EXTREME LEARNING MACHINE BERBASIS RECURRENT PADA DETEKSI PHISHING EMAIL

PROPOSAL TESIS

Untuk memenuhi sebagian persyaratan
memperoleh gelar Magister Komputer

Disusun oleh:
Wanda Athira Luqyana
NIM: 186150100111002

PROGRAM STUDI MAGISTER ILMU KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan tesis dengan judul "Extreme Learning Machine Berbasis Recurrent Pada Deteksi Phishing Email" dengan baik. Tesis ini disusun dan diselesaikan sebagai syarat dalam memperoleh gelar magister pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Proses penyelesaian tesis telah mendapat banyak bantuan maupun dukungan baik secara moral maupun materiil dari berbagai pihak. Oleh karena itu penulis mengucapkan banyak terima kasih kepada:

1. Bapak Dr. Eng. Fitra A. Bachtiar, S.T., M.Eng, selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan, ilmu, dan masukan dalam menyelesaikan tesis ini.
2. Ibu Dr. Lailil Muflikhah, S.Kom., M.Sc., selaku Dosen Pembimbing II yang telah memberikan bimbingan, arahan, ilmu, dan masukan dalam menyelesaikan tesis.
3. Kedua orangtua, suami, anak, serta adik penulis yang telah memberikan doa maupun dukungan kepada penulis dalam membantu kelancaran pengerjaan tesis penulis.
4. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agung Setia Budi, S.T., M.T., Ph.D selaku Ketua Program Studi Magister Fakultas Ilmu Komputer Universitas Brawijaya.
6. Seluruh bapak dan ibu dosen yang telah mendidik dan memberikan ilmu selama penulis menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
7. Teman-teman Program Studi Magister yang selalu memberikan semangat, dukungan, serta doa selama penulis menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
8. Serta seluruh pihak yang tidak dapat disebutkan oleh penulis satu per satu yang telah membantu kelancaran pengerjaan skripsi.

Semoga seluruh doa dan dukungan yang telah diberikan dibalas oleh Allah SWT. Penulis menyadari bahwa tesis ini tidak lepas dari kekurangan baik dalam format maupun isi. Oleh karena itu diharapkan kritik maupun saran yang membangun dalam proses untuk memperbaiki diri. Penulis berharap semoga skripsi yang telah ditulis dapat memberikan manfaat bagi semua pihak.

ABSTRAK

Email merupakan media terpenting pada masa kini, penggunaannya telah dibutuhkan untuk berbagai bentuk aktifitas sehari-hari manusia. Seluruh akses baik finansial maupun non-finansial telah membutuhkan *email* baik pribadi maupun non-pribadi. Sehingga tidak mengherankan bahwa *email* menjadi sasaran empuk untuk penyerangan kejahatan siber. *Phishing email* menjadi salah satu bentuk kejahatan siber yang telah mendunia. Berdasarkan data statistik didapatkan 55,53% telah dikirimkan *phishing email* di Indonesia. Berbagai kerugian finansial-pun telah berdampak pada korban. Upaya minimalisir *phishing email* telah dilakukan sejak dahulu namun seiring berkembangnya ilmu pengetahuan, *phishing email* juga telah berkembang. Sehingga diperlukan pula upaya yang telah dikembangkan untuk mengatasinya. *Recurrent Extreme Learning Machine Neural Network* (RELMNN) menjai metode yang diusulkan untuk mengatasi *phishing email*. Metode ELM yang berbasis *recurrent* dan dikombinasikan dengan *Pricipal Component Analysis* (PCA) sebagai metode untuk mereduksi dimensi pada teks dapat menghasilkan deteksi *phishing email* secara akurat dengan waktu komputasi yang singkat.

Kata kunci: *phishing email*, *cyber crime*, *Recurrent Extreme Learning Machine Neural Network*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR PERSAMAAN

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

1.1 Latar belakang

Teknologi Informasi dan Komunikasi (TIK) telah menjadi hal yang umum untuk digunakan. Seluruh masyarakat dari berbagai usia sudah tidak asing lagi dengan kehadiran teknologi, khususnya dimasa pandemi Covid-19 yang meningkatkan penggunaan teknologi. Masyarakat di Indonesia baik yang masih berada di usia sekolah TK, SD, SMP, SMA, Mahasiswa, para pekerja kantoran hingga orang tua yang berada di rumah sedang marak untuk berburu teknologi yang berupa gadget maupun internet. Namun kurangnya edukasi tentang penggunaan teknologi masih menjadi kendala yang dapat merugikan sebagian orang, khususnya orang tua. Berbagai macam penipuan kini banyak terjadi baik melalui email, pesan singkat, maupun media sosial.

Phishing email menjadi salah satu permasalahan pencurian data yang terjadi di seluruh bagian negeri dan telah memakan banyak korban. Berdasarkan data yang dikumpulkan oleh Pusopskamsinas, diketahui bahwa sebanyak 2549 kasus *phishing email* terjadi di Indonesia pada tahun 2020. Persentase kejadian *phishing email* di Indonesia meningkat semenjak pandemi Covid-19 dimulai. Dikatakan bahwa terdapat 55,53% *phishing email* yang dikirimkan pada jam kerja dan 44,37% diluar jam kerja (Anisatul Umah, 2021). Kurangnya pengetahuan mengenai keamanan data menjadi suatu alasan utama mengapa *phishing email* masih memakan korban, sehingga dalam menangani permasalahan tersebut penelitian untuk meminimalisir korban dari *phishing email* diperlukan. Dampak finansial telah dirasakan dari berbagai bisnis di Amerika. Berdasarkan keterangan FBI total kerugian telah mencapai 1,8 Juta dollar Amerika di Tahun 2020. Selain itu kerugian finansial secara personal dirasakan oleh salah satu orang yang terkait dengan *real estate*. Korban telah mengalami kerugian sebesar 22,893 dollar Amerika (Walser, 2021). Pada umumnya kerugian yang diakibatkan oleh *phishing email* berupa kerugian finansial, sehingga diperlukan suatu solusi yang dapat meminimalisir terjadinya *phishing* yang dapat menyerang suatu bisnis maupun personal. *Text mining* menjadi salah satu cara yang daapt mengidentifikasi *phishing email,* sehingga dapat meminimalisir adanya penipuan.

Text mining menjadi salah satu alternative yang dapat digunakan untuk mempermudah mengatasi *phishing email* yang marak terjadi. *Text mining* merupakan proses dalam mengubah data tidak terstruktur (teks) menjadi data yang terstruktur, sehingga dapat diketahui pola yang bermakna pada data tersebut. Berbagai macam pengaplikasian dari *text mining* telah diterapkan untuk menyelesaikan permasalahan, diantaranya adalah meningkatkan *customer experience* dengan mengolah data yang didapat berdasarkan *online review*. Selain itu *text mining* dapat mengatasi permasalahan manajemen resiko dengan memberikan wawasan terkait tren industry dan pasar keuangan dengan memantau pergeseran sentiment dan mengekstrak informasi dari laporan analisis. Sehingga*,* penggunaan *text mining* menjadi solusi yang efektif dan merupakan

tindakan yang preventif untuk meminimalisir kerugian yang terjadi pada setiap calon korban *phishing email*.

