Sistem Peringatan Dini Untuk Mendeteksi Spam Pada Email Menggunakan Domain Name System-Based Blackhole List (DNSBL) Filter Dan Support Vector Machine (SVM) Filter

*Early Warning System For Detecting Spam In Email Using Domain Name System-Based Blackhole List (DNSBL) Filter And Support Vector Machine (SVM) Filters*

Muhammad Hafidz1, Fahri Firdausillah2

Teknik Informatika S1, Universitas Dian Nuswantoro

E-mail: 1111201509000@mhs.dinus.ac.id, 2fahri.firdausillah@dsn.dinus.ac.id

Abstrak

*Email merupakan media komunikasi bagi pengguna dan penyedia jasa Internet yang efektif. Perusahaan-perusahaan besar mayoritas menggunakan email sebagai media komunikasi dengan dengan para pelangganya. Namun tidak semua email yang dikirim dapat sampai ke kotak masuk email para pelanggan. Banyak faktor yang mempengaruhi hal tersebut, diantaranya karena konten yang tidak sesuai kaidah penulisan yang baik, alamat email yang tidak valid, domain pengguna yang terdaftar dalam Blacklist dan sebagainya. Berdasarkan hal tersebut, diperlukan adanya perangkat lunak yang digunakan untuk pengecekan email yang akan dikirm untuk meningkatkan kemungkinan email sampai ke pelanggan. Penelitian ini menghasilkan sebuah perangkat lunak Backend API validator yang dikembangkan menggunakan bahasa pemrograman Java dan Kotlin. Perangkat lunak tersebut menggunakan metode Domain Name System-Based Blackhole List (DNSBL) dan Support Vector Machine (SVM) dalam melakukan validasi email sebelum dikirim.Melalui penelitian ini diharapkan dapat mengurangi prosentase email gagal terkirim akibat email terdeteksi sebagai spam.*

*Keywords: Backend API, Kotlin, Spam filter, E-mail*

Abstract

Email is a communication medium for users and effective Internet service providers. Large companies use email as a medium of communication with their customers. However, not all emails sent can go to customers' inboxes. Many factors influence this because the content does not comply with good update rules, invalid email addresses, user domains that are registered on the Black List and so on. Based on this, we need software that is used to check e-mail that will be sent to increase the value of e-mail to the customer. This research makes a Backend API validator software application developed using Java and Kotlin programming languages. The software uses the Domain Name System-based Blackhole List (DNSBL) and Support Vector Machine (SVM) methods to validate emails before they are sent. Through this research, it is expected to reduce the percentage of emails that are contacted as emails that are generated as spam.

*Keywords: Backend API, Kotlin, Spam filter, E-mail*

1. PENDAHULUAN

MTarget.co adalah perusahaan *startup* IT dengan model bisnis *SaaS (Software as a Service)* dari Jakarta. Kontribusi mereka dalam aktivitas pemasaran di perusahan adalah membangun email marketing automation software berbasis cloud, untuk membantu usaha kecil, menengah hingga perusahaan besar dalam mengembangkan bisnis mereka.[1]

Permasalahan yang sering terjadi baik pada perusahaan ataupun agensi *email marketing* seperti MTarget.co adalah tidak semua email yang dikirim dapat sampai ke kotak masuk email para pelanggan. Banyak email yang dikirim terdeteksi sebagai spam, dan akhirnya email tersebut mengalami *Bounce*. *Bounce Message* adalah email otomatis yang memberi tahu pengirim pesan sebelumnya bahwa pesan itu belum terkirim (atau masalah pengiriman lainnya terjadi).[2]

Semakin besar persentase *Bounce* untuk setiap email dikirim, semakin besar pula dampak yang negatif yang diterima baik bagi perusahaan ataupun pelanggan. Selain menurunkan nilai pemasaran, *IP address* ataupun *DNS address* dari perusahaan atau jasa pengiriman *email marketing* dapat di *banned* oleh server mail dari email pelanggan. Proses *whitelist*/pemurnian email yang diblokir cenderung rumit dan memakan waktu yang relatif lama tergatung dari bagaimana IP atau DNS tersebut terblokir. Hal ini dapat menghambat aktivitas marketing perusahaan yang berakibat menurunya omset perusahaan.

Menurut Hasan Alkahtani[3] taksonomi untuk penyaringan spam pada email secara umum dapat dibagi menjadi 2 yaitu *Reputation-Based Filtering* dan *Content-Based Filtering*. *Reputation-Based Filtering* adalah penyaringan spam yang penyebabnya bukan dari konten pada email. Penyaring tersebut membuat penilaian terhadap reputasi dari pengirim, penerima dan perantara dalam proses pengiriman pesan. Sedangkan *Content-Based Filtering* adalah penyaringan spam dengan menilai apakah konten email mengandung kata-kata atau pola berpontensi menyebabkan spam.

