

**DETEKSI SPAM PADA MEDIA SOSIAL X BERDASARKAN
POST DAN REPOST DENGAN MENGGUNAKAN
METODE RANDOM FOREST CLASSIFIER**

SKRIPSI

SURYANA MEISSARAH ZAINI SINAGA

191402055



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

MEDAN

2024

**DETEKSI SPAM PADA MEDIA SOSIAL X BERDASARKAN POST DAN
REPOST DENGAN MENGGUNAKAN METODE RANDOM FOREST
CLASSIFIER**

SKRIPSI

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah sarjana
Teknologi Informasi**

SURYANA MEISSARAH ZAINI SINAGA

191402055



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

PERSETUJUAN

Judul : DETEKSI SPAM PADA MEDIA SOSIAL X
BERDASARKAN POST DAN REPOST DENGAN
MENGUNAKAN METODE RANDOM FOREST
CLASSIFIER

Kategori : SKRIPSI

Nama Mahasiswa : SURYANA MEISSARAH ZAINI SINAGA

Nomor Induk Mahasiswa : 191402055

Program Studi : SARJANA (S-1) TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 11 Juli 2024

Komisi Pembimbing:

Pembimbing 2,

Pembimbing 1,



Fanindia Purnamasari S.TI., M.IT

NIP. 198908172019032023



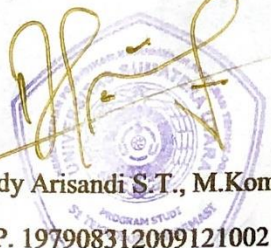
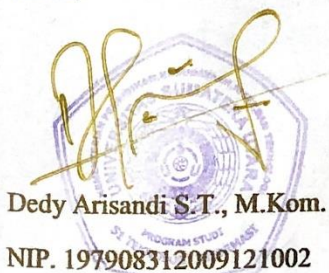
Ivan Jaya S.Si., M.Kom.

NIP. 198407072015041001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,



Dedy Arisandi S.T., M.Kom.

NIP. 197908312009121002

PERNYATAAN

**DETEKSI SPAM PADA MEDIA SOSIAL X BERDASARKAN POST DAN REPOST
DENGAN MENGGUNAKAN METODE RANDOM FOREST CLASSIFIER**

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dari ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Juli 2024

Suryana Meissarah Zaini Sinaga

191402055

UCAPAN TERIMA KASIH

Puji serta syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena limpahan berkat dan karunia-Nya sehingga karya tulis ini bisa terselesaikan sesuai ketentuan untuk dapat menyandang gelar S1 dari Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara dan memperoleh gelar Sarjana.

Dalam proses penyelesaian skripsi ini, penulis telah menerima begitu banyak bantuan, bimbingan, dukungan dan doa dari banyak pihak. Oleh karena itu penulis menghaturkan rasa terima kasih yang tulus dan dalam:

1. Kepada orang tua penulis, Bapak Hasoloan Sinaga dan Ibu Sarmalina Pardede yang selalu memberikan dukungan, didikan, dan motivasi kepada penulis agar tetap semangat hingga saat ini. Terimakasih sudah menemani perjalanan hidup penulis dan untuk doa yang sangat luar biasa sehingga penulis bisa menyelesaikan skripsi.
2. Kepada ketiga saudara Yoan Sinaga, Kartika Natasha Sinaga, dan Indah Angelita Sibaga yang senantiasa mendukung setiap proses. Terimakasih untuk segala pembelajaran, saran, solusi, motivasi serta doa baik yang diberikan kepada penulis.
3. Kepada Bapak Dr. Muryanto Amin, S.Sos., M.Si., selaku Rektor Universitas Sumatera Utara.
4. Kepada Ibu Dr. Maya Silvi Lydia, M.Sc selaku Dekan Fasilkom-TI Universitas Sumatera Utara.
5. Kepada Bapak Dedy Arisandi S.Kom., M.Kom selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
6. Kepada Bapak Ivan Jaya S.Si., M.Kom selaku Dosen Pembimbing 1 yang telah berkenan meluangkan waktu untuk memberikan arahan, bimbingan, motivasi serta kritik dan saran kepada penulis.
7. Kepada Ibu Fanindia Purnamasari S.TI., M.IT selaku Dosen Pembimbing 2 yang juga telah berkenan memberikan arahan, masukan, kritik dan saran kepada penulis.

8. Kepada seluruh Dosen Program Studi Teknologi Informasi yang sudah memberikan banyak pembelajaran dan pengetahuan kepada penulis selama perkuliahan.
9. Kepada Staff dan Pegawai Akademik Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang sudah membantu penulis melengkapi kebutuhan administrasi dalam menyelesaikan studi.
10. Kepada diri sendiri yang akhirnya sudah sampai di tahap ini. Terimakasih sudah bangkit ketika jatuh, bertahan hidup dengan baik di perantauan, dan untuk tidak menyerah dalam menyelesaikan skripsi ini semaksimal mungkin.
11. Kepada teman seperjuangan penulis semasa diperkuliahan, Grace Ogestin Pasaribu, Jogiana Simangunsong, Legi Maria Silaban, Sheren Siahaan, Sinthia Audrey, dan Godblessus Simaremare yang telah banyak membantu, saling mendukung dan mendoakan satu sama lain agar tetap semangat untuk menyelesaikan perkuliahan. Semoga kita semua bisa sukses bersama.
12. Kepada teman seperjuangan penulis, Margaretha Naibaho, Indah Nainggolan, dan Annisa Amaliah yang sudah membantu penulis dalam pembelajaran serta menyelesaikan skripsi ini.
13. Kepada sahabat penulis, Niken, Lady dan Grecia orang yang selalu menampung semua kesedihan, curhatan dan selalu menemani penulis.
14. Teman-teman dan semua pihak yang tidak dapat penulis tuliskan satu persatu yang sudah memberi support kepada penulis dan membantu menyelesaikan perkuliahan.

Semoga Tuhan Yang Maha Esa senantiasa selalu memberkati kita semua kedepannya.

Medan, 20 Juni 2024

Suryana Meissarah Zaini Sinaga

ABSTRAK

Di era globalisasi, teknologi berkembang pesat sehingga memudahkan komunikasi melalui media sosial. Salah satu media sosial yang banyak digunakan adalah X, sebelumnya dikenal sebagai Twitter. X memiliki dampak besar dalam industri, bisnis, dan politik, dengan 19,5 juta pengguna di Indonesia dari total 500 juta pengguna global. Namun, popularitasnya juga menarik *spammer* untuk melakukan kegiatan spam, seperti kampanye politik, penyebaran informasi menyesatkan, dan promosi tidak relevan. Spam adalah pesan massal yang tidak diinginkan oleh penerima, mengganggu privasi dan kenyamanan pengguna. Oleh karena itu, diperlukan penelitian untuk mendeteksi post spam dan bukan spam, guna meningkatkan kenyamanan dan keamanan pengguna X. Penelitian ini bertujuan untuk mendeteksi spam berbahasa Indonesia pada media sosial X berdasarkan post dan repost dengan menggunakan *Random Forest Classifier* dan TF-IDF. Penelitian ini menggunakan 2800 data yang merupakan post dan repost dari akun pengguna media sosial X. Tahapan preproses yang dilakukan pada penelitian ini yaitu menghapus variable yang tidak diinginkan, menghapus emoji, mengubah kata menjadi huruf kecil, penghapusan tanda baca atau symbol, normalisasi, stopword, serta tokenisasi. Penelitian ini menggunakan word embedding yaitu TF-IDF untuk mengubah kata yang ada dalam data menjadi vektor dan akan diidentifikasi menggunakan metode *Random Forest Classifier*. Metode evaluasi pada penelitian ini adalah *Confussion Matrix* dan menghasilkan akurasi sebesar 0,97. Berdasarkan hasil evaluasi yang diperoleh, maka dapat disimpulkan bahwa algoritma yang digunakan pada penelitian ini mampu mendeteksi post dan repost spam dengan kinerja yang tinggi.

Kata kunci: deteksi spam, *spammer*, *tf idf*, *random forest classifier*, *confussion matrix*

SPAM DETECTION ON SOCIAL MEDIA X BASED ON POST AND REPOST USING RANDOM FOREST CLASSIFIER

ABSTRACT

In the era of globalization, technology is rapidly advancing, it makes communication be easier through social media. One of the most widely used platform is X, formerly known as Twitter. X has had a huge impact on industry, business, and politic, with 19.5 million users in Indonesia out of a global total of 500 million. However, its popularity also attracts spammers who engage in activities such as political campaigns, dissemination of misleading information, and irrelevant promotions. Spam, defined as unwanted mass messages, disrupts user privacy and convenience. Therefore, research is needed to detect spam and non-spam posts, in order to enhance user the convenience and security of the users. This study aims to detect Indonesian-language spam on social media X based on posts and reposts using the Random Forest Classifier and TF-IDF. The study use 2800 data posts and reposts from X user accounts. Preprocessing stages included removing unwanted variables, emojis, change words to lowercase, removing punctuation or symbols, normalization, stop-word removal, and tokenization. The research employed TF-IDF for word embedding to convert words in the data into vector, which will be identified using the Random Forest Classifier method. The evaluation methods of this research is Confusion Matrix, resulting in an accuracy of 0.97. Based on the evaluation outcomes, it can be concluded that the algorithm used in this study effectively detects spam posts and reposts with high performance.

Keywords: spam detection, spammer, tf idf, random forest classifier, confusion matrix

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH.....	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	5
BAB 2	7
LANDASAN TEORI.....	7
2.1 Media Sosial.....	7
2.2 X.....	7
2.3 Spam.....	9
2.4 Text Pre-processing.....	10
2.5 Random Forest Classifier	11
2.6 Web scraping.....	13
2.7 Term Frequency dan Inverse Document Frequency	13
2.8 Penelitian Terdahulu	15
2.9 Perbedaan Penelitian	19
BAB 3	20

ANALISIS DAN PERANCANGAN SISTEM	20
3.1 Data yang Digunakan	20
3.2 Arsitektur Umum.....	22
3.3 Pre-processing	23
3.3.1 Cleaning	23
3.3.2 Case Folding.....	24
3.3.3 Punctuation Removal	25
3.4.4 Normalization.....	25
3.4.5 Stopword Removal.....	26
3.4.6 Stemming	27
3.4.7 Tokenizing.....	28
3.5 Word Embedding	29
3.6 Random Forest Classifier	30
3.7 Metode Evaluasi.....	31
3.8 Perancangan Sistem.....	32
3.8.4 Rancangan Tampilan Beranda	32
3.7.2 Rancangan Tampilan Training Data	33
3.7.3 Rancangan Tampilan Testing	34
BAB 4	36
IMPLEMENTASI DAN PENGUJIAN SISTEM.....	36
4.1 Implementasi Sistem	36
4.1.1 Spesifikasi Perangkat Keras.....	36
4.1.2 Spesifikasi Perangkat Lunak.....	36
5.2 Implementasi Perancangan Tampilan Interface	37
4.2.1 Tampilan Halaman Beranda	37
4.2.2 Tampilan Halaman Training.....	37
4.2.3 Tampilan Halaman Testing.....	39
4.2.4 Tampilan Halaman User Input.....	40
5.3 Implementasi Model.....	42
5.3.1 Menentukan Nilai Vektor Kata dengan TF-IDF.....	42

5.3.2	Pelatihan Model Random Forest Classifier	44
5.4	Hasil Pengujian Sistem.....	46
5.5	Hasil Evaluasi.....	50
BAB 5	53
KESIMPULAN DAN SARAN	53
5.1	KESIMPULAN	53
5.2	SARAN	53
DAFTAR PUSTAKA	54

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu	17
Tabel 3. 1 Dataset Post dan Repost Twitter	21
Tabel 3. 2 Pembagian Dataset.....	22
Tabel 3. 3 Penerapan Tahap <i>Cleaning</i>	24
Tabel 3. 4 Penerapan Tahap <i>Case Folding</i>	25
Tabel 3. 5 Penerapan Tahap <i>Punctuation Removal</i>	25
Tabel 3. 6 Penerapan Tahap <i>Normalization</i>	26
Tabel 3. 7 List Stopword Bahasa Indonesia.....	27
Tabel 3. 8 Penerapan Tahap <i>Stopword Removal</i>	27
Tabel 3. 9 Penerapan Tahap <i>Stemming</i>	28
Tabel 3. 10 Penerapan <i>Tokenizing</i>	29
Tabel 3. 11 Penerapan <i>Confussion Matrix</i>	31
Tabel 4. 1 Performansi <i>Hyperparameters Tuning</i>	45
Tabel 4. 2 Hasil Pengujian Sistem	46
Tabel 4. 3 Keterangan Post Spam <i>Confussion Matrix</i>	51
Tabel 4. 4 Keterangan Post Non-spam <i>Confussion Matrix</i>	51
Tabel 4. 5 Hasil Evaluasi	52

DAFTAR GAMBAR

Gambar 2. 1 Contoh post spam pada media social X	10
Gambar 3. 1 Arsitektur Umum	23
Gambar 3. 2 Embedding Matrix “ayo”	29
Gambar 3. 3 Rancangan tampilan beranda	32
Gambar 3. 4 Rancangan tampilan <i>training data</i>	33
Gambar 3. 5 Rancangan Tampilan Testing Data	34
Gambar 4. 1 Tampilan halaman beranda	37
Gambar 4. 2 Tampilan halaman <i>training</i> sebelum dilakukan proses training	38
Gambar 4. 3 Tampilan hasil <i>training</i> data	38
Gambar 4. 4 Tampilan halaman <i>testing</i> data	39
Gambar 4. 5 Tampilan halaman hasil proses <i>testing</i>	40
Gambar 4. 6 Tampilan hasil evaluasi.....	40
Gambar 4. 7 Tampilan Halaman User Input	41
Gambar 4. 8 Tampilan Halaman <i>User Input</i> (Post Spam)	41
Gambar 4. 9 Tampilan Halaman <i>User Input</i> (Post Non-spam)	42
Gambar 4. 10 Vektor dari kata ‘ayo’	43
Gambar 4. 11 Kamus kata dengan nilai rata-rata vektor	44
Gambar 4. 12 Implementasi metode klasifikasi Random Forest	45
Gambar 4. 13 Confusion Matrix	50

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di era globalisasi, perkembangan teknologi sudah sangat pesat. Teknologi membuat jarak tak lagi jadi masalah dalam berkomunikasi melalui media sosial. Berdasarkan laporan We Are Social, jumlah pengguna aktif media sosial di Indonesia sebanyak 191 juta orang pada Januari 2022. Jumlah itu telah meningkat 12,35% dibandingkan pada tahun sebelumnya yang sebanyak 170 juta orang. Salah satu media sosial yang banyak digunakan oleh masyarakat adalah X.

Media sosial X atau yang sebelumnya kita kenal dengan nama *Twitter* merupakan salah satu layanan media sosial populer yang digunakan sebagai sarana komunikasi yang menghubungkan antar pengguna. Pemilik *Twitter*, Elon Musk melakukan perubahan merek (*rebranding*) *Twitter* sejak 23 Juli 2023 dengan mengganti logo hingga perubahan nama aplikasi menjadi X. Pada 31 Juli 2023, *Twitter* akhirnya resmi berganti nama menjadi "X" di toko aplikasi *App Store*. X menjadi aplikasi pertama di *App Store* yang memiliki nama dengan satu karakter. Fitur *tweet* dan *retweet* juga diubah namanya menjadi *post* (posting) dan *repost* (posting ulang) namun dengan fungsi yang masih sama yaitu membagi informasi ke pengguna lain. Post adalah tulisan yang berisi teks, foto, video dan URL HTTP terbatas hingga 280 karakter bagi pengguna yang tidak premium dan 4000 karakter bagi pengguna X premium. Repost adalah aktivitas melakukan posting ulang sebuah postingan. X sebagai media sosial yang memiliki dampak besar dalam beberapa, seperti bidang industri, bisnis serta politik. Berdasarkan data PT. Bakrie Telecom, memiliki 19,5 juta pengguna di Indonesia dari total 500 juta pengguna global. Popularitas X ini menjadi target yang menarik pengguna yang tidak bertanggung jawab untuk melakukan kegiatan spam.

