

ANALISA JARINGAN SARAF TIRUAN DENGAN METODE *BACKPROPAGATION* UNTUK MENGETAHUI LOYALITAS KARYAWAN

Jasmir, S.Kom, M.Kom
Dosen tetap STIKOM Dinamika Bangsa Jambi

Abstrak

Karyawan atau tenaga kerja adalah bagian dari sistem dalam suatu perusahaan atau instansi yang juga merupakan salah satu yang berperan penting dalam mencapai visi misi dan tujuan dari sebuah perusahaan atau institusi. Hasil yang dicapai dari perusahaan pun tergantung juga dari loyalitas dan dedikasi dari karyawan atau tenaga kerja. Loyalitas kerja merupakan suatu interaksi terpadu secara serasi dari tiga faktor esensial, yaitu optimalisasi kemampuan, manajemen dan sumber daya yang dimiliki oleh tenaga kerja.

1.Latar Belakang Penelitian

Ilmu pengetahuan dan teknologi terus berkembang tak henti-hentinya, Seiring dengan itu pula hal-hal yang menarik untuk dibahas pun menjadi tantangan dalam mengambil keputusan, loyalitas kerja misalnya, hal tersebut akan menjadi vital bagi pimpinan dan karyawan, dan juga bagi instansi untuk mengamati kualitas dan loyalitas kerja karyawan atau karyawannya. Dalam mengetahui loyalitas tersebut dapat dibuat permodelan menggunakan komputer dengan berbasis pengetahuan menggunakan Jaringan Saraf Tiruan. Jaringan saraf tiruan merupakan salah satu sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Jaringan saraf tiruan mampu mengenali kegiatan dengan berbasis pada data masa lalu. Data masa lalu akan dipelajari oleh jaringan saraf tiruan sehingga mempunyai kemampuan untuk memberi keputusan terhadap data yang belum pernah dipelajari.

Penelitian ini membahas tentang perhitungan untuk mengetahui loyalitas kerja dengan menggunakan metode *BackPropagation* pada jaringan saraf tiruan. Dengan kerja sistem komputer lebih cepat, teliti dan akurat bila dibandingkan kerja manusia, hal ini mendorong percepatan pengembangan teknologi AI (*Artificial Intelligence*). Oleh sebab itu penulis tertarik untuk melakukan penelitian dengan judul “**Analisa Jaringan Saraf Tiruan dengan Metode *BackPropagation* untuk Mengetahui Loyalitas Karyawan**”.

2.Perumusan Masalah

Adapun rumusan masalahnya yaitu Bagaimana penerapan jaringan saraf tiruan dengan metoda *BackPropagation* untuk mengetahui loyalitas karyawan ?

3.Batasan Masalah

Adapun batasan masalah dalam penelitian ini mencakup :

1. Menganalisa jaringan saraf tiruan dengan metode *BackPropagation*.
2. Peneliti hanya membahas proses loyalitas karyawan.

4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Membuat model dengan *BackPropagation* untuk mengetahui loyalitas karyawan.
2. Membantu pihak terkait dalam hal loyalitas karyawan.

5. Tinjauan Pustaka

5.1. Jaringan Saraf Tiruan (JST)

Adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran / pelatihan pada otak manusia yang di implementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran / pelatihan. Jaringan saraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang di dasarkan atas asumsi sebagai berikut :

1. Pemrosesan informasi terjadi pada elemen sederhana yang di sebut *neuron*.
2. Isyarat mengalir di antara sel saraf/*neuron* melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan/mengalikan isyarat yang dikirim melaluinya.
4. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.

Valluru B. Rao dan Hayagriva V. Rao (1993) mendefinisikan : “Jaringan saraf sebagai sebuah kelompok pengolahan elemen dalam suatu kelompok yang khusus membuat perhitungan sendiri dan memberikan hasilnya kepada kelompok kedua atau berikutnya”.

Setiap sub-kelompok menurut gilirannya harus membuat perhitungan sendiri dan memberikan hasilnya untuk subgrup atau kelompok yang belum melakukan perhitungan. Pada akhirnya sebuah kelompok dari satu atau beberapa pengolahan elemen tersebut menghasilkan keluaran (*output*) dari jaringan.

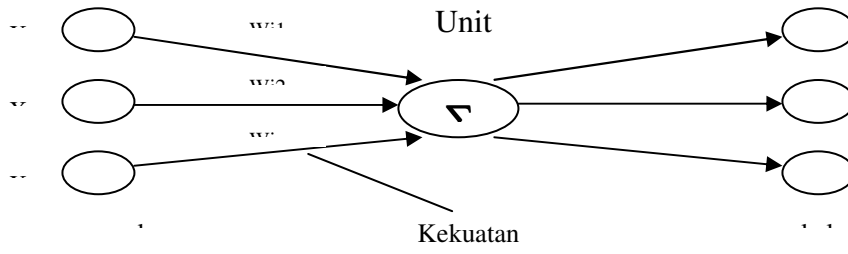
Setiap pengolahan elemen membuat perhitungan berdasarkan pada jumlah masukan (*input*). Sebuah kelompok pengolahan elemen di sebut *layer* atau lapisan dalam jaringan. Lapisan pertama adalah *input* dan yang terakhir adalah *output*. Lapisan di antara lapisan *input* dan *output* disebut dengan lapisan tersembunyi (*hidden layer*).

Jaringan saraf tiruan merupakan suatu bentuk arsitektur yang terdistribusi paralel dengan sejumlah besar *node* dan hubungan antar *node* tersebut. Tiap titik hubungan dari suatu *node* ke *node* yang lain mempunyai harga yang di asosiasikan dengan bobot. Setiap *node* memiliki suatu nilai yang diasosiasikan sebagai nilai aktivasi *node*.