Berdasarkan taksonomi tersebut penulis ingin mengimplementasikan beberapa metode untuk melakukan penyaringan spam pada email. Metode tersebut adalah *DNSBL Filtering* untuk implementasi *Reputation-Based Filtering* dan klasifikasi menggunakan metode *Support Vector Machine* untuk implementasi dari *Content-Based Filtering*.

Diperlukan langkah preventif agar email yang dikirim tidak mengalami *Bounce*. Solusi untuk permasalahan tersebut dapat berupa sistem yang memeriksa setiap email yang akan dikirim. Setiap email diperiksa apakah sudah memenuhi kriteria email yang lolos uji spam. Apabila memenuhi kriteria email tersebut akan di kirim ke pengguna dan sebaliknya. Hal tersebut lebih efektif daripada harus mendata setiap email yang mengalami *Bounce* untuk dikoreksi dan dikirim kembali. Selain itu langkah tersebut dapat meminimalisir *bandwith* yang digunakan beserta beban jaringan. Dari sistem tersebut diharapkan kerugian akibatnya *Bounce* dapat diminimalisir.

1. METODE PENELITIAN

2.1 Tinjauan Pustaka

2.1.1 Email

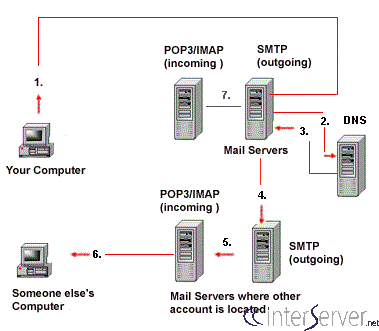
Surat elektronik (akronim: ratel, ratron, surel, atau surat-e) atau pos elektronik (akronim: pos-el.) atau imel (bahasa Inggris: *email*) adalah sarana kirim mengirim surat melalui jalur jaringan komputer (misalnya Internet).[4]

2.1.2 Mail Server

Mail Server adalah sebuah aplikasi komputer dimana aplikasi ini menerima email masuk dari pengguna lokal (orang-orang dalam domain yang sama) serta pengirim jarak jauh dan meneruskan email keluar untuk pengiriman. Komputer yang memasang aplikasi semacam itu juga dapat disebut sebagai Mail Server.Mail Server dibedakan menjadi 2 yaitu Mail Server yang digunakan untuk email keluar disebut sebagai MTA (Mail Transfer Agent) dan Mail Server untuk masuk, menggunakan protokol POP3 / IMAP disebut sebagai MDA (Mail Delivery Agent).

Dalam proses pengiriman email setidaknya menggunakan 3 protokol utama yaitu :

* IMAP adalah singkatan dari Internet Mail Access Protocol. Protokol ini digunakan saat menerima email. Ketika seseorang menggunakan IMAP, email akan ada di server dan tidak diunduh ke kotak surat pengguna dan dihapus dari server. Ini membantu untuk memiliki lebih sedikit memori yang digunakan di komputer lokal dan memori server meningkat.
* POP adalah singkatan dari Post Office Protocol. Protokol ini juga digunakan untuk email yang masuk. Perbedaan utama dengan kedua protokol adalah bahwa POP mengunduh seluruh email ke komputer lokal dan menghapus data di server setelah diunduh. Ini sangat membantu dalam server dengan memori yang lebih sedikit. Versi POP saat ini adalah POP3**.**
* SMTP adalah singkatan dari Simple Mail Transfer Protocol. Email dikirim menggunakan protokol ini. Diagram di bawah di bawah ini menjelaskan jalur yang diambil email dari pengirim ke penerima yang dituju :



Pertama pengirim harus memasukkan alamat email penerima bersama dengan pesan menggunakan aplikasi email. Ini harus dilakukan di komputer lokal. Setelah selesai dan tombol "Send" diklik, email akan menuju ke MTA (Mail Transfer Agent). Komunikasi ini dilakukan melalui protokol SMTP.

Langkah selanjutnya adalah pencarian DNS. Sistem mengirim permintaan untuk mencari tahu MTA yang sesuai dari penerima. Ini akan dilakukan dengan bantuan MX Record. Di zona DNS, untuk domain alamat penerima, akan ada MX Record (Mail Exchange Record). Ini adalah DNS Record yang menentukan server email suatu domain. Jadi, setelah pencarian DNS, respons diberikan ke server surat yang diminta dengan alamat IP dari server surat penerima. Dengan cara ini server mail 'to' diidentifikasi.

Langkah selanjutnya adalah mentransfer pesan di antara server surat. Protokol SMTP digunakan untuk komunikasi ini. Sekarang pesan sudah berada diMail Server penerima (MTA).

