

PENERAPAN BAGGING UNTUK PENINGKATAN AKURASI KLASIFIKASI TEMA TUGAS AKHIR MENGGUNAKAN ALGORITMA BACKPROPAGATION NEURAL NETWORK

Arfan Haqiqi Sulasmoro^{1,2}, Catur Supriyanto¹, Vincent Suhartono¹

¹Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang

²Program Studi Diploma Tiga Teknik Komputer, Politeknik Harapan Bersama, Tegal

Arfan.hqq@gmail.com, catur.supriyanto@dsn.dinus.ac.id, vishtn@gmail.com

Abstrak

Pertumbuhan jumlah mahasiswa membutuhkan penanganan dalam upaya menjaga keseimbangan kualitas kelulusan pada perguruan tinggi swasta. Pengelompokan tema tugas akhir mahasiswa di sebuah pendidikan tinggi swasta dengan melihat hasil studi untuk tiap-tiap matakuliah yang diberikan dengan melihat histori nilai tiap-tiap semester untuk matakuliah tertentu diharapkan dapat membantu dalam penyelesaian tugas akhir. Pada penelitian ini mencoba menerapkan metode *Neural network* dengan struktur *Backpropagation* untuk melakukan klasifikasi tema tugas akhir di program studi DIII Teknik Komputer Politeknik Harapan Bersama Tegal dengan *evaluation pattern* menggunakan aplikasi rapidminer 5.2. Penerapan metode ini bertujuan untuk mendapatkan hasil klasifikasi tema tugas akhir dengan BPNN. Hasil dari penelitian menunjukkan bahwa penerapan metode *bagging* pada proses *training* dapat meningkatkan nilai *accuracy* dalam mengklasifikasi tema tugas akhir menggunakan algoritma *Backpropagation Neural Network*.

Kata kunci : Klasifikasi, backpropagation, *neural network*, Bagging.

1. Pendahuluan

Perguruan tinggi memiliki peran yang sangat strategis di tengah-tengah masyarakatnya. Peran strategis yang meningkat membuat pendidikan tinggi sering dirumuskan kedalam tiga wacana besar, yaitu pengajaran (*knowledge transfer process*), riset (*research*) dan benteng peradaban. Ketiga peran tersebut tersirat dalam semangat tridharma perguruan tinggi atau kemudian lebih lanjut sering disebut dengan tridharma.

Angka partisipasi perguruan tinggi (persentase jumlah mahasiswa perguruan tinggi terhadap penduduk berumur 19 – 24 tahun) adalah sebanyak 5,3% pada tahun 1983 – 1984, meningkat menjadi 8,5% pada tahun 1988 – 1989, dan mencapai 11% pada tahun 1993 – 1994 [1].

Pertumbuhan jumlah mahasiswa membutuhkan penanganan dalam upaya menjaga keseimbangan kualitas kelulusan pada perguruan tinggi.

Ketidaktepatan mahasiswa dalam menyelesaikan studi yang salah satunya dikarenakan terkendala dengan menyelesaikan tugas akhir (TA), mulai dari pemilihan kompetensi yang akan diajukan sebagai bahan penelitian hingga proses

penyelesaian penyusunan TA yang tidak tepat waktu.

Dalam upaya meningkatkan jumlah lulusan tepat waktu dalam menyelesaikan studinya, dilakukan beberapa kajian penelitian, mulai dari pengidentifikasian data mahasiswa, model pembelajaran, hingga pada pengklasifikasian untuk model prediksi kelulusan [2][3].

Neural network adalah salah satu cabang dari *Artificial Intelligent*. Salah satu metode dalam *neural network* adalah metode *backpropagation*. Banyak aplikasi dari *neural network* diantaranya pada penerapan untuk prediksi (*forecasting*), pengenalan pola (*pattern recognition*), identifikasi dan simulasi[4]. *Backpropagation* adalah algoritma pembelajaran yang terawasi (*supervise learning*) dan biasanya digunakan oleh *perceptron* dengan banyak lapisan (*layer*) untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya (*hidden layer*). Metode *Backpropagation* merupakan metode yang sangat baik dalam menangani masalah pengenalan pola- pola kompleks.