Salah satu organisasi yang dikenal dan sering digunakan dalam paradigma jaringan saraf buatan adalah Perambatan Galat Mundur (*backpropagation*).

5.2 Model Neuron Jaringan Saraf Tiruan (JST)

Seperti halnya otak manusia, jaringan saraf juga terdiri dari beberapa neuron dan ada hubungan antara *neuron-neuron* tersebut. Beberapa *neuron* akan mentransformasikan informasi yang diterimanya melalui sambungan keluaran menuju *neuron-neuron* yang lain. Neuron ini dimodelkan dari penyederhanaan sel saraf manusia yang sebenarnya



Gambar 2.1 Struktur Unit Jaringan Saraf Tiruan

Gambar 2.1 memperlihatkan struktur unit pengolah jaringan saraf tiruan. Pada sisi sebelah kiri terlihat beberapa masukan yang menuju ke unit pengolah yang masing-masing datang dari unit-unit yang berbeda $x(n)$. Setiap sambungan mempunyai kekuatan hubungan terkait (bobot) yang disimbolkan dengan $w(n)$. Kekuatan hubungan pada tiap sambungan akan disesuaikan selama pelatihan sehingga pada akhir pelatihan dihasilkan jaringan saraf dengan bobot-bobot yang mantap. Unit pengolah akan membentuk penjumlahan berbobot dari tiap masukannya dan menggunakan fungsi ambang non linier (fungsi aktivasi) untuk menghitung keluarannya. Hasil perhitungan akan dikirimkan melalui hubungan keluaran seperti tampak pada gambar sisi sebelah kanan.

Sebagian jaringan saraf melakukan penyesuaian bobot-bobotnya selama menjalani prosedur latihan. Pelatihan dapat berupa pelatihan terbimbing (*supervised training*) di mana diperlukan pasangan masukan sasaran untuk tiap pola dilatihkan. Jenis kedua adalah pelatihan tak terbimbing (*unsupervised training*). Pada metode ini, penyesuaian bobot (sebagai tanggapan terhadap masukan) tak perlu disertai sasaran. Dalam pelatihan tek terbimbing, jaringan mengklasifikasikan pola-pola yang ada berdasarkan kategori kesamaan.

5.3. Fungsi Aktivasi

Fungsi aktivasi harus mempunyai beberapa karakteristik penting yaitu *continue*, terdefinisi, dan tidak monoton. Fungsi aktivasi merupakan bagian penting dalam tahap perhitungan keluaran dari suatu algoritma. Beberapa fungsi aktivasi yang digunakan dalam jaringan saraf tiruan adalah :

1. Fungsi Identitas

$$f(x)=x, \text{ untuk semua } x \dots\dots\dots(2.1)$$

2. Fungsi undak biner (dengan batas ambang)

$$f(x) = \begin{cases} 1 \text{ untuk } x \geq \theta \\ 0 \text{ untuk } x < \theta \end{cases} \dots\dots\dots(2.2)$$

3. Fungsi sigmoid

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \dots\dots\dots(2.3)$$

$$f'(x) = \sigma f(x)[1 - f(x)]$$

Dengan : σ : konstanta

4. Fungsi sigmoid bipolar

$$g(x) = 2 f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)} \dots\dots\dots(2.4)$$

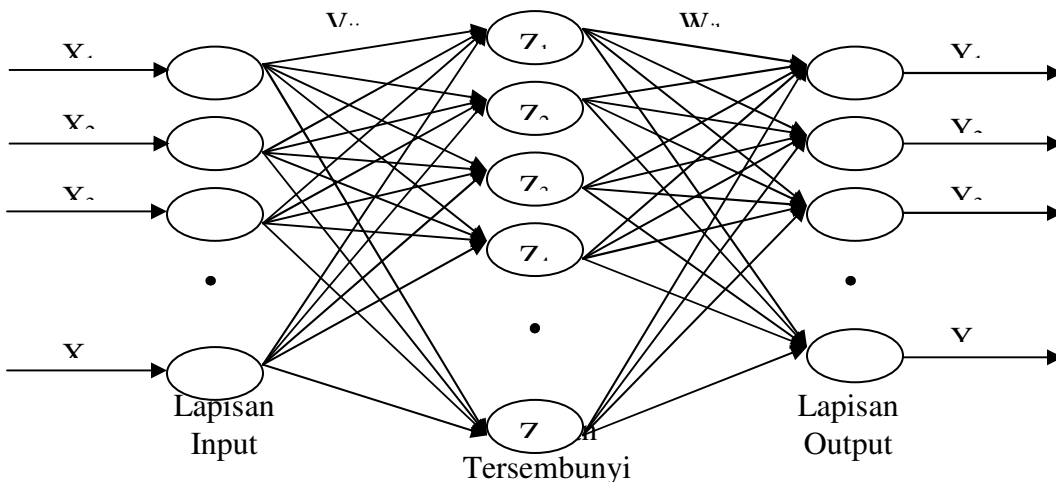
$$= \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)}$$

$$g'(x) = \frac{\sigma}{2} [1 + g(x)][1 - g(x)]$$

Dengan : σ : konstanta

5.4. Metode BackPropagation

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, *neuron-neuron* diaktifkan dengan menggunakan fungsi aktivasi sigmoid.



Gambar 2.5 Tiga Lapisan Backpropagation

5.4.1. Algoritma Backpropagation

Pada dasarnya, pelatihan dengan metode *Backpropagation* terdiri atas tiga langkah yaitu :

- a. Data dimasukkan ke *input* jaringan (*feedforward*)
- b. Perhitungan dan propagasi balik dari *error output* yang bersangkutan (*backward*).
- c. Pembaharuan (*adjustment*) bobot dan bias.

