PEMANFAATAN NEURAL NETWORK DENGAN METODE BACKPROPAGATION UNTUK MEMFILTER SPAM

Gusmelia Testiana

Dosen Program Studi Tadris Matematika Fakultas Tarbiyah IAIN Raden Fatah Palembang

Abstract : *Email Spam is email that is sent to thousands of email users in a sustainable manner. Typically, it contains unwanted promotional product, service, business, pornography, viruses and emails. This study uses Neural Network with a back propagation methods as email classification. The email will be extracted based on header that consists of content type, sender domains, the prefix which is based on the stages of extraction word, removing every stopword word, removing affixes and TFIDF calculation. The emails which are used in this study were taken from SpamAssassin. Based on this research, neural network which mostly used back propagation method are able to classify emails based on its category.*

Key words: ***email classification, email spam, neural network, back propagation method***

PENDAHULUAN

Perkembangan teknologi informasi, terutama internet yang menyediakan banyak layanan untuk pengguna, salah satunya email yang memberikan kemudahan bagi pemakainya dalam mengirimkan dan menerima pesan. Namun terdapat juga permasalahan atau gangguan bagi si pengguna email tersebut. Gangguan berupa email yang tidak diinginkan yang jumlahnya tidak hanya satu atau dua saja, tetapi bias belasan atau puluhan dalam sehari atau yang dikenal dengan spam mail, akibatnya banyak pengguna yang merasa tidak nyaman, karena mengahabiskan waktu, uang dan bandwidth. Selain pengguna internet itu sendiri, ISP (Internet Service Provider) sebagai penyedia layanan internet dan masyarakat umumnya merasa tidak nyaman dan dirugikan.

Spam mail biasanya berisikan promosi suatu barang, multi level marketing, menawarkan uang yang banyak tanpa harus bekerja dan bisa juga berisi propaganda. Melihat isinya maka spam mail biasanya digunakan oleh bagian marketing baik perusahaan online maupun perusahaan biasa untuk menawarkan produk atau jasanya dengan biaya operasi yang rendah, karena spam ini tidak memerlukan mailing list untuk mencapai para pelanggan-pelanggan yang diinginkan. Di satu sisi cara ini tidak bias disalahkan karena memang internet boleh dipergunakan sebagai tempat promosi, namun di sisi lain banyak pihak yang dirugikan, karena biasanya sangat mengganggu dan kadang-kadang membohongi.

Banyak cara yang telah dilakukan untuk menghindari spam mail diantaranya jangan pernah membalas email spam, jangan mengakses situs yang direkomendasikan spam. Hati-hati dalam menulis alamat email sebab banyak situs yang berperan sebagai kolektor email, menggunakan email lain sewaktu berkorespodensi untuk hal-hal di luar bisnis, mengaktifkan anti virus, selain itu diperlukan pemfilteran yang bertujuan untuk membedakan email spam atau bukan.

Berbagai macam teknik telah dilakukan untuk membedakan email spam ini, seperti teknik Bayesia Network yang merupakan bentuk dari probabilistic model grafik yang menyediakan basis untuk penciptaan suatu meta spam filter (Csaba Gulyas, 2006)

Neural Network merupakan bagian dari Artificial Intelligence yang dikenal sebagai machine learning yang merupakan metode dalam ilmu komputer yang melibatkan mekanisme adaptasi sehingga mesin (komputer) dapat belajar dari pengalaman. Neural Network ini diketahui memiliki kemampuan yang sangat baik untuk melakukan berbagai proses klasifikasi.

Maka dengan penjelasan di atas maka digunakan untuk memfilter email spam atau bukan dengan Neural Network dengan metode Back Propagation, karena algoritma Back Propagation sangat baik dalam menangani masalah pola-pola kompleks.

Perumusan Masalah

Berdasarkan Latar Belakang di atas maka muncul suatu masalah:

1. Bagaimana menerapkan Neural Network untuk memfilter pesan-pesan yang dikategorikan sebagai spam atau buka?
2. Bagaimana mengevaluasi kinerja dari Neural Network dalam memfilter email Spam?

