

# Penggunaan Jaringan Fungsi Basis Radial yang Dimodifikasi Untuk Analisis Efek Benturan Pada Logam Lurus

Luky Agus Hermanto, ST., MT.<sup>1)</sup>

1) Jurusan Teknik Komputer Institut Teknologi Adhi Tama Surabaya  
Email : luky\_a\_hermanto@yahoo.com

**Abstrak** - Benturan dengan kekuatan tertentu pada logam lurus akan mengakibatkan adanya perubahan bentuk dan perubahan kelurusan logam. Perubahan kelurusan ini mengakibatkan adanya kekurangan sempurna bentuk dan fungsi, sehingga diperlukan suatu tindakan pembenahan ke posisi semula (meluruskan) untuk mendapatkan bentuk dan fungsi yang sesuai keadaan semula. Tindakan ini akan semakin tepat apabila dilakukan analisa terlebih dahulu. Karena dengan adanya analisa akan didapatkan informasi tentang kondisi logam yang bisa disimpan dalam bentuk computer database dan digital videography yang dapat digunakan teknisi untuk strategi pembenahan kondisi logam.

Jaringan fungsi basis radial yang dimodifikasi, yaitu jaringan fungsi basis radial yang proses pembelajarannya menggunakan algoritma new EM-based training. Dengan algoritma new EM-based training diharapkan proses mendapatkan pembobot dalam pembelajaran menjadi lebih cepat.

**Kata kunci** : Analisis Benturan, Jaringan Fungsi Basis Radial, Pembenahan kondisi logam

## 1. PENDAHULUAN

Kondisi logam yang lurus atau rata cenderung sering digunakan dalam bidang mekanikal. Akan tetapi dalam perjalanan penggunaan logam tersebut sering kali terjadi hal yang tidak dapat dihindari. Benturan dapat menyebabkan fungsi logam menjadi menurun, sampai tidak dapat digunakan, bahkan dari segi estetika juga dapat menurunkan tingkat penampilan. Materi atau bahan logam memiliki tingkat pergeseran elastisitas berbeda-beda. Dalam beberapa kasus elastisitas ini masih dapat ditoleransi sehingga logam yang mengalami benturan dan menyebabkan bengkok dapat dilakukan perbaikan atau perawatan untuk mengembalikan ke posisi semula. Analisa ini sangat diperlukan untuk teknisi, dimana didapatkan informasi yang bisa disimpan dalam bentuk computer database dan digital videography yang dapat digunakan teknisi untuk strategi pembenahan kondisi.

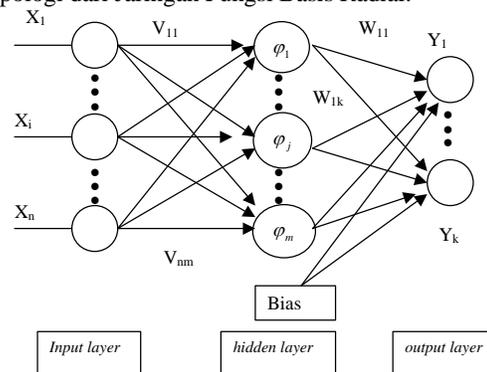
Jaringan fungsi basis radial merupakan modifikasi dari jaringan feedforward. Pada tahun 2001 tahun 1996, U. Pietruschka dan R. Brause telah mengusulkan penggunaan jaringan fungsi basis radial untuk proses kontrol pada industri karet. Jaringan fungsi basis

radial juga telah digunakan untuk pengenalan suara perintah yang terbatas (Juan Carlos Macera, Nipesh Shah, 2001), pengenalan wajah (Meng Joo Er, Shiqian Wu, Hock Lye Toh, 2002), konversi suara spectra (Carlo Drioli, 1900) dan registrasi elastik dari gambar medis (Mike Fornefeet, Karl Rohr, H. Siegfried Stiehl, 1999).

## 2. TINJAUAN PUSTAKA

### 2.1. Jaringan Fungsi Basis Radial

Jaringan Fungsi Basis Radial adalah jaringan yang memiliki dua lapisan yang bentuk output unit – nya berupa kombinasi linier dari fungsi basis yang dihitung dari hidden unit [6]. Berikut ini adalah topologi dari Jaringan Fungsi Basis Radial.



