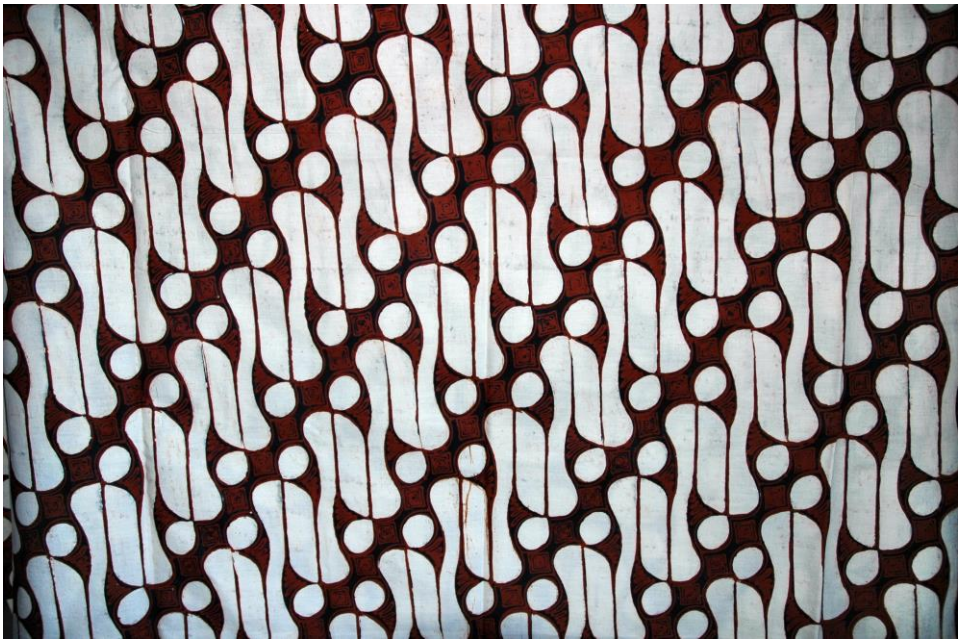
BAB 2
LANDASAN TEORI

2.1. Batik

Batik merupakan budaya yang telah lama berkembang dan dikenal oleh masyarakat Indonesia. Secara etimologi kata batik berasal dari bahasa Jawa, yaitu ”tik” yang berarti titik / matik (kata kerja, membuat titik) yang kemudian berkembang menjadi istilah ”batik” [1]. Motif batik di setiap daerah memiliki keunikan dan ciri khas masing-masing, baik dalam ragam hias maupun tata warna batik itu sendiri. Tradisi membatik awalnya merupakan tradisi turun menurun, sehingga motif batikannya pun dapat dikenali dan menjadi motif khas di daerah tertentu. Perkembangan batik di Indonesia memuncak pada tanggal 2 Oktober 2009 UNESCO saat ditetapkan bahwa batik adalah hak kebudayaan bangsa Indonesia. Sehingga sampai sekarang batik bisa dikatakan sudah mendarah daging, karena sering dipakai hampir dalam kegiatan Salah satu sarana bagi kalangan muda Indonesia dalam mengenal budaya batik dapat berkunjung ke Museum Tekstil yang berlokasi di Tanah Abang, Jakarta Pusat, sebagai lembaga edukatif kultural untuk melestarikan budaya tekstil tradisional, salah satunya yaitu batik.

Menurut H. Santosa Doellah [4] dalam bukunya yang berjudul “Batik: Pengaruh Zaman Dan Lingkungan” motif batik terbagi menjadi dua bagian besar, yaitu motif geometri dan non geometri. Pada motif geometri terbagi menjadi tiga jenis yaitu motif parang, motif ceplok, dan motif lereng. Motif batik geometri yang pertama yaitu motif parang. Motif parang merupakan motif yang memiliki pola yang terdiri atas satu atau lebih ragam hias yang tersusun membentuk garis-garis sejajar dengan sudut miring 45^{o}.



Gambar 2.1 Motif Batik Parang

Jenis motif geometri yang kedua adalah motif ceplok. Motif batik ceplok memiliki ciri-ciri dimana didalam batik tersebut terdapat gambar-gambar segi empat, lingkaran dan segala variasinya dalam membuat sebuah pola yang teratur.



Gambar 2.2 Motif Batik Ceplok

Jenis motif geometri yang ketiga adalah motif lereng (liris). Pada dasarnya motif lereng sama dengan motif parang tetapi memiliki perbedaan pada tidak adanya hias mlinjon dan hias gareng.



Gambar 2.3 Motif Batik Lereng

Selain itu pada motif non geometri dibagi menjadi empat jenis yaitu motif semen, motif lung lungan, motif buketan dan motif khusus. Motif non geometri yang pertama yaitu motif semen. Motif semen memiliki ragam hias utama yang merupakan ciri pola semen yaitu meru.

Motif non geometri yang kedua yaitu motif lung lungan. Sebagian besar motif lung lungan memiliki ragam hias utama serupa dengan motif semen. Yang membedakan motif lung-lungan dengan motif semen adalah ragam hias utama lung-lungan tidak selalu mengandung ragam hias meru.



Gambar 2.4 Motif Batik Lung Lungan

Motif non geometri yang ketiga yaitu motif buketan. Pola buketan mudah dikenali melalui rangkaian bunga atau kelopak bunga dengan kupu-kupu, burung, atau berbagai satwa kecil mengelilinginya.



Gambar 2.5 Motif Batik Buketan

Motif non geometri yang terakhir yaitu motif khusus. Motif khusus memuat motif yang tidak dapat dimasukkan ke dalam kelas yang lain. Motif ini banyak mempertemukan dua atau lebih motif lain yang digabung menjadi satu motif baru.



