

**IDENTIFIKASI *PHISING* PADA PESAN TEKS MENGGUNAKAN
ALGORITMA *SUPPORT VECTOR MACHINE* DENGAN
*ENSEMBLED BAGGING***

SKRIPSI

OLEH :

KELVIN NATHANIEL LUMBANRAJA

201402050



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

2024

**IDENTIFIKASI PHISING PADA PESAN TEKS MENGGUNAKAN ALGORITMA
SUPPORT VECTOR MACHINE DENGAN ENSEMBLED BAGGING**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

KELVIN NATHANIEL LUMBANRAJA
201402050



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

PERSETUJUAN

Judul : Identifikasi Phising pada Pesan Teks Menggunakan Support Vector Machine dengan Ensambled Bagging

Kategori : Skripsi

Nama Mahasiswa : Kelvin Nathanael Lumbanraja

Nomor Induk Mahasiswa : 201402050

Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi informasi Universitas Sumatera Utara

Medan, 04 Juli 2024

Komisi Pembimbing :

Pembimbing 2,

Ivan Jaya S.Si., M.Kom.

NIP. 198407072015041001

Pembimbing 1,

Dr. Erna Budhiarti Nababan M.IT

NIP. 196210262017042001

Diketahui/disetujui oleh

Program Studi S1 Teknologi Informasi

Ketua,

Dedy Arisandi, S.T., M.Kom.

NIP. 197908312009121002

PERNYATAAN

**IDENTIFIKASI PHISING PADA PESAN TEKS MENGGUNAKAN ALGORITMA
SUPPORT VECTOR MACHINE DENGAN ENSEMBLED BAGGING**

SKRIPSI

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

Medan, 04 Juli 2024



Kelvin Nathanael Lumbanraja
201402050

UCAPAN TERIMAKASIH

Penulis mengungkapkan rasa syukur dan terima kasih kepada Tuhan Yang Maha Esa yang telah mengaruniakan berkat-Nya dalam membantu penulis menuntaskan tugas akhir berupa skripsi sebagai persyaratan dalam meraih gelar sarjana dalam bidang Teknologi Informasi di Program Studi Teknologi Informasi, Universitas Sumatera. Judul skripsi ini adalah “Identifikasi Phising Pada Pesan Teks Menggunakan Algoritma Support Vector Machine Dengan Ensembled Bagging”. Pencapaian penyelesaian skripsi ini tidak mungkin terwujud tanpa kontribusi dari pihak-pihak yang terlibat dalam membantu penulis sepanjang langkah pelaksanaan skripsi. Karenanya, penulis hendak menyampaikan terima kasih yang tulus kepada :

1. Mama Tercinta, Herty Magdalena dan Papa Tercinta, Freddy Tabatua yang senantiasa memberikan dukungan, doa, dan kengahatan hati, yang telah membantu penulis untuk tetap kuat hingga sampai pada tahap penyelesaian skripsi ini.
2. Ibu Dr. Erna Budhiarti Nababan M.IT., selaku Dosen Pembimbing I yang telah memberi bimbingan, dukungan, motivasi, serta kritik dan saran yang konstruktif, yang telah diberikan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak Ivan Jaya S.Si., M.Kom., selaku Dosen Pembimbing II yang juga telah membimbing, memberikan motivasi, serta kritik dan saran kepada penulis dalam proses penyelesaian skripsi penulis ini.
4. Bapak Indra Aulia S.TI., M.Kom., selaku Dosen yang pernah memberikan bimbingan, arahan, saran dan kritik kepada penulis di awal penulisan, sehingga penulis mendapatkan inspirasi untuk memulai penulisan skripsi ini.
5. Bapak Dedy Arisandi S.T., M.Kom., selaku Ketua Prodi Teknologi Informasi, Universitas Sumatera Utara.
6. Ibu Dr. Maya Silvi Lydia B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

7. Bapak dan Ibu Dosen di Program Studi Teknologi Informasi, Universitas Sumatera Utara, yang telah memberikan pengetahuan yang berlimpah kepada penulis selama perkuliahan.
8. Bapak dan Ibu staff serta pegawai Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, yang telah memberikan dukungan sedari awal perkuliahan hingga penyelesaian skripsi penulis.
9. Angela Prima Lie, yang telah mendukung, menemani, dan memberi kekuatan di masa-masa sulit penulis hingga penyelesaian skripsi ini terwujud.
10. Jennifer Wijaya, yang telah membantu dalam proses penyempurnaan dataset dalam skripsi penulis sehingga dataset tersebut dapat diolah dengan baik.
11. Teman Teman ‘P Mabar’, Frans Mayandro, Pretty Ohara, Stephani Uli, Kevin Tulus, Yeftha El Imani yang telah saling memberikan bantuan, dukungan, dan semangat dalam menghadapi berbagai permasalahan selama perjalanan perkuliahan ini.
12. Christine Amanda, sebagai teman baik penulis yang telah membantu penulis melalui pemberian masukan dan berkenan menjawab pertanyaan-pertanyaan penulis selama proses pengembangan skripsi.
13. Teman Teman ‘voli hora hora’, Vicky Natanael, Felix Christian, Kevin Bangun, Jesika yang telah saling memberikan dukungan dan semangat dalam menghadapi berbagai permasalahan dalam perkuliahan ini.
14. Seluruh keluarga dan teman-teman lain yang tidak dapat disebutkan satu persatu, yang juga telah memberi dukungan dan semangat kepada penulis hingga berhasil menyelesaikan skripsi ini.

Medan, 04 Juli 2024

Penulis,



Kelvin Nathanael Lumbanraja

201402050

ABSTRAK

Phising merupakan salah satu kejahatan dalam sosial media yang mencuri data pribadi seseorang maupun sebuah industri. Phising dapat dikirim melalui berbagai bentuk dan salah satu nya melalui pesan teks, yang dinamakan smishing. Smishing berisikan pesan teks yang mencantumkan alamat email, nomor telepon, ataupun tautan website yang menarik agar penerima tidak menyadari kejahatan tersebut. Kejahatan seperti ini tentunya dapat merugikan penerima pesan dalam hal finansial ataupun kemanan data. Namun, Pengidentifikasi smishing masih sulit dilakukan karena pesan tersebut harus di identifikasi secara manual dimana proses tersebut memakan waktu yang lama. Dengan demikian, dibutuhkan suatu pendekatan yang dapat mengidentifikasi phising pada pesan teks secara cepat dan lebih akurat. Penelitian ini bertujuan untuk mengidentifikasi phising pada pesan teks menggunakan algoritma Support Vector Machine (SVM) dengan Ensembled Bagging. Metodologi yang digunakan meliputi pengumpulan data sebanyak 1600 teks phising dan tidak phising yang diperoleh dari penelitian sebelumnya dan kotak pesan peneliti, kemudian dilakukan preprocessing berupa cleaning, tokenisasi, eliminasi stopwords, dan stemming. Data kemudian dibagi menjadi data pelatihan dan data pengujian. Algoritma SVM digunakan sebagai model dasar dan dioptimalkan dengan Ensembled Bagging untuk meningkatkan akurasi identifikasi. Hasil penelitian menunjukkan bahwa pendekatan yang diusulkan berhasil mengidentifikasi phising dengan akurasi sebesar 95,2%, membuktikan bahwa kombinasi SVM dan Ensembled Bagging efektif dalam mengolah data pesan teks untuk mengidentifikasi phising.

Kata Kunci : *Phising, Smishing, Ensambled Bagging, Support Vector Machine, Pesan Teks.*

*IDENTIFICATION OF PHISING IN TEXT MESSAGES USING THE SUPPORT
VECTOR MACHINE ALGORITHM WITH ENSEMBLE BAGGING*

ABSTRACT

Phishing is a type of social media crime that steals personal data from individuals or industries. Phishing can be delivered in various forms, one of which is through text messages, known as smishing. Smishing contains text messages that include email addresses, phone numbers, or website links that attract the recipient without realizing the crime. Such crimes can harm the recipient financially or compromise data security. However, identifying smishing is still challenging because these messages must be identified manually, a process that takes a long time. Therefore, an approach is needed to detect phishing in text messages quickly and more accurately. This study aims to identify phishing in text messages using the Support Vector Machine (SVM) algorithm with Ensembled Bagging. The methodology includes collecting 1600 phishing and non-phishing texts obtained from previous research and the researcher's message box, then preprocessing the data through cleaning, tokenization, stopwords elimination, and stemming. The data is then divided into training and testing sets. The SVM algorithm is used as the base model and optimized with Ensembled Bagging to improve identification accuracy. The results show that the proposed approach successfully identifies phishing with an accuracy of 95.2%, proving that the combination of SVM and Ensembled Bagging is effective in processing text message data for phishing identification.

Keywords : *Phising, Smishing, Ensambled Bagging, Support Vector Machine, text message.*

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN	iv
UCAPAN TERIMAKASIH	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Manfaat Penelitian	3
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan	4
BAB 2 LANDASAN TEORI	6
2.1. Phising	6
2.2. Pesan Teks	7
2.3. Support Vector Machine	8
2.4. Ensembled Bagging	9
2.5. Cross Validation	10
2.6. GridSearch Cross Validation	11

2.7. Website	11
2.8. HTML	12
2.9. Bootstrap	13
2.10. JavaScript	14
2.11. Python	14
2.12. Flask	16
2.13. Confusion Matrix	16
2.14. Peneliti Terdahulu	18
BAB 3 ANALISIS DAN PERANCANGAN SISTEM	26
3.1. Data yang digunakan	26
3.2. Arsitektur Umum	28
3.2.1. Split Dataset	30
3.2.2. Pre Processing	30
3.2.3. Resampling	35
3.2.4. Data Training	36
3.2.5. Processing	36
3.2.6. Evaluasi	39
3.2.7. Deployment	40
3.2.8. Sistem Identifikasi	40
3.2.9. Output	40
3.3. Perancangan Sistem	41
3.3.1. Desain Halaman Cek Phising Teks	41
3.3.2. Desain Halaman Cek Phising File	42
3.3.3. Desain Halaman Training	43
3.3.4. Desain Halaman Testing	44
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM	46
4.1. Implementasi Sistem	46
4.1.1. Spesifikasi Perangkat Keras dan Perangkat Lunak	46
4.1.2. Penerapan Sistem Berbasis Web	46

4.2. Hasil Pelatihan Model	51
4.3. Hasil Pengujian model	60
4.4. Hasil Evaluasi Pengguna	62
BAB 5 KESIMPULAN DAN SARAN	66
5.1. Kesimpulan	66
5.2. Saran	66
DAFTAR PUSTAKA	68

DAFTAR TABEL

Tabel 2. 1 Confusion Matrix Dua Kelas (Qadrini et al., 2021)	17
Tabel 2. 2 Penelitian Terdahulu	22
Tabel 3. 1 Dataset Pesan Teks Normal dan Phising	27
Tabel 3. 2 Pemecahan Dataset	28
Tabel 3. 3 Implementasi Cleaning	31
Tabel 3. 4 Implementasi Tokenisasi	32
Tabel 3. 5 Implementasi Stop Word Elimination	34
Tabel 3. 6 Implementasi <i>Stemming</i>	35
Tabel 3. 7 Parameter Support Vector Machine	39
Tabel 3. 8 Confusion Matrix	39
Tabel 4. 1 Daftar Parameter	51
Tabel 4. 2 Hasil GridSearchCV	51
Tabel 4. 3 Hasil Rata-rata skor cross-validation dari Setiap Fold dengan Parameter Terbaik	55
Tabel 4. 4 Perhitungan data yang mengandung pesan normal (ham)	56
Tabel 4. 5 Perhitungan data yang mengandung pesan phising (smishing)	56
Tabel 4. 6 Hasil Komputasi Nilai Evaluasi	58
Tabel 4. 7 Hasil Komputasi Nilai Evaluasi SVM	59
Tabel 4. 7 Hasil Pengujian Model	60
Tabel 4. 8 Pernyataan salah oleh sistem dalam data testing	62
Tabel 4. 9 Tabel Pernyataan Evaluasi Pengguna	64
Tabel 4. 10 Hasil Evaluasi Pengguna	64
Tabel 4. 10 Hasil Evaluasi Pengguna (lanjutan)	65

DAFTAR GAMBAR

Gambar 2. 1 Hyperplane	9
Gambar 3. 1 Arsitektur Umum	29
Gambar 3. 2 pseudocode cleaning data	31
Gambar 3. 3 Pseudocode tokenisasi data	32
Gambar 3. 4 Pseudocode Stopwords data	33
Gambar 3. 5 Pseudocode stemming data	35
Gambar 3. 6 Diagram Alur Pelatihan Model	36
Gambar 3. 7 Contoh Matrix TF-IDF	37
Gambar 3. 8 Contoh Penempatan Nilai Fitur Data	37
Gambar 3. 9 Hyperplane dan Margin SVM	38
Gambar 3. 10 Desain Halaman Cek Phising Teks	41
Gambar 3. 11 Desain Halaman Cek Phising File	42
Gambar 3. 12 Desain Halaman Training	43
Gambar 3. 13 Desain Halaman Testing	44
Gambar 4. 1 Halaman Cek Phising Teks	47
Gambar 4. 2 Output Teks	47
Gambar 4. 3 Halaman Cek Phising File	48
Gambar 4. 4 Output Cek File	48
Gambar 4. 5 Halaman Training	49
Gambar 4. 6 Output Training	49
Gambar 4. 7 Halaman Testing	50
Gambar 4. 8 Output testing	50
Gambar 4. 9 Output Confusion Matrix	50
Gambar 4. 10 Confusion Matrix	55
Gambar 4. 11 Confusion Matrix Svm	59

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Phising merupakan salah satu kejahatan dalam sosial media yang mencuri data pribadi seseorang maupun sebuah industri. Menurut Badan Siber dan Sandi Negara (BSSN), terdapat kasus *phising* berbentuk *email* sebanyak 164.131 kasus di Indonesia pada tahun 2022. Penyerang akan mengirimkan pesan beserta tautan yang menarik agar penerima tidak menyadari kejahatan tersebut. Setelah Penerima sudah menekan link *phising* tersebut, maka penyerang berhasil menembus keamanan sistem perangkat penerima. Hal tersebut berakibat data pribadi milik penerima dapat diakses dengan mudah.

Selain *phising* melalui *email*, *phising* juga dapat dikirim melalui pesan teks. Menurut Kompas.com (2022), pesan teks masih menjadi sarana bagi para penjahat untuk melakukan *phising*. *Phising* yang dikirimkan melalui pesan teks dinamakan dengan *smishing*. *Smishing* ini berisikan pesan teks yang mencantumkan alamat *email*, nomor telepon, ataupun tautan *website*.

Beberapa penelitian sudah dilakukan terkait penelitian *phising*. Penelitian yang dilakukan oleh Purwanto et.al. (2020) berfokus pada PhishZip, algoritma baru yang menggunakan kompresi untuk temukan situs web phising. Beda dari metode umum, PhishZip lebih simpel dan gak perlu langkah awal khusus. Algoritma ini memanfaatkan zlib, yang berasal dari DEFLATE, algoritma kompresi data buat mengompresi situs web lewat HTTP. DEFLATE digunakan buat menyusun dua kamus kata umum di situs web phising dan non-phising. Kata-kata yang sering muncul punya nilai kemungkinan lebih tinggi. Setelah proses kompresi, PhishZip bisa mengklasifikasikan situs web jadi phising atau non-phising. Hasilnya menunjukkan PhishZip mampu mendeteksi situs web phising dengan baik: 80,04% benar, 18,25% salah, dengan akurasi total 80,89%.

Penelitian lainnya yang dilakukan oleh Espinoza et.al. (2019). Penelitian ini mendeteksi *phising* yang terjadi dalam email menggunakan siklus pemodelan *machine learning*. Proses pada penelitian ini dibagi menjadi 2 yaitu, pembuatan model gabungan menggunakan algoritma *Decision Tree* dan *Naive Bayes*. Selanjutnya akan

dilakukan validasi menggunakan algoritma *machine learning* lain. Hasil dari penggabungan dua algoritma dalam mendeteksi *email* pada *phising* ialah 96.77%.

Pada penelitian yang telah dilakukan oleh Sakib et.al. (2021). Penelitian ini memprediksi modus kelahiran menggunakan pengklasifikasi Ensembled Bagging. Penelitian ini bertujuan untuk memprediksi apakah seorang ibu akan melahirkan secara caesar atau melalui persalinan yang normal. Dataset yang digunakan berasal dari studi kasus yang berada di rumah sakit Bangladesh. Metode Ensembled Bagging akan menghasilkan model yang telah dilatih oleh algoritma pembelajaran mesin dengan data training. Pada kasus ini, peneliti menggunakan beberapa bantuan algoritma yaitu, Decision Tree, Naive Bayes, Support Vector Machine, dan KNN. Setelah semua model didapatkan, peneliti menguji model dengan data uji yang sudah dipisahkan sebelumnya. Dalam penelitian ini, dengan menggunakan Ensembled Bagging, hasil dari algoritma pembelajaran mesin meningkat. Akurasi Decision Tree sebesar 0.87 yang sebelumnya sebesar 0.85, lalu KNN sebesar 0.86 yang sebelumnya 0.79, dan Naive Bayes sebesar 0.84 yang sebelumnya 0.80. Metode Ensembled Bagging menggabungkan hasil dari model yang berbeda dan menggabungkannya menjadi model yang lebih baik dan lebih stabil.

Berdasarkan penelitian yang telah dilakukan, peneliti mengusulkan penggunaan algoritma support vector machine dan Ensembled Bagging. Pada penelitian sebelumnya, algoritma support vector machine berhasil melakukan prediksi dengan akurasi 86%. Pada penelitian sebelumnya juga, Ensembled Bagging berhasil meningkatkan akurasi dari algoritma Pembelajaran Mesin yang digunakan. Penelitian tersebut menunjukkan bahwa dengan adanya ensambled bagging, algoritma support vector machine akan lebih akurat dalam mengolah data.

Dengan penelitian yang sudah dilakukan, peneliti melakukan penelitian dengan judul “IDENTIFIKASI PHISING PADA PESAN TEKS MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE DENGAN ENSEMBLED BAGGING”.

1.2. Rumusan Masalah

Phising dalam sosial media masih banyak ditemukan hingga sekarang. Penyerang akan mengirimkan pesan yang menarik agar penerima tidak menyadari tindak kejahatan tersebut. Pesan tersebut berisikan teks dan attribut lain seperti alamat *email*,

nomor telepon, dan bisa juga tautan sebuah *website*. Saat penerima pesan menekan salah satu dari atribut tersebut maka, data pribadi penerima dapat dengan mudah diakses oleh penyerang dan kerugian finansial serta penyebaran data sensitif tidak dapat dihindari. Sehingga dibutuhkan suatu pendekatan yang dapat mengidentifikasi phising pada pesan teks dengan lebih akurat.

1.3. Tujuan Penelitian

Mengidentifikasi *phising* pada pesan teks menggunakan algoritma *support vector machine* dengan *ensembled bagging*.