Pengguna media sosial X harus mendaftar dan memiliki akun media sosial tersebut, namun masih banyak masyarakat yang menyalahgunakan akun tersebut sebagai media untuk menjadi *trending* topik, promosi atau meningkatkan popularitas dari seorang pemilik akun media sosial dengan cara kampanye politik, sarana protes, menyebarkan informasi dan berita yang menyesatkan, dan sebagainya. X dihadapkan pada berbagai masalah seperti gangguan privasi pengguna dan spam. Spam adalah pesan atau tulisan yang dikirimkan secara massal tanpa dikehendaki oleh penerimanya. Arti dari “secara massal” yaitu pesan atau tulisan yang merupakan bagian dari sekumpulan pesan atau tulisan yang memiliki isi yang sama (Spamhaus, 2004). Tindakan menyebarkan spam disebut dengan *spamming*, sedangkan orang yang melakukan *spamming* disebut *spammer*. Bentuk post spam sangat bervariasi, baik berupa hal negatif, post jualan yang tidak berhubungan, penggunaan hastag yang ngawur, bahkan link ke suatu website berbahaya tertentu dan lain sebagainya yang mengganggu kegiatan berselancar di sosial media X.

Dengan adanya masalah spam tersebut, perlu dilakukan penelitian untuk mendeteksi tweet *spam* dan bukan *spam*. Kasus spammer telah diteliti sebelumnya khususnya akun spammer pada media sosial X. Salah satunya adalah penelitian yang dilakukan oleh Andita Wahyuningtyas, et al., (2020) dengan judul Deteksi Spam pada X Menggunakan *Algoritme Naïve Bayes*. Penelitian ini bertujuan untuk mendeteksi post spam dan bukan spam. Hal tersebut dapat dilakukan dengan klasifikasi. Terdapat berbagai macam metode klasifikasi, salah satu metode dalam *data mining* untuk mengklasifikasikan spam dan bukan spam adalah *Naïve Bayes*. Penelitian ini mengumpulkan data spam dari X dengan mengidentifikasi terlebih dahulu akun yang diduga sebagai spammer. Penelitian ini menggunakan 70% data latih dan 30% data uji dengan metode klasifikasi *Naïve Bayes*. Akurasi hasil klasifikasi post spam dan bukan spam adalah 95.57%. Metode yang dapat digunakan untuk melakukan deteksi akun spammer juga telah lama dipelajari, salah satunya adalah metode Random Forest. Metode ini merupakan pengembangan dari algoritma *Classification and Regression Tree* (CART) dimana sering digunakan dan mampu menghasilkan tingkat akurasi yang tinggi dalam bidang text classification terbukti pada penelitian Klasifikasi Topik X menggunakan Metode *Random Forest* dan Fitur Ekspansi *Word2Vec* (Rafly Ghazali Ramli & Yuliant Sibaroni, 2022). Penelitian ini menghasilkan nilai akurasi sebesar 99,49%.

Didasari oleh latar belakang masalah yang diuraikan tersebut serta penelitian-penelitian terdahulu, penulis mengajukan sebuah penelitian yang akan menghasilkan sebuah website dengan metode Random Forest yang dapat mendeteksi akun spammer pada X. Penelitian ini diberi judul DETEKSI SPAM PADA X BERDASARKAN POST DAN REPOST DENGAN MENGGUNAKAN METODE RANDOM FOREST.

1.2 Rumusan Masalah

Penggunaan X sebagai salah satu media social yang paling populer di kalangan masyarakat seringkali mengakibatkan berbagai masalah terhadap kenyamanan pengguna seperti spammer dan gangguan privasi. Kegiatan spam dilakukan dengan banyak cara, seperti post spam hal-hal negative, berjualan bahkan penyebaran link yang berbahaya. Dengan adanya masalah spammer ini, perlu adanya pencegahan terhadap serangan akun spam salah satunya dengan dilakukan klasifikasi untuk menentukan akun spam dan bukan spam melalui aktivitas penggunaannya. Untuk mempermudah melakukan klasifikasi tersebut, diperlukan website yang dapat membantu dalam mengkategorikan post spam dan bukan secara otomatis dengan menggunakan teknologi kecerdasan buatan.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mendeteksi tweet apakah tweet tersebut merupakan *spam* atau *non spam* pada media sosial X berdasarkan aktivitas pengguna dengan menggunakan *TF-IDF* dan metode *Random Forest*.

1.4 Batasan Masalah

Dalam melakukan penelitian ini, penulis membuat batasan masalah sebagai berikut:

1. Deteksi dilakukan hanya pada akun publik, tidak berlaku untuk akun privat.
2. Perilaku pengguna yang diperhatikan pada penelitian ini berfokus pada dua atribut yaitu aktivitas *post* dan *repost*.
3. Aktivitas *post* dan *repost* yang dideteksi hanya berupa kata/kalimat yang menggunakan bahasa Indonesia.
4. Output dari penelitian ini adalah sistem berbasis website.

1.5 Manfaat Penelitian

Adapun manfaat penelitian ini adalah:

1. Membantu pengguna untuk mengetahui post spam pada X dan memberikan kredibilitas informasi yang terdapat di X.
2. Membantu pengguna untuk menghemat upaya menghindari pengguna X yang melakukan kegiatan spam agar tidak mengganggu aktivitas selancar di X.
3. Menjadi referensi penelitian selanjutnya dalam melakukan identifikasi spam pada media sosial X dengan menggunakan *Random Forest*.

1.6 Metodologi Penelitian

Tahapan-tahapan yang harus dilakukan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur adalah langkah awal dalam penelitian ini, dilakukan untuk mengembangkan banyak referensi dari berbagai sumber literatur seperti buku, artikel, skripsi, jurnal, dan sumber bacaan lainnya mengenai X, post spam, Machine Learning dan metode Random Forest.

2. Analisis Permasalahan

Setelah langkah studi literatur, selanjutnya akan dilakukan analisa permasalahan untuk memahami konsep Machine Learning yang akan diterapkan dalam penelitian untuk mengidentifikasi post spam pada X dengan menggunakan Random Forest.

3. Perancangan Sistem

Setelah langkah analisa permasalahan, selanjutnya akan dilakukan perancangan untuk sistem yang akan dibangun. Pada langkah ini, akan dibangun sebuah rancangan sistem, penentuan data training dan data testing, dan rancangan arsitektur berdasarkan analisa permasalahan yang sudah dilakukan sebelumnya.

4. Implementasi

Setelah langkah perancangan, selanjutnya adalah implementasi. Pada tahap ini, semua hasil dari rancangan sebelumnya yang telah dilakukan akan diimplementasikan untuk membuat sistem.

5. Pengujian Sistem

Setelah langkah implementasi, selanjutnya akan dilakukan proses pengujian kualitas kemampuan hasil implementasi tersebut dari metode yang digunakan yaitu Random Forest.

6. Penyusunan Laporan

Setelah proses pengujian, maka selanjutnya adalah penyusunan laporan dari keseluruhan penelitian yang sudah dilakukan.

1.7 Sistematika Penulisan

Sistematika dalam penulisan penelitian ini terdiri atas lima bagian utama, antara lain:

BAB 1: Pendahuluan

Bagian pendahuluan berisikan penjabaran aspek-aspek mendasar dari penelitian, mencakup latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2: Landasan Teori

Bagian landasan teori berisi teori-teori yang dibutuhkan untuk lebih mengerti dan dapat menyelesaikan permasalahan dan mencapai tujuan pada penelitian ini. Pada bab ini dijelaskan tentang spammer serta metode yang digunakan yaitu Random Forest, TF-IDF dan lain-lain. Selain itu, dalam bab ini juga dipaparkan penelitian yang pernah dilakukan sebelumnya yang berfungsi sebagai acuan penulis dalam melakukan penelitian ini.

BAB 3: Analisis dan Perancangan Sistem

Pada bagian ini berisi penjelasan tentang analisis masalah penelitian dan perancangan sistem menggunakan Random Forest dalam hal pendeteksian post dan repost berbahasa Indonesia yang mengandung unsur spam. Pada bab ini juga terdapat penjelasan terkait dengan arsitektur umum dari sistem yang akan dibangun.

BAB 4: Implementasi dan Pengujian Sistem

Bagian ini menjelaskan terkait dengan proses implementasi dari desain sistem yang telah disusun sebelumnya pada Bab 3. Selain itu, bab ini juga mencakup penjelasan mengenai hasil pengujian sistem yang telah dibuat, serta akan dilakukan proses evaluasi terhadap hasil yang telah diperoleh.

BAB 5: Kesimpulan dan Saran

Bagian ini berisi kesimpulan dari penelitian yang telah dilakukan, apakah tujuan dari penelitian ini tercapai, serta saran yang diutarakan oleh penulis sehingga memungkinkan pengembangan yang lebih lanjut pada penelitian berikutnya.

BAB 2

LANDASAN TEORI

2.1 Media Sosial

Media social adalah situs jaringan sosial berbasis web yang memungkinkan bagi setiap individu untuk membangun profil publik ataupun semi publik dalam sistem terbatas, daftar pengguna lain dengan siapa mereka terhubung, dan melihat serta menjelajahi daftar koneksi mereka yang dibuat oleh orang lain dengan suatu sistem (Henderi, 2007:3).

Teknologi-teknologi web baru memudahkan semua orang untuk membuat dan yang terpenting menyebarluaskan konten mereka sendiri. Post di Blog, X, Instagram, Facebook, atau video di YouTube yang dapat direproduksi dan dilihat oleh jutaan orang secara gratis. Pemasang iklan tidak harus membayar banyak uang kepada penerbit atau distributor untuk memasang iklannya. Sekarang pemasang iklan dapat membuat konten sendiri yang menarik dan dilihat banyak orang (Zarrella, 2010, h. 2).

2.2 X

X atau yang kerap kita kenal sebagai Twitter merupakan salah satu jejaring sosial online paling populer dan paling cepat berkembang yang didirikan pada tahun 2006. Selain merubah nama dan logo, fitur tweet dan retweet juga diubah namanya menjadi post dan repost namun dengan fungsi yang masih sama. X digunakan sebagai sarana komunikasi yang menghubungkan antar pengguna diseluruh belahan dunia. X telah muncul sebagai platform microblogging paling populer dimana pengguna dapat mengakses berbagi berita, media, meme, pandangan, dan pembaruan dalam bentuk post. Post adalah tulisan yang berisi teks, foto, video dan URL HTTP terbatas hingga 280 karakter bagi pengguna yang tidak premium dan 4000 karakter bagi pengguna X premium. Banyak mesin pencari populer seperti Yahoo, Microsoft Bing, dan Google menggunakan aliran X untuk melacak

pembaruan langsung dari kejadian di seluruh dunia untuk memberikan informasi secara praktis tanpa penundaan.

Saat ini X menyediakan X Premium yang merupakan layanan langganan premium untuk meningkatkan percakapan berkualitas di platform. X Premium adalah langganan berbayar opsional yang menambahkan tanda centang biru ke akun pengguna dan menawarkan akses lebih awal ke fitur-fitur tertentu, misalnya Edit postingan dan Postingan lebih panjang. X Premium diberi harga sesuai-lokasi mulai dari \$8/bulan atau \$84/tahun di negara yang tersedia untuk mendapatkan tanda centang biru serta akses lebih awal ke berbagai fitur.

Tampilan aplikasi X adalah sebagai berikut:

1. Notifikasi/Notifications: berisi pemberitahuan apabila ada mention masuk yang berupa follow, mention, replies, repost atau like.
2. Pesan/Direct Message: untuk mengirim pesan langsung kepada pengguna lain tanpa diketahui oleh orang lain.
3. Sorotan/Highlights: sorotan tersedia bagi user yang sudah verified atau berlangganan Twitter blue. Sorotan berisi postingan favorit/terbaik yang ingin ditampilkan di profil dalam satu tab.
4. Disukai/likes: berisi postingan-postingan yang disukai oleh pengguna
5. Trending topic: berisi informasi yang sedang ramai dibicarakan, kata/frasa maupun hashtag yang sedang trending akan muncul di kolom trending topic Twitter berdasarkan algoritma yang diminati.
6. Cari X/Search X: untuk menemukan post/kicauan, atau pengguna lain di X.
7. Komunitas/Community: dimulai dan dikelola oleh orang di X, admin dan moderator yang menegakkan peraturan Komunitas dan menjaga percakapan tetap informatif, relevan, dan menyenangkan. Orang yang menerima undangan untuk bergabung ke Komunitas akan menjadi anggota. Post dalam Komunitas dapat dilihat oleh siapa pun di X, tetapi hanya orang dalam Komunitas itu sendiri yang dapat berinteraksi dan berpartisipasi dalam diskusi. (X, 2023).
8. Posting/Post: untuk membuat status/post baru yang ingin kita update, bisa berupa teks, URL, gif, foto, video atau spaces.
9. Foto dan video: untuk menambahkan gambar atau video ketika kita akan membuat atau meng-update status.

10. *Timeline*: tempat menampung kicauan pengguna X. Post yang muncul di *timeline* berdasarkan kurun waktu.
11. Markah atau *Bookmark*: fitur markah memudahkan pengguna menyimpan post di timeline agar dapat diakses dengan mudah dan cepat kapan saja. Atau, pengguna dapat melihat markah yang disimpan dengan membuka tab “Markah” di menu navigasi. Untuk menghapus markah yang disimpan, sentuh ikon Markah dari detail sebuah Post dalam timeline Markah disimpan.

X memiliki kebijakan untuk melarang kegiatan spam pada aplikasinya (X, 2019). Hal ini dilakukan karena X ingin memberikan layanan yang aman, handal dalam memberikan informasi, dan memberikan kebebasan para pengguna untuk saling berkomunikasi. Beberapa contoh pelanggaran dalam kebijakan X yaitu spam dengan motivasi komersial, dimana biasanya diarahkan kepada web lain baik produk, layanan, ataupun program dengan menggunakan URL. Pengguna dilarang untuk membuat keterlibatan yang tidak sah dengan membuat akun atau konten yang terlihat aktif atau populer. Pengguna dilarang untuk melakukan koordinasi untuk mempengaruhi percakapan dengan menggunakan lebih dari satu akun (akun palsu), otomatisasi, ataupun penggunaan *script*.

Beberapa hal yang dianggap mengganggu informasi oleh X yaitu pengguna yang mengoperasikan akun lebih dari satu, hal ini ditandai dengan data yang identik atau konten yang sangat mirip. Pengguna menjalankan beberapa akun yang terlibat komunikasi yang bertujuan menaikkan ataupun memanipulasi suatu post dengan tujuan untuk membuat pembahasan menjadi menonjol. Pengguna melakukan pengiriman konten yang duplikat seperti post dan hashtag yang memiliki tingkat kemiripan tinggi.

2.3 Spam

Definisi spam dalam KBBI adalah surat yang dikirim tanpa diminta melalui internet, biasanya berisi iklan. Spam merupakan penyalahgunaan pengiriman pesan/teks tanpa dikehendaki oleh penerimanya, orang yang mengirimkan spam disebut spammer. (Christian, et al., 2016). Akun spammer cenderung melakukan aktivitas yang berlebihan, seperti mengirim banyak pesan otomatis (Aditya et al., 2019). Para spammer menuliskan berbagai komentar tentang bisnis mereka (promo/berjualan), atau link spam, dan berbagai hal lain yang tentu sangat mengganggu.

Pelaku spammer biasanya didorong oleh beberapa tujuan, seperti untuk menyebarkan iklan untuk menghasilkan penjualan suatu barang tertentu, menyebarkan konten pornografi, virus, phishing, atau sekadar untuk membahayakan sebuah sistem. Mereka tidak hanya mencemari pencarian real-time, tapi mereka bisa juga mengganggu statistik yang disajikan oleh tweet mining tools dan mengonsumsi sumber daya ekstra dari pengguna dan sistem. Spam hanya membuang perhatian manusia, mengingat spammer semakin meningkat. (Benevenuto, 2011).

Beberapa ciri dari akun spammer adalah: postingan mengandung informasi yang tidak jelas dan tidak relevan, terlalu banyak Hastag atau Tagar (#), Mengandung kata-kata atau frasa tertentu, isi post hanya promosi, mengandung tautan atau lampiran yang mencurigakan, mengirim informasi dari sumber yang tidak jelas atau tidak diketahui, pesan atau postingan yang tidak diinginkan dan masih banyak ciri lainnya (UrbanJabar.com, 2023). Contoh post spam dapat dilihat pada gambar 2.1.



Gambar 2. 1 Contoh post spam pada media social X

2.4 Text Pre-processing

Text Pre-processing merupakan proses untuk menyeleksi data agar data yang akan dipakai dan diolah dalam penelitian menjadi lebih terstruktur dan dapat memudahkan penelitian. Tahap ini dilakukan guna untuk membersihkan data yang awalnya merupakan hasil crawling dan belum diolah sehingga diubah menjadi data yang lebih teratur dan lebih dimengerti oleh mesin dengan tujuan untuk mendapatkan performa yang lebih baik dalam proses identifikasi.