Penelitian pada kasus *phishing email* telah banyak dilakukan, pendekatan *sender centric* menjadi salah satu metode yang diimplementasikan. Penelitian tersebut telah membuktikan bahwa dengan mengunakan pendekatan tersebut 98,7% dataset Nazarion telah diklasifikasikan sebagai *phishing email*. Namun akurasi yang diraih pendekatan *sender centric* hanya menyelesaikan pada data *banking* saja (Sanchez & Duan, 2012). Selain itu penelitian *phishing* dilakukan dengan membandingkan beberapa algoritme dengan mengimplementasikan *Lexical Feature*. Hasil penelitian yang didapatkan bahwa SVM yang mengimplementasikan *Lexical Feature* memiliki tingkat akurasi yang paling baik (Sindhu et al., 2020). Penelitian lain terkait *phishing email* dilakukan pula dengan membandingkan metode *machine learning* dan *deep learning* dalam memberikan solusi terhadap masalah. Dalam penelitian tersebut memiliki kesenjangan terhadap jumlah dataset yang digunakan yaitu, 3416 sebagai email yang berlabel *phishing email* dan 14950 merupakan *regular email* (Bagui et al., 2019).

Teks memiliki fitur data yang tinggi untuk dikomputasi, sehingga diperlukan solusi untuk mereduksi dimensi pada teks agar mempersingkat waktu komputasi dan memperbaiki tingkat akurasi. Penelitian terkait reduksi dimensi data telah dilakukan pada *phishing email.* Penelitian yang dilakukan dengan membandingkan teknik ekstraksi fitur dan seleksi fitur. Didapatkan berdasarkan hasil penelitian bahwa ekstraksi fitur *Principal Component Analysis* (PCA) dan L*atent Semantic Analysis* (LSA) merupakan pilihan yang tepat untuk memperbaiki performa klasifikasi (Zareapoor & K. R, 2015). Penelitian terkait ekstraksi fitur dilakukan juga untuk memprediksi penyakit diabetes. PCA dilakukan untuk meningkatkan kinerja dari metode K-Means yang membuktikan bahwa 25 data telah diklasifikasikan lebih benar (Zhu et al., 2019).

Berbagai penelitian terkait *phishing email* telah dilakukan oleh peniliti sebelumnya. Berbagai penelitian tersebut masih memiliki kekurangan dari segi waktu komputasi yang masih rendah. Selain itu masih terdapat permasalahan dengan objek teks. Permasalahan tersebut adalah teks memiliki banyak fitur sehingga dimensi data masih tinggi. Hal ini menjadi kekurangan dalam melakukan komputasi untuk menyelesaikan permasalahan. Oleh karena itu diperlukan suatu metode yang dapat mereduksi dimensi data yang tinggi dalam kasus teks.

Selain penelitian terkait objek *phishing email* dan ekstraksi fitur, *Extreme Learning Machine* ialah metode implementasi dari beberapa penelitian yang dibahas. Penelitian yang telah dilakukan adalah deteksi *phishing* website yang mengimplementasikan *improved Extreme Learning Machine*. Hasil penelitian memberikan akurasi terbaik dengan mengimplementasikan ADASYN + SDAE + NIOSELM secara berurutan (Yang et al., 2021). Penelitian ELM dilakukan pula pada permasalahan *imbalanced data*. ELM yang diimplementasikan setelah dilakukan *random sampling data*. Evaluasi G-Mean menunjukkan hasil yang baik dengan metode ELM yang datanya telah diseimbangkan menggunakan teknik *resampling* (Wang & Xing, 2016). Pengembangan metode ELM dilakukan pula dengan

menggunakan proses *recurrent*. Penelitian yang dilakukan untuk meramalkan beban listrik menggunakan metode Recurrent Extreme Learning Machine Neural Network. Berdasarkan hasil penelitian didapatkan bahwa nilai RMSE lebih tinggi dibandingkan dengan metode ELM, RNN, LR, kSR, kNNR, GPR dan GRNN dengan kecapatan komputasi yang relatif sama dengan ELM (Ertugrul, 2016).

ELM merupakan metode yang telah tepat dalam menyelesaikan berbagai macam kasus tak terkecuali pada permasalahan teks. Pengembangan dari metode ELM yaitu dengan menambahkan proses *recurrent.* Proses *recurrent* yang dikembangkan pada jaringan ELM berkerja dengan mengembalikan *output* sebagai *input* baru pada proses berikutnya. Sehingga proses penelitian pada permasalahan teks akan menggunakan *Recurrent Extreme Learning Machine Neural Network* yang berbasis *recurrent* sehingga dapat menyeleasikan permasalahan pada deteksi *phishing*. RELMN merupakan metode yang telah digunakan pada berbagai masalah pada penelitian yang berbasis *time series* yang belum memerhatikan banyaknya dimensi data dan waktu komputtasi. Tingginya dimensi data pada kasus teks kemudian diatasi dengan mengimplementasikan metode ekstrasi fitur PCA untuk mereduksi dimensi data, sehingga hasil penelitian diharapkan dapat menyelesaikan kasus teks dengan akurasi yang baik dan waktu komputasi yang singkat.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana pengaruh ekstraksi fitur PCA terhadap hasil klasifikasi *phishing email*?
2. Bagaimana pengaruh *hidden neuron* dan *context neuron* terhadap tingkat akurasi pada algoritme RELMNN?
3. Bagaimana kinerja metode RELMNN dibandingkan dengan metode umum seperti SVM untuk klasifikasi *phishing email*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah berikan, maka penelitian ini dilakukan dengan harapan dapat mencapai tujuan sebagai berikut:

1. Mengetahui pengaruh ekstraksi fitur terhadap hasil klasifikasi *phishing email*.
2. Mengetahui pengaruh *hidden neuron* dan *context neuron* terhadap tingkat akurasi yang dihasilkan oleh algoritme RELMNN dengan menggunakan ekstraksi fitur
3. Mengetahui kinerja RELMMNN pada *phishing email* yang dibandingkan dengan metode umum lainnya.

1.4 Manfaat

Berdasarkan tujuan yang telah ditunjukkan, diharapkan penelitian dapat memberikan manfaat sebagai berikut:

1. Sebagai bahan acuan untuk dilakukan penelitian selanjutnya dengan topik permasalahan yang sama.
2. Hasil penelitian dapat bermanfaat bagi user email dalam membedakan *phishing email* dan non *phishing email*.

1.5 Batasan masalah

Untuk melakukan penelitian secara spesifik dan jelas, diperlukan batasan yang diterapkan pada penelitian ini. Sehingga ditentukan batasan-batasan dalam melakukan penelitian, diantaranya adalah:

1. Penelitian deteksi *phishing email* diimplementasikan menggunakan bahasa Python.
2. Dataset yang digunakan diambil dari dua sumber, yaitu *Phishcorpus* dan *Enron Mail.*

1.6 Sistematika pembahasan

Sistematikan pembahasan pada penelitian ini tersusun atas enam bab yang dimulai dari bab pendahuluan hingga bab peneutup. Tujuan dari sistematika pembahasan yaitu memudahkan pembaca untuk memahami sistemastika dalam penelitian.

BAB 1 PENDAHULUAN

Bab ini membahas mengenai alasan utama dari topik yang diangkat sebagai objek penelitian. Alasan utama dituangkan dalam latar belakang masalah, rumusah masalah, tujuan dilakukannya penelitian, manfaat dari hasil penelitian, batasan masalah yang diterapkan pada penelitian, dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Membahas terkait teori-teori dasar sebagai referensi maupun pernyataan pendukung dilakukannya penelitian. Teori yang dibahas pada landasan kepustakaan meliputi *text mining, phishing email,* Extreme Learning Machine, Recurrent Extreme Learning Machine Neural Network.

BAB 3 METODOLOGI

Pada bab ini menjelaskan mengenai langkah-langkah penelitian dengan tahapan penelitian yang meliputi kebutuhan sistem, perancangan sistem, dan pengumpulan data.