1.4. Batasan Masalah

Penelitian tersebut memiliki beberapa batasan masalah agar penelitian tidak menyimpang dari tujuan yang akan dicapai. Batasan-batasan yang disebut adalah :

1. Datas yang digunakan merupakan dataset yang berisi pesan *phising* dan tidak *phising*.
2. Data yang digunakan berbentuk teks yang dimuat pada *file* berekstensi .csv.
3. Dataset dikumpulkan dari 3 sumber yaitu, penelitian terdahulu (jurnal internasional), artikel, dan juga pengumpulan manual.
4. Keluaran yang dihasilkan melalui penelitian ini mencakup pengimplementasian model dalam sistem berbasis web. Sistem tersebut mampu mengidentifikasi pesan yang dimasukkan oleh pengguna mengandung *phising* ataupun tidak dan juga dapat memberikan label pada data pesan berbentuk .csv yang belum memiliki label secara otomatis.

1.5. Manfaat Penelitian

Manfaat dari penelitian tersebut ialah :

1. Pesan teks langsung teridentifikasi apakah itu mengandung *phising* atau tidak.
2. Membantu pengguna terhindar dari kerugian finansial ataupun informasi karena pesan *phising*.
3. Bagi peneliti, lebih memahami algoritma *support vector machine* yang ditingkatkan dengan *ensembled bagging* dalam mengolah data.
4. Mengetahui kinerja dari algoritma *support vector machine* yang didukung oleh *ensembled bagging* dalam mengidentifikasi *phising*.

1.6. Metodologi Penelitian

Tahapan yang harus dilewati dalam pelaksanaan penelitian ini mencakup beberapa hal :

1. Studi Literatur

Studi literatur merupakan tahap peneliti mengumpulkan referensi dari berbagai sumber referensi seperti artikel, jurnal, skripsi, buku, dan juga sumber lainnya yang berisikan informasi pesan teks yang menganung phising, ensambled bagging, text processing, dan algoritma support vector machine.

2. Analisis Permasalahan

Pada tahap ini, analisis permasalahan akan dilakukan dengan menggunakan referensi yang telah didapatkan sebelumnya guna memahami algoritma support vector machine dan ensambled bagging. Hal tersebut dapat membantu pemahaman dalam sistem untuk mengidentifikasi phising pada pesan teks dalam penelitian ini.

3. Perancangan Sistem

Selanjutnya merupakan tahap perancangan sistem dimana dalam tahap ini, akan dilakukan perancangan arsitektur umum, pengumpulan dataset, desain antarmuka untuk sistem yang akan dibuat dalam penelitian ini.

4. Implementasi

Lalu akan dilakukan tahap implementasi rancangan yang telah disusun dari tahap sebelumnya untuk membangun sistem yang sesuai dengan tujuan penelitian.

5. Pengujian Sistem

Pengujian dilakukan dengan melakukan evaluasi akurasi ketepatan metode dalam penggunaan algoritma support vector machine dan ensambled bagging dalam mengidentifikasi pesan teks yang mengandung phising.

6. Penyusunan Laporan

Tahap terakhir dari penelitian ini merupakan penulisan sebuah laporan yang bertujuan untuk mencatat secara komprehensif seluruh proses penelitian yang telah dijalankan, dipresentasikan dalam bentuk laporan.

1.7. Sistematika Penulisan

Dalam Penelitian ini, penulisan dibagi menjadi lima bagian utama, meliputi :

Bab 1: Pendahuluan

Bagian awal, yang dikenal sebagai pendahuluan, merangkum latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan struktur penelitian.

Bab 2: Landasan Teori

Bagian ini membahas teori-teori yang berhubungan dengan penelitian guna membantu meningkatkan pemahaman serta menuntaskan pokok masalah dari penelitian yang dilakukan. Dalam landasan teori, *phising* pada pesan teks serta metode yang digunakan yaitu *support vector machine* dan *ensambled bagging* akan dijelaskan. Terdapat juga penelitian-penelitian sebelumnya yang dijadikan acuan dalam pelaksanaan penelitian ini.

Bab 3: Analisis dan Perancangan Sistem

Bagian ini berisi penjelasan dari permasalahan studi dan perencanaan sistem pengidentifikasi pesan teks mengandung phising menggunakan *support vector machine* dan *ensamble bagging*.

Bab 4: Implementasi dan Pengujian Sistem

Pada bagian ini dijelaskan tahap-tahap pengaplikasian dari rancangan sistem yang telah dibuat sebelumnya. Lalu keluaran dari pengujian serta evaluasi sistem juga dijelaskan pada baian ini.

Bab 5: Kesimpulan dan Saran

Segmen ini berisi rangkuman dari hasil penelitian. Apakah tujuan penelitian berhasil dicapai, serta penyampaian rekomendasi untuk penyempurnaan pada penelitian mendatang.

BAB 2

LANDASAN TEORI

2.1. Phising

Phishing merupakan kejahatan digital yang memanipulasi korbannya untuk memberikan informasi pribadi kepada pelakunya. *Phishing* melibatkan beberapa metode pengiriman *email* atau panggilan telepon, termasuk situs web jahat, promosi harga palsu, dan berbagai metode dan trik yang digunakan oleh penyerang untuk memata-matai korban (Alotaibi et.al., 2021).

Proses kerja *phising* dimulai dengan pelaku menentukan target korban dan merencanakan cara untuk mengumpulkan informasi yang berguna. Selanjutnya, mereka membuat email palsu atau halaman web yang menyerupai situs asli untuk menipu korban. Setelah berhasil mengirim pesan yang meyakinkan, pelaku memantau halaman web palsu untuk mengumpulkan data yang dimasukkan oleh korban. Data ini kemudian digunakan untuk melakukan kegiatan ilegal seperti pembelian barang tanpa izin atau penipuan.

Adapun jenis-jenis *phising* adalah sebagai berikut:

1. *Scam Phishing*: Upaya untuk memperoleh informasi pribadi seperti nomor rekening bank, password, dan nomor kartu kredit dengan cara mengelabui korban melalui *email*, SMS, atau media sosial. Pelaku sering mengirimkan link atau file yang mengandung malware atau telah dimodifikasi untuk mencuri informasi.
2. *Blind Phishing*: Jenis serangan phishing yang paling umum, di mana *email* dikirim secara massal tanpa strategi khusus. Pelaku hanya mengandalkan keberuntungan dan menunggu penerima masuk ke dalam jebakan penyerang.
3. *Spear Phishing*: Serangan phishing yang ditujukan kepada kelompok tertentu seperti pejabat pemerintah, pelanggan perusahaan, atau orang-orang tertentu. Tujuannya untuk masuk ke dalam database khusus untuk mencuri informasi rahasia ataupun data-data finansial.
4. *Clone Phishing*: Jenis serangan yang dilakukan dengan menduplikat website asli agar hasil duplikat tersebut mengelabui pengguna. *Website* palsu ini kemudian meminta pengguna agar memasukkan informasi pribadi yang nantinya akan disalahgunakan.

5. *Whaling*: Serangan phishing dengan sasaran eksekutif tingkat tinggi dan tokoh masyarakat, termasuk eksekutif bisnis, dengan tujuan mengganggu lembaga pemerintah. Serangan ini dilakukan dengan menyamar sebagai pejabat pengadilan atau mengumumkan informasi internal perusahaan.
6. *Vishing*: Jenis serangan *phishing* yang menggunakan suara (voice) untuk melakukan serangan serta mencari target. Pelaku sering menggunakan nomor telepon yang palsu atau VoIP untuk merahasiakan identitas mereka.
7. *Pharming*: Serangan *phishing* yang menggunakan DNS spoofing untuk menuntun korban ke halaman tiruan yang ditujukan untuk serangan kepada korban. DNS adalah sistem yang menerjemahkan domain menjadi alamat IP. Jika sistem ini disusupi, pengguna akan diarahkan ke halaman palsu melalui URL yang dimasukkan.
8. *Smishing*: Jenis *phishing* yang dilakukan melalui SMS. Pesan yang dikirimkan biasanya mendesak korban untuk melakukan sesuatu, seperti membayar sejumlah uang atau mengklaim hadiah lotre, dengan tujuan untuk mengelabui dan mencuri informasi pribadi.

2.2. Pesan Teks

Pesan teks merupakan informasi yang dikirimkan dari satu perangkat ke perangkat lain. Pesan teks biasanya bertujuan untuk membagikan informasi, memberikan penawaran, dan masih banyak lagi. Orang atau pengguna yang mengirim pesan disebut dengan *texter* (penulis teks).

Pesan teks adalah bentuk komunikasi tertulis yang biasanya dikirim melalui perangkat telekomunikasi seperti telepon seluler atau komputer. Dalam komunikasi teks, pengguna dapat menyampaikan informasi secara singkat dan langsung, memungkinkan komunikasi yang cepat dan efisien. Keunggulan pesan teks terletak pada kecepatan pengiriman dan kemudahan dalam menyampaikan pesan tanpa perlu berlama-lama.

Komunikasi teks juga memanfaatkan simbol-simbol non-verbal seperti emotikon atau emoji untuk mengekspresikan emosi atau nuansa dalam pesan. Meskipun menggunakan teks tertulis, pengguna dapat dengan mudah menambahkan nuansa atau konteks tambahan melalui simbol-simbol ini. Namun, pesan teks memiliki

keterbatasan dalam format, seperti batasan jumlah karakter atau keterbatasan dalam penggunaan media tambahan seperti gambar atau video.

Pesan teks juga dapat menjadi sarana untuk melakukan serangan phishing, salah satunya melalui metode yang disebut smishing. Smishing adalah serangan phishing yang dilakukan melalui SMS, di mana pelaku mencoba untuk mengelabui korban dengan mengirimkan pesan teks yang mengandung tautan berbahaya atau meminta korban untuk mengirimkan informasi pribadi.

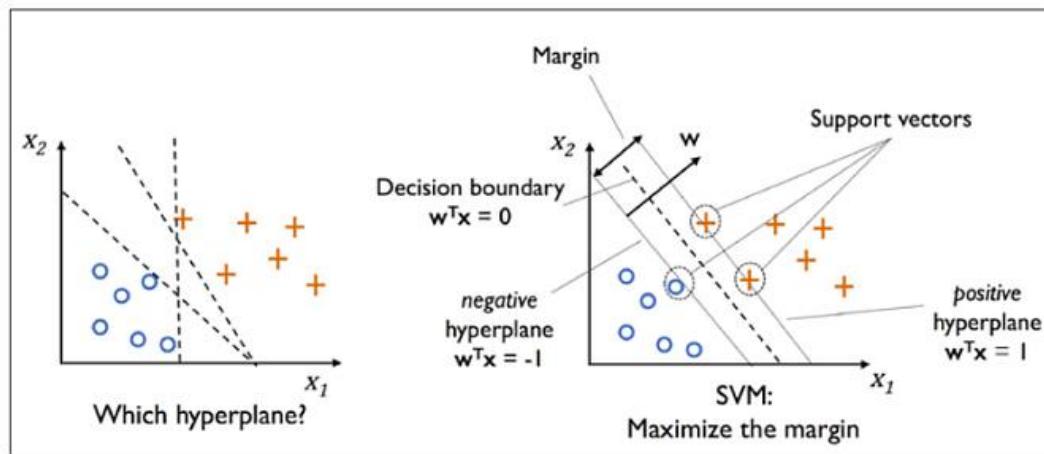
Pelaku smishing seringkali menggunakan pesan teks yang mengecoh, misalnya dengan memberitahu korban bahwa mereka telah memenangkan hadiah atau ada masalah dengan akun mereka yang perlu segera diselesaikan. Tujuan utama dari smishing adalah untuk mengambil informasi sensitif dari korban seperti nomor kartu kredit, password, atau informasi keuangan lainnya dari korban.

Oleh karena itu, penting bagi pengguna untuk selalu waspada terhadap pesan teks yang mencurigakan dan tidak mengklik tautan atau memberikan informasi pribadi tanpa verifikasi yang jelas. Menggunakan aplikasi keamanan yang dapat memfilter pesan teks berbahaya juga merupakan langkah yang bijaksana untuk melindungi diri dari serangan smishing.

2.3. Support Vector Machine

Support vector machine merupakan algoritma yang menganalisis data dan mengidentifikasi pola yang digunakan dalam klasifikasi data. Algoritma ini lebih menekankan pada korelasi hubungan kata. Dengan algoritma tersebut, data akan dibagi menjadi dua kelas yang berbeda.

SVM digunakan untuk mencari *hyperplane* optimal dengan maksud untuk memaksimalkan jarak antara kelas (Darmawan & Fauzan Dianta, 2023). *Hyperplane* sendiri ialah fungsi yang bertujuan untuk memisahkan antara kelas. Dalam dua dimensi, fungsi ini dikenal sebagai garis yang memisahkan kelas, sedangkan dalam tiga dimensi, dikenal sebagai bidang, dan dalam dimensi yang lebih tinggi disebut sebagai hyperplane.



Gambar 2. 1 Hyperplane

Dalam SVM, hyperplane yang ditemukan dapat dilihat seperti Gambar 2.1, terletak di tengah-tengah antara 2 kelas. Hal ini mengindikasikan bahwa jarak antara hyperplane dengan objek-objek data dari kedua kelas berbeda, yang terdekat dengan hyperplane dan diberi tanda bulat kosong, adalah maksimum (Adrian et al., 2021). Dalam konteks SVM, objek data terdekat yang menjadi support vector merupakan yang paling sulit diklasifikasikan karena posisinya yang hampir tumpang tindih dengan kelas lain. Oleh karena itu, hanya *support vector* ini yang dipertimbangkan dalam menemukan hyperplane optimal oleh SVM.

Dalam banyak kasus, data tidak dapat dipisahkan secara linear dalam ruang input. Soft margin SVM tidak dapat menemukan *hyperplane* pemisah yang memungkinkan untuk memiliki akurasi yang tinggi dan generalisasi yang baik dalam kasus ini. Oleh karena itu, diperlukan penggunaan kernel untuk mentransformasikan data ke dalam ruang dimensi yang lebih tinggi yang dikenal sebagai ruang kernel. Ruang kernel ini memungkinkan SVM untuk memisahkan data secara linear dalam dimensi yang lebih tinggi. Beberapa jenis fungsi kernel yang sering digunakan meliputi kernel linear, polynomial, dan *radial basis function* (RBF).

2.4. Ensembled Bagging

Ensemble Bagging Classifier adalah salah satu metode dalam pembelajaran mesin yang menggunakan teknik ensemble untuk meningkatkan kinerja model. Ensemble Bagging Classifier membangun beberapa model pembelajaran mesin yang independen secara paralel dan kemudian menggabungkan hasil prediksi dari model-model tersebut.

Proses ensemble bagging classifier dimulai dengan membuat beberapa subset acak dari data pelatihan dengan penggantian (bootstrap). Setiap subset ini digunakan untuk melatih model pembelajaran mesin yang berbeda. Proses pelatihan dilakukan secara paralel, yang membuat proses ini efisien dalam penggunaan sumber daya komputasi.

Setelah semua model terlatih, hasil prediksi dari setiap model digabungkan untuk menghasilkan prediksi akhir. Dalam klasifikasi, hasil prediksi dapat diambil berdasarkan mayoritas suara (misalnya, modus dari semua prediksi) atau dengan memberikan bobot pada setiap prediksi berdasarkan kepercayaan pada model tersebut. Ensemble Bagging Classifier seringkali menghasilkan kinerja yang lebih baik daripada model tunggal karena mampu mengurangi varians dan overfitting. Dengan menggunakan teknik ini, model dapat lebih baik dalam menggeneralisasi pola dari data yang belum pernah dilihat sebelumnya.

2.5. Cross Validation

Cross validation adalah teknik yang digunakan dalam machine learning dan pemodelan prediktif untuk mengevaluasi kinerja dan generalisasi model. Dalam *cross validation*, data dibagi menjadi subset, yang disebut fold, untuk melakukan pelatihan dan pengujian model secara berulang.

Teknik ini penting karena memanfaatkan data yang terbatas dengan membuat beberapa model untuk estimasi yang lebih akurat. *Cross validation* juga membantu mengatasi overfitting, di mana model terlalu spesifik pada data pelatihan sehingga kurang baik dalam menganalisis data baru (Xu, 2018).

Ada beberapa jenis *cross validation*, termasuk *K-Fold Cross Validation*, *Hold-Out Cross Validation*, *Stratified K-Fold Cross Validation*, *Leave-P-Out Cross Validation*, *Leave-One-Out Cross Validation*, *Monte Carlo (Shuffle-Split)*, dan *Time Series (Rolling Cross-Validation)*. Setiap jenis memiliki kegunaan dan kelebihannya sendiri, tergantung pada kebutuhan dan karakteristik data yang digunakan.

Dengan menggunakan *cross validation*, kita dapat memperoleh estimasi performa model yang lebih akurat dan dapat diandalkan pada data yang tidak terlihat sebelumnya. Ini membantu dalam pengembangan model yang lebih baik dan dapat diandalkan untuk digunakan dalam berbagai aplikasi machine learning dan pemodelan prediktif.

2.6. GridSearch Cross Validation

Grid Search Cross Validation adalah metode yang efektif dalam menemukan parameter optimal untuk model pembelajaran mesin (Kouate, 2020) . Metode ini menggabungkan dua konsep utama, yaitu Grid Search dan Cross Validation, untuk mencapai tujuan ini. Grid Search bekerja dengan melakukan iterasi melalui semua kombinasi parameter yang mungkin, sedangkan Cross Validation digunakan untuk mengevaluasi kinerja model dengan setiap kombinasi parameter tersebut.

Pertama-tama, Grid Search melakukan pencarian parameter terbaik dengan cara yang sistematis. Ini dilakukan dengan membuat "grid" dari parameter yang ingin dioptimalkan, kemudian melakukan iterasi melalui setiap kombinasi parameter. Setiap kombinasi parameter tersebut kemudian digunakan untuk melatih model dan mengevaluasi kinerjanya menggunakan metode Cross Validation.

Kedua, Cross Validation membantu memastikan bahwa model yang dihasilkan dari Grid Search dapat digeneralisasi dengan baik ke data yang belum pernah dilihat sebelumnya. Dengan membagi data menjadi subset dan melakukan pelatihan serta evaluasi pada setiap subset, Cross Validation memberikan perkiraan yang lebih baik tentang seberapa baik model akan berperforma pada data baru.

Dengan menggabungkan Grid Search dan Cross Validation, peneliti dapat meningkatkan kinerja model kita dengan menemukan parameter yang optimal dan memastikan bahwa model tersebut dapat digeneralisasi dengan baik. Ini sangat penting dalam mengembangkan model pembelajaran mesin yang efektif dan dapat diandalkan untuk berbagai aplikasi.

2.7. Website

Website atau situs web merupakan sekumpulan halaman web yang saling terhubung, biasanya tersimpan pada server yang sama, dan dapat diakses melalui internet menggunakan browser (Susilawati et al., 2020) . Website dirancang untuk menyediakan informasi, layanan, atau fitur kepada pengguna, yang dapat berupa teks, gambar, video, dan interaktivitas lainnya. Dalam konteks bisnis, pendidikan, hiburan, atau layanan publik, website berfungsi sebagai portal untuk memperluas jangkauan, meningkatkan keterlibatan, dan memfasilitasi transaksi atau komunikasi.