Sekarang, pesan ini ditransfer ke Mail Delivery Agentdan kemudian ditransfer ke komputer lokal penerima. Seperti yang telah kita lihat sebelumnya, dua protokol dapat digunakan di sini. Jika kami menggunakan POP3, maka seluruh email akan diunduh ke komputer lokal dan salinan di server akan dihapus. Jika protokol yang digunakan adalah IMAP, maka pesan email disimpan di server mail itu sendiri, tetapi pengguna dapat dengan mudah memanipulasi email di server mail seperti di komputer lokal. Inilah perbedaannya ketika menggunakan kedua protokol dan ini adalah bagaimana email Anda dikirimkan. Jika beberapa kesalahan terjadi untuk mengirim email, email akan tertunda. Ada antrian surat di setiap server surat. Email-email ini akan menunggu dalam antrian email. Server email akan terus mencoba mengirim ulang email. Setelah pengiriman email gagal secara permanen, server email dapat mengirim pesan email bouncing kembali ke alamat email pengirim.

2.1.3 Email Marketing

Email Marketing adalah email yang biasanya dikirim ke sekelompok orang dengan tujuan komersial. Email marketing sendiri biasanya memiliki konten berupa iklan, pengajuan kegiatan bisnis, permintaan penjualan atau donasi, dan dimaksudkan untuk membangun kesetiaan, kepercayaan, atau brand awareness. Email marketing dapat dikirim ke daftar prospek yang dibeli atau database pelanggan saat ini. Email marketing biasanya mengacu pada pengiriman pesan email dengan tujuan meningkatkan hubungan pedagang dengan pelanggan saat ini atau sebelumnya, mendorong loyalitas pelanggan dan mengulang bisnis, memperoleh pelanggan baru atau meyakinkan pelanggan saat ini untuk membeli sesuatu dengan segera, dan berbagi iklan pihak ketiga.[5]

2.1.4 Bounce Message

*Bounce Message* adalah email otomatis yang memberi tahu pengirim pesan sebelumnya bahwa pesan itu belum terkirim (atau masalah pengiriman lainnya terjadi).Pengirim terkadang menerima *bounce message* dari server emailnya sendiri, melaporkan bahwa ia tidak dapat mengirim pesan, atau meskipun telah menerima pesan tersebut, sekarang pesan itu tidak dapat dikirim, juga menerima tanggung jawab untuk mengirimkan DSN jika pengiriman gagal.Karena berbagai alasan, terutama spam dan email virus, pengguna mungkin menerima pesan pentalan yang salah dikirim sebagai tanggapan terhadap pesan yang sebenarnya tidak pernah mereka kirim.[6]

2.1.5 Kotlin

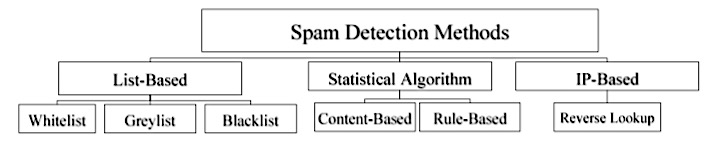
Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin Virtual Java ataupun menggunakan kompiler LLVM yang dapat pula dikompilasikan kedalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim programer dari JetBrains yang bermarkas di Rusia. Meskipun sintaksisnya tidak kompatibel dengan bahasa Java, Kotlin didesain untuk dapat bekerja sama dengan kode bahasa Java dan bergantung kepada kode bahasa Java dari Kelas Pustaka Java yang ada, seperti berbagai framework Java yang ada. Tim Pengembang memutuskan menamakannya Kotlin dengan mengambil nama dari sebuah pulau di Rusia, sebagaimana Java yang mengambil nama dari pulau Jawa di Indonesia. Setelah Google mengumumkan bahwa Kotlin menjadi bahasa kelas satu bagi Android, maka bersama Java dan C++, Kotlin menjadi bahasa resmi untuk pengembangan aplikasi-aplikasi Android.[7]

2.1.6 Sistem Peringan Dini

Menurut Waidyanatha sistem peringatan dini merupakan rangkaian sistem komunikasi informasi yang terdiri dari subsistem sensor, deteksi, keputusan, serta broker yang berurutan, bekerja untuk memprediksi gangguan yang tidak diinginkan yang berpotensi dapat mengganggustabilitas dunia nyata.[8] Diharapkan dengan adanya prediksi tersebut, dapat dilakukan penanggulan secara efektif dan *realtime*.

2.1.7 Sistem Peringan Dini pada Spam Email

Menurut Alireza[9] deteksi spam pada email dapat dilakukan sebelum email dikirim, yaitu dengan memindahkan sistem penyaringan spam ke mail server pengirim. Selain dapat mendeteksi spam dengan waktu yang lebih cepat, juga dapat menghindari penggunaan sumber daya jaringan berlebih. Beberapa metode tersebut dapat diklasifikasikan seperti bagan berikut :



Gambar 2. 1 Metode Deteksi Spam

1. List Based
   1. Whitelist

Dalam teknik ini, setiap pengguna menyimpan kontak emailnya dalam daftar yang disebut *Whitelist*. Oleh karena itu, setiap email yang diterima dengan alamat koresponden dari daftar ini diterima, dan semua alamat lain dari daftar ini dianggap tidak pasti.

