

SISTEM PERINGATAN DINI UNTUK MENDETEKSI SPAM PADA EMAIL MENGGUNAKAN DNSBL FILTER DAN SVM

by Muhammad Hafidz

Submission date: 01-Apr-2019 09:39PM (UTC+0700)

Submission ID: 1103680542

File name: ta1.docx (2.27M)

Word count: 8110

Character count: 56991

TUGAS AKHIR I

SISTEM PERINGATAN DINI UNTUK MENDETEKSI SPAM PADA EMAIL MENGGUNAKAN DNSBL FILTER DAN SVM

Diajukan untuk memenuhi salah satu syarat mengerjakan dan

menempuh ujian tugas akhir 2



Disusun Oleh :

NAMA : MUHAMMAD HAFIDZ

NIM : A11.2014.08602

Program Studi : Teknik Informatika-S1

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

SEMARANG

Tahun 2018

HALAMAN PERSETUJUAN TUGAS AKHIR I

Nama

: Muhammad Hafidz

NIM

: A11.2015.09000

Program Studi

: Teknik Informatika-S1

Fakultas

: Ilmu Komputer

Judul Tugas Akhir

: SISTEM PERINGATAN DINI UNTUK MENDETEKSI
SPAM PADA EMAIL MENGGUNAKAN DNSBL FILTER
DAN SVM

Tugas Akhir ini telah diperiksa dan disetujui,

Semarang, 23 Februari 2019

Menyetujui

Pembimbing

Fahri Firdausillah S.Kom M.CS

HALAMAN PENGESAHAN

Nama

: Muhammad Hafidz

NIM

: A11.2015.09000

Program Studi

: Teknik Informatika-S1

Fakultas

: Ilmu Komputer

Judul Tugas Akhir

: SISTEM PERINGATAN DINI UNTUK MENDETEKSI
SPAM PADA EMAIL MENGGUNAKAN DNSBL FILTER
DAN SVM

Tugas Akhir ini telah diujikan dan di pertahankan didepan Dewan Penguji pada Sidang tugas akhir tanggal 15 Oktober 2019. Menurut pandangan kami, tugas akhir ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugerahan gelar Sarjana Komputer (S.Kom)

Semarang, 23 Februari 2019

Ketua Penguji

Fahri Firdausillah S.Kom M.CS

HALAMAN RINGKASAN

Email merupakan media komunikasi bagi pengguna dan penyedia jasa Internet yang efektif. Perusahaan-perusahaan besar mayoritas menggunakan email sebagai media komunikasi dengan dengan para pelanggannya. 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 dikirim untuk meningkatkan kemungkinan email sampai ke pelanggan. Penelitian ini menghasilkan sebuah perangkat lunak Backend API validator yang dikembangkan menggunakan bahasa pemrograman Java dan Kotlin. Melalui penelitian ini diharapkan dapat mengurangi prosentase email gagal terkirim akibat email terdeteksi sebagai spam.

Kata kunci: Backend API, Java, Kotlin, filter spam, email.

DAFTAR ISI

HALAMAN RINGKASAN	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB II LANDASAN TEORI.....	5
2.1 Tinjauan Studi.....	5
2.2 Tinjauan Pustaka.....	11
2.2.1 Surat Elektronik	11
2.2.2 Mail Server	12
2.2.3 Cara Kerja Email	12
2.2.2.1 IMAP	12
2.2.2.2 POP	12
2.2.2.3 SMTP	13
2.2.4 Email Marketing	14
2.2.5 Bounce Message	15
2.2.6 Java	15
2.2.7 Kotlin	16
2.2.8 Sistem Peringatan Dini	16
2.2.9 Sistem Penanganan Dini pada Spam Email	18
2.2.10 Client-Server Model	20
2.2.11 World Wide Web	21
2.2.12 HTML	21
2.2.13 Hypertext Transfer Protocol.....	22
2.2.14 Web Service	22
2.2.15 REST API.....	22
2.2.16 DNS dan Tipe DNS.....	23
2.2.17 Jenis-jenis DNS Record	24
2.2.18 DNSBL	25

2.2.19	Jenis jenis DNSBL.....	26
2.2.20	Cara Kerja Pengecekan Domain berbasis DNSBL.....	27
2.2.21	Web Scraping	27
2.2.22	Machine Learning.....	28
2.2.23	Text Mining.....	28
2.2.24	Support Vector Machine (SVM)	30
2.2.25	Spam Prevention menggunakan Support Vector Machine (SVM)	33
2.3	Kerangka Pemikiran	34
	BAB III METODE PENELITIAN	35
3.1	Jenis Dan Sumber Data.....	35
3.2	Teknik Pengumpulan Data.....	35
3.3	Metode Pengembangan Sistem	35
3.4.1	Planning / Perencanaan.....	35
3.4.2	Design / Perancangan.....	36
3.4.3	Coding / Pengkodean.....	39
3.4.4	Testing / Pengujian.....	39
3.4	Metode Evaluasi	39
	DAFTAR PUSTAKA.....	41

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka.....	8
Tabel 2.2 Klasifikasi DNSBL yang digunakan CSAIL (Computer Science and Artificial Intelligence Laboratory).....	26

DAFTAR GAMBAR

Gambar 2.1 Jalur pengiriman email.....	13
Gambar 2.2 Alur sistem peringatan dini	17
Gambar 2.3 Metode Deteksi Spam	18
Gambar 2.4 Client-Server Model.....	20
Gambar 2.5 Contoh Syntax HTML	21
Gambar 3.1 Kerangka Sistem Peringatan Dini Deteksi Spam Pada Email	36

BAB I

PENDAHULUAN

1.1 Latar Belakang

MTarget.co adalah perusahaan startup IT dengan model bisnis SaaS (Software as a Service) dari Jakarta. Kontribusi mereka dalam aktivitas pemasaran di perusahaan adalah membangun email marketing automation software berbasis cloud, untuk membantu usaha kecil, menengah hingga perusahaan besar dalam mengembangkan bisnis mereka.[6]

Permasalahan yang sering terjadi baik pada perusahaan ataupun agensi email marketing seperti MTtarget.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). [7]

Bounce dengan penyebab internal seperti konten tidak sesuai kaidah penulisan yang baik, alamat email yang tidak valid, domain pengguna yang terdaftar dalam Blacklist dan sebagainya dikenal dengan *Hard Bounce*. Sedangkan *Soft Bounce* merupakan *Bounce* yang diakibatkan oleh faktor eksternal seperti penuhnya *Inbox*, server penerima email sedang *down* atau luring, isi pesan yang terlalu besar dan sebagainya.

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 mail server dari email pelanggan. Proses *whitelist/pemurnian* email yang diblokir cenderung rumit dan memakan waktu yang relatif lama tergantung dari bagaimana IP atau DNS tersebut terblokir. Hal ini dapat menghambat aktivitas marketing perusahaan yang berakibat menurunnya omset perusahaan.