Sejarah website dimulai pada awal tahun 1990-an, ketika Tim Berners-Lee, seorang ilmuwan komputer Inggris di CERN, mengembangkan HTML (*HyperText*

(Markup Language) sebagai bahasa markup untuk membuat halaman web dan HTTP (*HyperText Transfer Protocol*) sebagai protokol untuk mentransfer data di web. Situs web pertama, yang masih aktif hingga hari ini, diluncurkan pada tahun 1991. Sejak saat itu, teknologi website telah berkembang pesat, dengan pengenalan CSS (*Cascading Style Sheets*) untuk styling halaman, JavaScript untuk interaktivitas, dan berbagai framework dan teknologi backend yang telah meningkatkan dinamika dan fungsionalitas website (Homepage & Aji Prasetyo, 2022).

Pengembangan website melibatkan penggunaan bahasa pemrograman dan teknologi seperti HTML, CSS, JavaScript, dan berbagai framework seperti React, Angular, atau Vue.js untuk frontend. Untuk bagian backend, website dapat menggunakan bahasa seperti Python, PHP, Ruby, atau Java, dan teknologi server seperti Node.js. Penggunaan database seperti MySQL, PostgreSQL, atau MongoDB adalah umum untuk menyimpan data yang kemudian dapat diakses dan dikelola melalui website. Penerapan ini memungkinkan pembuatan website yang tidak hanya informatif tetapi juga interaktif, mendukung aplikasi web yang kompleks dan dinamis.

2.8. HTML

HTML (*Hypertext Markup Language*) merupakan bahasa markup standar yang digunakan untuk menciptakan dan mendesain halaman web. HTML menginstruksikan browser tentang cara menampilkan konten web, termasuk teks, gambar, dan media lainnya (Rozi et al., 2022). Melalui elemen-elemen HTML, pengembang dapat strukturisasi konten dengan judul, paragraf, list, link, dan berbagai komponen interaktif lainnya. Karena fungsinya yang fundamental dalam pembuatan website, HTML menjadi dasar dalam pengembangan web dan merupakan keterampilan esensial bagi setiap pengembang web yang ingin merancang situs web yang efektif dan mudah diakses (Sari et al., 2022).

HTML (*Hypertext Markup Language*) juga memungkinkan integrasi dengan bahasa pemrograman lainnya, seperti CSS (*Cascading Style Sheets*) untuk mengatur tata letak dan tampilan halaman web, serta JavaScript untuk menambahkan interaktivitas dan fungsionalitas dinamis. Dengan menggunakan kombinasi HTML, CSS, dan JavaScript, pengembang dapat menciptakan pengalaman web yang menarik dan responsif bagi pengguna.

Selain itu, HTML juga mendukung pengembangan web yang responsif, artinya halaman web dapat menyesuaikan diri dengan berbagai perangkat dan ukuran layar. Hal ini dapat dilakukan dengan menggunakan fitur-fitur HTML seperti meta tag viewport dan elemen-elemen HTML yang responsif seperti grid dan flexbox. Dengan demikian, pengembang dapat memastikan bahwa pengalaman pengguna tetap optimal, baik di desktop maupun perangkat mobile.

Secara keseluruhan, HTML adalah fondasi dari pengembangan web modern dan menjadi keterampilan yang sangat berharga bagi siapa pun yang tertarik dalam membangun situs web. Dengan pemahaman yang baik tentang HTML, seseorang dapat membuat halaman web yang efektif, mudah diakses, dan menarik bagi pengguna.

2.9. Bootstrap

Bootstrap merupakan salah satu kerangka kerja (framework) front-end yang sedang populer digunakan oleh pengembang web saat ini. Framework ini memungkinkan pengembang untuk dengan cepat membangun tampilan situs web yang responsif dan menarik tanpa harus menulis kode dari awal. Dengan menyediakan kumpulan komponen UI, grid system, dan gaya CSS yang siap pakai, Bootstrap mempercepat proses pengembangan dan memastikan konsistensi dalam tampilan antarmuka pengguna (Sari et al., 2022).

Fitur responsif yang diperkenalkan oleh Bootstrap pada tahun 2012 telah membuatnya semakin diminati oleh pengembang web. Fitur ini memungkinkan tata letak situs web beradaptasi dengan baik pada berbagai perangkat, mulai dari desktop hingga smartphone, tanpa mengorbankan pengalaman pengguna. Dengan demikian, Bootstrap membantu memastikan bahwa situs web yang dikembangkan dengan framework ini dapat diakses dengan baik oleh pengguna yang mengakses melalui perangkat mobile, yang semakin meningkat jumlahnya.

Keunggulan Bootstrap tidak hanya terletak pada kemampuannya dalam menciptakan situs web responsif, tetapi juga dalam kemudahan penggunaannya. Framework ini menyediakan dokumentasi yang komprehensif dan contoh kode yang mudah diikuti, sehingga memungkinkan pengembang pemula untuk memahami dan menggunakan Bootstrap dengan cepat. Hal ini menjadikan Bootstrap pilihan yang

populer bagi pengembang web yang ingin menghasilkan tampilan situs web yang modern dan responsif dengan waktu pengerjaan yang efisien.

2.10. JavaScript

JavaScript ialah bahasa pemrograman yang terintegrasi dalam dokumen HTML dan berfungsi di sisi klien web. Sejak awal sejarah internet, JavaScript telah menjadi bahasa pemrograman utama untuk pengembang web. JavaScript memberikan kemampuan tambahan kepada HTML dengan memungkinkan eksekusi perintah di sisi pengguna, yaitu di browser, bukan di server web. Dengan JavaScript, pengembang web dapat membuat situs web yang dinamis dan interaktif, dengan kemampuan untuk berinteraksi dengan elemen HTML dan CSS, serta melakukan validasi formulir dan manipulasi data.

Salah satu keunggulan utama JavaScript adalah kemampuannya untuk berjalan di sisi klien, atau di browser pengguna, bukan di server web. Hal ini memungkinkan aksi yang cepat dan responsif tanpa perlu mengirim permintaan ke server, yang pada akhirnya meningkatkan pengalaman pengguna. Sejak awal sejarah internet, JavaScript terus berkembang dan menjadi lebih kuat dengan dukungan berbagai library dan framework, seperti React, Angular, dan Vue, yang meningkatkan fungsionalitas dan mempermudah pengembangan aplikasi web yang rumit.

JavaScript juga memiliki keunggulan dalam hal fleksibilitas dan adaptabilitas. Bahasa pemrograman ini dapat digunakan untuk mengembangkan berbagai jenis aplikasi web, mulai dari aplikasi sederhana hingga yang sangat kompleks. Selain itu, JavaScript juga mendukung pengembangan aplikasi lintas platform, yang memungkinkan pengembang untuk membuat aplikasi yang dapat berjalan di berbagai sistem operasi dan perangkat. Dengan kombinasi kemudahan penggunaan, responsifitas, dan fleksibilitasnya, JavaScript tetap menjadi salah satu bahasa pemrograman yang paling dominan dan relevan dalam dunia pengembangan web.

2.11. Python

Python adalah bahasa pemrograman yang dapat mengeksekusi instruksi secara langsung (interpretatif) dengan menggunakan paradigma pemrograman berorientasi objek (Object Oriented Programming) dan memiliki semantik dinamis yang meningkatkan keterbacaan syntax (Romzi & Kurniawan, 2020). Python sering

dianggap sebagai bahasa yang memiliki kemampuan tinggi, menggabungkan kemampuan yang kuat dengan sintaksis yang jelas, serta didukung oleh berbagai pustaka standar yang luas dan komprehensif. Meskipun termasuk dalam kategori bahasa pemrograman tingkat tinggi, Python dirancang agar mudah dipelajari dan dipahami.

Python memiliki fitur menarik yang membuatnya layak untuk dipelajari. Salah satunya, Python memiliki tata bahasa dan skrip yang sangat mudah dipelajari. Selain itu, Python juga dilengkapi dengan sistem pengelolaan data dan memori otomatis. Modul dalam Python juga selalu diperbarui secara berkala. Selain itu, Python juga memiliki banyak fasilitas pendukung. Bahasa pemrograman Python dapat dijalankan pada berbagai sistem operasi seperti Linux, Microsoft Windows, Mac OS, Android, Symbian OS, Amiga, Palm, dan lainnya.

Python diciptakan dan dikembangkan oleh Guido Van Rossum, seorang programmer asal Belanda, di kota Amsterdam pada tahun 1990. Pada tahun 1995, Python mengalami pengembangan lebih lanjut untuk meningkatkan kompatibilitasnya oleh Guido Van Rossum. Kemudian, pada awal tahun 2000, Python mengalami beberapa pembaruan versi hingga mencapai Versi 3, yang masih digunakan hingga saat ini. Nama "Python" sendiri diambil dari acara televisi populer yang disukai oleh Guido van Rossum, yaitu "Monty Python's Flying Circus".

Python diminati karena dianggap mudah dipelajari, bahkan oleh pemula. Kode Python mudah dibaca dan mampu mengeksekusi fungsi-fungsi kompleks karena adanya banyak library standar yang dapat diakses dengan mudah. Pengembangan program menggunakan Python bisa dilakukan dengan cepat dan menggunakan kode yang lebih sedikit, bahkan untuk program dengan skala yang sangat rumit. Python juga mendukung multi-platform dan memiliki sistem pengelolaan memori otomatis yang serupa dengan Java.

Namun, Python memiliki kelemahan dalam hal kecepatan eksekusi yang cukup lambat. Python juga kurang mendukung pengembangan untuk platform Android dan iOS. Selain itu, Python memiliki keterbatasan dalam akses ke basis data dan tidak cocok untuk tugas-tugas yang membutuhkan penggunaan memori intensif atau pemrosesan *multi-core/multi-processor*.

2.12. Flask

Flask adalah kerangka kerja aplikasi web ringan yang ditulis dalam bahasa pemrograman Python, dirancang untuk membuat aplikasi web dengan cepat dan mudah, fokus pada kesederhanaan dan fleksibilitas. Konsep dasar Flask meliputi routing, pengguna dapat menentukan bagaimana aplikasi web merespons permintaan dari pengguna berdasarkan URL yang diminta, menggunakan dekorator Python. Templating juga penting, memungkinkan pemisahan tampilan dari logika aplikasi, dengan mesin template Jinja2 yang memungkinkan pembuatan file template HTML dengan struktur dasar halaman web dan penggunaan variabel, loop, dan kontrol alur untuk tampilan dinamis.

Flask menyediakan berbagai objek yang memudahkan penanganan permintaan HTTP, seperti objek request untuk mengakses data yang dikirim oleh pengguna dan objek session untuk menyimpan informasi sesi pengguna antar permintaan. Respons juga dapat diatur dengan berbagai macam jenis respons yang dapat dikembalikan oleh aplikasi, seperti teks sederhana, HTML, JSON, atau file statis, menggunakan objek response.

Flask juga mendukung penggunaan ekstensi, memungkinkan penambahan fungsionalitas tambahan ke aplikasi, seperti Flask-SQLAlchemy untuk mengakses database, Flask-WTF untuk mengelola formulir web, dan Flask-Login untuk otentikasi pengguna.

2.13. Confusion Matrix

Confusion matrix adalah salah satu alat evaluasi yang sering digunakan dalam klasifikasi dalam pembelajaran mesin untuk menggambarkan kinerja model. Confusion Matrix merupakan alat ukur berbentuk matriks yang digunakan untuk memperoleh keakuratan klasifikasi kelas secara keseluruhan karena algoritma yang digunakan (Qadrini et al., 2021). Bentuk sederhana confusion matrix dapat dilihat pada tabel 2.1.

Tabel 2. 1 Confusion Matrix Dua Kelas (Qadrini et al., 2021)

Confusion Matrix		Nilai Sebenarnya	
		True	False
Nilai	True	TP (<i>True Positif</i>) <i>Correct result</i>	FP (<i>False Positive</i>) <i>Unexpected result</i>
	False	FN (<i>False Negative</i>) <i>Negative</i>) <i>Missing result</i>	TN (<i>True Negatif</i>) <i>Correct Absence of result</i>

Confusion matrix terdiri dari empat komponen utama, yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). True Positive adalah jumlah kasus di mana model dengan benar memprediksi kelas positif. True Negative adalah jumlah kasus di mana model dengan benar memprediksi kelas negatif. False Positive adalah jumlah kasus di mana model salah memprediksi kelas positif padahal sebenarnya adalah negatif. False Negative adalah jumlah kasus di mana model salah memprediksi kelas negatif padahal sebenarnya adalah positif.

Dengan menggunakan confusion matrix, kita dapat menghitung beberapa metrik evaluasi penting seperti akurasi, presisi, recall, dan F1-score. Akurasi menunjukkan proporsi prediksi yang benar terhadap total prediksi. Presisi mengukur seberapa banyak dari prediksi positif yang benar-benar positif. Recall mengukur seberapa banyak dari total kasus positif yang berhasil diidentifikasi dengan benar oleh model. F1-score adalah rata-rata harmonis dari presisi dan recall, memberikan gambaran keseimbangan antara keduanya.

Perhitungan nilai akurasi, presisi, recall, dan F1-score dapat diuraikan sebagai berikut :

$$Akurasi = \frac{True\ Positives+True\ Negatives}{Total\ Population} \quad (2.1)$$

$$Precision = \frac{True\ Positives}{True\ Positives+False\ Positives} \quad (2.2)$$

$$Recall = \frac{True\ Positives}{True\ Positives+False\ Negatives} \quad (2.3)$$

$$F1_{ham} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (2.4)$$

2.14. Peneliti Terdahulu

Pada penelitian sebelumnya yang dilakukan oleh Purwanto et.al. (2020) penelitian ini berfokus pada sebuah algoritma baru berbasis kompresi untuk mendeteksi situs web *phising* yang dinamakan *PhishZip*. Dibandingkan dengan pendekatan pembelajaran mesin standar untuk klasifikasi, metode berbasis kompresi lebih mudah diterapkan dan tidak memerlukan pra pemrosesan. Phishzip ini menggunakan pustaka zlib, yang didasarkan pada algoritma kompresi *DEFLATE*. *DEFLATE* adalah algoritma kompresi data yang digunakan untuk mengompresi situs web dalam *HTTP*. Algoritma *DEFLATE* akan mengkompresi 2 kamus yang berisikan kata-kata yang umum digunakan dalam situs *web phising* dan *non-phising*. Dengan demikian, kata yang lebih sering muncul akan diberi nilai kemungkinan yang lebih tinggi. Setelah pengkompresian dilakukan maka akan keluar keputusan apakah situs *web* diklasifikasikan sebagai *phising* atau *non-phising*. Dengan melakukan klasifikasi menggunakan algoritma kompresi, algoritma ini dapat mendeteksi situs *web phising* dengan tingkat positif sejati sebesar 80,04%, tingkat positif palsu 18,25%, dan akurasi sebesar 80,89%.

Terdapat juga penelitian terdahulu yang dilakukan oleh Espinoza et.al. (2019). Penelitian ini bertujuan untuk mendeteksi phising yang terjadi dalam email menggunakan algoritma pembelajaran mesin. Data set yang digunakan ialah dataset yang telah diekstraksi dari PhisTank Dataset. Algoritma yang digunakan dalam penelitian ini ialah penggabungan antara Decision Tree dan Naive Bayes. Untuk email yang terinfeksi phising, akan melewati beberapa prosedur yaitu, Email akan dikirim ke database yang menyimpan fitur, dimana 18 keanehan dari setiap email diekstraksi dan matriks data yang disimpan dalam sebuah file berekstensi CSV. Matriks ini dikirim bersama data Naive Bayes Classifier untuk pelatihan model. Jika hasil tersebut dibawah 95%, maka pemeriksaan ulang akan dilakukan dengan menggunakan Decision Trees Classifier. Pengklasifikasian ini akan mengirimkan pesan respons dengan persentasi prediksi yang dicapai oleh algoritma pelatihan. Lalu jika email telah terinfeksi phising, maka email tersebut akan dikirim ke email karantina. Lalu pada tahap akhir, proses sebelumnya akan mengambilkan sebuah pesan yang memuat hasil dari setiap email yang dianalisa. Dengan menggunakan penggabungan ini, Pengklasifikasian phising mendapatkan tingkat positif yang benar sebesar 95,15%,

negatif yang sebenarnya sebesar 96,83%, tingkat kesalahan sebesar 3,21%, dan akurasi sebesar 96,78%.

Penelitian sebelumnya yang dilakukan oleh Aung et.al. (2021). Penelitian ini dilakukan untuk mendeteksi penipuan kartu kredit berdasarkan metrik kinerja. Penelitian ini menggunakan algoritma Random Forest. Dataset yang digunakan pada penelitian ini merupakan dataset yang tidak seimbang, oleh karena penyeimbangan dataset dilakukan dengan menggunakan Synthetic Minority Oversampling Technique (SMOT). Dataset yang digunakan berjumlah 2 dimana dataset pertama berisi transaksi yang terjadi dalam 2 hari. Dataset ini berisi 31 fitur numerik, dan salah satu nya memiliki label yang jika berinilai 1 maka terjadi penipuan, sedangkan 0 jika tidak terjadi penipuan. Total data yang dimiliki ialah 492 penipuan dari 284.807 data. Dataset kedua merupakan dataset Jerman dimana memiliki 300 data penipuan dan 700 data tidak penipuan. Penelitian menggunakan WEKA dan Python sebagai sarana mengolah datanya. Kedua dataset akan melewati tahap pra-pemrosesan yang akan menormalkan data yang ada. Setelah data dinormalkan, dataset akan dilatih dengan beberapa algoritma pembelajaran mesin seperti, Random Forest, KNN, SVM, Naive Bayes, dan Regresi Logistik. Dalam penggunaan Pyhton pada dataset pertama, Algoritma Random Forest mendapatkan hasil yang lebih baik dalam jumlah instance dibanding dengan 4 algoritma lainnya sebanyak 85.398. Pada dataset kedua, Random Forest juga mendapatkan hasil pengklasifikasian dengan benar terbanyak sebesar 228 data. Pada dataset 1 dengan menggunakan WEKA, Random forest menjadi algoritma dengan hasil terbaik yang mendapatkan tingkat akurasi sebesar 0.9998. Lalu pada dataset 2 mendapatkan tingkat akurasi sebesar 0.76.

Terdapat penelitian terdahulu yang dilakukan oleh Sakib et.al. (2021). Penelitian ini bertujuan untuk memprediksi jenis kelahiran apakah kelahiran itu persalinan normal atau operasi caesar. Dataset yang digunakan ialah dataset yang berasal dari studi kasus di negara Bangladesh. Dalam penelitian ini, peneliti mengidentifikasi faktor-faktor penting yang berpengaruh terhadap persalinan secara caesar. Dataset dibagi menjadi 2 bagian yaitu 80% untuk data pelatihan dan 20% untuk data pengujian. Peneliti menggunakan ukuran evaluasi seperti Recall, Presisi, dan F1 Score, dan akurasi. Lalu peneliti menggunakan Receiver Operating Characteristic (ROC) untuk menunjukkan peforma model. Setelah melakukan proses

pengklasifikasian, terlihat bahwa algoritma dengan ensembled bagging mengalami peningkatan akurasi. Algoritma decision tree mendapatkan akurasi sebesar 0.85 dan decision tree dengan metode bagging mendapatkan akurasi sebesar 0.87. Algoritma KNN mendapatkan akurasi 0.79, sedangkan KNN-Bagging mendapatkan nilai 0.86. Pada algoritma SVM tidak mengalami kenaikan, algoritma ini mendapat akurasi 0.86. Lalu untuk Naive Bayes mendapatkan akurasi sebesar 0.80 dan Naive Bayes-Bagging mendapatkan akurasi 0.84.