Text preprocessing perlu dilakukan dengan tujuan untuk mengurangi karakter-karakter yang tidak memiliki makna dimana karakter-karakter tersebut dapat mengurangi kinerja dari algoritma yang digunakan. Adapun proses yang terjadi dalam text preprocessing antara lain *cleaning*, *case folding*, *normalization*, *stopword removal*, *stemming* dan *tokenizing*.

2.5 Random Forest Classifier

Random Forest adalah salah satu algoritma *Supervised Learning* yang dikeluarkan oleh Breiman pada tahun 2001, dan biasanya digunakan untuk menyelesaikan masalah yang berhubungan dengan klasifikasi, regresi, dan lainnya. Random Forest merupakan metode atau algoritma dari teknik pohon keputusan yang digunakan untuk pengklasifikasi dan regresi. Metode ini merupakan sebuah esemble (kumpulan) pohon keputusan sebagai dasar dari Random Forest Classifier yang dibangun dan dikombinasikan. Salah satu aspek penting dalam metode Random Forest adalah penggunaan *bootstrap sampling* untuk membangun pohon prediksi. Setiap pohon keputusan dalam Random Forest memprediksi secara acak, dan Random Forest itu sendiri melakukan prediksi keputusan berdasarkan hasil voting dari seluruh pohon dalam ensemble. Model dari Random Forest terdiri dari K pohon keputusan. Setiap pohon memilih variabel x yang paling sesuai dengan pohon tersebut.

Dalam proses klasifikasi, suara tunggal diberikan kepada pohon yang dipilih, dan prediksi akhir dilakukan berdasarkan mayoritas suara dari pohon-pohon tersebut. Hal ini memungkinkan Random Forest dapat mengatasi masalah overfitting dan memberikan hasil prediksi yang lebih stabil dan akurat. Random Forest Classifier juga memiliki kemampuan untuk melakukan pengklasifikasian dengan variabel target yang memiliki kelas diskrit serta regresi dengan variabel target yang bersifat berkelanjutan. Metode ini sangat populer dan efektif dalam banyak aplikasi, termasuk dalam analisis data, pemrosesan citra, dan bioinformatika. Kelebihan lain dari Random Forest adalah kemampuannya dalam menangani data yang tidak seimbang dan kemampuan untuk mengidentifikasi pentingnya setiap variabel dalam klasifikasi. Dengan kombinasi dari pohon keputusan acak, Random Forest mampu memberikan prediksi yang handal dan dapat diandalkan. Adapun tahapan yang dilakukan random forest sebagai berikut:

a. Bagging

Dilakukan pemilihan sampel acak dari data latih dengan penggantian. Ini berarti setiap sampel dapat dipilih lebih dari sekali, dan beberapa sampel dapat diabaikan. Ini memungkinkan setiap pohon dalam Random Forest memiliki data pelatihan yang berbeda-beda. Dengan menggabungkan berbagai sampel acak menghasilkan keragaman di antara pohon-pohon dalam hutan, yang mendukung pembentukan model yang lebih kuat (Rahmi et al., 2023).

b. Pertumbuhan pohon

Proses pertumbuhan pohon dimulai dengan memilih variabel acak sebagai pembagi yang menghasilkan pemisahan terbaik antara kelas dalam data pelatihan. Dalam setiap langkah, data pelatihan dibagi berdasarkan nilai pembagi yang dipilih, membentuk cabang-cabang baru dalam pohon (Rahmi et al., 2023).

c. Prediksi

Setelah semua pohon tumbuh, Random Forest digunakan untuk membuat prediksi. Setiap pohon dalam hutan memberikan prediksi berdasarkan fitur input. Jika menggunakan Random Forest untuk klasifikasi, prediksi akhir dihasilkan dengan memilih kelas yang paling sering muncul di antara prediksi pohon-pohon tersebut. Jika kita menggunakan Random Forest untuk regresi, prediksi akhir dihasilkan dengan mengambil rata-rata dari prediksi pohon-pohon tersebut. Dengan menggabungkan hasil prediksi dari pohon-pohon yang berbeda, Random Forest dapat menghasilkan prediksi yang lebih stabil dan akurat daripada menggunakan satu pohon saja (Jatmiko et al., 2019).

Entropy dibutuhkan dalam metode Random Forest untuk mengukur tingkat ketidakmurnian atau ketidakaturan suatu himpunan data. Dalam konteks Random Forest, entropy digunakan untuk menentukan atribut mana yang paling baik dalam memisahkan data menjadi kelompok yang homogen. Semakin rendah nilai entropy, semakin baik atribut tersebut dalam melakukan pemisahan. Dengan mempertimbangkan entropy, Random Forest dapat memilih atribut yang paling informatif dan efektif dalam membangun pohon keputusan, sehingga meningkatkan akurasi dan kualitas prediksi dari model (Sandag, 2020).

$$Entropy(Y) = -\sum p(c|Y) \log_2 p(c|Y) \quad (1)$$

Keterangan:

Y = Himpunan kasus

(c|Y) = Proporsi nilai Y terhadap kelas c.

$$Entropy(Y) = \sum_{v \in \text{values}(a)} \frac{Y_v}{Y_a} Entropy(Y_v) \quad (2)$$

Keterangan:

Values (a) = Nilai yang mungkin dalam himpunan kasus a.

Y_v = Subkelas dari Y dengan kelas v yang berhubungan kelas a.

Y_a = Semua nilai yang sesuai dengan a.

2.6 Web scraping

Web scraping merupakan proses pengambilan sebuah dokumen semi terstruktur dari internet, umumnya berupa halaman-halaman web dalam bahasa markup seperti HTML atau XHTML dan menganalisis dokumen tersebut untuk diambil data tertentu dari halaman tersebut untuk digunakan bagi kepentingan lain (Turland, 2010). Secara teknis web scraping tidak termasuk dalam bidang data mining karena data mining mengandung upaya untuk memahami pola semantik atau tren dalam kumpulan data yang besar yang telah diperoleh. Aplikasi web scraping berfokus hanya pada mendapatkan data dengan cara pengambilan dan ekstraksi.

Langkah-langkah melakukan web *scraping* yaitu sebagai berikut:

- a. Mempelajari atau observasi terhadap struktur html halaman web target.
- b. Ekstraksi potongan-potongan data yang relevan dari halamannya.
- c. Penyaringan, pemrosesan data untuk disimpan dalam database.

2.7 Term Frequency dan Inverse Document Frequency

Term Frequency (TF) dan *Inverse Document Frequency* (TF-IDF) merupakan metode untuk menghitung bobot kata yang umum digunakan dalam sistem temu kembali informasi. Metode ini akan menghitung nilai *Term Frequency* (kemunculan

kata) dan *Inverse Document Frequency* (IDF) pada setiap token atau kata di dalam dokumen (Arfian, 2016).

Cara kerja TF-IDF:

1. Term Frequency (TF):

TF mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Representasinya adalah jumlah kemunculan kata tersebut dalam dokumen.

Contoh: Jika kata "data" muncul sebanyak 5 kali dalam dokumen, nilai TF untuk "data" adalah 5.

2. Inverse Document Frequency (IDF):

IDF mengukur seberapa penting suatu kata dalam seluruh koleksi dokumen. Ini dilakukan dengan menghitung logaritma dari jumlah total dokumen dibagi oleh jumlah dokumen yang mengandung kata tersebut.

Rumus IDF: $\log \left(\frac{\text{jumlah total dokumen}}{\text{jumlah dokumen yang mengandung kata}} \right)$

Contoh: Jika terdapat 1000 dokumen dalam koleksi dan kata "data" muncul di 100 dokumen, nilai IDF untuk "data" adalah $\log \frac{1000}{100}$

3. TF-IDF:

Nilai TF-IDF untuk suatu kata dalam suatu dokumen dihitung dengan mengalikan nilai TF dengan nilai IDF untuk kata tersebut.

Rumus TF-IDF: $TF\text{-}IDF = TF \times IDF$

Contoh: Jika TF untuk "data" dalam suatu dokumen adalah 5 dan IDF untuk "data" adalah 2, maka nilai TF-IDF untuk "data" dalam dokumen tersebut adalah 10.

Rumus untuk menghitung nilai idf dapat dilihat pada persamaan berikut:

$$IDF_j = \log \left(\frac{D}{df_j} \right) \quad (3)$$

Keterangan:

D = jumlah semua dokumen

Df_j = jumlah dokumen yang mengandung kata atau tf

Rumus untuk menghitung bobot kata pada dokumen dapat dilihat pada persamaan berikut:

$$W_{dt} = TF_{dt} * IDF_t \quad (4)$$

Keterangan:

d = dokumen ke- d

T = kata ke- t dari kata kunci

W = bobot dokumen ke- d terhadap kata ke- t

tf = banyaknya kata yang dicari pada sebuah dokumen

IDF = kemunculan kata di banyak dokumen

2.8 Penelitian Terdahulu

Beberapa penelitian terkait yang dilakukan sebelumnya yaitu oleh Andita Wahyuningtyas, et al., (2020) dengan judul Deteksi Spam pada X Menggunakan *Algoritme Naïve Bayes*. Penelitian ini bertujuan untuk mendeteksi post spam dan bukan spam. Penelitian ini menggunakan 70% data latih dan 30% data uji dengan metode klasifikasi *Naïve Bayes*. Akurasi hasil klasifikasi post spam dan bukan spam adalah 95.57%.

Sebelumnya banyak penelitian yang telah dilakukan, diantaranya penelitian oleh Agus Tiyanasyah Syah, et al., (2020). Klasifikasi Komentar Spam Pada Instagram Menggunakan Metode *Support Vector Machine*. Tujuan dari penelitian ini adalah membangun sebuah system yang dapat mengklasifikasi bahwa suatu komentar adalah *spam* atau *nonspam* secara otomatis. Metode yang digunakan adalah *Support Vector Machine* (SVM). Data komentar yang digunakan pada penelitian ini dikumpulkan dari komentar-komentar pada foto atau video yang dibagikan oleh *public figure* atau artis yang memiliki pengikut (*followers*) diatas 1 juta pengikut. Dari hasil penelitian diketahui bahwa metode SVM dengan *kernel linier* memberikan hasil yang terbaik dengan error terkecil yaitu 2.8% dan ketepatan klasifikasi sebesar 97.33%.

Kemudian pada penelitian oleh Antonius Rachmat C & Yuan Lukito pada tahun (2017). Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes. Penelitian ini mengambil komentar *Instagram* dan membangun dataset dari public figur Indonesia yang memiliki lebih dari satu juta pengikut. Dengan menggunakan preprocessing (*tokenization, stop word removal, dan stemming*), pembobotan TF-IDF, dan pembelajaran yang diawasi. Metode *Naive Bayes* digunakan untuk mendeteksi komentar spam dalam Bahasa Indonesia. *Naive Bayes* menghasilkan tingkat akurasi 74,31% pada dataset yang tidak seimbang dan tingkat akurasi 77,25% data seimbang. Hasil ini menunjukkan bahwa Naïve Bayes bisa digunakan untuk membangun *dektektor* otomatis komentar spam dalam bahasa Indonesia di *Instagram* dengan tingkat akurasi tinggi.

Pada penelitian Rafly Ghazali Ramli & Yuliant Sibaroni (2022). Klasifikasi Topik X menggunakan Metode *Random Forest* dan Fitur Ekspansi *Word2Vec*. Pengguna social media X biasanya hanya tertarik pada post yang termasuk dalam jenis topik tertentu. Post hanya memuat 280 karakter, membuat klasifikasi post menjadi banyak tantangan, karena post yang pendek dan kurang fokus pada topik. Solusi untuk menyelesaikan tantangan tersebut dalam penelitian ini menggunakan fitur ekspansi agar memperkaya teks sehingga tampak seperti dokumen teks berukuran besar. Metode yang dipilih pada fitur ekspansi adalah *Word2Vec*, untuk mengelompokkan vektor dari kata-kata yang mirip menjadi satu di dalam ruang vektor, artinya mendeteksi kemiripan secara matematis. Penulis menggunakan metode *Random Forest* untuk klasifikasi data post pada penelitian ini, karena terkenal menjaga ketidakseimbangan data di kelas yang berbeda, terutama kumpulan data yang sangat besar. Hasil dari nilai akurasi dan F1-Score pada fitur ekspansi menggunakan algoritma Random Forest menunjukkan bahwa hasil dari keseluruhan nilai mengalami peningkatan sehingga nilai tertingginya ada pada fitur top 5 dengan menggunakan kamus kata data berita + tweet sebesar 99,49%.

Pada penelitian John Cardiff & Elena Shushkevich pada tahun (2018). *Misogyny Detection and Classification in English Tweets: The Experience of the ITT Team*. Misogini adalah salah satu bentuk diskriminasi terhadap wanita. Pernyataan misogini adalah pernyataan yang diungkapkan sebagai ekspresi atau prasangka kebencian terhadap perempuan, baik disampaikan dari laki-laki maupun sesama perempuan, dan baik secara verbal ataupun tidak, dapat diwujudkan secara linguistik dalam berbagai

cara. Melakukan klasifikasi pernyataan misogini menjadi dua kelas menggunakan TF-IDF sebagai pembobotan kata, dilanjutkan dengan metode *Logistic Regression*, *Support Vector Machines*, dan *Naive Bayes* dan *Logistic Regression* (NB+LR) yang menghasilkan akurasi tertinggi yaitu 78% dan 76%.

Lalu pada penelitian oleh Muhammad Hanafiah, et al., (2019). Klasifikasi Spam Post Pada X Menggunakan Metode Naïve Bayes (Studi Kasus: Pemilihan Presiden 2019). Penelitian ini mengklasifikan sebuah post ke dalam dua kelas yaitu *spam* dan *non-spam* dengan studi kasus pilpres 2019. Penelitian ini menggunakan Naïve Bayes dengan preprocessing dan Naïve Bayes tanpa preprocessing, masing-masing menghasilkan nilai akurasi 76,34% dan 74,14%.

Tabel 2. 1 Penelitian Terdahulu

No	Penulis	Judul	Tahun	Keterangan
1	Andita Wahyuningtyas., et al	Deteksi Spam pada Twitter Menggunakan Algoritme Naïve Bayes	2020	Penelitian ini bertujuan untuk mendeteksi post spam dan bukan spam. Penelitian ini menggunakan 70% data latih dan 30% data uji dengan metode klasifikasi <i>Naïve Bayes</i> . Akurasi hasil klasifikasi post spam dan bukan spam adalah 95.57%.
2	Agus Tiyansyah Syah., et al	Klasifikasi Komentar Spam Pada Instagram Menggunakan Metode Support Vector Machine	2020	Tujuan dari penelitian ini adalah membangun sebuah system yang dapat mengklasifikasi bahwa suatu komentar adalah <i>spam</i> atau <i>nonspam</i> secara otomatis. Data komentar yang digunakan pada penelitian ini dikumpulkan dari komentar-komentar pada foto atau video yang dibagikan o-leh <i>public figure</i> atau artis yang memiliki pengikut (<i>followers</i>) diatas 1 juta pengikut. Dari hasil penelitian diketahui bahwa metode SVM dengan <i>kernel linier</i> memberikan hasil yang terbaik dengan error terkecil yaitu 2.8% dan ketepatan klasifikasi sebesar 97.33%.