Gambar 2.6 Motif Batik Campuran (1)



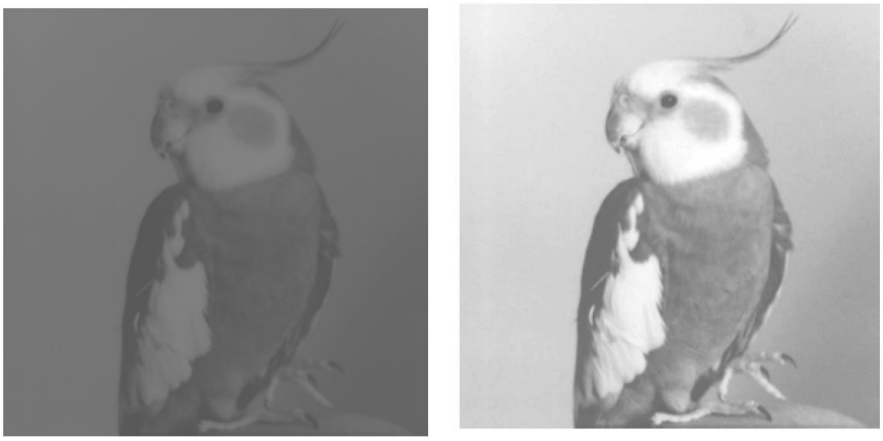
Gambar 2.7 Motif Batik Campuran (2)

2.2. Pengolahan Citra

Citra merupakan istilah lain untuk gambar, sebagai salah satu komponen multimedia memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Ada sebuah peribahasa yang berbunyi "sebuah gambar bermakna lebih dari seribu kata" (*a picture is more than a thousand words*) [10].

Secara harfiah, citra adalah gambar pada bidang dwimatra (dua dimensi). Meskipun sebuah citra kaya informasi, namun seringkali citra yang dimiliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*).

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Sebagai contoh, citra burung nuri pada Gambar 2.8 (a) tampak agak gelap, lalu dengan operasi pengolahan citra kontrasnya diperbaiki sehingga menjadi lebih terang dan tajam (b).



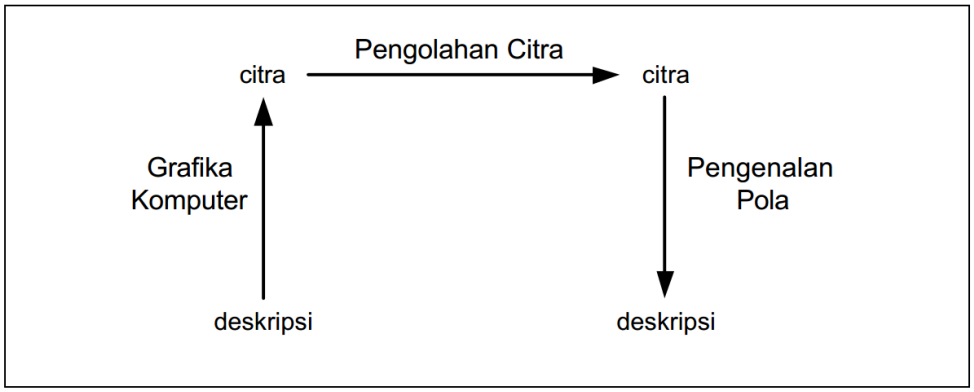
(a) (b)

Gambar 2.8 Perbaikan Kontras Citra

Di dalam bidang komputer, sebenarnya ada tiga bidang studi yang berkaitan dengan data citra, namun tujuan ketiganya berbeda, yaitu:

1. Grafika Komputer (*computer graphics*).
2. Pengolahan Citra (*image processing*).
3. Pengenalan Pola (*pattern recognition/image interpretation*).

Hubungan antara ketiga bidang (grafika komputer, pengolahan citra, pengenalan pola) ditunjukkan pada gambar 2.9.



Gambar 2.9 Tiga Bidang Studi Yang Berkaitan Dengan Citra

Grafika Komputer bertujuan menghasilkan citra (lebih tepat disebut grafik atau *picture*) dengan primitif-primitif geometri seperti garis, lingkaran, dan sebagainya. Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar. Contoh data deskriptif adalah koordinat titik, panjang garis, jari-jari lingkaran, tebal garis, warna, dan sebagainya. Grafika komputer memainkan peranan penting dalam visualisasi dan *virtual reality*.

Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan. Termasuk ke dalam bidang ini juga adalah pemampatan citra (*image compression*).

Pengenalan Pola mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar

mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra.

2.2.1. Grayscale

Untuk mengkonversi citra RGB ke *grayscale* dapat dilakukan dengan persamaan berikut.

X = (R+G+B)/3 (2.1)

R = warna merah pada suatu piksel

G = warna hijau pada suatu piksel

B = warna biru pada suatu piksel

Di setiap pikselnya terdapat masing-masing nilai pada R (merah), G (hijau), dan B (biru), sehingga ketika digunakan persamaan tersebut piksel-piksel akan hanya akan memiliki warna gradasi dari warna putih sampai hitam.

2.2.2. Histogram Equalization

Histogram didefinisikan sebagai probabilitas statistik distribusi setiap tingkat abu-abu dalam gambar digital. Persamaan *histogram* (HE) adalah teknik yang sangat populer untuk peningkatan kontras gambar.

$$S_k = \frac{(L-1)}{n} \sum_{j=0}^{k} n_{rj} \quad (2.2)$$

S_k = output HE

k = nilai yang muncul dalam citra

L = derajat keabuan

n = jumlah seluruh piksel dalam citra

n_{rj} = jumlah yang muncul di setiap nilai k

2.3. Kecerdasan Buatan

2.3.1. Pengertian Kecerdasan Buatan

Kecerdasan Buatan atau AI (*Artificial Intelligence*) merupakan suatu ilmu yang mempelajari bagaimana membuat komputer melakukan sesuatu pada suatu kejadian atau peristiwa yang mana orang melakukannya dengan baik. AI merupakan proses di mana peralatan mekanik dapat melaksanakan kejadian kejadian dengan menggunakan pemikiran atau kecerdasan seperti manusia [11].

AI adalah sub bagian dari ilmu komputer yang merupakan suatu teknik perangkat lunak yang pemrogramannya dengan cara menyatakan data, pemrosesan data dan penyelesaian masalah secara simbolik, dari pada secara numerik. Masalah-masalah dalam bentuk simbolik ini adalah masalah-masalah yang sering dijumpai dalam kehidupan sehari-hari. Masalah-masalah ini lebih berhubungan dengan simbol dan konsep simbol dari pada dengan angka-angka. Di sini dengan kecerdasan buatan diusahakan untuk membuat komputer seakan dapat berpikir secara cerdas.