Penelitian selanjutnya yang dilakukan oleh Espinoza et.al. (2020). Penelitian ini bertujuan untuk mendeteksi ulasan palsu pada ulasan restoran menggunakan ensemble learning. Pendekatan berbasis pembelajaran mesin sudah dilakukan dan dapat mendeteksi ulasan yang bersifat menipu ataupun palsu. Dataset yang digunakan berasal dari data yang dikumpulkan dari sebuah restoran dimana berisi ulasan palsu dari tiga restoran. Untuk mengembangkan klasifikasi yang baik, peneliti menggunakan 3 pendekatan ensemble yaitu boosting, bagging, dan stacking. Klasifikasi ini menggunakan pembelajaran mesin pohon keputusan, random forest, SVM, Extreme Gradient-Boosting Trees, dan multilayer perceptron. Dari pengklasifikasian yang dilakukan metode ensemble diterapkan pada algoritma SVM dan MLP. Hasil akhir pada klasifikasi menunjukkan bahwa algoritma dengan bagging ensemble memiliki tingkat kesalahan pelatihan yang lebih sedikit, dan juga tingkat kesalahan uji yang lebih rendah. Algoritma SVM mendapat kesalahan pelatihan sebesar 0.239 dan kesalahan uji data sebesar 0.409. Sedangkan SVM dengan bagging ensemble memiliki kesalahan pelatihan sebesar 0.227 dan kesalahan uji data sebesar 0.318.

Penelitian selanjutnya yang dilakukan oleh Shukla et.al. (2020). Penelitian ini bertujuan untuk mendeteksi phising dalam url menggunakan metode pembelajaran mendalam yang dikombinasikan dengan algoritma tradisional. Penelitian ini menggunakan dataset berita dari Reuters dan dataset lainnya untuk mengevaluasi kinerja model yang diusulkan. Metode yang dikembangkan melibatkan penggunaan arsitektur LSTM untuk mengekstraksi fitur dari teks, yang kemudian digunakan sebagai input untuk algoritma klasifikasi tradisional seperti SVM dan Naive Bayes. Prosesnya dimulai dengan preprocessing teks, termasuk tokenisasi, penghapusan

stopwords, dan stemming. Setelah preprocessing, teks tersebut diubah menjadi representasi vektor menggunakan teknik embedding seperti Word2Vec. Vektor-vektor ini kemudian dimasukkan ke dalam model LSTM untuk pelatihan. Hasil dari LSTM digunakan sebagai fitur tambahan untuk algoritma SVM dan Naive Bayes. Dengan kombinasi tersebut, model berhasil mendapatkan akurasi sebesar 95.1%.

Selain itu, penelitian yang dilakukan oleh Aich et.al.(2018), melakukan penelitian yang bertujuan untuk mendeteksi spam pada teks menggunakan berbagai algoritma pembelajaran mesin. Penelitian ini menggunakan dua dataset, yaitu dataset JSC dan UCI, untuk mengevaluasi kinerja model yang diusulkan. Metode yang dikembangkan melibatkan penggunaan algoritma seperti Adaboost, Bagging, J48, dan SVM. Prosesnya dimulai dengan preprocessing teks dan dilanjutkan dengan penyeimbangan dataset menggunakan teknik SMOTE. Algoritma yang digunakan kemudian diterapkan pada dataset yang telah diproses untuk pelatihan dan pengujian. Hasil penelitian menunjukkan bahwa SVM memberikan performa terbaik dalam menangani dataset yang tidak seimbang, dengan Recall (SPAM) sebesar 84.1 dan Recall (HAM) sebesar 99 pada dataset JSC, menghasilkan rata-rata akurasi kelas sebesar 90.94%. Penelitian ini menyimpulkan bahwa penggunaan SVM sangat efektif untuk deteksi spam pada teks, terutama pada dataset yang tidak seimbang.

Adapun penelitian lain berjudul “*Application of Bagging Ensemble Classifier based on Genetic Algorithm in the Text Classification of Railway Fault Hazards*” Li et.al. (2019). Penelitian ini bertujuan untuk mengklasifikasikan bahaya kesalahan kereta api menggunakan algoritma klasifikasi ensambel berbasis Bagging yang dioptimalkan dengan Algoritma Genetika. Dataset yang digunakan dalam penelitian ini adalah data teks yang berisi informasi berharga terkait keselamatan kereta api, yang diubah menjadi vektor menggunakan metode TF-IDF. Algoritma yang digunakan adalah penggabungan antara Decision Tree sebagai model dasar dan Bagging Ensemble Classifier untuk meningkatkan akurasi klasifikasi. Pada langkah pertama, TF-IDF digunakan untuk mengekstraksi fitur teks dan mengonversinya menjadi vektor. Selanjutnya, Decision Tree digunakan untuk mengklasifikasikan data. Bagging Ensemble Classifier kemudian melakukan pelatihan sampel acak pada vektor teks yang dihasilkan oleh TF-IDF. Penggunaan TF-IDF dan algoritma decision tree

mendapatkan *precision* sebesar 70.12%, *recall* sebesar 67.45%, dan *F1-Score* sebesar 68.76%. Lalu untuk TF-IDF dan menggunakan bagging ensemble clasifier mendapatkan *precision* sebesar 82.91%, *recall* sebesar 88.37%, dan *F1-Score* sebesar 85.55%. Dengan penggunaan bagging ensamble clasifier, klasifikasi pada model mengalami kenaikan yang signifikan dalam akurasi dan keandalan.

Tabel 2. 2 Penelitian Terdahulu

No	Nama	Metode	Keterangan
1	Rizka Purwanto, PhisZip dan kompresi Arindam Pal, Alan DEFLATE Blair, Sanjay Jha (2020)	PhisZip dan kompresi DEFLATE	Membuat algoritma bernama PhisZip yang mengompresi kamus (kata kunci) dengan algoritma DEFLATE. Hasil akhir mendapatkan akurasi sebesar 80,89%.
2	Bryan Espinoza, Je'ssica Simba, Walter Fuertes, Eduardo Benavides, Roberto Andrade, Theofilos Toulkeridis (2019)	Decision Tree dan Naive Bayes	Mendeteksi serangan phising pada email dengan menggabungkan 2 algoritma. Tahap pertama data dilatih dengan Naive bayes. Lalu jika hasil belum mencapai 95% maka data akan dilatih lagi menggunakan Decision tree sehingga email yang terdeteksi phising akan dikarantina. Akurasi yang didapatkan sebesar 96,78%.
3	Maung Hein Aung, Random Forest, Penelope Tane Seluka, penormalan SMOT, Jean Tiana Rose Fuata, Maria Josephine WEKA dan Python, Tikoisuva, Matalita Sereinana Cabelawa,	Random Forest, SMOT, WEKA dan Python, serta perbandingan dengan algoritma	Mendeteksi penipuan kartu kredit berdasarkan metrik kinerja. Dataset dinormalkan menggunakan SMOT lalu dilatih dengan 2 software yang berbeda yaitu WEKA

Tabel 2.2. Penelitian Terdahulu (Lanjutan)

No	Nama	Metode	Keterangan
	Ravneil Nand (2021)	pembelajaran mesin lain.	dan python. Dalam pelatihan data menggunakan Python, Random forest mendapatkan hasil terbaik dengan memiliki jumlah pengklasifikasian benar terbanyak yaitu 85.401 dan 223. Saat menggunakan WEKA, Random forest juga mendapatkan akurasi tertinggi yaitu sebesar
4	Md. Sakib Bin Alam, Muhammed J. A. Patwary, Maruf hassan (2021)	Decision Tree, KNN, SVM, Naive Bayes, dan algoritma pembelajaran mesin menggunakan ensamble bagging.	Memprediksi persalinan dengan menggunakan ensemble bagging sebagai peningkatan peforma algoritma pembelajaran mesin yang lain. Algoritma dengan ensembled bagging mengalami peningkatan akurasi. Algoritma decision tree mendapatkan akurasi sebesar 0.85 dan decision tree dengan metode bagging mendapatkan akurasi sebesar 0.87.
5	Luis Gutierrez- Espinoza, Faranak Abri, Akbar Siami Namin, Keith S. Jones, dan David R. W. Sears.	SVM, MLP, Random Forest, Decision Tree, Ensamble Learning.	Mendeteksi ulasan palsu pada restoran dengan menggunakan algoritma pembelajaran mesin yang didukung oleh ensamble bagging. Pada akhir proses pengklasifikasi algoritma

Tabel 2.2. Penelitian Terdahulu (Lanjutan)

No	Nama	Metode	Keterangan
6	Shrishti Shukla dan LSTM, Word2Vec, Pratyush Sharma (2020)	svm dan naive bayes	yang didukung dengan ensamble bagging memiliki tingkat kesalahan pelatihan yang lebih kecil yakni 0.227 dari yang sebelumnya 0.239, dan tingkat kesalahan uji menjadi 0.318 dari yang sebelumnya 0.409.
7	Payal Aich, Manju Venugopalan, dan Deepa Gupta (2018)	SMOTE dan SVM	Mendeteksi phising dalam URL dengan menggabungkan arsitektur LSTM dan algoritma tradisional. Teks diprepareses dan diubah menjadi vektor menggunakan Word2Vec, kemudian dilatih dengan LSTM. Hasilnya digunakan sebagai fitur untuk SVM dan Naive Bayes, menghasilkan akurasi sebesar 95.1%.
			Mendeteksi spam pada teks menggunakan berbagai algoritma pembelajaran mesin. Teks diprepareses dan dataset diseimbangkan dengan SMOTE. SVM memberikan performa terbaik dengan Recall (SPAM) 84.1 dan Recall (HAM) 99 pada dataset JSC, dengan rata-rata akurasi 90.94%.

Tabel 2.2. Penelitian Terdahulu (Lanjutan)

No	Nama	Metode	Keterangan
8	LI Xinqin, LI Ping, SHI Tianyun, dan ZHOU Wen (2019)	TF-IDF, Tree, Ensamble Clasifier	Mendeteksi bahaya kesalahan kereta api menggunakan algoritma klasifikasi ensambel Bagging yang dioptimalkan dengan Algoritma Genetika. Data teks diubah menjadi vektor dengan TF-IDF dan diklasifikasikan menggunakan Decision Tree dan Bagging Ensemble. Hasil menunjukkan TF-IDF + Decision Tree: precision 70.12%, recall 67.45%, F1-Score 68.76%. TF-IDF + Bagging Ensemble: precision 82.91%, recall 88.37%, F1-Score 85.55%. Bagging Ensemble meningkatkan akurasi dan keandalan klasifikasi.

Perbedaan penelitian ini dengan penelitian yang telah dilakukan sebelumnya ialah, penelitian ini akan mengidentifikasi *phising pada* pesan teks. Penelitian ini juga menggunakan *ensamble learning* yaitu *ensamble bagging* untuk mendukung algoritma yang digunakan, yakni *support vector machine*. Dataset yang digunakan merupakan dataset yang digabung dari beberapa sumber, yaitu dari mendeley, artikel penelitian sebelumnya dan pengumpulan manual pada kotak pesan. Lalu hasil penelitian ini dapat melakukan pelabelan otomatis terhadap data pesan teks dalam bentuk .csv dimana pesan teks tersebut dikumpulkan dalam 1 file namun tidak memiliki label.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Data yang digunakan

Dataset yang digunakan dalam penelitian ini merupakan dataset yang berisi kumpulan pesan teks normal (ham) dan pesan teks phising (smishing) . Dataset ini diambil dari beberapa sumber yaitu penelitian, artikel, dan pengumpulan pada kotak sms. Penelitian yang telah dilakukan sebelumnya oleh, Sandhya Mishra, dengan judul “*Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis*”. Dataset ini dikumpulkan dengan cara mengubah gambar yang diperoleh dari internet menjadi teks menggunakan kode *python*. Dataset yang didapat dari penelitian ini berbahasa inggris yang peneliti ubah (translate) ke bahasa indonesia agar fungsi sistem yang akan dibangun berguna bagi masyarakat indonesia. Data yang telah ditranslate akan digunakan sebagai bahan penelitian.

Dataset yang akan digunakan memiliki label, dimana label ini akan memberi tahu apakah pesan tersebut termasuk pesan normal (ham) ataupun pesan phising (smishing). Pesan-pesan tersebut mengandung beberapa atribut seperti link (url), nomor telepon, ataupun alamat email yang bersifat phising dan tidak. Dataset ini berjumlah 5.971 data teks yang diberi label pesan normal (ham), spam, maupun pesan phising (Smishing). Dataset tersebut mencakup 489 pesan spam, 638 pesan phising (smishing), dan 4844 pesan normal (ham). Dari dataset ini peneliti menggunakan 1000 data dimana berisikan 638 pesan phising (smishing) dan 362 pesan normal (ham).

Selanjutnya, dataset juga dikumpulkan dari artikel yang ditulis oleh Wibisono, (2018). Artikel ini berisikan dataset yang memiliki 3 kelas yaitu, pesan teks normal, pesan teks penipuan, dan pesan teks promosi. Penulis memilah data dari dataset tersebut sehingga beberapa data dapat digunakan sebagai bahan penelitian. Lalu, penulis juga mengumpulkan beberapa pesan teks normal dan pesan teks phising dari kotak sms peneliti. Dari artikel dan pengumpulan pesan pada kotak pesan, dataset yang didapatkan berjumlah 600 data. Lalu dataset digabung menjadi satu dengan total data 1600, dimana pesan normal (ham) sebanyak 645 data dan pesan phising

(smishing) sebanyak 955 data. Dataset ini yang akan digunakan peneliti sebagai dasar penelitian untuk mengidentifikasi phising pada pesan teks.

Tabel 3. 1 Dataset Pesan Teks Normal dan Phising

Label	Teks
Smishing	Mohon untuk tetap berada di rumah. Untuk mendorong warga agar tetap berada di rumah, setiap warga berhak mendapatkan dana darurat sebesar 305.96 atau lebih
Smishing	ANDA TELAH MENANG! Sebagai pelanggan Vodafone yang kami hargai, komputer kami telah memilih ANDA untuk memenangkan hadiah sebesar £150. Untuk mendapatkannya mudah sekali. Cukup hubungi 09061743386
Smishing	Apple ID:[BUXCX7GBVwWCcOD Notifikasi Terakhir untuk Apple 1D anda akan berakhir hari ini. Cegah hal ini dengan mengkonfirmasi Apple ID anda di http://verifyapple.uk . Apple Inc
Smishing	Ini adalah kali kedua kami mencoba untuk menghubungi anda. Anda telah memengangkan hadiah sebesar 750 Poundsterling. Untuk mengeklaim sangat mudah, telepon 08712101359 SEKARANG! Hanya 10p per menit. BT-tarif-nasional
Smishing	Mohon HUBUNGI 08712402902 segera. Ada pesan penting menanti anda.
Smishing	Untuk pemegang voucher, untuk mengeklaim tawaran minggu ini. Mohon kunjungi http://www/wtlp.co.uk/text melalui PC. SnK berlaku
Ham	Nomorku di LUTON 0125698789. Telepon aku kalau kamu di sini!
Ham	Sebenarnya aku sudah menghapus laman lamaku.. Sekarang aku blogging di magicalsongs.blogspot.com
Ham	Transaksi NEFT dengan nomor referensi 456367 sebesar Rs.9000 telah diberikan ke rekening penerima pada 5 Januari jam 3 siang.
Ham	Anda telah mendaftarkan Shinco sebagai pembayar. Masuk ke icicibank.com dan masukkan URN untuk mengonfirmasi. Hati-hati penipuan. Jangan mengirimkan atau memberitahukan URN kepada siapapun

Tabel 3.1 Dataset Pesan Teks Normal dan Phising(lanjutan)

Label	Teks
Ham	Biarkan aku saja yang melakukannya. Memangnya kamu mau bawa? Berat sekali. Apakah nomor 98321561 familiar denganmu?
Ham	Akun anda telah berhasil diisi ulang sebesar INR 100,00. Saldo akun prabayar KeralaCircle anda sekarang Rs 101,00. ID Transaksi anda KR 3567

Selanjutnya dilakukan pembagian dataset dari dataset yang telah dipilih menjadi 2 bagian, yaitu data training dan data testing. Pembagian dataset memiliki proporsi 70:30, dimana 70% untuk data training dan 30% untuk data testing. Detail pemecahan dataset terdapat pada tabel 3.2.

Tabel 3. 2 Pemecahan Dataset

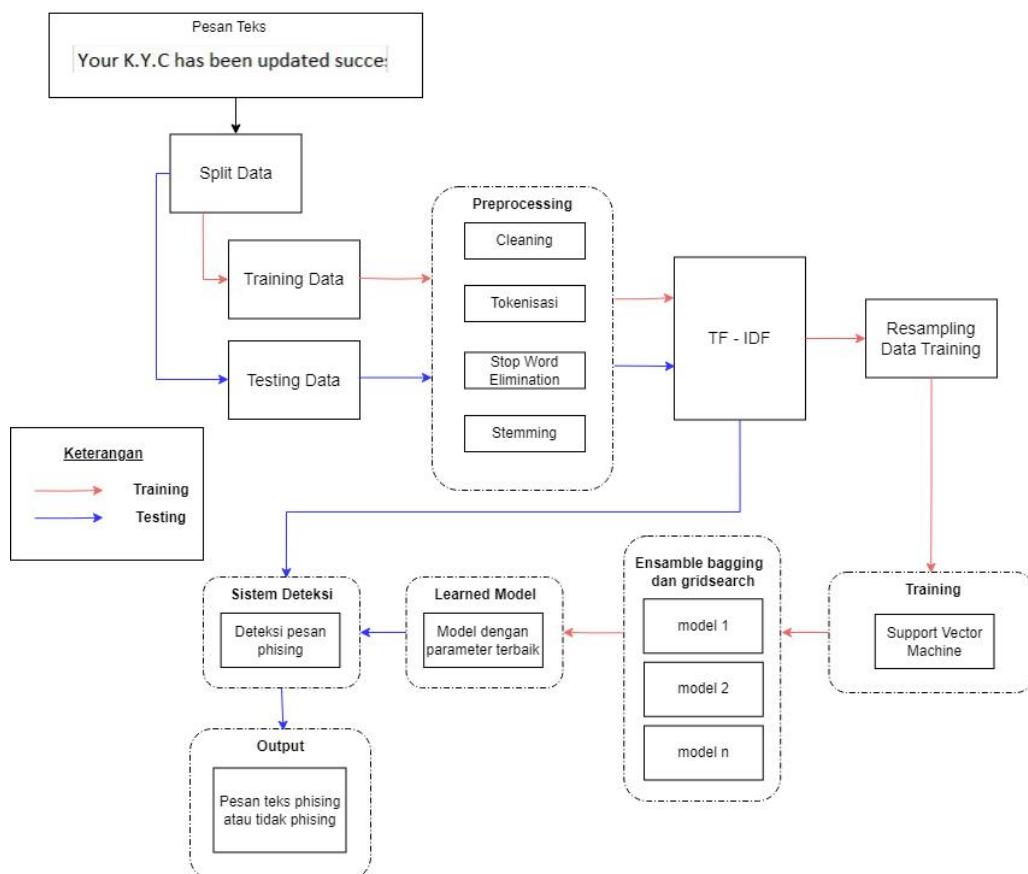
Data Training	Data Testing
1119	481

3.2. Arsitektur Umum

Teknik yang digunakan penulis pada penelitian ini memiliki beberapa tahap, yakni : *preprocessing*, dimana pesan teks dalam dataset tersebut akan diolah melalui beberapa proses guna meningkatkan peforma model dalam mengenali dataset. Tahap *Preprocessing* terbagi menjadi 4 bagian, yaitu *data cleaning*, *tokenisasi*, *stop word elimination*, dan *stemming*. Setelah melalui proses ini, dataset akan diproses dalam tahap TF-IDF (Term Frequency - Inverse Document Frequency), dimana tahap ini akan mengubah teks menjadi numerik berdasarkan bobot seberapa sering dan jarangnya kata pada teks tersebut muncul.