Ruang Lingkup

Supaya pembahasan tidak terlalu eluas maka dibatasi pembahasan sebagai berikut:

1. Pemfilteran spam dilihat dari header dengan bagian content type berformat teks dan HTML dalam bahasa Inggris, sender domain dan awalan subjeknya.
2. Pemfilteran berfokus kepada kata pada pesan, tidak memperhitungkan lampiran dan gambar yang ada pada email.
3. Pengevaluasian kinerja Neural Network dengan Algoritma Back Propagation menggunakan Visual C#

Tujuan dan Manfaat

Tujuan dari penelitian ini adalah:

1. Menentukan parameter input yang digunakan Neural Network dengan Algoritma Back Propagation
2. Membangun sebuah prototype system dengan menggunakan Visual C# untuk mengevaluasi kinerja dari Neural Network dengan Algoritma Back Propagation untuk memfilter Email Spam.

Manfaat dari penelitian ini adalah:

1. Mempermudah pengguna email untuk menyaring jenis pesan yang diterima.
2. Menghindari pemakaian waktu, uang dan bandwidth yang banyak.

LANDASAN TEORI

1.1 Neural Network

Neural network atau dikenal juga dengan Jaringan Syaraf Tiruan adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut (Sri Kusumadewi, 2003). Sedangkan menurut Agustinus Noertjahyana dan Julia (2002), *Neural network* adalah sistem pengolah informasi yang mempunyai karakteristik mirip dengan jaringan syaraf biologis, yaitu jaringan syaraf pada otak manusia.

2.2 Algoritma *Backpropagation*

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 sigmoid, yaitu:

$$f(x) = \frac{1}{1+e^{-x}} \qquad (2.7)$$

a. Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).

b. Kerjakan langkah-langkah berikut selama kondisi berhenti bernilai *false*.

1. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

Feedfoward:

a. Tiap-tiap unit *input* (X_i i=1,2,3,…,n) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
b. Tiap-tiap unit tersembunyi (Z_i j=1,2,3,…,p) menjumlahkan sinyal-sinyal *input* terbobot:

$$z_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \qquad (2.8)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$z_j = f(z_in_j) \qquad (2.9)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*)

c. Tiap-tiap unit *output* (Y_k, k=1,2,3,...,m) menjumlahkan sinyal-sinyal *input* berbobot.

$$y_in_k = w_{0k} + \sum_{i=1}^{p} z_i w_{jk} \qquad (2.10)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output*nya:

$$y_k = f(y_in_k) \qquad (2.11)$$

Dan kirimkan sinyal tersebut ke semua unit di lapisan atasnya (unit-unit *output*).

Backpropagation

d. Tiap-tiap unit *output* (Y_k, k=1,2,3,...,m) menerima target pola yang berhubungan dengan pola *input* pembelajran, hitung informasi errornya:

$$\delta_k = (t_k - y_k) f'(y_in_k) \qquad (2.12)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

$$\Delta w_{jk} = \alpha \delta_k z_j \qquad (2.13)$$

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

$$\Delta w_{0k} = \alpha \delta_k \qquad (2.14)$$

Kirimkan δ_k ini ke unit-unit yang addi lapisan bawahnya.

e. Tiap-tiap unit tersembunyi (Z_j, j=1,2,3,...,p) menjumlahkan delta *input*nya (dari unit-unit yang berada pada lapisan di atasnya):

$$\delta_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \qquad (2.15)$$

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

$$\delta_j = \delta_in_j f'(z_in_j) \qquad (2.16)$$

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

$$\Delta v_{jk} = \alpha \delta_j x_i \qquad (2.17)$$

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

$$\Delta v_{0j} = \alpha \delta_j \qquad (2.17)$$

f. Tiap-tiap unit *output* (Y_k, k=1,2,3,...,m) memperbaiki bias dan bobotnya (j=0,1,2,...,p):

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \qquad (2.18)$$

Tiap-tiap unit tersembunyi (Z_j, j=1,2,3,...,p) memperbaiki bias dan bobotnya (i=0,1,2,...,n):

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij} \qquad (2.19)$$

2. Tes kondisi berhenti

2.3 *Email*

Email (*electronic mail*) adalah penyampaian pesan dan *file* melalui suatu jaringan komputer (Selly, *et al*, 2007). Saat ini, email merupakan metode komunikasi utama untuk penggunaan perorangan maupun bisnis. Namun terdapat juga permasalahan atau gangguan bagi si pengguna *email* tersebut. Gangguan berupa *email* yang tidak diinginkan yang jumlahnya tidak hanya satu atau dua saja, tetapi bisa belasan atau puluhan dalam sehari atau yang dikenal dengan *spam mail*.