Penelitian yang berhubungan dengan *forecasting* dengan metode *Backpropagation* sudah banyak dilakukan, diantaranya

klasifikasi dan peramalan cuaca oleh Arti R. Naik dan Prof S.K. Pathan [5], Peramalan kecepatan angin jangka pendek dengan *feed forward* BPNN oleh K. G. Upadhyay *et.al* [6], klasifikasi bentuk tidak beraturan oleh Shih-Wei Lin *et.al* [7], identifikasi gejala penderita kolesterol melalui pola iris mata dengan metode *back propagation neural network* oleh Malakut Banu Hutomo *et.al* [8]. Dari beberapa penelitian yang sudah dilakukan dapat diambil kesimpulan bahwa algoritma BPNN dapat digunakan untuk mengatasi problem klasifikasi dan dapat digunakan untuk model *forecasting*. dengan baik.

Akurasi pada model algoritma *data mining* menentukan baik tidaknya model tersebut, model yang memiliki akurasi yang rendah dapat berakibat salah dalam mengklasifikasikan sebuah data sehingga dapat berakibat fatal jika model tersebut dipakai dalam Sistem Pendukung Pengambilan Keputusan (SPPK). Untuk meningkatkan nilai akurasi hasil dari klasifikasi dengan menggunakan algoritma BPNN akan diterapkan algoritma *bagging* pada proses *training*, untuk mengukur kinerja digunakan *confusion matrix*. Dan untuk Validasi menggunakan *X-validation*. Diharapkan dari penelitian ini akan diperoleh peningkatan nilai akurasi dari pengklasifikasian menggunakan algoritma BPNN dengan penerapan algoritma *bagging* sehingga dapat diperoleh model klasifikasi untuk tema tugas akhir pada program studi DIII Teknik Komputer Politeknik Harapan Bersama berdasarkan beberapa data nilai matakuliah tiap semester.

2. Metode Penelitian

Dalam mengimplementasikan *Neural Network* ada beberapa parameter yang harus ditentukan. Selama ini tidak ada pedoman yang baku dalam menentukan parameter *Neural Network*. sehingga metode yang dipakai adalah metode eksperimen [9]. Penelitian eksperimen ini menggunakan penerapan BPNN untuk mengklasifikasi Tema tugas akhir mahasiswa yaitu tema multimedia, *programming*, dan *hardware*.

Sumber data yang dihimpun oleh peneliti adalah data sekunder dimana data sekunder adalah informasi yang telah

dikumpulkan dan biasanya tersedia dalam bentuk dipublikasikan atau dalam media elektronik [10] yang diperoleh dari nilai matakuliah mahasiswa mulai semester 1 sampai dengan semester 5 dengan tema tugas akhirnya.

Untuk mendapatkan data yang berkualitas, ada beberapa teknik yang dapat dilakukan, antara lain:

a. Data Validation

Untuk mengidentifikasi jika terdapat data yang tidak lengkap (kosong), maka dilakukan penghapusan terhadap data tersebut. Data awal diambil dengan mengumpulkan data nilai dari angkatan 2007 hingga 2009. Kurikulum tiap-tiap tahun angkatan berbeda. Sehingga dipilih hanya matakuliah yang ada pada tiap kurikulumnya.

b. Data Size Reduction and Dicitization

Untuk memperoleh dataset dengan jumlah atribut dan *record* yang lebih sedikit tetapi bersifat informatif, dapat dilakukan dengan menghapus atribut yang tidak informatif.

Metode yang diusulkan, dataset akan dibagi menjadi 10 bagian. Secara bergantian, setiap bagian secara berurutan dijadikan sebagai data uji, sedangkan bagian lain sebagai data latih.

Adapaun data yang digunakan adalah data sekunder dimana diambil dari data BAA (Biro Administrasi Akademik) yang berupa data nilai akademik dari mahasiswa angkatan 2007 sampai dengan 2009. Data-data tersebut disesuaikan tiap-tiap kurikulumnya. Jadi nilai yang diambil adalah nilai untuk matakuliah yang ada pada tiap kurikulumnya sampai dengan semester 5.

