

# **Penerapan Jaringan Saraf Tiruan Propagasi Balik Studi Kasus Pengenalan Jenis Kopi**

***Bernard Renaldy Suteja***

*Jurusan S1 Teknik Informatika*

*Fakultas Teknologi Informasi, Universitas Kristen Maranatha*

*Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164*

*Email: [bernardjogja@gmail.com](mailto:bernardjogja@gmail.com)*

---

## **Abstract**

*Testing the taste of coffee traditionally, will depend on human taste sensor, which is not stable and relied on physical and mental conditions. In this situation, only qualitative measures can be determined. To increase reliability in the evaluation of taste of coffee, an electronic taste sensor system can be used to measure objectively. Neural network approach is very appropriate to apply pattern recognition method in an electronic taste sensor system. This paper discusses about neural network with back propagation for recognizing the kind of coffee. Back propagation architecture was chosen because can produce many tastes as output, so alternative identifications can be seen ( in the form of correlation factor).*

***Keywords :*** *Neural network, Back propagation*

---

## **1. Pendahuluan**

Evaluasi rasa produksi kopi pabrikan biasanya secara tradisional bergantung pada indera pengecap manusia. Tetapi sistem pengecap manusia biasanya tidak stabil, tergantung kondisi fisik atau mental yang bersangkutan pada saat itu, dan hanya ukuran kualitatif yang bisa ditetapkan. Untuk memungkinkan evaluasi rasa produksi kopi dengan kehandalan tinggi yang kontinu, sistem sensor pengecap elektronik yang menghasilkan pengukuran obyektif bisa digunakan. Meniru indera pengecap atau pembau pada manusia (keduanya menggunakan metode penginderaan yang sama), maka sebuah sistem sensor pengecap adalah sebuah larikan yang terdiri dari beberapa sensor pengecap tunggal (pada hidung manusia terdapat tujuh sel pembau dengan kemampuan membedakan ratusan jenis aroma(Gopel,1989)).

Pada dasarnya sebuah rasa yang ada adalah kombinasi satu sampai lima dari lima rasa dasar (dengan perbandingan kuantitas beragam), yaitu asam (dihasilkan ion hidrogen), asin (a.l dihasilkan NaCl dan KCl), pahit (a.l dihasilkan *caffeine*, *quinine*), manis (a.l dihasilkan *sucrose*, *glucose*), dan yang terakhir *umami*, istilah bahasa Jepang yang menggambarkan rasa gurih (dihasilkan MSG). Rasa kopi adalah campuran dari rasa asam dan pahit (Fukunaga, 1995).

Sistem sensor akan mengekstrak karakteristik sebuah rasa dalam bentuk bentukan pola sinyal. Untuk kelas rasa yang sama, bentukan pola akan mirip (secara visual maupun perhitungan kuantitatif). Tentunya sensor yang digunakan di sini harus selektif terhadap kelima jenis rasa tersebut, atau mempunyai sifat selektivitas global (*global selectivity*). Ini berbeda dengan sifat umum sensor kimia

(*chemosensor*) yang mempunyai sifat selektivitas lokal (hanya selektif terhadap aroma atau rasa tertentu). Sistem sensor pengecap atau pembau biologis mamalia/manusia juga menghasilkan hal yang serupa, dan menurut Moulton (Gopel,1989) pengenalan rasa/bau dilakukan oleh semacam metode pengenalan pola di dalam otak.

Pendekatan jaringan saraf tiruan (JST) sangat cocok untuk menerapkan metode pengenalan pola untuk melengkapi sistem sensor pengecap elektronik. Paper ini melengkapi sebuah program JST propagasi balik pengenalan jenis kopi, yang dibuat dengan bahasa Visual Basic 6.0. Arsitektur propagasi balik dipilih karena dapat sekaligus menampilkan berbagai ragam rasa sebagai keluaran, sehingga identifikasi-identifikasi alternatif dapat dilihat (dalam bentuk faktor korelasi). Arsitektur lain seperti Hebb, *heteroassociative*, dan *autoassociative* hanya menetapkan keluaran yang menyatakan apakah sebuah pola yang diuji adalah termasuk pola kelas yang disimpan jaringan atau tidak. Pada program ini masukan ditentukan sebanyak tujuh *node*, sesuai dengan cacah sensor dalam larikan, dan satu *hidden layer* sebanyak lima *node*. Cacah keluaran yang menyatakan cacah kelas rasa kopi ditentukan sebanyak tiga *node* (untuk biji kopi tumbuk asal Indonesia, dari luar Indonesia, dan kopi kemasan pabrik). Program ini tidak diarahkan sebagai aplikasi yang bisa digunakan pabrik kopi, tetapi lebih ditujukan untuk memperlihatkan bagaimana JST menangani pengelompokan kelas rasa kopi.

## **2. Dasar Teori**

Pada bagian ini akan dibahas dasar teori yang menyangkut arsitektur JST propagasi balik, dan algoritma untuk menghitung target pelatihan jaringan. Pengambilan data dan implementasinya dalam program akan dibahas pada bagian berikutnya.