BAB 4 PERANCANGAN

Menjelaskan mengenai perancangan sistem yang diterapkan pada penelitian, yaitu perancangan RELMNN pada klasifikasi *phishing email.*

BAB 5 PENGUJIAN DAN ANALISIS

Bab lima membahas terkait pengujian yang dilakukan pada sistem, seihngga dapat mengetahui tingkat keberhasilan sistem dalam memberikan suatu solusi. Hasil pengujian kemudian dianalisis untuk diketahui alasan-alasan yang terjadi terkait hasil pengujian.

BAB 6 PENUTUP

Bab enam membahas terkait setiap proses dan hasil yang didapatkan selama penelitian berlangsung akan dibahas secara ringkas pada kesimpulan. Saran yang diperoleh dari hasil penelitian akan dicantumkan dengan harapan jika terdapat penelitian serupa akan memberikan hasil yang jauh lebih baik dari penelitian sebelumnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab landasan kepustakaan disajikan beberapa teori pendukung untuk penelitian. Teori yang akan dibahas mengenai *text mining, phishing,* dan algoritme Recurrent Extreme Learning Machine Neural Network.

2.1 Kajian Pustaka

Penelitian terdahulu terkait deteksi *phishing email* telah dilakukan, diantaranya penelitian yang dilakukan oleh Sanchez & Duan ditahun 2012. Penelitian dilakukan dengan menggunakan pendekatan *sender centric* yang memfokuskan penelitian pada pengirim informasi. Penelitian tersebut dilakukan untuk mendeteksi phishing email bank dengan melakukan 2 (dua) tahap yaitu, memilah pesan *banking* dan *non-banking* menggunakan SVM sebagai algoritme klasifikasi. Pendekatan *sender centric* memiliki aturan-aturan untuk mengidentifikasi pengirim email yang mencurigakan. Hasil penelitian didapatkan bahwa SVM memiliki tingkat akurasi yang tinggi dalam melakukan klasfikasi pesan *banking* dan *non-banking*. Selain itu performa yang dihasilkan dari aturan-aturan *sender centric* dalam mendeteksi *phishing email* cukup baik. Hal tersebut ditunjukkan dengan sebesar 98,7% dataset Nazarion telah diklasifikasikan sebagai *phishing email* (Sanchez & Duan, 2012).

Deteksi *phishing* dilakukan juga pada penelitian yang membandingkan beberapa algoritme. *Random Forest*, SVM, dan *Neural Network* dengan penggunaan *Backpropagation* menjadi algoritme-algoritme terbaik untuk dilakukan penelitian. Implementasi *Lexical Feature* menjadi bagian dari proses klasifikasi yang berguna dalam menyempurnakan hasil klasifikasi dari setiap metode. Berdasarkan penelitian yang dilakukan *Lexical Feature* mampu menyempurnakan hasil klasifikasi menjadi lebih baik dan memberikan hasil akurasi terbaik pada algoritme SVM(Sindhu et al., 2020).

Penelitian phishing email dilakukan juga pada teknik reduksi dimensi. Dalam mendeteksi *phishing email* memiliki kesulitan dikarenakan data yang diteliti merupakan data teks yang umumnya memiliki dimensi data yang tinggi. Mengatasi permasalahan tersebut penelitian terkait reduksi dimensi data dilakukan dengan membandingkan dua teknik reduksi dimensi yaitu, teknik ekstraksi fitur dan seleksi fitur. Pada penelitian tersebut digunakan *Chi-Square* dan *Information Gain Ratio* sebagai teknik seleksi fitur, sedangkan untuk ekstraksi fitur menggunakan *Principal Component Analysis* (PCA) dan *Latent Semantic Analysis* (LSA). Tahapan penilitian dilakukan dengan mengimplementasikan J48 *Decision Tree* sebagai metode klasifikasi. Hasil penelitian didapatkan bahwa penggunaan teknik ekstraksi fitur PCA dan LSA memberikan performa yang baik pada hasil klasifikasi *phishing email* (Zareapoor & K. R, 2015).

Selain itu penelitian terkait algoritme *machine learning* telah dilakukan. Salah satunya penelitian yang dilakukan untuk mendeteksi *phishing* pada *website* dengan mengimplementasikan *improved Extreme Learning Machine* untuk

mengoptimalkan proses *training*, sehingga dapat berjalan dengan cepat dan mendapatkan hasil akurasi yang optimal. Penelitian dilakukan menggunakan 60000 *dataset website* normal dan 5000 *website phishing*. Peneliti menerapkan 3 (tiga) aspek untuk proses ekstraksi, yaitu *surface features*, *topological features*, dan *deep features*. Penelitian melakukan pengujian pada algoritme *Stacked Denoising Auto Encoder* (SDAE) yang terbukti mampu untuk mereduksi dimensi karena nilai error rendah. Pengujian dilakukan pada algoritme klasifier yang membandingkan NIOSELM + SDAE + ADASYN. Hasil akurasi terbaik didapatkan pada proses deteksi yang mengimplementasikan ADASYN + SDAE + NIOSELM secara berurutan (Yang et al., 2021).

Algoritme ELM menjadi metode *machine learning* yang digunakan pada permasalahan *imbalanced data*. Penelitian dilakukan dengan teknik resampling sebagai solusi untuk mengatasi *imbalanced data*, sehingga kelas dengan jumlah data yang sedikit dapat tetap memberikan informasi yang layak. Proses penelitian dilakukan dengan mengimplementasikan algoritme ELM secara umum pada data asli (tanpa diubah), ELM yang diimplementasikan setelah *dilakukan random sampling data*, dan implementasi ELM yang telah diintegrasikan dengan metode resampling. Proses *resampling* pada penelitian tersebut melakukan dengan metode *under-sampling*, dimana dataset kelas terbesar akan disamakan jumlahnya dengan *dataset* kelas terkecil yang dilakukan sebanyak M-kali untuk mencapai titik *equilibrium* (Wang & Xing, 2016).

Pada tahun 2016 telah dilakukan penelitian terkait pengembangan ELM dilakukan dengan mengimplementasikan metode *recurrent*. Penelitian dilakukan pada meramalkan beban listrik yang mengimplementasikan metode Recurrent Extreme Learning Machine Neural Network RELMNN. Berdasarkan hasil penelitian didapatkan bahwa nilai RMSE lebih tinggi dibandingkan dengan metode ELM, RNN, LR, kSR, kNNR, GPR dan GRNN dengan kecapatan komputasi yang relatif sama dengan ELM (Ertugrul, 2016).

2.2 Phishing

Phishing merupakan kata yang mulanya muncul di tahun 1990. Peretas menggunakan “ph” sebagai pengganti ‘f’ yang berfungsi untuk mengubah *fishing* menjadi *phishing*. Kata tersebut memiliki maksut memancing, hal tersebut mengacu pada tindakan penyerang untuk menarik pengguna agar mengunjungi *website* palsu memalui *email* palsu sehingga mendapatkan informasi pribadi korban (Chawla & Singh Chouhan, 2014). Tindakan *phishing* memiliki definisi yang tidak pasti pada mulanya. Namun dengan dengan seiring dengan berkembangnya zaman, serangan *phishing* memiliki makna bahwa merupakan proses dari tindakan untuk mengelabui penerima agar melakukan sesuatu sesuai dengan kehendak dari penyerang (Alkhalil et al., 2021). Selain itu definisi *phishing* lainnya ialah tindakan yang tergolong sebagai *cybercrime* atau kejahatan dunia maya oleh seseorang yang mengaku sebagai lembaga terpercaya agar memberikan informasi terkait data pribadi. Pelaku menghubungi korbannya melalui email, telepon, maupun

pesan teks agar mendapatkan identitas pribadi, informasi terkain perbankan, serta kata sandi (KnowBe4, 2021). Kasus terkait gugatan mengenai *phishing* pertama kali pada tahun 2004, yang dilakukan oleh seorang remaja. Remaja tersebut membuat tiruan *website* "America Online" sehingga dapat memperoleh informasi dan mengakses kartu kredit para korbannya (KnowBe4, 2021).

