarg-Marquardt* (trainlm/LM) yaitu metode yang dirancang dengan menggunakan turunan kedua tanpa harus menghitung matriks Hessian, melainkan matriks Jacobian yang dapat dihitung dengan

teknik propagasi balik standar yang tentu saja lebih sederhana dibanding dengan menghitung matriks Hessian.

1.2.9 Algoritma *Conjugate Gradient Fletcher-Reeves*

Algoritma *conjugate gradient* dikembangkan oleh E. Stiefel dan M.R. Hestenes. Pertama kali metode ini digunakan untuk menyelesaikan persamaan linear atau persamaan matriks secara iteratif (Widyastuti, 2004).

Masalah optimasi nonlinier tanpa kendala merupakan pencarian nilai minimum dari fungsi bernilai $f(x)$ yaitu:

$$min\ f(x), x \in R^n \quad 0.6$$

Dimana f adalah fungsi nonlinier yang kontinu dan terdiferensialkan dengan gradien $g_x = \nabla f_x$ Masalah tersebut dapat diselesaikan secara numerik, yaitu dengan cara iteratif. Iterasi yang dijalankan dinotasikan dengan X_k, k =1, 2, ... Setiap iterasi dideskripsikan sebagai berikut:

$$x_{k+1} = x_k + \alpha_k d_k \quad 0.7$$

α_k > 0 adalah panjang langkah *(steplength)* yang ditentukan oleh suatu pencarian garis, d_k didefinisikan oleh:

$$d_k = \begin{cases} -g_k & if\ k = 1 \\ -g_k + \beta_k d_{k-1}, & if\ k > 1 \end{cases} \quad 0.8$$

Dengan $g_k = g(x_k)$ adalah gradien dari f di titik x_k, β_k adalah parameter yang jika digunakan untuk meminimumkan fungsi kuadratik yang konveks ketat, maka arah pencarian d_k dan d_{k+1} merupakan konjugat berdasarkan Hessian dari fungsi objektif (Zhang, Zhou, & Li, 2006). (Dai, Liao, & Li, 2004) menyatakan

bahwa jika fungsi objektifnya adalah fungsi kuadratik yang konveks ketat dan menggunakan pencarian garis eksak, maka metode *conjugate gradient* menghasilkan arah-arah pencarian yang konjugat satu dengan lainnya. Terdapat banyak formula β_k yang terkenal, salah satunya yaitu *Fletcher-Reeves* (FR). Formula untuk algoritma *Conjugate Gradient Backpropagation Fletcher Reeves* (Traincgf) yaitu :

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \quad 0.9$$