Menurut Hasan Alkahtani [8] 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 berpotensi 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*.

DNSBL merupakan adalah daftar alamat IP yang dicurigai mengirim spam dan digunakan untuk mencegah pesan email yang tidak diinginkan mencapai penerima yang tidak curiga. Satu hal yang penting untuk disebutkan adalah blacklist itu sebenarnya tidak memblokir pesan pengirim, tetapi justru sebagai tolak ukur. Penyedia menggunakan informasi ini dari berbagai layanan daftar hitam bersama dengan metrik internal untuk membuat keputusan untuk memblokir pesan atau tidak. [9]

Dalam *Machine Learning*, *Support Vector Machine* (SVM) merupakan model pembelajaran supervised dengan algoritma pembelajaran asosiasi yang menganalisis data untuk keperluan analisis klasifikasi ataupun regresi. Diberikan serangkaian contoh pelatihan, masing-masing ditandai sebagai milik satu atau yang lain dari dua kategori, algoritma pelatihan SVM membangun model yang memberikan contoh baru untuk satu kategori atau yang lain, menjadikannya sebagai pengelompokan linear biner non-probabilistik (walaupun metode seperti skala Platt ada untuk menggunakan SVM dalam pengaturan klasifikasi probabilistik). [10]

Berdasarkan hal tersebut, 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 diprinsipalitaskan apakah sudah memenuhi kriteria email yang lolos uji spam. Apabila memenuhi kriteria email tersebut akan dikirim 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. Oleh karena itu, pada penelitian ini penulis akan melakukan implementasi *DNSBL Filtering* dan *Support Vector Machine(SVM)* pada pembuatan sistem menggunakan Odoo, dimana dapat disampaikan jika penulis akan membuat tugas akhir yang berjudul “SISTEM PERINGATAN DINI UNTUK MENDETEKSI SPAM PADA EMAIL MENGGUNAKAN DNSBL FILTER DAN SVM”

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan diatas diketahui bahwa terdapat beberapa teknik untuk mengklasifikasi spam pada email. Namun, teknik klasifikasi tersebut belum diterapkan ke dalam sistem yang sebenarnya. Oleh karena itu didapatkan rumusan masalah sebagai berikut

1. Bagaimana langkah preventif untuk mendekripsi spam pada email secara dini ?
2. Bagaimana membangun Backend API yang menerapkan Support Vector Machine (SVM) dan DNSBL Filter ?

1.3 Batasan Masalah

Pada penelitian ini, terdapat beberapa batasan terhadap masalah yang akan diselesaikan supaya penilitian dapat dilakukan secara lebih terarah dan lebih dapat dipertanggung jawabkan. Batasan-batasan yang dimaksud antara lain :

1. Menggunakan DNSBL Filter dan Support Vector Machine sebagai metode penyaringan spam.
2. Menggunakan DNSBL dari dnsbl.info
3. Dataset berupa kumpulan email dalam bahasa inggris.
4. Menggunakan Kotlin sebagai bahasa utama dalam pembuatan sistem.
5. Menggunakan Restful API sebagai metode Web Service dengan hasil berupa JSON.
6. Aplikasi yang dikembangkan dalam penelitian ini berbasis Backend API.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah :

1. Mencari langkah preventif untuk mendeteksi spam pada email secara dini.
2. Membangun Backend API yang menerapkan Support Vector Machine (SVM) dan DNSBL Filter.

1.5 Manfaat Penelitian

Setelah melaksanakan penelitian, diharapkan penulis dapat memberi manfaat antara lain :

1. Mencegah email bukan spam gagal terkirim ke inbox penerima.
2. Mengurangi potensi email mengalami *Bounce* atau terdeteksi sebagai spam.
3. Meminimalisir bandwith yang digunakan beserta beban jaringan akibat *Bounce*.
4. Mencegah keputusan *email service provider* untuk melakukan banned IP atau DNS pengirim dan hal-hal merugikan lainnya.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Penelitian dalam bidang penyaringan email spam sudah banyak dikembangkan sehingga menghasilkan beberapa produk yang baru dan bermanfaat bagi pengguna. State of the art atau penelitian sebelumnya, akan membahas mengenai persamaan dan perbedaan yang ada dalam penelitian sebelumnya. Dimana persamaan dan perbedaan akan dilihat dari objek yang diteliti, serta metode penelitiannya. Penyusunan data state of the art akan disusun dalam bentuk data matrik, sehingga dapat memperlihatkan perbandingan antara penelitian sebelumnya dengan penelitian yang akan dilakukan. Berikut merupakan referensi yang digunakan :

1. Penelitian oleh Hasan [8]. Dalam penelitiannya yaitu *A Taxonomy of Email SPAM Filters*, penelitian tersebut membahas mengenai taksonomi untuk filter email spam.
2. Penelitian oleh Kamini [15]. Dalam penelitiannya yaitu *A Multi-layer Model to Detect Spam Email at Client Side*, penelitian tersebut membahas mengenai penerapan filter ganda yaitu filter menggunakan SpamBayes dan filter terhadap ukuran dan struktur dokumen seperti tanggal dan waktu email, bidang subjek, hyperlink, digit angka, jumlah kata, penggunaan karakter khusus pada konten email untuk filter email spam. Filter ganda tersebut dapat meningkatkan performa hingga 60% jika dibandingkan hanya dengan menggunakan SpamBayes saja.

3. Penelitian oleh Karthika [14]. Dalam penelitiannya yaitu *Latent Semantic Indexing Based SVM Model for Email Spam Classification*, peneliti melakukan perbandingan beberapa metode Machine Learning dalam melakukan filter terhadap email spam. Metode-metode tersebut diantaranya adalah SVM + TF-IDF, SVM + LSI, Neural Network, PSO dan Firefly + Bayes. Hasil dari perbandingan tersebut menunjukan bahwa SVM + LSI memiliki presisi dan penarikan paling tinggi dan akurasi sedikit dibawah PSO.
4. Keempat oleh Tu Ouyang [11]. Dalam penelitiannya yaitu *A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise*, peneliti membangun penyaringan spam sebagai *pipeline* dengan menerapkan penyaringan berbasis DNSBL, penyaringan berbasis fitur paket SYN, penyaringan berbasis karakteristik lalu lintas dan berdasarkan pesan konten. Mereka menemukan bahwa *pipeline* tersebut bekerja sebaik pengaturan operasional. Selain itu setiap lapisan pipeline juga dapat bekerja dalam waktu yang lama dan dalam beberapa kasus tertentu dalam beberapa kasus, lapisan selanjutnya dapat mengimbangi kinerja yang buruk di lapisan sebelumnya.
5. Kelima oleh Simranjit Kaur Tuteja [12]. Dalam penelitiannya yaitu *A Survey on Classification Algorithms for Email Spam Filtering* melakukan review terhadap beberapa metode *Machine Learning* yang dapat digunakan untuk melakukan filter terhadap email spam. Metode tersebut antara lain : Neural Network (NN), Support Vector Machine (SVM) Classifier, Naïve Bayesian (NB) Classifier, J48 Classifier. Percobaan dilakukan berdasarkan ukuran data yang berbeda dan ukuran fitur yang berbeda. Hasil klasifikasi akhir harus ‘1’ jika akhirnya adalah spam, jika tidak, ‘0’. Klasifikasi Naive Bayesian menunjukkan hasil yang baik, tetapi Neural Network dan SVM tidak menunjukkan hasil yang baik dibandingkan dengan J48 atau classifier Naïve Bayesian. Neural Network dan SVM tidak sesuai untuk dataset untuk membuat keputusan biner. Dari percobaan ini, mereka dapat menemukan bahwa classifier J48 sederhana dapat memberikan hasil klasifikasi yang lebih baik untuk pemfilteran email spam.