Selanjutnya, data yang telah dipilih, disimpan dalam bentuk .csv serta sudah melewati tahap *preprocessing* akan dibagi menjadi 2 bagian , yaitu *data training* dan *data testing*. Penggunaan *data training* ditujukan sebagai data untuk melatih model machine learning dengan algoritma yang telah dipilih penulis. Lalu, Setelah model berhasil terbentuk dengan baik, model tersebut akan diuji dengan menggunakan *data testing*.

Untuk tahap berikutnya, dilakukan *resampling* pada *data training* sebelum melakukan pelatihan model. Resampling dilakukan untuk mengatasi ketidakseimbangan data pada dataset yang digunakan. Jumlah salah satu kelas yang lebih sedikit akan ditingkatkan melalui tahap ini agar model tidak cenderung lebih baik dalam memprediksi kelas lainnya yang memiliki jumlah lebih banyak. Terakhir, *data training* yang telah di *resampling* akan digunakan untuk pelatihan model menggunakan algoritma *Support Vector Machine*. Algoritma *svm* akan dibantu dengan teknik *ensemble bagging* untuk meningkatkan kinerja model yang akan dihasilkan. Lalu saat model telah berhasil dilatih dengan *support vector machine* dan menggunakan bantuan *ensemble bagging*, model akan diuji dengan menggunakan *data testing* yang telah dibagi sebelumnya. Keluaran dari penelitian ini adalah pengidentifikasiannya bahwa pesan teks tersebut termasuk pesan teks phising (smishing) dan pesan teks normal (ham). Arsitektur umum dari proses penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Arsitektur Umum

3.2.1. *Split Dataset*

Dataset akan dibagi menjadi 2 bagian yaitu training data dan testing data. Dataset akan dibagi dengan teknik holdout method. Dengan Teknik ini, dataset akan dibagi menjadi rasio 70:30. Dimana rasio 70 untuk data training dan rasio 30 untuk data testing. Training data merupakan data yang akan digunakan untuk melatih algoritma yang digunakan. Sedangkan data testing merupakan dataset yang disediakan untuk menguji apakah algoritma yang telah dilatih dapat mengidentifikasi data testing dengan benar setelah dilatih terlebih dahulu oleh training data.

3.2.2. *Pre Processing*

Pre-processing atau pra-pemrosesan dilakukan untuk mengklasifikasikan data dan untuk memudahkan pemrosesan data. Tahap ini terdiri dari 4 langkah yakni, *cleaning*, tokenisasi, *stop word elimination*, dan *stemming*.

3.2.2.1. *Cleaning*

Tahap cleaning merupakan tahap awal dalam preprocessing yang bertujuan untuk membersihkan teks dari elemen-elemen yang tidak relevan atau mengganggu dalam analisis. Dalam implementasi ini, digunakan fungsi `preprocess_text(text)` yang menerima input teks dan mengembalikan teks yang telah dibersihkan. Fungsi ini melakukan beberapa tindakan, seperti menghapus tag HTML, menghapus angka, menghapus spasi ekstra, dan mengonversi teks ke huruf kecil. Proses cleaning pada data dapat dilihat melalui *pseudocode* yang ada pada gambar 3.2.

```

FOR EACH row IN dataframe:
    SET row['Preprocess'] TO EMPTY STRING
    FOR EACH character IN row ['comment']:
        IF character IS NOT AN HTML TAG:
            #Menghapus HTML tags
            ADD character TO row ['Preprocess']
    END FOR
    #Mengubah teks menjadi huruf kecil
    SET row ['Preprocess'] TO LOWERCASE(row['Preprocess'])
    #Menghapus spasi berlebihan
    SET row['Preprocess'] TO REMOVE_EXTRA_SPACES(row['Preprocess'])
    #Menghapus angka
    SET row['Preprocess'] TO REMOVE_NUMBERS(row['Preprocess'])
END FOR

```

Gambar 3. 2 pseudocode cleaning data

Berikut merupakan hasil dari data cleaning yang dilakukan pada dataset ditampilkan pada tabel 3.3.

Tabel 3. 3 Implementasi Cleaning

Sebelum Implementasi <i>Cleaning</i>	Setelah Implementasi <i>Cleaning</i>
Anda telah memenangkan hadiah uang sebesar £200 atau bahkan £1000 . Untuk mengeklaim hadiah ANDA telepon gratis pada nomor 08000407166 (18+) 2 stop getstop pada 88333 PHP	anda telah memenangkan hadiah uang sebesar £ atau bahkan £ . untuk mengeklaim hadiah anda telepon gratis pada nomor (+) stop getstop pada php

3.2.2.2. Tokenisasi

Proses tokenisasi merupakan proses yang akan membagi pesan berupa kalimat atau paragraf menjadi beberapa bagian. Tahap ini membagi pesan berdasarkan angka, spasi, dan tanda baca agar dapat dianalisis. Proses tokenisasi pada data dapat dilihat melalui pseudocode pada gambar 3.3.

```

#Fungsi untuk melakukan tokenisasi kata
FUNCTION TokenizeText(inputText):
    # Tokenisasi kata
    tokens ← SPLIT_TEXT_INTO_WORDS(inputText)
    Return tokens
END FUNCTION

#Fungsi untuk membagi teks menjadi kata-kata
FUNCTION SPLIT_TEXT_INTO_WORDS(text)
    # Implementasi untuk membagi teks berdasarkan spasi atau tanda baca
    wordList ← SPLIT_BY_WHITESPACE_AND_PUNCTUATION(text)
    RETURN wordList
END FUNCTION

```

Gambar 3. 3 Pseudocode tokenisasi data

Pseudocode di atas menggambarkan langkah-langkah yang dilakukan dalam kode program. Fungsi *word_tokenize_wrapper* digunakan untuk memecah teks menjadi token menggunakan *word_tokenize* dari NLTK. Setiap token kemudian dimasukkan ke dalam daftar *text_tokens* untuk setiap baris dalam dataframe. Berikut hasil implementasi text yang telah di tokenisasi dapat dilihat pada tabel 3.4.

Tabel 3. 4 Implementasi Tokenisasi

Sebelum Implementasi Tokenisasi	Setelah Implementasi Tokenisasi
anda telah memenangkan hadiah uang sebesar £ atau bahkan £ . untuk mengeklaim hadiah anda telepon gratis pada nomor (+) stop getstop pada php	'anda', 'telah', 'memenangkan', 'hadiah', 'uang', 'sebesar', '£', 'atau', 'bahkan', '£', '.', 'untuk', 'mengeklaim', 'hadiah', 'anda', 'telepon', 'gratis', 'pada', 'nomor', '(', '+', ')', 'stop', 'getstop', 'pada', 'php'

3.2.2.3. Stop Word Elimination

Penghapusan kata henti dilakukan untuk menghilangkan kata-kata yang dianggap tidak penting atau umum dalam teks. Kata-kata ini seringkali tidak memberikan

informasi yang berguna dalam analisis teks. Proses *stop word elimination* dapat dilihat melalui pseudocode pada gambar 3.4.

```

FUNCTION MainProcess(dataset)
    #Mengambil daftar stopwords bahasa Indonesia
    stopwordsList ← GET_INDONESIAN_STOPWORDS()
    #Menambahkan stopwords tambahan
    additionalStopwords ← [“kata1”, “kata2”, “kata3”]
    stopwordsList ← ADD_ADDITIONAL_STOPWORDS(stopwordsList,
    additionalStopwords)
    #Inisiasi list untuk menyimpan pesan yang telah dihapus stopwordsnya
    cleanedDataset ← EMPTY_LIST()
    #Iterasi melalui setiap pesan dalam dataset
    FOR each message IN dataset
        tokens ← TokenizeText(message)
        cleanedMessage ← RemoveStopwords(tokens, stopwordsList)
        ADD cleanedMessage TO cleanedDataset
    END FOR
    RETURN cleanedDataset
END FUNCTION

FUNCTION RemoveStopwords(tokens, stopwordsList)
    #Inisialisasi list untuk menyimpan token yang bukan stopwords
    filteredTokens ← EMPTY_LIST()
    #Iterasi melalui setiap token
    FOR each token IN tokens
        IF token IS NOT IN stopwordsList THEN
            ADD token TO filteredTokens
        END IF
    END FOR
    RETURN filteredTokens
END FUNCTION

```

Gambar 3. 4 Pseudocode Stopwords data

Langkah pertama adalah mendapatkan daftar stopwords dalam bahasa Indonesia dari NLTK. Kemudian, kata-kata tambahan yang dianggap tidak relevan juga ditambahkan ke dalam daftar stopwords. Daftar stopwords tersebut kemudian diubah menjadi bentuk set untuk efisiensi dalam pencarian. Fungsi stopwords_removal digunakan untuk menghapus stopwords dari daftar kata-kata. Setiap baris dalam kolom 'text_tokens' kemudian diproses dengan fungsi ini, dan hasilnya disimpan dalam kolom baru 'teks_tokens_WSW'. Hasil implementasi dapat dilihat pada tabel 3.5 dibawah ini.

Tabel 3. 5 Implementasi Stop Word Elimination

Sebelum Implementasi Stop Word Elimination	Setelah Implementasi Stop Word Elimination
'anda', 'telah', 'memenangkan', 'hadiyah', 'uang', 'sebesar', '£', 'atau', 'bahkan', '£', '.', 'untuk', 'mengeklaim', 'hadiyah', 'anda', 'telepon', 'gratis', 'pada', 'nomor', '(', '+', ')', 'stop', 'getstop', 'pada', 'php'	'memenangkan', 'hadiyah', 'uang', '£', '£', '.', 'mengeklaim', 'hadiyah', 'telepon', 'gratis', 'nomor', '(', '+', ')', 'stop', 'getstop', 'php'

3.2.2.4. *Stemming*

Langkah terakhir ialah stemming. Stemming berarti menghilangkan imbuhan pada setiap kata dalam pesan menjadi kata dasar. Hal ini dilakukan agar menghindari ejaan kata yang tidak tepat. Proses *stemming* dapat dilihat melalui pseudocode yang ada pada gambar 3.5.

```

FUNCTION StemTokens(tokens)
    #Inisialisasi list untuk menyimpan pesan yang telah di-stemming
    stemmedDTOkens ← EMPTY_LIST()
    #Iterasi melalui setiap token
    FOR each token IN token
        stemmedTokens ← STEM_WORD(token)
        ADD stemmedToken TO stemmedTokens
    END FOR
    RETURN stemmedTokens
END FUNCTION

#Fungsi untuk stemming kata
FUNCTION STEM_WORD(word)
    stemmedWORD ← APPLY_STEMMING_ALGORITHM(word)
    Return stemmedWord
END FUNCTION

```

Gambar 3. 5 Pseudocode stemming data

Setelah teks melewati berbagai proses seperti pseudocode diatas, Hasil implementasi stemming dapat dilihat pada tabel 3.6.

Tabel 3. 6 Implementasi *Stemming*

Sebelum Implementasi <i>Stemming</i>	Setelah Implementasi <i>Stemming</i>
'memenangkan', 'hadiyah', 'uang', '£', '£', '.', 'mengeklaim', 'hadiyah', 'telepon', 'gratis', 'nomor', '(', '+', ')', 'stop', 'getstop', 'php'	'menang', 'hadiyah', 'uang', "", "", "", 'klaim', 'hadiyah', 'telepon', 'gratis', 'nomor', "", "", "stop", 'getstop', 'php'

3.2.3. *Resampling*

Dalam tahap resampling ini, digunakan teknik Random Over Sampling (ROS) dari pustaka imbalanced-learn (imblearn) untuk menangani ketidakseimbangan kelas dalam dataset. ROS secara acak memilih sampel dari kelas minoritas (kelas yang kurang representatif) dengan penggandaan (duplikasi) sampel, sehingga jumlah sampel dalam kelas minoritas akan seimbang dengan jumlah sampel dalam kelas

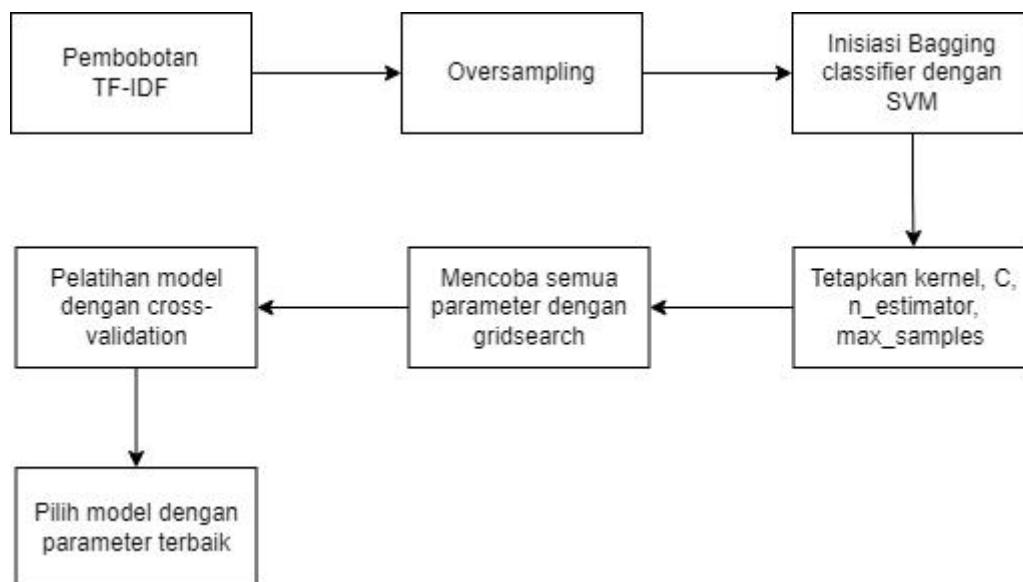
majoritas. Hal ini dilakukan untuk mencegah model yang dibuat cenderung memprediksi kelas mayoritas karena distribusi yang tidak seimbang dalam dataset.

3.2.4. Data Training

Setelah data training sudah terbentuk, maka data tersebut akan digunakan untuk melatih algoritma *svm*. Algoritma ini akan mengecek apakah data-data tersebut termasuk *linier* atau *non-linier*. Setelah itu algoritma ini akan mencari hyperplane terbaik untuk memisahkan data pesan phising atau tidak.

3.2.5. Processing

Pada tahapan ini, Support Vector Machine digunakan sebagai base model dalam proses pembuatan model. Algoritma ini mengatasi kesulitan dalam kategorisasi data dengan menciptakan batas keputusan yang lebih adaptif. Proses alur pelatihan model dari input hingga terpilihnya model terbaik dapat dilihat pada gambar 3.6 dibawah ini.



Gambar 3. 6 Diagram Alur Pelatihan Model

Teks tersebut akan diubah menjadi vektor fitur numerik, penjelasan terperinci mengenai proses operasional pelatihan diuraikan sebagai berikut.

1. Data yang berbentuk teks akan melalui proses encoding (pembobotan) dek yang berbentuk matrix. Matrix hasil encoding akan digunakan sebagai data training bagi model *svm*. Contoh matrix dapat dilihat pada gambar 3.7 dibawah ini.

	15 float64	coding float64	collection flo...	courses float64	data float64	file float
Doc1	0	0	0.4557324363227 4113	0	0.2908881107344 8395	
Doc2	0	0	0	0.4747707959705 7625	0.3030400490826 257	
Doc3	0	0	0	0	0	0.4217647
Doc4	0.3322643119288 0906	0.3322643119288 0906	0	0	0.2120800062890 7964	

Gambar 3. 7 Contoh Matrix TF-IDF

2. Sebelum melakukan training, Hasil Matrix tersebut akan melalui proses *resampling* dengan teknik *Random Over Sampler*, dimana proses ini akan memperbanyak data dari salah satu kelas agar seimbang dengan kelas lain dengan cara menyalin secara random data lainnya.
3. Matrik numerik yang telah melalui proses *resampling*, akan dilatih menggunakan algoritma *support vector machine*. Pelatihan ini melalui 3 tahap agar tercipta model yang dapat memisahkan 2 kelas, yaitu :
 - a) Menyusun Data Point

Algortima *support vector machine* akan menjadikan matrix numerik sebagai data point (vektor) dengan menggunakan persamaan :

$$\omega * x + b \quad (3.1)$$

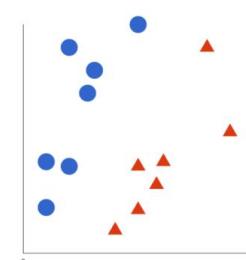
Di mana :

ω = vektor bobot yang menentukan orientasi hyperplane

x = vektor fitur dari pembobotan (hasil TF-IDF)

b = bias atau offset dari hyperplane

Data point (vektor) akan ditempatkan berdasarkan nilai fiturnya dalam ruang fitur (feature space). Contoh penempatan titik bedasarkan nilai fiturnya dapat dilihat pada gambar 3.8.



Gambar 3. 8 Contoh Penempatan Nilai Fitur Data

b) Mencari Hyperplane dan Margin

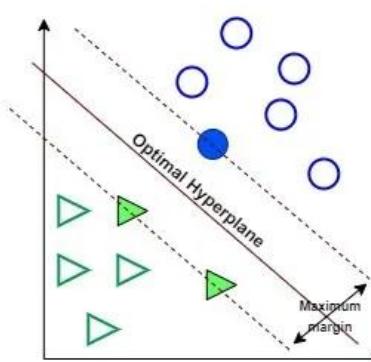
Setelah semua data dalam matrix numerik masuk kedalam ruang fitur, *support vector machine* akan mencari hyperplane dengan persamaan:

$$\omega * x + b = 0 \quad (3.2)$$

Hyperplane akan memisahkan data point menjadi 2 kelas berdasarkan posisi data point dalam ruang fitur *svm*. Sebagai contoh :

1. Jika $\omega * x + b > 0$, maka data point tersebut diklasifikasikan ke dalam satu kelas (misalnya, non-phising).
2. Jika $\omega * x + b < 0$, maka data point tersebut diklasifikasikan ke dalam kelas lainnya (misalnya, phising).