Gambar 1: Topologi Jaringan Fungsi Basis Radial

Pelatihan jaringan jaringan fungsi basis radial terdiri atas dua tahap :

#### 1. Tahap Clustering Data

Pada tahap ini, data di-cluster atau dikelompokkan berdasarkan kedekatan tertentu, misalnya kedekatan warna antara dua pixel, kedekatan antara dua titik dan sebagainya. Penentuan cluster dengan sendirinya akan menghasilkan center atau pusat dari kelompok data. Jumlah cluster menentukan jumlah hidden unit yang dipakai.

Dalam menentukan center, ada dua cara yang bisa dipakai. Cara yang paling mudah ialah dengan menentukan center secara acak dari kelompok data. Cara yang lebih sulit, tetapi lebih baik adalah dengan algoritma clustering. Tahapan ini merupakan tahapan yang bersifat unsupervised.

2. Tahap pembaharuan pembobot

Jaringan syaraf tiruan menyimpan pengetahuannya dalam bobot *neuron-neuron*-nya. Pelatihan tahap berikutnya berfungsi mendapatkan nilai bobot *neuron-neuron*-nya. Pada tahap ini, ada serangkaian perhitungan yang diperlukan untuk memperbaharui pembobot. Pada tahap ini juga dibutuhkan training beserta targetnya. Tahap ini merupakan tahap yang bersifat *supervised*. Langkah-langkah tahapan ini sebagai berikut:

- a. Menentukan sinyal *input ke hidden layer* dan menghitung nilai fungsi aktivasinya pada tiap *hidden layer*. Dengan menggunakan rumus

$$\varphi(\|x_m - t_j\|) = \exp(-\|x_m - t_j\|^2 / \sigma^2) \dots(1)$$

dimana

m = 1,2,3... sesuai dengan jumlah *training pattern*

j = 1,2,3... sesuai dengan jumlah *hidden unit*

X = vektor *input*

t = vektor data yang dianggap *center*

- b. Menyusun matriks, *Gaussian* dari perhitungan langkah 1.

$$G = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \dots & \varphi_{1C} \\ \varphi_{21} & \varphi_{22} & \dots & \varphi_{2C} \\ \dots & \dots & \dots & \dots \\ \varphi_{M1} & \varphi_{M2} & \dots & \varphi_{MC} \end{bmatrix} \dots\dots\dots(2)$$

dimana

M vektor *input* ke M

C *hidden unit* ke C

Kolom terakhir biasanya diisi bias.

- c. Menghitung bobot baru (W) dengan mengalikan *pseudoinverse* dari matrik G, dengan vektor target (d) dari *data training*. Rumusnya:

$$W = (G^T G)^{-1} G^T d \dots\dots\dots(3)$$

- d. Untuk menghitung nilai *output* dari jaringan, digunakan rumus:

$$y(x) = \sum_{i=1}^2 w_i G(\|x - t_i\|) + b \dots\dots\dots(4)$$

dimana b nilai bobot bias  $G(x) = \varphi(x)$

**2.2. Algoritma Ekspektasi Maksimisasi**

Algoritma *Expectation Maximization* (EM) adalah suatu metoda umum untuk estimasi maximum likelihood dari parameter untuk data tidak lengkap. Dalam rumus asli algoritma EM, model dasar yang dinyatakan dalam hubungan Y dan  $Z = \{Y, V\}$ , dimana Y disebut data yang tidak lengkap, V data yang hilang, dan himpunan kombinasi Z disebut data yang lengkap. Tujuan dari algoritma adalah menemukan himpunan parameter, yang mana akan memaksimalkan kemungkinan dari nilai yang diobservasi Y. Maksimisasi dilaksanakan dalam kaitan dengan distribusi gabungan Y dan Z.

