

BAB II

KAJIAN PUSTAKA

A. Kanker Serviks

Kanker serviks adalah sel kanker yang terbentuk pada daerah leher rahim. Leher rahim yaitu suatu daerah organ reproduksi wanita yang menghubungkan rahim ke vagina. Kanker serviks biasanya berkembang secara perlahan dari waktu ke waktu. Sebelum kanker muncul di leher rahim, sel-sel serviks melalui serangkaian perubahan di mana sel-sel yang tidak normal mulai muncul di jaringan serviks. Ketika sel-sel berubah dari sel normal ke sel-sel abnormal, hal itu disebut displasia. Seiring waktu, displasia yang tidak diobati dapat berubah menjadi kanker (www.cancer.gov).

1. Penyebab Kanker Serviks

Penelitian telah menemukan beberapa faktor yang dapat meningkatkan risiko kanker serviks, diantaranya yaitu (*American Cancer Society*, 2014):

a. Infeksi HPV

Penyebab utama terjadinya kanker serviks disebabkan oleh infeksi *Human Papilloma Virus* (HPV). HPV adalah kumpulan lebih dari 150 virus yang berhubungan, yang dapat menginfeksi sel-sel pada permukaan kulit dan dapat ditularkan melalui kontak kulit seperti vagina, anal, atau oral seks. Virus HPV berisiko rendah dapat menimbulkan penyakit kutil kelamin (*genital ward*) yang dapat sembuh dengan sendirinya dengan kekebalan tubuh. Namun pada virus HPV berisiko tinggi, seperti tipe 16, 18, 31, 33 dan 45 dapat mengubah permukaan sel-

sel vagina menjadi tidak normal. Bila tidak segera diobati, infeksi virus HPV ini dalam jangka panjang dapat menyebabkan terbentuknya sel-sel pra-kanker serviks.

b. Merokok

Peluang kemungkinan wanita yang merokok untuk terkena kanker serviks adalah sekitar dua kali lebih besar dibandingkan wanita yang tidak merokok.

c. Melemahnya Sistem Kekebalan Tubuh

Infeksi HIV (virus penyebab AIDS) atau mengonsumsi obat yang menekan sistem kekebalan tubuh dapat meningkatkan risiko kanker serviks. Sistem kekebalan tubuh penting dalam menghancurkan sel-sel kanker dan memperlambat pertumbuhannya.

d. Infeksi Chlamydia

Chlamydia adalah jenis bakteri yang dapat menginfeksi sistem reproduksi. Infeksi Chlamydia dapat menyebabkan risiko yang lebih tinggi untuk terkena kanker serviks.

e. Diet Rendah Buah dan Sayuran

Wanita yang diet buah-buahan dan sayuran kemungkinan mengalami peningkatan risiko untuk kanker serviks. Vitamin yang terkandung pada buah dan sayuran sangat dibutuhkan untuk menjaga kestabilan dan kesehatan tubuh.

f. Penggunaan Alat Kontrasepsi dalam Jangka Panjang

Menggunakan alat kontrasepsi, seperti kontrasepsi oral dan pil KB

untuk waktu yang lama (5 tahun atau lebih) sedikit meningkatkan risiko kanker serviks. Namun, risiko menurun dengan cepat ketika wanita berhenti menggunakan alat kontrasepsi tersebut.

g. Memiliki Beberapa Kehamilan Jangka Penuh

Wanita yang telah memiliki 3 atau lebih kehamilan jangka penuh memiliki peningkatan risiko mengembangkan kanker serviks. Penelitian telah menunjukkan perubahan hormon selama kehamilan sebagai kemungkinan yang membuat perempuan lebih rentan terhadap infeksi HPV atau pertumbuhan kanker.

h. Mengalami Kehamilan pada Usia Muda

Ketika wanita dengan usia dibawah 17 tahun memiliki kehamilan jangka penuh, mereka hampir 2 kali lebih mungkin untuk terkena kanker serviks di kemudian hari dibanding wanita yang menunggu untuk hamil sampai usia mereka 25 tahun atau lebih.

i. Pemakaian DES (diethylstilbestrol)

DES dapat meningkatkan risiko kanker serviks pada perempuan yang mengonsumsi obat ini sebelum kelahiran. DES diberikan kepada beberapa wanita hamil di Amerika Serikat antara sekitar 1940 dan 1971 untuk mencegah keguguran.

j. Kemiskinan

Banyak wanita yang tidak mampu dan tidak mempunyai akses ke layanan-layanan medis yang memadai. Ketika wanita tersebut menderita pra-kanker serviks, penyakit biasanya tetap tidak terdiagnosa dan tidak

diobati sampai penyakit itu berkembang menjadi kanker serviks dan menyebar ke bagian-bagian lain dari tubuh.

Beberapa faktor lain yang dapat meningkatkan risiko terkena kanker serviks adalah faktor kebersihan, kurangnya tes *pap smear* secara teratur, sedang terkena penyakit menular lain (misalnya herpes), memiliki banyak pasangan seksual, terkena paparan bahan kimia, dll (Tim kanker serviks, 2010: 7-9).

2. Gejala Kanker Serviks

Pada tahap awal, biasanya kanker serviks tidak menimbulkan gejala. Gejala sering tidak disadari hingga kanker telah berkembang lebih jauh dan telah menyebar ke daerah di sekitarnya. Gejala yang timbul akan semakin banyak apabila kanker serviks telah memasuki stadium lanjut. Gejala-gejala kanker serviks antara lain:

- a. Ada bercak atau pendarahan setelah hubungan seksual
- b. Ada bercak atau pendarahan di luar masa haid
- c. Ada bercak atau pendarahan pada masa menopause
- d. Mengalami masa haid yang lebih berat dan lebih panjang dari biasanya
- e. Keluarnya bau menyengat yang tidak bisa dihilangkan walaupun sudah diobati (Tim kanker serviks, 2010: 9)
- f. Keputihan yang tidak normal dari vagina, dengan ciri diantaranya: kental, warna kuning/kecoklatan, dapat berbau busuk dan/atau gatal
- g. Rasa sakit saat berhubungan seksual (www.cancerhelp.com).

3. Deteksi Dini Kanker Serviks

Pendeteksian kanker serviks dapat dilakukan dengan

menggunakan beberapa tes seperti berikut (Tim kanker serviks, 2010: 12-13):

a. Tes *Pap Smear*

Pap smear merupakan sebuah tes sederhana yang dapat digunakan untuk mengamati sel-sel pada permukaan serviks. Tes *pap smear* dilakukan dengan pengambilan sampel sel pada permukaan serviks yang kemudian dikirim ke laboratorium untuk diuji. Pengujian dilakukan dengan menggunakan mikroskop untuk melihat apakah ada perubahan sel atau tidak. Apabila ditemukan sel-sel abnormal pada permukaan serviks maka disarankan untuk menjalani tes lebih lanjut, seperti tes kolposkopi atau biopsi.

b. Tes IVA

IVA adalah singkatan dari Inspeksi Visual dengan Asam asetat, merupakan metode pemeriksaan dengan mengoles serviks dengan asam asetat. Setelah serviks diolesi dengan asam asetat, kemudian diamati apakah ada perubahan warna yang terjadi atau tidak. Perubahan warna dapat ditunjukkan dengan munculnya warna putih pada area serviks tertentu. Jika tidak ada perubahan warna, maka dapat dianggap tidak ada infeksi pada serviks. Tes ini dapat dilakukan hanya untuk deteksi dini.

c. Tes Kolposkopi

Tes kolposkopi adalah pemeriksaan yang dilakukan menggunakan alat bantu kolposkop untuk melihat keseluruhan leher rahim secara lebih mendetail. Kolposkop menggunakan cahaya terang dan lensa pembesar untuk membuat jaringan lebih mudah dilihat. Pemeriksaan ini akan dianjurkan jika hasil tes *pap smear* atau IVA menunjukkan adanya sel serviks abnormal.

d. Biopsi

Biopsi adalah proses pengambilan sel-sel abnormal dari daerah yang dicurigai pada serviks. Sampel jaringan ini kemudian akan dianalisis dengan mikroskop.

e. Biopsi Kerucut

Biopsi kerucut dilakukan dengan pengambilan sebuah sampel jaringan berbentuk kerucut. Pemeriksaan menggunakan biopsi kerucut memungkinkan ahli patologi untuk melihat apakah ada sel-sel abnormal dalam jaringan di bawah permukaan serviks. Anestesi atau bius total biasanya dilakukan selama tes tersebut.