2.3.2. Kategori Dasar AI

Teknik pemrograman dengan kecerdasan buatan melakukan prosesnya sama dengan apa yang dilakukan oleh otak manusia. Kecerdasan buatan juga meniru proses belajar manusia di mana informasi yang baru diserap dan dimungkinkan untuk digunakan sebagai referensi pada waktu yang akan datang. Di sini informasi yang baru dapat disimpan tanpa harus mengubah cara kerja pikiran atau mengganggu seluruh fakta-fakta yang sudah ada. Sehingga dengan kecerdasan buatan dimungkinkan untuk membuat program di mana setiap bagian dari program benar-benar independen. Di sini setiap bagian dari program seperti potongan-potongan informasi dalam otak manusia. Secara umum kecerdasan buatan dibagi menjadi tiga kategori dasar, yaitu:

1. Sistem Berbasis Pengetahuan atau sistem pakar (*Expert System/Knowledge Based System*), yaitu program komputer yang berisi pengetahuan manusia yang digunakan untuk menyelesaikan masalah dalam domain tertentu.

2. Sistem bahasa alami (*Natural Language System)*, yaitu pemrograman yang mengerti bahasa manusia.
3. Sistem dengan kemampuan memahami (*Perception System*), yaitu sistem untuk penglihatan, pembicaraan atau sentuhan.

2.3.3. Implementasi AI

AI dapat diimplementasikan ke dalam berbagai bidang, yaitu :

1. *Machine Vision :* Bertujuan pada pengenalan pola dalam beberapa jalan yang sama sebagai kegiatan sistem visual/indera manusia.
2. *Robotics :* Difokuskan pada produksi alat-alat mekanik yang dapat mengendalikan gerak. Sebagai contoh: sebuah robot sederhana dapat bergerak atau berpindah ke depan, belakang, kanan atau kiri atau berpindah tempat ke ruangan berbeda. Sebuah robot sebenarnya buta akan bentuk urutan dari aksi bila tanpa usaha untuk mengganti komponennya atau bisa mendeteksi dan memperbaiki kesalahan dalam rencananya akan menjadi sulit bila tanpa kecerdasan. Sering sebuah robot akan diformulasikan pada sebuah rencana dasar pada informasi yang tidak lengkap dan benar dalam menjalankan sebuah rencana.
3. *Speech Processing :* Bertujuan pada pengenalan dan sintesa pembicaraan manusia.
4. *Theorem Proving :* Usaha untuk membuktikan secara otomatis masalah-masalah dalam matematika dan logika.
5. *General Problem Solving :* Bertujuan pada pemecahan kelas-kelas dari masalah-masalah yang ditekankan dalam sebuah bahasa formal.
6. *Pattern Recognation:* Difokuskan pada pengenalan dan klasifikasi dari pola-pola.
7. *Game Playing:* Pembuatan program-program bermain permainan.
8. *Machine Learning :* Bertujuan pada produksi mesin-mesin yang mengakumulasi pengetahuan dengan contoh-contoh observasi.
9. *Learning* merupakan sebuah persoalan sulit untuk program AI, dalam mencapai kesuksesan diperlukan dalam pemecahan persoalan. Bagian

kemampuan untuk mempelajari komponen terpenting dari jalan kecerdasan. Sebuah sistem pakar harus berkemampuan ekstensif dan dapat menghitung kerugian dalam memecahkan sebuah persoalan.

10. *Planning* adalah aspek terpenting pendukung untuk mendesain atau merancang robot-robot dengan kemampuan menyelesaikan tugasnya dengan tingkat fleksibelitas dan tanggap terhadap dunia luar.
11. *Neural Network atau Parallel Distributed* : teknik-teknik terbaik untuk merepresentasikan pengetahuan dan merancang algoritma pencarian yang hati-hati untuk implementasi kecerdasan.

2.4. Jaringan Syaraf Tiruan

2.4.1. Pengertian Jaringan Syaraf Tiruan

Jaringan syaraf tiruan, bersama sistem pakar dan perangkat lunak, ternyata memberikan solusi persoalan dunia industri, telekomunikasi dan informasi dengan berbagai aplikasi masa kini maupun masa depan. Dalam bidang ilmu pengetahuan jaringan syaraf tiruan sudah sejak lama dibicarakan banyak orang. Mulai dikenal akhir tahun 1940-an, jaringan syaraf masuk dalam blok perkembangan teknologi komputer. Meski begitu, anehnya perkembangan teknologi komputer itu pulalah yang jadi penghambat berkembangnya ilmu jaringan syaraf. Meski riset dan pengembangan teknologi komputer terus berjalan, jaringan syaraf kurang begitu mendapat perhatian.

Ternyata kini jaringan syaraf tiruan dapat menjawab beberapa persoalan dunia telekomunikasi, industri maupun informasi, yang tak terlintas sebelumnya. Dan melihat prospeknya di masa depan, para ahli yang sangat fanatik pada model komputer digital konvensional, boleh jadi berbalik menyesali diri.

Jaringan syaraf tiruan (JST) adalah sistem pengolahan informasi yang didasari fisolofi struktur perilaku syaraf makhluk hidup [13]. Dengan begitu, jaringan syaraf tiruan tak diprogram selayaknya mekanisme pada komputer digital konvensional. Begitu juga dari segi arsitekturnya. Dalam arsitekturnya, jaringan syaraf mempelajari bagaimana menghasilkan keluaran yang diinginkan pada saat diberikan sekumpulan masukan. Proses ini dilakukan secara internal, yaitu dengan

memerintahkan sistem untuk mengidentifikasikan hubungan antar masukan kemudian mempelajari respon tersebut. Dengan metoda pensintesisan hubungan, jaringan syaraf dapat mengenal situasi yang sedang dan telah dijumpai sebelumnya. Berbeda dengan proses internal, proses eksternal lebih tergantung pada aplikasinya. Sistem bisa menggunakan umpan balik eksternal atau sinyal tanggapan yang diinginkan, untuk membentuk prilaku jaringan. Ini disebut sebagai *supervised learning*. Dengan cara lain, jaringan dapat membangkitkan sinyal tanggapan yang diinginkan sendiri dalam skenario *unsupervised learning*.