2.4 *Spam*

Spam merupakan *email* atau informasi pada kelompok warta (*newsgroup*) yang tidak diinginkan yang dikirim ke banyak penerima atau kelompok warta sekaligus. *Spam* merupakan surat sampah pada *internet*. Isi *spam* bervariasi, mulai dari menjual produk atau layanan, mempromosikan kesempatan bisnis, hingga mengiklankan materi yang tidak sopan (Selly, *et al*, 2007).

2.5 Pemprosesan Text

Pemprosesan text adalah suatu prosedur dalam mengolah dokumen teks di mana dapat dibagi ke dalam lima hal:

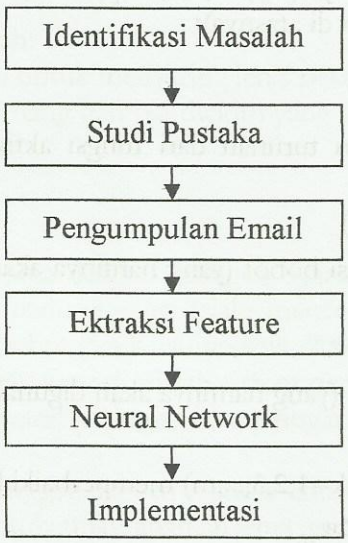
1. *Lexical analysis*, yaitu berhubungan dengan perlakukan terhadap teks dalam hal angka, tanda hubung, tanda baca dan huruf kapital atau bukan.
2. Membuang *stopword* yaitu melakukan *filtering* terhadap kata-kata yang sering muncul dalam dokumen.
3. *Stemming* yaitu pembuangan imbuhan (prefix dan suffix) dari sebuah kata sehingga didapatkan bentuk kata dasarnya.
4. Penyeleksian dari *index* kata untuk menentukan kata apa saja yang akan digunakan dalam mengindeks.
5. Konstruksi dari struktur istilah seperti kamus (*thesaurus*).

Pemprosesan text dilakukan untuk pengekstrak fitur atau ciri dari suatu *email spam*. Fitur-fitur tersebut menjadi suatu *wordlist* yang akan menjadi acuan sistem yang dalam mengidentifikasi *email spam*.

METODOLOGI PENELITIAN

Kerangka Kerja

Gambar 3.1 merupakan kerangka kerja (*Frame Work*) yang digunakan dalam penelitian ini:



Gambar 3.1 Kerangka Kerja

Uraian Kerangka Kerja

Uraian kerangka kerja adalah isi dari masing-masing kerangka kerja, mulai dari identifikasi masalah, studi pustaka, pengumpulan email, ekstraksi feature, Neural Network dan implementasi.

1. Identifikasi Masalah

   Identifikasi masalah merupakan tahap awal dari penyusunan penelitian ini. Masalah yang dihadapi adalah bagaimana memfilter email yang dapat dikategorikan sebagai spam atau bukan.

2. Studi Pustaka

   Studi pustaka yang dilakukan meliputi pengetahuan tentang spam, email spam filtering, pemrosesan teks, data processing, indexing, ekstrasi cirri (feature), pengetahuan tentang Neural Network khususnya teknik back propagation.

3. Pengumpulan Email

   Email yang digunakan dalam penelitian ini diambil dari http://spamassasin.apache.org/publiccorpus.

4. Ektraksi Feature
   Fitur yang akan diekstrak dari email adalah bagian header yang mengandung informasi seperti pengirim, penerima, tanggal pengiriman dan lain-lain, dan bagian isi pesan dari email tersebut. Ke dua bagian ini mengandung informasi yang digunakan untuk membedakan antara spam atau bukan spam.
5. Neural Network
   Struktur Neural Network yang digunakan adalah Neural network yang terdiri dari tiga layer (input layer, hidden layer dan output layer). Jumlah neuron pada input layer disesuaikan dengan jumlah data vector, sedangkan untuk hidden layer dibuat variasi jumlah neuron untuk menguji mana yang lebih baik. Output layer akan merepresentasikan email spam dan bukan spam. Algoritma pembelajaran yang digunakan adalah back propagation. Neural Network ini digunakan untuk dapat membedakan antara spam atau bukan spam (ham)
6. Implementasi
   Pada tahap implementasi untuk melakukan filtering email spam dengan menggunakan Visual C#
   Bahan Penelitian
   Bahan dari penelitian ini adalah kumpulan email-email yang di ambil dari http://spamassasin.apache.org/publiccorpus.
   Dan peralatan yang digunakan diantaranya:
   1. Komputer PC dengan Spesifikasi minimum Pentium 4 3GHz dengan RAM 1GB dan Harddisk 160GB
   2. Perangkat Lunak Visual C#