Saat umpan maju (*feedforward*), setiap unit *input* (X_i) akan menerima sinyal *input* dan akan diteruskan sinyal tersebut ke semua unit pada lapisan tersembunyi. Setiap unit tersembunyi menjumlahkan sinyal-sinyal *input* terbobot dan digunakan fungsi aktivasi untuk menghasilkan sinyal inputnya (Z_j) dan dikirimkan ke semua unit *output*. Pada semua unit *output* dilakukan penjumlahan atas sinyal-sinyal input terbobot yang diterimanya dan digunakan fungsi aktivasi untuk menghasilkan sinyal *output*nya (Y_k).

Setelah perambatan maju selesai dikerjakan maka jaringan melakukan perambatan galat mundur. Sinyal-sinyal *output* (Y_k) akan menerima target pola yang berhubungan dengan pola *input* pembelajaran dengan menghitung target error atau galat (δ_k), mengoreksi bobot-bobotnya (Δw_{jk}) dan mengoreksi bobot biasnya (Δw_{0k}). Kirimkan target *error* atau galat (δ_k) ke unit-unit yang ada dilapisan bawahnya. Tiap-tiap unit tersembunyi (Z_j) menjumlahkan delta *input*nya (δ_k) dan menghasilkan target *error* atau galat digunakan fungsi aktivasi (δ_j). Kemudian dihitung koreksi bobot-bobotnya (ΔV_{ij}) untuk memperbaiki nilai (V_{ij}) dan koreksi bobot biasnya (ΔV_{0j}) untuk memperbaiki nilai (V_{0j}). Pada tiap-tiap unit *output* (Y_k) diperbaiki bias dan bobotnya W_{jk} (baru) dan W_{0k} (baru) dan pada tiap-tiap unit tersembunyi (Z_j) diperbaiki bias dan bobotnya V_{ij} (baru) dan V_{0j} (baru). Setelah langkah-langkah diatas selesai dilakukan dan akan berhenti sampai tes kondisi berhenti.

Notasi-notasi yang akan digunakan pada algoritma *Backpropagation* yaitu:

- x Data untuk *inputan*
 $x = (x_1, \dots, x_i, \dots, x_n)$
- t Target untuk *output*
- α *Learning rate* yaitu parameter yang mengontrol perubahan bobot selama pelatihan. Jika *Learning rate* besar, jaringan semakin cepat belajar tetapi hasilnya kurang akurat. *Learning rate* biasanya dipilih antara 0 dan 1.
- Z_j *Hidden unit* (lapisan tersembunyi) ke j. Sinyal *input* pada Z_j dilambangkan dengan z_{inj} . Sinyal *output* (aktivasi) untuk Z_j dilambangkan dengan z_j .
- V_{0j} Bias untuk *hidden unit* ke j.
- V_{ij} Bobot antara unit *input* ke i dan *hidden unit* ke j.
- Y_k Unit *output* ke k. Sinyal *input* ke Y_k dilambangkan dengan y_{ink} . Sinyal *output* (aktivasi) untuk Y_k dilambangkan dengan y_k .
- W_{0k} Bias untuk unit *output* ke k.
- W_{jk} Bobot antara *hidden unit* ke j dan unit *output* ke k.
- δ_k Faktor koreksi *error* untuk bobot w_{jk} .
- δ_j Faktor koreksi *error* untuk bobot v_{ij} .

Secara detail langkah-langkah pelatihan *BackPropagation* sebagai berikut :

- Langkah 1 : Inisialisasi bobot dan bias. Baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka disekitar 0 dan 1 atau -1 (bias positif atau negatif).
- Langkah 2 : Jika *stop condition* belum terpenuhi maka jalankan langkah 2-10.
- Langkah 3 : Untuk setiap *data training*, lakukan langkah 4-9
 - a. Umpan maju (*feedforward*) :
- Langkah 4 : Tiap-tiap unit input ($x_i, i = 1, 2, 3, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada diatasnya (lapisan tersembunyi).
- Langkah 5 : Tiap-tiap *hidden unit* ($z_j, j = 1, \dots, p$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya.

$$z_{inj} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \dots\dots\dots(2.5)$$

Dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan.

$$z_j = f(z_{inj}) \dots\dots\dots(2.6)$$

Langkah 6 : Setiap unit *output* (Y_k , $k = 1, \dots, m$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasnya.

$$y_{in_k} = W_{0k} + \sum_{j=1}^p Z_j W_{jk} \quad \dots\dots\dots(2.7)$$

Dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit *output* yang bersangkutan.

$$y_k = f(y_{in_k}) \quad \dots\dots\dots(2.8)$$

Lalu mengirim sinyal *output* ini ke seluruh unit pada unit *output*.

b. Propagasi *error* (*backpropagation of error*) :

Langkah 7 : Setiap *output* (Y_k , $k = 1, \dots, m$) menerima suatu target *pattern* (*desired output*) yang sesuai dengan *input training pattern* untuk menghitung kesalahan (*error*) antara target dengan *output* yang dihasilkan jaringan.

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad \dots\dots\dots(2.9)$$

Kemudian hitung koreksi *error* (yang nantinya akan digunakan untuk memperoleh nilai W_{jk}).

$$\Delta W_{jk} = \alpha \delta_k z_j \quad \dots\dots\dots(2.10)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai W_{0k}).

$$\Delta W_{0k} = \alpha \delta_k \quad \dots\dots\dots(2.11)$$

Kirimkan ini ke unit-unit yang berada pada lapisan dibawahnya.