Lalu, *svm* juga mencari dan memaksimalkan margin maksimum untuk meningkatkan generalisasi model dan membuatnya lebih tahan terhadap overfitting. Margin maksimum merupakan jarak antara data point kelas satu dengan kelas lainnya yang terdekat dengan hyperplane. Untuk ilustrasi hyperplane dan margin dapat dilihat pada gambar 3.9.



Gambar 3. 9 Hyperplane dan Margin SVM

c) Model Terlatih

Setelah *svm* memiliki hyperplane dan margin maksimum, model akan menggunakan hyperplane yang didapat untuk mengklasifikasikan teks apakah teks tersebut termasuk phising atau tidak.

4. Langkah selanjutnya, menginisiasi bagging clasifier. Teknik ini merupakan teknik ensamble dimana, beberapa model *svm* akan dilatih pada subset data yang berbeda dan menggabungkan hasil prediksinya. Jumlah subset data akan diatur dalam parameter bagging clasifier.

5. Lalu, menggunakan parameter grid untuk mencari kombinasi parameter terbaik pada svm dan bagging clasifier. Parameter yang dipakai ialah, ‘C’, ‘gamma’, ‘kernel’ untuk Support Vector Machine, dan ‘n_estimators’, dan ‘max_samples’ Bagging Classifier. Detail parameter yang digunakan dapat dilihat pada tabel 3.7 dibawah ini.

Tabel 3. 7 Parameter Support Vector Machine

Parameter	Nilai
C	0.1, 1, 10, 100
Gamma	scale, auto
Kernel	linear, rbf, poly
N_estimator	10, 20

6. Proses pelatihan juga melibatkan cross-validation dimana dilakukan sebanyak 5 fold. Hal ini bertujuan untuk memastikan bahwa model tidak overfitting dan menghasilkan model hyperplane yang baik.
7. Setelah semua parameter telah di melewati proses training, maka akan dicari kombinasi hyperparameter terbaik berdasarkan kinerja cross validation. Sehingga model dengan kombinasi hyperparameter terbaik dapat mengidentifikasi apakah teks tersebut phising atau non-phising.

3.2.6. Evaluasi

Pada tahapan ini dilakukan pengujian *model performance* dari kedua model yang telah dihasilkan menggunakan confusion matrix. *Confussion matrix* adalah metode penghitungan yang membandingkan hasil klasifikasi dengan data sebenarnya. Matrik ini menunjukkan tingkat akurasi dalam persentase dan berguna sebagai acuan untuk menilai performa algoritma klasifikasi (Hasanah et al., 2019).

Tabel 3. 8 Confusion Matrix

Kelas	<i>Smishing</i>	<i>Ham</i>
<i>Smishing</i>	50	10
<i>Ham</i>	5	100

Menurut Ainurrohman (Ainurrohmah, 2021) *Confusion Matrix* dalam kinerjanya dapat dinilai dengan empat parameter: TP, FN, FP, dan TN. Sebagai contoh pada tabel 3.8, terdapat 50 pesan yang mengandung smishing dan diidentifikasi oleh sistem sebagai smishing. Maka, TP = 50. Lalu, terdapat 100 pesan yang mengandung ham dan diidentifikasi oleh sistem sebagai ham. Maka, TN = 100. Terdapat juga 5 pesan yang mengandung ham namun teridentifikasi oleh sistem sebagai smishing. Maka, FP = 5. Serta 10 pesan yang mengandung smishing namun diidentifikasi oleh sistem sebagai ham. Maka, FN = 10.

3.2.7. Deployment

Setelah model Bagging Classifier yang menggunakan SVM sebagai base model berhasil dilatih dan dievaluasi dengan baik, langkah selanjutnya adalah melakukan deployment model untuk digunakan dalam produksi. Dalam konteks ini, Flask digunakan sebagai kerangka kerja untuk deployment model. Flask adalah kerangka kerja aplikasi web yang ringan dan sederhana, cocok untuk digunakan dalam deployment model Machine Learning.

Proses deployment dimulai dengan mengembangkan aplikasi web sederhana menggunakan Flask. Aplikasi ini akan memiliki satu endpoint untuk menerima input data yang akan diklasifikasikan oleh model. Setelah menerima input data, aplikasi akan memproses data tersebut menggunakan model Bagging Classifier yang telah dilatih sebelumnya. Hasil klasifikasi akan dikembalikan sebagai respons dari endpoint tersebut.

3.2.8. Sistem Identifikasi

Algoritma yang telah dilatih menjadi learned model, akan dimasukkan kedalam sistem identifikasi. Sistem inilah yang akan mengidentifikasi data testing.

3.2.9. Output

Setelah model yang telah dilatih dimasukkan kedalam sistem tersebut, maka data testing akan diuji kedalam sistem. Output dari sistem ini merupakan pendekripsi apakah pesan tersebut phising atau tidak.

3.3. Perancangan Sistem

Pada tahap ini, akan dijelaskan mengenai rancangan antarmuka sistem yang berfokus pada identifikasi pesan teks *phising* ataupun normal. Sistem ini dikembangkan dalam bentuk website. Penjelasan rancangan ini bertujuan untuk memberikan pengguna gambaran menyeluruh tentang sistem yang sedang dikembangkan dan berfungsi sebagai panduan melalui proses pengembangan sistem.

3.3.1. Desain Halaman Cek Phising Teks

Halaman cek *phising* teks adalah halaman untuk pengguna mengidentifikasi apakah pesan termasuk *phising* atau tidak. Pengguna dapat memasukkan pesan dalam bentuk teks kedalam kotak yang disediakan. Setelah pengguna memasukkan teks yang ingin diperiksa, pengguna dapat mengidentifikasi dengan menekan tombol “Cek Pesan”. Lalu, Pesan akan teridentifikasi sebagai pesan *phising (smishing)* ataupun pesan normal (ham). Desain halaman cek phising teks diilustrasikan oleh gambar 3.10.



Gambar 3. 10 Desain Halaman Cek Phising Teks

Keterangan :

- Fungsi label 1 adalah untuk menginisiasi halaman cek phising dalam bentuk teks.
- Fungsi label 2 adalah menjadi tempat bagi pengguna untuk memasukkan teks yang akan diidentifikasi.

- c. Fungsi label 3 adalah sebagai tombol untuk melakukan pengecekan terhadap pesan yang sudah dimasukkan.
- d. Fungsi label 4 adalah notifikasi apakah pesan tersebut termasuk pesan phising (smishing) atau pesan normal (ham).

3.3.2. Desain Halaman Cek Phising File

Halaman cek phising file ini merupakan halaman yang dapat digunakan oleh pengguna untuk melakukan identifikasi pesan teks dalam bentuk .csv . Setelah memasukkan file yang dipilih, pengguna akan menekan tombol upload dan cek file. Lalu, pesan dalam file tersebut akan ditampilkan pada tabel dibawah dan juga label yang telah berisikan hasil identifikasi dari sistem. Pengguna dapat melihat tabel tersebut dan juga dapat mendownload hasil identifikasi dalam bentuk file .csv ketika menekan tombol download dibawah tabel. Desain halaman cek phising file diilustrasikan oleh gambar 3.11.

Gambar 3. 11 Desain Halaman Cek Phising File

Keterangan :

- a. Label 1 adalah untuk menginisiasi halaman cek phising dalam bentuk file.

- b. Label 2 adalah wadah pengguna untuk mengupload file berbentuk .csv yang akan diidentifikasi.
- c. Label 3 adalah tombol upload yang akan melakukan proses identifikasi pada file.
- d. Label 4 adalah tabel hasil yang ditampilkan sesuai dengan isi dari file yang diidentifikasi, beserta labelnya (smishing ataupun ham).
- e. Label 5 adalah tombol download yang dapat digunakan pengguna untuk download hasil identifikasi dalam bentuk file .csv juga.

3.3.3. Desain Halaman Training

Halaman training ini merupakan halaman yang digunakan oleh pengguna untuk melakukan pelatihan model melalui unggahan *file dataset* yang memiliki format .csv. Setelah berhasil mengunggah file, pengguna dapat memilih tombol training untuk memulai proses training pada model. Setelah proses selesai, halaman training akan menyajikan tabel yang berperan sebagai alat bantu bagi pengguna untuk membandingkan data sebelum dan setelah proses *preprocessing*. Desain halaman training dapat dilihat pada gambar 3.8.

Cek Phising

Home **Training** Testing Cek Phising Teks Cek Phising File

1

Training

Untuk melakukan proses training masukkan dataset berbentuk .csv

2 Input Dataset

Upload dan Mulai Training **3**

HASIL PREPROCESSING

Teks 4	Preprocessing

Gambar 3. 12 Desain Halaman Training

Keterangan :

- a. Label 1 adalah untuk menginisiasi halaman training.
- b. Label 2 adalah wadah pengguna untuk mengupload file dataset berbentuk .csv.
- c. Label 3 adalah tombol upload yang akan melakukan proses training.
- d. Label 4 adalah tabel hasil yang menampilkan teks beserta hasil preprocessing dari teks tersebut.

3.3.4. Desain Halaman Testing

Halaman testing ini merupakan halaman yang digunakan oleh pengguna untuk melakukan pengujian model melalui unggahan *file testing* yang memiliki format .csv. Setelah berhasil mengunggah file, pengguna dapat memilih tombol testing untuk memulai proses pengujian pada model. Setelah proses selesai, halaman testing akan menyajikan tabel yang berisi pesan teks, teks hasil *preprocessing*, label asli, dan label hasil prediksi. Halaman ini juga menampilkan informasi tambahan seperti evaluasi bedasarkan *confusion matrix* yang mencakup nilai *Precision*, *Recall*, *F1-Score*, dan *Accuracy*. Desain halaman testing dapat dilihat pada gambar 3.13.

Teks	Preprocessing	Label	Deteksi

Gambar 3. 13 Desain Halaman Testing

Keterangan :

- a. Label 1 adalah untuk menginisiasi halaman testing.
- b. Label 2 adalah wadah pengguna untuk mengupload file data uji berbentuk .csv.
- c. Label 3 adalah tombol upload yang akan melakukan proses testing.
- d. Label 4 adalah tabel hasil yang menampilkan teks, hasil *preprocessing*, label asli, dan label prediksi.

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Pengembangan sistem pengidentifikasi spam dilakukan penerapan algoritma *Support Vector Machine* yang dikombinasikan dengan *Ensemble Bagging*. Proses ini melibatkan komponen penunjang yaitu perangkat keras dan perangkat lunak yang meliputi:

4.1.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Berikut merupakan spesifikasi dari perangkat keras yang digunakan dalam penelitian ini:

1. Laptop Asus ROG Strix GL503GE
2. Prosesor Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz
3. Daya tampung 12 GB

Adapun spesifikasi dari perangkat lunak yang digunakan dalam penelitian ini meliputi:

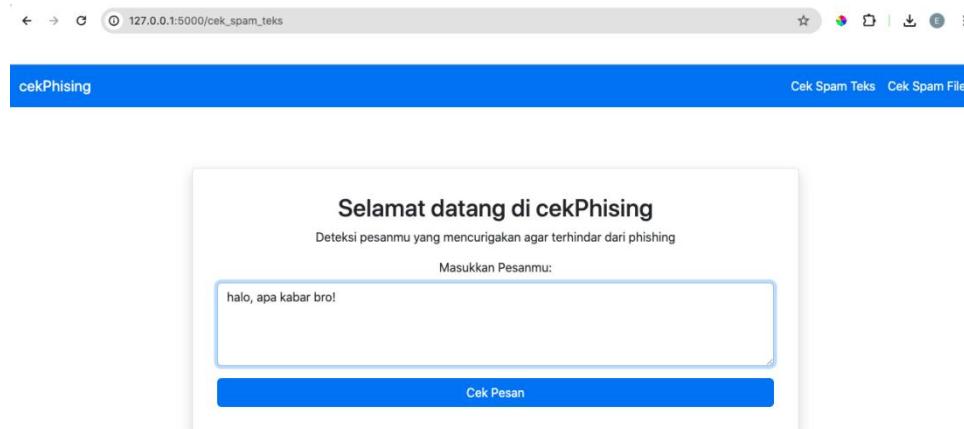
1. OS: Windows 10 Home Single Language
2. Bahasa pemrograman Python versi 3.9.6
3. IDE: *Google colab* dan *Microsoft Visual Studio Code*

4.1.2. Penerapan Sistem Berbasis Web

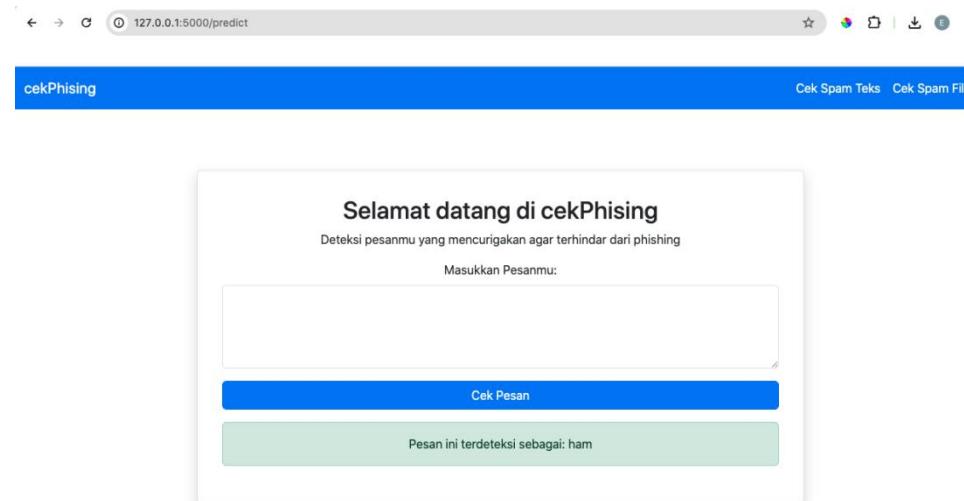
Berikut merupakan implementasi dari rancangan sistem berbasis web yang telah dijelaskan pada bab sebelumnya. Sistem ini terdiri dari empat menu yaitu cek phising teks, cek spam file, *training*, dan *testing*.

1. Halaman Cek Phising Berdasarkan Teks

pada halaman cek *phising* teks, pengguna dapat melakukan pengecekan pesan phising dengan menginputkan teks ke dalam *textbox* yang telah disediakan (gambar 4.1). Setelah menginputkan teks yang akan dilakukan pengecekan, pengguna dapat mengklik tombol cek pesan. Selanjutnya, sistem akan memproses teks lalu menampilkan output di bawah *button* “cek pesan” berupa hasil prediksi (gambar 4.2). Jika outputnya merupakan “ham”, maka akan tampil alert hijau. Sedangkan jika outputnya merupakan “smishing”, maka akan tampil alert merah.



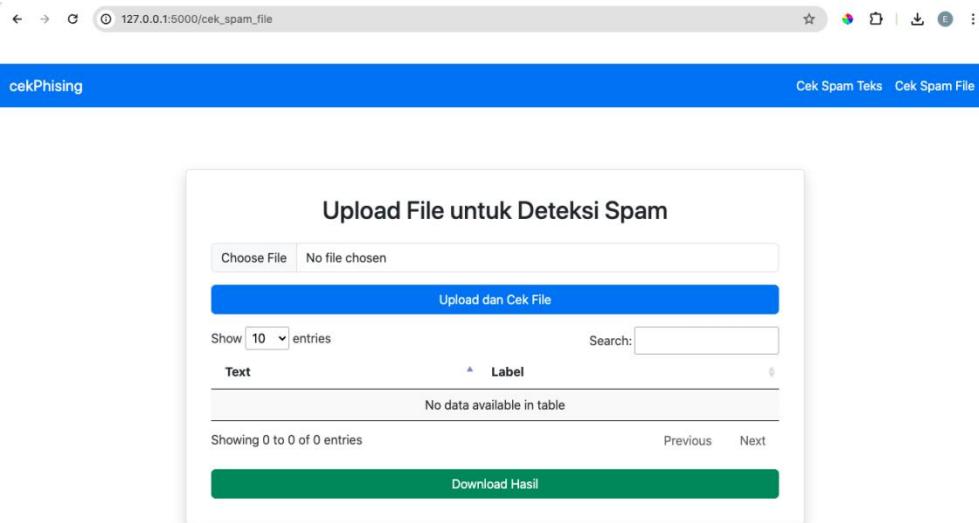
Gambar 4. 1 Halaman Cek Phising Teks



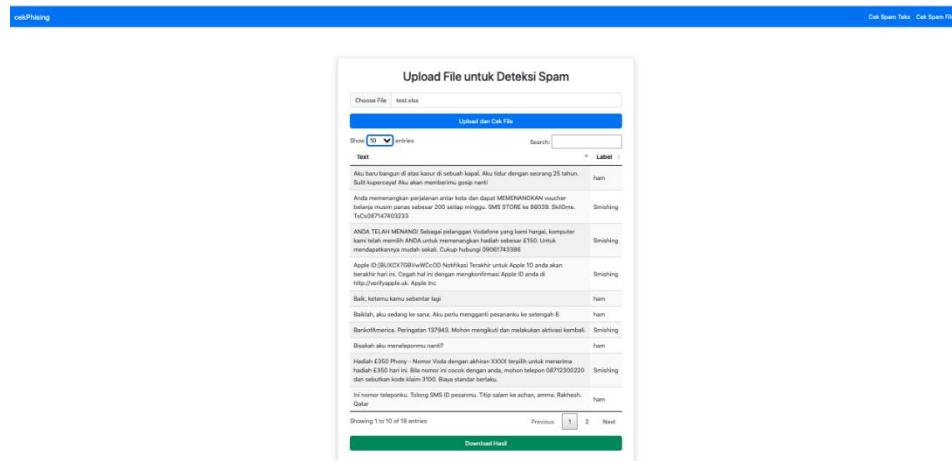
Gambar 4. 2 Output Teks

2. Halaman Cek Phising Berdasarkan File

Halaman cek phising file memungkinkan pengguna untuk menginputkan beberapa teks dalam satu file excel. Pengguna dapat mengunggah file excel yang berisi teks (gambar 4.3). Selanjutnya sistem akan memproses file tersebut kemudian menampilkan hasil prediksi ke dalam tabel yang tersedia (gambar 4.4). selain itu, pengguna juga dapat mengunduh output hasil prediksi dengan mengklik tombol “Download Hasil”.