### **2.1 Jaringan Saraf Tiruan Propagasi Balik**

Salah satu arsitektur jaringan saraf tiruan yang sangat populer adalah *multilayer feedforward networks*. Secara umum, jaringan seperti ini terdiri dari sejumlah unit neuron sebagai lapisan *input*, satu atau lebih lapisan simpul-simpul neuron komputasi *hidden* (lapisan tersembunyi), dan sebuah lapisan simpul-simpul neuron komputasi *output*. Sinyal *input* dipropagasikan ke arah depan (arah lapisan *output*), lapisan demi lapisan. Jenis jaringan ini adalah hasil generalisasi dari arsitektur *perceptron* satu lapisan, jadi biasa disebut sebagai *multilayer perceptron* (MLPs).

*Error back propagation* adalah algoritma MLPs yang menggunakan prinsip *supervised learning*. Propagasi balik (ke arah lapisan *input*) terjadi setelah jaringan menghasilkan *output* yang mengandung *error*. Pada fase ini seluruh bobot *synaptic* (yang tidak memiliki aktivasi nol) dalam jaringan akan disesuaikan untuk mengoreksi/memperkecil *error* yang terjadi (*error correction rule*). Untuk pelatihan jaringan, pasangan fase propagasi ke depan dan balik dilakukan secara berulang untuk satu set data latihan, kemudian diulangi untuk sejumlah *epoch* (satu sesi lewatan untuk seluruh data latihan dalam sebuah proses pelatihan jaringan) sampai *error* yang terjadi mencapai batas kecil toleransi tertentu atau nol.

Fungsi aktivasi yang digunakan pada arsitektur ini adalah yang menghasilkan nilai kontinu, jadi *output* jaringan juga akan bernilai kontinu. Fungsi aktivasi yang umum digunakan disini adalah *sigmoidal* (*biner* atau *bipolar*, mengacu pada pembuatan program, akan ditunjukkan yang bipolar, *range*:[-1,1]) :

$$f(v_n) = y_n = \frac{2}{1 + \exp(-v_n)} - 1 \quad (1)$$

$y_j$  adalah nilai aktivasi setiap neuron/*node*, dan  $v_j$  adalah fungsi integrasi untuk setiap neuron, dimana :

$$v_n = \sum_m w_{mn} y_{mn} - \theta_n \quad (2)$$

$w_{ij}$  adalah bobot synaptic *link*-link antar neuron, dan  $\theta_n$  adalah *threshold* (nilai ambang) yang dimiliki setiap neuron. Untuk menyederhanakan perhitungan, pada setiap lapisan (*input*, *hidden*, dan *output*) diberikan neuron tambahan yang selalu memiliki nilai aktivasi 1, dan bobot synaptic setiap *link*-nya dengan neuron tertentu sama dengan nilai *threshold* neuron tersebut. Sehingga  $v_n$  menjadi :

$$v_n = \sum_m w_{mn} y_{mn} \quad (3)$$

Untuk pelatihan, diberikan satu set data latihan ( $s$   $t$ ) dimana  $s$  adalah *input* dan  $t$  adalah target (*output*) yang valid untuk  $s$ . *Error* terjadi bila *output* yang dihasilkan jaringan dengan bobot synaptic yang digunakan saat itu tidak sama dengan target (pada satu atau lebih neuron *output*). Sinyal *error* pada sebuah neuron *output* adalah :

$$e_n = t_n - y_n \quad (4)$$

Jumlahan kuadrat sinyal *error* digunakan sebagai acuan untuk melihat apakah jaringan sudah terlatih dengan baik atau tidak. Semakin kecil nilainya menunjukkan bobot synaptic setiap *link* semakin menyesuaikan untuk menghasilkan *output* yang benar. ( $n$  = jumlah *output*).

$$E = \frac{1}{2} \sum_n e_n^2 \quad (5)$$

*Error* kuadrat rerata (*average squared error*) adalah rata-rata  $E$  terhadap cacah data latihan ( $P$ ).

$$E_v = \frac{1}{P} \sum_P E_P \quad (6)$$

### 2.1.1. Derivasi untuk Algoritma Propagasi Balik

Jumlahan kuadrat sinyal *error* (rumus 4) adalah fungsi dari seluruh bobot pada jaringan. Gradien  $E$  adalah vektor yang terdiri dari derivatif-derivatif parsial dari  $E$  terhadap setiap bobot. Vektor ini menunjukkan arah dimana  $E$  akan meningkat. Arah kebalikannya akan menunjukkan pengurangan pada  $E$  (Fausett, 1994). Dengan demikian bobot akan menyesuaikan diri untuk memperkecil  $E$  sebesar :

$$\Delta w_{mn} = -\frac{\partial E}{\partial w_{mn}} \quad (7)$$

Konstanta *learning rate* (laju belajar) dapat disertakan sebagai berikut ( $1 \geq \eta > 0$ ) :

$$\Delta w_{mn} = -\eta \frac{\partial E}{\partial w_{mn}} \quad (8)$$

Jika dalam sebuah arsitektur MLPs terdapat  $n$  lapisan, dengan lapisan ke- $n$  adalah lapisan *output*, maka penyesuaian bobot untuk setiap lapisan adalah sebagai berikut ( indeks lapisan ke- $n$  adalah  $a$ , lapisan ke- $(n-1)$  adalah  $b$ , ke- $(n-2)$  adalah  $c$ , ke- $(n-3)$  adalah  $d...$  dst) :