Pada *training 2 layer feedforward networks* dengan menggunakan pendekatan *EM-based*. Dengan mempertimbangkan jaringan fungsi basis radial dengan *Gaussian unit* yang mendekati satu dimensi pemetaan  $g(x) : R \rightarrow R$  sebagai berikut:

$$\bar{g}(x) = \sum_{i=1}^G \lambda_i o_i(x) \dots\dots\dots(5)$$

dimana i adalah indeks dari unit-unit fungsi basis radial,  $\lambda_i$  amplitudo dan  $o_i(x)$  adalah fungsi aktivasi dari setiap unit, yang diberikan oleh :

$$o_i(x) = \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right) \dots\dots\dots(6)$$

Masalah *training* berisi dalam estimasi amplitudo,  $\lambda_i$ , pusat  $\mu_i$  dan *variance*  $\sigma_i^2$  dari model fungsi basis radial yang diberikan oleh himpunan *input* dan berkorespondensi *noisy output*  $(x_k, y_k)$ . Observasi *noisy* mempunyai karakteristik model berikut:

$$y_k = \sum_i g_i(x_k; \theta_i) + e_k \dots\dots\dots(7)$$

dimana  $g_i(x_k, \theta_i) = \lambda_i o_i(x_k)$ , dan biasanya, mengasumsikan bahwa  $e_k$  adalah *zero-mean white Gaussian Noisy* dari *variance*  $\sigma^2$ . Maka *log-likelihood* dari parameter diberikan oleh:

$$L(\theta; x; y) = K - \frac{1}{2\sigma^2} \sum_k \left( y_k - \sum_i g_i(x_k, \theta_i) \right)^2 \dots\dots\dots(8)$$

dimana K adalah konstanta yang dapat diabaikan untuk proses estimasi,  $\theta = \{\theta_1, \dots, \theta_G\}$  dan  $\theta_i = (\lambda_i, \mu_i, \sigma_i)$ .

Prosedur perhitungan yang lebih efisien untuk memperoleh estimasi *maximum likelihood* adalah berbasis pada algoritma EM. Sebuah pilihan baik untuk data hilang atau data tidak lengkap dari ini algoritma perlu untuk menyederhanakan maksimisasi dari *likelihood*. Bagian seleksi yang berguna untuk masalah ini yang telah diusulkan oleh Feder dan Weinstein (1998): data lengkap diperoleh dari dekomposisi setiap observasi komponen sinyal G, menurut

$$z_{k,i} = g_i(x_k, \theta_i) + e_{k,i}, i = 1..G \dots\dots\dots(9)$$

dimana residual  $e_{k,i}$  diperoleh dari dekomposisi *residual total*  $e_k = y_k - \sum_i g_i(x_k; \theta_i)$  kedalam komponen G, yaitu

$$e_{k,i} = t_i e_k \quad i = 1, \dots, G \quad \forall k \dots\dots\dots(10)$$

Menurut Feder dan Weinstein (1998), ini menunjukkan bahwa *decoupling variables*  $t_i$  dapat sembarang konstanta, batasan jumlahan 1. Sebuah dekomposisi dari persamaan residual antara semua *neuron-neuron* diusulkan

$$t_i = \frac{1}{G}, i = 1 \dots G \dots\dots\dots(11)$$

yang terakhir persamaan (11) mendekomposisi residual algoritma EM ini, untuk *training 2 layer feedforward networks*, dapat disimpulkan sebagai berikut:

E-step: for  $i=1 \dots G$  hitung

$$\hat{z}_{k,i} = g_i(x_k, \theta_i) + t_i \left( y_k - \sum_{j=1}^G g_j(x_k; \theta_j) \right) \dots(12)$$

**M-Step: for  $i=1 \dots G$  evaluasi**

$$\{\lambda_i, \mu_i, \sigma_i^2\} = \arg \min_{\theta_i} \sum_k (\hat{z}_{k,i} - g_i(x_k; \theta_i))^2 \quad (13)$$

Dimana iterasi notasi indeks telah dihilangkan. Pencatatan masalah dari *training global* sebuah jaringan fungsi basis radial dengan  $G$  neuron telah dipecahkan kedalam  $G$  masalah yang lebih sederhana dari *training* sebuah *neuron* tunggal.