Latar belakang dikembangkannya JST karena pada pemrograman beberapa aplikasi seperti *image recognition* (pengenalan citra), *speech recognition* (pengenalan suara), *weather forecasting* (peramalan cuaca) ataupun permodelan tiga dimensi, tak dapat dengan mudah dan akurat diterapkan pada set instruksi komputer biasa. Atas dasar itu, maka diterapkan arsitektur komputer khusus yang dimodel berdasar otak manusia.

2.4.2. Model Sel Syaraf (*Neuron*)

Satu sel syaraf terdiri dari tiga bagian, yaitu: fungsi penjumlah (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*). Secara matematis dapat digambarkan sebuah *neuron* k dengan menuliskan pasangan persamaan sebagai berikut :

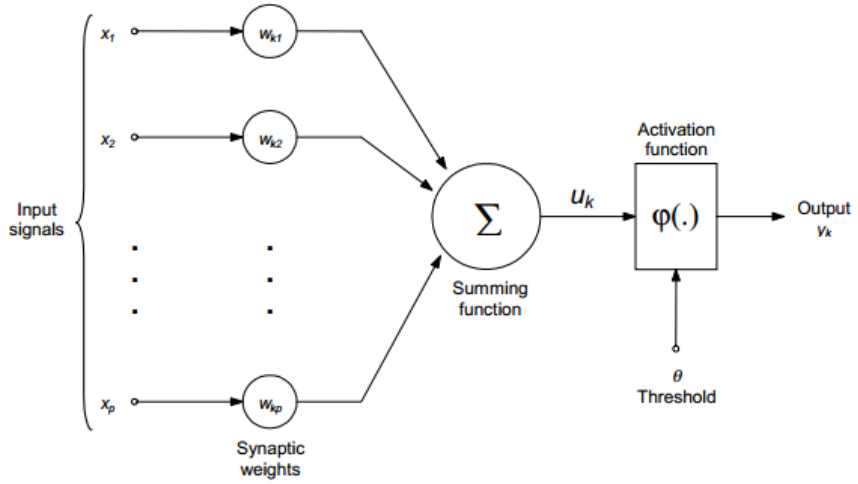
$$u_k = \sum_{j=1}^{p} w_{kj} x_j \tag{2.3}$$

dan

$$y_k = \varphi(u_k - \theta_k) \tag{2.4}$$

Dimana x1, x2, …, xp adalah sinyal input ; wk1, wk2, …, wkp adalah bobot-bobot sinaptik dari *neuron* k; uk adalah *linear combiner output*; θk adalah *threshold*; μ(.) adalah fungsi aktivasi; dan yk adalah sinyal *output* dari *neuron*. Penggunaan *threshold* memberikan pengaruh adanya *affine transformation* terhadap *output* uk dari *linear combiner* seperti yang ditunjukkan pada gambar 2.10.

$$v_k = u_k - \theta_k$$

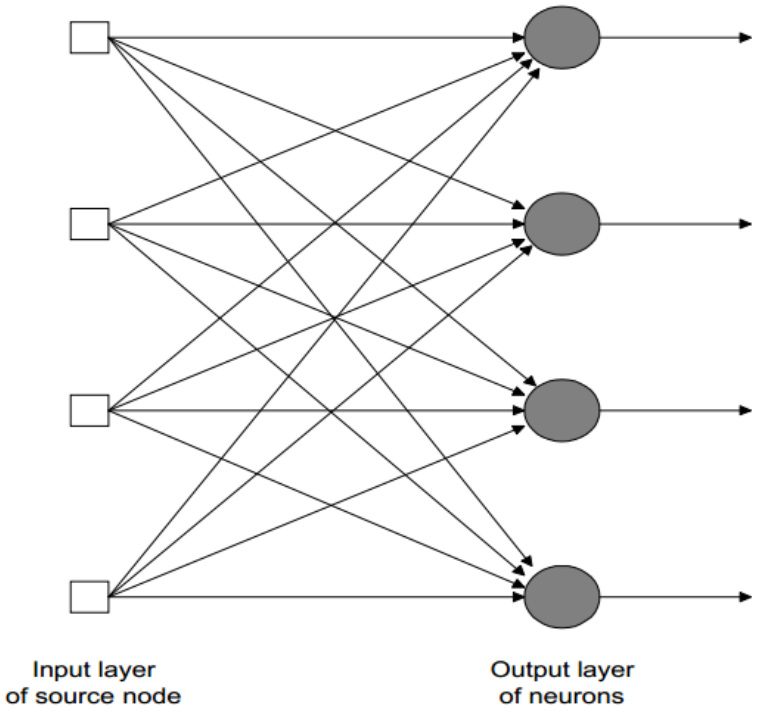


Gambar 2.10 Model Matematis Nonlinier Dari Suatu Neuron

2.4.3. Arsitektur Jaringan

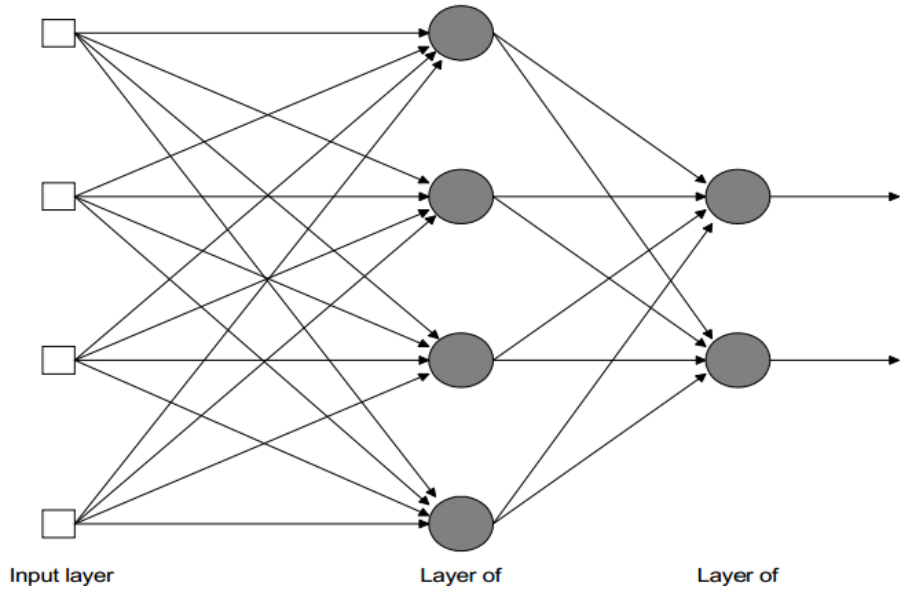
Pola dimana *neuron-neuron* pada JST disusun berhubungan erat dengan algoritma pembelajaran yang digunakan untuk melatih jaringan.