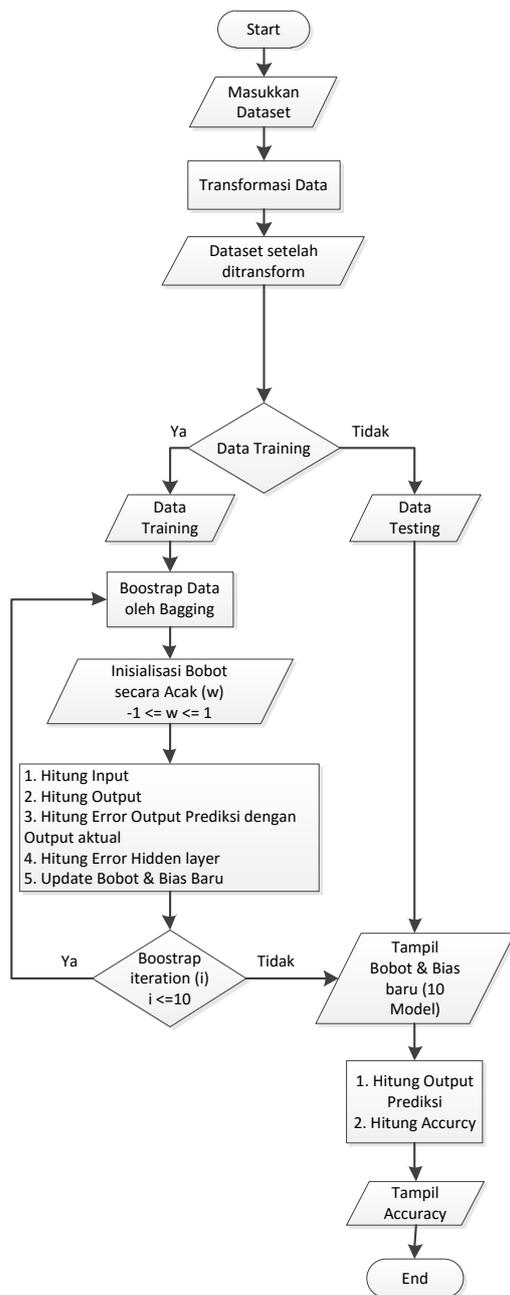
Data nilai yang sudah dikumpulkan dan sudah ditransformasikan dari nilai huruf (*string*) ke nilai bobot (*numeric*) kemudian ditransformasikan agar memenuhi standar perhitungan fungsi sigmoid biner yaitu nilai yang dimasukkan harus dalam *range* 0 dan 1. Adapun tabel normalisasi nilai bobot dapat dilihat pada tabel 1.

Metode yang diusulkan adalah *Backpropagation Neural Network* untuk tahapan klasifikasinya dan untuk peningkatan nilai akurasinya menggunakan *Bagging*

(*Bootstrap Aggregation*) dapat dilihat pada gambar 1.

Tabel 1 Normalisasi

Bobot Nilai	Hasil Transformasi [$x' = \frac{0.8(x-a)}{b-a} + 0,1$]
4	0,900
3	0,633
2	0,367
1	0,100



Gambar 1 Flowchart Model yang Diusulkan

Pada saat penentuan *Learning Rate* (LR) dan momentum serta *Training Cycle* (TC) dilakukan dengan mengubah nilai-nilai tersebut dari 0,1 hingga 0,9 untuk *learning rate* dan *momentum*, sedangkan untuk *training cycle* diawali dari nilai *default* yang diberikan rapidminer 5.2.

Dan untuk meningkatkan nilai akurasi pengklasifikasian tema tugas akhir maka diterapkan algoritma *Bagging (Bootstrap Aggregating)*. Penerapan *Bagging* pada tahap proses *training* digunakan untuk mengestimasi hasil dari pemilihan pengklasifikasi saat klasifikasi dilakukan *voting* terbanyak berdasarkan model-model yang dihasilkan.