Selain beberapa tes di atas, terdapat juga tes lain yang dapat dilakukan untuk mendeteksi kanker serviks, yaitu Tes HPV. Tes HPV adalah tes laboratorium yang digunakan untuk memeriksa DNA dalam beberapa jenis infeksi HPV. Sel dikumpulkan dari leher rahim dan DNA dari sel diperiksa untuk mengetahui apakah ada infeksi yang disebabkan oleh jenis HPV yang terkait dengan kanker serviks. Tes ini juga dapat dilakukan jika hasil tes *pap smear* menunjukkan adanya sel-sel serviks yang abnormal (www.cancer.gov).

4. Stadium Kanker Serviks

Stadium kanker adalah cara bagi paramedis untuk merangkum seberapa jauh kanker telah menyebar. Stadium kanker serviks dinyatakan dalam angka Romawi 0 s/d IV. Secara umum, angka yang lebih rendah menunjukkan semakin kecil kemungkinan kanker telah menyebar. Angka yang lebih tinggi, seperti stadium IV menunjukkan kanker yang lebih serius (www.cancerhelps.com).

Menurut Tim Kanker Serviks, stadium kanker serviks dibagi menjadi 5 (Tim kanker serviks, 2010: 17-21), yaitu:

a. Stadium 0 atau Normal

Stadium ini disebut juga “Carsinoma-in-situ” yang berarti “kanker yang berada di tempatnya”, belum menyerang bagian lain. Pada stadium ini, perubahan sel yang tidak wajar hanya ditemukan pada permukaan serviks. Stadium ini termasuk kondisi pra-kanker yang bisa diobati dengan tingkat kesembuhan mendekati 100%.

b. Stadium I

Stadium I berarti bahwa kanker baru berada di leher rahim. Stadium ini dibagi menjadi stadium IA dan stadium IB. Kanker serviks stadium IA terdiri dari kanker serviks stadium IA1 dan IA2. Begitu juga kanker serviks stadium IB terdiri dari kanker serviks stadium IB1 dan IB2. Pada stadium IA, pertumbuhannya begitu kecil sehingga kanker hanya bisa dilihat dengan sebuah mikroskop atau kolposkop. Pada Stadium IA1, kanker telah tumbuh kurang dari 3 mm ke dalam jaringan serviks, dan lebarnya kurang dari 7 mm. Pada stadium IA2, kanker telah tumbuh antara 3 sampai 5 mm ke dalam jaringan-jaringan serviks, tetapi lebarnya masih kurang dari 7 mm. Pada stadium IB, area kanker lebih luas, tetapi kanker masih berada dalam jaringan serviks dan biasanya masih belum menyebar. Kanker ini biasanya bisa dilihat tanpa menggunakan mikroskop, tetapi tidak selalu demikian. Pada stadium IB1, kanker tidak lebih besar dari 4 cm. Pada stadium IB2, kanker lebih besar dari 4 cm (ukuran horizontal).

c. Stadium II

Pada Stadium II, kanker mulai menyebar keluar dari leher rahim menuju ke jaringan-jaringan di sekitarnya. Akan tetapi kanker masih belum tumbuh ke dalam otot-otot atau ligamen dinding panggul, atau menuju ke vagina bagian bawah. Stadium II dibagi menjadi stadium IIA dan stadium IIB. Pada stadium IIA, kanker telah menyebar ke vagina bagian atas. Stadium IIA dibagi lagi menjadi stadium IIA1 dan stadium IIA2. Pada stadium IIA1 kanker berukuran 4 cm atau kurang, dan pada stadium IIA2 kanker berukuran lebih dari 4 cm. Pada Stadium IIB, kanker telah menyebar ke dalam jaringan di sekitar serviks. Kanker serviks stadium IIA dapat diobati dengan pembedahan, radioterapi, atau gabungan keduanya. Kanker serviks stadium IIB biasanya diobati dengan gabungan radioterapi dan kemoterapi.

d. Stadium III

Pada stadium III, kanker serviks telah menyebar jauh dari serviks menuju ke dalam struktur di sekitar daerah panggul. Kanker mungkin telah tumbuh ke dalam vagina bagian bawah dan otot-otot serta ligamen yang melapisi dinding panggul. Stadium ini dibagi menjadi stadium IIIA dan stadium IIIB. Pada Stadium IIIA, kanker telah menyebar ke sepertiga bagian bawah dari vagina tetapi masih belum ke dinding panggul. Pada stadium IIIB, kanker telah tumbuh menuju dinding panggul dan memblokir satu atau kedua saluran pembuangan ginjal Stadium ini biasanya diobati dengan radioterapi dan kemoterapi.

e. Stadium IV

Kanker serviks stadium IV adalah kanker yang paling parah. Kanker telah menyebar ke organ-organ tubuh di luar serviks dan rahim. Stadium ini dibagi menjadi stadium IVA dan stadium IVB. Pada stadium IVA, kanker telah menyebar ke organ-organ seperti kandung kemih dan dubur. Pada stadium IVB, kanker telah menyebar ke organ-organ tubuh yang sangat jauh, misalnya paru-paru. Pada stadium ini kanker diobati dengan pembedahan, radioterapi, kemoterapi atau kombinasi dari ketiganya.

B. Citra Digital

Sebuah citra dapat didefinisikan sebagai fungsi pada bidang dua dimensi $f(x, y)$, dimana (x, y) adalah koordinat spasial (bidang), dan amplitudo f pada sembarang pasangan koordinat (x, y) disebut intensitas dari citra pada titik tersebut. Ketika (x, y) dan nilai intensitas dari f adalah semuanya terbatas atau *discrete quantities* (dapat direpresentasikan secara numerik dengan nilai-nilai diskrit), citra tersebut dinamakan citra digital (Gonzales & Woods, 2002: 1).

Menurut Rinaldi Munir (2004: 18-19) pada umumnya citra digital berbentuk persegi panjang yang dimensi ukurannya dinyatakan sebagai lebar kali panjang atau disimbolkan dengan $P \times Q$. Citra digital yang berukuran $P \times Q$ biasanya direpresentasikan dengan matrik yang berukuran $P \times Q$, yang ditunjukkan sebagai berikut:

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, Q - 1) \\ f(1,0) & f(1,1) & \dots & f(1, Q - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(P - 1,0) & f(P - 1,1) & \dots & f(P - 1, Q - 1) \end{bmatrix} \quad (2.1)$$

Elemen pada matriks merupakan elemen citra digital yang disebut dengan piksel. Oleh karena itu, citra yang berukuran $P \times Q$ berarti mempunyai PQ buah piksel.