1. *Single-Layer Feedforward Networks*



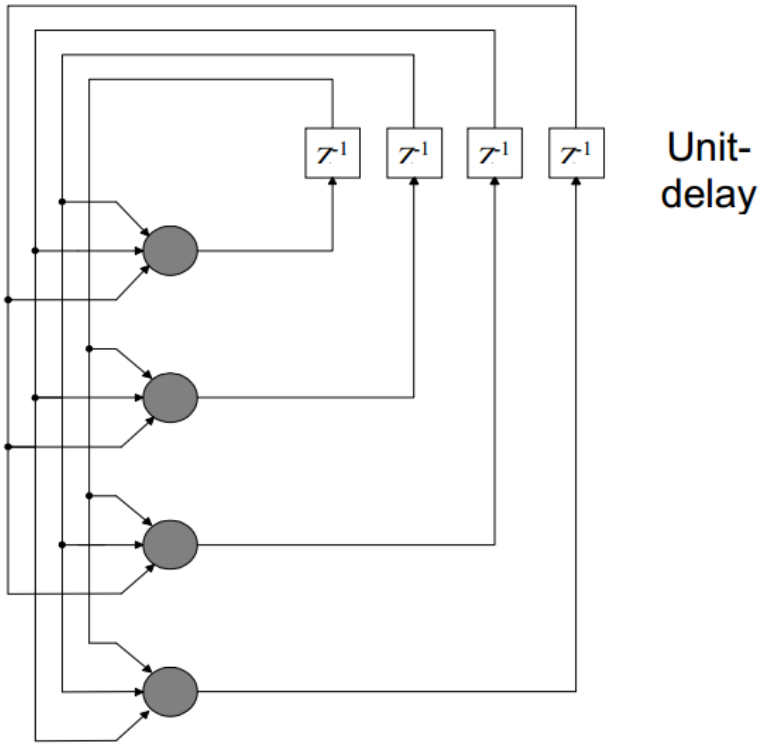
Gambar 2.11 *Feedforward Network* dengan satu lapisan neurons

2. *Multi-Layer Feedforward Networks*



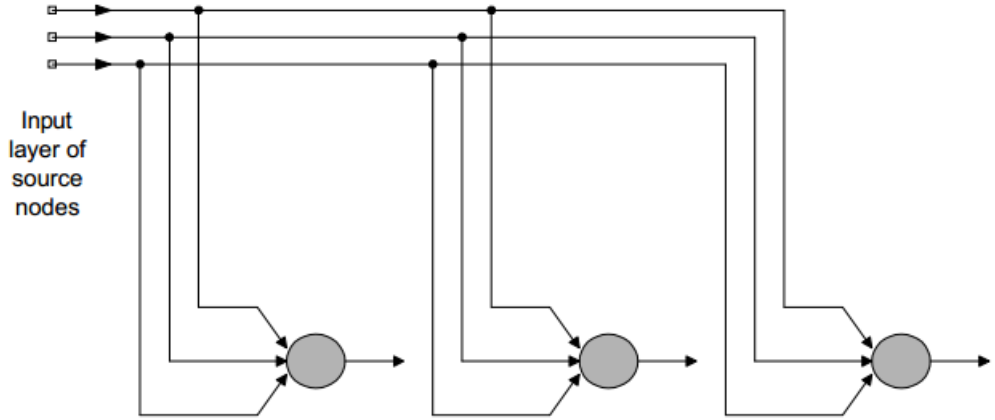
Gambar 2.12 *Feedforward Network* dengan lebih dari satu lapisan neurons