* 1. Greylist

Pada langkah pertama, semua email yang diterima ditolak. Karena kebijakan ini, spammer tidak mencoba mengirim ulang email yang ditolak karena memakan waktu lama bagi mereka. Sebagai gantinya, spammer lebih memilih untuk mencari alamat email lain tanpa pemfilteran *Greylist*.

* 1. Blacklist

Dalam pemfilteran *Blacklist*, alamat IP dan nama domain dari server pengirim disimpan dalam daftar yang disebut *Blacklist* dan email dari alamat IP dan domain tersebut diblokir. Kemudian, berdasarkan kebijakan pihak penerima, email dari alamat IP *Blacklisted* dihapus atau dikirim ke folder spam.

1. Statisfical Algorithm
   1. Content-Based

*Content-Based Filtering*adalah teknik penyaringan yang menggunakan *machine learning*. Untuk mendapatkan hasil yang memuaskan, administrator server mail perlu melatih filter untuk menjalankan fungsinya. Pemfilteran ini mulai berfungsi berdasarkan beberapa kata yang telah ditentukan setelah email diterima seluruhnya. Kata-kata khusus ini dikumpulkan oleh laporan statistik berdasarkan kata-kata dan frasa yang dikumpulkan dari email spam.

* 1. Rule-Based

Penyaringan berdasarkan aturanmirip dengan yang berbasis konten dengan beberapa perbedaan. Teknik ini bekerja melalui beberapa aturan dan regulasi tertentu. Dengan aturan ini, filter memutuskan untuk meneruskan atau memblokir email yang diterima.

1. IP-Based
   1. Reverse Lookup

Di *reverse lookup*, juga dikenal sebagai *reverse DNS (Domain Name System) lookup*, *host* dikaitkan dengan alamat *IP (Internet Protocol)* yang diberikan. Dengan menggunakan metode ini, penerima dapat mengkonfirmasi identitas nama domain pengirim.

2.1.8 Client-Server Model

Merupakan sebuah aplikasi yang terdistribusi dimana memiliki server yang bekerja sebagai penyedia layanan atau *resource* dan client yang menggunakan layanan tersebut.[10] Secara umum relasi yang dimiliki server dengan client ialah *one-to-many*, jadi sebuah server bisa diakses banyak client dalam waktu yang bersamaan. Berikut gambaran dari *Client- Server Model*. Biasanya client disebut juga frontend dan server disebut backend dimana kedua buah sistem ini saling berhubungan dan menjadi sebuah sistem yang utuh.

2.1.9 World Wide Web

Perkembangan akses internet sangatlah pesat, hal ini menjadi salah satu bukti bahwa teknologi juga ikut berkembang[11]. Disisi yang sama pertumbuhan sistem informasi juga sangat cepat. Dengan hal ini seluruh sistem informasi lebih gampang untuk diakses. Seluruh sistem informasi yang dapat diakses menggunakan web browser disebut dengan halaman web(*web page*). Dalam bahasa ilmiah halaman web disebut juga *World Wide Web* atau biasa disingkat dengan WWW[12].

2.1.10 HTML

Sebuah teknologi informasi berbasis situs web tidak bisa terlepas dari teknologi bernama HTML. HTML merupakan teknologi dasar untuk membangun sebuah halaman web(*web page*). HTML digunakan untuk mendefinisikan atau mentranslasikan konten dari halam web tersebut, seperti link, paragraf, gambar, heading, dan lain sebagainya.[13]

2.1.11 Web Service

Merupakan salah satu bentuk *Client-Server* model yang termasuk ke dalam Interoperabilitas dengan melakukan komunikasi melalui *World Wide Web (WWW)* dan *HyperText Transfer Protocol (HTTP)*. *Web Service* menyediakan sebuah layanan yang dapat diakses oleh semua platform dan kerangka kerja.[14] *Web serivce* dapat menerima dan menyimpan informasi dalam format seperti HTTP, XML, SSL, SMTP, SOAP, dan JSON.

2.1.12 REST API

Merupakan sekumpulan fungsi yang mana developer dapat melakukan kegiatan *request* dan *response*.[15] Ada enam aturan dimana sebuah sistem dikatakan REST API, berikut aturan aturan tersebut.[16]

2.1.13 DNS dan Tipe DNS

Sistem Penamaan Domain (bahasa Inggris: (Domain Name System; DNS) adalah sebuah sistem yang menyimpan informasi tentang nama host ataupun nama domain dalam bentuk basis data tersebar (distributed database) di dalam jaringan komputer, misalkan: Internet. DNS menyediakan alamat IP untuk setiap nama host dan mendata setiap server transmisi surat (mail exchange server) yang menerima surel (email) untuk setiap domain. Menurut browser Google Chrome, DNS adalah layanan jaringan yang menerjemahkan nama situs web menjadi alamat internet.