Secara umum citra digital terbagi menjadi tiga jenis, yaitu citra biner (*binary image*), citra keabuan (*grayscale*), dan citra warna (*true color*). Berikut ini penjelasan dari masing-masing jenis citra digital (Darma Putra, 2010: 40-44):

1. Citra biner merupakan citra yang paling sederhana. Citra ini hanya memiliki dua kemungkinan nilai piksel, yaitu hitam dan putih. Warna hitam ditunjukkan dengan nilai 0 dan warna putih ditunjukkan dengan nilai 1. Citra ini juga sering disebut citra B&W (*black and white*) atau citra monokrom.
2. *Grayscale* merupakan citra yang mampu menghasilkan gradasi warna abu-abu dari warna hitam hingga warna putih. Citra ini memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan). Nilai ini digunakan untuk menunjukkan nilai intensitas citra. Nilai 0 untuk warna hitam, nilai 256 untuk warna putih dan nilai antara 0 sampai dengan 256 untuk warna antara hitam dan putih (keabuan).
3. *True color* atau *rgb* merupakan perpaduan dari ketiga komponen utama pembentuk warna yang dikenal sebagai warna *rgb*. *Rgb* terdiri dari tiga warna dasar yaitu merah (*red*), hijau (*green*) dan biru (*blue*) yang berukuran sama.

C. *Preprocessing* Citra

Preprocessing citra merupakan proses awal yang dilakukan untuk memperbaiki kualitas citra dengan menggunakan teknik-teknik pengolahan citra.

Salah satu teknik pengolahan citra yang dapat digunakan dalam *preprocessing* adalah perbaikan citra (*image enhancement*). Perbaikan citra diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk, misalnya mengalami derau (*noise*), citra terlalu terang atau gelap, citra kurang tajam, kabur, dan sebagainya. Melalui teknik *preprocessing* inilah kualitas citra diperbaiki sehingga dapat digunakan untuk aplikasi lebih lanjut (Rinaldi Munir, 2004: 91).

Perbaikan citra dapat dilakukan dengan beberapa operasi. Operasi pada citra digital pada dasarnya adalah memanipulasi elemen-elemen matriks. Elemen matriks yang dimanipulasi dapat berupa elemen tunggal (piksel), sekumpulan elemen yang berdekatan, atau keseluruhan elemen matriks (Rinaldi Munir, 2004: 41). Berikut adalah operasi-operasi dasar pada pengolahan citra digital:

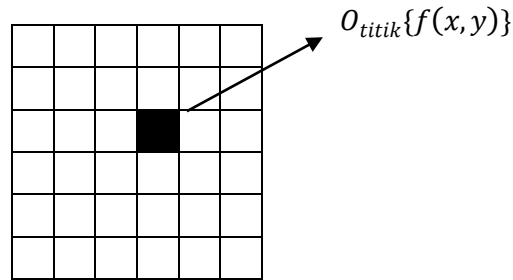
1. Operasi Titik

Operasi titik dilakukan pada piksel tunggal di dalam citra. Operasi ini terdiri dari pengaksesan piksel pada lokasi yang diberikan, memodifikasinya dengan operasi *linear* atau *non linear*, dan menempatkan nilai piksel baru pada lokasi yang bersesuaian di dalam citra yang baru. Operasi ini diulangi untuk keseluruhan piksel di dalam citra (Rinaldi Munir, 2004: 41).

Beberapa teknik perbaikan citra melalui operasi titik adalah *intensity adjustment*, *histogram equalization*, dan *thresholding*. Secara matematis, operasi titik dinyatakan sebagai berikut:

$$f_B(x, y) = O_{titik}\{f_A(x, y)\} \quad (2.2)$$

yang dalam hal ini f_A dan f_B masing-masing adalah citra masukan dan citra keluaran, O_{titik} dapat berupa operasi *linear* atau *non linear*. Maksud dari operasi *linear* adalah operasi yang dapat dinyatakan secara matematis sebagai persamaan *linear*. Sedangkan persamaan *non linear* merupakan kebalikan dari persamaan *linear*.



Gambar 2.1 Operasi Titik

2. Operasi Spasial

Operasi spasial atau lokal merupakan teknik perbaikan citra melalui penggunaan suatu kernel konvolusi 2-dimensi. Kernel adalah matrik persegi berukuran $(a \times b)$ dengan elemen-elemennya merupakan bilangan-bilangan yang membentuk pola tertentu. Ukuran kernel dapat berbeda-beda, namun pada umumnya berukuran kecil, seperti 2×2 , 3×3 , 5×5 , dan sebagainya. Kernel dilakukan pada proses konvolusi, oleh karena itu kernel disebut dengan *convolution window* (jendela konvolusi). Sedangkan konvolusi merupakan operator sentral pengolah citra dan telah digunakan secara luas pada berbagai piranti lunak pengolah citra (Darma Putra, 2010: 137-138).

Menurut Rinaldi Munir (2004: 46) operasi spasial menghasilkan citra keluaran yang intensitas suatu pikselnya bergantung pada piksel-piksel

disekitarnya. (Gambar 2.2). Formula operasi spasial ditunjukkan pada persamaan berikut:

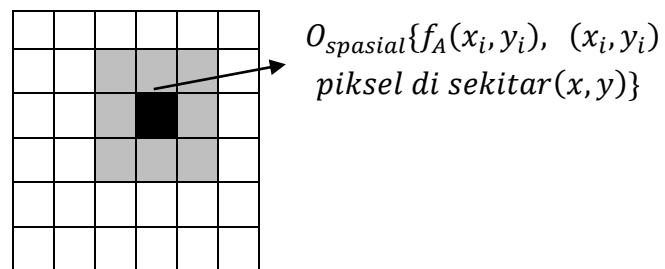
$$f_B(x, y) = O_{spasial}\{f_A(x_i, y_i); \quad (x_i, y_i) \in N(x, y)\} \quad (2.3)$$

dengan,

N = *neighborhood* adalah himpunan piksel-piksel yang berada disekitar (x, y)

f_A = citra masukan atau citra A

f_B = citra keluaran atau citra B



Gambar 2.2 Operasi Spasial

Beberapa metode yang menggunakan operasi spasial adalah *neighborhood averaging*, *median filtering* dan *high-pass filtering*.

a. *Neighborhood averaging*

Pada prinsipnya, filter yang digunakan dalam *neighborhood averaging* (filter mean) merupakan salah satu jenis filter *linear*, yang bekerja dengan cara mengganti intensitas suatu piksel dengan rata-rata dari nilai piksel tersebut dan nilai piksel tetangganya. Proses *neighborhood averaging* menghasilkan citra keluaran dengan pengurangan “transisi” ketajaman dalam intensitas. Karena *random noise* dan tepi biasanya berisi transisi ketajaman (*sharp*) dalam level intensitas, yang paling jelas dari proses ini adalah pengurangan *noise* dan pengaburan tepi (Gonzales & Woods, 2002: 119).

Perintah umum yang diberikan pada matlab untuk melakukan *neighborhood averaging* dengan kernel satuan berukuran (axb) adalah:

```
kernel=ones(a)/ab;  
  
J=imfilter(I, kernel);  
  
figure,imshow(J);
```

Contoh perintah matlab untuk melakukan *neighborhood averaging* dengan kernel satuan berukuran 3×3 adalah:

```
kernel=ones(3)/9;  
  
J=imfilter(I, kernel);  
  
figure,imshow(J);
```

b. *Median Filtering*

Median filtering atau filter median merupakan salah satu jenis filter *non linear* yang bekerja dengan mengganti nilai suatu piksel pada citra asal dengan nilai median dari piksel tersebut dan lingkungan tetangganya. Filter median mampu memberikan kemampuan pengurangan *noise* yang sangat bagus dengan memperhatikan *smoothing* yang sedikit, dibandingkan dengan filter mean pada ukuran yang sama. Pada bagian tertentu, filter median juga baik untuk menghilangkan *salt-and-pepper noise* karena sifat median yang menjauhi hitam dan putih (Gonzalez & Woods, 2002: 123).