3. Hasil dan Pembahasan

a. Pengumpulan data

Dalam penelitian ini hanya menggunakan data sekunder yang diperoleh dari sumber lain yang sudah ada, dalam hal ini adalah Biro Administrasi Akademik (BAA) Politeknik Harapan Bersama. Data sekunder yang digunakan adalah data nilai hasil studi mahasiswa angkatan 2007, 2008, dan 2009 program studi DIII Teknik Komputer Politeknik Harapan Bersama Tegal.

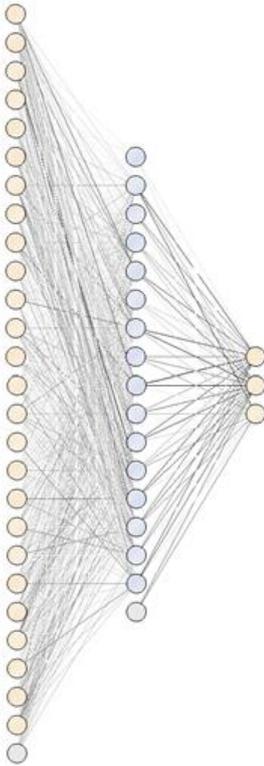
b. Proses Percobaan

Data disiapkan dalam *file excel*. Diimport dan dimasukkan dalam *repository* dari Rapidminer. Pengujian tahap pertama dilakukan dengan mengkombinasikan nilai *training cycle*, *learning rate*, dan momentum.

c. Analisis

Pada analisa arsitektur jaringan yang digunakan adalah arsitektur jaringan lapis banyak (*multilayer*).

Desain arsitektur dapat dilihat pada gambar 2, dimana desain arsitektur terdiri dari tiga lapis, yaitu satu *input layer* dengan 26 *neurons* 1 *threshold*, satu *hidden layer* dengan 16 *neurons* 1 *threshold* dan satu *output layer* dengan 3 *neuron*.



Gambar 2 Desain Arsitektur Jaringan (26-16-3)

Input berupa dataset yang merupakan nilai matakuliah yang berjumlah 26, terdiri dari matakuliah dengan kode a02, a03, a04, a05, a06, a07, a08, a09, a10, a12, a13, a14, a15, a16, a17, a20, a21, a22, a24, a26, a.27, a28, a29, a30, a31, a32, dan a33. Tiap matakuliah yang diinputkan akan diuji dan diberikan *hidden layer* oleh rapidminer dengan jumlah 16 *neurons hidden layer* dengan 1 neuron *hidden layer threshold*. Kemudian hasil outputnya akan mengarah pada ketiga *output* yaitu *hardware, programming*, dan *multimedia*.

Pertama diurutkan terlebih dahulu untuk kode matakuliah yang tidak urut dan meloncat-loncat kedalam model *x1, x2*, dan seterusnya.

Tabel 2 Normalisasi

Kode	Nama Matakuliah	Kode baru
a02	Bahasa Inggris	X1
a03	Matematika	X2
a04	Konsep Teknologi Informasi	X3
a05	Otomatisasi Perkantoran	X4
a06	Komputer Multimedia	X5
a07	Sistem Operasi	X6
a08	Algoritma dan Pemrograman	X7
a09	Desain Grafis	X8
a10	Elektronika Analog	X9
a12	Pemrograman Berorientasi Objek 1	X10
a13	Pemrograman Komputer (VB)	X11
a14	Perangkat Keras Komputer	X12
a15	Fotografi	X13
a16	Tipografi	X14
a17	Sistem Digital	X15
a20	Pemrograman Web	X16
a21	Pemrograman Berorientasi Objek 2	X17
a24	Analisa dan Perancangan Sistem	X18
a26	Animasi	X19
a27	Ilustrasi	X20
a28	CAD	X21
a29	Sistem Basis Data	X22
a30	Jaringan Komputer 1	X23
a31	Audio Visual	X24
a32	Jaringan Komputer 2	X25
a33	Branding & Corporate Identity	X26

Kemudian ambil *record* pertama untuk memulai proses penghitungan algoritma *backpropagation neural network*. Seperti pada tabel 3.