Link menuju lapisan  $n$  :

$$\frac{\partial E}{\partial w_{ba}} = \frac{\partial E}{\partial e_a} \frac{\partial e_a}{\partial y_a} \frac{\partial y_a}{\partial v_a} \frac{\partial v_a}{\partial w_{ba}}$$

$$E = \frac{1}{2} \sum_a e_a^2 \quad ; \quad \frac{\partial E}{\partial e_a} = e_a$$

$$e_a = t_a - y_a \quad ; \quad \frac{\partial e_a}{\partial y_a} = -1$$

$$y_a = f(v_a) \quad ; \quad \frac{\partial y_a}{\partial v_a} = y_a^1 = f^1(v_a)$$

$$v_a = \sum_b w_{ba} y_{ba} \quad ; \quad \frac{\partial v_a}{\partial w_{ba}} = y_{ba}$$

$$\frac{\partial E}{\partial w_{ba}} = -e_a f^1(v_a) y_{ba} \quad (9)$$

$$\delta_n = e_a f^1(v_a) \quad ; \quad \frac{\partial E}{\partial w_{ba}} = -\delta_n y_{ba} \quad ; \quad \delta = \text{gradien lokal}$$

Link menuju lapisan (n-1) :

$$\frac{\partial E}{\partial w_{cb}} = \frac{\partial E}{\partial y_{ba}} \frac{\partial y_{ba}}{\partial v_b} \frac{\partial v_b}{\partial w_{cb}} ;$$

$$\frac{\partial E}{\partial y_{ba}} = \frac{\partial E}{\partial e_a} \frac{\partial e_a}{\partial v_a} \frac{\partial v_a}{\partial y_{ba}} = \sum_a e_a \frac{\partial e_a}{\partial v_a} \frac{\partial v_a}{\partial y_{ba}}$$

$$e_a = t_a - y_a = t_a - f(v_a) ; \quad \frac{\partial e_a}{\partial v_a} = -f'(v_a)$$

$$v_a = \sum_b w_{ba} y_{ba} ; \quad \frac{\partial v_a}{\partial y_{ba}} = w_{ba}$$

$$y_{ba} = f(v_b) ; \quad \frac{\partial y_{ba}}{\partial v_b} = y'_{ba} = f'(v_b)$$

$$v_b = \sum_c w_{cb} y_{cb} ; \quad \frac{\partial v_b}{\partial w_{cb}} = y_{cb}$$

$$\frac{\partial E}{\partial w_{cb}} = -y_{cb} f'(v_b) \sum_a e_a f'(v_a) w_{ba} \tag{10}$$

$$\delta_{n-1} = f'(v_b) \sum_a \delta_n w_{ba} ; \quad \frac{\partial E}{\partial w_{cb}} = -\delta_{n-1} y_{cb}$$

Link menuju lapisan(n-2) :

$$\frac{\partial E}{\partial w_{dc}} = \frac{\partial E}{\partial y_{cb}} \frac{\partial y_{cb}}{\partial v_c} \frac{\partial v_c}{\partial w_{dc}}$$

$$\frac{\partial E}{\partial y_{cb}} = \frac{\partial E}{\partial y_{ba}} \frac{\partial y_{ba}}{\partial v_b} \frac{\partial v_b}{\partial y_{cb}} = \sum_b \sum_a e_a \frac{\partial e_a}{\partial v_a} \frac{\partial v_a}{\partial y_{ba}} \frac{\partial v_b}{\partial y_{cb}}$$

$$v_b = \sum_c w_{cb} y_{cb} ; \quad \frac{\partial v_b}{\partial y_{cb}} = w_{cb}$$

$$y_{cb} = f(v_c) ; \quad \frac{\partial y_{cb}}{\partial v_c} = y'_{cb} = f'(v_c)$$

$$v_c = \sum_d w_{dc} y_{dc} ; \quad \frac{\partial v_c}{\partial w_{dc}} = y_{dc}$$

$$\frac{\partial E}{\partial w_{dc}} = -y_{dc} f'(v_c) \sum_b f'(v_b) \left( \sum_a e_a f'(v_a) \right) w_{cb} \quad (11)$$

$$\delta_{n-2} = f'(v_c) \sum_b \delta_{n-1} w_{cb} \quad ; \quad \frac{\partial E}{\partial w_{dc}} = -\delta_{n-2} y_{dc}$$

Algoritma di atas berlaku rekursif untuk *link-link* menuju lapisan-lapisan sebelumnya, sampai ke lapisan sebelum lapisan *input*. Sebagai contoh, nilai penyesuaian bobot berikutnya (*link* menuju lapisan ( $n-3$ )) :

$$\frac{\partial E}{\partial w_{ed}} = -y_{ed} f'(v_d) \sum_c f'(v_c) \left( \sum_b f'(v_b) \left( \sum_a e_a f'(v_a) \right) \right) w_{dc} \quad (12)$$