6. Keenam oleh D.S.Silnov [13]. Dalam penelitiannya yaitu *An Analysis of Modern Approaches to the Delivery of Unwanted Emails (Spam)* mengulas cara kerja beberapa tools untuk menangkap spam di tingkat jaringan. Teknologi tersebut termasuk DNSBL (memeriksa untuk melihat apakah alamat IP dimasukkan dalam daftar hitam), memeriksa catatan PTR dari alamat IP dan domain email, SPF dan DKIM baru-baru ini dan beberapa lainnya.

Tabel 2.1 Tinjauan Pustaka

No.	Nama	Tahun	Judul	Metode	Hasil
1.	Hasan Al-kahtani	2019	A Taxonomy of Email SPAM Filters	<ul style="list-style-type: none">• Black Lists• White Lists• Challenge-Response Systems (CRS)• Origin Diversity Analysis• Implicit Techniques• Explicit Techniques• Rule Based Filters• Statistical Filters• Genetic Algorithms• Artificial Immune System• Artificial Neural Networks• Clustering Techniques• Decision Tree Technique• Honey Pots• Zombie-Based Approach	Taksonomi mengenai penyaringan spam pada email yang berisi berbagai macam teknik yang telah digunakan atau diusulkan untuk melawan SPAM, dan filter SPAM mana yang harus digunakan

2.	Kamini (Simi) Bajaj	2017	A Multi-layer Model to Detect Spam Email at Client Side	Model multi-layer yang menggunakan SpamBayes dan penyaringan non-teksual yang mengeksplorasi teknik pembelajaran mesin alternatif.	Model multi-layer ini meningkatkan akurasi klasifikasi dan menghilangkan email abu-abu menjadi spam dan email ham.
3	Karthika Renuka	2014	Latent Semantic Indexing Based SVM Model for Email Spam Classification	SVM + LSI	Hasil dari perbandingan tersebut menunjukan bahwa SVM + LSI memiliki presisi dan penarikan paling tinggi dan akurasi sedikit dibawah PSO.
4.	Tu Ouyang	2014	A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise	Penyaringan spam sebagai <i>pipeline</i> dengan menerapkan penyaringan berbasis DNSBL, penyaringan berbasis fitur paket SYN, penyaringan berbasis karakteristik lalu lintas dan berdasarkan pesan konten.	Mereka menemukan bahwa <i>pipeline</i> tersebut bekerja sebaik pengaturan operasional. Selain itu setiap lapisan pipeline juga dapat bekerja dalam waktu yang lama dan dalam beberapa kasus tertentu dalam beberapa kasus, lapisan selanjutnya dapat mengimbangi kinerja yang buruk di lapisan sebelumnya.
5.	D. S. Silnov	2016	An Analysis of Modern Approaches to the Delivery of Unwanted Emails (Spam)	DNSBL Memeriksa catatan PTR dari alamat IP dan domain email SPF DKIM	Metode baru untuk menyingkirkan spam pada email di tingkat jaringan

6.	Simranjit Kaur Tuteja	2016	A Survey on Classification Algorithms for Email Spam Filtering	Neural Network (NN), Support Vector Machine (SVM) Classifier, Naive Bayesian (NB) Classifier, J48 Classifier.	Classifier J48 sederhana dapat 1 memberikan hasil klasifikasi yang lebih baik untuk pemfilteran email spam.
----	-----------------------	------	--	---	---

2.2 Tinjauan Pustaka

2.2.1 Surat Elektronik

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).

Struktur alamat surel, sebagai contoh:

surelsaya@surabaya.vibriel.net.id

Keterangan:

- surelsaya: nama kotak surat (*mailbox*) atau nama pengguna (*username*) yang ingin dituju dalam *mailserver*
- surabaya.vibriel.net.id: nama *mailserver* tempat pengguna yang dituju, rinciannya:
 - surabaya: *subdomain* (milik pemegang nama *domain*), biasanya merujuk ke suatu komputer dalam lingkungan pemilik *domain*
 - vibriel: nama *domain*, biasanya menunjukkan nama perusahaan/organisasi/perorangan (Vibriel)
 - net: *second level domain*, menunjukkan bahwa *domain* ini termasuk kategori *networking* (net)
 - id: *top level domain*, menunjukkan bahwa *domain* ini terdaftar di otoritas *domain* Indonesia (id) [16]

Email merupakan media komunikasi bagi pengguna dan penyedia jasa Internet yang efektif. Menurut The Radiacti Group, Inc, sebuah perusahaan riset market teknologi, pengguna aktif email pada tahun 2015 telah mencapai 2.6 Milyar. [1] Jumlah ini lebih banyak daripada Sosial Media seperti Facebook yang berjumlah 1.7 Milyar[2] dan Twitter yang berjumlah 313 Juta. [3]

2.2.2 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).