3	Antonius Rachmat C & Yuan Lukito	Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes	2017	Penelitian ini mengambil komentar <i>Instagram</i> dan membangun dataset dari public figur Indonesia yang memiliki lebih dari satu juta pengikut. Metode <i>Naive Bayes</i> digunakan untuk mendeteksi komentar spam dalam Bahasa Indonesia. <i>Naive Bayes</i> menghasilkan tingkat akurasi 74,31% pada dataset yang tidak seimbang dan tingkat akurasi 77,25% data seimbang. Hasil ini menunjukkan bahwa Naïve Bayes bisa digunakan untuk membangun <i>dektektor</i> otomatis komentar spam dalam bahasa Indonesia di <i>Instagram</i> dengan tingkat akurasi tinggi.
4	Rafly Ghazali Ramli & Yuliant Sibaroni	Klasifikasi Topik Twitter menggunakan Metode Random Forest dan Fitur Ekspansi Word2Vec	2022	Penelitian ini melakukan klasifikasi post pada X menggunakan metode <i>Random Forest</i> untuk klasifikasi data post dan menggunakan fitur ekspansi <i>Word2Vec</i> untuk mengelompokkan vektor dari kata-kata yang mirip menjadi satu di dalam ruang vektor, artinya mendeteksi kemiripan secara matematis.
5	John Cardiff & Elena Shushkevich	Misogyny Detection and Classification in English Tweets: The Experience of the ITT Team	2018	Melakukan klasifikasi pernyataan misogini menjadi dua kelas menggunakan TF-IDF sebagai pembobotan kata, dilanjutkan dengan metode Logistic Regression, Support Vector Machines, dan Naive Bayes dan Logistic Regression (NB+LR) yang menghasilkan akurasi tertinggi yaitu 78% dan 76%.
6	Muhammad Hanafiah, et al	Klasifikasi Spam Tweet Pada Twitter Menggunakan Metode Naïve	2019	Penelitian ini mengklasifikan sebuah post ke dalam dua kelas yaitu <i>spam</i> dan <i>non-spam</i> dengan studi kasus pilpres 2019. Penelitian ini menggunakan Naïve Bayes dengan preprocessing dan Naïve Bayes tanpa

		Bayes (Studi Kasus: Pemilihan Presiden 2019)		preprocessing, masing-masing menghasilkan nilai akurasi 76,34% dan 74,14%.
--	--	--	--	--

2.9 Perbedaan Penelitian

Penelitian ini memiliki perbedaan dibandingkan dengan penelitian terdahulu dalam beberapa aspek. Metode yang diterapkan pada penelitian ini adalah Random Forest Classifier sedangkan pada penelitian sebelumnya, metode yang digunakan dalam mendeteksi post spam berbeda dengan penelitian ini. Terlebih lagi, penelitian ini memanfaatkan word embedding, yaitu TF-IDF, untuk mengubah teks menjadi representasi vector. Pada penelitian ini juga melakukan deteksi pada post dan repost secara umum yang menggunakan bahasa Indonesia.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Bab ini akan membahas tentang analisis serta desain rancangan sistem untuk identifikasi akun palsu dan akun asli pada instagram. Bagian pertama bab ini, terdapat penjelasan mengenai teknik pengambilan data serta analisa data pada dataset yang digunakan dalam proses penelitian. Bagian selanjutnya pada bab ini, terdapat penjelasan alur dan struktur sistem yang akan dijelaskan dalam bentuk arsitektur umum dan flowchart. Pada bagian berikutnya, terdapat penjelasan mengenai algoritma yang digunakan dalam proses pembangunan system. Bagian akhir bab ini, terdapat bagian rancangan desain user interface yang menjelaskan dan menampilkan secara singkat tentang desain user interface pada program.

3.1 Data yang Digunakan

Penelitian ini menggunakan data yang berisikan post dan repost dalam bahasa Indonesia. Data tersebut akan disimpan dalam file berformat .csv yang berisi 2800 data post dan repost yang dimana post dan repost tersebut berasal dari hasil *scraping* data dari postingan akun di media sosial X. Data diperoleh dari akun-akun pengguna berikut: @MoniccaFU, Jackie_22, Syukron, Ilyshan98, uSer_23910, dan xdd3319_.

Dalam data yang dipakai pada penelitian ini terdapat tweet dan label yaitu untuk menentukan post tersebut mengandung unsur spam maupun tidak mengandung unsur spam. Pada proses *scraping* data, penulis memakai beberapa keyword untuk melakukan pelabelan spam dan bukan spam berdasarkan penelitian terdahulu yang dilakukan Septiandri & Wibisono (2017). Selain itu, dalam penelitian ini juga dibantu oleh seorang guru bahasa Indonesia bernama Sorta Pardede, S.Pd. untuk melakukan pelabelan pada data yang ada. Terdapat beberapa karakteristik yang digunakan untuk melakukan pelabelan manual spammer, karakteristik tersebut adalah sebagai berikut (Yang, dkk., 2011): (1) Spam yang berisi link aktif. Hal ini dilakukan untuk mempromosikan sebuah website dengan cara menautkan link aktif berupa URL. (2) Spam yang berisi promosi atau menawarkan produk tertentu. Spam ini masih berkaitan dengan jenis yang pertama, yaitu berisi link aktif yang menawarkan promosi produk

tertentu. (3) Kesamaan tweet dengan tweet sebelumnya. Hal ini dilihat berdasarkan kumpulan tweet yang telah di-posting oleh pengguna Twitter, jika tiap tweet memiliki kesamaan konten atau kemiripan kemunculan kata yang digunakan maka akun tersebut dapat dikategorikan sebagai bot spammer. (4) Spam seringkali memakai banyak hashtag. Hashtag memudahkan pencarian tweet, atau memperbesar peluang untuk menjadi trending topic (Verma & Sofat, 2014).

Di dalam data yang di pakai dalam penelitian ini terdapat 2 kolom utama, yaitu kolom post/repost dan kolom label skor. Label skor digunakan untuk menentukan apakah post/repost tersebut mengandung unsur spam maupun tidak mengandung unsur spam. Untuk setiap post/repost yang mengandung unsur *spam* akan diberi label 1 dan yang tidak mengandung unsur spam (*non-spam*) akan diberi label 0. Berikut dataset post dan repost spam dan non-spam secara lebih detail, dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Dataset Post dan Repost Twitter

No	Tweet	Label Score
1	uSer_23910,"Hai Kak, @Hasegawahutaru, Follow yuk @InfoMakassarID paling update seputar Makassar, menarik dan menambah wawasan :), pasti di Folback!",1	0
2	Ilyshan98,Selamat #HariKebangkitanNasional Ã°Ã°,â€¡Ã°Ã°,â€¡Ã°Ã°,â€¡Ã°Ã°,â€¡Ã°Ã°,0	0
3	Syukron,korupsi Puskesmas harus diusut. https://t.co/yRSsRoRUZw #SumutWaspada #recekantwitter #GanjarTakTakutPakDirmanÃ¢â€,Ã¢â€! https://t.co/xNdfcnlDr8,0	1
4	Ilyshan98,"RT @slankdotcom: ""Buat sesuatu yang baru, ambil semua resiko! Inilah saat yang tepat, mimpikan jauh terdepan"" ~ Indonesia Now	0
5	MoniccaFU,"""Bangkitkan rasa dan semangat persatuan, kesatuan. Maju terus Indonesia nan jaya. Terus berkarya membangun bangsa.Ã¢â€,Ã¢â€! https://t.co/IJiKNbvrNI "	0
...
27 97	Jackie_22,"Caranya gampang, yuk kita sama sama lawan info HOAX! #NasionalismeZamanNow",0	0

27 98	MoniccaFU,"Hari kebangkitan Nasional	0
27 99	Bersatu kita padu	0
28 00	xdd3319_,FREE ONGKIR Cream Pemutih Wajah Pria AURABEAUTY Complete Whitening Series https://t.co/AgiF47Jzkt https://t.co/J0G6zHvblf,1	1

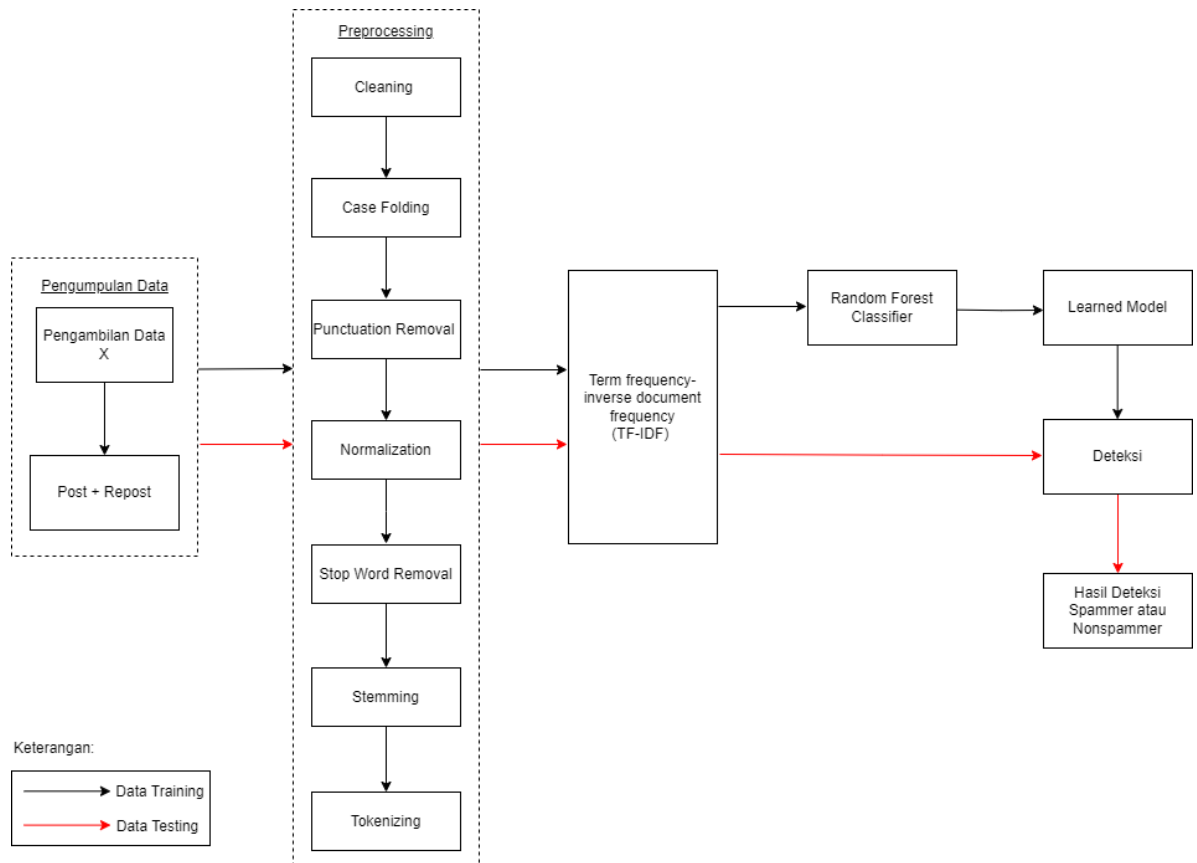
Dataset pada tabel 3.1 akan dibagi menjadi 2 bagian yaitu data pelatihan (training) dan data uji (testing). Dataset akan dibagi sebesar 80% data training dan 20% data testing. Selanjutnya data training akan dipecah lagi menjadi dua, yakni data pelatihan utama dan data validasi. Dimana 80% merupakan data latih dan 20% adalah data validasi. Pembagian jumlah dataset tersebut dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Pembagian Dataset

Data Training	Data Testing
2240	560

3.2 Arsitektur Umum

Dalam penelitian ini penulis akan melakukan beberapa tahapan antara lain: pengumpulan data, data yang dikumpulkan akan disimpan dalam format .csv (*comma seprated values*) dan akan digunakan sebagai input dalam perancangan sistem. Data yang sudah diperoleh, akan dibagi menjadi data training dan data testing. Selanjutnya, data yang telah dikumpulkan akan melalui tahapan preprocessing yang terdiri atas *cleaning case folding, punctuation removal, normalization, stopword removal, stemming* dan *tokenizing*. Setelah data tersebut melalui tahapan preprocessing, data tersebut akan melalui tahapan word embedding yang bertujuan untuk mengubah kata menjadi sebuah vector atau array yang terdiri atas kumpulan angka. Pada penelitian ini penulis menggunakan TF-IDF. Selanjutnya data tersebut akan melalui proses training menggunakan Random Forest yang akan menghasilkan suatu model yang nantinya akan dipakai pada proses testing. Output dari sistem pada penelitian ini adalah spammer dan bukan spammer. Berikut penjelasan tentang arsitektur umum dilihat pada Gambar 3.1.



Gambar 3. 1 Arsitektur Umum

3.3 Pre-processing

Setelah dilakukan pengumpulan data, langkah selanjutnya adalah *pre-processing* data. Tahap ini dilakukan untuk mempersiapkan teks menjadi data yang dapat diolah pada tahap-tahap berikutnya. Hasil *pre-processing* mendapatkan teks yang bersih dari *noise*. Tahap preprocessing ini dilakukan karena data post/repost yang tidak baku serta masih terdapat *noise* pada data teks tersebut. Proses *pre-processing* juga dilakukan untuk mendapatkan parameter-parameter yang diinginkan pada penelitian. Berikut ada tahap *pre-processing* yang dilakukan pada penelitian deteksi post dan repost spam pada media sosial X.

3.3.1 Cleaning

Cleaning merupakan suatu tahap yang dilakukan untuk menghapus beberapa variable yang tidak digunakan pada proses penelitian sehingga nantinya dapat meminimalisir tingkat kegagalan pada proses identifikasi di dalam penelitian. Variabel tersebut misalnya seperti URL, *emoticon*, angka, dan lainnya yang dianggap tidak digunakan

dalam data. Penerapan proses cleaning dapat dilihat pada *pseudocode* berikut ini dan hasilnya dapat dilihat pada Tabel 3.3.

```
data['Preprocess'] = data['Tweet'].str.replace("&am", " ")
data['Preprocess'] = data['Preprocess'].str.replace(">", " ")
data['Preprocess'] = data['Preprocess'].str.replace("\\\\n", " ")
data['Preprocess'] = data['Preprocess'].astype(str).apply(lambda x:
    x.encode('ascii', 'ignore').decode('ascii'))
data['Preprocess'] = data['Preprocess'].str.replace(" ", " ")
data['Preprocess'] = data['Preprocess'].str.replace("'", '')
data['Preprocess'] = data['Preprocess'].str.replace('@[\w]+', '')
data['Preprocess'] = data['Preprocess'].str.replace("\\\\n", " ")
data['Preprocess'] = data['Preprocess'].str.replace("\n", " ")
data['Preprocess'] = data['Preprocess'].str.replace("\r", " ")
data['Preprocess'] = data['Preprocess'].str.replace(r'((?i)\b(
    (?:(https|http?://|www\d{0,3}[.])|[a-z0-9.-]+[.])|[a-zA-z]{2,4}/)
    (?:(^s(<>)+|((^s(<>)+|((^s(<>)+\)))*)\))+?:\(((^s(<>)+\)))
    <>+|((^s(<>)+\)))*)\|^s!()\[\]\{\};:'.', '<>«»“”‘’')',
    "", regex=True)
data['Preprocess'] = data['Preprocess'].str.replace("'", '')
data['Preprocess'] = data['Preprocess'].str.replace("\\\\x[a-zA-z0-9]
[a-zA-z0-9]", "", regex=True)
```

Tabel 3. 3 Penerapan Tahap *Cleaning*

Sebelum tahap cleaning	Sesudah tahap cleaning
uSer_23910,"Hai Kak, @Hasegawahutaru, Follow yuk @InfoMakassarID paling update seputar Makassar, menarik dan menambah wawasan :), pasti di Folback!",	Hai Kak, @Hasegawahutaru, Follow yuk @InfoMakassarID paling update seputar Makassar, menarik dan menambah wawasan, pasti di Folback

3.3.2 Case Folding

Pada tahapan case folding dilakukan penyeragaman huruf dengan cara mengubah huruf besar (*uppercase*) pada post/repost menjadi huruf kecil (*lowercase*). Proses ini bertujuan untuk membuat karakter menjadi lebih sederhana, sehingga karakter yang tidak diinginkan menjadi lebih mudah untuk dihapus karena huruf kecil dan huruf besar memiliki bentuk yang berbeda pada text processing. Contoh penerapan case folding dapat dilihat pada *pseudocode* berikut dan hasilnya terdapat pada Tabel 3.4.

```
data['Preprocess'] = data['Preprocess'].str.lower()
```

Tabel 3. 4 Penerapan Tahap *Case Folding*

Sebelum tahap case folding	Sesudah tahap case folding
Hai Kak, @Hasegawahutaru, Follow yuk @InfoMakassarID paling update seputar Makassar, menarik dan menambah wawasan, pasti di Folback	hai kak, @hasegawahutaru, follow yuk @infomakassarid paling update seputar makassar, menarik dan menambah wawasan, pasti di folback

3.3.3 Punctuation Removal

Tahapan ini adalah proses menghapus tanda baca ataupun simbol yang ada pada kalimat untuk membersihkan data yang akan diproses sehingga karakter yang tidak diinginkan tidak akan ikut dalam proses. Hal ini akan membuat data menjadi lebih sederhana. Penerapan *punctuation removal* dapat dilihat pada *pseudocode* berikut dan hasilnya pada Tabel 3.5.

```
data['Preprocess'] = data['review'].str.replace('[^a-zA-Z0-9]+',
        ' ')
data['Preprocess'] = data['Preprocess'].str.replace('[^a-zA-Z]+',
        ' ')
```

Tabel 3. 5 Penerapan Tahap *Punctuation Removal*

Sebelum tahap punctuation removal	Sesudah tahap punctuation removal
hai kak, @hasegawahutaru, follow yuk @infomakassarid paling update seputar makassar, menarik dan menambah wawasan, pasti di folback	hai kak hasegawahutaru follow yuk infomakassarid paling update seputar makassar menarik dan menambah wawasan pasti di folback