Serangan *phishing* dilakukan dengan dua tipe yaitu, *deceptive phishing* dan *technical subterfuge*. *Deceptive phishing* merupakan tipe paling umum yang dilakukan untuk mengelabui korbannya. Pada tipe ini *phisher* atau pelaku *phishing* menggunakan trik *social engineering* dengan cara membuat scenario atau menggunakan metode teknis dalam memikat korban, hal ini dilakukan dengan meyakinkan korban mengenai keabsahan email palsu (Alkhalil et al., 2021).

Jenis-jenis *deceptive phishing* ialah:

a. *Phishing Email*. Merupakan email palsu yang dikirimkan kepada korban secara acak oleh pihak yang berpura-pura sebagai institusi ataupun orang yang terpecaya untuk memikat korban agar memberikan informasi pribadi.
b. *Spoofed website.* Phishers menggunakan website palsu yang menyerupai website asli.
c. *Phone phishing. Phishing* yang dilakukan melalui panggilan telepon atau pesan teks, yang mana *phishers* mengaku sebagai seseorang yang dikenal oleh korban ataupun sebagai sumber terpercaya. Hal ini dilakukan untuk memandu korban agar memberikan *password* ataupun nomor PIN akun bank.
d. *Social media.* Phishing dilakukan dengan membajak akun media sosial, peniruan identitas, penipuan, dan penyebaran malware.

Selain *deceptive phishing*, terdapat jenis-jenis *phishing* dari tipe *technical subterfuge*. Jenis-jenis *technical subterfuge* ialah:

a. Malware Based Phishing. *Phishing* ini dilakukan dengan menjalankan software berbahaya yang telah didownload oleh korban, sehingga dapat menyerang mesin milik korban.
b. DNS- Based Phishing. *Phishing* yang mengganggu sistem nama domain sehingga korban diarahnya ke situs web berbahaya dengan mencemari cache DNS.
c. Content Injection. *Phishing* yang dilakukan dengan memasukkan konten palsu pada website asli atau website yang sah.
d. Man in the Middle. *Phishers* menyisipkan komunikasi antar dua pihak (korban dan website sah) untuk mendapatkan informasi dari kedua belah pihak.
e. Search Engine Phishing. Teknik yang dilakukan dengan membuat website berbahaya dengan tawaran menarik dan menggunakan taktik Search

Engine Optimization agar terindex secara legal, sehingga dapat muncul ketika korban melakukan pencarian produk ataupun layanan.

f. URL Attacks. Teknik yang dilakukan dengan meyakinkan korban untuk mengakses link yang telah terhubung dengan server berbahaya.

Phishing email merupakan tindakan criminal yang dilakukan oleh *phisher* untuk mengejar keuntungan finansial baik secara langsung maupun tidak. Kejahatan yang dilakukan kepada siapapun dengen mengirimkan email sebanyak mungkin dengan harapan ada orang yang terkena umpan dari *phisher*. Berbagai jenis *email* dikirimkan sebagai umpan untuk memakan korban, sebagai contohnya banyak *phisher* melakukan rekayasa sosial yang berpura-pura sebagai CEO sehingga efektif untuk mereka menjaring korban (Datacom, 2020).

2.3 *Text Mining*

Text mining adalah ilmu yang bertujuan untuk memproses teks agar menjadi informasi yang diperoleh dari peramalan pola dan kecenderungan melalui pola statistik. Teks yang diolah bisa berupa teks terstruktur dan teks tidak terstruktur. *Text mining* mengacu pada *information retrieval*, *data mining, machine learning*, statistik dan komputasi linguistic (Jiawei et al., 2012). *Text mining* bertujuan untuk menganalisis pendapat, sentimen, evaluasi, sikap, penilaian, emosi seseorang seihngga dapat diketahui apakah berkenaan dengan suatu topik, layanan, organisasi, individu, atau kegiatan tertentu (Liu, 2012). Penggunaan dari *text mining* dilakukan untuk klasterisasi, klasifikasi*, information retrieval*, dan *information extraction* (Kogan & Berry, 2010).

2.4 Pre-processing

Pre-processing merupakan tahap awal dari *text mining* untuk mengubah data sesuai dengan format yang dibutuhkan. Proses ini dilakukan untuk menggali, mengolah dan mengatur infomasi dan untuk menganalisis hubungan tekstual dari data terstruktur dan data tidak testruktur (Nugroho, 2016). Persiapan data dilakukan untuk diolah pada *knowledge discovery*. Tahapan dari *Pre-processing* meliputi *case folding, data cleaning,* normalisasi bahasa*, stopword removal, stemming,* tokenisasi.

2.4.1 Case Folding

Tahap awal adalah *case folding* yang bertujuan untuk mengubah setiap bentuk kata menjadi sama. Hal ini dilakukan dengan mengubah kata menjadi *lower case* atau huruf kecil. Contoh dari *case folding* ditunjukkan pada Tabel 2.1.

Tabel 2.1 *Case Folding*

Data Awal	Data *Case Folding*
Best regards MR.Nardhamuni Ganas Chief Auditor Ned Bank	best regards mr.nardhamuni ganas chief auditor ned bank

PRIVILEGE AND CONFIDENTIALITY NOTICE: The information contained in this e-mail is privileged and confidential and is for the exclusive use of the addressee(s). Any person who receives this e-mail is entitled to handle it over to the addressee, informed that such person may not, disclose or reproduce the contents thereof. If you have received this communication, please notify without delay or delete the message if not interested	privilege and confidentiality notice: the information contained in this e-mail is privileged and confidential and is for the exclusive use of the addressee(s). any person who receives this e-mail is entitled to handle it over to the addressee, informed that such person may not, disclose or reproduce the contents thereof. if you have received this communication, please notify without delay or delete the message if not interested

2.4.2 Cleaning Data

Cleaning data merupakan proses pembersihan kata dengan menghilangkan delimiter koma (,), titik (.), dan tanda baca lainnya. Pembersihan kata bertujuan untuk mengurangi *noise*.

Tabel 2.2 *Cleaning Data*

Data Awal	***Cleaning Data***
best regards mr.nardhamuni ganas chief auditor ned bank privilege and confidentiality notice: the information contained in this e-mail is privileged and confidential and is for the exclusive use of the addressee(s). any person who receives this e-mail is entitled to handle it over to the addressee, informed that such person may not, disclose or reproduce the contents thereof. if you have received this communication, please notify without delay or delete the message if not interested	best regards mr nardhamuni ganas chief auditor ned bank privilege and confidentiality notice the information contained in this email is privileged and confidential and is for the exclusive use of the addressees any person who receives this e-mail is entitled to handle it over to the addressee informed that such person may not disclose or reproduce the contents thereof if you have received this communication please notify without delay or delete the message if not interested

2.4.3 Stopword Removal

Stopword merupakan daftar kata umum yang tidak memiliki arti penting dan tidak digunakan. Pada proses ini kata umum akan dihapus untuk mengurangi jumlah kata yang disimpan oleh sistem (Manning et al., 2009).