### 2.2.2. Algoritma Propagasi Balik

Menurut Haykin (1994), prosedur algoritma propagasi balik akan mengikuti sekuen sebagai berikut :

1. Inisialisasi ; menentukan konfigurasi jaringan, kemudian menetapkan seluruh bobot *synaptic* dan *threshol*d dengan nilai acak kecil yang terdistribusi secara seragam.
2. Menyiapkan data pelatihan ; untuk setiap pasangan ( $s \ t$ ) dilakukan langkah 3 (komputasi ke depan) dan langkah 4 (komputasi balik) secara berurutan.
3. Komputasi ke depan ; *input* yang tersedia adalah nilai aktivasi bagi neuron-neuron sesudah lapisan *input*. Kemudian untuk lapisan-lapisan berikutnya nilai aktivasi dihasilkan kemudian di propagasikan dengan memakai fungsi aktivasi *sigmoid* ( rumus 1 ). Pada lapisan *output* nilai aktivasinya adalah sebagai keluaran.
4. Komputasi balik ; seluruh bobot *synaptic* disesuaikan untuk memperkecil *error*. Mulai dari *link-link* yang menuju lapisan output, sampai *link-link* yang menuju lapisan *hidden* pertama. Untuk penyesuaian bobot ini antara lain bisa menggunakan rumus 7 – rumus 10, tergantung posisi *link*. Untuk mempercepat konvergensi, bisa ditambahkan parameter momentum ( $\alpha$ ). Untuk data latihan ke- $p$  :

$$w_{mn}(p) = w_{mn}(p-1) + \Delta w_{mn}(p) + \alpha(w_{mn}(p-1) - w_{mn}(p-2))$$

$$= w_{mn}(p-1) + \Delta w_{mn}(p) + \alpha \Delta w_{mn}(p-1) \quad (13)$$

5. Iterasi ; iterasi dilakukan untuk sejumlah *epoch*, jika masih terjadi *error*, sampai *average squared error* (rumus 5) mencapai batas kecil toleransi tertentu atau nol.
6. Langkah 2 s/d 4 adalah algoritma untuk melatih jaringan. Untuk pengujian pola, cukup dilakukan komputasi ke depan satu lewatan (langkah 3).

### 2.2.3 Metode Korelasi untuk Pencocokan Pola (Pattern Matching)

Faktor korelasi antara pola uji dan pola kalibrasi suatu kelas rasa dapat digunakan sebagai target pelatihan jaringan yang menggunakan fungsi aktivasi *bipolar sigmoid*. Nilai +1 dari faktor korelasi menunjukkan obyek yang sama. Nilai yang mendekati itu menunjukkan obyek yang makin serupa. Nilai -1 menunjukkan dua obyek yang benar-benar berbeda. Nilai aktivasi *bipolar sigmoid* akan menunjukkan *range* yang sama.

Untuk suatu set data yang tersedia (berisi ragam pola bahan-bahan yang menghasilkan kelas rasa yang sama, jadi mempunyai pola yang mirip), nilai kalibrasi dapat ditentukan melalui langkah-langkah sebagai berikut (Gopel, 1989) :

1.  $N_{ik}$  adalah sinyal rerata terhadap cacah ragam pola ( $n$ ) yang mirip (bisa juga rerata terhadap ragam konsentrasi untuk bahan yang sama, karena sinyal adalah fungsi linear dari konsentrasi, yang biasanya mengandung nonlinearitas).

$$N_{ik} = \frac{1}{n} \sum_n N_{ikn} \quad (14)$$

2.  $N_k$  adalah rerata  $N_{ik}$  terhadap cacah sensor yang digunakan ( $i$ ).

$$N_k = \frac{1}{i} \sum_i N_{ik} \quad (15)$$

3. Nilai kalibrasi adalah ( $N_{ik} - N_k$ ).

Sebuah pola tunggal yang akan diuji menggunakan kuantitas yang dihitung dengan cara yang sama, ( $S_{ipn} - S_{pn}$ ), tetapi untuk  $n = 1$ ;  $i$  : merupan iterasi pelatihan

$$S_{pn} = \frac{1}{i} \sum_n S_{ipn} \quad (16)$$

Dengan demikian faktor korelasi dapat dihitung sebagai berikut :

$$\rho = \frac{\sum_i (S_{ipn} - S_{pn})(N_{ik} - N_k)}{\sqrt{\sum_i (S_{ipn} - S_{pn}) \sum_i (N_{ik} - N_k)}} \quad (17)$$