Langkah 8 : Tiap-tiap unit tersembunyi (z_j , $j = 1, \dots, p$) menjumlahkan delta *inputnya* (dari unit-unit yang berada pada lapisan diatasnya).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad \dots\dots\dots(2.12)$$

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

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad \dots\dots\dots(2.13)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai V_{ij}).

$$\Delta V_{ij} = \alpha \delta_j X_i \quad \dots\dots\dots(2.14)$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai V_{0j}).

$$\Delta V_{0j} = \alpha \delta_j \quad \dots\dots\dots(2.15)$$

c. Perbaharui (adjuatment) bobot dan bias :

Langkah 9 : Tiap-tiap unit *output* (Y_k , $k = 1, \dots, m$) memperbaiki bias dan bobotnya ($j = 0, 1, 2, \dots, p$).

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad \dots\dots\dots(2.16)$$

$$W_{0k}(\text{baru}) = W_{0k}(\text{lama}) + \Delta W_{0k} \quad \dots\dots\dots(2.17)$$

Tiap-tiap unit tersembunyi (z_j , $j = 1, \dots, p$) memperbaiki bias dan bobotnya ($i = 0, 1, 2, \dots, n$).

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad \dots\dots\dots(2.18)$$

$$V_{0j}(\text{baru}) = V_{0j}(\text{lama}) + \Delta V_{0j} \dots\dots\dots(2.19)$$

Langkah 10 : Tes Kondisi berhenti (*stop condition*).

5.4.2.Kondisi Berhenti Pada Metode *BackPropagation*

Jika tes kondisi telah terpenuhi, pelatihan jaringan dapat dihentikan. Untuk menentukan kondisi berhenti terdapat dua cara yang biasa di pakai yaitu sebagai berikut.

1. Dengan membatasi iterasi yang diinginkan dilakukan. Misalnya, jaringan akan dilatih sampai dengan iterasi ke-500, yang dimaksud satu iterasi adalah perulangan langkah ke-4 sampai langkah ke-9 untuk semua *training* data yang ada.
2. Dengan membatasi *error*. Pada metode *backpropagation*, dipakai metode *Mean Square Error* untuk menghitung rata-rata antara *output* yang dikehendaki, pada *training* data dengan *output* yang dihasilkan oleh jaringan. Misalnya, *error* telah mencapai 0,01 (1%) maka pelatihan dihentikan. Cara memeriksa *stopping condition* dengan *Mean Square Error* adalah sebagai berikut :
 - a. Dengan bobot yang ada saat itu, lakukan set umpan maju (langkah ke-4 sampai dengan langkah ke-6) dimana *inputnya* diambil dari *input training set*. Langkah tersebut dilakukan untuk semua data *training* yang ada.
 - b. Kemudian, dicari selisih antara target *output* (T_k) dengan *output* jaringan (Y_k) dan diimplementasikan pada persamaan *Mean Square Error*. Jika terdapat m *training* data, maka:

$$\text{Mean Square Error} = E = 0,5 * \{ (t_{k1} - y_{k1})^2 + (t_{k2} - y_{k2})^2 + \dots + (t_{km} - y_{km})^2 \} \quad (2.20)$$

Setelah pelatihan selesai, jika kemudian jaringan diberi *input*, jaringan akan dapat menghasilkan *output* seperti yang diharapkan. Cara mendapatkan output adalah dengan mengimplementasikan metode *backpropagation*, tetapi hanya pada bagian umpan majunya.

5.4.3.Pengujian Data

Setelah proses pembelajaran dilakukan berdasarkan langkah-langkah algoritma *backpropagation*, maka diperoleh nilai bobot akhir dan nilai bias akhir yaitu dari iterasi terakhir dengan nilai *error* atau galat 0,01. Nilai bobot dan bias inilah yang akan digunakan dalam melakukan pengujian terhadap data dengan perambatan maju (*forward backpropagation*). Adapun langkah-langkah pengujian data adalah sebagai berikut:

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

Langkah 2 : Tiap-tiap *hidden unit* ($z_j, j = 1,\dots,p$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya.

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots(2.21)$$

Dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan.

$$z_j = f (z_in_j) \dots\dots\dots(2.22)$$

Langkah 3 : Setiap unit *output* ($Y_k, k = 1,\dots,m$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasnya.

$$y_{in_k} = w_{0k} + \sum_{j=1}^p z_j w_{jk} \dots\dots\dots(2.23)$$

Dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit *output* yang bersangkutan.

$$y_k = f(y_{in_k}) \dots\dots\dots(2.24)$$

Lalu mengirim sinyal *output* ini ke seluruh unit pada unit-unit *output*.

5.5. Penilaian Pelaksanaan Pekerjaan Untuk Mengetahui Loyalitas Kerja Karyawan

Loyalitas kerja merupakan suatu interaksi terpadu secara serasi dari tiga faktor esensial, yaitu optimalisasi kemampuan, manajemen dan sumber daya yang dimiliki oleh tenaga kerja. Agar bisa mengetahui tingkat loyalitas kerja karyawan maka dilakukan penilaian pelaksanaan pekerjaan dalam jangka waktu penilaian yang telah ditentukan.