Tabel 2.3 *Stopword Removal*

Data Awal	Data *Stopword Removal*
best regards mr nardhamuni ganas chief auditor ned bank privilege and confidentiality notice the information contained in this email is privileged and confidential and is for the exclusive use of the addressees any person who receives this e-mail is entitled to handle it over to the addressee informed that such person may not disclose or reproduce the contents thereof if you have received this communication please notify without delay or delete the message if not interested	best regards mr nardhamuni ganas chief auditor ned bank privilege confidentiality notice information contained email privileged confidential exclusive addressees person receives e-mail entitled handle addressee informed person disclose reproduce contents thereof received communication please notify without delay delete message interested

2.4.4 Tokenization

Tokenization adalah proses untuk memotong document menjadi pecahan kecil yang dapat berupa bab, sub-bab, paragraf, kalimat, dan kata (token). Pada proses ini akan menghilangkan *whitespace.*

Tabel 2.4 *Tokenization*

Data Awal	Data *Tokenization*
best regards mr nardhamuni ganas chief auditor ned bank privilege confidentiality notice information contained email privileged confidential exclusive addressees person receives e-mail entitled handle addressee informed person disclose reproduce contents thereof received communication please notify without delay delete message interested	[‘best’, ‘regards’, ‘mr’, ‘nardhamuni’, ‘ganas’, ‘chief’, ‘auditor’, ‘ned’, ‘bank’, ‘privilege’, ‘confidentiality’, ‘notice’, ‘information’, ‘contained’, ‘email’, ‘privileged’, ‘confidential’, ‘exclusive’, ‘addressees’, ‘person’, ‘receives’, ‘e-mail’, ‘entitled’, ‘handle’, ‘addressee’, ‘informed’, ‘person’, ‘disclose’, ‘reproduce’, ‘contents’, ‘thereof’, ‘received’, ‘communication’, ‘please’, ‘notify’, ‘without’, ‘delay’, ‘delete’, ‘message’, ‘interested’]

2.4.5 Pembobotan TF-IDF

Pembobotan *Term Frequency-Inverse Document Frequency* (TF-IDF) adalah metode yang digunakan untuk menghitung bobotan setiap kata yang telah diekstrak. Penggunaan metode ini umumnya dilakukan untuk mengihitung kata umum yang ada pada *information retrieval*. Model pembobotan TF-IDF merupakan metode yang mengintegrasikan model *term frequency (tf)* dan *inverse document frequency (idf),* dimana *term frequency (tf)* merupakan proses untuk menghitung jumlah kemunculan term dalam satu dokumen dan *inverse document frequency (idf)* digunakan untuk menghitung term yang muncul di berbagai dokumen(komentar) yang dianggap sebagai term umum, yang dinilai tidak penting (Akbari et al., 2012).

Proses awal yang dilakukan dalam pembobotan TF-IDF dilakukan dengan menghitung *term frequency* $tf_{t,d}$. Dimana t menunjukkan term dalam dokumen d yang berfungsi untuk menunjukkan kemunculan term t pada dokumen d. Hal ini berpengaruh dalam bobot term yang akan semakin tinggi ketika banyak term yang muncul dalam suatu dokumen. Nilai dari *tf* akan dihitung bobotnya dengan rumus *weighting term frequency (*W_{tf}*)*. Rumus tersebut ditunjukkan pada persamaan 2.1.

$$\boldsymbol{W_{tf_{t,d}}} = \begin{cases} \mathbf{1} + \boldsymbol{log_{10}\, tf_{t,d}}\,, & \boldsymbol{if\ tf_{t,d} > 0} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \tag{2.1}$$

Banyaknya kata yang muncul pada dokumen, umumnya merupakan nilai *term frequency* dari kata yang tidak penting. Untuk menghindari pembobotan pada kata tidak penting maka digunakan pembobotan *document frequency* yang bermaksud untuk menghitung jumlah dokumen yang mengandung term t.

Dari nilai term pada setiap dokumen yang telah ditemukan akan dilakukan proses kebalikan dari pembobotan *document frequency*. Proses pembobotan ini disebut dengan *inverse document frequency*, yang menyatakan bahwa frekuensi dari term yang rendah pada banyak dokumen akan memberikan bobot paling tinggi. Perhitungan ini ditunjukkan dengan rumus persamaan 2.2.

$$\boldsymbol{idf_t} = \boldsymbol{log_{10}} \frac{\boldsymbol{N}}{\boldsymbol{df_t}} \tag{2.2}$$

Perhitungan pembobotan TF-IDF merupakan perkalian yang dilakukan dari pembobotan term *frequency* dengan *inverse document frequency*. Hal ini ditujunkkan pada rumus persamaan 2.3.

$$\boldsymbol{W_{t,d}} = \boldsymbol{W_{tf_{t,d}}} \times \boldsymbol{idf_t} \tag{2.3}$$

Keterangan:

$W_{tf_{t,d}}$ = bobot kata dalam setiap dokumen

$tf_{t,d}$ = jumlah kemunculan kata t pada dokumen d

N = jumlah dokumen pada kumpulan dokumen

df = jumlah dokumen yang mengandung term

idf_t = bobot inverse dari nilai df

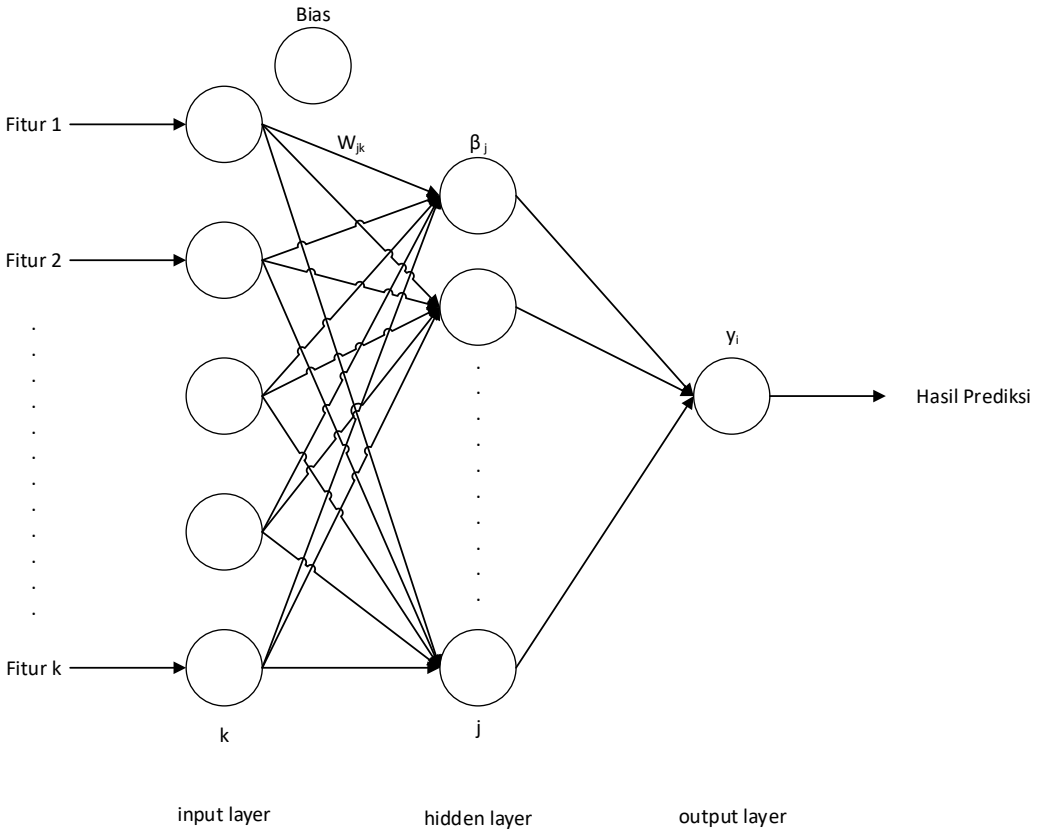
$W_{t,d}$ = pembobotan TF-IDF

2.5 Extreme Learning Machine

Extreme Learning Machine atau dikenal dengan ELM merupakan algoritme pembelajaran yang sederhana dan efisien untuk jaringan syaraf feedforward dengan single layer. ELM diusulkan sebagai metode yang dapat memperbaiki kekurangan yang ada pada algoritme *backpropagation*. Metode pembelajaran tersebut diperkenalkan oleh Huang dkk yang memiliki keunggulan pada *learning speed* (Huang et al., 2006). Proses perhitungan pada ELM dilakukan dengan inisialisasi parameter *weight* dan bias secara random sehingga memiliki *learning speed* yang cepat.