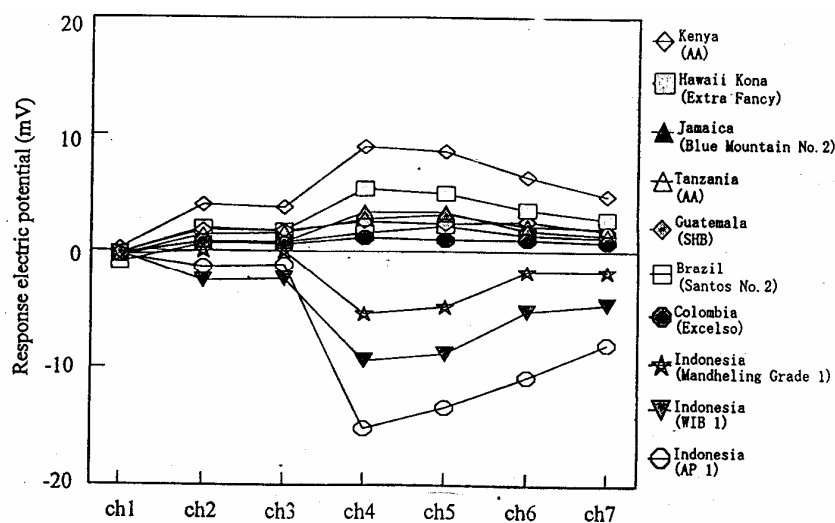
### 3. Perancangan Arsitektur JST Propagasi Balik, Pembuatan dan Pengujian Program

#### 3.1 Kopi yang digunakan

Di sini akan digunakan berbagai macam jenis kopi tumbuk, yaitu dari biji kopi asal Indonesia, dari luar Indonesia, dan berbagai macam kopi kemasan komersial yang banyak dijumpai di pasaran. Seperti telah disebutkan, pembuatan program tidak ditujukan untuk dikembangkan bagi penggunaan kalangan pabrik, tetapi hanya mensimulasikan bagaimana JST propagasi balik menangani pengelompokan berbagai jenis kopi.

Pengukuran sinyal berbagai jenis kopi di sini dilakukan oleh Taiji Fukunaga dkk (1995), dengan sampel sebelas jenis kopi tumbuk yang berasal dari berbagai negara, antara lain Salvador (CS), Brazil (Santos No.2), Guatemala(SHB), Jamaica (Blue Mountain No.2), Hawaii Kona (Extra Fancy), Kenya(AA), Tanzania(AA), Colombia(Excelso), Indonesia Mandheling(Grade 1), Indonesia WIB 1, dan Indonesia AP 1. Setelah keseluruhan biji kopi dipanggang, kemudian ditumbuk halus, 8 gr dari masing-masing sampel dimasukkan dalam gelas berisi 150 ml air panas bersuhu 95° C. Kemudian diaduk tiga putaran, dibiarkan mendingin sampai 60° C, dan sampel siap diukur.

Sensor pengecap yang digunakan terdiri dari tujuh sensor, yaitu sesuai urutan : DOP(*dioctylphosphate*), DOP:TOMA=9:1, DOP:TOMA=5:5, DOP:TOMA= 3:7, TOMA(*trioctyl methyl ammonium chloride*), OA (*oleic acid*), dan OAm(*oleyl amine*). Penamaan didasarkan jenis *lipid membrane* (lapisan bahan berlemak) yang digunakan masing-masing sensor. Sinyal adalah perubahan potensial (mV) relatif terhadap potensial nol kopi asal Salvador(CS), jadi data yang diambil adalah untuk sepuluh jenis kopi. Pengukuran sinyal sepuluh jenis kopi tersebut dilakukan dua kali dengan prosedur rotasi. Sinyal rerata digunakan sebagai sinyal yang membentuk pola.

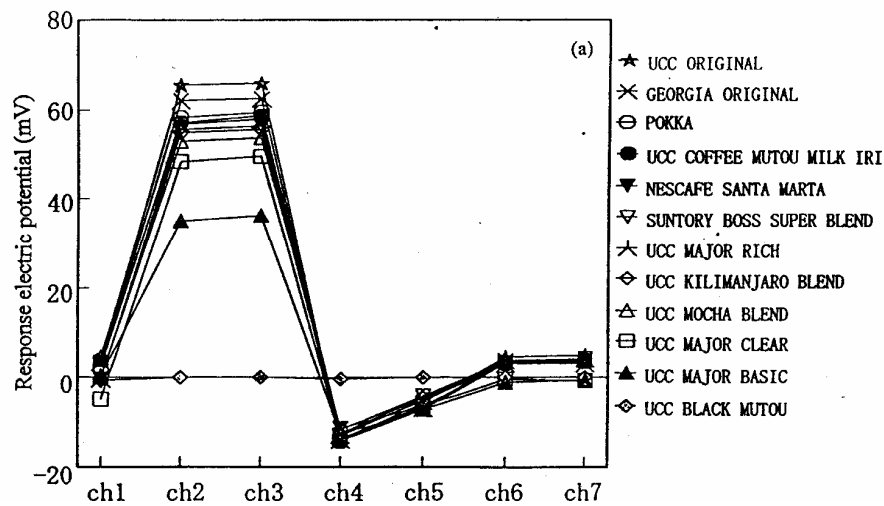


Gambar 1 Pola respon sensor terhadap 10 jenis biji kopi



Pada gambar 1 terlihat kopi asal luar Indonesia secara visual memiliki bentuk pola yang mirip. Sedangkan kopi asal Indonesia adalah cerminannya di kuadran negatif. Bagi sistem pengenalan pola yang kita buat, ini adalah pola yang berbeda. Dengan demikian dapat ditentukan nilai kalibrasi untuk dua jenis kopi tumbuk, yaitu asal Indonesia dan luar Indonesia.