2.2.3 Cara Kerja Email

Dalam proses pengiriman email setidaknya menggunakan 3 protokol utama yaitu :

2.2.2.1. IMAP

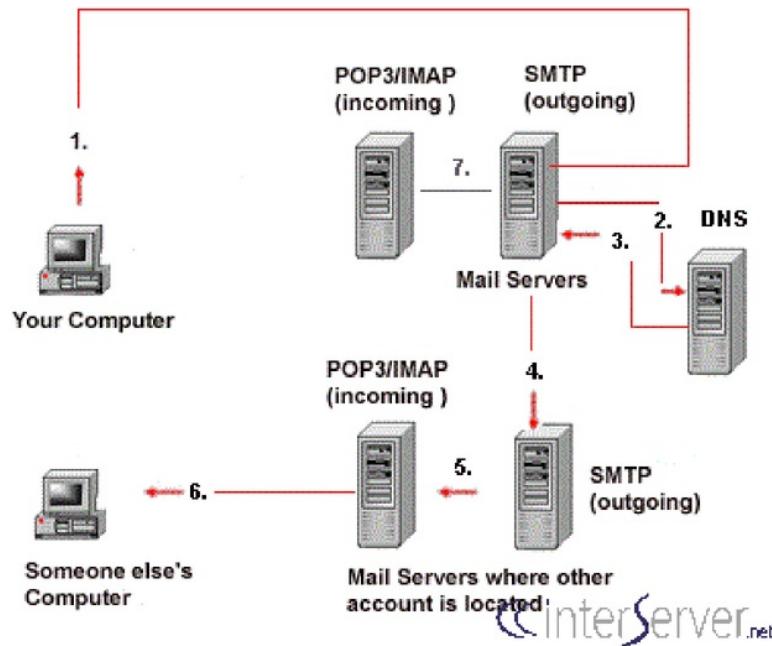
IMAP adalah singkatan dari Internet Mail Access Protocol. Protokol ini digunakan saat menerima email. Ketika seseorang menggunakan IMAP, email akan ada di server dan tidak diunduh ke kotak surat pengguna dan dihapus dari server. Ini membantu untuk memiliki lebih sedikit memori yang digunakan di komputer lokal dan memori server meningkat.

2.2.2.2. POP

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.

2.2.2.3. SMTP

SMTP adalah singkatan dari Simple Mail Transfer Protocol. Email dikirim menggunakan protokol ini. Diagram di bawah ini menjelaskan jalur yang diambil email dari pengirim ke penerima yang dituju :



Gambar 2.1 Jalur pengiriman email

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 di Mail Server penerima (MTA). Sekarang, pesan ini ditransfer ke Mail Delivery Agent dan 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.

2.2.4 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. [4] Dengan email marketing, perusahaan dapat mengirim informasi mengenai profil dan produk-produk mereka dengan cepat dan murah. Email marketing juga merupakan salah satu media marketing yang memiliki tingkat investasi yang tinggi. Menurut The Direct Marketing Association (DMA) pada tahun 2018 setiap rupiah yang diinvestasikan di email marketing, Return of Investment-nya adalah 3228%. [5]

2.2.5 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. [7]

2.2.6 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya

dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi. [17]

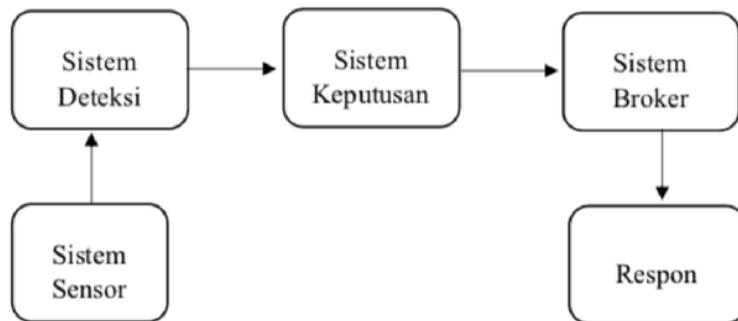
2.2.7 Kotlin

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. [17] 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. [18] 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. [19]

2.2.8 Sistem Peringatan 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 mengganggu stabilitas dunia nyata. [20] Diharapkan dengan adanya prediksi tersebut, dapat dilakukan penanggulangan secara efektif dan realtime.



Gambar 2.2 Alur sistem peringatan dini

Sistem Sensor merupakan bagian dimana sistem menerima informasi masukan. Sumber dan bentuk masukan sangat tergantung dengan domain dari sistem peringatan dini itu sendiri. Sebagai contoh, saat ini banyak pemanfaatan media sosial sebagai sumber masukan untuk penanggulangan bencana alam, militer, krisis ekonomi dan lain sebagainya.

Sistem Deteksi merupakan bagian dimana sistem mengekstraksi data dari kumpulan data yang didapat oleh sensor. Sebagai contoh, sistem peringatan gempa menggunakan akan membaca data dari seismograf dan mendeteksi apakah dari data seismograf tersebut menunjukkan terjadi gempa serta sistem dapat membedakan gempa sesungguhnya dengan gempa yang disebabkan oleh ledakan.

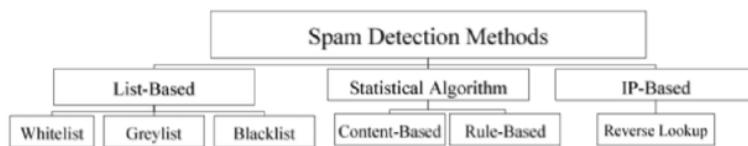
Sistem Keputusan merupakan bagian dimana sistem akan menentukan apakah sistem akan memberikan respon terhadap suatu gangguan atau tidak.

Sistem Broker merupakan sebuah sistem yang memiliki tugas untuk mengirimkan hasil keputusan ke bagian sistem respon. Sistem ini akan merubah data hasil keputusan ke dalam bentuk standard yang dapat di terima oleh sistem respon dan menentukan pesan mana yang akan dikirm.

Sistem respon merupakan keluaran dari sistem peringatan dini yang biasanya berbentuk pesan peringatan. Dalam pesan peringatan tersebut selain memberitahukan tentang resiko yang akan terjadi, diberikan juga instruksi berikutnya guna menaggulangi dampak gangguan.

2.2.9 Sistem Penanganan Dini pada Spam Email

Menurut Alireza [22] 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 2.3 Metode Deteksi Spam

1. List Based

a. Whitelist

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

b. 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.

c. Blacklist

Dalam pemfilteran Blacklist, alamat IP dan nama domain dari server pengirim disimpan dalam daftar yang disebut Daftar Hitam 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.

2. Statisfical Algorithm

a. 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.

b. Rule-Based

Penyaringan berdasarkan aturan mirip 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.

3. IP-Based

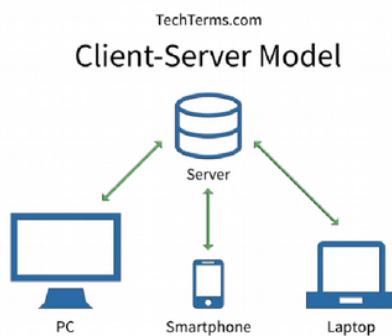
a. Reverse Lookup

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.

2.2.10 Client-Server Model

Merupakan sebuah aplikasi yang terdistribusi dimana memiliki server yang bekerja sebagai penyedia layanan atau resource dan Client yang menggunakan layanan tersebut [21]. 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.



Gambar 2.4 Client-Server Model

- a. Client-side: merupakan sebuah aplikasi sisi klien yang dijalankan dengan sebuah *device* yang menerima inputan dari pengguna. Aplikasi sisi klien ini juga menyiapkan data atau informasi yang dibutuhkan pengguna, setelah pengguna memasukan informasi, data akan diririm ke server atau yang biasanya disebut *request*.
- b. Server-side: merupakan sebuah aplikasi sisi server yang mana berfungsi sebagai menerima *request* dari aplikasi sisi klien yang langsung memproses *request* tersebut dan mengirimkan tanggapan sesuai dengan permintaan aplikasi sis klien atau biasa disebut *response*.