### 2.3. Analisis Efek Benturan

Dengan menggunakan foto X-Ray kondisi dari suatu efek benturan terhadap logam dapat di ketahui, sehingga dapat ditentukan diagnosa berkenaan dengan afek kerusakan logam, perencanaan perawatan, evaluasi ramalan dan hasil perawatan. Sumber kesalahan yang utama di dalam analisa efek benturan pada logam adalah meliputi film radiografis perbesaran, mengusut, mengukur, merekam dan hal menonjol identifikasi. Analisa efek benturan pada logam *computer-aided* adalah lebih cepat di dalam analisa dan didapatnya data dibanding metoda konvensional. Banyak program penguji efek benturan pada logam telah dikembangkan untuk melaksanakan analisa efek benturan pada logam secara *computer-aided digitizing*.

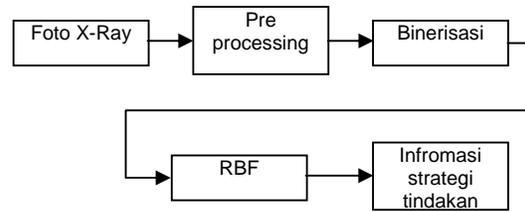


Gambar 2:

Hasil foto X-Ray sebuah logam setelah terjadi benturan

## 3. PEMBAHASAN

Secara umum perangkat lunak untuk analisis efek benturan mempunyai struktur sebagai berikut:



Gambar 3:

Struktur perangkat lunak untuk analisa efek benturan

### 3.1. Preprocessing

Pada tahap ini dilakukan proses proses pendeteksian tepi. Tujuan dari proses ini adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek didalam citra.

Untuk mendeteksi tepi gambar di atas dilakukan dengan menggunakan algoritma *Sobel* dbrikut program *sobel* yang digunakan :

```

procedure
TFormutama.SobelProcess1Click(Sender:
TObject);
var sx:array [-1..1,-1..1] of Integer;
    sy:array [-1..1,-1..1] of Integer;
    jx,jy: Double;
    wd,x,y,k,l,i,j,lx,ly,w,r,g,b: Integer;
begin
  sx[-1,-1]:=-1; sx[0,-1]:=-2; sx[1,-1]:=-1;
  sx[-1,0]:=0; sx[0,0]:=0; sx[1,0]:=0;
  sx[-1,1]:=1; sx[0,1]:=2; sx[1,1]:=1;

  sy[-1,-1]:=-1; sy[0,-1]:=0; sy[1,-1]:=1;
  sy[-1,0]:=-2; sy[0,0]:=0; sy[1,0]:=2;
  sy[-1,1]:=-1; sy[0,1]:=0; sy[1,1]:=1;

  lx:=Asli.Width;
  ly:=Asli.Height;

  for i:=0 to lx do
  begin
    for j:=0 to ly do
    begin
      jx:=0; jy:=0;
      for k:=-1 to 1 do
      begin
        for l:=-1 to 1 do
        begin
          x:= i + k;
          y:= j + l;
          If (x >= 0) And (x <= lx) And (y >=
0) And (y <= ly) Then
            begin
              r:=GetRValue(Asli.Canvas.Pixels[x,y]);
              g:=GetRValue(Asli.Canvas.Pixels[x,y]);
              b:=GetRValue(Asli.Canvas.Pixels[x,y]);
              w:=(r+g+b) div 3;
              jx:=jx + w * sx[k,l];
              jy:=jy + w * sy[k,l];
            end;
          end;
        end;
      end;
      wd:=round(sqrt(sqr(jx)+sqr(jy)));
      if wd > 255 then wd:=255
      else if wd < 0 then wd:=0;
      Image1.Canvas.Pixels[x,y]:=RGB(wd,wd,wd);
    end;
  end;
  palsu:=Image1.Picture.Bitmap;
  