2.1.14 Jenis-jenis DNS Record

Beberapa kelompok penting dari data yang disimpan di dalam DNS adalah sebagai berikut:

* A record atau catatan alamat memetakan sebuah nama host ke alamat IP 32-bit (untuk IPv4).
* AAAA record atau catatan alamat IPv6 memetakan sebuah nama host ke alamat IP 128-bit (untuk IPv6).
* CNAME record atau catatan nama kanonik membuat alias untuk nama domain. Domain yang di-alias-kan memiliki seluruh subdomain dan rekod DNS seperti aslinya.
* MX record ataucatatan pertukaran surat memetakan sebuah nama domain ke dalam daftar *mail exchange server* untuk domain tersebut.
* PTR record atau catatan penunjuk memetakan sebuah nama host ke nama kanonik untuk host tersebut. Pembuatan rekod PTR untuk sebuah nama host di dalam domain in-addr.arpa yang mewakili sebuah alamat IP menerapkan pencarian balik DNS (*reverse DNS lookup*) untuk alamat tersebut. Contohnya (saat penulisan / penerjemahan artikel ini), www.icann.net memiliki alamat IP 192.0.34.164, tetapi sebuah rekod PTR memetakan 164.34.0.192.in-addr.arpa ke nama kanoniknya: referrals.icann.org.
* NS record atau catatan server nama memetakan sebuah nama domain ke dalam satu daftar dari server DNS untuk domain tersebut. Pewakilan bergantung kepada rekod NS.
* SOA record atau catatan otoritas awal (*Start of Authority*) mengacu server DNS yang menyediakan otorisasi informasi tentang sebuah domain Internet.
* SRV record adalah catatan lokasi secara umum.
* Catatan TXT mengizinkan administrator untuk memasukan data acak ke dalam catatan DNS; catatan ini juga digunakan di spesifikasi *Sender Policy Framework*.[17]

2.1.15 DNSBL

*Domain Name System-based Blackhole List (DNSBL)* atau *Real-time Blackhole List (RBL)* merupakan daftar hitam dari domain yang terdeteksi mengirim email spam. Sebagian besar perangkat lunak mail server dapat dikonfigurasi untuk menolak atau menandai pesan yang berasal dari domain yang sudah tercatat dalam daftar blokir. Istilah “*Blackhole List*” sendiri berasal dari istilah “*blacklist*” dan “*blocklist*”.[18]

2.1.16 Jenis jenis DNSBL

DNSBL sendiri dapat dikelompokkan berdasarkan fokus *blaclist* dan cara pemeliharaan domain yang terblaclist sebagai berikut:

Tabel 2 Klasifikasi DNSBL yang digunakan CSAIL (Computer Science and Artificial Intelligence Laboratory)

|  |  |  |
| --- | --- | --- |
| Fokus Blacklist | Cara Pemeliharaan | Blacklist |
| Spammer | Konservatif | sbl.spamhaus.org |
| Proxy terbuka | Konservatif | opm.blitzed.org |
| Relay terbuka | Konservatif | rbl.maps.vix.com, list.dsbl.org, multihop.dsbl.org, relays.mail-abuse.org, relays.osirusoft.com,relays.visi.com, relays.orbs.org, relays.ordb.org |
| Relay terbuka | Agresif | unconfirmed.dsbl.org, dnsbl.sorbs.net |
| Serangan Virus/Exploitasi | Agresif | xbl.spamhaus.org, cbl.abuseat.org |
| Netblock ISP/Negara | Agresif | {argentina,att,...}.blackholes.us,dul.maps.vix.com, dul.dnsbl.sorbs.net, dynablock.easynet.nl, blackholes.easynet.nl, dialups.mail-abuse.org |
| RFC  Violators | Mix | {dsn,ipwhois,whois,abuse,postmaster,bogusmx.rfc-ignorant.org |
| Mix | Mix | sbl-xbl.spamhaus.org, bl.spamcop.net, dnsbl.njabl.org |
| Commercial | Commercial | hil.habeas.com, sa-hil.habeas.com, query.bondedsender.org, sa-other.bondedsender.org, sa-trusted.bondedsender.org |
| Unknown | Unknown | rbl.dorkslayers.com, rbl.debian.net |

2.1.16 Cara Kerja Pengecekan Domain berbasis DNSBL

Berikut merupakan alur proses pengecekan domain berbasis DNSBL :

1. DNS lookup tipe A pada domain yang akan di cek untuk mendapatkan ip. Sebagai contoh DNS lookup pada google.com akan mendapatkan ip *172.217.194.113*.
2. Ubah susunan ip dari a.b.c.d menjadi d.c.b.a. Sehingga susunanya menjadi *113.194.217.172*.
3. Konkat dengan DNSBL yang ingin di test. Sebagai contoh DNSBL yang akan dipakai adalah sbl.spamhaus.org. Sehingga susunanya menjadi *113.194.217.172.sbl.spamhaus.org.*
4. DNS lookup tipe TXT pada ip yang sudah di konkat. Apabila hasilnya sukses maka domain tersebut telah terblokir pada DNSBL yang terkait dan sebaliknya.[19]