Penilaian pelaksanaan pekerjaan pada Karyawan Negeri Sipil diatur dalam Peraturan Pemerintah Republik Indonesia Nomor 10 Tahun 1979. Unsur-unsur dalam Penilaian Pelaksanaan Pekerjaan menurut PP Nomor 10 Tahun 1979 tersebut yang kemudian dijadikan objek dalam penelitian ini adalah:

- a. Kesetiaan
Kesetiaan adalah ketaatan dan pengabdian kepada Pancasila, Undang-Undang Dasar 1945, Negara dan Pemerintah.
- b. Prestasi Kerja
Prestasi kerja adalah hasil kerja yang dicapai oleh seorang Karyawan Negeri Sipil dalam melaksanakan tugasnya yang dibebankan kepadanya.
- b. Tanggung Jawab
Tanggung Jawab adalah kesanggupan seorang Karyawan Negeri Sipil menyelesaikan pekerjaan yang diserahkan kepadanya dengan sebaik-baiknya dan tepat pada waktunya serta berani memikul resiko atas keputusan yang diambilnya atau tindakan yang dilakukannya.
- c. Ketaatan
Ketaatan adalah kesanggupan seorang Karyawan Negeri Sipil untuk mentaati segala peraturan perundang-undangan dan peraturan kedinasan yang berlaku, mentaati perintah kedinasan yang diberikan oleh atasan yang berwenang, serta kesanggupan untuk tidak melanggar larangan yang ditentukan.
- d. Kejujuran
Kejujuran adalah ketulusan hati seorang Karyawan Negeri Sipil dalam melaksanakan tugas dan kemampuan untuk tidak menyalah gunakan wewenang yang diberikan kepadanya.
- e. Kerja sama
Kerja sama adalah kemampuan seorang Karyawan Negeri Sipil untuk bekerja bersama-sama dengan orang lain dalam menyelesaikan sesuatu tugas yang ditentukan, sehingga mempunyai daya guna dan hasil guna yang sebesar-besarnya.
- f. Prakarsa

Prakarsa adalah kemampuan seorang Karyawan Negeri Sipil untuk mengambil keputusan, langkah-langkah, atau melaksanakan sesuatu tindakan yang diperlukan dalam melaksanakan tugas pokok tanpa menunggu perintah dari atasan.

g. **Kepemimpinan**

Kepemimpinan adalah kemampuan seorang Karyawan Negeri Sipil untuk meyakinkan orang lain sehingga dapat dikerahkan secara maksimal untuk melaksanakan tugas pokok.

3. Analisa dan Hasil

3.1. Proses Pembelajaran

Jaringan saraf tiruan mencoba untuk mensimulasikan kemampuan otak manusia untuk belajar. Jaringan saraf tiruan juga tersusun atas *neuron-neuron* dan dendrite. Tidak seperti model biologis, jaringan saraf tiruan memiliki struktur yang tidak dapat diubah, dibangun oleh sejumlah *neuron* dan memiliki nilai tertentu yang menunjukkan seberapa besar koneksi antara *neuron* (bobot).

Perubahan yang terjadi selama proses pembelajaran adalah perubahan nilai bobot. Pada saat pembelajaran dilakukan pada input yang berbeda, maka nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Apabila nilai ini telah tercapai mengindikasikan bahwa tiap-tiap *input* telah berhubungan dengan *output* yang diharapkan.

3.2 Analisa Kebutuhan

Analisis kebutuhan merupakan analisis yang dibutuhkan dalam membuat sistem perhitungan hasil loyalitas karyawan yang berupa *input*, *output*, fungsi-fungsi yang dibutuhkan dan antar muka yang diinginkan. *Input* atau masukan dari sistem perhitungan prediksi loyalitas karyawan adalah sebagai berikut :

3.2.1 Inisialisasi Data Input

Pada tahap ini, data yang dibutuhkan untuk kasus ini adalah Daftar Penilaian Pelaksanaan Pekerjaan dari Karyawan, dimana data tersebut sebelum diimplementasikan menjadi sebuah masukan/*input* yang dapat dimengerti oleh program maka harus diinisialisasi terlebih dahulu. Dari hasil pengambilan data pada Daftar Penilaian Pelaksanaan Pekerjaan tersebut, diambil 8 variabel yang akan menghasilkan *outputan* sistem yaitu loyalitas baik atau loyalitas kurang baik, adalah :

1. Kesetiaan.
2. Prestasi Kerja.
3. Tanggung Jawab.
4. Ketaatan.
5. Kejujuran.
6. Kerjasama.
7. Prakarsa.
8. Kepemimpinan.

Di dalam penilaian pelaksanaan pekerjaan, dikelompokkan nilai berdasarkan jenjang penilaian dibawah ini :

1. Nilai 91 – 100 (Amat Baik).
2. Nilai 76 – 90 (Baik).
3. Nilai 61 – 75 (Cukup).

4. Nilai 51 – 60 (Sedang).
5. Nilai 51 kebawah (Kurang).

Dari jenjang penilaian diatas, maka dibuatlah pola perhitungan yang lebih kecil dimana *range* yang diberikan adalah 1 dan 0, dengan diasumsikan nilai 76 – 100 diberi nilai 1 dan nilai 75 kebawah diberi nilai 0. Semua variabel dihitung menggunakan konsep penilaian yang telah ditentukan tersebut.

3.2.2 Penetapan *Input*

Dalam masalah ini, untuk mengetahui loyalitas karyawan diperlukan data yang diambil dari Daftar Penilaian Pelaksanaan Pekerjaan yang didalamnya terdapat unsur-unsur penting dalam menentukan loyalitas baik atau loyalitas kurang baik dari seorang karyawan.

Data-data dari Daftar Penilaian Pelaksanaan Pekerjaan ini terdapat 8 unsur penting dalam penilaian karyawan yang kemudian ini dijadikan variabel-variabel input dan nantinya bisa menghasilkan output setelah melalui tahapan-tahapan proses. Variabel-variabel input tersebut :

X1	= Kesetiaan.
X2	= Prestasi Kerja
X3	= Tanggung Jawab
X4	= Ketaatan
X5	= Kejujuran
X6	= Kerjasama
X7	= Prakarsa
X8	= Kepemimpinan

3.2.3 Proses

Pada sistem prediksi loyalitas karyawan ini jalur proses kerjanya menggunakan jaringan saraf tiruan (JST), dimana proses kerja pada jaringan saraf tiruan menerapkan alur kerja dari arsitektur jaringan lapis banyak (*multilayer*) dengan metode *Backpropagation*.