```

```
Image2.Picture.Bitmap:=palsu;
end;
end;
```

Setelah dilakukan proses deteksi tepi langkah selanjutnya dalam *preprocessing* ini adalah menghilangkan *noise* dari gambar. Proses menghilangkan noise pada pada gambar sefalometri dilakukan dengan menggunakan *median filtering*. Berikut ini subprogram *median filtering* yang digunakan untuk menghilangkan *noise*:

```
Begin
  w := FormCitra.Image.Picture.Width;
  h := FormCitra.Image.Picture.Height;
  if
    (FormCitra.Image.Picture.Bitmap.PixelFormat =
    pf8bit)
  then
    begin
      SetLength(Ki, w, h);
      SetLength(Ko, w, h);
      for y := 0 to h-1 do
        begin
          PC :=
          FormCitra.Image.Picture.Bitmap.ScanLine[y];
          PH :=
          FormHasil.Image.Picture.Bitmap.ScanLine[y];
          for x := 0 to w-1 do
            begin
              Ki[x, y] := PC[x];
              Ko[x, y] := PH[x];
            end;
          end;
          if (RadioButton9.Checked) then
            M := 1;
          if (RadioButton25.Checked) then
            M := 2;
          if (RadioButton49.Checked) then
            M := 3;
          if (RadioButton121.Checked) then
            M := 5;
          for x := M to w-1-M do
            for y := M to h-1-M do
              begin
                for u := -M to M do
                  for v := -M to M do
                    Urutan[((2*M+1)*u+v+((2*M+1)*(2*M+1)+1) div 2]
                      := Ki[x-u,y-v];
                    for i := 2 to (2*M+1)*(2*M+1) do
                      for j := (2*M+1)*(2*M+1) downto
                    i do
                      if (Urutan[j] < Urutan[j-1])
                    then
                      begin
                        temp := Urutan[j];
                        Urutan[j] := Urutan[j-1];
                        Urutan[j-1] := temp;
                      end;
                    Ko[x,y] :=
                    Urutan[((2*M+1)*(2*M+1)+1) div 2];
                    end;
                    for y := 0 to h-1 do
                      begin
                        PH :=
                        FormHasil.Image.Picture.Bitmap.ScanLine[y];
                        for x := 0 to w-1 do
                          PH[x] := Ko[x, y];
                        end;
                        Ki := nil;
                        Ko := nil;
                      end;
                    End;
                    End;
```

Setelah setelah menghilangkan noise langkah selanjutnya adalah menyederhanakan gambar sefalometri menjadi gambar hitam putih dengan menggunakan operasi *threshold*. Berikut ini adalah program *threshold* yang digunakan untuk menyederhanakan gambar:

```
procedure
TFormrbf.ButtonThresholdClick(Sender:
TObject);
  var x,y,temp,tengah:Integer;
      palsu:TBitmap;
  begin
    palsu:=TBitmap.Create;
    palsu.Assign(Asli);
    palsu.Dormant; palsu.FreeImage;
    tengah:=170;
    try
      for y:= 0 to palsu.Height -1 do
        begin
          P := palsu.ScanLine[y];
          x:= 0;
          repeat
            temp:=p[x];
            if temp < tengah then
              begin
                p[x]:= 0;
              end else
                begin
                  p[x]:= 255;
                end;
                inc(x,1);
                until x>= 3*(Image1.Width);
            end;
          finally
            Asli.Assign(palsu);
            Image1.Picture.Assign(Asli);
            ButtonThreshold.Enabled:=false;
          end;
        end;
      end;
```

### 3.2. Ekstraksi ( Binerisasi )

Setelah objek-objek yang ada berhasil disederhanakan pada tahap sebelumnya, tahap berikutnya adalah mengekstraksi ciri-ciri tertentu dari setiap objek. Pada tahap ini gambar dibagi menjadi baris sebanyak M dan kolom sebanyak N yang akan didapatkan matrik berukuran M x N.

Pada tahap ini dilakukan pembentukan matrik-matrik yang nantinya digunakan pada tahap klasifikasi. Matrik ini nanti akan digunakan untuk proses training. Matriks-matriks ini diperoleh dengan cara mengganti warna hitam dengan angka 1 dan putih dengan angka 0. Dari matrik-matrik ini nantinya digunakan untuk identifikasi bagian-bagian dari sefalimetri.