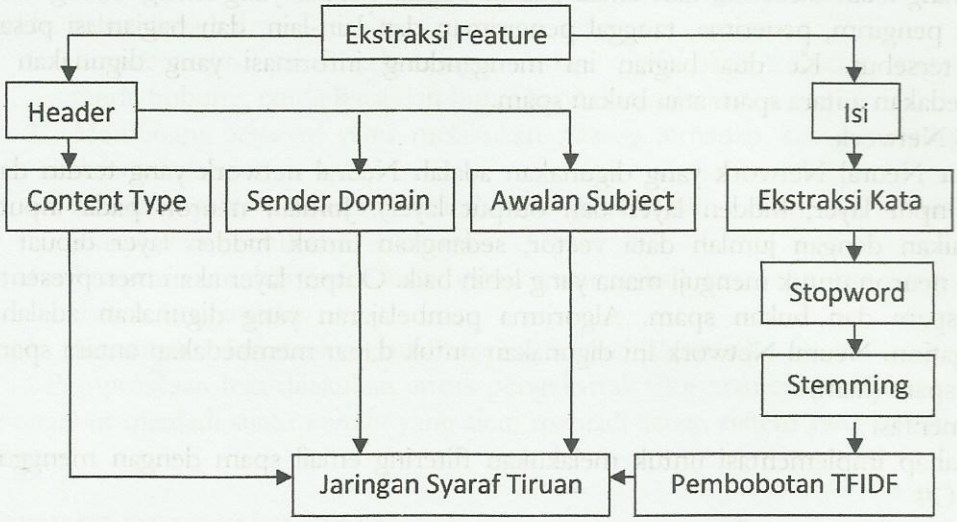
4.1 Analisis Sistem

Analisa sistem merupakan suatu uraian dari sistem informasi yang utuh menjadi suatu bagian yang terbentuk dari komponen yang berguna untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan suatu perbaikan dalam penggunaan *neural network* untuk memfilter *email spam.*

Sebelum *spam filtering* dapat digunakan, pertama harus dilakukan proses *training* terhadap kumpulan *email-email* yang telah digolongkan sebagai *spam* dan bukan *spam* (*ham*) pada jaringan syaraf tiruan. Sehingga sistem yang akan dibangun dapat mendeteksi apakah sebuah *email* digolongkan sebagai *spam* dan *ham.*

Sistem yang akan dirancang ini melewati beberapa langkah pemrosesan teks yang akan dikategorikan berdasarkan *header*, yang terdiri dari bagian *content type*, *sender domain* dan awalan subjek. Dan berdasarkan isi dari *email*, ekstraksi kata, *stopword*, *stemming*, pembobotan kata dengan metode TFIDF. Dari *header* dan isi *email* ini akan dijadikan *input* untuk *neural network.*

Aliran kerja sistem dapat dilihat pada Gambar 4.1.



Gambar 4.1 Aliran kerja sistem

4.1.1 Ekstraksi *Feature*

Untuk melakukan *ekstraksi feature* dengan menggunakan salah satu *email spam.*

Date: Wed, 05 Jun 2002 23:11:54
To: yyyy@netnoteinc.com
Subject: URGENT HELP
MIME-Version: 1.0
Content-Type: text/plain;charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-Keywords:
Dear,sir
I am Chief Tony Anenih,the minister of Works and Housing under the government of President Olusegun Obasanjo. I am 51 years of age, married with one wife and five children. I have served my beloved country Nigeria as a minister of Works and Housing for some years now.

Gambar 4.2 Contoh *Email Spam* untuk Analisa

Dari contoh *email* di atas akan dilakukan *ekstraksi feature.* Fitur yang akan diekstrak dari *email* adalah bagian *header* yang mengandung informasi seperti pengirim, penerima, tanggal pengiriman dan lain-lain. Pada bagian *header* fitur yang akan diekstrak yaitu *content type*, *sender domain*, awalan subjek. Dan bagian isi pesan dari *email* tersebut dimulai dengan ekstraksi kata sampai pembobotan kata dengan metode TFIDF kemudian di proses oleh jaringan syaraf tiruan.

4.1.2 *Header*

Dari bagian *header* yang mengandung informasi seperti pengirim, penerima, tanggal pengiriman dan lain-lain.