Tabel 3 Data set Record Pertama

X1	0,900
X2	0,900
X3	0,633
X4	0,633
X5	0,900
X6	0,900
X7	0,633
X8	0,633
X9	0,900
X10	0,633
X11	0,900
X12	0,900
X13	0,633
X14	0,633
X15	0,633
X16	0,633
X17	0,367
X18	0,900
X19	0,633
X20	0,900
X21	0,633
X22	0,900
X23	0,633
X24	0,633
X25	0,633
X26	0,900
Y	Multimedia

Adapun untuk tahapannya dapat diilustrasikan sebagai berikut:

Tahap Pelatihan:

- 1). Tahap 0: Inisialisasi bobot dan bias
 Pada tahapan ini bobot dipilih secara acak nilainya yaitu diantara *range* [-1, 1]. adapun bobot v_{ij} seperti tampak pada tabel 4 berikut :

Tabel 4 Penentuan bobot v_{ij} untuk 16 *neuron hidden layer*

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
x1	-0,1	0,1	0,4	0,2	0,2	0,7	0,2	0,1	0,2	0,9	0,8	0,2	-0,2	0,8	-0,1	0,1
x2	0,2	0,4	0,1	0,1	-0,2	0,4	0,2	0,9	-0,1	0,9	-0,1	0,1	0,2	0,8	-0,2	-0,2
x3	0,1	0,2	0,1	0,2	0,1	-0,1	0,2	0,4	0,1	0,8	0,2	0,1	0,2	-0,1	0,8	0,4
x4	0,2	0,1	-0,2	-0,2	0,2	-0,1	0,2	0,2	0,1	0,4	0,2	0,2	0,2	0,4	0,1	0,8
x5	0,2	-0,2	0,2	0,4	0,2	-0,1	0,2	0,1	0,2	-0,2	0,2	0,4	-0,2	0,8	-0,1	0,2
x6	0,1	0,2	0,8	0,7	0,2	0,2	0,2	-0,1	0,8	0,7	-0,2	0,2	0,4	-0,1	0,2	0,2
x7	0,4	-0,2	0,2	0,1	0,8	0,7	-0,7	0,2	0,2	-0,7	0,2	0,1	-0,2	0,2	0,4	0,8
x8	0,1	-0,2	0,4	0,1	0,2	0,1	-0,2	0,9	0,1	0,4	0,2	0,9	-0,1	0,1	-0,2	-0,2
x9	0,1	0,2	-0,1	0,4	0,1	-0,2	0,2	0,4	-0,7	0,1	0,2	0,2	0,1	0,4	0,2	0,8
x10	0,4	-0,1	0,4	0,1	0,2	-0,7	0,1	0,2	0,1	-0,2	0,2	0,1	0,2	0,4	0,1	0,1
x11	0,8	0,7	0,1	-0,1	0,1	0,1	0,1	-0,2	0,2	0,4	0,1	0,4	0,2	0,8	0,7	0,8
x12	0,2	0,8	0,9	0,8	0,7	0,8	0,7	-0,1	0,8	0,1	0,2	0,4	0,1	0,2	0,4	0,1
x13	0,1	0,2	0,2	0,8	0,2	0,1	-0,1	0,1	0,2	0,9	0,1	0,4	0,2	0,2	0,4	-0,1
x14	0,1	0,4	0,2	0,2	0,1	0,2	0,2	0,4	0,1	0,2	0,1	0,2	0,1	0,2	-0,1	0,2
x15	-0,2	0,2	0,4	-0,7	0,2	0,2	0,1	0,4	0,2	0,1	0,4	-0,1	0,8	-0,7	0,2	0,8
x16	0,2	0,1	0,2	0,8	0,1	-0,7	0,1	0,2	0,1	0,2	0,8	-0,2	0,8	-0,1	0,8	-0,1
x17	0,2	0,4	0,1	0,2	0,8	0,2	-0,7	0,2	0,4	0,1	-0,1	0,2	0,8	0,7	0,9	-0,7
x18	-0,2	0,8	-0,1	0,2	0,1	0,1	-0,2	0,1	0,4	0,2	0,2	0,2	0,8	0,9	0,8	0,8
x19	0,2	0,2	0,1	-0,1	0,2	0,1	-0,7	0,9	0,1	0,1	0,2	-0,7	0,8	-0,7	-0,2	0,1
x20	0,8	-0,1	-0,2	0,8	-0,1	0,8	0,2	0,4	0,1	0,2	-0,2	0,8	-0,1	0,1	0,2	0,2
x21	-0,2	0,8	-0,1	0,8	0,1	0,1	0,2	0,1	0,2	0,4	0,1	0,2	0,1	0,2	0,1	0,4
x22	0,4	0,2	0,8	0,9	0,2	0,8	0,2	0,9	0,8	0,2	0,1	0,2	0,4	0,2	0,4	0,1
x23	0,1	0,4	0,2	0,8	0,9	-0,2	0,1	0,2	-0,1	0,8	0,1	-0,2	0,1	0,4	0,1	0,9
x24	0,9	0,2	-0,1	0,1	0,2	0,8	0,9	0,1	-0,7	0,2	0,8	0,8	0,9	0,8	0,2	0,2
x25	0,2	-0,1	0,2	-0,1	0,1	0,2	0,1	0,8	0,7	0,8	0,7	-0,1	0,2	-0,1	-0,7	-0,2
x26	0,1	0,8	-0,1	0,8	0,9	0,1	0,4	0,2	0,2	0,2	0,8	0,9	0,1	0,4	0,2	0,2
bias	0,2	0,2	0,4	0,1	-0,2	0,4	-0,1	0,2	0,1	0,1	0,2	0,2	0,8	0,7	0,8	0,1