2.2.11 World Wide Web

Perkembangan akses internet sangatlah pesat, hal ini menjadi salah satu bukti bahwa teknologi juga ikut berkembang [22]. 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 [23].

2.2.12 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 [24]. Berikut merupakan contoh syntax dari HTML.

```
1  <!DOCTYPE html>
2  <html>
3  |  <head>
4  |  |  <title></title>
5  |  </head>
6  |  <body>
7
8  |  </body>
9  </html>
10
```

Gambar 2.5 Contoh Syntax HTML

2.2.13 Hypertext Transfer Protocol

Merupakan sebuah protokol *application layer* untuk mengirim atau menerima sebuah dokumen seperti HTML dan lain lain. HTTP digunakan untuk menyambungkan antara web browser dan web server. HTTP juga digunakan sebagai penghubung antara *client-server model*, dimana *client* meminta tanggapan(*response*) dengan menggunakan permintaan(*request*) [25].

2.2.14 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 [26]. Web service dapat menerima dan menyimpan informasi dalam format seperti HTTP, XML, SSL, SMTP, SOAP, dan JSON.

2.2.15 REST API

Merupakan sekumpulan fungsi yang mana developer dapat melakukan kegiatan request dan response [27]. Ada enam aturan dimana sebuah sistem dikatakan REST API, berikut aturan aturan tersebut [28].

- a. Client-Server : Secara arsitektur REST memisahkan pemrosesan sistem menjadi dua komponen. Server merupakan komponen yang menyediakan layanan dan menanggapi permintaan untuk service tersebut. Client merupakan komponen yang terhubung ke server untuk melakukan permintaan ke server.
- b. Stateless : Server tidak melihat status sesi dari Client. Setiap Request yang dikirim melalui Client harus berisi seluruh informasi

- yang dibutuhkan agar server dapat mengerti apa yang harus dikirim ke Client.
- c. Cacheable : Response yang dikirim oleh server harus cacheable. Hal ini bertujuan untuk menghindari request yang tidak diperlukan.
 - d. Uniform Interface : Dengan perbedaan komponen dari sistem REST untuk melakukan komunikasi dari kedua komponen memerlukan standar yang sama(Uniform Interface). Hal ini juga mengurangi efisiensi dalam mengirim informasi, karena informasi yang merupakan bentuk standar sedangkan dari pihak aplikasi client memiliki kebutuhan yang berbeda.
 - e. Layered System : Sistem ini berada di layer yang berbeda. Satu layer hanya bisa berinteraksi dengan layer terdekatnya. Tetapi dari komponen komponen sistem tidak perlu mengerti satu sama lain, asalkan keduanya bekerja dengan baik maka komunikasi data juga akan bekerja.

2.2.16 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.

DNS menyediakan pelayanan yang cukup penting untuk Internet, ketika perangkat keras komputer dan jaringan bekerja dengan alamat IP untuk mengerjakan tugas seperti pengalaman dan penyaluran (*routing*), manusia pada umumnya lebih memilih untuk menggunakan nama host dan nama domain, contohnya adalah

penunjukan sumber universal (URL) dan alamat surel. Analogi yang umum digunakan untuk menjelaskan fungsinya adalah DNS bisa dianggap seperti buku telepon internet di mana saat pengguna mengetikkan www.indosat.net.id di peramban web maka pengguna akan diarahkan ke alamat IP 124.81.92.144 (IPv4) dan 2001:e00:d:10:3:140::83 (IPv6).

2.2.17 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' atau catatan 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*.

Jenis catatan lainnya semata-mata untuk penyediaan informasi (contohnya, catatan LOC memberikan letak *lokasi fisik* dari sebuah host, atau data ujicoba (misalkan, catatan WKS memberikan sebuah daftar dari server yang memberikan servis yang dikenal (*well-known service*) seperti HTTP atau POP3 untuk sebuah domain [29].

2.2.18 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”.

Terdapat lusinan DNSBL yang ada, dimana menggunakan array besar yang berisi kriteria dari alamat yang terdaftar dan tidak. DNSBL dapat berisi alamat komputer zombie atau mesin yang dikhurasukan untuk mengirim spam, Penyedia Layanan Internet/Internet Service Provider (ISP) yang bersedia menjadi host spammer atau yang mengirim spam ke sistem honeypot [30].

2.2.19 Jenis jenis DNSBL

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

Tabel 2.2 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.dsbl.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

2.2.20 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. [31]

2.2.21 Web Scraping

Web scraping, web harvesting atau web data extraction adalah data scraping untuk mengekstrak data dari website [32]. 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 [33].

2.2.22 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 perilaku 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 [34].

2.2.23 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.

Text mining memiliki alur dan proses yang hampir serupa dengan data mining pada umumnya. Yang pertama adalah data preparation, pada

tahapan ini data yang digunakan akan dipilah terlebih dahulu. Selain itu pada tahapan ini dilakukan juga proses preprocessing dokumen (kategorisasi teks, ekstraksi informasi, ekstraksi istilah). terdapat beberapa hal yang dilakukan dalam preprocessing dokumen sebagai berikut, diantaranya :

1. Tokenisasi

Tokenisasi merupakan proses memecah kalimat menjadi kumpulan token atau kata. proses ini akan menghilangkan beberapa karakter seperti “ ”(spasi), “,”, “.” dan tanda baca lainnya.

2. Stop word removal

Proses ini akan menghilangkan beberapa kata yang tidak berpengaruh seperti “di”, “ke”, “yang”, dan lain sebagainya.

3. Stemming

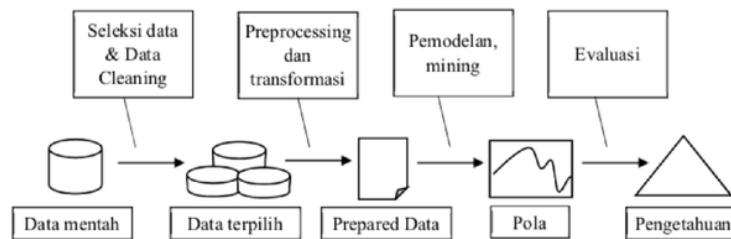
Proses ini akan merubah suatu token ke dalam bentuk dasarnya. Contohnya “menyapu” menjadi “sapu”. Bila tidak ditemukan rule yang mengatur bentuk dasar dari sebuah token maka token tersebut tidak akan berubah.