Fitur yang akan diekstrak dari bagian *header* adalah sebagai berikut:

1. *Content type*
   Pada bagian header ini akan diperiksa tipe dari isi pesan, apakah hanya teks biasa atau berisi tag-tag HTML.
2. *Sender Domain*
   Melakukan ekstraksi dari alamat pengirim *email*, apakah *domain*nya *com* atau *domain* yang lain.
3. Awalan Subjek

Sistem akan melakukan ekstraksi pada subjek *email*, apakah diawali oleh kata "Re" yang menunjukkan *email* tersebut balasan dari seseorang atau tidak.

Dari contoh *email* pada Gambar 4.2 diperoleh hasil bagian *header*nya sebagai berikut:

Tabel 4.1: Hasil dari Bagian *Header*

Content Type	*Sender domain*	Awalan subjek
Text	mail.com	-

4.1.3 Isi

Untuk proses ekstrkasi *feature* selanjutnya dilihat dari bagian isi. Pada bagian ini terlebih dulu dibuat *wordlist* yang berisi kata-kata yang sering digunakan oleh *email spam*.

4.1.4 Ekstraksi Kata

Hasil ekstraksi kata dari isi email pada Gambar 4.2 ada pada Tabel 4.2 berikut ini:

Tabel 4.2: Hasil Ekstraksi Kata

I	Children	Money	Look	Me
Am	Have	In	Into	Detail
Chief	Served	Any	Real	About
Tony	My	Lucrative	Estate	How
Anenih	Beloved	Or	Property	Can

4.1.6 Proses *Stemming*

Proses berikutnya melakukan proses *stemming* pada masing masing kata tersebut. *Stemmer* yang digunakan adalah *Porter's Stemmer* ada pada lampiran. Sehingga kata-kata tersebut terlihat pada Tabel 4.4 berikut:

Tabel 4.4: Kata Hasil *Stemming*

Kata Awal	Kata Setelah Proses Stemming	Kata Awal	Kata Setelah Proses Stemming
Chief	Chief	Professional	Profess
Tony	Tony	Pursiut	Pursiut
Anenih	Anenih	Personally	Personal

4.1.7 Pembobotan TFIDF

Langkah selanjutnya adalah melakukan pembobotan kata dengan teknik TFIDF. Algoritma TFIDF dapat dikembangkan sebagai berikut:

1. Membaca seluruh kata dalam *email*
   Email yang akan dibaca diambil dari *spamassassin* yang akan digunakan dalam percobaan. *Email* yang digunakan dikategorikan menjadi *email spam* dan bukan spam (*ham*).

2. Menghitung kemunculan setiap kata dalam *email* (TF)

$$TF(k) = \frac{\textit{Jumlah kata k disebuah email}}{\textit{Jumlah kata dalam email}} \quad (4.1)$$

3. Menghitung banyaknya *email* yang mengandung kata k (DF)

$$DF(k) = \frac{\textit{Jumlah email yang mengandung kata k}}{} \quad (4.2)$$

4. Mengitung TFIDF

$$TFIDF(k) = TF(k) \times IDF \quad (4.3)$$

$$IDF = \log(1 + \frac{nDoc}{DF(k)}) \quad (4.4)$$

$$TFIDF(k) = TF(k) x \log(1 + \frac{nDoc}{DF(k)}) \quad (4.5)$$

dimana *TF* adalah jumlah munculnya kata *k* dalam satu *email* dibagi dengan jumlah kata pada email, *DF* adalah jumlah email yang mengandung kata *k* dibagi dengan total jumlah email dan *nDoc* adalah jumlah *email* yang banyaknya berbeda untuk masing-masing percobaan. Untuk perhitungan *TFIDF* dapat digunakan program pada lampiran. Pada Tabel 4.5 dapat dilihat hasil pembobotan kata dari *email*.