3. *Recurrent Networks*

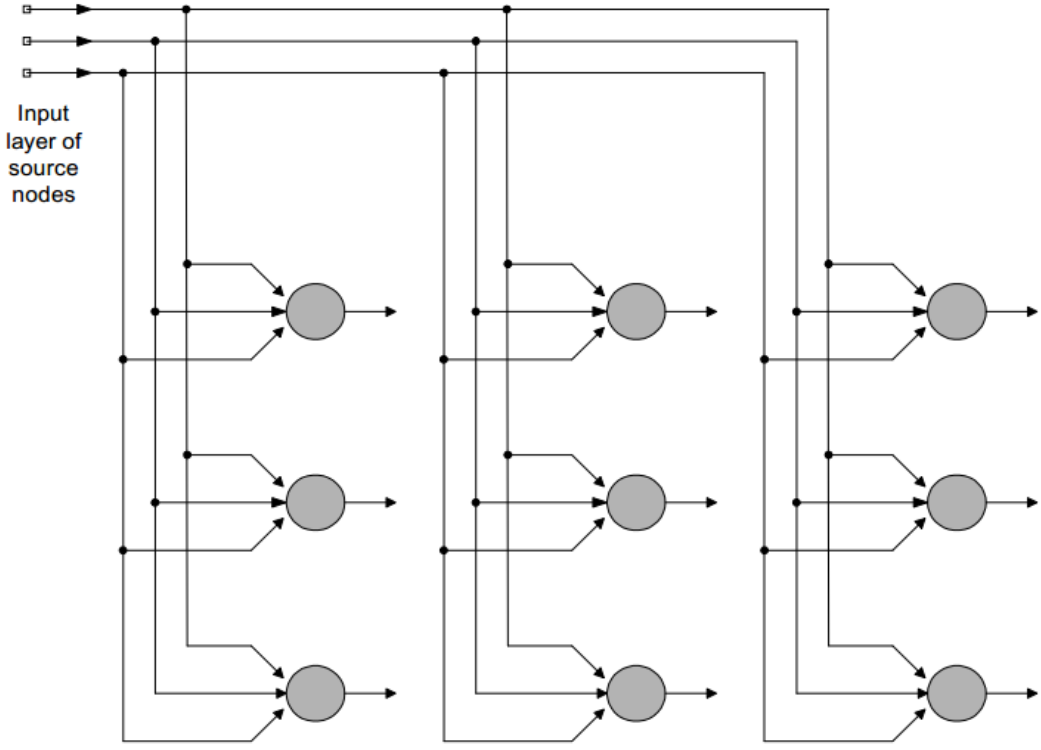


Gambar 2.13 *Recurrent network*

4. *Lattice Structure*



Gambar 2.14 *Lattice* satu dimensi dengan 3 *neuron*



Gambar 2.15 *Lattice* dua dimensi dengan 3 kali 3 *neuron*

2.4.4. Proses Pembelajaran

Proses pembelajaran pada JST adalah suatu proses dimana parameter-parameter bebas JST diadaptasikan melalui suatu proses perangsangan

berkelanjutan oleh lingkungan dimana jaringan berada. Definisi proses pembelajaran ini menyebabkan urutan kejadian sebagai berikut:

1. JST dirangsang oleh lingkungan.
2. JST mengubah dirinya sebagai hasil rangsangan ini.
3. JST memberikan respon dengan cara yang baru kepada lingkungan, disebabkan perubahan yang terjadi dalam struktur internalnya sendiri.

2.4.5. Algoritma Pembentukan JST

Tahap-tahap pembentukan JST dapat dilakukan dengan cara berikut [14].

1. Inisialisasi bobot = 0.
2. Menetapkan maksimum epoh, target error, dan learning rate
3. Inisialisasi Epoh = 0, MSE = 1.
4. Kerjakan langkah-langkah berikut selama (Epoh < maksimum epoh) dan (MSE > target error):
   a. Epoh = Epoh + 1 (2.5)
   b. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:
      - Tiap-tiap unit input (x_i, i=1,2,3…,n) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembuyi)
      - Tiap-tiap unit pada suatu lapisan tersembunyi (z_j, j=1,2,3…,p) menjumlahkan sinyal-sinyal input terbobot:

        $$z_in_j = b1_j + \sum_{i=1}^{n} x_i \, v_{ij} \quad (2.6)$$

        Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

        $z_j = f(z_in_j)$ (2.7)

        dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output)
      - Tiap-tiap unit output (y_k, k=1,2,3,…,m) menjumlahkan sinyal-sinyal input terbobot.

$$y_in_k = b2_k + \sum_{i=1}^{p} z_i \, w_{jk} \quad (2.8)$$

Gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$y_k = f(y_in_k)$ (2.9)

dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit output)

- Tiap-tiap unit output (yk, k=1,2,3,…,m) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya:

  $\delta 2_k = (t_k - y_k)\, f'(y_in_k)$ (2.10)

  $\varphi 2_{jk = k}\, z_j$ (2.11)

  $\beta 2_k = \delta_k$ (2.12)

  kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

  $\Delta w_k = \alpha\, \varphi 2 j_k$ (2.13)

  hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai $b2_k$):

  $\Delta b2_k = \alpha \beta 2_k$ (2.14)

  Langkah di atas juga dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi error dan suatu laposan tersembunyi ke lapisan tersembunyi sebelumnya.

- Tiap-tiap unit tersembunyi (z_j, j=1,2,3,…,p) menjumlahkan delta inputnya (dari unit-unit berada pada lapisan di atasnya):

  $$z_in_j = \sum_{k=1}^{m} \delta 2_k \, w_{jk} \quad (2.15)$$

  Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi error:

  $\delta 1_{j =} \delta_in_j\, f'(z_in_j)$ (2.16)

  $\varphi 1_{ij} = \varphi 1_j\, x_j$ (2.17)

  $\beta 1_j = \delta 1_j$ (2.18)

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{ij} = \alpha \varphi 1_{ij} \quad (2.19)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai $b1_j$):

$$\Delta b1_j = \alpha \beta 1_j \quad (2.20)$$

- Tiap-tiap unit output (Y_k, k = 1,2,3,…,m) memperbaiki bias dan bobotnya (j=0,1,2,…,p):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.21)$$

$$b2_k(\text{baru}) = b2_k(\text{lama}) + \Delta b2_k \quad (2.22)$$

tiap-tiap unit tersembunyi (z_j, j=1,2,3,…,p) memperbaiki bias dan bobotnya (i=0,1,2,…,n):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.23)$$

$$b1_j(\text{baru}) = b1_j(\text{lama}) + \Delta b1_j \quad (2.24)$$

c. Hitung MSE

2.5. Algoritma Levenberg Marquardt

Algoritma Levenberg Marquardt dikembangkan oleh Kenneth Levenberg dan Donald Marquardt, memberikan solusi numerik untuk masalah meminimalkan fungsi nonlinear [16]. Di bidang jaringan syaraf tiruan, algoritma ini cocok untuk pelatihan kecil dan menengah.

Algoritma Levenberg Marquadt merupakan pengembangan dari algoritma *backpropagation*. Algoritma ini dibangun untuk mengatasi beberapa kekurangan yang ada pada algoritma *backpropagation* dengan memanfaatkan teknik optimisasi numerik standar yaitu menggunakan pendekatan matriks *Jacobian.* Tujuan dari Levenberg Marquadt adalah meminimalkan total *error*. Pada algoritma *backpropagation* proses *update* bobot dan bias menggunakan *negative gradient descent* secara langsung, sedangkan algoritma Levenberg Marquardt menggunakan pendekatan matrik Hesian (H) yang dapat dihitung dengan:

$$H = J^T J \quad (2.25)$$

Sedangkan *gradient* dapat dihitung dengan:

$$g = J^T e \quad (2.26)$$

Dalam hal ini J merupakan sebuah matrik *jacobian* yang berisikan turunan pertama dari *error* jaringan terhadap bobot dan bias jaringan. Perubahan pembobot dapat dihitung dengan:

$$\Delta W = [J^T J + \mu I] - J^T e \quad (2.27)$$

Sehingga perbaikan pembobot dapat ditentukan dengan:

$$W = W + \Delta W \quad (2.28)$$

$$W = W + [J^T J + \mu I] - J^T e \quad (2.29)$$

Keterangan :

W = fungsi bobot-bobot jaringan dan bias

e = vektor yang menyatakan semua error pada output jaringan

μ = konstanta learning

I = matrik identitas

2.6. Inisialisasi Bobot Awal Dengan Metode Nguyen Widrow

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf dalam mencapai minimum global terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Jikai nilai bobot awal terlalu besar, maka input ke setiap lapisan tersembunyi atau lapisan output akan jatuh pada daerah dimana turunan fungsi sigmoidnya akan sangat kecil. Jika nilai bobot awal awal terlalu kecil, maka input ke setiap lapisan tersembunyi atau lapisan output akan sangan kecil yang akan mengakibatkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara *random* dengan nilai antara -0,5 sampai 0,5.