Dalam filter median, dimensi kernel yang digunakan berukuran ganjil, seperti 3×3 , 5×5 , 7×7 , dan 9×9 . Semakin kecil ukuran kernel yang digunakan

akan semakin dapat meningkatkan kualitas suatu citra, karena dengan menggunakan kernel yang lebih kecil proses yang digunakan lebih detail, dan menghasilkan *smoothing* yang sedikit. Oleh karena itu, filter median lebih sering menggunakan dimensi kernel 3×3 (Murinto & Bachrudin Muchtar, 2012: 660).

Contoh 2.1. Tinjau jendela berupa kelompok piksel 3×3 (berbentuk kotak diarsir) pada sebuah citra pada Gambar 2.3(a). Piksel yang sedang diproses adalah yang mempunyai intensitas 35. Urutkan piksel-piksel tersebut mulai dari intensitas terkecil hingga intensitas terbesar:

9 10 10 10 **10** 10 11 12 35

Median dari kelompok tersebut adalah 10 (dicetak tebal). Titik tengah dari Gambar 2.3(a) yang bernilai 35 sekarang diganti dengan nilai median (10). Hasil dari filter median 3×3 diperlihatkan pada gambar Gambar 2.3(b). Jadi, filter median menghilangkan piksel yang sangat berbeda dengan piksel tetangganya (dianggap *noise*).

13	10	15	14	18
12	10	10	10	15
11	11	35	10	10
13	9	12	10	12
13	12	9	8	10

(a) Piksel bernilai 35 terkena *noise*

13	10	15	14	18
12	10	10	10	15
11	11	10	10	10
13	9	12	10	12
13	12	9	8	10

(b) 35 diganti dengan median dari kelompok piksel 3×3

Gambar 2.3 Penghilangan Derau (*Noise*) dengan Filter Median 3×3

Sebelum dilakukan *filtering*, citra digital yang sebelumnya berwarna harus diubah dulu dalam format *grayscale* karena yang difilter adalah intensitas dari warna hitam dan putih pada citra digital tersebut. Perintah matlab untuk mengkonversi citra *rgb* ke citra *grayscale* adalah:

```
R=rgb2gray(I);
```

Sedangkan perintah umum matlab untuk melakukan filter median berukuran (*axb*) adalah:

```
J=medfilt2(I,[a b]);  
  
figure,imshow(J);
```

Kode berikut digunakan untuk menampilkan *noise* pada citra dengan *salt-and-pepper noise* dengan parameter *noisy density* 0,2.

```
IN=imnoise(I,'salt & pepper',0.02);
```

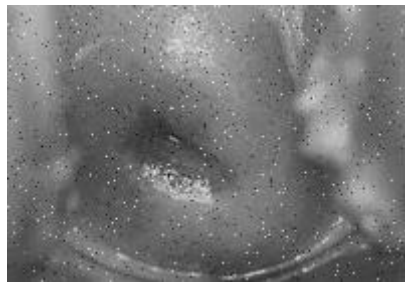
Contoh 2.2. Penggunaan teknik operasi spasial filter median pada citra *N-1.jpg* yang ditunjukkan pada Gambar 2.4. Gambar 2.4 (a) merupakan citra asli. Gambar 2.4 (b) merupakan citra asli dengan *noise salt-and-pepper* yang ditunjukkan dengan bercak-bercak atau *spike* pada gambar. Sedangkan Gambar 2.4 (c) menunjukkan citra *noise salt-and-pepper* yang telah dikonversi ke format citra *grayscale*. Hasil operasi spasial filter median 3×3 terhadap Gambar 2.4 (c) ditunjukkan oleh Gambar 2.4 (d). Tampak bahwa filter median mampu memberikan hasil yang baik untuk menghapus *noise* berupa bercak-bercak putih dengan tetap memperhatikan *smoothing* yang sedikit.



(a) Citra asli



(b) Citra dengan *noise salt-and-pepper*



(c) Citra *grayscale* dari gambar (b)



(d) Citra hasil filter median 3x3
dari gambar (c)

Gambar 2.4 Hasil *Noise*, *Grayscale* dan Filter Median pada Citra

Perintah yang diberikan pada matlab adalah:

```
I=imread('N-1.jpg');  
im=imnoise (I, 'salt & pepper', 0.02);  
R=rgb2gray(im);  
med=medfilt2 (R, [3 3]);  
figure,imshow(I);  
figure,imshow(im);  
figure,imshow(R);  
figure,imshow(med);
```

c. *High-pass filtering*

Sebagaimana pada proses pengolahan sinyal satu dimensi, *high-pass filtering* dua dimensi akan melewatkan komponen citra frekuensi tinggi dan

meredam komponen citra frekuensi rendah (Darma Putra, 2010: 145).

Contoh perintah untuk melakukan high-pass filtering:

```
hpf1=[ 1 -2 1;-2 5 -2; 1 -2 1];  
  
hpf2=[ 0 -1 0;-1 5 -1; 0 -1 0];  
  
hpf3=[-1 -1 -1;-1 9 -1;-1 -1 -1];  
  
J1=uint8(conv2(double(I),hpf1,'same'));  
  
J2=uint8(conv2(double(I),hpf2,'same'));  
  
J3=uint8(conv2(double(I),hpf3,'same'));
```

3. Operasi Transformasi

Teknik perbaikan citra dengan operasi transformasi berbeda dengan metode operasi titik dan operasi spasial. Operasi transformasi dilakukan dengan cara:

- a. mentransformasikan citra asal ke dalam domain yang sesuai dengan proses perbaikan citra,
- b. melakukan proses perbaikan citra pada domain tersebut,
- c. mengembalikan citra ke domain spasial untuk diproses lebih lanjut.

Salah satu metode transformasi yang sering digunakan dalam proses pengolahan citra adalah *Fast Fourier Transform* (FFT). Transformasi ini memindahkan informasi citra dari domain spasial ke dalam domain frekuensi, magnituda dan fasa.

D. Ekstraksi Fitur

Menurut Darma Putra (2010: 243) ekstraksi fitur merupakan proses pengolahan citra untuk mendapatkan fitur atau karakteristik suatu objek yang membedakannya dari objek yang lain. Fitur dari suatu citra menyatakan ciri dari permukaan objek dalam citra tersebut, ciri ini berisi informasi tentang komposisi struktur permukaan objek dan hubungan antara permukaan objek dengan lingkungan sekitarnya. Analisis fitur kemudian dilakukan untuk mengidentifikasi karakteristik suatu citra yang dapat digunakan untuk menginterpretasi citra.