Tabel 4.5 Hasil Pembobotan Kata

Kata	TF	IDF	TF-IDF	nDoc
Chief	0.125	4.094345	0.5117931	240
Tony	0.375	3.871201	1.4517	240
Anenih	0.125	5.480639	0.6850799	240
Minister	0.25	4.787492	1.196873	240
Works	0.02702703	4.094345	0.110658	240

Setelah melakukan pembobotan kata, maka semua kata tersebut diurut berdasarkan abjad dan diberi nomor, dapat dilihat pada Tabel 4.6 berikut:

Tabel 4.6 Hasil Pembobotan Kata yang Telah Terurut

Kata-ke	Kata	TF	IDF	TF-IDF	nDoc
1	*Anenih*	0.125	5.480639	0.68508	240
2	*Chief*	0.125	4.094345	0.51179	240
3	*Minister*	0.25	4.787492	1.19687	240
4	*Tony*	0.375	3.871201	1.4517	240
5	*Works*	0.02702703	4.094345	0.11066	240

4. 2 Analisa Kebutuhan

4.2.1 *Input*

Data yang dibutuhkan sebagai data *input* pada penelitian ini adalah:

1. *Content Type*
2. *Sender Domain*
3. Awalan Subjek
4. TFIDF
5. TF
6. DF
7. n(*email*)
8. k

Nilai dari data *input* dapat diambil dari hasil perhitungan yang telah dilakukan oleh pada Tabel 4.6 di atas. Data *input* agar dapat di olah maka diinisialisasi terlebih dahulu dengan variabel *input* sebagai berikut:

1. X1 = *Content Type*
   Text = 1
   Html = 0
2. X2 =*Sender Domain*
   Domain com = 1
   Domain lain = 0
3. X3 =Awalan Subjek
   Berawalan = 1
   Tidak Berawalan = 0
4. X4 = TFIDF
   Kata terbobot *spam* = 1
   Kata terbobot *ham* = 0
5. X5 = TF
   Kata yang sering muncul = 1
   Kata yang jarang muncul =0
6. X6 = IDF
   Mengandung kata k = 1
   Tidak mengandung kata k = 0
7. X7 = n(*email*)
   Jumlah *email* = 1

8. X8 = k

Kata k = 1

Kata bukan k = 0

Target dari *filtering email spam* adalah dengan melihat dari masing-masing perubahan yang terjadi pada masing-masing masukan. Adapun model data untuk pelatihan adalah sebagai berikut :

Tabel 4.7 Model Data untuk Pelatihan

No	Data Nilai Input								
	X1	X2	X3	X4	X5	X6	X7	X8	Target
1	1	1	1	1	1	1	1	1	1
2	1	1	1	0	1	0	1	0	0
3	0	0	0	1	1	1	1	1	1
4	1	0	0	1	1	1	1	0	1
5	0	1	1	0	0	0	0	1	0

4.2.2 Proses

Pada sistem pemfilteran *email* ini jalur proses kerjanya menggunakan *neural network* menerapakan alur kerja dari arsitektur jaringan *multilayer* dengan metode *backpropagation. Neural network* yang terdiri dari tiga lapisan yaitu: *input layer* yang terdiri dari delapan masukan yaitu *content type*, *sender domain*, awalan subjek, bobot kata (TFIDF), jumlah kata (TF), jumlah *email* yang mengandung kata tersebut (IDF), jumlah email (n) dan kata itu sendiri (k), *hidden layer* terdiri dari tiga sel syaraf dan *output layer* terdiri dari dua sel syaraf. Lapisan masukan digunakan untuk menampung variabel yaitu X1, X2, X3, X4, X5, X6, X7, X8, sedangkan dua lapisan keluaran digunakan untuk merepresentasikan pemfilteran dengan dua kategori.

4.2.3 *Output*

Output dari sistem pemfilteran *email spam* ini adalah dari *email* yang ditampung dalam variabel Y1 dan Y2 sebagai berikut:

1. Y1 = *spam*
2. Y2 = bukan *spam* (*ham*)

Hasil *output* yang diinginkan adalah:

1. Y1 = *spam* = 1
2. Y2 = bukan *spam* (*ham*) = 0

4.3 Cara Kerja *Backpropagation*

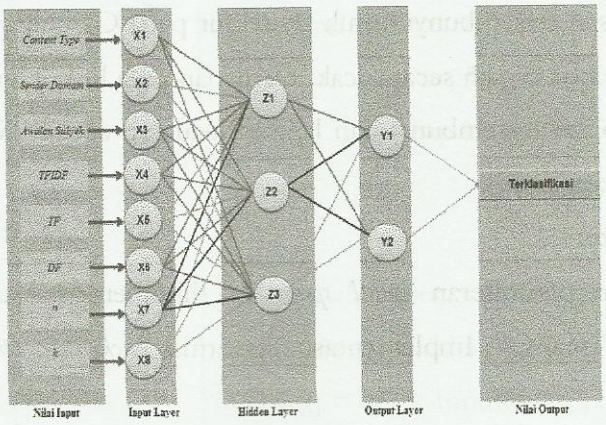
Cara kerja *neural network* dengan metode *backpropagation* yaitu *gradient* unit-unit yang tersembunyi diturunkan dari *error* yang disosialisasikan dengan unit *output*. Hal ini diberikan karena nilai target *hidden neuron* tidak diberikan