3.4.4 Normalization

Proses ini memiliki tujuan untuk menormalisasi setiap kata yang terdapat dalam data. Setiap kata dalam data yang merupakan singkatan dan *typo* akan diperbaiki serta dinormalisasi menjadi kata yang terstruktur sesuai KBBI sehingga dapat diproses. Proses akan menjadi lebih mudah apabila struktur kalimat lebih spesifik. Pada penelitian ini penulis menggunakan *dictionary* yang sudah disediakan sebelumnya. Proses ini dilakukan dengan cara menormalisasikan kata-kata singkatan seperti “abis” menjadi “habis”, “ad” menjadi “ada”, “adlh” menjadi “adalah”, “ae” menjadi “saja”, “bbm” menjadi “bahan bakar minyak” dan kata-kata *typo* lainnya yang tidak sesuai

dalam Bahasa Indonesia. Tetapi *dictionary* yang dipakai dalam proses *normalization* ini memiliki kekurangan yaitu terdapat beberapa kata yang tidak ternormalisasi dengan baik akibat singkatan dan *typo* yang berlebihan. Penerapan *normalization* dilihat pada *pseudocode* dan Tabel 3.6.

```
function normalize_text(text, stdword_, nonstdword_):
    text = text.split(" ")
    for i in range(length(text)):
        if text[i] in nonstdword_:
            index = index(nonstdword_, text[i])
            text[i] = stdword_[index]
    return join(" ", map(string, text))
```

Tidak Baku	Baku
Abis	Habis
Abg	Abang
dng	Dengan
yuk	Ayo

Tabel 3. 6 Penerapan Tahap *Normalization*

Sebelum tahap normalization	Sesudah tahap normalization
hai kak hasegawahutaru follow yuk infomakassarid paling update seputar makassar menarik dan menambah wawasan pasti di folback	hai kak hasegawahutaru follow ayo infomakassarid paling update seputar makassar menarik dan menambah wawasan pasti di folback

3.4.5 Stopword Removal

Stop word removal merupakan tahap yang dilakukan untuk menghilangkan kata-kata umum atau yang tidak dibutuhkan dalam penelitian, biasanya dapat diabaikan tetapi tidak mengubah arti kalimat. Tahap ini dilakukan dengan tujuan untuk membuat kalimat menjadi lebih sederhana, agar dapat mempercepat proses training tanpa mengubah arti dari kalimat tersebut. Kata hubung yang terdapat dalam data tidak akan mempengaruhi arti dari kalimat yang ada. Kata hubung dalam Bahasa Indonesia hanya memperjelas konteks yang hanya dipahami oleh manusia, tetapi mesin tidak dapat memahaminya. Proses stop word removal dilakukan melalui pengecekan hasil parsing deskripsi. Jika kata-kata tersebut termasuk tidak penting (*stoplist*) maka akan di *remove*. Contohnya

“dan”, “atau”, “dan lain-lain”, “ke”. *Library* yang digunakan pada tahapan ini adalah NLTK (*Netural Language Tool Kit*). Daftar *stopword* yang digunakan dimuat dalam Tabel 3.7 serta contoh penerapan proses *stopword removal* ini dapat dilihat pada *pseudocode* berikut dan hasilnya pada Tabel 3.8.

```
stop_words = list(stopwords.words('indonesian'))
stop_words = stop_words + ["rt", "retweet", "url", "user"]

data['Preprocess'] = data['review']

for stop_word in stop_words:
    regex_stopword = r"\b" + stop_word + r"\b"
    data['Preprocess'] = data['Preprocess'].str.replace(
        regex_stopword, '')
```

Tabel 3. 7 List Stopword Bahasa Indonesia

List Stopword
aduh, dan, di, alah, alamak, cih, deh, doing, dong, gih, haha, hai, halo, hehe, hihi, kah, kan, ke, kok, lah, loh, mah, nah, nya, oh, pun, sih, sip, tuh, uhuk, uhuy, ups, waduh, wah, wkwk, woi, wow, ya, yaelah, yang, yuhu, ,,tiktok, terima, kasih, up, no, baru, tagih, sms, by, cok, kartu, ok

Tabel 3. 8 Penerapan Tahap *Stopword Removal*

Sebelum tahap stopwords removal	Sesudah tahap stopwords removal
hai kak hasegawahutaru follow ayo infomakassarid paling update seputar makassar, menarik dan menambah wawasan, pasti di folback	kak hasegawahutaru follow ayo infomakassarid paling update seputar makassar menarik menambah wawasan pasti folback

3.4.6 Stemming

Stemming digunakan untuk merubah kata-kata menjadi bentuk dasar, mengurangi kata-kata ke yang sama bentuk dasarnya dengan cara mengurangi imbuhan, sehingga kata-kata tersebut dapat dianggap sama dalam analisis teks. Adapun contoh imbuhan yang akan dihilangkan antara lain Inflection Suffixes (“-nya”, ”ku”, ”mu”, ”-kah”), imbuhan awalan atau prefix (“be”, ”ke”, ”di”), dan imbuhan turunan (“kan”, ”-i”, ”-an”). Setiap penggunaan imbuhan yang terdapat dalam data akan dihapus untuk mendapatkan variasi kata dasar yang akurat tanpa harus menghilangkan makna pada kata tersebut.

Contoh, kata "berjalan", "berjalanlah", dan "berjalanlah" dapat di-stem menjadi "jalan". Proses *stemming* ini digunakan *library* Sastrawi. Contoh penerapan proses stemming dilihat pada pseudocode berikut dan hasilnya pada Tabel 3.9.

```
factory = StemmerFactory()
stemmerID = factory.create_stemmer()

function stemming(text, stemmer_id):
    stemmed = stemmer_id.stem(text)
    return stemmed

data['Preprocess'] = data['review']

data['Preprocess'] = data['Preprocess'].map(lambda com:
    stemming(com, stemmerID))
```

Tabel 3. 9 Penerapan Tahap *Stemming*

Sebelum tahap stemming	Sesudah tahap stemming
kak hasegawahutaru follow ayo infomakassarid paling update seputar makassar menarik menambah wawasan pasti folback	kakak hasegawahutaru follow ayo infomakassarid baru putar makassar tarik tambah wawas folback

3.4.7 *Tokenizing*

Proses yang dilakukan untuk memisahkan kalimat menjadi token (kata per kata). Tokenisasi merupakan proses penting dalam pengolahan data yang bertujuan untuk membagi data menjadi bagian-bagian kecil yang disebut token. Pada tahap ini, kalimat-kalimat dalam data akan dipisahkan menjadi kata-kata yang merupakan unit terkecil dalam data. Adapun tujuan dari tokenisasi adalah mengubah teks yang telah dibersihkan menjadi format yang dapat dimengerti dan diolah oleh mesin. Token dapat berupa kata, frasa ataupun karakter yang terpisah. Contoh penerapan *tokenizing* dapat dilihat pada pseudocode berikut dan hasilnya pada Tabel 3.10.

```
function word_tokenize(text):
    return tokenize_text_into_words(text)

data['Preprocess_tokenized'] = data['review'].apply(word_tokenize)

return data
```

Tabel 3. 10 Penerapan *Tokenizing*

Sebelum tahap tokenizing	Sesudah tahap tokenizing
kak hasegawahutaru follow ayo infomakassarid paling update seputar makassar menarik menambah wawasan pasti folback	“kakak”, “hasegawahutaru”, “follow”, “ayo”, “infomakassarid”, “baru”, “putar”, “makassar”, “tarik”, “tambah”, “wawas”, “folback”

3.5 Word Embedding

Word embedding merupakan sebuah metode untuk mengkonversi serangkaian kata yang berupa karakter alphanumeric ke dalam bentuk vektor atau *array* yang berisi angka atau bilangan. Penelitian ini menggunakan TF-IDF sebagai *word embedding*.

Term frequency-inverse document frequency (TF-IDF) merupakan teknik pembobotan berbasis statistik dengan menggabungkan dua konsep dalam perhitungannya, yaitu frekuensi kemunculan kata dan *inverse*. Pembobotan TF-IDF sering diterapkan pada permasalahan penggalian informasi. Ide dasar TF-IDF adalah memberikan bobot pada setiap kalimat, selanjutnya kalimat tersebut diurutkan berdasarkan bobot teratas dengan bobot paling besar akan dipilih sebagai hasil. Bobot kalimat diperoleh dari penjumlahan bobot term pada sebuah kalimat.

Pada penelitian ini, setiap kata dalam korpus data akan diubah menjadi vector dengan dimensi 500. Vector ini dikenal sebagai embedding matrix yang berfungsi sebagai representasi numeric untuk tiap-tiap kata. Proses ini akan memberikan pemahaman komputer terhadap makna dan hubungan antar kata-kata dalam teks. Selanjutnya, embedding matrix ini akan menjadi bobot dalam pemodelan data. Pada gambar 3.2 diberi contoh vektor atau bobot kata dari kata “ayo”.

```
array([[0.43590711 0.         0.         0.         0.         0.
        0.         0.         0.         0.         0.         0.
        0.         0.46756168 0.         0.         0.         0.
        0.         0.         0.         0.         0.         0.
        0.46756168 0.         0.         0.         0.         0.
        0.62937424 0.         0.         0.         0.         0.
        0.         0.         0.         0.46501068 0.         0.
        0.         0.46756168 0.         0.         0.         0.
        0.         0.         1.         0.         1.         0.
        0.         0.         0.         1.         0.         0.
        0.         0.         0.         0.         0.         0.
        0.         0.         0.         0.         0.         0.
        0.         0.         0.         0.         0.         0.
        0.         0.         0.         0.         0.46756168 0.
        0.         0.         0.         0.         0.76957964 0.
        0.         0.         0.         0.         0.         ]]
```

Gambar 3. 2 Embedding Matrix “ayo”

3.6 Random Forest Classifier

Metode Random Forest merupakan algoritma *Ensemble Learning* yang menggunakan dan membangun struktur *Tree* dalam tahapannya. Dalam penggunaannya, dibangun *Decision Tree* dengan memilih atau mengambil data secara acak. Untuk menentukan kelas suatu data, dalam Random Forest menggunakan sistem voting dari hasil berdasarkan *Decision Tree* tersebut. Random Forest (RF) merupakan metode yang dapat meningkatkan hasil akurasi, karena dalam mengembangkan simpul untuk setiap node yang dilakukan secara acak. Metode ini digunakan untuk membangun *Decision Tree* yang terdiri dari *root node*, *internal node*, dan *leaf node* dengan mengambil atribut dan data secara acak sesuai ketentuan yang diberlakukan. Pada penelitian ini, penulis menggunakan algoritma Random Forest Classifier dengan menginstall *library Scikit-learn*. *Scikit-learn* menyediakan variabel tambahan dengan model, yang menunjukkan kepentingan relatif atau kontribusi setiap fitur dalam prediksi dan secara otomatis menghitung skor relevansi setiap fitur dalam *training*.

```
import sklearn
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
clf = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
clf.fit(X_train, Y_train)
```

Untuk visualisasi, digunakan kombinasi antara library *matplotlib* dan *seaborn*, karena *seaborn* dibangun di atas *matplotlib*, library tersebut menawarkan sejumlah tema yang disesuaikan dan menyediakan jenis plot tambahan. *Matplotlib* adalah *superset* dari *seaborn*. Kedua library tersebut menghasilkan visualisasi yang baik.

Sebelum dilakukan proses klasifikasi sistem, akan dilakukan penggantian rasio data latih dan data uji terlebih dahulu agar hasil yang diberikan lebih optimal. Pada sistem klasifikasi dilakukan pengulangan eksekusi program sebanyak 3 kali yang diambil nilai rata-rata akurasi nya dan menggunakan data uji dan data latih yang diperbandingkan dengan 20:80. Lalu, diambil paling tinggi akurasinya dari percobaan penggantian rasio pada data latih dan data uji.

3.7 Metode Evaluasi

Metode evaluasi dilakukan untuk mengetahui keakuratan dan performa sistem ketika melakukan deteksi akun platform media sosial X dengan menggunakan algoritma Random Forest. Performansi suatu model identifikasi dapat dievaluasi dengan perhitungan beragam cara seperti *accuracy*, *precision*, *recall*, dan *F1-Score*. *Accuracy* merupakan rasio prediksi yang sesuai dengan aktual baik yang kelasnya positif atau negatif. *Precision* merupakan rasio perbandingan antara kelas yang diprediksi benar positif dengan seluruh kelas hasil yang diprediksi positif. *Recall* adalah rasio perbandingan antara kelas yang diprediksi benar positif dengan seluruh data yang memiliki kelas aktual positif. *F1-Score* adalah rata-rata perhitungan *precision* dan *recall*. Penelitian ini menggunakan perhitungan *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk evaluasi sistem yang dibangun. Penerapan *confussion matrix* dapat dilihat pada Tabel 3.11.

Tabel 3. 11 Penerapan *Confussion Matrix*

Classification		Actual Values	
		Positif	Negatif
Predicted Values	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Keterangan:

TP (*True Positive*) : Jumlah banyaknya data yang diprediksi positif oleh model dan kelas aktualnya bernilai positif

TN (*True Negative*) : Jumlah banyaknya data yang diprediksi negatif oleh model dan kelas aktualnya bernilai negatif

FP (*False Positive*) : Jumlah banyaknya data yang diprediksi positif oleh model dan kelas aktualnya bernilai negatif

FN (*False Negative*) : Jumlah banyaknya data yang diprediksi negatif oleh model dan kelas aktualnya bernilai positif.

Perhitungan yang digunakan untuk melakukan evaluasi besaran nilai akurasi, *precision*, *recall*, dan *f1-score* dijabarkan dalam persamaan seperti di bawah ini:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \dots \dots \dots (3.1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \dots \dots \dots (3.2)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \dots \dots \dots (3.3)$$

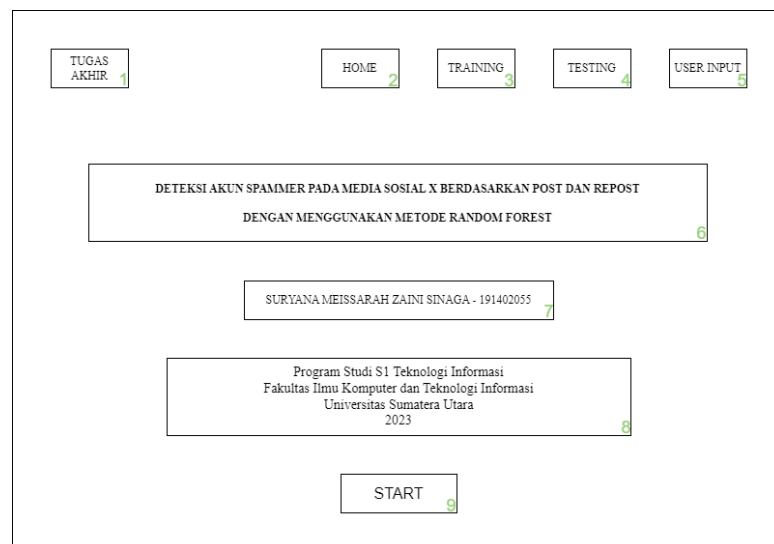
$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Data} \times 100\% \dots \dots \dots (3.4)$$

3.8 Perancangan Sistem

Tahapan perancangan sistem menjelaskan tentang perancangan menu dan alur sistem serta mempersiapkan antarmuka aplikasi deteksi spam pada media social X berdasarkan post dan repost dengan menggunakan algoritma *Random Forest Classifier*. Adapun antarmuka sistem yang dibangun pada penelitian ini berbasis *single page website*. Perancangan antarmuka sistem ini dibuat untuk memudahkan pengguna untuk mengoperasikan sistem.

3.8.4 Rancangan Tampilan Beranda

Tampilan beranda merupakan halaman utama ataupun tampilan yang pertama kali muncul saat sistem dijalankan. Pada tampilan beranda akan ditampilkan informasi tentang judul penelitian, identitas penulis, dan beberapa *button* yang dapat terhubung ke halaman lain. Rancangan tampilan beranda ditunjukkan pada Gambar 3.3 berikut.



