1. Pendahuluan

1.1 Latar belakang

Saat ini, membaca *e-mail* sudah menjadi keperluan bagi banyak orang mulai dari kalangan profesional sampai kalangan pribadi. Alasan banyak orang menggunakan *e-mail* antara lain karena cepat, murah, dan efisien. Tetapi adakalanya kita menerima *e-mail* dari sumber yang tidak jelas dan yang tidak kita inginkan. *E-mail* inilah yang disebut *spam* [4].

Pada dekade yang lalu, *worms* menjadi masalah utama bagi para pengguna *e-mail*. Tetapi dalam beberapa tahun terakhir ini, *spam* 'menyerbu' layanan internet paling banyak digunakan ini (*e-mail*) [4]. Pada Desember 2003, BBC News memperkirakan sekitar 40% dari semua *e-mail* yang terkirim diidentifikasi sebagai *spam*, dan proses identifikasi dan penghapusan *spam* oleh perusahaan-perusahaan di Inggris memakan waktu rata-rata 1 jam/pekerja/hari. Para *spammer* melakukan penyebaran spam dengan cara tiap pesan dikirim sekaligus dalam jumlah yang sangat besar. Hal itu dilakukan dengan harapan sekian presentase dari pesan (yang mayoritas berisi iklan suatu produk) yang terkirim akan merespon balik dan membeli produknya. The Wall Street Journal memperkirakan bahwa tingkat respon sebesar 0.0001% sudah dapat menghasilkan *profit* bagi *spammer* tersebut. Oleh karena itu, tidak heran jika *spam* dapat dijadikan sarana promosi produk yang efektif, cepat, dan tentu saja murah.

Berbagai macam strategi telah dicoba untuk menyelesaikan masalah ini. Secara garis besar strategi-strategi tersebut dapat dibagi menjadi 2 teknik, yaitu teknik *prevention* dan *cure*. Teknik *prevention* bertujuan untuk mencegah terkirimnya *spam* dengan mengimplementasian sejumlah kendali dan pemeriksaaan pada sistem *e-mail* global yang akan menyulitkan para *spammer* dalam mengirim pesan dalam jumlah besar. Tetapi masalah utama teknik ini adalah implementasi kendali pada sistem global yang kompleks. Teknik lainnya yaitu *cure* yang bertujuan mencegah masuknya *spam* ke *inbox* si penerima. Teknik ini biasa dikenal sebagai *spam filtering*. Cara kerjanya yaitu dengan mem-*filter spam* menggunakan sejumlah *rule* (aturan). Jika kita sudah memiliki aturan yang jelas untuk membedakan *spam* atau bukan, maka kita tinggal menggunakan aturan tersebut dengan memanfaatkan pencocokan pola. Tetapi, tentu saja akan merepotkan jika kita harus memperbaharui aturan tersebut agar tetap berlaku untuk menyaring pesan baru dengan strategi *spamming* yang semakin bervariasi dan "kreatif".

Oleh karena itu, diperlukan metode *Artificial Intelligence* (AI) yang memungkinkan kita membuat daftar aturan secara otomatis berdasarkan data-data yang dimiliki. Metode AI yang paling cocok untuk kasus ini adalah dengan menggunakan teknik *learning*. Dengan teknik ini, kita dapat secara otomatis menemukan aturan yang diharapkan dapat berlaku umum untuk data-data yang belum pernah diketahui [1].

Salah satu *learning algorithm* yang dapat dipakai untuk membangkitkan aturan ini adalah *Grammatical Evolution* (GE). GE adalah metode yang dapat merepresentasikan solusi berbasiskan *grammar*. *Grammar* dalam metode ini

menggunakan notasi *Backus Naur Form* (BNF) [3]. BNF dapat dikodekan dengan mudah dan mewakili semua bahasa [2]. *Grammar* yang dibangun akan berisikan elemen-elemen yang dapat membentuk aturan-aturan untuk kemudian digunakan dalam *spam filtering*.

1.2 Perumusan masalah

Dari uraian di atas dapat dirumuskan beberapa permasalahan utama, yaitu:
1. Bagaimana menentukan atribut yang akan dimasukkan ke dalam daftar aturan.
2. Bagaimana menentukan *grammar* yang cocok untuk kasus *spam filtering*.
3. Bagaimana memanfaatkan metode GE untuk membangkitkan aturan.
4. Bagaimana menentukan fungsi fitness yang tepat.
5. Bagaimana melakukan pre-processing terhadap data uji.

Hipotesa yang akan dibuktikan yaitu bahwa *grammar* yang mengandung elemen yang lebih kompleks menghasilkan akurasi yang lebih tinggi.

Batasan masalah yang digunakan dalam menyelesaikan permasalahan di atas antara lain sebagai berikut:
1. Data uji menggunakan *corpus SpamAssasin* yang berisikan *spam* dan *ham* yang berasal dari http://spamassassin.apache.org/publiccorpus/.
2. Aturan yang digunakan terdiri dari beberapa atribut statistik, seperti frekuensi kemunculan kata/karakter, jumlah resipien, dsb.
3. *Spam* dan *ham* diletakkan di direktori yang berbeda agar dapat digunakan sebagai acuan untuk menghitung akurasi klasifikasi.
4. Parameter performansi diukur berdasarkan:
   - Akurasi yaitu presentase dari jumlah klasifikasi data yang cocok dibagi jumlah sampel data total. Semakin tinggi akurasi semakin baik.
   - *False positive* yaitu kesalahan identifikasi ham sebagai spam. Semakin rendah *false positive* semakin baik.

1.3 Tujuan

Tujuan dari pembuatan TA ini adalah sebagai berikut:
1. Menerapkan metode GE untuk *spam filtering*.
2. Menganalisa pengaruh penentuan *grammar* terhadap akurasi *spam filtering*.
3. Menganalisa pengaruh penentuan parameter evolusi terhadap akurasi *spam filtering*.

1.4 Metodologi penyelesaian masalah

Ada beberapa tahap yang dilakukan dalam menyelesaikan masalah ini, yaitu:
1. Pengumpulan atribut
   Dalam tahap ini dilakukan studi data untuk menentukan atribut-atribut mana saja yang nantinya akan dimasukkan ke dalam rule.
2. Penentuan parameter dan grammar
   Parameter-parameter evolusi seperti jumlah populasi, generasi maksimum, crossover rate, mutation rate, dll ditentukan dalam tahap ini. Pembangunan grammar yang cocok dengan masalah juga dilakukan.
3. Pembangkitan rule dengan GE

Proses ini disebut juga dengan proses pelatihan, yaitu menggunakan sejumlah data latih dari corpus untuk kemudian dijadikan acuan pengukuran akurasi sementara. Akurasi tersebut dihitung dengan menggunakan acuan rule sementara pada data latih. Nilai ini digabungkan dengan nilai false positive sementara kemudian dijadikan sebagai nilai fitness. Tahap ini akan menghasilkan satu rule yang akan dijadikan acuan untuk tahap klasifikasi.

4. Klasifikasi
   Tahap ini membutuhkan data uji dan rule. Data uji berasal dari corpus dengan jumlah tertentu dan rule berasal dari tahap sebelumnya.
5. Analisa hasil
   Hasil klasifikasi kemudian dianalisa performansinya berdasarkan parameter evolusi dan grammar.