Berikutnya, dilakukan pengukuran menggunakan sistem sensor yang sama untuk kopi kemasan komersial. Diambil duabelas sampel kopi kemasan yang masing-masing sinyalnya diukur relatif terhadap potensial kopi bermerk Black Motou, yang merupakan kopi bebas gula. Sebelas sampel lainnya adalah yang mengandung rasa susu. Sensor 9:1 dan 5: 5 (ch2 & ch3) memberikan respon yang sangat sensitif terhadap rasa ini (lihat Gambar 2).

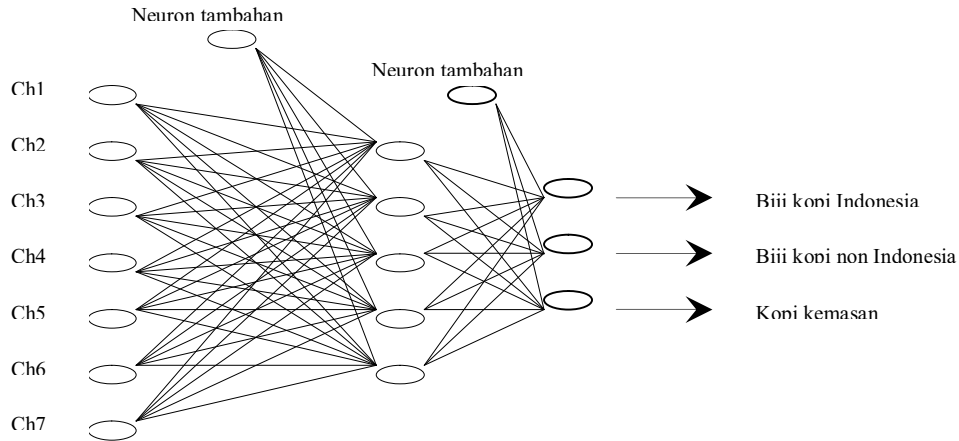


Gambar 2 Pola respon sensor terhadap 11 jenis kopi komersial

Kemudian ditentukan nilai kalibrasi untuk jenis kopi ketiga ini. Dalam perhitungan ini UCC Black Motou dikecualikan karena memiliki potensial hampir sama untuk semua sensor, yang dengan demikian dijadikan sebagai acuan relatif pengukuran respon untuk kopi yang lain.

### 3.2 Perancangan Arsitektur JST Propagasi Balik

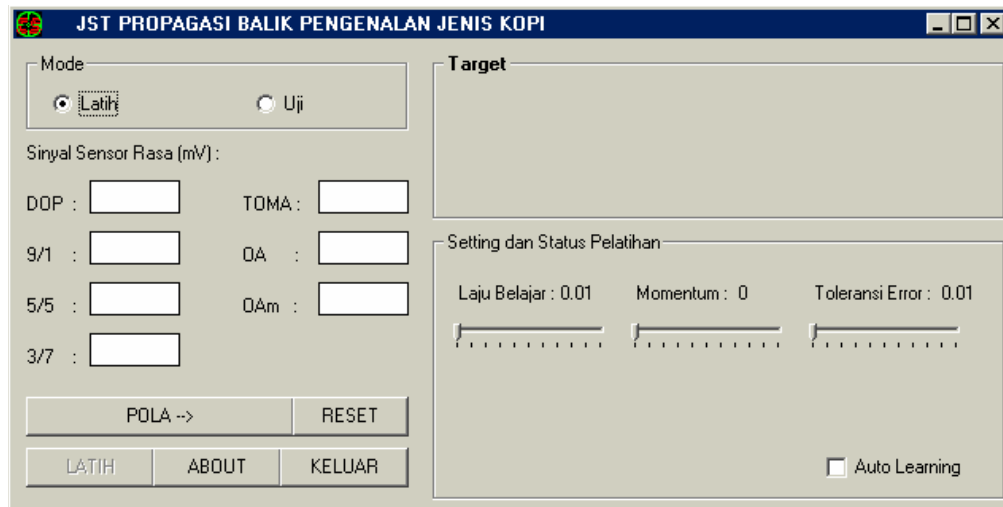
Arsitektur JST yang dipergunakan di sini adalah propagasi balik dengan cacah *input* tujuh *node*, sesuai jumlah sensor, sebuah *hidden layer* dengan cacah *node* lima, dan *output* dengan cacah *node* tiga, sesuai dengan tiga kelas kopi yang akan dikelompokkan. Seluruh *link synaptic* terhubung penuh (*fully connected*), dengan fungsi aktivasi *bipolar sigmoid*. Untuk meniadakan *threshold* dalam perhitungan, setiap lapisan diberikan *node* tambahan dengan nilai aktivasi selalu 1, dan bobot *synaptic* sama dengan nilai *threshold node* tempat ia terhubung.