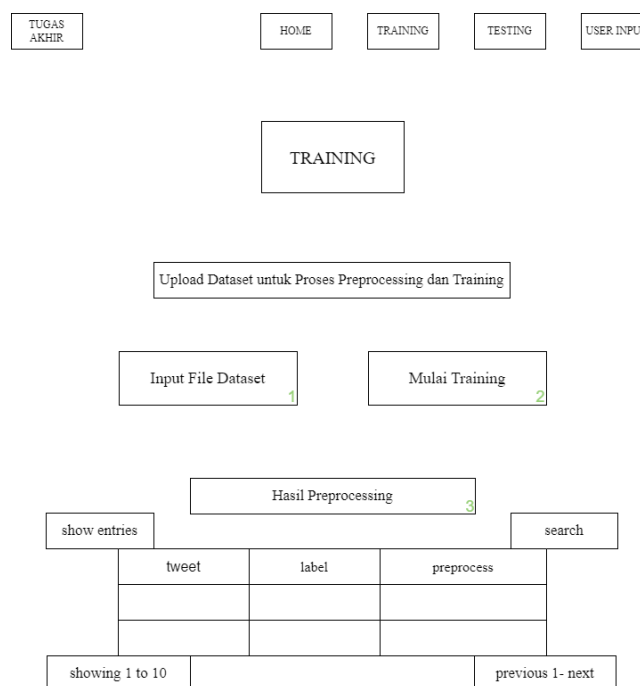
Gambar 3. 3 Rancangan tampilan beranda

Keterangan:

1. Label 1 merupakan *button* yang menampilkan informasi penggunaan tugas
2. Label 2 merupakan *button* yang mengarahkan pengguna ke halaman beranda
3. Label 3 merupakan *button* yang mengarahkan pengguna ke halaman training
4. Label 4 merupakan *button* yang mengarahkan pengguna ke halaman testing
5. Label 5 merupakan *button* yang mengarahkan pengguna ke halaman user input
6. Label 6 merupakan *button* yang menampilkan informasi judul penelitian
7. Label 7 merupakan *button* yang menampilkan identitas penulis
8. Label 8 merupakan *button* yang menampilkan informasi program studi dan fakultas
9. Label 9 merupakan *button* yang mengarahkan pengguna untuk memulai ke halaman *training*.

3.7.2 Rancangan Tampilan Training Data

Halaman *training* data memiliki peran untuk memungkinkan pengguna melakukan pelatihan dengan cara mengunggah file *dataset* yang sudah disiapkan sebelumnya dalam format .csv. Setelah file diunggah, pengguna dapat memilih tombol “Mulai Training” untuk mulai melakukan proses *training*. Kemudian proses *training* akan berlangsung dan akan mengolah data. Setelah semua proses *training* selesai dilaksanakan akan ditampilkan hasil data sebelum dan sesudah proses *preprocessing* beserta grafik laju akurasi dan *loss*. Desain dari tampilan halaman *training* ditunjukkan pada Gambar 3.4 berikut.



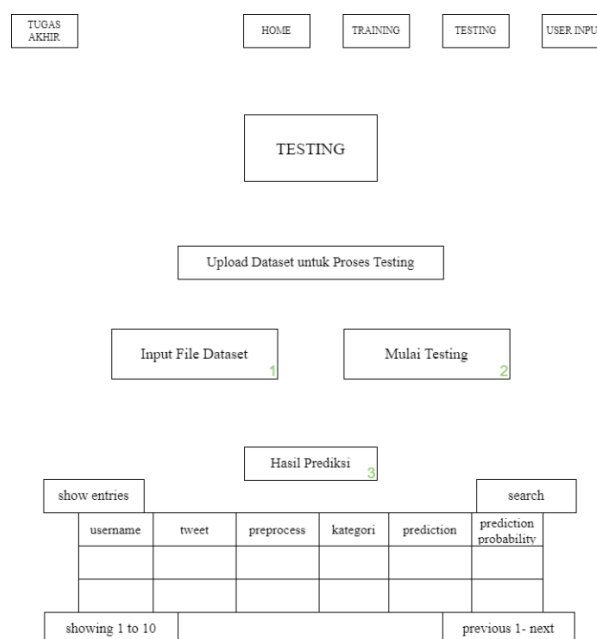
Gambar 3. 4 Rancangan tampilan *training data*

Keterangan:

1. Label 1 merupakan *button* untuk mengunggah file *dataset training*
2. Label 2 merupakan *button* untuk memulai proses *training*
3. Label 3 menampilkan hasil data *training*, sebelum dan sesudah proses *pre-processing*

3.7.3 Rancangan Tampilan Testing

Halaman *testing* memiliki peran bagi pengguna untuk menjalankan proses pengujian dengan cara mengunggah file *dataset* yang telah disediakan sebelumnya dalam format .csv. Setelah file diunggah, pengguna dapat memilih tombol “Mulai Testing” untuk mulai melakukan proses *testing*. Kemudian proses *testing* akan berlangsung dan akan menghasilkan informasi post yang mengandung spam dan post yang tidak mengandung spam berdasarkan model yang telah dilatih sebelumnya menggunakan algoritma Random Forest. Di halaman ini, hasil deteksi akan disajikan dalam bentuk tabel yang terdiri dari enam kolom, yakni *username*, *post/tweet*, *preprocess*, kategori, *prediction* dan *prediction probability*. Pada halaman ini juga menampilkan *Summary Probabilitas* Akun Spam dan Non Spam yang berisi 3 kolom, yaitu *username*, *average probability* dan kategori. Selain itu, pada halaman ini juga akan menampilkan *Confussion Matrix* yang berisi *Precision*, *Recall*, *F1-Score*, dan *Support*. Desain dari tampilan halaman *testing* dapat dilihat pada Gambar 3.5 berikut.



Gambar 3. 5 Rancangan Tampilan Testing Data

Keterangan:

1. Label 1 merupakan button untuk mengunggah file dataset training
2. Label 2 merupakan button untuk memulai proses testing
3. Label 3 menampilkan hasil data training, sebelum dan sesudah proses *testing*

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dibahas mengenai implementasi dari rancangan desain sistem yang telah disusun pada bab sebelumnya. Implementasi sistem melibatkan penggunaan perangkat lunak dan perangkat keras yang telah ditentukan seiring dengan penerapan pembuatan website deteksi. Selain itu, pada bab ini juga memfokuskan pembahasan mengenai pengujian sistem yang telah dikembangkan.

4.1 Implementasi Sistem

Dalam pengembangan sistem deteksi spam pada platform media sosial X berdasarkan post dan repost dengan menggunakan metode Random Forest Classifier, digunakan perangkat keras dan perangkat lunak sebagai elemen pendukung. Beberapa elemen tersebut mencakup:

4.1.1 Spesifikasi Perangkat Keras

Rincian mengenai spesifikasi perangkat keras yang diterapkan dalam proses pengembangan sistem ini dapat diuraikan sebagai berikut:

1. Processor AMD A9-9425 RADEON R5, 5 COMPUTE CORES 2C +3G 3.10GHz
2. Kapasitas memory (RAM) 4.00 GB.
3. HDD 1TB

4.1.2 Spesifikasi Perangkat Lunak

Dibawah ini rincian mengenai spesifikasi perangkat lunak yang diimplementasikan selama tahap pengembangan sistem:

1. Operating System: Windows 10 Home Single Language 64 bit
2. Bahasa pemrograman *Python* versi 3.8.6
3. Microsoft Visual Studio Code
4. Library bahasa pemrograman: *Flask* versi 2.1.2, *Gensim* versi 3.8.3, *Keras* versi 2.4.3, *matplotlib* versi 3.3.1, *nltk* versi 3.6.2, *numpy* versi 1.19.5, *pandas* versi

5. 1.1.5, *pickles* versi 0.1.1, *Sastrawi* versi 1.0.1, *scikit-learn* versi 0.24.2, *seaborn* versi 0.10.1, dan *BeautifulSoup*.

5.2 Implementasi Perancangan Tampilan Interface

Berikut adalah implementasi dari perancangan *interface* (antarmuka) pengguna yang telah dipaparkan dalam bab sebelumnya akan dijabarkan pada bagian ini.

4.2.1 Tampilan Halaman Beranda

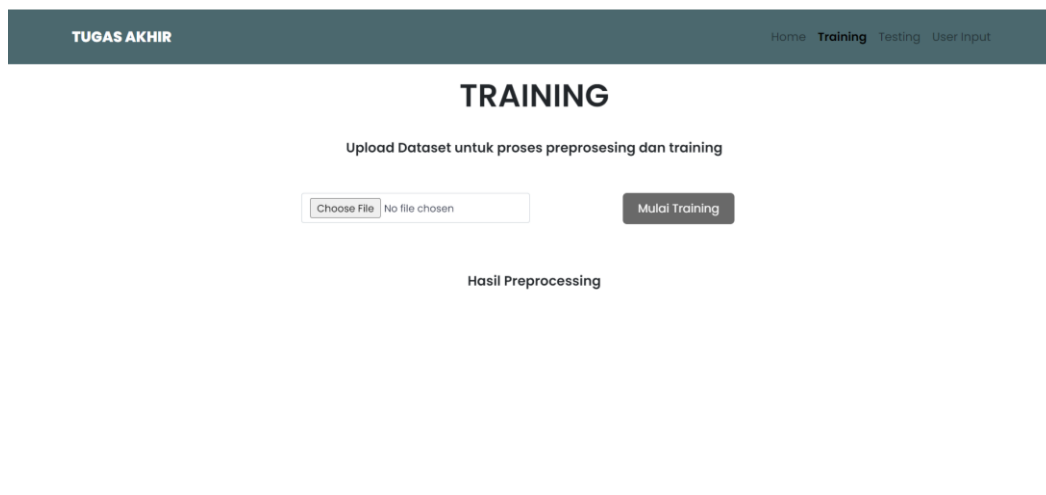
Berikut adalah representasi visual dari halaman utama yang telah direncanakan pada bab sebelumnya. Pada halaman ini terdapat judul penelitian dan rincian informasi mengenai peneliti. Pada halaman ini juga terdapat button home dan button yang dapat menghubungkan ke halaman berikutnya, yaitu untuk proses *training*, *testing* dan *user input*. Tampilan halaman beranda ditunjukkan pada Gambar 4.1



Gambar 4. 1 Tampilan halaman beranda

4.2.2 Tampilan Halaman Training

Berikut adalah antarmuka atau halaman yang digunakan untuk melaksanakan proses *training*. Agar pengguna dapat dengan mudah mengoperasikannya, tampilan desain antarmuka dibuat dengan sederhana. Pada halaman ini, pengguna hanya perlu mengunggah dataset yang telah disiapkan sebelumnya, dan kemudian menekan button “Mulai Training” yang disediakan untuk memulai proses preprocessing dan *training*. Hasilnya adalah pembentukan suatu model. Tampilan halaman sebelum memulai proses *training* ditunjukkan pada Gambar 4.2



Gambar 4. 2 Tampilan halaman *training* sebelum dilakukan proses training

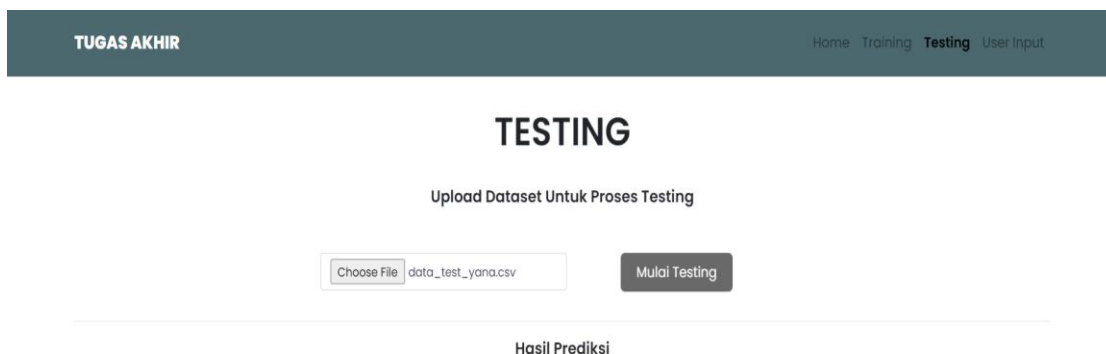
Setelah proses *training* data selesai dilakukan, maka pada halaman *training* akan menampilkan data hasil *training* dalam bentuk tabel yang terdiri dari 3 bagian tabel yaitu *tweet*, label *spam*, dan hasil *preprocessing*. Tabel ini berfungsi agar pengguna dapat membandingkan tweet yang belum melalui tahapan *preprocessing* dan yang sudah melalui tahapan *preprocessing*. Pada halaman *training* data terdapat juga fitur tambahan yaitu *search* dan *pagination* guna untuk memudahkan pengguna ketika ingin mencari kalimat atau kata kunci tertentu. Tampilan hasil *preprocessing* dapat dilihat pada Gambar 4.3.

TUGAS AKHIR			Home	Training	Testing
Hasil Preprocessing					
Show	10	entries	Search: <input type="text"/>		
Tweet	Label Spam	Preprocess			
"Bangkitkan rasa dan semangat persatuan, kesatuan. Maju terus Indonesia nan jaya. Terus berkarya membangun bangsa.â€¦ https://t.co/JlKNbvrNI	0	bangkit semangat satu satu maju indonesia nan jaya karya bangun bangsa			
"Bangsa yang besar adalah bangsa yang menghormati jasa pahlawannya" Selamat #harikebangkitannasional. Selalu bersatâ€¦ https://t.co/YC4ebJhdtu	0	bangsa bangsa hormat jasa pahlawan selamat harikebangkitannasional bersat			
"Bangunlah suatu dunia di mana semua bangsa hidup dalam damai dan persaudaraan." Selamat Hari Kebangkitan Nasionalâ€¦ https://t.co/CJLqThLZS0	0	bangun dunia bangsa hidup damai saudara selamat bangkit nasional			
"Buat sesuatu yang baru, ambil semua resiko! Inilah saat yang tepat, mimpikan jauh terdepan" - Indonesia Now Selamatâ€¦ https://t.co/DLyMQ2Kr0c	0	ambil risiko mimpi depan indonesia now selam			
"Harapan saya sebagai istri tentu mas Emil semakin sayang keluarga, keinginannya diijabah, tetap menjadi emil yangâ€¦ https://t.co/GbeFMp3fce	0	harap istri mas emil sayang keluarga ingin diijabah emil			

Gambar 4. 3 Tampilan hasil *training* data

4.2.3 Tampilan Halaman Testing

Berikut adalah antarmuka atau halaman yang diperuntukkan untuk menjalankan proses *testing*. Seperti halaman *training*, pada halaman ini, pengguna hanya perlu menyisipkan *dataset* yang telah disiapkan sebelumnya dan mengaktifkan tombol yang tersedia untuk memulai proses pengujian. Tata letak sebelum dimulainya pengujian dapat ditemukan pada Gambar 4.4, sebagaimana dijelaskan secara visual.



Gambar 4. 4 Tampilan halaman *testing* data

Halaman *testing* ini akan melakukan proses deteksi post yang termasuk spam dan bukan spam dengan menggunakan algoritma *Random Forest Classifier*. Model yang telah dilatih dan disimpan pada tahapan proses *training* sebelumnya akan melalui proses deteksi. Hasil dari pengujian akan ditampilkan dalam bentuk tabel dengan lima kolom, yakni *username*, *tweet*, *preprocessing*, *kategori*, *prediction* dan *prediction probability*. Tampilan hasil dari proses *testing* dapat dilihat pada Gambar 4.5.

TUGAS AKHIR

Home Training **Testing** User Input

Hasil Prediksi

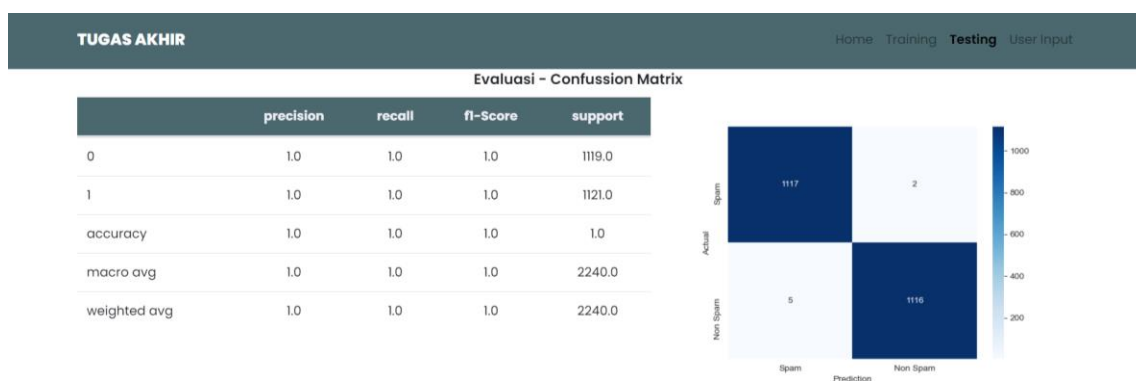
Show 10 entries

Search:

Username	Tweet	Preprocess	Kategori	Prediction	Prediction Probability
Ilyshan98	Kemerdekaan hanya didapat dan dimiliki oleh bangsa yg jiwanya berkobar-kobar dengak tekak Merdeka #HariKebangkitanNasional #PM1117	merdeka milik bangsa jiwa kobar kobar dengak tekak merdeka harikebangkitannasional gera mahasiswa islam indonesia	0	0	0.01
Ilyshan98	Oh ini #HariKebangkitanNasional kirain hari kebangkitan rasa : Eeeaaa	harikebangkitannasional kira bangkit eeeaaa	0	0	0.0
Ilyshan98	mari bersama - sama membangkitkan rasa dan semangat persatuan untuk Indonesia yng lebih baik .. Bangkitlah Indonesia! https://t.co/DiZ3jY4bes	mari bangkit semangat satu indonesia bangkit indonesia	0	0	0.0
Ilyshan98	Mendahulukan Indonesia, saling menguatkan sesama anak bangsa @ZUL_Hasan #HariKebangkitanNasional #ZulkifliHasan	dahulu indonesia kuat anak bangsa zul hasan harikebangkitannasional zulkiflihasan	0	0	0.02

Gambar 4. 5 Tampilan halaman hasil proses *testing*

Hasil model yang telah melalui proses *training* dan *testing*, selanjutnya akan melalui proses metode evaluasi yaitu menggunakan *Confussion Matrix*. Hasil yang akan ditampilkan yaitu berupa visualisasi gambar *heat-map* dan tabel agar pengguna mudah untuk membaca hasil evaluasi dalam deteksi spam. Tampilan hasil evaluasi ditunjukkan pada gambar 4.6.