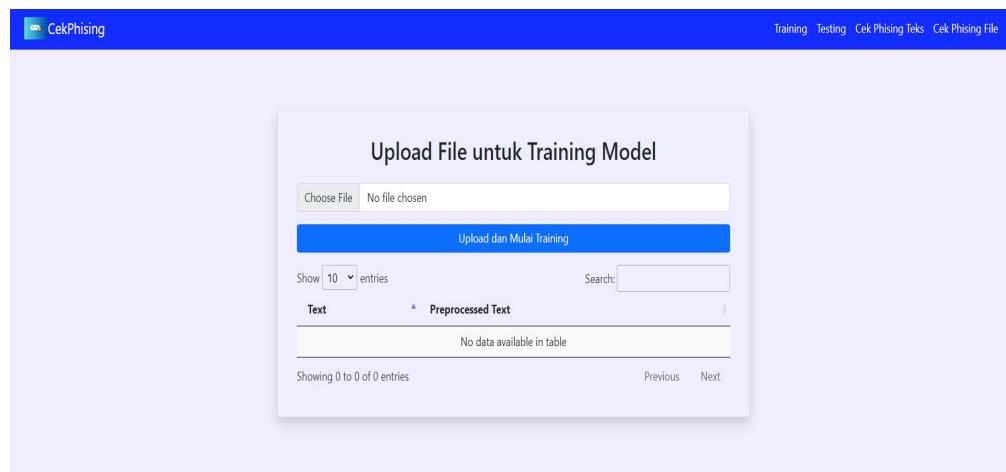
Gambar 4. 3 Halaman Cek Phising File



Gambar 4. 4 Output Cek File

3. Halaman Training

Halaman training memungkinkan pengguna untuk mengupload data training untuk melatih model. Pengguna dapat mengunggah file excel yang berisi pesan teks beserta label (gambar 4.5). Selanjutnya sistem akan menjalankan proses pelatihan model menggunakan *data training* tersebut. Setelah selesai, halaman akan menampilkan output berupa tabel pesan teks beserta hasil *preprocessing*-nya(gambar 4.6).



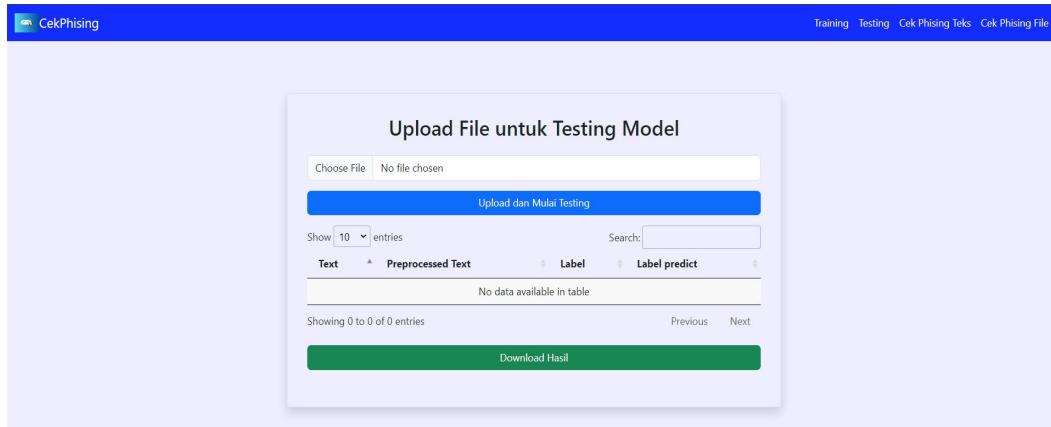
Gambar 4. 5 Halaman Training

Upload File untuk Training Model	
Choose File	train.xlsx
Upload dan Mulai Training	
Show 10 entries	Search:
Text	Preprocessed Text
Susun programmu dengan baik	susun programmu
<=> Tap the link to log in. DON'T SHARE IT, for security reasons. WITH ANYONE. Not even Gojek. https://login.gojek.com/ln/4ZxhJlpbo9q4PKeZqV9ubl	tap the link to log in . do n't share it , for secur reason , with anyone . not even gojek . http : //login.gojek.com /ln /4ZxhJlpbo9q4PKeZqV9ubl
Telepon dari 08702490080 memberitahuanda untuk menelpn 09066353152 untuk mengeklaim hadiah 5000. Anda harus memasukkan semua detail pribadi dan mobile anda. Hati-hati!	telepon memberitahuanda menelpn mengeklaim hadiah . memasukkan detail pribadi mobil . hati-hati !
"Ayo cek dan beli kembali paket internet kamu yang lalu. Cek di	" ayo cek beli paket internet . cek tsel.me/combo-sakti

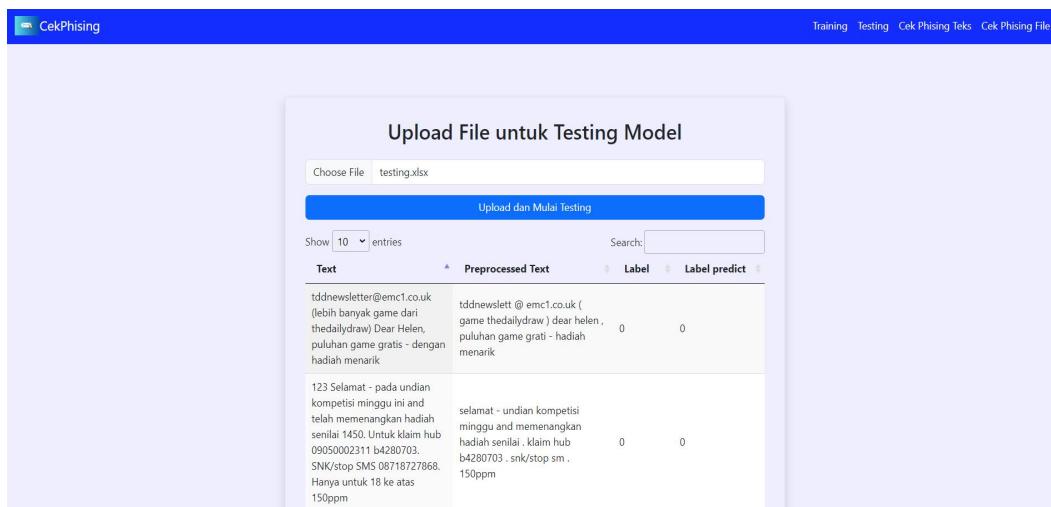
Gambar 4. 6 Output Training

4. Halaman Testing

Halaman testing memungkinkan pengguna untuk menguji model menggunakan data uji. Pengguna dapat mengunggah file excel yang berisi pesan teks dan label asli (gambar 4.7). Selanjutnya sistem akan menjalankan proses pengujian model. Setelah selesai, halaman akan menampilkan output berupa tabel pesan teks, hasil prediksinya (gambar4.8) dan informasi *confusion matrix* (gambar 4.9).



Gambar 4. 7 Halaman Testing



Gambar 4. 8 Output testing



Gambar 4. 9 Output Confusion Matrix

4.2. Hasil Pelatihan Model

Penelitian ini menggunakan SVM (*Support Vector Machine*) sebagai model dasar atau *base model* dan *Bagging Classifier* sebagai metode ensemble untuk meningkatkan kinerja prediksi. SVM digunakan karena kemampuannya dalam menangani data dengan jumlah fitur yang besar dan kemampuannya dalam menangani data non-linear melalui penggunaan fungsi kernel. Sedangkan, *Bagging Classifier* digunakan untuk mengurangi varians model dengan cara melatih beberapa model pada sampel data yang berbeda dan kemudian menggabungkan hasil prediksi dari model-model tersebut. Guna mendapatkan parameter yang paling optimal, peneliti menggunakan *GridSearchCV* untuk melakukan kombinasi parameter dan menentukan kombinasi parameter yang menghasilkan akurasi tertinggi. Adapun parameter yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 4. 1 Daftar Parameter

Parameter	Value
<i>Base_estimator_C</i>	[0.1, 1, 10, 100]
<i>Base_estimator_gamma</i>	[‘scale’, ‘auto’]
<i>Base_estimator_kernel</i>	[‘linear’, ‘rbf’, ‘poly’]
<i>N_estimators</i>	[10, 20]
<i>Max_samples</i>	[0.5, 1.0]

Dari parameter pada tabel 4.1 tersebut, selanjutnya dilakukan kombinasi yang hasilnya dapat dilihat pada tabel berikut:

Tabel 4. 2 Hasil GridSearchCV

C	gamma	kernel	max_samples	n_estimators	Mean_test_score
0,1	scale	linear	0,5	10	0,906
0,1	scale		0,5	20	0,899
0,1	scale		1	10	0,930
0,1	scale		1	20	0,925
0,1	scale	rbf	0,5	10	0,915
0,1	scale		0,5	20	0,906
0,1	scale		1	10	0,903
0,1	scale		1	20	0,892

C	gamma	kernel	max_samples	n_estimators	mean_test_score
0,1	scale	poly	0,5	10	0,850
0,1	scale		0,5	20	0,916
0,1	scale		1	10	0,853
0,1	scale		1	20	0,862
0,1	auto	linear	0,5	10	0,896
0,1	auto		0,5	20	0,897
0,1	auto		1	10	0,929
0,1	auto		1	20	0,924
0,1	auto	rbf	0,5	10	0,842
0,1	auto		0,5	20	0,842
0,1	auto		1	10	0,831
0,1	auto		1	20	0,839
0,1	auto	poly	0,5	10	0,499
0,1	auto		0,5	20	0,500
0,1	auto		1	10	0,499
0,1	auto		1	20	0,499
1	scale	linear	0,5	10	0,955
1	scale		0,5	20	0,963
1	scale		1	10	0,963
1	scale		1	20	0,967
1	scale	rbf	0,5	10	0,962
1	scale		0,5	20	0,961
1	scale		1	10	0,969
1	scale		1	20	0,969
1	scale	poly	0,5	10	0,936
1	scale		0,5	20	0,937
1	scale		1	10	0,940
1	scale		1	20	0,939
1	auto	linear	0,5	10	0,954
1	auto		0,5	20	0,963
1	auto		1	10	0,969

C	gamma	kernel	max_samples	n_estimators	mean_test_score
1	auto	rbf	1	20	0,966
1	auto		0,5	10	0,890
1	auto		0,5	20	0,914
1	auto		1	10	0,903
1	auto		1	20	0,900
1	auto	poly	0,5	10	0,499
1	auto		0,5	20	0,500
1	auto		1	10	0,501
1	auto		1	20	0,500
10	scale	linear	0,5	10	0,957
10	scale		0,5	20	0,961
10	scale		1	10	0,963
10	scale		1	20	0,966
10	scale	rbf	0,5	10	0,960
10	scale		0,5	20	0,961
10	scale		1	10	0,968
10	scale		1	20	0,970
10	scale	poly	0,5	10	0,942
10	scale		0,5	20	0,938
10	scale		1	10	0,941
10	scale		1	20	0,943
10	auto	linear	0,5	10	0,960
10	auto		0,5	20	0,963
10	auto		1	10	0,962
10	auto		1	20	0,963
10	auto	rbf	0,5	10	0,891
10	auto		0,5	20	0,893
10	auto		1	10	0,895
10	auto		1	20	0,900
10	auto	poly	0,5	10	0,500
10	auto		0,5	20	0,499

C	gamma	kernel	max_samples	n_estimators	mean_test_score
10	auto	poly	1	10	0,501
10	auto		1	20	0,499
100	scale	linear	0,5	10	0,959
100	scale		0,5	20	0,955
100	scale		1	10	0,964
100	scale		1	20	0,966
100	scale	rbf	0,5	10	0,964
100	scale		0,5	20	0,964
100	scale		1	10	0,969
100	scale		1	20	0,970
100	scale	poly	0,5	10	0,931
100	scale		0,5	20	0,930
100	scale		1	10	0,938
100	scale		1	20	0,941
100	auto	linear	0,5	10	0,957
100	auto		0,5	20	0,957
100	auto		1	10	0,963
100	auto		1	20	0,962
100	auto	rbf	0,5	10	0,884
100	auto		0,5	20	0,894
100	auto		1	10	0,913
100	auto		1	20	0,905
100	auto	poly	0,5	10	0,500
100	auto		0,5	20	0,500
100	auto		1	10	0,568
100	auto		1	20	0,585

Mean_test_score yang didapatkan pada tabel 4.2 diatas merupakan Rata-rata skor cross-validation dari lima fold data testing yang menggunakan setiap kombinasi parameter yang telah ditentukan. Kesimpulan dapat diambil bahwa kernel terbaik dalam pelatihan model *support vector machine* ini ialah jenis rbf. Kernel ini terpilih

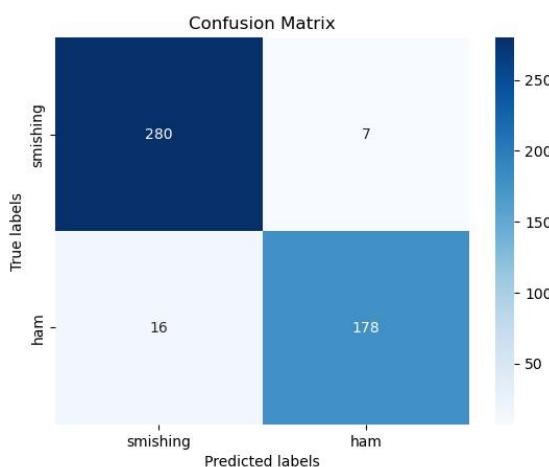
dengan akurasi tertinggi dikarenakan kernel *rbf* mampu membagi kelas data menjadi 2 bagian dengan sangat baik. Kombinasi parameter spesifik untuk kernel *rbf* ini ialah nilai C sebesar 100 dan nilai gamma ialah *scale*.

Selanjutnya, kombinasi parameter *ensamble bagging* terbaik ialah *max_sample* = 1 dan *n_estimator* = 20. Model dengan gabungan kombinasi parameter dari support vector machine dan *ensamble bagging* ini berhasil mencapai akurasi sebesar 95.2%. Hal ini berarti model memiliki akurasi sebesar 95.2% dalam mengidentifikasi *phising* pada pesan teks dengan benar. Rincian mengenai nilai akurasi dari setiap fold dengan kombinasi parameter terbaik ini dapat dilihat pada tabel 4.3 dibawah ini.

Tabel 4. 3 Hasil Rata-rata skor cross-validation dari Setiap Fold dengan Parameter Terbaik

Fold	Mean_test_score
1	0.966
2	0.974
3	0.963
4	0.974
5	0.974

Adapun *confusion matrix* yang dihasilkan oleh kombinasi tersebut adalah sebagai berikut.



Gambar 4. 10 Confusion Matrix

Dari *confusion matrix* di atas, model berhasil memprediksi 280 teks smishing dengan benar dan 7 teks diprediksi salah. Selain itu, model juga berhasil memprediksi 178

teks ham dan 16 teks diprediksi salah. Nilai-nilai tersebut dijabarkan dalam tabel 4.3 dan 4.4 berikut.

Tabel 4. 4 Perhitungan data yang mengandung pesan normal (ham)

No.	Pesan Teks Normal	Akumulasi
1	<i>True Positive</i>	178
2	<i>True Negative</i>	280
3	<i>False Positive</i>	7
4	<i>False Negative</i>	16

Tabel 4. 5 Perhitungan data yang mengandung pesan phising (smishing)

No.	Pesan Teks Phising (smishing)	Akumulasi
1	<i>True Positive</i>	280
2	<i>True Negative</i>	178
3	<i>False Positive</i>	16
4	<i>False Negative</i>	7

Adapun perhitungan akurasi, *precision*, *recall*, dan *f1 score* adalah sebagai berikut:

$$Akurasi = \frac{True\ Positives + True\ Negatives}{Total\ Population} \times 100\%$$

$$Akurasi = \frac{178 + 280}{481} \times 100\%$$

$$Akurasi \approx 0.952 \times 100\%$$

$$Akurasi \approx 95.2\%$$

Total populasi merupakan total keseluruhan dari data uji. Sehingga $178+280+7+16 = 481$. Dari perhitungan di atas, dapat dilihat bahwa akurasi dari model yang dihasilkan adalah 95.2%.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \times 100\%$$

Untuk kelas “ham”:

$$Precision_{ham} = \frac{178}{178 + 7} \times 100\%$$

$$Precision_{ham} \approx 0.962 \times 100\%$$

$$Precision_{ham} \approx 96.2\%$$

Untuk kelas “smishing”:

$$Precision_{smishing} = \frac{280}{280 + 16} \times 100\%$$

$$Precision_{smishing} \approx 0.945 \times 100\%$$

$$Precision_{smishing} \approx 94.5\%$$

Dari hasil perhitungan di atas, dapat diketahui bahwa *precision* untuk kelas “ham” adalah sekitar 96.2% dan *precision* untuk kelas *precision* untuk kelas “smishing” adalah sekitar 94.5%.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \times 100\%$$

Untuk kelas “ham”:

$$Recall_{ham} = \frac{178}{178 + 16} \times 100\%$$

$$Recall_{ham} \approx 0.917 \times 100\%$$

$$Recall_{ham} \approx 91.7\%$$

Untuk kelas “smishing”:

$$Recall_{smishing} = \frac{280}{280 + 7} \times 100\%$$

$$Recall_{smishing} \approx 0.975 \times 100\%$$

$$Recall_{smishing} \approx 97.5\%$$

Dari hasil perhitungan di atas, dapat diketahui bahwa *recall* untuk kelas “ham” adalah sekitar 91.7% dan *recall* untuk kelas *recall* untuk kelas “smishing” adalah sekitar 97.5%.

Dari hasil perhitungan *recall* dan *precision* selanjutnya dapat dihitung *f1 score*. Perhitungannya adalah sebagai berikut:

$$F1_{ham} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\%$$

Untuk kelas “ham”:

$$F1_{ham} = 2 \times \frac{0.962 \times 0,917}{0,962 + 0,917} \times 100\%$$

$$F1_{ham} = 2 \times \frac{0,882}{1,879} \times 100\%$$

$$F1_{ham} = 2 \times 0,469 \times 100\%$$

$$F1_{ham} \approx 0,938 \times 100\%$$

$$F1_{ham} \approx 93.8\%$$

Untuk kelas “smishing”:

$$F1_{smishing} = 2 \times \frac{0.945 \times 0,975}{0,945 + 0,975} \times 100\%$$

$$F1_{smishing} = 2 \times \frac{0,921}{1,920} \times 100\%$$

$$F1_{smishing} = 2 \times 0,479 \times 100\%$$

$$F1_{smishing} \approx 0,959 \times 100\%$$

$$F1_{smishing} = 95.9\%$$

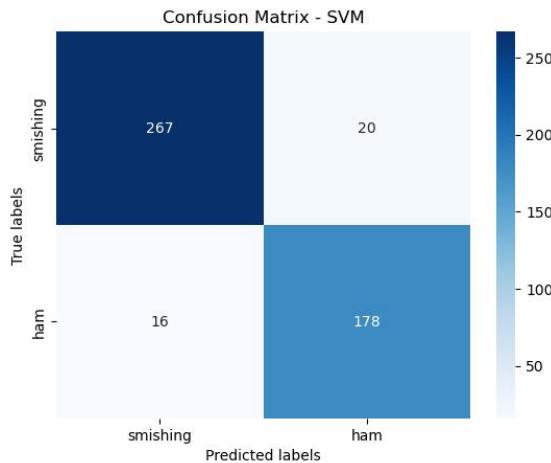
Dari hasil perhitungan di atas, dapat diketahui bahwa F1 score untuk kelas “ham” adalah 93.8% dan F1 score untuk kelas smishing adalah 95.9%.

Perhitungan data dari teks yang mengandung phising (smishing) dan teks normal (ham) selanjutnya disajikan pada tabel 4.2 berikut ini.