Salah satu metode ekstraksi fitur yang telah banyak digunakan adalah metode GLCM (*Gray Level Co-occurrence Matrix*). GLCM adalah suatu matriks yang menggambarkan peluang munculnya pasangan dua buah piksel dengan intensitas tertentu dalam jarak dan arah tertentu pada suatu citra. Haralick, Shanmugam & Dinstein (1973: 619) mengusulkan berbagai jenis fitur yang dapat diekstraksi dari GLCM, beberapa diantaranya yaitu: *energy* atau *angular second moment*, *contrast*, *correlation*, *sum of squares* atau *variance*, *inverse difference moment*, *sum average*, *sum variance*, *sum entropy*, *entropy*, *difference variance*, dan *difference entropy*, *maximum probability*, *homogeneity* dan *dissimilarity*. Berikut adalah formula untuk masing-masing fitur ekstraksi:

1. *Energy*

Energy atau juga disebut *angular second moment* adalah ukuran keseragaman distribusi intensitas suatu citra. Nilai *energy* akan tinggi ketika citra memiliki keseragaman distribusi intensitas piksel yang baik. *Energy*

bernilai antara 0 sampai 1. *Energy* dapat diperoleh menggunakan persamaan berikut (Mohanaiah, Sathyanarayana & Gurukumar, 2013: 2):

$$Energy = \sum_{i=1}^L \sum_{j=1}^L p^2(i, j) \quad (2.4)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

2. *Entropy*

Fitur ini mengukur ketidakteraturan atau keacakan distribusi intensitas dari suatu citra. *Entropy* merupakan kebalikan dari *energy*, sehingga semakin baik distribusi intensitas suatu citra (semakin tinggi nilai *energy*) maka nilai *entropy* akan semakin rendah. Rumus *entropy* adalah sebagai berikut (Gadkari, 2004: 13):

$$Entropi = - \sum_{i=1}^L \sum_{j=1}^L p(i, j) \log p(i, j) \quad (2.5)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

3. *Contrast*

Contrast adalah fitur yang digunakan untuk mengukur perbedaan dari seperangkat piksel yang saling berdekatan pada citra. Semakin acak nilai intensitas suatu citra maka perbedaan antar pikselnya akan semakin tinggi (nilai *contrast*-nya tinggi). *Contrast* merepresentasikan tekstur suatu citra. Nilai *contrast* yang tinggi menyatakan tekstur citra yang kasar dan nilai *contrast*

yang rendah menyatakan tekstur citra yang halus. *Contrast* bernilai antara 0 sampai dengan 1. *Contrast* didefinisikan sebagai (Gadkari, 2004: 13):

$$Contrast = \sum_{i=1}^L \sum_{j=1}^L (i - j)^2 p(i, j) \quad (2.6)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

4. *Sum of squares (Variance)*

Variance adalah ukuran heterogenitas atau variasi elemen-elemen matriks GLCM. Fitur ini akan menempatkan bobot yang relatif tinggi pada elemen yang memiliki nilai yang berbeda dari nilai rata-rata $p(i, j)$. Berikut adalah rumus untuk menghitung *variance* (Albregtsen, 2008: 6):

$$Variance = \sum_{i=1}^L \sum_{j=1}^L (i - \mu)^2 p(i, j) \quad (2.7)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

μ = nilai rata-rata $p(i, j)$.

5. *Correlation*

Correlation adalah ukuran ketergantungan *linear* tingkat keabuan antar piksel satu sama lain. Nilai *correlation* akan tinggi ketika gambar memiliki homogenitas piksel yang sangat baik (Albregtsen, 2008: 6). *Correlation* didefinisikan sebagai berikut:

$$Correlation = \sum_{i=1}^L \sum_{j=1}^L \frac{\{(ij)p(i, j)\} - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.8)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

μ_x = nilai rata-rata elemen kolom (p_x) pada citra

μ_y = nilai rata-rata elemen baris (p_y) pada citra

σ_x = standar deviasi elemen kolom (p_x) pada citra

σ_y = standar deviasi elemen baris (p_y) pada citra

6. Inverse Difference Moment (IDM)

IDM adalah ukuran dari homogenitas lokal. Nilai IDM tinggi ketika tingkat keabuan (*grey level*) lokal seragam (Mohanaiah, Sathyanarayana & Gurukumar, 2013: 2). IDM dapat diperoleh menggunakan persamaan berikut:

$$IDM = \frac{\sum_{i=1}^L \sum_{j=1}^L p(i,j)}{1 + ((i-j)^2)} \quad (2.9)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

7. Sum average

Rumus *sum average* adalah sebagai berikut (Haralick, Shanmugam & Dinstein, 1973: 619):

$$SA = \sum_{k=2}^{2L} \{ k p_{x+y}(k) \} \quad (2.10)$$

dengan,

$$p_{x+y}(k) = \sum_{i=1}^L \sum_{j=1}^L p(i, j); i + j = k$$

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

8. *Sum entropy*

Sum entropy adalah ukuran dari jumlah perbedaan lokal pada citra. Nilai *sum entropy* tinggi ketika tingkat keabuan (*grey level*) lokal tidak seragam (Abouelatta, 2013: 217). Rumus *sum entropy* adalah sebagai berikut (Haralick, Shanmugam & Dinstein, 1973: 619):

$$SE = - \sum_{k=2}^{2L} p_{x+y}(k) \log\{ p_{x+y}(k) \} \quad (2.11)$$

dengan,

$$p_{x+y}(k) = \sum_{i=1}^L \sum_{j=1}^L p(i, j); i + j = k$$

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

9. *Sum variance*

Sum variance menunjukkan heterogenitas lokal dari suatu citra. Nilai *sum variance* tinggi ketika tingkat keabuan (*grey level*) lokal tidak seragam (Abouelatta, 2013: 217). Berikut adalah persamaan dari *sum variance* (Haralick, Shanmugam & Dinstein, 1973: 619):

$$SV = \sum_{k=2}^{2L} (k - SE)^2 p_{x+y}(k) \quad (2.12)$$

dengan,

$$p_{x+y}(k) = \sum_{i=1}^L \sum_{j=1}^L p(i, j); i + j = k$$

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

10. *Difference variance*

Difference variance menunjukkan selisih *variance* antara dua piksel yang berdekatan. Secara matematis rumus *sum variance* ditunjukkan oleh persamaan berikut (Haralick, Shanmugam & Dinstein, 1973: 619):

$$\begin{aligned} DV &= \text{varians dari } p_{x-y}(k) & (2.13) \\ &= \sum_{k=0}^{L-1} (k - \sum_{k=0}^{L-1} k p_{x-y}(k))^2 p_{x-y}(k) \end{aligned}$$

dengan,

$$p_{x-y}(k) = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j) ; |i - j| = k$$

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

11. *Difference entropy*

Difference entropy mengukur selisih *entropy* antara dua piksel yang berdekatan. Rumus *difference entropy* adalah sebagai berikut (Haralick, Shanmugam & Dinstein, 1973: 619):

$$DE = - \sum_{k=0}^{L-1} p_{x-y}(k) \log\{p_{x-y}(k)\} \quad (2.14)$$

dengan,

$$p_{x-y}(k) = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i, j) ; |i - j| = k$$

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

12. *Maximum probability*

Maximum probability menghitung tingkat keabuan (*gray level*) yang memiliki peluang maksimum di GLCM. *Maximum probability* dirumuskan sebagai berikut (Soh & Tsatsoulis, 1999: 781):

$$MP = \max_{i,j} p(i,j) \quad (2.15)$$

dengan,

$p(i,j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

13. Homogeneity

Homogeneity mengukur kehomogenan variasi intensitas dari suatu citra.

Homogeneity merupakan kebalikan dari *contrast*, sehingga nilai *homogeneity* akan tinggi ketika citra memiliki homogenitas yang sangat baik (piksel yang sangat mirip) atau nilai *contrast* yang rendah. *Homogeneity* bernilai antara 0 sampai 1. Rumus *homogeneity* yaitu (Kaur. J, Garg & Kaur. D, 2014: 2329):

$$H = \frac{\sum_{i=1}^L \sum_{j=1}^L p(i,j)}{1+|i-j|} \quad (2.16)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i,j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

14. Dissimilarity

Dissimilarity mirip dengan *contrast*, yaitu mengukur perbedaan tiap piksel. *Dissimilarity* akan menjadi tinggi apabila wilayah lokal memiliki *contrast* yang tinggi, dan sebaliknya. *Dissimilarity* bernilai antara 0 sampai dengan 1. Berikut adalah definisi dari *dissimilarity* (Soh & Tsatsoulis, 1999: 781):

$$D = \sum_{i=1}^L \sum_{j=1}^L |i-j| p(i,j) \quad (2.17)$$

dengan,

L = banyaknya tingkat keabuan dari citra

$p(i, j)$ = peluang tingkat keabuan pada baris ke- i dan kolom ke- j

E. *Neural Network* (NN)

Neural Network (NN) atau jaringan syaraf tiruan adalah sistem pemroses informasi yang terinspirasi oleh jaringan syaraf biologi. *Neural network* pertama kali diperkenalkan pada tahun 1943 oleh seorang ahli syaraf Warren McCulloch dan seorang ahli logika Walter Pitts. Hingga saat ini, *neural network* sudah banyak diaplikasikan dalam berbagai permasalahan, seperti pengenalan pola (misal huruf, angka, suara atau tanda tangan), *signal processing* untuk menekan *noise* dalam saluran telepon, peramalan, klasifikasi, optimisasi, dll (Siang, 2005: 5 & 31).