Gambar 4. 6 Tampilan hasil evaluasi

4.2.4 Tampilan Halaman User Input

Halaman user input dirancang untuk memungkinkan pengguna melakukan deteksi post yang merupakan spam dan bukan spam tanpa perlu mengunggah dataset dalam format .csv. Pada halaman masukan pengguna ini, pengguna hanya diperbolehkan untuk memasukkan satu tweet pada *text box* yang telah disediakan. Tampilan dari halaman user input dapat dilihat pada Gambar 4.7.

The screenshot shows a web application interface with a dark blue header containing 'TUGAS AKHIR' and navigation links 'Home', 'Training', 'Testing', and 'User Input'. The main heading is 'Data Testing - Uji Coba Realtime'. Below it, the instruction 'Masukkan Data Post' is displayed. A large text input field contains the placeholder text 'enter post'. To the right of the input field is a dark blue button labeled 'Mulai Testing'. Below the input field, the text 'Hasil Prediksi Real Time' is visible.

Gambar 4. 7 Tampilan Halaman User Input

Setelah proses deteksi selesai, maka sistem akan menampilkan hasil deteksi dari sebuah kalimat yang sudah di *input* sebelumnya oleh pengguna. Tampilan hasil proses dari *user input Realtime* dan hasilnya apakah termasuk *spam* atau bukan *spam* akan ditampilkan pada Gambar 4.8 dan Gambar 4.9.

The screenshot shows the same web application interface as Gambar 4.7, but with the input field containing the text 'openbooking bungasolo naomisoloo jakartaselatan wa kulinerlendir'. A 'Reset' button is now visible next to the input field. Below the input field, the text 'Hasil Prediksi Real Time' is displayed. A table with two columns, 'Input' and 'Prediksi', is shown. The table contains one row with the input text and the prediction '1'. The table is paginated, showing 'Showing 1 to 1 of 1 entries'. A search bar is located at the top right of the table area. The table has a dark blue header with 'Input' and 'Prediksi' columns. The input text is 'openbooking bungasolo naomisoloo jakartaselatan wa kulinerlendir' and the prediction is '1'. The table is paginated, showing 'Showing 1 to 1 of 1 entries'. A search bar is located at the top right of the table area. The table has a dark blue header with 'Input' and 'Prediksi' columns. The input text is 'openbooking bungasolo naomisoloo jakartaselatan wa kulinerlendir' and the prediction is '1'. The table is paginated, showing 'Showing 1 to 1 of 1 entries'. A search bar is located at the top right of the table area.

Gambar 4. 8 Tampilan Halaman *User Input* (Post Spam)

Gambar 4. 9 Tampilan Halaman *User Input* (Post Non-spam)

5.3 Implementasi Model

Setelah melalui tahap pelatihan, langkah selanjutnya adalah menguji sistem untuk mengevaluasi kinerja dari model yang telah dibuat. Pengujian ini melibatkan proses pelatihan model dan pengujian model untuk memastikan bahwa sistem yang telah dibangun berfungsi sebagaimana yang diharapkan.

5.3.1 Menentukan Nilai Vektor Kata dengan TF-IDF

Untuk menentukan nilai vektor kata dengan menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*), ada beberapa langkah yang perlu diikuti. TF-IDF adalah teknik yang berguna untuk mengubah teks menjadi fitur numerik yang dapat diproses oleh model *machine learning*. Dengan menggabungkan frekuensi kemunculan kata dalam dokumen dengan kelangkaan kata di seluruh *korpus*, TF-IDF membantu menyoroti kata-kata yang penting untuk pengklasifikasian atau analisis teks lainnya.

Adapun tahapan dalam penerapan TF-IDF yaitu:

1. Menghitung vektor TF-IDF, di mana setiap baris mewakili sebuah dokumen (tweet) dan setiap kolom mewakili sebuah kata. Nilai dalam vektor adalah skor TF-IDF untuk setiap kata dalam setiap dokumen. Gambar 4.10 menunjukkan contoh nilai vektor dari kata 'ayo'.

[0.	0.	0.	0.50404022	0.	0.43590711
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.62937424
0.	0.	1.	0.	0.	0.45233183
0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	0.	0.	0.	0.
0.50404022	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.70480901	0.	0.	0.	0.	0.
0.	0.	0.	0.36099079	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.34685662	0.	0.	0.	0.
0.46501068	0.	0.	0.	0.	0.43590711
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.36099079	0.	0.
0.	0.	0.	0.	0.	0.34685662
0.43590711	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	1.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	0.	0.	0.34685662	0.
0.	0.	0.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.43590711	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	1.	0.78131142	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.50404022	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.36099079	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.46756168	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.46756168	0.

Gambar 4. 10 Vektor dari kata ‘ayo’

2. Nilai rata-rata yang dihasilkan dari semua vektor kata “ayo” yang ada yaitu 0.46416724. Jadi setiap kata yang ada dalam korpus akan diwakili ataupun memiliki satu angka yang merupakan hasil rata-rata vektornya. Angka yang mewakili satu kata tersebut selanjutnya akan disimpan ke dalam file untuk dijadikan sebuah kamus kata dengan nilai rata-rata vektornya. Gambar 4.11 berikut akan menunjukkan contoh beberapa kata yang ada dalam kamus kata dengan *average* (nilai rata-rata) vektornya.


```
'aduk': 0.372985408,
'agen': 0.32630278,
'ajar': 0.32373618,
'ayo': 0.46416724,
'bahasa': 0.23682337,
'bangkit': 0.89361574,
'info': 0.71605416,
'putar': 0.470769832,
'wujud': 0.35980261,
```

Gambar 4. 11 Kamus kata dengan nilai rata-rata vektor

3. Kemudian semua kata yang ada pada data akan diganti dan diwakili oleh nilai rata-rata vektor yang ada pada kamus kata sebelumnya. Contoh kata yang telah melalui proses seperti ['ayo ', 'bangkit'] akan diganti dan diwakili menjadi [0.46416724, 0.89361574] dan selanjutnya akan diproses dengan menggunakan algoritma *Random Forest Classifier*.

5.3.2 Pelatihan Model Random Forest Classifier

Algoritma Random Forest Classifier memiliki parameter yang dapat di-*tuning* untuk meningkatkan performansi model yang dibangun. Parameter-parameter ini dikenal dengan *hyperparameters*. *Hyperparameters* ini dapat di-*tuning* untuk menghasilkan model dengan performansi yang lebih baik (Clara, 2021).

Hyperparameters tuning Random Forest Classifier dilakukan pada seluruh kombinasi parameter. *Hyperparameters* algoritma *Random Forest Classifier* yang digunakan pada penelitian ini adalah sebagai berikut:

1. Parameter *n_estimators*, merupakan jumlah pohon yang dibangun algoritma sebelum mengambil voting maksimum atau rata-rata prediksi. Pada penelitian ini, penulis menggunakan parameter *n_estimators* dengan rentang 10 sampai 100.
2. Parameter *max_depth*, adalah kedalaman maksimum dari setiap pohon keputusan, untuk membatasi kedalaman pohon dapat mencegah overfitting. Nilai yang umum *max_depth* berkisar dari antara 7 hingga 30, tetapi ini bisa bervariasi tergantung pada dataset. Pada penelitian ini, peneliti telah mencari nilai terbaik untuk *max_depth* yaitu sebesar 25.
3. Parameter *max_features*, mendefinisikan jumlah maksimal banyaknya fitur yang dipilih secara acak yang akan dipertimbangkan ketika melakukan percabangan pada

tree. Pada penelitian ini, peneliti menggunakan parameter *max features* = ‘sqrt’ (by default atau nilai umum).

4. Parameter *criterion*, mendefinisikan jenis *split* yang akan dilakukan pada setiap *node* di pohon keputusan. Pada penelitian ini, menggunakan parameter *criterion* ‘gini’ untuk mendapatkan nilai *gini impurity* dalam membangun pohon keputusan.

Pada implementasi ini *decision tree* sebagai *basic learner* untuk *random forest* menggunakan implementasi dari pustaka *Scikit-learn*. Implementasi dari penggunaan *Random Forest* ini dapat dilihat pada Gambar 4.12.