Nguyen Widrow merupakan modifikasi sederhana bobot-bobot dan bias ke unit tersembunyi yang mampu meningkatkan kecepatan jaringan dalam proses pelatihan jaringan. Metode ini secara sederhana dapat diimplementasikan dengan prosedur sebagai berikut.

1. Tetapkan :

$$\beta = 0.7(p)^{1/n} \quad (2.30)$$

dimana

n = jumlah neuron pada lapisan input

p = jumlah neuron pada lapisan tersembunyi

β = faktor penskalaan

2. Kerjakan untuk setiap unit pada lapisan tersembunyi (j = 1, 2…., p).
   a. Inisialisasi bobot-bobot dari lapisan input ke lapisan tersembunyi:
   b. Hitung: $||V_j||$
   c. Set bias :

      $b1_j$ = bilangan *random* antara -β sampai β.

2.7. Pengujian JST

Pada proses pengujian, tahap yang dilakukan hanya sampai tahap maju saja, tidak ada tahap mundur dan tahap modifikasi bobot. Seluruh bobot *input* diambil dari nilai bobot terakhir yang diperoleh dari proses pelatihan. Pada tahap pengujian ini, jaringan diharapkan dapat mengenali pola berdasarkan data baru yang diberikan (generalisasi).

2.8. Diagram Arus Data (*Data Flow Diagram*)

Data flow diagram adalah suatu gambaran secara logika, *data flow diagram* biasa digunakan untuk membuat sebuah model sistem informasi dalam bentuk proses-proses yang saling berhubungan yang disebut dengan aliran data.

1. Arus Data (*Data Flow*)

Arus data ini menunjukan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. Pada DFD, arus data digambarkan dengan tanda panah.

2. Proses

Proses adalah kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Pada DFD, proses digambarkan dalam simbol lingkaran.

3. Kesatuan Luar (*External Entity*)

Kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang akan memberikan masukan (*input*) atau menerima keluaran (*output*) dari sistem. Kesatuan luar digambarkan dengan simbol kotak persegi panjang.

4. File

File digambarkan dengan dua garis *horizontal*.

Konsep dasar DFD dapat dilakukan dengan analisa *Top Down*, yaitu pemecahan sistem yang besar menjadi beberapa sub-sub sistem yang lebih kecil DFD terdiri dari :

1. Diagram Konteks

Diagram Konteks adalah diagram arus data yang berfungsi untuk menggambarkan yang dirancang disuatu objek, diagram konteks ini menggambarkan secara global atau menyeluruh dari suatu sistem informasi keterkaitan aliran-aliran data antara system dengan bagian-bagian luar. Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut.

2. Middle Level

Merupakan pemecahan dari tiap-tiap proses yang mempunyai fungsi sama. Pada *middle level* diagaram 0 dipecah menjadi diagram 1,2,3 dan seterusnya yang merupakan penguraian dari diagram konteks.

3. Lowest Level

Merupakan pemecahan dari *data flow* yang ada pada *middle level*. Pemecahan tersebut masih tetap mempunyai fungsi yang sama dari level sebelumnya. Untuk *Lowest Level*, pemberian nomor diagram terdiri dari bagian *middle level*.

2.9. MATLAB

MATLAB (*Matrix Laboratory*) adalah sebuah program untuk analisis dan komputasi numerik, merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks [15]. Pada awalnya, program ini merupakan *interface* untuk koleksi rutin-rutin

numerik proyek LINPACK dan EISPACK, dikembangkan menggunakan bahasan FORTRAN. Namun sekarang, program ini merupakan produk komersial dari perusahaan Mathworks, Inc. yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (terutama untuk fungsi-fungsi dasar MATLAB).

MATLAB telah berkembang menjadi sebuah *environment* pemrograman yang canggih dan berisi fungsi-fungsi *built-in* untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. MATLAB juga berisi *toolbox* yang berisi fungsi-fungsi tambahan untuk aplikasi khusus. MATLAB bersifat *extensible*, dalam arti bahwa seorang pengguna dapat menulis fungsi baru untuk ditambahkan di *library* jika fungsi-fungsi *built-in* yang tersedia tidak dapat melakukan tugas tertentu. MATLAB dapat menggunakan teknik berorientasi objek dalam desain program. Teknik berorientasi objek dapat menyederhanakan pemrograman tugas yang melibatkan struktur data khusus atau sejumlah besar fungsi yang berinteraksi dengan jenis data khusus.

Dalam lingkungan perguruan tinggi teknik, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Di industri, Matlab merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tinggi, pengembangan dan analisanya. Fitur-fitur Matlab sudah banyak dikembangkan, dan lebih dikenal dengan nama *toolbox*. *Toolbox* ini merupakan kumpulan dari fungsi -fungsi Matlab (*M-files*) yang telah dikembangkan ke suatu lingkungan kerja Matlab untuk memecahkan masalah dalam kelas *particular*. Area-area yang sudah bisa dipecahkan dengan *toolbox* saat ini meliputi pengolahan sinyal, sistem kontrol, *neural networks*, *fuzzy logic*, *wavelets*, dan lain-lain.

Kelengkapan pada Sistem Matlab sebagai sebuah sistem, Matlab tersusun dari 5 bagian utama:

1. *Development Environment*.

   Development Environment merupakan sekumpulan perangkat dan fasilitas yang membantu anda untuk menggunakan fungsi-fungsi dan file-file Matlab. Beberapa perangkat ini merupakan sebuah *graphical user interfaces* (GUI).