Jaringan saraf yang terdiri dari 3 lapisan, yaitu lapisan masukan/*input* terdiri atas 8 variabel masukan unit sel saraf, lapisan tersembunyi terdiri dari atas 6 unit sel saraf dan lapisan keluaran/*output* terdiri atas 1 sel saraf. Lapisan masukan digunakan untuk menampung 8 variabel yaitu X1 sampai dengan X8, sedangkan lapisan keluaran digunakan untuk mempresentasikan pengelompokkan pola, nilai 0 untuk Loyalitas Kurang Baik dan nilai 1 untuk Loyalitas Baik.

3.2.4 *Output*

Output atau keluaran dari sistem perhitungan prediksi loyalitas karyawan adalah loyalitas baik dan loyalitas kurang baik berdasarkan 8 variabel *input*, sehingga dibutuhkan 1 *node output*. Hasil *output* yang diinginkan berupa nilai 0 (Loyalitas Kurang Baik) dan 1 (Loyalitas Baik), dimana : jika *output* bernilai $\leq 0,5$ maka dianggap 0, sedang jika *output* bernilai $>0,5$ maka dianggap bernilai 1.

3.3 Fungsi-fungsi yang dibutuhkan

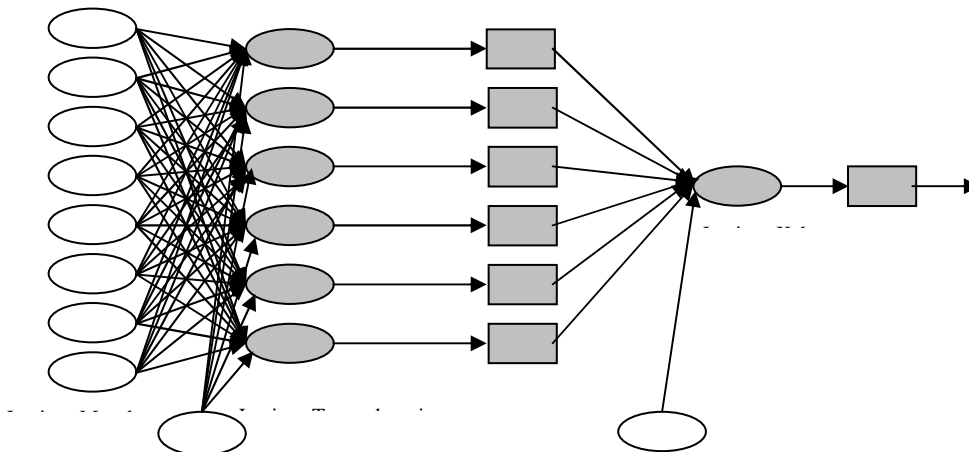
Fungsi-fungsi yang harus ditangani oleh sistem perhitungan hasil prediksi loyalitas Karyawan ini adalah menerima masukan data-data, memberikan/menampilkan data hasil pengolahan, membuat *login* dan *password* untuk pengamanan data perhitungan jaringan saraf tiruan untuk mengetahui loyalitas karyawan.

Langkah-langkah Pembuatan Aplikasi Jaringan Saraf Tiruan

1. Mengumpulkan data untuk pelatihan dan pengujian jaringan saraf tiruan. Semakin banyak data dapat diperoleh, semakin baik jaringan dapat menyelesaikan masalahnya.
2. Data yang diperoleh dipisahkan menjadi 2 bagian yang bagian pertama digunakan untuk melatih jaringan saraf dan bagian kedua dipergunakan untuk menguji hasil kerja jaringan saraf tiruan.
3. Pemilihan arsitektur dan algoritma pelatihan jaringan. Banyaknya sel *input* dan *output* disesuaikan dengan masalah yang akan diselesaikan.
4. Inisialisasi parameter jaringan saraf : Bobot, bias, konstanta belajar, target *error* dan lain-lain.
5. Penginputan data pelatihan dan melatih data tersebut. Ini dilakukan untuk melatih jaringan saraf tiruan untuk menemukan titik konvergensinya. Konvergensi jaringan saraf tiruan ditandai dengan telah tercapainya galat yang diinginkan.
6. Setelah jaringan saraf konvergen maka dilakukan pengujian terhadap data yang telah dilatihkan dan data baru yang belum pernah dilatihkan.
7. Selanjutnya jaringan saraf tiruan diimplementasikan sebagai sebuah sistem yang siap untuk menyelesaikan masalah.

3.6 Metode *Backpropagation* Untuk Mengetahui Loyalitas Karyawan

Pada sistem prediksi loyalitas karyawan ini menggunakan proses kerja jaringan saraf tiruan (JST), dimana proses kerja ini menerapkan alur kerja dari arsitektur jaringan lapis banyak (*multi layer*) dengan metode *Backpropagation*.



Gambar 3.2 Arsitektur Jaringan *Backpropagation* untuk Mengetahui Loyalitas Karyawan

Dari gambar diatas dapat dibuat contoh proses perhitungan mengetahui loyalitas karyawan. Jaringan saraf tersebut terdiri dari 3 lapisan, yaitu lapisan masukan/*input* yang terdiri atas 8 variabel *neuron*, lapisan tersembunyi terdiri atas 6 *neuron* dan lapisan keluaran/*output* terdiri atas 1 *neuron*. Lapisan masukan digunakan untuk menampung 8 variabel yaitu X_1 sampai dengan X_8 , sedangkan lapisan keluaran digunakan untuk mempresentasikan 1 variabel pengelompokan pola keluaran yaitu 1 untuk loyalitas baik dan 0 untuk loyalitas kurang baik.