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=100, random_state=42, max_depth=25)
clf.fit(X_train, Y_train)
```

<ipython-input-18-80c8c14054ca>:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. For more information, see https://www.python.org/dev/peps/pep-0185/#column-vector-vs-array
clf.fit(X_train, Y_train)

RandomForestClassifier
RandomForestClassifier(max_depth=25, random_state=42)

Gambar 4. 12 Implementasi metode klasifikasi Random Forest

Model yang dibangun akan diukur performansinya menggunakan akurasi, *precision*, *recall*, dan *F1-Score*. Performansi kombinasi *hyperparameters tuning* yang dilakukan dapat dilihat dalam Tabel 4.1 dimana S dinyatakan untuk *spam* dan NS untuk *non spam*.

Tabel 4. 1 Performansi *Hyperparameters Tuning*

number of trees	max features	Criterion	max depth	Precision		Recall		F1-score		accuracy
				S	NS	S	NS	S	NS	
10	‘sqrt’	‘gini’	7	96%	77%	71%	97%	82%	86%	84%
20	‘sqrt’	‘gini’	7	96%	78%	72%	96%	83%	87%	85%
30	‘sqrt’	‘gini’	10	96%	83%	88%	97%	88%	89%	89%
40	‘sqrt’	‘gini’	10	95%	86%	84%	86%	89%	90%	90%
50	‘sqrt’	‘gini’	15	95%	89%	89%	95%	92%	92%	92%
60	‘sqrt’	‘gini’	15	95%	90%	89%	96%	92%	93%	92%
70	‘sqrt’	‘gini’	20	95%	94%	94%	95%	94%	94%	94%
80	‘sqrt’	‘gini’	20	95%	94%	94%	95%	95%	95%	95%
90	‘sqrt’	‘gini’	25	95%	95%	95%	95%	95%	95%	95%
100	‘sqrt’	‘gini’	25	96%	98%	98%	96%	97%	97%	97%

Hyperparameters tuning pada algoritma Random Forest yang menggunakan distribusi parameter pada tabel 4.1 menunjukkan bahwa akurasi terbaik terdapat pada kombinasi terakhir yaitu sebesar 97%. Nilai *F1-Score* terbaik sebesar 97% untuk spam dan 97% untuk non spam. Untuk *recall* terbaik sebesar 98% untuk spam dan 96% untuk non spam. Nilai *precision* tertinggi sebesar 96% untuk spam dan 98% untuk non spam. Semuanya ada pada kombinasi terakhir yaitu dengan *number of trees* = 100. Hal ini menunjukkan bahwa model yang dibangun dengan menggunakan algoritma *Random Forest Classifier* sangat bergantung pada jumlah pohonnya. Semakin banyak jumlah pohon, maka akan semakin tinggi akurasi yang didapatkan.

Pencarian *hyperparameters tuning* terbaik dilakukan dengan prinsip *trial and error*. Seluruh kombinasi *tuning* dilakukan dengan menggunakan *criterion* ‘gini’ yang mengacu pada penggunaan *CART Decision Tree*. Seluruh *hyperparameters tuning* juga dilakukan menggunakan *maximum features* dengan nilai ‘sqrt’ atau akar dua dari total fitur yang ada pada data.

5.4 Hasil Pengujian Sistem

Hasil model yang sudah dipelajari dan disimpan setelah melalui proses *training* selanjutnya akan diuji dengan data *testing* yang telah disediakan sebelumnya. *TF-IDF* dan *Random Forest Classifier* akan mengidentifikasi data training yang sudah ada sebelumnya dan selanjutnya akan diuji untuk memperoleh hasil evaluasi terhadap algoritma yang digunakan. Dalam pengujian ini menggunakan data akan melalui tahap *pre-processing* yaitu data *cleaning*, *case folding*, *punctuation removal*, *normalization*, *stopword removal*, *stemming*, dan *tokenizing*. Setelah dilakukannya tahap *pre-processing*, setiap kata akan diubah menjadi indeks yang berupa urutan *integer*, *integer* ini akan menghasilkan prediksi menggunakan model yang telah di latih sebelumnya. Hasil pengujian sistem dapat diperhatikan pada Tabel 4.2.

Tabel 4. 2 Hasil Pengujian Sistem

Post	Pre-processing	Label Score	Prediction
Oh ini #HariKebangkitanNasional kirain hari kebangkitan rasa :" Eeeaaa	harikebangkitannasion al kira bangkit eeeaaa	0	0

#HariKebangkitanNasional Aplikasi android yang menghasilkan uang hingga 30\$/hari https://t.co/nkdmvsY9Fz	harikebangkitannasional aplikasi android hasil duit	1	1
Lakukan perjuangan itu dgn tindakan nyata. Berkarya nyata!Â selamat #HariKebangkitanNasional bangsaku tumpah darahku	laku juang tindak nyata karya nyata selamat harikebangkitannasional bangsa tumpah darah	0	0
Hai @mutiaraa707 Pengen tau siapa yang sering stalkingin akun kamu? Cek disini aja kak http://t.co/31v-XDHunN9	mutiaraa tau stalkingin akun cek kakak	1	1
Hai @nadia_rahmawaty Pengen tau siapa yang sering stalkingin akun kamu? Cek disini aja kak http://t.co/31vXDHunN9	nadia rahmawaty tau stalkingin akun cek kakak	1	1
Para pemuda sebagai calon pemimpin, agar pemimpin pandai merasa, merasakan kesulitan hidup rakyat, mendengar rakyatâ€¦ https://t.co/zAB47vrPc5	pemuda calon pimpin pimpin pandai rasa sulit hidup rakyat dengar rakyat	0	0
#TarawihInstagramable aku masih butuh penjelasan yang benar lagi badak	tarawihinstagramable butuh jelas badak	0	1
Setiap saya pulang, Ibu selalu membisikkan jangan korupsi ya! #GanjarTakTakutPakDirman	pulang bisik korupsi ganjartaktakutpakdirman	0	0
BONUS REBATE UNTUK SPORTBOOK 0.75% Ayo Gabung Disini :	bonus rebate sportbook ayo gabung	1	1

https://t.co/F9vivPckha Whatsapp: +6282168210878 BBM:â€ https://t.co/0YyGrorM2Y	whatsapp bahan bakar minyak		
.	.	.	.
.	.	.	.
.	.	.	.
Cerita-cerita lucu, bagi mereka yang sedang berpuasa di awal bulan ramadan ini. #EraId #Ramadan #Milenialâ€ https://t.co/SMzn6MgD0r	cerita cerita puasa ramadan eraid ramadan milenial	0	1
kekalahan yang memalukan deh https://t.co/z6Z9zEbXhk #HariKebangkitanNasional #SemangatPuasa #Harlah7Moedaâ€ https://t.co/DobhyKij3u	kalah malu harikebangkitannasion al semangat puasa harlah moeda	1	0
Hai Kak, @_anggraini22, mau dapat info terbaru seputar Makassar ? follow @InfoMakassarID ya.. Pasti di Folback !	kakak anggraini info baru putar makassar follow infomakassarid folback	1	1
Superhero Wanita dalam kehidupan nyata ? ðŸ™ #KartuJitu #AgenSBOBET #AgenMaxbet #AgenBola #EfekJomblo #Persebayaâ€ https://t.co/kfdgcAsDBL	superhero wanita hidup nyata kartujitu agensbobet agenmaxbet agenbola efekjomblo baya	1	1
Hallo, @VhyraeeL_Tandi, mau dapat info terbaru seputar Makassar ? follow	vhyraeel tandi info baru putar makassar follow infomakassarid folback	1	1

@InfoMakassarID ya.. Pasti di Folback !			
Ada cerita lucu @Staquf ketemu @MikePenceVP di sini https://t.co/8wg4AP0BsE #NasionalismeZamanNow #Indonesia	cerita staquf ketemu mikepencevp nasionalismezamanno w Indonesia	1	0
hai kak,@Twentytwo994,Ada Info Nih Buat Temen-Temen yg Pingin Bisa Bahasa Inggris tanpa Kursus Info http://t.co/cZm3xRMJQ4	kakak twentytwo info nih teman teman bahasa inggris kursus info	1	1
Siang semua . Jangan lupa Login dan bermain di https://t.co/F3xSj6ogMU siang ini ya ^_^ #Indonesiaâ€ https://t.co/Y857ObRUxB	siang lupa login main siang Indonesia	1	0

Pada saat melakukan pengujian, ditemukan adanya ketidaktepatan sistem dalam mengklasifikasikan data karena beberapa faktor tertentu. Penulis menganalisis beberapa faktor yang kemungkinan merupakan penyebab terjadinya kesalahan dalam proses prediksi, yaitu sebagai berikut:

1. Dalam post tweet menggunakan narasi kalimat yang variatif/bukan template sehingga dideteksi sebagai bukan spam oleh sistem.

Contoh:

Siang semua . Jangan lupa Login dan bermain di <https://t.co/F3xSj6ogMU> siang ini
ya ^_^ #Indonesiaâ€ <https://t.co/Y857ObRUxB>

Post tweet diatas dideteksi sebagai bukan spam dikarenakan menggunakan narasi bahasa yang variatif dan tidak terlihat seperti template seperti post biasa yang isinya hanya ingin mengingatkan dan menyapa walaupun isinya mengandung hastag dan link berisi spam.

2. Saat melakukan pengujian pada *user input* masih terjadi kesalahan prediksi pada model dikarenakan data yang digunakan masih kurang variatif, akibatnya model kurang mengenal pola-pola dari setiap kalimat.
3. Dalam post tweet terjadi penghapusan link aktif pada saat proses *pre-processing* sehingga dideteksi sebagai bukan spam oleh sistem.

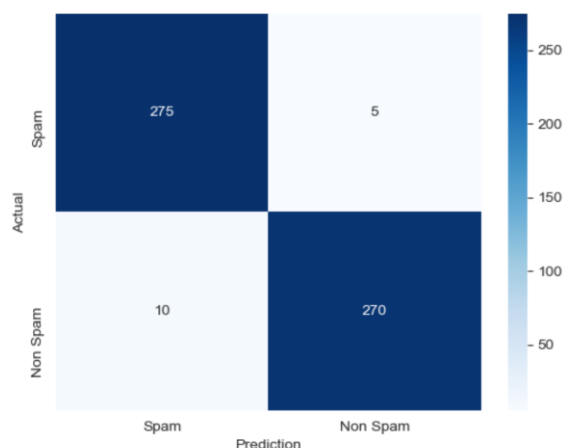
Contoh:

Siang semua . Jangan lupa Login dan bermain di <https://t.co/F3xSj6ogMU> siang ini ya ^_^ #Indonesiaâ€¦ | <https://t.co/Y857ObRUxB>

Post tweet diatas dideteksi sebagai bukan spam dikarenakan link yang dijadikan bahan promosi dihapus pada saat proses pre-processing.

5.5 Hasil Evaluasi

Evaluasi dilakukan pada keseluruhan model Random Forest Classifier yang parameternya telah melalui proses *tuning*. Pendekatan evaluasi menjadi acuan untuk menghitung deteksi spam dan bukan spam menggunakan TF-IDF dan Random Forest Classifier. Penelitian ini mengadopsi *Confusion Matrix* untuk mengukur nilai *precision*, *recall*, *f-score*, dan *accuracy*. Informasi ditampilkan dengan menggunakan visualisasi *heatmap* untuk mempermudah melihat pola berdasarkan TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*), yang dapat kita lihat pada Gambar 4.10



Gambar 4. 13 Confusion Matrix

Dengan melihat visualisasi *heatmap* pada gambar 4.10 maka dapat dihitung jumlah TPT (True Positive Post), TNP (True Negative Post), FPP (False Positive Post) dan FNP

(*False Negative Post*). Tabel 4.3 dan 4.4 menunjukkan keterangan pernyataan spam dan non-spam pada metode evaluasi *Confussion Matrix*.

Tabel 4. 3 Keterangan Post Spam *Confussion Matrix*

No	Post Spam	Jumlah
1	TPP (True Positive Post)	275
2	FPP (False Positive Post)	10
3	TNP (True Negative Post)	270
4	FNP (False Negative Post)	5

Tabel 4. 4 Keterangan Post Non-spam *Confussion Matrix*

No	Post Non-spam	Jumlah
1	TPP (True Positive Post)	270
2	FPP (False Positive Post)	5
3	TNP (True Negative Post)	275
4	FNP (False Negative Post)	10

Berdasarkan hasil deteksi post spam dan post non-spam pada Tabel 4.3 dan 4.4, maka evaluasi dapat dilakukan dengan menghitung nilai *Precision*, *Recall*, *F-measure*, dan *Accuracy* dengan menggunakan persamaan yang telah dijelaskan pada bab sebelumnya.

Kalkulasi *Precision*:

$$Precision \text{ Post Non-Spam (0)} = \frac{TPP0}{TPP0 + \sum FPP0} = \frac{270}{270+5} = 0,98$$

$$Precision \text{ Post Spam (1)} = \frac{TPP1}{TPP1 + \sum FPP1} = \frac{275}{275+10} = 0,96$$

Kalkulasi *Recall*:

$$Recall \text{ Post Non-Spam (0)} = \frac{TPP0}{TPP0 + \sum FNP0} = \frac{270}{270+10} = 0,96$$

$$Recall \text{ Post Spam (1)} = \frac{TPP1}{TPP1 + \sum FNP1} = \frac{275}{275+5} = 0,98$$

Kalkulasi *FI-score*:

$$F1\text{-score Post Non-Spam (0)} = \frac{2 \times \text{Recall (0)} \times \text{Precision (0)}}{\text{Recall (0)} + \text{Precision (0)}} = \frac{2 \times 0.96 \times 0.98}{0.96 + 0.98} = 0,97$$

$$F1\text{-score Post Spam (1)} = \frac{2 \times \text{Recall (1)} \times \text{Precision (1)}}{\text{Recall (1)} + \text{Precision (1)}} = \frac{2 \times 0.98 \times 0.96}{0.98 + 0.96} = 0,97$$

Total keseluruhan akurasi:

$$\text{Accuracy} = \frac{TPP+TNP}{TPP+TNP+FPP+FNP} \times 100\% = \frac{275+270}{275+270+10+5} \times 100\% = 0,973 \times 100\% = 97,3\%$$

Berdasarkan hasil evaluasi diatas dengan menghitung nilai *Precision*, *Recall*, *F1-Score*, dan Akurasi pada post dan non-spam maka dapat diambil kesimpulan sebagai berikut dilampirkan pada Tabel 4.5.

Tabel 4. 5 Hasil Evaluasi

	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Spam (1)	0,96	0,98	0,97
Non-spam (0)	0,98	0,96	0,97

Berdasarkan table diatas, nilai *precision* lebih tinggi pada pernyataan non-spam yaitu sebesar 98% dan pada pernyataan spam memiliki nilai *precision* yang lebih rendah yaitu sebesar 96%. Hal ini menunjukkan bahwa rasio yang digunakan untuk memprediksi dengan rata-rata 97% akurat terhadap *True Positive* (TP) pada post spam dan non-spam. Model yang dibuat juga telah sesuai, dapat dibuktikan melalui nilai *recall*, yaitu nilai pada post spam sebesar 98% dan pada non-spam sebesar 96%. Nilai *F1-Score* sama pada kedua pernyataan yaitu 97% menunjukkan bahwa rata-rata perhitungan *precision* dan *recall* sudah sesuai sehingga menghasilkan akurasi yang baik yaitu sebesar 97%. Berdasarkan akurasi yang telah dicapai pada penelitian ini, diketahui bahwa sistem dapat mendeteksi post spam dan non-spam menggunakan Random Forest dengan cukup baik.

BAB 5

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang dapat ditarik terkait dengan deteksi spam pada media sosial X berdasarkan post dan repost dengan menggunakan *Random Forest Classifier* adalah sebagai berikut:

1. Hasil kinerja algoritma Random Forest Classifier dan TF-IDF sebagai *word embedding* menunjukkan performa yang baik dengan memperoleh akurasi sebesar 97% yang telah dievaluasi menggunakan *Confussion Matrix*.
2. Berdasarkan hasil evaluasi yang didapat, maka algoritma Random Forest Classifier dengan *word embedding* TF-IDF dapat bekerja dengan baik untuk melakukan deteksi post spam dan non spam.

5.2 SARAN

Peneliti mengakui bahwa kajian yang telah dilakukan masih memiliki ruang untuk perbaikan yang signifikan. Dalam menanggapi hal ini, penulis ingin menyampaikan beberapa saran konstruktif kepada peneliti yang berkeinginan untuk melanjutkan atau meningkatkan kualitas penelitian ini, meliputi:

1. Memperluas jangkauan bahasa pada post yang dapat dideteksi sehingga tidak terbatas hanya pada post yang menggunakan bahasa Indonesia saja.
2. Selain mengimplementasikan aplikasi berbasis website, dapat dilakukan eksplorasi untuk mengembangkan aplikasi dalam bentuk lain, seperti platform Android.
3. Memanfaatkan penggunaan atribut data yang lebih beragam seperti: foto profil atau avatar, tahun bergabung, *following*, *followers*, dll guna meningkatkan proses deteksi.
4. Mengimplementasikan fitur deteksi tidak hanya pada akun yang bersifat publik, tetapi juga berlaku untuk akun yang *private*.

DAFTAR PUSTAKA

- Aditya, C. S. K., Hani'ah, M., Fitrawan, A. A., Arifin, A. Z., & Purwitasari, D. (2019). Deteksi Bot Spammer pada Twitter Berbasis Sentiment Analysis dan Time Interval Entropy. *Jurnal Buana Informatika*, 7(3), 179–186. <https://doi.org/10.24002/jbi.v7i3.656>
- Andita Wahyuningtyas., Sitanggang, Imas Sukaesih., Khotimah, Husnul., (2020). “Deteksi Spam pada Twitter Menggunakan Algoritme Naïve Bayes”, *Jurnal Ilmu Komputer Agri-Informatika* volume 7 no 1 halaman 31 – 40. eISSN: 2654-9735. 2020. <http://journal.ipb.ac.id/index.php/jika>
- Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. ((Rachmat & Lukito, 2017)). “Detecting Spammers on Twitter”. *Proc. Collaboration Electron. Messaging Anti-Abuse Spam Conf.* (CEAS), 6, 3156–3164. <https://doi.org/10.1021/bi972148>
- BPS. (2023). 06300.2313. Statistik Telekomunikasi Indonesia 2022. <https://www.bps.go.id/publication/2023/08/31/131385d0253c6aae7c7a59fa/statistik-telekomunikasi-indonesia-2022.html#:~:text=Abstraksi,10%20persen%20di%20tahun%202021.>
- Eshraqi, N., Jalali, M., & Moattar, M. H. (2015). “Detecting spam tweets in Twitter using a data stream clustering algorithm”. *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, 347–351. <https://doi.org/10.1109/ICTCK.2015.7582694>
- Hanafiah, Muhammad., Herdiani, Anisa., & Astuti, Widi. (2019). “Klasifikasi Spam Tweet Pada Twitter Menggunakan Metode Naïve Bayes (Studi Kasus: Pemilihan Presiden 2019)”. *e-Proceeding of Engineering* Vol.6, No.2 ISSN : 2355-9365, 2019. <https://openlibrary.telkomuniversity.ac.id/pustaka/153032/klasifikasi-spam-tweet-pada-twitter-menggunakan-metode-na-ve-bayes-studi-kasus-pemilihan-presiden-2019-.html>

- Mathiarasi, B., & Shyni, Emilin., (2014). "Detection of Spam links in Twitter", *International Journal of Engineering Research & Technology (IJERT)* ISSN: 2278-0181 Vol. 3 Issue 3, March, 2014. <https://www.ijert.org/detection-of-spam-links-in-twitter>
- Priyatno, Arif Mudi. (2020). "Deteksi Akun Spammer Berdasarkan Hashtag dan Aktifitas Komunitas pada Twitter". *TESIS – IF* 185401. <https://repository.its.ac.id/78225>
- Rachmat, Antonius., & Lukito, Yuan. (2017). "Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes". *Ultimatics: Jurnal Ilmu Teknik Informatika* Vol 9 No 1 (2017). <https://doi.org/10.31937/ti.v9i1.564>
- Rachmat, Antonius., & Lukito, Yuan. (2016). "SENTIPOL: Dataset Sentimen Komentar Pada Kampanye PEMILU Presiden Indonesia 2014 dari Facebook Page", in *Konferensi Nasional Teknologi Informasi dan Komunikasi 2016*. pp. 218-228. <https://knastik.ukdw.ac.id/2016/makalah/artikel/e7-j1.pdf>
- Ramli, Rafly Ghazali & Sibaroni, Yuliant. (2022). "Klasifikasi Topik Twitter menggunakan Metode Random Forest dan Fitur Ekspansi Word2Vec", *e-Proceeding of Engineering* : Vol.9, No.1 Februari, eISSN: 2355-9365, 2022. <https://repository.telkomuniversity.ac.id/pustaka/177451/klasifikasi-topik-twitter-menggunakan-metode-random-forest-dan-fitur-ekspansi-word2vec.html>
- Rofi'ah, Ekawati. (2017). "Self Disclosure (Pengungkapan Diri) sRemaja Perempuan Melalui Media Sosial Twitter". *SKR FIS* 2-051800499. 2018. <http://repository.ub.ac.id/id/eprint/9886/>
- Septiandri, A. A. & Wibisono, O. (2017). Detecting Spam Comments on Indonesia's Instagram Posts. *Journal of Physics: Conference Series*, (pp. 67-73).

- Syam, Agus Tiyanasyah., dkk., (2020). “Klasifikasi Komentar Spam Pada Instagram Menggunakan Metode Support Vector Machine”, *Jurnal Buffer Informatika* volume 6 no 2, pp. 2527-4856, 2020.
<https://journal.uniku.ac.id/index.php/buffer>
- Urbanjabar.com. (2023). Awas Ini Ciri-Ciri Spamming Yang Harus Kamu Tahu. Diakses pada 26 September 2023,
<https://www.urbanjabar.com/featured/927928241/awas-ini-ciri-ciri-spamming-yang-harus-kamu-tahu>.
- Verma, M., & Sofat, S. (2014). Techniques to Detect Spammers in Twitter-A Survey. *International Journal of Computer Applications*, 85(10), 27-32,
https://www.researchgate.net/publication/262992888_Techniques_to_Detect_Spammers_in_Twitter-_A_Survey.
- X. (2023). “Komunitas di X”. Diakses pada 18 September 2023,
<https://help.twitter.com/id/using-twitter/communities>.
- X. (2023). “Tentang X Premium”. Diakses pada 19 September 2023,
<https://help.twitter.com/id/using-twitter/twitter-blue>



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR : 2722/UN5.2.14.D/SK/SPB/2024
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 11 Juli 2024 perihal permohonan ujian skripsi:
Nama : SURYANA MEISSARAH ZAINI SINAGA
NIM : 191402055
Program Studi : Sarjana (S-1) Teknologi Informasi
Judul Skripsi : Deteksi SPAM Pada Media Sosial X Berdasarkan Post dan Repost Dengan Menggunakan Metode Random Forest Classifier
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026
- MEMUTUSKAN
- Menetapkan :
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
Ketua : Mohammad Fadly Syah Putra, B.Sc., M.Sc.
NIP: 198301292009121003
Sekretaris : Dr. Sawaluddin, M.IT
NIP: 195912311998021000
Anggota Penguji : Ivan Jaya S.Si., M.Kom.
NIP: 198407072015041001
Anggota Penguji : Fanindia Purnamasari S.TI,M.IT
NIP: 198908172019032023
Moderator : -
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

- Tembusan :
1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
 2. Yang bersangkutan
 3. Arsip

Medan
Ditandatangani secara elektronik oleh:
Dekan



Maya Silvi Lydia
NIP 197401272002122001