2.1.17 Web Scraping

*Web scraping, web harvesting* atau *web data extraction* adalah *data scraping* untuk mengekstrak data dari website.[20] Perangkat lunak web scrapper akan mengakses *World Wide Web* melalui protokol HTTP atau melalui web browser. *Web scrapping* dapat dilakukan menggunakan *bot* atau *web crawler*.[21]

2.1.18 Machine Learning

*Machine Learning*, cabang dari kecerdasan buatan, adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritme yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris, seperti dari sensor data basis data. Sistem pembelajar dapat memanfaatkan contoh data untuk menangkap ciri yang diperlukan dari probabilitas yang mendasarinya (yang tidak diketahui). Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara variabel yang diamati. Fokus besar penelitian pemelajaran mesin adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Kesukarannya terjadi karena himpunan semua peri laku yang mungkin, dari semua masukan yang dimungkinkan, terlalu besar untuk diliput oleh himpunan contoh pengamatan (data pelatihan). Karena itu pembelajar harus merampatkan (generalisasi) perilaku dari contoh yang ada untuk menghasilkan keluaran yang berguna dalam kasus-kasus baru.[22]

2.1.19 Text Mining

Untuk dapat mengklasifikasikan suatu teks kita memerlukan sebuah teknik yang dapat mengambil informasi pada data yang berupa teks. Oleh karena itu, kita memerlukan teknik yang disebut text mining. Text mining merupakan sebuah pengembangan teknik data mining yang mana text mining dapat mencari pola tertentu pada kumpulan data tidak terstruktur (text) dalam suatu dokumen. Secara keseluruhan text mining melibatkan data mining, machine learning, information retrieval, dan natural language processing. Hasil dari sebuah proses text mining adalah sebuah pengetahuan berupa pola yang dapat diterapkan untuk menyelesaikan masalah.

2.1.20 TF IDF

Dalam information retrieval, tf-idf atau TFIDF, kependekan dari term-frequency-inverse document frequency, adalah statistik numerik yang dimaksudkan untuk mencerminkan betapa pentingnya sebuah kata untuk sebuah dokumen dalam kumpulan atau kumpulan.[23] Hal ini sering digunakan sebagai faktor pembobotan dalam pencarian pengambilan informasi, penambangan teks dan pemodelan pengguna.

2.1.21 Chi Square

Dalam teori probabilitas dan statistika, distribusi khi-kuadrat (bahasa Inggris: Chi-square distribution) atau distribusi χ² dengan k derajat bebas adalah distribusi jumlah kuadrat k peubah acak normal baku yang saling bebas. Distribusi ini seringkali digunakan dalam statistika inferensial, seperti dalam uji hipotesis, atau dalam penyusunan selang kepercayaan.[24]

2.1.22 Support Vector Machine (SVM)

Support Vector Machine merupakan supervised learning model yang digunakan untuk melakukan analisa data dalam proses klasifikasi atau regeresi linear. Pada proses klasifikasi metode pembelajaran ini akan memetakan data pada suatu bidang dan membedakan kategori yang terpisah secara jelas dengan jarak yang selebar-lebarnya. Kategori yang berbeda tersebut dipisahkan oleh hyperplane.[25]

2.2 Jenis Dan Sumber Data

Mengumpulkan beberapa jenis data sebagai acuan penelitian. Data sekunder didapatkan dari studi literatur dan dokumen penelitian terkait sebelumnya. Studi pustaka dilakukan untuk mencari 2 jenis data yaitu dataset email spam & ham dan daftar DNSBL.

2.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah Studi Pustaka. Dataset email spam & ham diambil dari Enron-Spam Dataset, sedangkan daftar DNSBL diambil dari hasil *web scrapping* dari website https://en.wikipedia.org/wiki/Comparison\_of\_DNS\_blacklists.

2.4 Implementasi

Melakukan implementasi dari hasil analisis untuk membangun backend API web service dengan menerapkan arsitektur Restful.

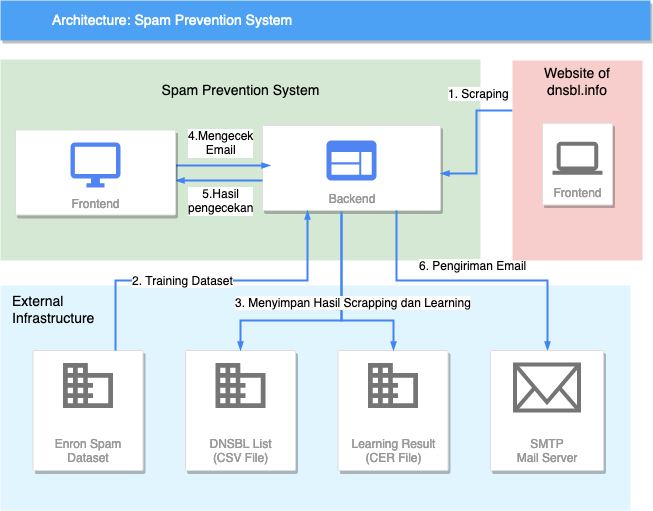
2.5 Eksperimen dan Cara Pengujian Metode

Melakukan eksperimen dan pengujian metode pada web service yang telah dibangun menggunakan arsitektur Restful.