4. Term weighting

Term weighting atau pembobotan adalah sebuah proses untuk menentukan bobot dari setiap kata. pada proses ini data yang sebelumnya berupa data yang tidak terstruktur sudah berubah menjadi data intermediate yang lebih terstruktur.

Tahap berikutnya adalah pemodelan, pada tahap ini data yang telah di preprocessing telah menjadi data intermediate yang lebih terstruktur. lalu data intermediate tadi dianalisa menggunakan teknik analisa representasi intemediate (seperti analisis distribusi, clustering, analisis tren, klasifikasi dan association rules).

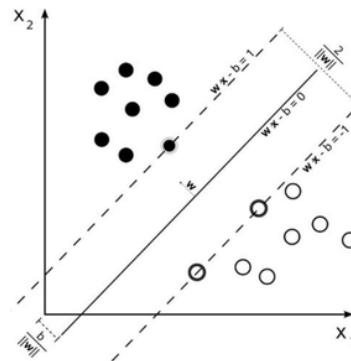
Tahap yang terakhir adalah evaluasi, pada tahap ini dilakukan visualisasi dari hasil pemodelan.



Gambar 2.6 Proses pencarian pengetahuan untuk text mining

2.2.24 Support Vector Machine (SVM)

Support Vector Machine merupakan supervised learning model yang digunakan untuk melakukan analisa data dalam proses klasifikasi atau regresi 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. Berikut adalah gambaran klasifikasi menggunakan svm.



Gambar 2.7 Gambaran klasifikasi menggunakan support vector machine

Hyperplane merupakan sebuah garis atau bidang yang memisahkan 2 buah kategori yang berbeda. Sebuah hyperplane dapat kita cari dengan menggunakan rumus sebagai berikut :

$$w \cdot x_i - b = 0 \quad \dots(1)$$

w pada persamaan diatas merupakan vektor yang menentukan bobot. Sementara itu x_i merupakan vektor sampel data dan b merupakan bias. Pada klasifikasi biner, hasil klasifikasi akan menghasilkan true (+1) dan false (-1). Untuk klasifikasi tersebut berlaku rumus sebagai berikut:

$$f(x) = \begin{cases} +1, & \text{jika } w \cdot x_i - b \geq 0 \\ -1, & \text{jika } w \cdot x_i - b \leq 0 \end{cases} \quad \dots(2)$$

Namun, agar hasil klasifikasi menjadi lebih baik kita perlu memaksimalkan nilai margin. Margin sendiri merupakan jarak antara titik-titik positif dan negatif terdekat di sekitar hyperplane. titik-titik positif dan negatif terdekat di sekitar hyperplane ini biasa disebut dengan support vector. Untuk menentukan margin dapat digunakan persamaan berikut :

$$f(x) = \begin{cases} +1, & \text{jika } w \cdot x_i - b \geq 1 \\ -1, & \text{jika } w \cdot x_i - b \leq -1 \end{cases} \quad \dots(3)$$

Pada persamaan di atas batas atas dan batas bawah tidak lagi 0 seperti persamaan. Hal ini dikarenakan $-1 < w \cdot x_i - b < 1$ menyatakan margin. Persamaan dapat disederhanakan menjadi :

$$y_i(w \cdot x_i - b) \geq 1 \quad \dots(4)$$

Pada umumnya dataset yang digunakan dalam klasifikasi teks dapat dipisahkan secara linear. Namun, tidak menutup kemungkinan bahwa dataset yang kita miliki tidak dapat dipisahkan secara linear. Oleh karena itu kita perlu membuat model yang dapat menghiraukan beberapa data tidak sesuai guna menambah akurasi. Untuk melakukan nya kita dapat menambahkan sebuah variabel slack (ξ). Teknik ini disebut soft margin.

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i \quad \xi_i > 0 \quad \dots(5)$$

Dari persamaan di atas, agar nilai margin menjadi maksimum kita harus meminimalkan problem sebagai berikut :

$$\min \quad \frac{1}{2} w \cdot w + C \sum_{i=1}^n \xi_i$$

$$s.t. \quad y_i(w \cdot x_i - b) \geq 1 - \xi_i \quad \xi_i > 0 \quad \dots (6)$$

Pada persamaan diatas C merupakan parameter yang menentukan seberapa banyak data yang dibiarkan misclassify. Kita dapat mempermudah penyelesaian persamaan di atas dengan mengubahnya kedalam bentuk lagrange sebagai berikut :

$$L(w, \xi, b, \alpha) = \frac{1}{2} w \cdot w + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i [1 - \xi_i - y_i(w \cdot x_i - b)] \quad \dots (7)$$

α_i pada persamaan di atas merupakan lagrange multiplier yang memiliki nilai 0 atau positif. Dari penyelesaian lagrange di atas kita bisa mendapatkan w sebagai berikut :

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad \dots (8)$$

jika $0 < \alpha_i < C$ maka x_i merupakan support vector. kemudian Nilai w dan b yang didapat digunakan pada persamaan (2) untuk proses klasifikasi. Jika nilai w pada persamaan (8) kita substitusikan dengan w yang ada pada persamaan (2) maka didapatkan fungsi objektif sebagai berikut :

$$f(x) = \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle - b \quad \dots (9)$$

Persamaan diatas merupakan fungsi objektif untuk menyelesaikan masalah secara linear. Untuk masalah yang tidak dapat di selesaikan secara linear (non-linear), kita dapat mengganti perhitungan dot product x_i, x dengan sebuah fungsi kernel lain seperti kernel radial base function atau sigmoid. Fungsi kernel ini dapat digunakan untuk membantu memecahkan masalah non-linear.

2.2.25 Spam Prevention menggunakan Support Vector Machine (SVM)

Karthika mengusulkan teknik klasifikasi spam email menggunakan Latent Semantic Indexing Based SVM Model. Pada awalnya, dataset input diberikan pada langkah pra-pemrosesan yang menghilangkan *stop word* dan tanda baca sehingga kata kunci yang lebih relevan diperoleh. Kata kunci yang diekstraksi kemudian diberikan kepada ekstraksi fitur di mana, Term Frequency (TF) dan Invers Document Frequency (IDF) memproses kata kunci yang diambil dari langkah-langkah pra-pemrosesan. Setelah pembentukan matriks fitur, dimensi yang sesuai untuk klasifikasi yang lebih baik ditemukan menggunakan model LSI yang memetakan ruang fitur ke ruang LSI menggunakan analisis berbasis korelasi. Akhirnya, ruang LSI diberikan ke algoritma SVM yang melatih berdasarkan pola yang diberikan dalam ruang pelatihan LSI. Dalam fase pengujian, email masukan yang diwakili dalam ruang LSI diklasifikasikan sebagai Spam atau Ham berdasarkan hyperplane optimal yang dihasilkan dalam pelatihan SVM. Dengan menggunakan langkah-langkah ini, klasifikasi spam dan email ham telah dilakukan secara efektif.