Berikut ini adalah program yang digunakan untuk ekstraksi:

```
for j:=0 to n-1 do
Begin
  for i:=0 to m-1 do
    Begin
      A[i,j]:=Asli.Canvas.Pixels[i,j];
    End;
  end;
end;
```

### 3.3. Klasifikasi

Sebelum melakukan analisis efek benturan maka matriks-matriks hasil ekstraksi yang telah diperoleh

akan dilakukan proses *training*. Pada proses *training* akan diperoleh ciri-ciri titik yang digunakan pada proses analisis efek benturan. Ciri-ciri titik tersebut akan disimpan dalam bentuk matriks pembobot. Proses *training* dilakukan dengan menggunakan jaringan syaraf tiruan jaringan fungsi basis radial yang dimodifikasi. Modifikasi jaringan fungsi basis radial dilakukan agar proses pencarian pembobot agar lebih cepat. Modifikasi ini dilakukan dengan menambahkan algoritma *Expectation Maximization*. Setelah diperoleh ciri-ciri titik maka akan bisa dilakukan prediksi sudut-sudut dan jarak antar titik tertentu yang nantinya digunakan pada analisis efek benturan.

#### 4. HASIL DAN PEMBAHASAN

Dengan menggunakan pembobot hasil training yang diterapkan pada specimen (image) test diperoleh besaran sudut-sudut dari posisi setelah terjadi benturan terhadap posisi awal yang diambil dari penentuan garis lurus sebagai kondisi semula/awal normal. Dengan kombinasi beberapa ordinat terhadap kondisi normal didapatkan tingkatan/klasifikasi nilai efek benturan. Dan ini bisa diartikan sebagai suatu tingkatan tindakan yang dapat dilakukan untuk proses penanganan lebih lanjut dari logam tersebut.

#### 5. KESIMPULAN

Dari hasil analisa menggunakan data test spesimen didapatkan nilai beragam sesuai tingkatan efek benturan. Klasifikasi ini dapat untuk menentukan tindakan apa yang seharusnya dilakukan untuk penanganan kondisi logam untuk kembali ke posisi normal/lurus.

#### DAFTAR REFERENSI

- [1] Carlo Drioli, *Radial Basis Function Networks For Conversion Of Sound Spectra*, Workshop on Digital Audio Effects, 1999.
- [2] Dan W. Patterson, *Artificial Neural Network Theory and Applications*, John Wiley and Sons, Inc. 1995.
- [3] Juan Carlos Macera and Nipesh Shah, *Voice Command Recognition Using Radial Basis Function Neural Networks*, University of Nevada Reno, 2001.
- [4] Mike Fornefett, Kalr Rohr, and H. Siegfried Stiehl, *Elastic Registration of Medical Images Using Radial Basis Functions with Compact Support*, IEEE Computer Society, pp 402-407, 1999.
- [5] Marcelino Lazar, Ignacio Santamaria and Carlos Pantaleon, *A New EM-Based Training Algorithm For RBF Networks*, Elsevier.com/locate, Neural Networks 16, 2002
- [6] Meng Joo Er, Shiqian Wu, Juwei Lu and Hock Lye Toh, *Face Recognition With Radial Basis Function (RBF) Neural Networks*, Transaction On Neural Networks, 2002.
- [7] Moody, J. and C.J. Darken, "Fast learning in networks of locally tuned processing units", *Neural Computations*, vol. 1, No. 2, pp. 281-294, 1989
- [8] Neil Alldrin, Andrew Smith and Doug Turnbull, *Classifying Facial Expression With Radial Basis Function Networks Using Gradient Descent and K-means*, University of California, 2003.
- [9] Santoro, M, et al, *Comparison of measurements made on digital and Plaster Models*, American Journal of Orthodontics and Dentofacial Orthopedics, Volume 124, 2002
- [10] Setiawan, Kuswara., *Paradigma Sistem Cerdas*, Bayumedia Publishing, Malang, 2003
- [11] Su Hongtao, David Dagan Feng and Zhao Rong-chun, *Face Recognition Using Multi-Feature and Radial Basis Function Networks*, Australian Computer Society, 2003.