Neural network dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi (Fausett, 1994: 3), dengan asumsi bahwa:

- (a) Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*)
- (b) Sinyal dikirimkan diantara *neuron-neuron* melalui penghubung-penghubung
- (c) Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal
- (d) Untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi (biasanya *non linear*) yang dikenakan pada jumlahan *input* yang diterima.

Mengaktifkan *neural network* sama halnya mengaktifkan setiap *neuron* yang digunakan dalam jaringan. *Neuron* adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian *neural network*. Untuk mengaktifkan *neuron* digunakan fungsi tertentu yang disebut dengan fungsi aktivasi (Fausett, 1994:17).

1. Fungsi Aktivasi

Argumen dalam fungsi aktivasi adalah *in input* (kombinasi *linear input* dan bobotnya). Jika terdapat n buah *input* (x_1, x_2, \dots, x_n) yang masing-masing memiliki bobot v_1, v_2, \dots, v_n , maka fungsi aktivasinya adalah $f(in) = f(\sum_{i=1}^n x_i v_i)$. Terdapat beberapa fungsi aktivasi yang dapat digunakan sebagai pengaktif *neural network* (Fausett, 1994: 17-19), antara lain:

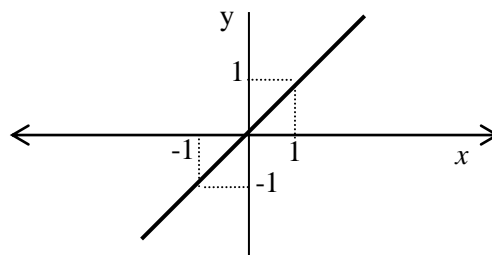
a. Fungsi Identitas (*Linear*)

Pada fungsi identitas, nilai *output* yang dihasilkan sama dengan nilai *inputnya* (Gambar 2.5). Fungsi identitas sering dipakai apabila menginginkan output berupa sembarang bilangan riil. Fungsi identitas dirumuskan sebagai:

$$y = f(x) = x, \quad x \in R \quad (2.18)$$

Dengan turunan pertama fungsinya adalah:

$$f'(x) = 1 \quad (2.19)$$



Gambar 2.5 Fungsi Identitas

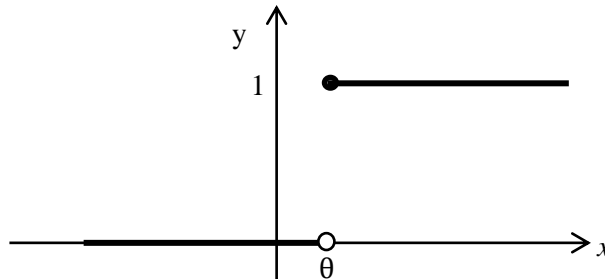
Pada Matlab R2010b, sintaks untuk fungsi aktivasi ini adalah *purelin*.

b. Fungsi Undak Biner (*Hard Limit*)

Fungsi undak biner digunakan untuk mengkonversikan *input* suatu variabel yang bernilai kontinu ke suatu *output* biner (0 atau 1). Fungsi undak biner (dengan *threshold* θ) dirumuskan sebagai berikut:

$$y = f(x) = \begin{cases} 1, & x \geq \theta \\ 0, & x < \theta \end{cases} \quad (2.20)$$

Grafik fungsi undak biner tampak pada Gambar 2.6 berikut:



Gambar 2.6 Fungsi Undak Biner

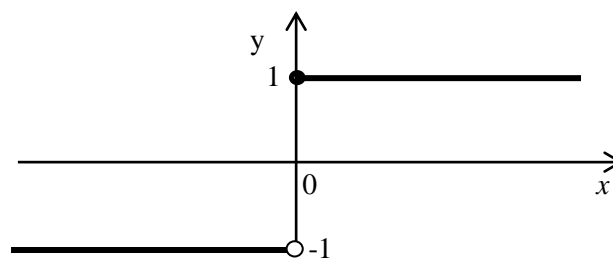
Pada Matlab R2010b, sintaks untuk fungsi aktivasi ini adalah *hardlim*.

c. Fungsi Bipolar (*Symmetric Hard Limit*)

Fungsi bipolar hampir sama dengan fungsi undak biner, perbedaannya terletak pada nilai *output* yang dihasilkan. Nilai *output* fungsi bipolar berupa 1 dan -1. Fungsi bipolar dirumuskan sebagai berikut:

$$y = f(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (2.21)$$

Grafik fungsi bipolar dapat dilihat pada Gambar 2.7 berikut :



Gambar 2.7 Fungsi Bipolar

Pada Matlab R2010b, sintaks untuk fungsi aktivasi ini adalah *hardlims*.

d. Fungsi Sigmoid Biner

Fungsi sigmoid biner sering digunakan karena nilai fungsinya yang terletak antara 0 dan 1 dan dapat diturunkan dengan mudah. Fungsi sigmoid biner dirumuskan sebagai berikut:

$$y = f(x) = \frac{1}{1+e^{-x}}, x \in R \quad (2.22)$$

Dengan turunan pertama fungsinya adalah:

$$f'(x) = \frac{e^{-x}}{1+2e^{-x}+e^{-2x}} \quad (2.23)$$

Pada Matlab R2010b, sintaks untuk fungsi aktivasi ini adalah *logsig*.

e. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, perbedaannya terletak pada rentang nilai *output*nya. Rentang nilai *output* fungsi sigmoid bipolar adalah -1 sampai 1. Fungsi sigmoid bipolar dirumuskan sebagai berikut:

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}}, x \in R \quad (2.24)$$

Dengan turunan pertama fungsinya adalah:

$$f'(x) = \frac{2e^{-x}}{1+2e^{-x}+e^{-2x}} \quad (2.25)$$

Pada Matlab R2010b, sintaks untuk fungsi aktivasi ini adalah *tansig*.

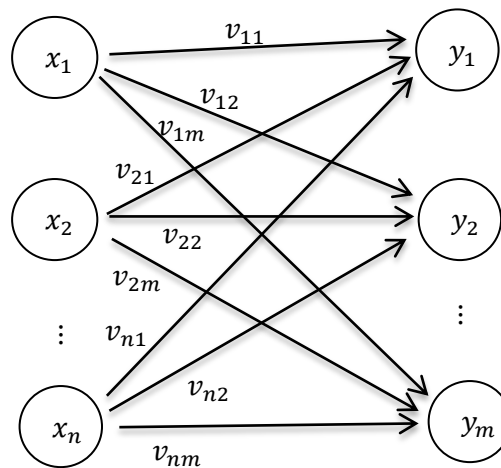
2. Arsitektur Jaringan

Arsitektur jaringan akan menentukan keberhasilan target yang dicapai karena tidak semua permasalahan dapat diselesaikan dengan arsitektur yang sama. Secara umum terdapat tiga jenis arsitektur jaringan atau arsitektur NN

yang sering dipakai, antara lain (Fausset, 1994: 12-15):

a. NN Layer Tunggal (*Single Layer Network*)

Dalam NN layer tunggal, sekumpulan *input neuron* dihubungkan langsung dengan sekumpulan *output*-nya. Semua unit *input* dihubungkan dengan semua unit *output*, meskipun dengan bobot yang berbeda-beda. Tidak ada unit *input* yang dihubungkan dengan unit *input* lainnya. Demikian pula pada unit *output*, tidak ada hubungan antar unit-unitnya.