1. HASIL DAN PEMBAHASAN

3.1 Metode yang Diusulkan

Rancanan sistem peringatan dini untuk mendeteksi spam pada email yang lama kurang lebih seperti berikut:



Gambar 1 Desain Web Service

Alur dari Sistem Peringatan Dini Untuk Mendeteksi Spam Pada Email sebagaimana dilihat pada gambar 3.1 diatas dijelaskan sebagai berikut:

1. Proses scrapping DNSBL

Proses ini dilakukan untuk mengambil data DNSBL dengan melakukan scrapping pada laman https://en.wikipedia.org/wiki/Comparison\_of\_DNS\_blacklists, dijelaskan sebagai berikut :

1. Backend melakukan HTTP Request ke laman https://en.wikipedia.org/wiki/Comparison\_of\_DNS\_blacklists mendapatkan laman tersebut dalam format HTML.
2. Backend melakukan ekstraksi HTML tersebut menghasilkan daftar DNSBL.
3. Daftar DNSBL kemudian disimpan kedalam file berekstensi csv.
4. Proses trainingdataset email spam dan ham

Proses ini dilakukan untuk membuat pemodelan menggunakan metode Support Vector Machine, dijelaskan sebagai berikut :

1. Backend membaca setiap file dari folder dataset email, ham & spam.
2. Backend mengubah dataset menjadi fitur data menggunakan DF.
3. Backend melakukan feature selection menggunakan Chi Square.
4. Backend melakukan pembobotan kata menggunakan TF IDF.
5. Backend melakukan pemodelan menggunakan Support Vector Machine.
6. Hasil training / model kemudian disimpan dalam file berekstensi crt.
7. Proses pengecekan email

Proses ini dilakukan untuk menentukan apakah email termasuk spam atau tidak, dijelaskan sebagai berikut :

* + 1. User mengirim email dari aplikasi Frontend Web.
    2. Backend membaca domain dari alamat email pengirim.
    3. Backend melakukan pengecekan menggunakan DNSBL Filter, apabila terdeteksi sebagai spam maka akan ditampilkan pesan ke aplikasi Frontend Web, jika tidak lanjut ke tahap berikutnya.
    4. Backend membaca konten email.
    5. Backend melakukan text mining dengan melakukan stemming, tokenisasi dan stop word removal pada konten email.
    6. Backend melakukan prediksi pada hasil text mining tersebut.
    7. Backend melakukan prediksi menggunakan SVM Filter apabila terdeteksi sebagai spam maka akan ditampilkan pesan ke aplikasi Frontend Web, jika tidak lanjut ke tahap berikutnya

1. Proses pengambilan keputusan terhadap hasil pengecekan email

Setelah proses pengecekan email selesai, backend akan mengeluarkan dokumen yang berisi hasil pengecekan. Dari hasil tersebut user dapat menentukan apakah email tersebut layak untuk di kirim atau direvisi.

3.3 Metode Evaluasi

Evaluasi dilakukan untuk mengetahui apakah aplikasi yang telah dibangun dapat berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan. Dalam penelitian ini penulis melakukan evaluasi dengan menggunakan metode *white-box testing*. Untuk melakukan pengujian *white-box* penulis menulis kode unit testing. White-box testing dilakukan untuk memastikan setiap komponen berjalan sesuai tujuan awal. Pengujian ini dilakukan pada bagian fungsi-fungsi utama. Pengujian dilakukan dengan memasukan inputan ke setiap *endpoint* dan dibandingkan dengan keluaran yang dibutuhkan. Beberapa komponen yang akan diuji adalah fungsi pengecekan domain, fungsi scrapping, penambahan, pengurangan & penambilan dnsbl, training dataset, pengecekan konten spam dan mengirim email.White-box testing dilakukan menggunakan JUnit, *unit testing framework* untuk bahasa pemrograman yang menggunakan *Java Virtual Machine (JVM)* seperti Java dan Kotlin. Setelah semua sudah sesuai maka Backend API yang sudah dibangung siap untuk diluncurkan.

Selain menggunakan *white-box testing* penulis juga akan menggunakan *black-box testing* pada penelitian kali ini. *Black-Box Testing* merupakan bentuk pengecekan sistem berdasarkan spesifikasi kebutuhan dari sistem itu sendiri dan tanpa melakukan pengecekan kode.*Black-Box Testing* murni melakukan pengecekan berdasarkan dari tampilan pengguna.[26] Blackbox digunakan untuk memastikan integrasi antar komponen dapat berjalan dengan baik.

1. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Setelah melakukan penerapan DNSBL dan Support Vector Machine (SVM) pada sistem peringatan dini untuk mendeteksi spam pada email, kesimpulan dari penelitian ini adalah :

* SVM, TF-IDF, Chi Square dan DNSBL filter dapat diimplementasikan pada sistem peringatan dini pendeteksi spam pada email. Sistem peringatan dini ini direalisasikan ke dalam bentuk sebuah pengaya web browser.
* Sistem yang telah dibuat menggunakan SVM dan DNSBL sebagai teknik klasifikasi telah diimplementasikan secara benar dan sesuai dengan hasil yang diharapkan.