2.3 Kerangka Pemikiran

Berdasarkan teori yang telah dipaparkan sebelumnya, penelitian akan dilakukan dengan kerangka pemikiran sebagai berikut :

Tabel 2.3Kerangka Pemikiran

Problem
<ul style="list-style-type: none">• Belum adanya sistem deteksi dini email spam sebagai bentuk pencegahan pada MTTarget
Approach
<ul style="list-style-type: none">• Menerapkan metode DNSBL Filtering dan klasifikasi SVM untuk membuat sistem deteksi dini email spam sebagai bentuk pencegahan
Development
<ul style="list-style-type: none">• Server Side : Kotlin dengan menggunakan framework vertx
Evaluation and Validation
<ul style="list-style-type: none">• Mengecek seluruh fitur yang ada menggunakan WEB UI
Result
<ul style="list-style-type: none">• Sistem Peringatan Dini Untuk Mendeteksi Spam Pada Email Menggunakan Dnsbl Filter Dan Svm

BAB III

METODE PENELITIAN

Metode penelitian yang digunakan dalam penulisan Tugas Akhir ini, sebagai berikut :

3.1 Jenis Dan Sumber Data

Penulis telah mengumpulkan beberapa jenis data sebagai acuan penelitian. Data tersebut termasuk dalam data sekunder, yaitu data yang dijadikan landasan teori dan penunjang yang diperoleh peneliti dari sumber yang sudah ada. 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.

3.2 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 dnsbl.info.

3.3 Metode Pengembangan Sistem

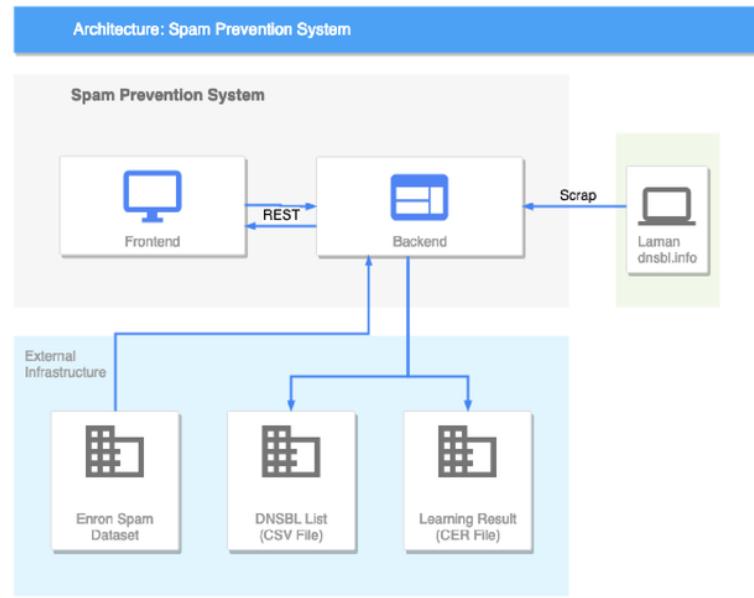
3.4.1 Planning / Perencanaan

Perencana atau *planning* merupakan tahapan awal dimana peneliti mengumpulkan seluruh kebutuhan, fitur utama, keluaran dari sistem (*output*), dan fungsionalitasnya. Berikut adalah kebutuhannya:

1. Sistem yang dikembangkan merupakan Backend API atau aplikasi sisi server(*server-side*).
2. Fitur utama dari Backend API ini merupakan untuk pengecekan domain dari tiap DNSBL dan pengecekan konten spam.
3. Backend API yang penulis kembangkan menggunakan teknologi web service RESTful.
4. Data yang dikeluarkan dari Backend API berformat JSON.

3.4.2 Design / Perancangan

Perancangan dalam pemrograman ekstrim ini memiliki prinsip yaitu penyederhanaan atau *simplicity*. Perancangan yang sederhana selalu memakan waktu yang singkat dibanding perancangan yang komplek. Jika perancangan sederhana masih menemui kesulitan maka bisa menggunakan solusi *spike* dimana pengembang bisa langsung menerapkan atau mengimplementasikan prototipe perancangan dan dilanjutkan evaluasi. Rancangan sistem peringatan dini untuk mendeteksi spam pada email yang lama kurang lebih seperti berikut:



Gambar 3.1 Kerangka Sistem Peringatan Dini Deteksi Spam Pada Email

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

a. Proses scrapping

Proses ini dilakukan untuk mengambil data DNSBL dengan melakukan scrapping pada laman dnsbl.info, dijelaskan sebagai berikut :

1. Backend melakukan HTTP Request ke laman dnsbl.info mendapatkan laman tersebut dalam format HTML.
2. Backend melakukan ekstraksi HTML tersebut menghasilkan daftar DNSBL.
3. Daftar DNSBL kemudian disimpan kedalam file berekstensi csv.

b. Proses training

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 melakukan text mining dengan melakukan stemming, tokenisasi dan stop word removal.
3. Backend melakukan pemodelan berdasarkan hasil text mining tersebut.
4. Hasil training / model kemudian disimpan dalam file berekstensi crt.

- c. Proses prediksi menggunakan DNSBL Filter
 - 1. Backend membaca domain dari alamat email pengirim.
 - 2. 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*.
 - 3. Ubah susunan ip dari a.b.c.d menjadi d.c.b.a. Sehingga susunanya menjadi *113.194.217.172*.
 - 4. Backend mengambil daftar DNSBL dari file csv.
 - 5. Untuk setiap DNSBL pada daftar, konkat ip dengan DNSBL tersebut. Sebagai contoh DNSBL yang akan dipakai adalah *sbl.spamhaus.org*. Sehingga susunanya menjadi *113.194.217.172.sbl.spamhaus.org*.
 - 6. DNS lookup tipe TXT pada ip yang sudah di konkat. Apabila hasilnya sukses maka domain tersebut telah terblokir pada DNSBL yang terkait dan sebaliknya.
 - 7. Lakukan pengulangan terhadap setiap DNSBL pada daftar.

- d. Proses prediksi menggunakan SVM Filter

- 1. Backend melakukan text mining dengan melakukan stemming, tokenisasi dan stop word removal pada konten email.
- 2. Backend melakukan prediksi pada hasil text mining tersebut.

- e. Proses prediksi keseluruhan

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 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 prediksi menggunakan SVM Filter apabila terdeteksi sebagai spam maka akan ditampilkan pesan ke aplikasi Frontend Web, jika tidak lanjut ke tahap berikutnya.
6. Backend melakukan pengiriman email.

3.4.3 Coding / Pengkodean

Setelah tahap perencanaan dan perancangan, selanjutnya masuk ke tahap pengkodean yang harus sesuai dengan tahap perancangan yang sudah ditulis diatas. Kali ini penulis akan menggunakan bahasa pemrograman Kotlin untuk pembangunan sistem peringatan dini untuk mendeteksi spam pada email untuk MTTarget.co. Penulis menggunakan bahasa pemrograman Kotlin karena MTTarget.co menggunakan Bahasa tersebut sebagai bahasa utama dalam pengembangan aplikasi backend mereka.