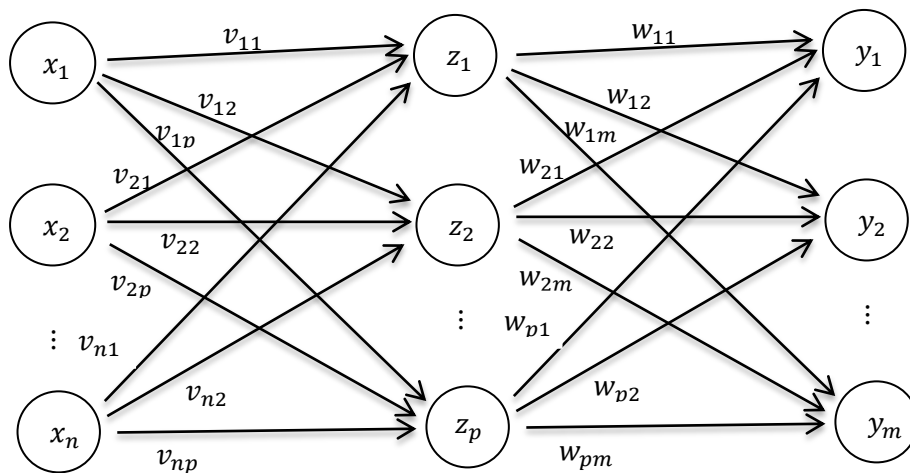


Lapisan *Input* Bobot Lapisan *Output*

Gambar 2.8 Arsitektur NN Layer Tunggal

b. NN Layer Jamak (*Multi-layer Network*)

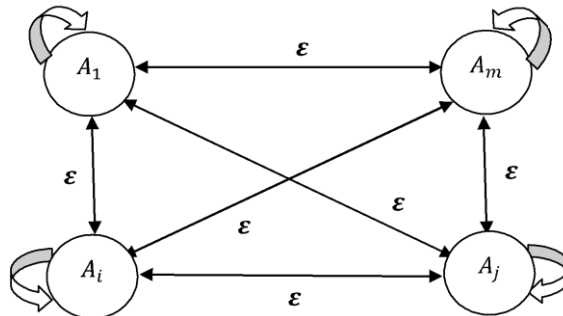
Dalam NN layer jamak, selain unit *input* dan *output*, ada unit-unit lain pada layer tertentu yang disebut layer tersembunyi. Dimungkinkan pula ada beberapa layer tersembunyi. Sama seperti pada unit *input* dan *output*, unit-unit dalam satu layer tidak saling berhubungan. NN layer jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layer tunggal, meskipun kadangkala proses pembelajarannya lebih lama.



Gambar 2.9 Arsitektur NN Layer Jamak

c. NN Layer Kompetitif (*Competitive Layer Network*)

NN dengan layer kompetitif memiliki bentuk yang berbeda dengan NN layer tunggal dan layer jamak. Pada NN dengan layer kompetitif, antar *neuron* dapat saling dihubungkan.



Gambar 2.10 Arsitektur NN Layer Kompetitif

3. Algoritma Pembelajaran

Algoritma pembelajaran dalam NN didefinisikan sebagai metode untuk menentukan nilai-nilai bobot penghubung yang tepat dalam mengirimkan suatu informasi antar *neuron* (Siang, 2005: 3).

Siang (2005: 28-29) menyatakan bahwa terdapat 2 metode pembelajaran *neural network* berdasarkan strategi pembelajarannya, yaitu:

a. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

Dalam pembelajaran ini, tidak terdapat target *output* yang akan mengarahkan proses pembelajaran. Dalam pembelajarannya, perubahan bobot jaringan dilakukan berdasarkan parameter tertentu dan jaringan dimodifikasi menurut ukuran parameter tersebut. Tujuan pembelajaran ini adalah untuk mengelompokkan unit-unit yang hampir sama ke dalam suatu area tertentu sesuai nilai input yang diberikan. Algoritma pembelajaran *neural network* yang termasuk *unsupervised learning* antara lain: *Kohonen Self- Organizing Maps*, *Learning Vextor Quantization*, dan *Counterpropagation*.

b. Pembelajaran Terawasi (*Supervised Learning*)

Dalam pembelajaran terawasi terdapat sejumlah data (*input – target output*) yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan. Pasangan data tersebut berfungsi untuk melatih jaringan hingga diperoleh bentuk yang terbaik. Pasangan data yang diberikan akan memberikan informasi yang jelas tentang bagaimana sistem harus mengubah dirinya untuk meningkatkan kinerjanya.

Pada setiap pembelajaran, suatu *input* diberikan ke jaringan. Jaringan akan memproses dan mengeluarkan *output*. Selisih antara *output* jaringan dengan target (*output* yang diinginkan) merupakan kesalahan (*error*) yang terjadi. Jaringan akan memodifikasi bobot sesuai kesalahan tersebut. Jaringan

perceptron, ADALINE, dan *backpropagation* adalah algoritma pembelajaran yang termasuk dalam pembelajaran terawasi. Namun yang sering digunakan dalam proses klasifikasi adalah algoritma pembelajaran *backpropagation*.

F. Algoritma *Backpropagation*

Algoritma *backpropagation* atau propagasi balik merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan layar jamak. Algoritma ini didasarkan pada pembelajaran koreksi kesalahan (*error*) yang diperoleh dengan membandingkan *output* dan target yang diinginkan. Kemudian *error* yang diperoleh ini digunakan untuk mengubah nilai bobot-bobot antar *neuron* dalam arah mundur (*backward*). Sebelumnya, tahap perambatan maju (*forward*) harus dikerjakan terlebih dahulu untuk mendapatkan nilai *output* (Haykin, 1999: 178).

Backpropagation adalah metode *neural network* yang populer karena kehandalannya dalam menangani masalah pengenalan pola-pola kompleks dengan baik. Algoritma *backpropagation* dibagi menjadi dua bagian, yaitu algoritma pembelajaran dan algoritma pengujian. Seperti halnya model NN yang lain, *backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pembelajaran serta kemampuan jaringan untuk memberi respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pembelajaran (Siang, 2005: 97).

Algoritma pembelajaran *backpropagation* meliputi tiga tahap yaitu *feedforward* atau perambatan maju, *backpropagation* atau propagasi balik, dan

perubahan bobot. Pada saat *feedforward*, *input* dihitung maju mulai dari lapisan *input* sampai lapisan *output* menggunakan fungsi aktivasi yang sudah ditentukan. Fungsi aktivasi yang dipakai pada algoritma *backpropagation* harus memenuhi beberapa syarat, yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Fungsi-fungsi yang memenuhi ketiga syarat tersebut adalah fungsi sigmoid biner, fungsi identitas, dan fungsi sigmoid bipolar. Selanjutnya tahap *backpropagation* atau propagasi balik, pada tahap ini kesalahan yang terjadi dihitung menggunakan selisih antara nilai *output* jaringan dengan target yang diinginkan. Kesalahan tersebut dipropagasikan balik untuk mendapatkan *error* yang minimal, dimulai dari lapisan *output* sampai ke lapisan *input*. Tahap ketiga adalah perubahan bobot dan bias. Tahap ini dilakukan untuk menurunkan kesalahan yang terjadi (Siang, 2005: 99-100).