4.2 Saran

Penulis sadari bahwa apa yang penulis buat saat ini belumlah sempurna, oleh karena itu penulis memberikan saran kepada pengembang sistem peringatan dini untuk mendeteksi spam pada emailtahap selanjutnya. Berikut merupakan saran yang diberikan penulis:

1. Untuk menambah fitur atau metode lain yang dapat meningkatkan efektivitas filter spam.
2. Mengurangi waktu pemfilteran dari masing masing fitur.
3. Melakukan optimasi sehingga sistem peringatan dinitersebut menjadi lebih sempurna.

DAFTAR PUSTAKA

1. MailTarget. “*MailTarget Company Profile”.* 2018.
2. “*Bounce Message”*. [Online]. Available: <https://en.wikipedia.org/wiki/Bounce_message> [Diakses Juni 3, 2018]
3. Alkahtani, H. “A Taxonomy of Email SPAM Filters”.
4. “Surat Elektronik”. [Online]. Available: <https://id.wikipedia.org/wiki/Surat_elektronik> [Diakses Juni 3, 2018]
5. “Email Marketing”. [Online]. Available: [https://en.wikipedia.org/wiki/Email\_marketing](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiKn7DK0f7jAhW0guYKHWjFBygQFjAAegQIARAB&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FEmail_marketing&usg=AOvVaw2A_jqaMgPubnikwEY9dhEn) [Diakses Juni 3, 2018]
6. “Bounde Message”. [Online]. Available: <https://en.wikipedia.org/wiki/Bounce_message> [Diakses Juni 3, 2018]
7. “Kotlin”. [Online]. Available: <https://id.wikipedia.org/wiki/Kotlin_(bahasa_pemrograman>) [Diakses Juni 3, 2018]
8. Waidyanatha, N. “Towards a typology of integrated functional early warning systems”. *Int. J. Crit. Infrastructures* *, 6* (1), 31-51. 2010.
9. Pour, A. N. “MINIMIZING THE TIME OF SPAM MAIL DETECTION BY RELOCATING FILTERING SYSTEM TO THE SENDER MAIL SERVER”. 2012.
10. “*Client–server model*”. [Online]. Available: <https://techterms.com/definition/client-server_model> [Diakses Maret 10, 2019]
11. “Number Of Internet Users Worldwide”. [Online]. Available: [https://www.statista.com/statistics/273018/number- of-internet-users-worldwide/](https://www.statista.com/statistics/273018/number-%20of-internet-users-worldwide/) [Diakses Maret 11, 2019]
12. “*Apa itu World Wide Web ?”. [Online]. Available:* <http://faculty.petra.ac.id/dwikris/docs/desgrafisweb/www/4-apaitu_www.html> [Diakses Maret 11, 2019]
13. “*HTML”*.[ Online]. Available: <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML> [Diakses]
14. “*What Are Web Services?”.[Online].* Available: <https://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>. [Diakses 11, 2019]
15. Deering, S. (t.thn.). *Do you know what a REST API is?* . Dipetik Maret 11, 2019, dari <https://www.sitepoint.com/developers-rest-api/>
16. Gustavsson, E. S. Efficent data communication between a webclient and a cloud environment,. 2016.
17. “Domain Name System”. [Online]. Available: <https://en.wikipedia.org/wiki/Domain_Name_System>.
18. “*DNSBL”*. Dipetik Juli 21, 2018, dari <https://en.wikipedia.org/wiki/DNSBL>
19. Jung, J. (2004). An Empirical Study of Spam Traffic and the Use of DNS Black Lists.
20. Boeing, G. (2016). New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings. *Journal of Planning Education and Research* .
21. Wikipedia. (t.thn.). *Web Scrapping*. Dipetik Oktober 21, 2018, dari <https://en.wikipedia.org/wiki/Web_scraping>
22. Wikipedia. (t.thn.). *Pemelajaran Mesin*. Dipetik Maret 11, 2019, dari <https://id.wikipedia.org/wiki/Pemelajaran_mesin>
23. Rajaraman, A. (2011). *Mining of Massive Datasets.*
24. Wikipedia. (t.thn.). *Distribusi khi-kuadrat* . Diambil kembali dari Wikipedia: <https://id.wikipedia.org/wiki/Distribusi_khi-kuadrat>
25. Wikipedia. (t.thn.). *Support Vector Machine*. Dipetik Februari 26, 2019, dari <https://en.wikipedia.org/wiki/Support-vector_machine>
26. Nidhra, S. (2012). BLACK BOX AND WHITE BOX TESTING TECHNIQUES –ALITERATURE REVIEW. *International Journal of Embedded Systems and Applications (IJESA)* *, 2* (2), 29-50.