2.5.1 Arsitektur Jaringan ELM

Setiap metode Jaringan Syaraf Tiruan memiliki arsitektur jaringan yang berbeda, ELM memiliki arsitektur jaringan dengan satu *hidden layer*. Gambar 2.1 menunjukkan arsitektur jaringan dari ELM.



Gambar 2.1 Arsitektur Jaringan ELM

Sumber: (Cholissodin et al., 2017)

2.5.2 Proses *Training*

Proses *training* dilakukan untuk memberikan pelatihan dengan menggunakan data latih, sehingga pada proses *training* mendapatkan nilai bobot yang optimal.

Tahapan yang dilakukan pada proses *training* pada Extreme Learning Machine adalah:

1. Inisialisasi nilai secara random pada matriks W_{jk} yang berperan sebagai nilai input bobot yang dibatasi dengan nilai *range* [-1, 1]. Nilai random tersebut disimpan dalam bentuk array berukuran *j* (jumlah *hidden neuron*) x *k* (jumlah *input neuron*). Kemudian inisialisasi nilai secara random untuk matriks bias b dengan *range* nilai [0,1] yang berukuran 1x (jumlah *hidden neuron*).
2. Hitung matriks H_{train} yang ditunjukkan pada Persamaan 2.4.

   $$H_{train} = g\ (\sum_{k=1}^{n} X_{train_{ik}} .\ {W_{jk}}^{T} + b_j) \quad (2.4)$$

   Keterangan:

   $\mathrm{H}_{\mathrm{train}}$: Matriks *output hidden layer* pada *training*

   X_{train} : Data *training*

   g : Fungsi aktivasi

   W^T : Transpose matriks *input weights*

   b : Bias

   i : Urutan data

   j : Jumlah *hidden neuron*

   k : Jumlah *input neuron*
3. Hitung matriks ${H_{train}}^{+}$ *Moore-Penrose Generalized Inverse* menggunakan Persamaan 2.5.

   $$\boldsymbol{{H_{train}}^{+}} = (\boldsymbol{{H_{train}}^{T}} . \mathrm{H}_{\mathrm{train}})^{-1} .\ \boldsymbol{{H_{train}}^{T}} \quad (2.5)$$

   Keterangan:

   ${H_{train}}^{+}$: Matriks *Moore-Penrose Generalized Inverse*

   ${H_{train}}^{T}$: Transpose matriks H_{train}
4. Hitung matriks *Beta* ($\hat{\beta}$) menggunakan Persamaan 2.6.

   $$\boldsymbol{\hat{\beta}} = \boldsymbol{{H_{train}}^{+} . T_{train}} \quad (2.6)$$

   Keterangan:

   $\hat{\beta}$: Matriks *output weights*

   T_{train} : Matriks label *training*
5. Hitung matriks $\hat{Y}_{train}$ menggunakan Persamaan 2.7.

   $$\boldsymbol{\hat{Y}_{train}} = \boldsymbol{H_{train} . \hat{\beta}} \quad (2.7)$$

   Keterangan:

   $\hat{Y}_{train}$: Matriks hasil prediksi *training*

6. Menentukan label hasil *training* menggunakan Persamaan 2.8.

$$\boldsymbol{L_{train} = arg^{row}_{max}\,(Y_{train})} \quad (2.8)$$

Keterangan:

L_{test} : Label *output classifier* ELM

arg^{row}_{max} : Mengambil indeks nilai maksimum

7. Hitung evaluasi *training*

2.5.3 Proses *Testing*

Proses *testing* dilakukan setelah proses *training* dilakukan dengan menggunakan data uji. Proses *testing* dilakukan untuk menguji hasil *training* sehingga dapat diketahi akurasi dari program yang digunakan. Tahapan dari proses *testing* adalah:

1. Diketahui nilai bobot W_{jk}, bias b, dan beta $\hat{\beta}$ yang didapatkan dari hasil *training* untuk diinputkan dalam proses *testing.*
2. Hitung matriks H_{test} yang ditunjukkan pada Persamaan 2.9.

$$\boldsymbol{H_{test} = g\,(\sum_{k=1}^{n} X_{test_{ik}} \,.\, W_{jk}{}^{T} + b_j)} \quad (2.9)$$

Keterangan:

$\mathrm{H_{test}}$: Matriks *output hidden layer* pada *testing*

X_{test} : Data *testing*

g : Fungsi aktivasi

W^T : Transpose matriks *input weights*

b : Bias

i : Urutan data

j : Jumlah *hidden neuron*

k : Jumlah *input neuron*

3. Hitung matriks $\hat{Y}_{test}$ dengan nilai $\hat{\beta}$ yang telah diketahui melalui proses *training.* Perhitungan dilakukan menggunakan Persamaan 2.10.

$$\boldsymbol{\hat{Y}_{test} = H_{test} \,.\, \hat{\beta}} \quad (2.10)$$

Keterangan:

$\hat{Y}_{test}$: Matriks hasil prediksi *testing*

4. Menentukan label hasil *testing* menggunakan Persamaan 2.11.

$$\boldsymbol{L_{test} = arg^{row}_{max}\,(Y_{test})} \quad (2.11)$$

Keterangan:

L_{test} : Label *output classifier* ELM

arg_{max}^{row} : Mengambil indeks nilai maksimum

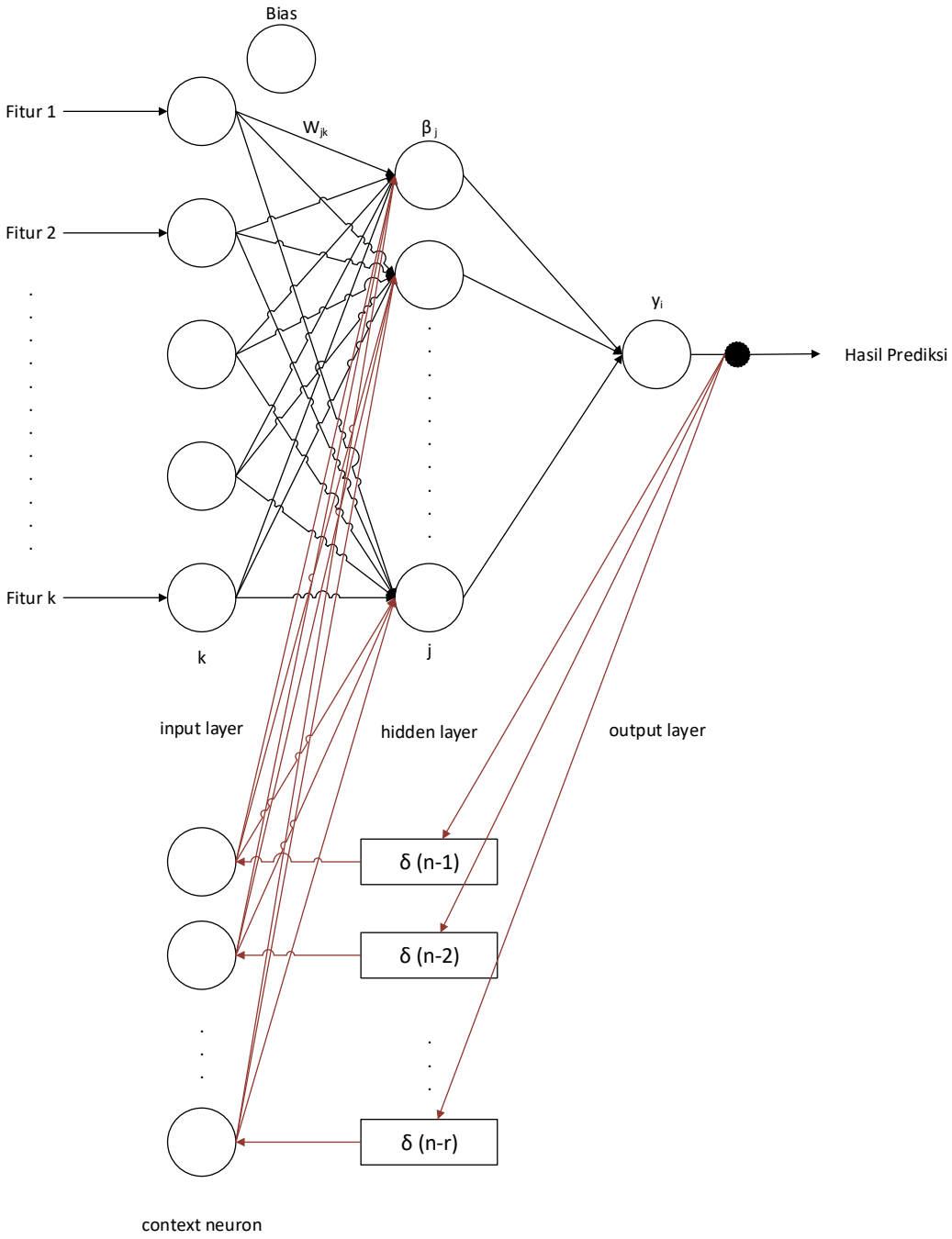
5. Hitung evaluasi *testing.*

2.6 Recurrent Extreme Learning Machine Neural Network

Recurrent Extreme Learning Machine Neural Network merupakan metode yang digunakan pada ELM dengan menambahkan mekanisme *recurrent* didalamnya. Cara kerja *recurrent* yaitu membentuk koneksi antar node secara berurutan sehingga jaringan syaraf dapat terhubung satu sama lain seperti rantai (Ghelani, 2019). Secara singkat proses *recurrent* pada ELM digunakan dengan memberikan koneksi antara *input* dan *output*, sehingga *output* jaringan yang dihasilkan akan bergantung dengan nilai input sebelumnya.

2.6.1 Arsitektur Jaringan RELMNN

Arsitektur jaringan pada RELMNN tidak jauh berbeda dengan arsitektur jaringan pada ELM. Perbedaan yang terletak pada proses *recurrent* terlihat pada arsitektur dengan adanya *context neuron*. Fungsi dari *context neuron* ialah menyimpan nilai dari *delayed output*, sehingga *neuron* ini seperti *nput* tambahan yang akan dioperasikan pada ELM. Arsitektur RELMNN dapat dilihat pada Gambar 2.2.



Gambar 2.2 Arsitektur Jaringan RELMNN

Sumber: (Ertugrul, 2016)

2.6.2 Proses *Training*

Tahapan *training* yang dilakukan pada RELMNN tidak jauh berbeda dengan tahapan ELM. Proses *training* RELMNN adalah:

1. Inisialisasi matriks *delay* δ dengan Persamaan 2.12.

$$\boldsymbol{\delta_{ir} = T\ (i - (k + r) + k} \quad (2.12)$$

Keterangan:

δ_{jk} : *delay* pada urutan data ke-j dan kolom ke-r

T : matriks label *training*

k : jumlah *input neuron*

i : urutan data

r : urutan *context neuron*

2. Membuat nilai *random* pada matriks $W'_{j(k+r)}$ sebagai bobot *input* dengan *range* nilai [-1,1]. Kemudian membuat nilai *random* untuk matriks bias b dengan range [0,1] dalam array berukuran 1 x jumlah *hidden neuron*.
3. Menghitung nilai matriks H_{train} yang ditunjukkan pada Persamaan 2.13.

$$\boldsymbol{H_{train} = g\ (\sum_{k=1}^{n}[X_{train_{ik}}, \delta]\ .\ W'^{T}_{j(n+r)} + b_j)} \quad (2.13)$$

Keterangan:

$\mathrm{H}_{\mathrm{train}}$: Matriks *output hidden layer* pada *training*

X_{train} : Data *training*

$[X_{train_{ik}}, \delta]$: Matriks gabungan X_{train} dan δ

δ : Matriks delay

g : Fungsi aktivasi

W'^{T} : Transpose matriks *input weights*

b : Bias

i : Urutan data

j : Jumlah *hidden neuron*

k : Jumlah *input neuron*

r : Urutan *context neuron*

4. Hitung matriks $H_{train}{}^{+}$ *Moore-Penrose Generalized Inverse* menggunakan Persamaan 2.5.
5. Hitung matriks *Beta* ($\hat{\beta}$) menggunakan Persamaan 2.6.
6. Hitung matriks $\hat{Y}_{train}$ menggunakan Persamaan 2.7.
7. Menentukan label hasil *training* menggunakan Persamaan 2.8.
8. Hitung evaluasi *training*

2.6.3 Proses *Testing*

Proses *testing* dilakukan untuk mengetahui kinerja dari metode yang telah diimplementasikan pada proses *training*. Tahapan-tahapan yang dilakukan pada proses *testing* adalah:

1. Diketahui nilai bobot$W'_{j(k+r)}$, bias b, dan beta $\hat{\beta}$ yang didapatkan dari hasil *training* untuk diinputkan dalam proses *testing.*
2. Inisialisasi matriks delay δ menggunakan Persamaan 2.12
3. Hitung matriks H_{test} yang ditunjukkan pada Persamaan 2.14.

$$\boldsymbol{H_{test} = g\ (\sum_{k=1}^{n}[X_{test_{ik}}, \delta]\ .\ W'_{j(n+r)}{}^{T} + b_j)} \quad (2.14)$$

Keterangan:

$\mathrm{H}_{\mathrm{test}}$	: Matriks *output hidden layer* pada *training*
X_{test}	: Data *training*
$[X_{test_{ik}}, \delta]$	: Matriks gabungan X_{train} dan δ
δ	: Matriks delay
g	: Fungsi aktivasi
W'^{T}	: Transpose matriks *input weights*
b	: Bias
i	: Urutan data
j	: Jumlah *hidden neuron*
k	: Jumlah *input neuron*
r	: Urutan *context neuron*

4. Hitung matriks $\widehat{\mathrm{Y}}_{\mathrm{test}}$ dengan menggunakan Persamaan 2.10.
5. Menentukan label hasil testing menggunakan Persamaan 2.11.
6. Hitung evaluasi testing.

2.7 Principal Component Analysis

Pemrosesan teks merupakan hal yang sulit, dikarenakan teks yang telah diolah pada tahap *pre-processing text* memiliki kelemahan yaitu teks memiliki dimensi yang tinggi. *Principal Component Analysis* (PCA) merupakan salah satu metode yang dapat mengurangi dimensi data yang tinggi, seperti pada data berbentuk teks. PCA merupakan metode yang dapat memaksimalkan pengurangan dimensi namun tetap meminimalkan kehilangan informasi dari data tersebut (Uğuz, 2011).