Tabel 3.1 Sample Data Penilaian Pelaksanaan Pekerjaan

X1	X2	X3	X4	X5	X6	X7	X8	Target
0.75	0.5	0.5	0.5	0.5	0.5	0.5	0	0

Ketentuan-ketentuan yang digunakan :

- Jumlah *neuron* pada *input layer* = 8
- Jumlah *neuron* pada *hidden layer* = 6
- Jumlah *neuron* pada *output layer* = 1
- Learning rate* (α) = 0.5
- Maksimum epoch = 50.000
- Target *error* = 0.01

3.6.1 Inisialisasi Bobot Dan Bias

Baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka disekitar 0 dan 1 atau -1 (bias positif atau negatif).

Tabel 3.2 Bobot Awal Input ke Hidden ($V_{11} \dots V_{86}$)

$V_{11}..V_{16}$	$V_{21}..V_{26}$	$V_{31}..V_{36}$	$V_{41}..V_{46}$	$V_{51}..V_{56}$	$V_{61}..V_{66}$	$V_{71}..V_{76}$	$V_{81}..V_{86}$
0.5893	0.8772	0.2681	0.7397	0.8059	0.8189	0.673	0.3149
0.8647	0.1612	0.1775	0.7464	0.3816	0.817	0.2275	0.6564
0.6894	0.2732	0.5768	0.0804	0.612	0.1414	0.5704	0.6923
0.2878	0.5437	0.4982	0.0763	0.4368	0.0935	0.2545	0.903
0.0936	0.9608	0.2852	0.0598	0.7408	0.0672	0.3659	0.3186
0.6289	0.1744	0.1559	0.1065	0.4939	0.0603	0.8371	0.4787

Tabel 3.3 Bobot Awal Bias ke Hidden ($V_{01} \dots V_{06}$)

V_{01}	V_{02}	V_{03}	V_{04}	V_{05}	V_{06}
0.9889	0.774	0.7496	0.1977	0.1436	0.9136

Tabel 3.4 Bobot Awal Hidden ke Output ($W_1 \dots W_6$)

W_1	W_2	W_3	W_4	W_5	W_6
0.7667	0.7405	0.552	0.7236	0.4117	0.5421

Tabel 3.5 Bobot Awal Bias ke Output (W_0)

W_0
0.2965

3.6.2 Pembelajaran

a. Umpan maju (*feedforward*)

Langkah 1 : Penjumlahan Terbobot $\rightarrow Z_{in_j} = V_{0j} + \sum_{j=1}^n X_i V_{ij}$

$$Z_{in_1} = V_{01} + V_{11} * X_{11} + V_{21} * X_{12} + V_{31} * X_{13} + V_{41} * X_{14} + V_{51} * X_{15} + V_{61} * X_{16} + V_{71} * X_{17} + V_{81} * X_{18}$$

$$Z_{in_1} = 0.9889 + (0.5893 * 0.75) + (0.8772 * 0.5) + (0.2681 * 0.5) + (0.7397 * 0.5) + (0.8059 * 0.5) + (0.8189 * 0.5) + (0.673 * 0.5) + (0.3149 * 0)$$

$$Z_{in_1} = 3.5225$$

Langkah 2 : Pengaktifan $\rightarrow Z_j = f (Z_{in_j})$

$$Z_1 = \frac{1}{1 + e^{-Z_{in_1}}}$$

$$= 1 / (1 + \exp (-3.5225))$$

$$= 0.9713$$

Langkah 3 : Perkalian $\rightarrow Y_{in_k} = W_{0k} + \sum_{j=1}^p Z_j W_{jk}$

$$Y_{in} = W_{01} + W_{11} * Z_1 + W_{21} * Z_2 + W_{31} * Z_3 + W_{41} * Z_4 + W_{51} * Z_5 + W_{61} * Z_6$$

$$Y_{in} = 0.2965 + (0.7667 * 0.9713) + (0.7405 * 0.9357) + (0.552 * 0.9163) + (0.7236 * 0.7966) + (0.4117 * 0.8106) + (0.5421 * 0.9088)$$

$$= 3.6427$$

Langkah 4 : Pengaktifan $\rightarrow Y_j = f (Z_{in_j})$

$$Y = \frac{1}{1 + e^{-Y_{in}}}$$

$$= 1 / (1 + \exp (- 3.6427))$$

$$= 0.9745$$

Langkah 5 : Membandingkan Kuadrat *Error* dengan *Target Error*

$$Error = Target (i) - Y$$

$$Error = 0 - 0.9745$$

$$Error = - 0.9745$$

$$Jumlah \text{ Kuadrat } Error = Kuadrat \text{ Error} + (Error)^2$$

$$Jumlah \text{ Kuadrat } Error = 0 + (- 0.9745)^2 = 0.9497$$

Jika syarat berhenti belum terpenuhi yaitu jumlah kuadrat *error* < target *error* (0.01) atau belum sampai maksimum *epoch* (50.000) maka lanjut ke langkah berikutnya.

b. Propagasi *error* (*backpropagation of error*)

Langkah 6 : Menghitung informasi kesalahan (*error*) antara target dengan *output* yang dihasilkan jaringan $\rightarrow \delta_i = (t_i - y_i) f' (y_{in_i})$

$$\delta = (t_1 - y_1) * y_1 * [1 - y_1]$$

$$\delta = (0 - 0.9745) * 0.9745 * [1 - 0.9745]$$

$$\delta = -0.0242$$

Langkah 7 : Menghitung koreksi bobot $\rightarrow \Delta W_{jk} = \alpha \delta_k Z_j$

$$\Delta W_1 = \alpha * \delta * Z_1$$

$$\Delta W_1 = 0.5 * -0.0242 * 0.9713$$

$$\begin{aligned}\Delta W_1 &= -0.0118 \\ \Delta W_0 &= \alpha * \delta \\ \Delta W_0 &= 0.5 * -0.0242 \\ \Delta W_0 &= -0.0121\end{aligned}$$