Sedangkan untuk bobot awal untuk *hidden layer* dengan *output* adalah ditentukan seperti pada tabel 5 dimana $Y1$, $Y2$, dan $Y3$ adalah *output* sedangkan $Z1$, $Z2$, dan seterusnya hingga $Z16$ adalah *Hidden layer*-nya.

Tabel 5 Penentuan bobot w_{ij} untuk 3 output

	Y1	Y2	Y3
Z1	-0,2	0,2	0,1
Z2	0,4	0,3	0,4
Z3	0,1	0,2	0,2
Z4	0,9	-0,3	0,8
Z5	0,3	-0,3	0,1
Z6	0,1	0,5	-0,1
Z7	-0,7	0,2	0,4
Z8	0,1	0,7	0,2
Z9	0,3	0,9	0,1
Z10	-0,1	0,6	0,3
Z11	0,7	0,1	0,1
Z12	0,6	0,3	-0,1
Z13	0,3	0,9	0,2
Z14	0,2	-0,7	0,8
Z15	0,1	0,2	0,9
Z16	0,3	0,1	-0,6
Bias	0,1	0,4	0,6

- 2). Tahap 1: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai dengan 8
- 3). Tahap 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3 sampai dengan 8

Fase I → Feed Forward

- 4). Tahap 3: Tiap unit masukan menerima sinyal dan meneruskan ke unit tersembunyi di atasnya
- 5). Tahap 4: Hitung semua keluaran di unit tersembunyi (Z_j), untuk $j=1, \dots, p$:

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

$$f(Z_{in1}) = \frac{1}{1 + e^{-(Z_{in1})}} \dots\dots\dots (2)$$

- 6). Tahap 5: Hitung semua jaringan di unit keluaran (y_k), Untuk $k=1, \dots, m$:

$$Y_{in_k} = W_{0k} + \sum_{i=1}^p Z_i W_k \dots\dots\dots (3)$$

$$Y_k = f(Y_{in_k}) = \frac{1}{1 + e^{-(Y_{in_k})}} \dots\dots\dots (4)$$

Fase II → Feed Backward

- 7). Tahap 6: Hitung faktor δ pada *neuron output* Y_k

$$\delta_k = (t_k - Y_k) f'(Y_{in_k}) \dots\dots\dots (5)$$

δ_k merupakan kesalahan yang akan dipakai dalam perubahan bobot *layer* dibawahnya (langkah 7)
 Hitung suku perubahan bobot W_{jk} (dengan *learning rate* $\alpha = 0,9$), yang akan dipakai nanti untuk merubah bobot W_{jk} sebagai berikut:

$$\Delta W_{kj} = \alpha * \delta_k * Z_j \dots\dots\dots (6)$$

- 8). Tahap 7: Hitung penjumlahan kesalahan di *hidden neuron* (δ) berdasarkan kesalahan di setiap unit tersembunyi z_j ($j=1,2,3, \dots, p$)

$$\delta_{inj} = \sum_{k=1}^m \delta_k W_j \dots\dots\dots (7)$$

Hasil dari delta *error* kemudian dihitung informasi *error* keluaran (δ) berdasarkan perkalian fungsi aktivasi dan turunannya disertai dengan perubahan bobot (ΔV) dan bias (ΔV_0) pada unit keluarannya:

$$\delta_j = \delta_{inj} * f'(Z_{in_j}) \dots\dots\dots (8)$$

$$\Delta V_{jk} = \alpha * \delta_j * X_i \dots\dots\dots (9)$$

$$\Delta V_{0j} = \alpha * \delta_j \dots\dots\dots (10)$$

Fase III Modifikasi Bobot

- 9). Tahap 8: Hitung semua perubahan bobot
 Perubahan bobot unit keluaran ($k = 1 ; j = 0, 1, \dots, 7$)

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \dots\dots (11)$$

Perubahan bobot unit tersembunyi:

$$V_{ik}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \dots\dots (12)$$

Setelah pelatihan selesai dilakukan, jaringan dapat digunakan untuk pengenalan pola. Dalam hal ini, hanya Propagasi Maju (tahap 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Penambahan Momentum

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain (*outlier*).

Dengan menggunakan suku *momentum* ($\mu = 0,9$) dan *learning rate* ($\alpha = 0,9$).

Pada tahap 8: Perubahan bobot unit keluaran dengan momentum (μ) menjadi :

$$w_{jk}(T+1) = w_{jk}(T) + \alpha \delta_k$$

$$z_j + \mu(w_{jk}(T) - w_{jk}(T-1)) \dots\dots\dots (13)$$

Keterangan:

$w_{jk}(T)$ = bobot mula-mula pola kedua (hasil iterasi pola pertama)

$w_{jk}(T-1)$ = bobot mula-mula pada iterasi pola pertama (bobot awal)

μ = *momentum*

α = *learning rate*

d. Hasil Percobaan

Percobaan dilakukan dengan menggunakan *dataset* yang sudah ditransformasi/dinormalisasi. Proses *training* dilakukan dibatasi hanya pada *Training Cycle*: 250, 500, 1000, dan 1500. Untuk *Learning rate* dan *momentum* diberikan nilai

dengan mengkombinasikan diantara keduanya mulai dari 0,1 hingga 0,9.

Adapun hasil percobaan dapat dilihat pada tabel 6 untuk hasil eksperimen tanpa menambahkan algoritma *bagging* dengan hasil nilai terendah untuk tiap-tiap *training cycle* 250, 500, 1000 dan 150. Dan tabel 7 adalah hasil dengan nilai tertinggi untuk tiap-tiap *training cycle* 250, 500, 1000 dan 1500.

Tabel 6 Accuracy Minimal default

TC	LR	Momentum	Accuracy
250	0,4	0,4	60.07%
500	0,5	0,3	60.05%
1000	0,5	0,3	59.80%
1500	0,3	0,5	60,30%

Tabel 7 Accuracy Maximal default

TC	LR	Momentum	Accuracy
250	0,9	0,9	71.17%
500	0,9	0,9	70,91%
1000	0,9	0,9	73,91%
1500	0,9	0,9	72.66%

e. Improvement dengan Bagging

Secara umum tujuan dari penerapan metode *Bagging* adalah untuk meningkatkan nilai akurasi dari suatu pengujian dataset [8]. Pengujian dilakukan dengan mengacu hasil *eksperiment* tertinggi, yaitu dengan *Training Cycle* 1000, *Learning rate* 0.9, dan momentum 0,9. Model *bagging* diletakkan pada bagian *training* data, dengan jumlah validasi 5. Diperoleh hasil akurasinya meningkat yaitu 74,14%. Diperoleh data *improvement* sebesar 74,14-73,91=0,23.