Gambar 3 Skema JST yang digunakan

### 3.3 Pembuatan Program

Antarmuka program dapat dilihat pada gambar 4 sebagai berikut :



Gambar 4 Antarmuka program JST Pengenalan Jenis Kopi

Disediakan dua mode aplikasi, yaitu mode latih dan mode uji. Mode uji menggunakan komputasi ke depan satu lewatan (*one pass forward computation*) dengan menggunakan nilai bobot-bobot *synaptic* terakhir yang dipunyai jaringan pada saat itu. Mode latih adalah mode untuk melatih jaringan, untuk menyesuaikan bobot *synaptic* agar memberikan keluaran yang sesuai target yang diberikan. Jadi untuk mode ini digunakan proses komputasi ke depan dan balik (*two pass*

computation) untuk seluruh data latihan, dan diiterasikan sampai bobot konvergen. Target dengan range (-1,1) dihitung menggunakan rumus 17.

Ada dua macam cara pengisian data latihan, yaitu secara manual, atau memilih cara *auto learning*, dimana data latihan yang digunakan, secara otomatis adalah pola-pola kalibrasi yang tersedia (pola pada gambar 1 dan gambar 2). Urutan data latihan untuk setiap *epoch* dibuat acak. Parameter-parameter latihan (*learning rate*, momentum) dapat disesuaikan menurut keinginan pengguna. Nilai-nilai ini akan tetap untuk sebuah sesi latihan. Toleransi *average squared error* untuk menghentikan latihan juga dapat ditentukan oleh pengguna.

Setelah sebuah sesi latihan selesai, bobot yang sudah di-update disimpan dalam *file* bobot bernama *lat2.dat* pada direktori C:\JST Bimo(18453). Bobot ini akan digunakan untuk komputasi latihan maupun uji berikutnya.

### 3.3 Pengujian Program

#### 3.3.1 Melatih Jaringan

Pengisian data latihan secara manual adalah dengan satu persatu memasukkan data sinyal untuk secara berurut melingkar : biji kopi Indonesia, biji kopi non Indonesia, dan kopi kemasan. Untuk setiap pemasukan data, target yang dihitung dengan rumus 17 akan ditampilkan. Nilai-nilai kalibrasi telah lebih dahulu dihitung dan digunakan sebagai konstanta-konstanta dalam program. Di sini bisa dikatakan pola diuji dengan metode korelasi. Hasil pengenalan adalah target yang ditampilkan. Jadi pada bagian ini juga bisa dilakukan pengujian dengan tidak menggunakan metode JST.

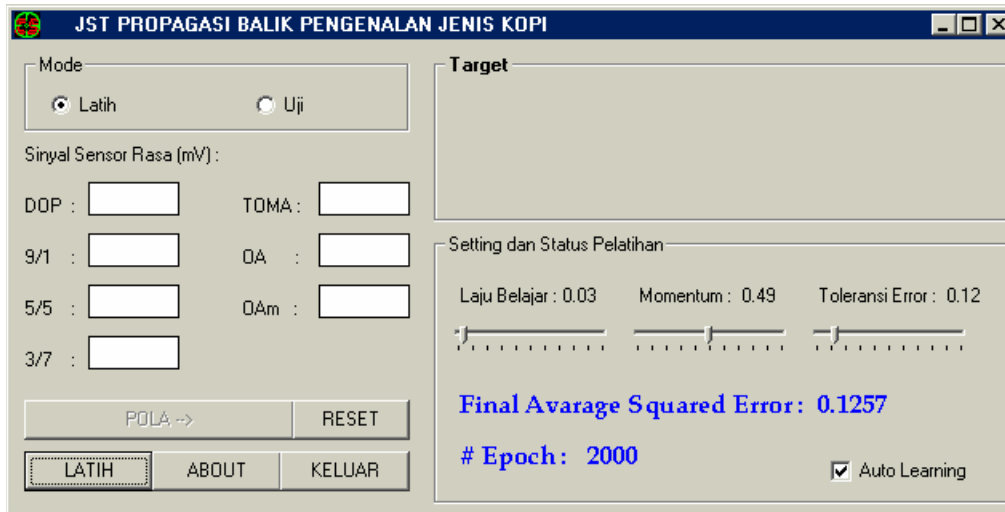


**Gambar 5** Target yang dihitung dengan metode korelasi untuk data latihan kopi kemasan dengan merk “Nescafe Santa Marta”

Gambar 5 memperlihatkan target untuk data latihan kopi kemasan “Nescafe Santa Marta”. Dari tampilan terlihat hasil perhitungan metode korelasi yang bisa diandalkan untuk melatih jaringan.

Penentuan nilai bobot *synaptic* adalah dengan menggunakan mode latihan *auto learning*, dengan 9 data latihan. Yaitu: biji kopi Indonesia (Mandheling Grade1, WIB1, AP1), biji kopi non Indonesia(AA, Extra Fancy, Blue Mountain No.2), dan kopi kemasan(UCC Original, Georgia Original, Pokka). Alasan pada dua kelas terakhir ditetapkan tiga macam adalah agar jumlah sampel untuk setiap kelas sama (untuk biji kopi Indonesia hanya tersedia tiga data).