Langkah 8 : Menjumlahkan delta *inputnya* $\rightarrow \delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk}$

$$\begin{aligned}\delta_{in_1} &= \delta * W_1 \\ \delta_{in_1} &= -0.0121 * 0.7667 \\ \delta_{in_1} &= -0.0093\end{aligned}$$

Langkah 9 : Menghitung galat informasinya $\rightarrow \delta_j = \delta_{in_j} f'(Z_{in_j})$

$$\begin{aligned}\delta_1 &= \delta_{in_1} * Z_1 * (1 - Z_1) \\ \delta_1 &= -0.0093 * 0.9713 * (1 - 0.9713) \\ \delta_1 &= -0.00026\end{aligned}$$

Langkah 10 : Menghitung koreksi bobot $\rightarrow \Delta V_{ij} = \alpha \delta_j X_i$

$$\begin{aligned}\Delta V_{11} &= \alpha \delta_1 X_1 \\ \Delta V_{11} &= 0.5 * -0.00026 * 0.75 \\ \Delta V_{11} &= -0.0000975\end{aligned}$$

Langkah 11 : Menghitung koreksi bias $\rightarrow \Delta V_{0j} = \alpha \delta_j$

$$\begin{aligned}\Delta V_{01} &= \alpha \delta_1 \\ \Delta V_{01} &= 0.5 * -0.00026 \\ \Delta V_{01} &= -0.00013\end{aligned}$$

Langkah 12 : Tiap unit tersembunyi memperbaiki bobot dan biasnya \rightarrow

$$\begin{aligned}V_{ij}(\text{baru}) &= V_{ij}(\text{lama}) + \Delta V_{ij} \text{ dan } V_{0j}(\text{baru}) = V_{0j}(\text{lama}) + \Delta V_{0j} \\ V_{11}(\text{baru}) &= V_{11}(\text{lama}) + \Delta V_{11} \\ V_{11}(\text{baru}) &= 0.5893 + -0.0000975 \\ V_{11}(\text{baru}) &= 0.5892\end{aligned}$$

$$\begin{aligned}V_{01}(\text{baru}) &= V_{01}(\text{lama}) + \Delta V_{01} \\ V_{01}(\text{baru}) &= 0.9889 + -0.00013 \\ V_{01}(\text{baru}) &= 0.9888\end{aligned}$$

Langkah 13 : Tiap unit *Output* memperbaiki bobot dan biasnya $\rightarrow W_{jk}(\text{baru}) =$

$$\begin{aligned}W_{jk}(\text{lama}) + \Delta W_{jk} \text{ dan } W_{0k}(\text{baru}) &= W_{0k}(\text{lama}) + \Delta W_{0k} \\ W_1(\text{baru}) &= W_1(\text{lama}) + \Delta W_1 \\ W_1(\text{baru}) &= 0.7667 + -0.0118 \\ W_1(\text{baru}) &= 0.7549 \\ W_0(\text{baru}) &= W_0(\text{lama}) + \Delta W_0 \\ W_0(\text{baru}) &= 0.2965 + -0.0121 \\ W_0(\text{baru}) &= 0.2844\end{aligned}$$

Pada data kedua juga dilakukan operasi yang sama dengan menggunakan bobot-bobot akhir hasil pengolahan data. Proses ini dilakukan secara berulang sampai maksimum *epoch* (50.000) atau kuadrat *error* < target *error* (0.01).

3.6.3 Output

Setelah kita mendapatkan nilai pelatihan pada iterasi yang ke yang berhenti pada saat target *error* terpenuhi, maka bobot dan bias akhir dari iterasi tersebut kita

masukkan pada nilai untuk pengujian. Setelah mendapatkan nilai Y baru kita bandingkan dengan *threshold* (nilai ambang/*teta*). *Output* dari sistem prediksi loyalitas karyawan adalah 1 (loyalitas baik) atau 0 (loyalitas kurang baik), dimana jika Y bernilai $> \textit{threshold}$ 0.5 maka dianggap 1, sedangkan jika Y bernilai $\leq \textit{threshold}$ 0.5 maka dianggap bernilai 0.

4. Kesimpulan

1. Sistem aplikasi JST merupakan sebuah simulasi dari koleksi model saraf biologi yang bisa digunakan untuk mengolah data hasil penilaian pekerjaan dengan menggunakan algoritma *Backpropagation* sehingga pelatihan jaringan yang dilakukan berulang-ulang membuat jaringan dapat semakin baik dalam mengetahui loyalitas karyawan.
2. Sistem ini dapat memberikan pelayanan yang baik dalam menginput, memproses dan memberikan keluaran dari data-data karyawan dan penilaian karyawan.
3. Dengan sistem komputerisasi maka sistem aplikasi JST mengetahui loyalitas karyawan ini mampu menerima dan mengolah data yang ada dengan hasil lebih cepat, tepat dan akurat.

DAFTAR PUSTAKA

- Kusumadewi, Sri., 2004, *Membangun Jaringan Syaraf Tiruan*, Yogyakarta : Graha Ilmu.
- Michael Negnevitsky, *Artificial Intelligent A Guide to Intelligent System*, Addison Wesley
- Suryadi H.S, Pengantar Sistem Pakar , Universitas Guna Darma
- Anita Desiani dan Muhammad Arhami, *Konsep Kecerdasan Buatan* Penerbit Andi
- Sri Kusumadewi dan Sri Hartati, *Neuro-Fuzzy Integrasi Sistem Fuzzy dan Jaringan Syaraf* Penerbit Graha Ilmu.