Hasil percobaan dapat dilihat pada tabel 8 untuk hasil terendah dan tabel 9 untuk hasil tertinggi dimana eksperimen menambahkan algoritma *bagging* untuk tiap-tiap *training cycle* 250, 500, 1000 dan 1500.

Tabel 8 Accuracy Minimal Bagging

TC	LR	Momentum	Accuracy
250	0,4	0,4	69,71%
500	0,5	0,3	69,95%
1000	0,5	0,3	69,46%
1500	0,3	0,5	70,94%

Tabel 9 Accuracy Maximal Bagging

TC	LR	Momentum	Accuracy
250	0,9	0,9	73,89%
500	0,9	0,9	74,38%
1000	0,9	0,9	74,14%
1500	0,9	0,9	73,64%

4. Kesimpulan

Dengan penerapan *bagging* dalam proses *training*, dapat meningkatkan nilai akurasi pengujian *dataset* nilai mahasiswa untuk klasifikasi multimedia, *programming*, dan *hardware* dengan metode *Backpropagation Neural Network*.

5. Daftar Pustaka

- [1] H.A.R. Tilaar, *Manajemen Pendidikan Nasional: Kajian Pendidikan Masa Depan*. Bandung: Remaja Rosdana Karya, 1994.
- [2] J. F. Ulysses, M. Teknik, I. Universitas, and A. Jaya, "Data Mining Classification Untuk Prediksi Lama Masa Studi Mahasiswa Berdasarkan Jalur Penerimaan Dengan Metode Naive Bayes," no. 125301917, pp. 1–8, 2008.
- [3] N. Komang, S. Julyantari, and I. K. D. Suryawan, "Data Mining Prestasi Akademik Dengan Naive Bayes Berdasarkan Attribut Importance (AI)," *J. Sist. DAN Inform.*, pp. 75–85, 2013.
- [4] Asep Sholahudd, "Penerapan Neural Network Tentang Metode Backpropagation Pada Pengenalan Pola Huruf," in *Proceedings Komputer dan sistem Intelcjen(KoMMm2002)*, 2002, pp. 82–89.
- [5] A. R. Naik and P. S. K. Pathan, "Weather Classification and Forecasting using Back Propagation Feed-forward Neural Network," *Int. J. Sci. Res. Publ.*, vol. 2, no. 12, pp. 1–3, 2012.
- [6] K. G. Upadhyay, a K. Choudhary, and M. M. Tripathi, "Short-term wind speed forecasting using feed-forward back-propagation neural network," *Int. J. Eng. Sci. Technol.*, vol. 3, no. 5, pp. 107–112, 2011.

- [7] S. Lin, S. Chou, and S. Chen, "Irregular shapes classification by back-propagation neural networks," *Int J Adv Manuf Technol*, no. 43, 2006.
- [8] M. B. Hutomo, A. Sugiarto, and E. A. Sarwoko, "Identifikasi Gejala Penderita Kolesterol Melalui Pola Iris Mata Dengan Metode Back Propagation Neural Network Seminar dan Call," *Pap. Munas Aptikom Semin. dan Call Pap. Munas Aptikom*, pp. 274–280, 2010.
- [9] G. Wu, Y. Ren, Y. Li, H. Kwak, and S. Jang, "Research on Parameter Optimization of Neural Network," *Int. J. Hybrid Inf. Technol.*, vol. 2, no. 1, pp. 81–90, 2009.
- [10] Hansen, Derek L., Ben Shneiderman, Marc A. Smith, *Analyzing Social Media Networks With Nodexl Insights from a Connected World*. Burlington: ELSEVIER, 2011.