4.3.1 Arsitektur *Neural Network* yang Didesain

Arsitektur jaringan syaraf tiruan yang digunakan adalah *backpropagation.* terdiri dari:

1. Lapisan *input* dengan 8 simpul masing-masing untuk nilai *content type* (X1), *sender domain* (X2), awalan subjek (X3), TFIDF (X4), TF (X5), IF (X6), DF (X7), n (X8), k (X9).
2. Lapisan tersembunyi dengan jumlah simpul 3 simpul, yang ditentukan sendiri.
3. Lapisan *output* dengan 2 simpul yaitu *spam* dan bukan *spam* (*ham*).

Sehingga arsitekturnya dapat dilihat pada Gambar 4.3 berikut:



Gambar 4.3 Arsitektur *Neural Network* yang Didesain

Keterangan :

$X_1 - X_8$ = *node* pada *input layer*

Z_1, Z_2, Z_3 = *node* pada *hidden layer*

Y_1, Y_2 = *node* pada *output layer*

4.3.2 Perancangan *Neural Network* dengan Algoritma Backpropagation

Pada tahap perancangan *neural network* dengan algoritma *backpropagation* ini hasil analisis yang telah di dapatkan akan diterapkan pada langkah-langkah algoritma *backpropagation* dengan menggunakan fungsi aktivasi sigmoid, sebagai berikut:

1. Tahap *Initialization*

   Merupakan tahapan untuk mendefenisikan atau menset awal nilai untuk variabel-variabel yang diperlukan seperti nilai *input*, *weight*, *output* yang diharapkan, *learning rate* (α), θ dan sebagainya.

2. Tahap *Activation*

   Pada tahap ini akan dilakukan dua kegiatan yaitu menghitung *actual output* pada *hidden layer* dan menghitung *actual output* pada *output layer.*

3. Tahap *Weigth Training*

   Pada tahap ini dilakukan dua kegiatan yaitu menghitung *error gradient* pada *output layer* dan menghitung *gradient* pada *hidden layer.*

4. Tahap *Iteration*

Tahap terakhir ini adalah tahapan pengujian dimana *error* yang diharapkan belum ditemukan maka akan kembali lagi kepada ketahapan ke dua *activation*.

Arsitektur yang dipilih dimisalkan seperti pada Gambar 4.4 dengan jumlah simpul pada lapisan masukan yaitu 2 dari 8 data input, dimana masing-masing lapisannya sebagai contoh adalah X1, X2. Jumlah simpul pada lapisan tersembunyi ada 2. Jumlah simpul pada lapisan keluaran 1 yang digunakan untuk mempresentasikan pola. Untuk membentuk jaringan syaraf tiruan, terlebih dahulu dilakukan inisialisasi bobot awal. Bobot awal yang menghubungkan simpul-simpul pada lapisan masukan dan lapisan tersembunyi untuk arsitektur pada Gambar 4.4 adalah W= (W13,W14, W23, W24) dan bobot bias dipilih secara acak. Demikian pula bobot awal yang menghubungkan simpul-simpul pada lapisan tersembunyi dan lapisan keluaran adalah W= (W35, W45) dan bobot biasnya juga dipilih secara acak.

4.3.3 Penerapan Algoritma *backpropagation*

Algoritma *backpropagation* untuk sistem pemfilteran *email spam* ini diimplementasikan dengan menggunakan bahasa pemrograman Visual C#. Implementasi Algoritma *backpropagation* adalah sebagai berikut :

Nama proses	Backpropagation
Deskripsi	Menghitung perubahan nilai bobot dan bias berdasarkan nilai error