Termasuk didalamnya adalah Matlab *desktop* dan *command window*, *command history*, sebuah *editor* dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, *files*, dan *search path*.

2. Matlab *Mathematical Function Library*.

   Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti *sum, sin, cos, complex arithmetic*, sampai dengan fungsi-fungsi yang lebih kompeks seperti *matrix inverse, matrix eigenvalues, bessel functions*, dan *fast fourier transforms*.

3. Matlab *Language*.

   Merupakan suatu *high-level matrix/array language* dengan *control flow statements, functions*, data *structures, input/output*, dan fitur-fitur *object-oriented programming*. Ini dimungkinkan untuk melakukan kedua hal baik "pemrograman dalam lingkup sederhana" untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang komplek.

4. *Graphics*.

   Matlab memiliki fasilitas untuk menampilkan *vector* dan *matrices* sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dikensi dan data tiga dimensi, *image processing*, *animation*, dan *presentation graphics*. Ini juga melibatkan fungsi level rendah yang dimungkinkan untuk memunculkan grafik mulai dari benutk yang sederhana sampai dengan tingkatan *graphical user interfaces* pada aplikasi Matlab.

5. Matlab *Application Program Interface* (API).

   Merupakan suatu *library* yang memungkinkan program yang telah ditulis dalam bahasa C dan Fortran mampu berinterakasi dengan Matlab. Ini melibatkan fasilitas untuk pemanggilan *routines* dari Matlab (*dynamic linking*), pemanggilan Matlab sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan MAT-*files*.

MATLAB banyak digunakan pada :

1. Matematika dan Komputasi
2. Pengembangan dan Algoritma
3. Pemrograman Modeling, simulasi, dan pembuatan prototipe
4. Analisa data, eksplorasi, dan visualisasi
5. Analisis numerik dan statistik
6. Pengembangan aplikasi teknik

2.10. Cross Validation

Untuk tujuan replikasi, yang dibutuhkan adalah mendapatkan jumlah dari data latih dan pasangan data tes dari dataset X (setelah lebih dahulu memisahkan sebagian data sebagai data uji). Untuk mendapatkannya, jika sampel X cukup besar, dapat dibagi secara acak ke dalam K bagian, lalu secara acak memisahkan masing-masing bagian ke dalam 2 kelompok dan menggunakan setengah data pertama untuk data latih dan setengah data lainnya untuk data uji. Pengujian ini dilakukan berulang dengan menggunakan data split yang berbeda, dinamakan *cross validation*.

K-fold *cross-validation* adalah sebuah pengujian dimana dataset X dibagi secara acak kedalam K bagian dengan ukuran yang sama, Xi, i=1,...,K. Untuk memproses setiap pasangnya diambil terlebih dahulu salah satu bagian dari K bagian sebagai data tes, dan mengkombinasikan sisanya sebanyak K-1 bagian sebagai data latih. Proses pengujian dilakukan sebanyak K kali [5].

Dalam contoh kasus yang memiliki banyak prediksi, hasil pada data uji sering ditampilkan sebagai confusion matrix dua dimensi dengan baris dan kolom untuk setiap kelas. Setiap elemen matrix menunjukkan jumlah data uji. Pada bagian baris adalah kelas data uji yang sebenarnya dan pada bagian kolom adalah kelas prediksi[20]. Contoh confusion matrix pada dapat dilihat pada tabel berikut.

Tabel 2.1 Contoh *Confusion Matrix*

	Hasil Identifikasi			
Target Identifikasi	**a**	**b**	**c**	**d**
A	4	0	0	0
B	1	3	0	0
c	1	0	2	1
d	0	0	0	4

Berikut ini adalah penjelasan pada tabel 2.1

Target a teridentifikasi menjadi a sebanyak 4
teridentifikasi menjadi b sebanyak 0
teridentifikasi menjadi c sebanyak 0
teridentifikasi menjadi d sebanyak 0

Target b teridentifikasi menjadi a sebanyak 1
teridentifikasi menjadi b sebanyak 3
teridentifikasi menjadi c sebanyak 0
teridentifikasi menjadi d sebanyak 0

Target c teridentifikasi menjadi a sebanyak 1
teridentifikasi menjadi b sebanyak 0
teridentifikasi menjadi c sebanyak 2
teridentifikasi menjadi d sebanyak 1

Target d teridentifikasi menjadi a sebanyak 0
teridentifikasi menjadi b sebanyak 0
teridentifikasi menjadi c sebanyak 0
teridentifikasi menjadi d sebanyak 4

Contents

BAB 2 7

2.1. Batik 7

2.2. Pengolahan Citra 12

2.3. Kecerdasan Buatan 15

2.4. Jaringan Syaraf Tiruan 17

2.5. Algoritma Levenberg Marquardt 24

2.6. Inisialisasi Bobot Awal Dengan Metode Nguyen Widrow 25

2.7. Pengujian JST 26

2.8. Diagram Arus Data (*Data Flow Diagram*) 26

2.9. MATLAB 27

2.10. Cross Validation 30

Gambar 2.1 Motif Batik Parang 8

Gambar 2.2 Motif Batik Ceplok 8

Gambar 2.3 Motif Batik Lereng 9

Gambar 2.4 Motif Batik Lung Lungan 10

Gambar 2.5 Motif Batik Buketan 10

Gambar 2.6 Motif Batik Campuran (1) 11

Gambar 2.7 Motif Batik Campuran (2) 11

Gambar 2.8 Perbaikan Kontras Citra 12

Gambar 2.9 Tiga Bidang Studi Yang Berkaitan Dengan Citra 13

Gambar 2.10 Model Matematis Nonlinier Dari Suatu Neuron 19

Gambar 2.11 *Feedforward Network* dengan satu lapisan neurons 19

Gambar 2.12 *Feedforward Network* dengan lebih dari satu lapisan neurons 20

Gambar 2.13 *Recurrent network* 20

Gambar 2.14 *Lattice* satu dimensi dengan 3 *neuron* 21

Gambar 2.15 *Lattice* dua dimensi dengan 3 kali 3 *neuron* 21

Tabel 2.1 Contoh *Confusion Matrix* 31