Langkah-langkah algoritma pembelajaran *backpropagation* adalah sebagai berikut (Fausett, 1994: 294-297):

Langkah 0 : Inisialisasi bobot-bobot (ambil bobot awal menggunakan nilai *random* yang cukup kecil, biasanya diinisialisasi secara *random* antara -0.5 sampai 0.5, atau -1 sampai dengan 1, atau interval yang lainnya.

Langkah 1 : Menetapkan parameter pembelajaran seperti maksimum *epoch*, target *error*, dan *learning rate* (α),
Inisialisasi *epoch* = 0, dan MSE = 1.

Langkah 2 : Kerjakan langkah-langkah berikut selama (*Epoch* < Maksimum *Epoch*) dan (MSE < Target *Error*).

Fase I : *Feedforward*

Langkah 3 : Setiap *neuron input* ($x_i, i = 1, 2, 2, \dots, n$) menerima sinyal *input* x_i dan meneruskan sinyal tersebut ke semua *neuron* yang ada di lapisan atasnya (lapisan tersembunyi).

Langkah 4 : Setiap *neuron* pada lapisan tersembunyi ($z_j, j = 1, 2, \dots, p$) menjumlahkan sinyal-sinyal *input* terbobot

$$z_in_j = v_{0j} + \sum_{i=1}^n v_{ij} x_i \quad (2.26)$$

Menggunakan fungsi aktivasi untuk menghitung sinyal *outputnya*,

$$z_j = f(z_in_j) \quad (2.27)$$

Mengirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit *output*).

Langkah 5 : Setiap unit *output* ($y_k, k = 1, 2, \dots, m$) menjumlahkan sinyal-sinyal *input* terbobot.

$$y_in_k = w_{0k} + \sum_{j=1}^p w_{jk} z_j \quad (2.28)$$

Menggunakan fungsi aktivasi untuk menghitung sinyal *output-nya*,

$$y_k = f(y_in_k) \quad (2.29)$$

Mengirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit *output*).

Fase II : *Backpropagation*

Langkah 6 : Setiap unit *output* ($y_k, k = 1, 2, \dots, m$) menerima target pola yang berhubungan dengan pola *input* pembelajaran, kemudian hitung informasi *error-nya*:

$$\delta_k = (t_k - y_k) f'(y_in_k) \quad (2.30)$$

δ_k adalah unit *error* yang akan dipakai dalam perubahan bobot lapisan di bawahnya. Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki bobot w_j) dengan laju pembelajaran α .

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j \quad (2.31)$$

Hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki bobot w_0):

$$\Delta w_{0k} = \alpha \cdot \delta_k \quad (2.32)$$

Mengirimkan sinyal tersebut ke unit-unit pada lapisan sebelumnya.

Langkah 7 : Setiap unit tersembunyi ($z_j, j = 1, 2, \dots, p$) menjumlahkan hasil perubahan *input*-nya dari unit-unit yang berada di lapisan atasnya

$$\delta_{in_j} = \sum_{k=1}^p \delta_k w_{jk} \quad (2.33)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (2.34)$$

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

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (2.35)$$

Hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}):

$$\Delta v_{0j} = \alpha \cdot \delta_j \quad (2.36)$$

Fase III : Perubahan bobot dan bias

Langkah 8 : Setiap unit *output* (y) memperbaiki bias dan bobot-bobotnya

$$(j = 1, 2, \dots, p)$$

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.37)$$

Setiap unit tersembunyi ($z_j, j = 1, 2, \dots, p$) memperbaiki bias dan bobot-bobotnya ($i = 1, 2, \dots, n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.38)$$

Langkah 9 : Uji syarat berhenti.

Setelah pembelajaran selesai dilakukan, jaringan dapat digunakan untuk pengujian atau pengenalan pola. Dalam hal ini, hanya *feedforward* atau perambatan maju (Fase I: langkah 4 dan 5) yang digunakan untuk menentukan *output* jaringan.

G. Ketepatan Hasil Klasifikasi

Pengujian ketepatan klasifikasi dilakukan setelah sistem selesai melakukan pembelajaran. Hal ini dilakukan untuk menguji sejauh mana sistem dapat memberikan ketepatan dalam hasil klasifikasi menggunakan data lain (data baru yang tidak digunakan selama pembelajaran). Untuk menghitung ketepatan hasil klasifikasi sistem, dapat digunakan tiga ukuran statistik. Tiga ukuran statistik tersebut adalah sensitivitas, spesifisitas, dan akurasi. Ketiga ukuran statistik tersebut digunakan untuk mengetahui seberapa bagus dan terpercaya hasil klasifikasi yang telah dilakukan. Kemungkinan yang dapat terjadi dalam hasil klasifikasi ditunjukkan dalam Tabel 2.1 berikut:

Tabel 2.1 Hasil Klasifikasi (Sharma & Mukharje, 2014: 22):

<i>Test</i>	<i>Disease</i>		<i>Sum</i>
	<i>Present</i>	<i>Absent</i>	
<i>Positive</i>	True Positive (TP)	False Positive (FP)	TP + FP
<i>Negative</i>	False Negative (FN)	True Negative (TN)	FN + TN
<i>Sum</i>	TP + FN	FP + TN	TP + FN + FP + TN

Keterangan:

- TP : pasien memiliki penyakit dan hasil klasifikasi menyatakan pasien memiliki penyakit,
- FN : pasien memiliki penyakit dan hasil klasifikasi menyatakan pasien tidak memiliki penyakit,
- FP : pasien tidak memiliki penyakit dan hasil klasifikasi menyatakan pasien memiliki penyakit,
- TN : pasien tidak memiliki penyakit dan hasil klasifikasi menyatakan pasien tidak memiliki penyakit.

1. Sensitivitas

Akobeng (2006: 339) menyatakan bahwa sensitivitas adalah proporsi pasien berpenyakit yang bereaksi positif terhadap pengujian penyakit tersebut. Sensitivitas hanya mampu memberitahu seberapa baik klasifikasi tersebut untuk mengidentifikasi penyakit ketika hanya melihat pasien berpenyakit. Rumus sensitivitas adalah sebagai berikut (Sharma & Mukharje, 2014: 22):

$$Sensitivitas = \frac{TP}{(TP+FN)} \times 100\% \quad (2.39)$$

b. Spesitifitas

Spesitifitas adalah proporsi pasien tanpa penyakit yang bereaksi negatif terhadap pengujian penyakit tersebut. Spesitifitas hanya mampu memberitahu seberapa baik klasifikasi tersebut untuk mengidentifikasi penyakit ketika hanya melihat pasien yang tidak memiliki penyakit (Akobeng, 2006: 339). Spesitifitas dapat diperoleh menggunakan rumus berikut (Sharma & Mukharje, 2014: 22):

$$\text{Spesitifitas} = \frac{TN}{(TN+FP)} \times 100\% \quad (2.40)$$

c. Akurasi

Akurasi adalah kemampuan tes untuk mengidentifikasi dengan tepat kondisi pasien sebenarnya (hasil positif maupun hasil negatif yang tepat). Akurasi sistem dapat diperoleh dengan menghitung hasil klasifikasi yang tepat dibagi total keseluruhan data. Secara matematis dapat dinyatakan dengan:

$$\text{Akurasi} = \frac{\text{Banyaknya klasifikasi yang tepat}}{\text{Total keseluruhan data}} \times 100\% \quad (2.41)$$