Logika proses	int i, j;
	double total;
	//Perbaiki error hidden layer
	for (i = 0; i < HiddenNum; i++) {
	total = 0.0;
	for (j = 0; j < OutputNum; j++){
	total += HiddenLayer[i].Weights[j] *
	OutputLayer[j].Error;
	}
	HiddenLayer[i].Error = total;
	}
	//Perbaiki error input layer
	for (i = 0; i < InputNum; i++){
	total = 0.0;
	for (j = 0; j < HiddenNum; j++){

```
            total += InputLayer[i].Weights[j] *
                HiddenLayer[j].Error;
        }
        InputLayer[i].Error = total;
    }
    //Merubah bobot pada input layer
    for (i = 0; i < InputNum; i++){
        for (j = 0; j < PreInputNum; j++){
            PreInputLayer[j].Weights[i] +=
                learningRate * InputLayer[i].Error *
                PreInputLayer[j].Value;
        }
    }
    //Merubah bobot pada hidden layer
    for (i = 0; i < HiddenNum; i++){
        for (j = 0; j < InputNum; j++) {
            InputLayer[j].Weights[i] +=
                learningRate * HiddenLayer[i].Error *
                InputLayer[j].Output;
        }
    }
    //Merubah bobot pada output layer
    for (i = 0; i < OutputNum; i++){
        for (j = 0; j < HiddenNum; j++){
            HiddenLayer[j].Weights[i] +=
                learningRate * OutputLayer[i].Error *
                HiddenLayer[j].Output;
        } }
```

4.3.3 *Training*

Pada penelitian ini digunakan 2926 *email* sebagai data *training* (1829 *ham* dan 1097 *spam*). Proses *training* akan menggunakan aplikasi Visual C# dengan proses:

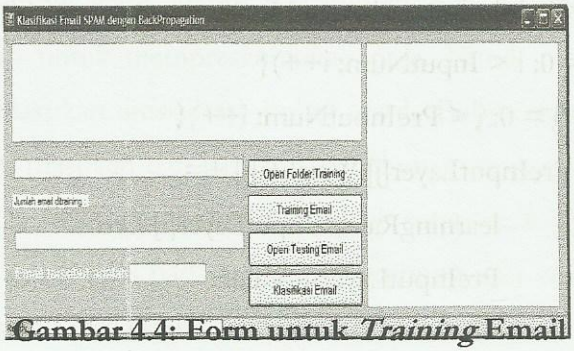
1. *Open Folder Training*

   Membuka file *training* yang telah ada, baik jenis ham dan spam.

2. *Training* email

Melakukan *training* terhadap email yang ada, sehingga proses *training* dapat dijalankan dan dapat melihat jumlah email yang telah di *training*.

Pada Gambar 4.4 merupakan bentuk form untuk melakukan proses *training* terhadap *email spam* dan *ham*.



Gambar 4.4: Form untuk *Training* Email

KESIMPULAN DAN SARAN

a. Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan beberapa hal yaitu:

1. Neural network dapat mengidentifikasi suatu email apakah terdapat email spam atau bukan.
2. Dalam melakukan pemfilteran terhadap email spam, maka cirri yang dieksekusi sebagai input bagi neural network adalah bagian header yang terdiri dari content type, sender domain dan awalan subject dan bagian isi dengan ektraksi kata, stopword, stemming dan pembobotan kata dengan TFIDF.
3. Performance sistem bergantung dari keberagaman email yang ada pada proses training. Makin beragam email yang ada pada proses training, makin baik performansi sistem.
4. Tingkat keberhasilan dari metode Backpropagation sangat baik dalam memfilter email spam. Namun masih ada jenis email yang tidak sesuai dengan kategorinya.
5. Persentasi keberhasilan yang diperoleh setelah pengujian untuk membedakan email spam atau bukan dengan metode neural network sebesar 99,5%.

b. Saran

1. Untuk pengembangan selanjutnya lakukan perbandingan antara penggnaan neural network dengan teknik lainnya.
2. Memperhitungkan attachment dan gambar, di mana saat ini banyak email spam menyelipkan kata-kata yang diidentifikasi sebagai spam ke dalam bentuk gambar.

3. Pengembangan selanjutnya email yang dibedakan tidak hanya berformat bahasa Inggris, tetapi berbahasa Indonesia.

DAFTAR PUSTAKA

Drucker *et al.* 1999. *Support Vector Machine for Spam Categorization.* IEEE Transactions on Neural Networks, Vol. 10, No. 5.

Metsis *et al.* 2006. *Spam Filtering with Naive Bayes – Which Naive Bayes?.* Di dalam: CEAS 2006 - *Third Conference on Email and AntiSpam.*

Marvin Chandra Wijaya, Semuil Tjiharjadi, 2008 Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008) Auditorium Universitas Gunadarma

Sulaiman, Agus; 2007; *Koneksi Database Dengan ADODC*; Jakarta.