Pada gambar 6 diperlihatkan hasil pelatihan untuk 9 data, dimana target dihitung secara otomatis. Disini bobot akan konvergen pada *average squared error(ase)* sebesar **0.1257**, setelah melalui beberapa kali latihan dengan laju belajar dan momentum terus dikurangi (iterasi dibatasi sampai 2000 *epoch* untuk menjaga jika bobot akan konvergen pada *error* yang tidak lebih kecil dari toleransi yang diberikan). Pada latihan terakhir ini laju belajar ditetapkan sebesar **0.03**, momentum sebesar **0.49**, dan toleransi *error (ase)* **0.12** (*setting* pada latihan pertama adalah : laju belajar : **0.34**, momentum : **0.9**, toleransi *error* : **0.13**, *ase* akan mencapai **0.1265** pada 38 *epoch*).



Gambar 6 Pelatihan dengan *autolearning*

### 3.3.2 Pengujian Jaringan dan Pembahasan

Jaringan akan diuji untuk dua buah pola, masing-masing adalah biji kopi non Indonesia(Kenya (AA)), dan kopi kemasan (Nescafe Santa Marta).

Pada gambar 7 diperlihatkan hasil pengujian kopi Kenya (AA) pada jaringan yang sudah dilatih. Keluarannya adalah (-0.9999976, 0.8276284, -0.9999991) untuk target (-0.934845, 0.9940835, -0.3901919).



**Gambar 7** Pengujian kopi Kenya AA

Berikut jaringan diuji dengan “Nescafe Santa Marta”(gambar 8). Keluaran yang dihasilkan bisa dilihat pada gambar 8, yaitu (0.9999999, -0.9041493, 1) untuk target (0.6678578, -0.4178505, 0.9967795).



**Gambar 8** Hasil pengujian “Nescafe Santa Marta”

Tidak semua sampel uji yang tersedia diidentifikasi mendekati benar (antara lain karena  $ase = 0.1257$ ). Menurut Fausett (1994), untuk melatih jaringan dengan cacah bobot/link dalam jaringan ( $W$ ) dengan tingkat keakuratan  $1 - \epsilon$ , diperlukan cacah data latihan ( $P$ ) sebanyak :

$$P \geq \frac{W}{\epsilon} \quad 3.1$$

Jadi untuk mencapai akurasi misalnya 90% (90% data uji yang diambil dari ruang sampel yang sama dengan data latihan akan diukur secara akurat) pada jaringan dengan cacah bobot 50 diperlukan paling tidak 500 data latihan.

Selain itu ada beberapa cara menurut Haykin(1994), yang dapat meningkatkan performansi jaringan. Yaitu yang digunakan dalam pembuatan program ini : urutan data latihan untuk setiap *epoch* dibuat acak, dan penggunaan fungsi pemetaan tertentu untuk menentukan target latihan (*prior information*). Penggunaan ini akan memberikan data pelatihan yang cukup handal bagi jaringan, dan mempercepat konvergensi bobot.

#### **4. Kesimpulan**

Arsitektur JST propagasi balik dengan fungsi aktivasi kontinu dapat memberikan performansi yang cukup baik untuk mengelompokkan kelas-kelas rasa kopi tertentu berdasarkan faktor korelasi. Penggunaan arsitektur JST lain untuk bidang ini akan membawa kesulitan dalam mencirikan rasa sebagai keluaran (rasa dibentuk dari kombinasi lima rasa dasar dengan ragam variasi komposisi tak berhingga, sehingga membuat penentuan cacah bit informasi biner menjadi tidak mungkin).

Penggunaan data latihan yang semakin banyak akan memberikan keakuratan pengukuran jaringan terhadap data-data uji yang diambil dari ruang sampel yang sama dengan data latihan tersebut. Program ini masih membutuhkan data latihan yang lebih banyak untuk meningkatkan akurasinya. Di masa mendatang, diharapkan didapatkan lebih banyak data untuk jenis-jenis kopi yang sama yang diukur dengan prosedur dan sistem sensor yang juga sama.

#### **5. Daftar Pustaka**

- [Fau94] Fausett, L..(1994). *Fundamentals of Neural Networks : Architecture, Algorithms, and Applications*. Prentice Hall Englewood Cliffs.
- [Fuk96] Fukunaga, T., Toko, K., Mori, S., Nakabayashi, Y., Kanda, M., (1996). *Quantification of Taste of Coffee Using Sensors with Global Selectivity*, *Sensors and Materials*, Vol. 8, No.1(1996), 047-056.
- [Gop89] Gopel, W., Hesse, J., Zemel, J.N., (1989). *Sensors : A Comprehensive Survey*, VCH Verlagsgesellschaft.
- [hay94] Haykin, S., (1994). *Neural Networks : A Comprehensive Foundation*, MacMillan College Publishing Company.