Tabel 4. 6 Hasil Komputasi Nilai Evaluasi

	Precision	Recall	F1-Score
Pesan teks normal (ham)	96.2%	91.7%	93.8%
Pesan teks mengandung phising (smishing)	94.5%	97.5%	95.9%

Untuk pelatihan menggunakan algoritma support vector machine tanpa ensamble bagging, Parameter terbaik yang terpilih ialah support vector machine dengan nilai C sebesar 10 dan kernel linear. Berikut hasil pengujian pada data testing menggunakan model yang telah dilatih tanpa ensamble bagging, dapat dilihat pada gambar 4.11.



Gambar 4. 11 Confusion Matrix Svm

Dengan confusion matrix seperti pada gambar, akurasi dapat dihitung dengan rumus yang sebelumnya digunakan. Berikut perhitungan akurasi yang didapatkan.

$$Akurasi = \frac{True\ Positives + True\ Negatives}{Total\ Population} \times 100\%$$

$$Akurasi = \frac{267 + 178}{481} \times 100\%$$

$$Akurasi = \frac{445}{481} \times 100\%$$

$$Akurasi \approx 0.925 \times 100\%$$

$$Akurasi \approx 92.5\%$$

Dari perhitungan di atas, dapat dilihat bahwa akurasi dari model yang dihasilkan adalah 92.5%. Lalu hasil metrik lainnya seperti, *precision*, *recall*, dan *f1-score* dapat dilihat pada tabel 4.7.

Tabel 4. 7 Hasil Komputasi Nilai Evaluasi SVM

	Precision	Recall	F1-Score
Pesan teks normal (ham)	89.9%	91.7%	90.8%
Pesan teks mengandung phising (smishing)	94.3%	93.0%	93.6%

Dari hasil perhitungan metrik diatas, model svm tanpa ensamble bagging mengalami penurunan metrik sebesar 0% sampai 6.3%.

4.3. Hasil Pengujian model

Guna memastikan model bekerja dengan baik, perlu dilakukan pengujian model ke data teks secara langsung. Pada penelitian ini pengujian kinerja model menggunakan 30% data asli yang diperoleh dari pembagian data *training* dan *testing*. Namun dalam sub bab ini hanya ditampilkan beberapa data saja. Berikut merupakan hasil pengujian:

Tabel 4. 7 Hasil Pengujian Model

Text	Aktual	Prediksi
Mohon untuk tetap berada di rumah. Untuk mendorong warga agar tetap berada di rumah, setiap warga berhak mendapatkan dana darurat sebesar 305.96 atau lebih	Smishing	Smishing
BankofAmerica. Peringatan 137943. Mohon mengikuti dan melakukan aktivasi kembali.	Smishing	Smishing
Anda memenangkan perjalanan antar kota dan dapat MEMENANGKAN voucher belanja musim panas sebesar 200 setiap minggu. SMS STORE ke 88039. SkilGme. TsCs087147403233	Smishing	Smishing
ANDA TELAH MENANG! Sebagai pelanggan Vodafone yang kami hargai, komputer kami telah memilih ANDA untuk memenangkan hadiah sebesar £150. Untuk mendapatkannya mudah sekali. Cukup hubungi 09061743386	Smishing	Smishing
Hadiah £350 Phony - Nomor Voda dengan akhiran XXXX terpilih untuk menerima hadiah £350 hari ini. Bila nomor ini cocok dengan anda, mohon telepon 08712300220 dan sebutkan kode klaim 3100. Biaya standar berlaku.	Smishing	Smishing
Apple ID:[BUXCX7GBVwWCcOD] Notifikasi Terakhir untuk Apple 1D anda akan berakhir hari ini. Cegah hal ini dengan mengkonfirmasi Apple ID anda di http://verifyapple.uk . Apple Inc	Smishing	Smishing
Nomor Voda dengan akhiran 7548 terpilih untuk menerima hadiah sebesar £350 hari ini. Bila nomor anda sesuai, mohon	Smishing	Smishing

Tabel 4. 7 Hasil Pengujian Model (Lanjutan)

Text	Aktual	Prediksi
telepon 08712300220 dengan menyebutkan kode klaim 4041. Biaya standar berlaku		
Itu keren, aku ingin memuaskanmu	ham	ham
Bisakah aku meneleponmu nanti?	ham	ham
Ini nomor teleponku. Tolong SMS ID pesanmu. Titip salam ke achan, amma. Rakhesh. Qatar	ham	ham
Selamat pagi, tolong panggil saya tuan.	ham	ham
LOL baiklah aku juga memikirkan itu haha	ham	ham
Oke pa. Bukan masalah :)	ham	ham
Baiklah, aku sedang ke sana. Aku perlu mengganti pesananku ke setengah 8	ham	ham
Maaf tentang itu, ini telepon teman saya dan saya tidak menulis itu. Dengan cinta, Kate	ham	ham
Baik, ketemu kamu sebentar lagi	ham	ham
Aku baru bangun di atas kasur di sebuah kapal. Aku tidur dengan seorang 25 tahun. Sulit kupercaya! Aku akan memberimu gosip nanti	ham	ham
selamat!! Anda mendapatkan 100jt	Smishing	Smishing

Hasil pengujian menunjukkan bahwa model memiliki kinerja yang sangat baik dalam mengklasifikasikan pesan teks sebagai ham atau *smishing*. Untuk pesan ham (bukan *smishing*), model dengan tepat mengidentifikasi semua pesan sebagai ham. Di sisi lain, untuk pesan smishing, model juga berhasil mengklasifikasikan sebagian besar pesan sebagai *smishing*. Dengan demikian, hasil pengujian ini menunjukkan bahwa model yang dihasilkan sangat baik dalam membedakan antara pesan ham dan *smishing*.

Terdapat beberapa pesan yang teridentifikasi salah oleh sistem identifikasi ini. Berikut dapat diuraikan kesalahan yang terjadi pada data testing dalam tabel 4.4 dibawah ini.

Tabel 4. 8 Pernyataan salah oleh sistem dalam data testing

Pesan Teks	Label	Identifikasi	Keterangan
AngpaoPoinSenyum! Dptkan Vchr Lottemart 50rb dg tukar 2500poin (normal 5000poin) sd 11Feb16. Ketik *123*7887*2*5# TERBATAS! Info: www.indosatooredoo.com/angpao	Ham	Smishing	salah
Selamat Natal & Tahun Baru 2024! Aktifkan NSP special NARU buat lebih meriah perayaannmu, cuma 3ribuan bs dapetin hadiah jutaan rupiah! Hub *121*2024# S&k	Ham	Smishing	salah
Bagi yg tertarik lanjut ke gemastik 2016 ini tempat diskusinya : http://line.me/R/ti/g/xxxxx	Ham	Smishing	salah
Nomor AS anda : 6282360005477 layanan ini dikenakan tariff Rp 100. info CS:188 Promo: Hindari gangguan penelpon tak dikenal, aktifkan Call Manager hub *500*17#	Ham	Smishing	salah
Selamat anda terdftr diProg GRATIS Nelp 1000Mnt&1000SMS keISAT+Internet hg 1000MB CUMA dg IsiUlang mulai 10rb. isiUlang pulsamu&Nikmati bonusnya.Cekbns: *555*3#	Smishing	Ham	salah
Pelanggan, Apple ID anda kadaluarsa hari ini. Cegah dengan konfirmasi Apple ID di ataligaid daal.au	Smishing	Ham	salah
pembayaran anda sudah jatuh tempo silahkan melapor/M-HUB ke Kantor 082327327705 terima kasih	Smishing	Ham	salah

Kesalahan pendekripsi pada data testing berikut disebabkan oleh adanya Variasi pesan teks normal yang mirip dengan pesan teks *phising*. Variasi ini membuat model kesulitan dalam membedakan antara pesan normal dan phishing, terutama ketika pesan tersebut memiliki pola atau struktur yang serupa. Hal ini dapat terjadi dikarenakan keterbatasan sumber dalam pengumpulan data. Begitu juga untuk pesan *phising* yang terdeteksi sebagai pesan normal, dimana pesan teks tersebut memiliki variasi yang mirip dengan pesan normal pada data *training*. Kesalahan klasifikasi tersebut dapat dihitung dengan menggunakan data false positive (FP) dan false negatives (FN). Berikut perhitungan kesalahan klasifikasi (toleransi kesalahan kesalahan klasifikasi).

$$\text{Persentasi kesalahan klasifikasi} = \left(\frac{\text{False Positive} + \text{False Negatives}}{\text{Total Prediksi}} \right) \times 100\%$$

$$\text{Persentasi kesalahan klasifikasi} = \left(\frac{16 + 7}{481} \right) \times 100\%$$

$$\text{Persentasi kesalahan klasifikasi} = \left(\frac{23}{481} \right) \times 100\%$$

$$\text{Persentasi kesalahan klasifikasi} \approx 4.78\%$$

Dari perhitungan yang telah dilakukan, Persentase toleransi kesalahan klasifikasi adalah sekitar 4.78%.

4.4. Hasil Evaluasi Pengguna

Untuk mengevaluasi pengalaman pengguna dalam menggunakan website yang telah dibangun untuk identifikasi phishing, dilakukan survei terhadap sejumlah pengguna. Evaluasi ini mencakup aspek-aspek berikut :

1. **Kegunaan (Usability):** Pengguna menilai kemudahan navigasi dan penggunaan fitur-fitur di website. Survei mencakup pertanyaan tentang kemudahan akses ke halaman pelatihan dan pengujian, serta kemudahan dalam memahami hasil yang ditampilkan.
2. **Kepuasan (Satisfaction):** Pengguna diminta menilai tingkat kepuasan mereka terhadap desain antarmuka, kecepatan respon sistem, dan kejelasan informasi yang disajikan.

3. **Kepercayaan (Trust):** Pengguna menilai seberapa percaya mereka terhadap hasil klasifikasi yang diberikan oleh sistem. Pertanyaan survei mencakup persepsi pengguna terhadap akurasi dan reliabilitas sistem.

Evaluasi dilakukan dengan menggunakan skala likert sebagai penilaian evaluasi. Berikut merupakan pernyataan yang telah disusun peneliti untuk mengetahui nilai evaluasi dari pengguna dapat dilihat pada tabel 4.9.

Tabel 4. 9 Tabel Pernyataan Evaluasi Pengguna

No	Pernyataan
1	Navigasi dalam website mudah digunakan.
2	Fitur - fitur dalam website ini mudah digunakan.
3	Hasil identifikasi yang ditampilkan mudah dipahami.
4	Pengguna puas dengan desain antarmuka website ini.
5	Pengguna puas dengan kecepatan respon sistem.
6	Pengguna percaya pada hasil identifikasi yang diberikan oleh sistem.
7	Pengguna merasa bahwa sistem ini akurat dan dapat diandalkan.
8	Informasi yang disajikan di website ini jelas dan informatif.

Setelah dilakukan evaluasi terhadap 10 pengguna, didapatkan hasil penilaian pengguna dengan penilaian skala likert yang dapat dilihat pada tabel 4.10 dibawah ini.

Tabel 4. 10 Hasil Evaluasi Pengguna

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Navigasi dalam website mudah digunakan.	0	0	2	5	3
2	Fitur - fitur dalam website ini mudah digunakan.	0	0	2	5	3
3	Hasil identifikasi yang ditampilkan mudah dipahami.	0	0	1	6	3

Tabel 4. 11 Hasil Evaluasi Pengguna (lanjutan)

No	Pernyataan	Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
4	Pengguna puas dengan desain antarmuka website ini.	0	0	1	5	4
5	Pengguna puas dengan kecepatan respon sistem.	0	0	2	4	4
6	Pengguna percaya pada hasil identifikasi yang diberikan oleh sistem.	0	0	1	6	3
7	Pengguna merasa bahwa sistem ini akurat dan dapat diandalkan.	0	0	1	5	4
8	Informasi yang disajikan di website ini jelas dan informatif.	0	0	1	5	4

Hasil evaluasi menunjukkan bahwa mayoritas pengguna merasa puas dengan kegunaan dan keandalan sistem, Desain antarmuka dan kecepatan respon sistem mendapat penilaian yang baik, namun ada ruang untuk peningkatan terutama dalam kecepatan respon.

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem, dapat disimpulkan bahwa :

1. Model yang dikembangkan menggunakan algoritma Support Vector Machine (SVM) yang dikombinasikan dengan Ensemble Bagging mampu mengidentifikasi pesan teks sebagai ham atau Smishing dengan sangat baik.
2. Pada awalnya dataset yang digunakan sebanyak 1000 data, yang diambil dari penelitian sebelumnya. Dengan dataset tersebut, peneliti mendapatkan model dengan akurasi 98%. Lalu, saat dataset ditambahkan menjadi 1600 data, peneliti mendapatkan model dengan akurasi 95.2%. Akurasi turun sebesar 3.8% dikarenakan data yang ditambahkan memiliki variasi yang mirip satu sama lain.
3. Pelatihan dan pengujian model dengan support vector machine tanpa bantuan ensamble bagging berhasil mencapai akurasi sebesar 92.5%. Hal ini menunjukkan bahwa dengan bantuan ensamble bagging, akurasi dari algoritma support vector machine dapat ditingkatkan sebanyak 2.7% menjadi 95.2%.
4. Model memberikan perlindungan yang efektif terhadap serangan phising dan meningkatkan keamanan pengguna dalam berkomunikasi melalui pesan teks.
5. Dengan demikian, model ini dapat diandalkan dalam memfilter pesan phising melalui pesan teks.

5.2. Saran

Adapun sejumlah saran terkait pengembangan kinerja sistem dalam penelitian penelitian selanjutnya meliputi:

1. Dalam penelitian ini, data yang digunakan berjumlah 1600 namun masih kurang dalam variasi data sehingga beberapa teks dengan variasi yang mirip masih belum teridentifikasi *phising*. Untuk menghindari hal tersebut, disarankan agar penelitian selanjutnya memperbanyak variasi dataset serta jumlah data yang digunakan.

2. Eksplorasi teknik *preprocessing* yang lebih canggih, seperti penggunaan *word embeddings* atau teknik NLP lainnya, dapat membantu meningkatkan pemahaman dan representasi teks yang lebih baik, sehingga meningkatkan kinerja model.
3. Menggabungkan beberapa model atau teknik klasifikasi, seperti kombinasi SVM dengan model lain atau *ensemble learning*, dapat membantu meningkatkan kinerja model dengan memanfaatkan kekuatan masing-masing pendekatan.
4. Melakukan optimasi hyperparameter yang lebih intensif dan eksploratif dapat membantu menemukan kombinasi hyperparameter yang lebih baik untuk meningkatkan kinerja model.

DAFTAR PUSTAKA

- Adrian, M. R., Putra, M. P., Rafialdy, M. H., & Rakhmawati, N. A. (2021). Perbandingan Metode Klasifikasi Random Forest dan SVM Pada Analisis Sentimen PSBB. *Jurnal Informatika Upgris*, 7(1). <https://doi.org/10.26877/jiu.v7i1.7099>
- Ainurrohmah. (2021). Akurasi Algoritma Klasifikasi pada Software Rapidminer dan Weka. PRISMA, Prosiding Seminar Nasional Matematika, 4, 493–499.
- Aung, M. H., Seluka, P. T., Fuata, J. T. R., Tikoisuva, M. J., Cabealawa, M. S., & Nand, R. (2020). Random Forest Classifier for Detecting Credit Card Fraud based on Performance Metrics. *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. IEEE.
- Basit, A., Zafar, M., Javed, A. R., & Jalil, Z. (2020, November 5). A Novel Ensemble Machine Learning Method to Detect Phishing Attack. Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020. <https://doi.org/10.1109/INMIC50486.2020.9318210>
- Bin Alam, M. S., Patwary, M. J. A., & Hassan, M. (2021). Birth Mode Prediction Using Bagging Ensemble Classifier: A Case Study of Bangladesh. *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. IEEE.
- Darmawan, Z. M. E., & Fauzan Dianta, A. (2023). Implementasi Optimasi Hyperparameter GridSearchCV Pada Sistem Prediksi Serangan Jantung Menggunakan SVM. *Online Teknologi: Jurnal Ilmiah Sistem Informasi*, 13(1).
- Espinoza, B., Simba, J., Fuertes, W., Benavides, E., Andrade, R., & Toulkeridis, T. (2019). Phishing Attack Detection: A Solution Based on the Typical Machine

- Learning Modeling Cycle. *2019 International Conference on Computational Science and Computational Intelligence* (CSCI). IEEE.
- Gutierrez-Espinoza, L., Abri, F., Siami Namin, A., Jones, K. S., & Sears, D. R. W. (2020). Ensemble Learning for Detecting Fake Reviews. *2020 IEEE 44th Annual Computers, Software, and Applications Conference* (COMPSAC). IEEE.
- Homepage, J., & Aji Prasetyo, S. (2022). IJCIT (Indonesian Journal on Computer and Information Technology) Pembuatan Website Untuk Deteksi Penyakit Umum Menggunakan Metode Certainty Factor. In IJCIT (Indonesian Journal on Computer and Information Technology) (Vol. 7, Issue 1).
- Kouate, P. M. (2020, September 11). Machine Learning: GridSearchCV & RandomizedSearchCV. <Https://Towardsdatascience.Com/Machine-Learning-Gridsearchcv-Randomizedsearchcv-D36b89231b10>.
- Purwanto, R., Paly, A., Blair, A., & Jha, S. (2020). PhishZip: A New Compression-based Algorithm for Detecting Phishing Websites. *2020 IEEE Conference on Communications and Network Security* (CNS). IEEE.
- Rangga Gelar Guntara. (2023). Aplikasi Deteksi Phising Berbasis Android Menggunakan Metode Pengembangan Perangkat Lunak DSRM. *Jurnal Minfo Polgan*, 12(1), 303–310. <https://doi.org/10.33395/jmp.v12i1.12379>
- Romzi, M., & Kurniawan, B. (2020). PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA (Issue 2).
- Rozi, F., Haryanti, T., & Fahriani, N. (2022). RANCANG BANGUN WEBSITE PROFIL SEKOLAH TAUD-SAQU ASHABUL QUR'AN SURABAYA BERBASIS HTML. In *Jurnal Ilmiah Computing Insight* (Vol. 4, Issue 1).

Sari, I. P., Qathrunada, F., Lubis, N., & Anggraini, T. (2022). Perancangan Sistem Absensi Pegawai Kantoran Secara Online pada Website Berbasis HTML dan CSS.

Staff, R. (2023). Tutorial Membuat Python Virtual Environment dan Contohnya.
<Https://Revou.Co/Panduan-Teknis/Python-Virtual-Environment>.

Susilawati, T., Yuliansyah, F., Romzi, M., & Aryani, R. (2020). MEMBANGUN WEBSITE TOKO ONLINE PEMPEK NTHREE MENGGUNAKAN PHP DAN MYSQL. JTIM: Jurnal Teknik Informatika Mahakarya.

Wibisono, Y. (n.d.). Dataset Klasifikasi Bahasa Indonesia (SMS Spam) & Klasifikasi Teks dengan Scikit-Learn. <Https://Yudiwbs.Wordpress.Com/2018/08/05/Dataset-Klasifikasi-Bahasa-Indonesia-Sms-Spam-Klasifikasi-Teks-Dengan-Scikit-Learn/>.

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross- Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing* (2018). Springer.