3.4.4 Testing / Pengujian

Selanjutnya merupakan tahap pengujian Backend API, pengujian ini dilakukan dengan cara *white-box testing* untuk megecek seluruh fungsi dari sistem peringatan dini untuk mendeteksi spam pada email untuk MTTarget.co. 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 dibangun siap untuk diluncurkan.

3.4 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.

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[35]. Blackbox digunakan untuk memastikan integrasi antar komponen dapat berjalan dengan baik.

DAFTAR PUSTAKA

- [1] MailTarget, "MailTarget Company Profile," 2018.
- [2] Wikipedia, "Bounce Message," [Online]. Available: https://en.wikipedia.org/wiki/Bounce_message. [Diakses 3 Juni 2018].
- [3] H. Alkahtani, "A Taxonomy of Email SPAM Filters".
- [4] Wikipedia, "DNSBL," [Online]. Available: <https://en.wikipedia.org/wiki/DNSBL>. [Diakses 21 Juli 2018].
- [5] Wikipedia, "Support Vector Machine," [Online]. Available: https://en.wikipedia.org/wiki/Support-vector_machine. [Diakses 26 Februari 2019].
- [6] K. Bajaj, "A Multi-layer Model to Detect Spam Email at Client Side," 2016.
- [7] K. R. D, "Latent Semantic Indexing Based SVM Model for Email Spam Classification," *Journal of Scientific & Industrial Research*, vol. 73, pp. 437-442, 2014.
- [8] T. Ouyang, "A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise," *Computer Networks*, 2013.
- [9] S. K. Tuteja, "A Survey on Classification Algorithms for Email Spam Filtering," *International Journal of Engineering Science and Computing*, vol. 6, no. 5, pp. 5937-5940, 2016.
- [10] D. S. Silnov, "An Analysis of Modern Approaches to the Delivery of Unwanted Emails (Spam)," *Indian Journal of Science and Technology*, vol. 9, 2016.
- [11] Wikipedia, "Surat Elektronik," [Online]. Available: https://id.wikipedia.org/wiki/Surat_elektronik. [Diakses 20 Mei 2018].
- [12] THE RADICATI GROUP, INC., "Email Statistics Report, 2015-2019," 2015.
- [13] Facebook, "Facebook Q4 2018 Results," 2018.
- [14] Twitter, "Selected Company Financials and Metrics," 2018.
- [15] Wikipedia, "Email Marketing," [Online]. Available: https://en.wikipedia.org/wiki>Email_marketing. [Diakses 30 05 2018].
- [16] R. Aldighieri, "Marketer email tracker," DMA, 2018.
- [17] Wikipedia, "Java," [Online]. Available: <https://id.wikipedia.org/wiki/Java>. [Diakses 21 Maret 2019].
- [18] J. Heiss, "The Advent of Kotlin: A Conversation with JetBrains' Andrey Breslav," Oracle Technology Network, April 2013. [Online]. Available: <http://www.oracle.com/technetwork/articles/java/breslav-1932170.html>. [Diakses 2 Februari 2014].
- [19] Mobius, "Андрей Бреслав — Kotlin для Android: коротко и ясно," 8 Januari 2015. [Online]. [Diakses 28 Mei 2017].
- [20] Android, "Kotlin," [Online]. Available: <https://developer.android.com/kotlin/index.html>. [Diakses 21 Juli 2018].

- [21] N. Waidyanatha, “Towards a typology of integrated functional early warning systems,” *Int. J. Crit. Infrastructures*, vol. 6, no. 1, pp. 31-51, 2010.
- [22] A. N. Pour, “MINIMIZING THE TIME OF SPAM MAIL DETECTION BY RELOCATING FILTERING SYSTEM TO THE SENDER MAIL SERVER,” *International Journal of Network Security & Its Applications (IJNSA)*, vol. 4, no. 2, pp. 53-62, 2012.
- [23] Tech Term, “Client–server model,” [Online]. Available: https://techterms.com/definition/client-server_model. [Diakses 10 Maret 2019].
- [24] N. o. i. u. w. f. t. 2. (. millions), Statista, [Online]. Available: <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>. [Diakses 11 Maret 2019].
- [25] Petra, “Apa itu World Wide Web ?,” [Online]. Available: <http://faculty.petra.ac.id/dwikris/docs/desgrafisweb/www/4-apaitu-www.html>. [Diakses 11 Maret 2019].
- [26] Mozilla, “HTML,” [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML. [Diakses 11 Maret 2019].
- [27] Mozilla, “HTTP,” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP>. [Diakses 9 Maret 2019].
- [28] Oracle, “What Are Web Services?,” [Online]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>. [Diakses 11 Maret 2019].
- [29] S. Deering, “Do you know what a REST API is?,” [Online]. Available: <https://www.sitepoint.com/developers-rest-api/>. [Diakses 11 Maret 2019].
- [30] E. S. a. K. Gustavsson, “Efficient data communication between a webclient and a cloud environment,,” 2016.
- [31] Wikipedia, “Sistem Penamaan Domain,” [Online]. Available: https://id.wikipedia.org/wiki/Sistem_Penamaan_Domain. [Diakses 11 Juli 2018].
- [32] Wikipedia, “DNSBL,” [Online]. Available: <https://en.wikipedia.org/wiki/DNSBL>. [Diakses 21 Juli 2018].
- [33] J. Jung, “An Empirical Study of Spam Traffic and the Use of DNS Black Lists,” 2004.
- [34] G. Boeing, “New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings,” *Journal of Planning Education and Research* , 2016.
- [35] Wikipedia, “Web Scrapping,” [Online]. Available: https://en.wikipedia.org/wiki/Web_scraping. [Diakses 21 Oktober 2018].
- [36] Wikipedia, “Pemelajaran Mesin,” [Online]. Available: https://id.wikipedia.org/wiki/Pemelajaran_mesin. [Diakses 11 Maret 2019].
- [37] S. Nidhra, “BLACK BOX AND WHITE BOX TESTING TECHNIQUES – ALITERATURE REVIEW,” *International Journal of Embedded Systems and Applications (IJESA)*, vol. 2, no. 2, pp. 29-50, 2012.
- [38] D. Allen, “Windows To Linux,” dalam *Network+ Guide To Networks*, Prentice Hall, 2010, p. 192.

SISTEM PERINGATAN DINI UNTUK MENDETEKSI SPAM PADA EMAIL MENGGUNAKAN DNSBL FILTER DAN SVM

ORIGINALITY REPORT



MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

11%

★ id.wikipedia.org

Internet Source

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off