Dalam mentransformasikan dataset yang telah ada pada tahapan *pre-processing* menjadi dataset baru yang memiliki dimensi lebih kecil dibandingkan sebelumnya, perlu melalui proses menggunakan PCA. Tahapan-tahapan PCA adalah (Zhu et al., 2019):

1. Hitung nilai *mean global*

$$\overline{\boldsymbol{x}} = \frac{\sum_{i=1}^{n} \boldsymbol{x_i}}{\boldsymbol{n}} \quad (2.15)$$

Keterangan:

$\bar{x}_n$: nilai *mean global* pada fitur ke-*n*

x_i : nilai dari data urutan ke-*i*

i : urutan data

n : jumlah data

2. Hitung nilai *varian*

$$\mathbf{s^2} = \frac{\sum_{i=1}^{n} (\boldsymbol{x_i} - \overline{\boldsymbol{x}})}{\boldsymbol{n-1}} \quad (2.16)$$

Keterangan:

$\mathbf{s^2}$: nilai *varian*

3. Hitung *covarians*

$$\boldsymbol{X^{nxn}} = (\boldsymbol{x_{ij}}, \boldsymbol{x_{ij}} = \boldsymbol{cov}\left(\boldsymbol{Dim_i}, \boldsymbol{Dim_j}\right)) \quad (2.17)$$

Keterangan:

X^{nxn} : matriks data baris ke-*n* dan kolom ke-*n*

Dim_i : dimensi pada baris ke-*i*

Dim_j : dimensi pada kolom ke-*j*

4. Hitung *eigen value* dan *eigen vector*

$$\boldsymbol{Ax} = \boldsymbol{\lambda x} \quad (2.18)$$

Dikarenakan *eigen vector* berkorespondensi dengan λ yang merupakan *eigen value* matriks A yang merupakan *non-zero vector*, maka didapatkan persamaan

$$(\boldsymbol{\lambda I} - \boldsymbol{A})\boldsymbol{x} = \mathbf{0} \quad (2.19)$$

Keterangan:

A : matriks ukuran *n*x*n*

x : *eigen vector* dari A

λ : *eigen value*

2.8 Nilai Evaluasi

Evaluasi merupakan tahapan dalam upaya untuk mengukur keberhasilan suatu sistem dengan membandingkan hasil perolehan implementasi dengan kriteria standar yang telah ditetapkan (Parikh & M.M, 2009). Umumnya untuk mengevaluasi hasil implementasi pada sentimen analisis menggunakan *confusion*

matrix. Pengukuran evaluasi dilakukan berdasarkan *confusion matrix* yang diperlihatkan pada Tabel 2.5.

Tabel 2.5 *Confusion Matrix*

Classification	***Predicted Positives***	***Predicted Negatives***
Actual Positive Cases	*Number of True Positive Cases (TP)*	*Number of False Negative Cases (FN)*
Actual Negatives Cases	*Number of False Positive Cases (FP)*	*Number of True Negative Cases (TN)*

Sumber: (Asch, 2013)

Dari Tabel 2.1 diketahui bahwa *true positive* merupakan jumlah dokumen yang prediksi kelasnya bernilai positif dan kelas aktualnya bernilai positif. *False negative* adalah jumlah dokumen yang diprediksi menjadi kelas negatif oleh sistem, namun kelas aktual dari dokumen adalah positif. *False positive* adalah jumlah dokumen yang diprediksi sebagai kelas positif oleh sistem tetapi kelas aktualnya adalah negatif. Sedangkan *true negative* ialah jumlah dokumen kelas yang diberikan oleh sistem dan kelas aktualnya bernilai sama, yaitu negatif.

Perhitungan yang dilakukan dalam tahap evaluasi berupa *Accuracy, Precision, Recall*, dan *F-Measure* didefinisikan pada persamaan berikut:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TN+TP}}{\boldsymbol{TN+TP+FP+FN}} \quad (2.20)$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (2.21)$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (2.22)$$

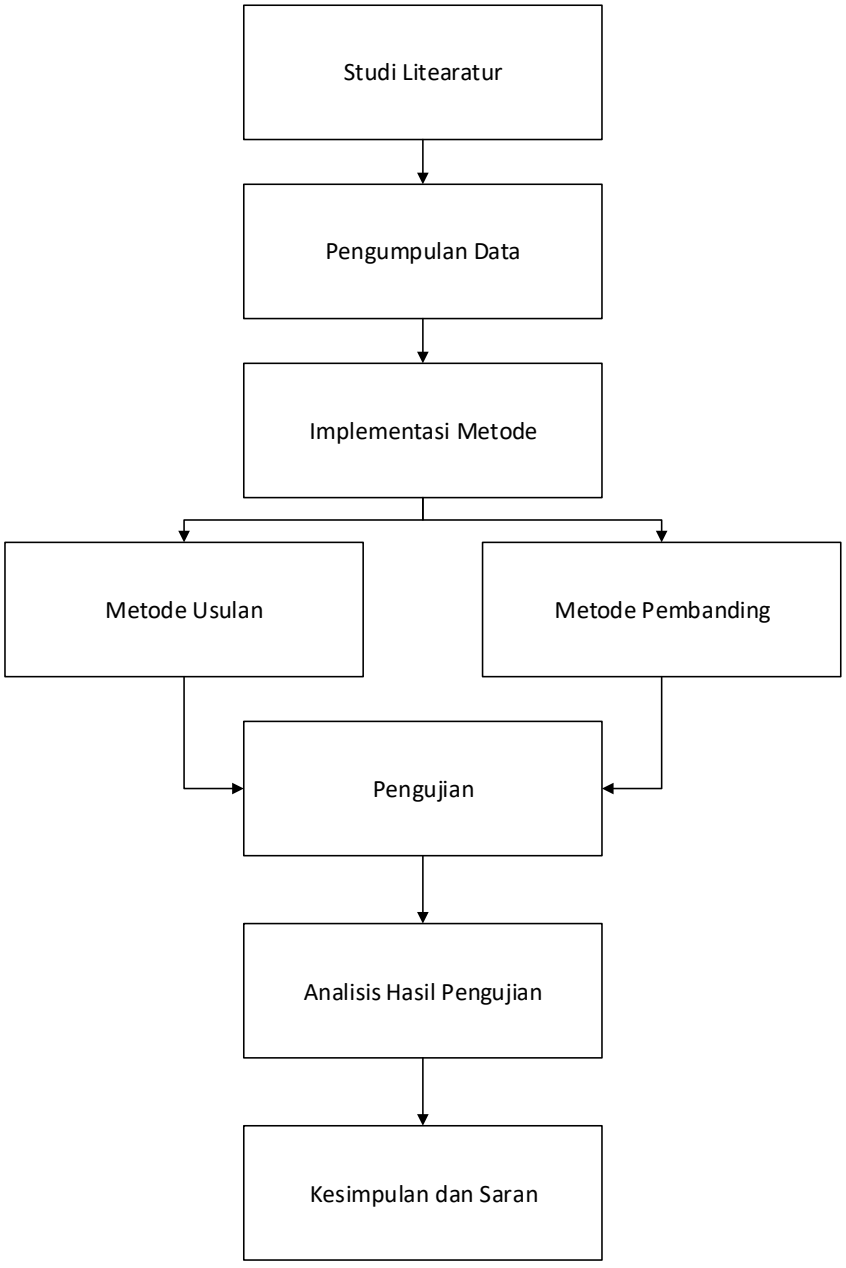
$$\boldsymbol{F-Measure} = \frac{\boldsymbol{2\times Precision\times Recall}}{\boldsymbol{Precision+Recall}} \quad (2.23)$$

Accuracy merupakan evaluasi yang dilakukan dengan menghitung seluruh keadaaan yang diprediksikan dengan nilai yang benar terhadap seluruh keadaan yang diprediksi. Proses evaluasi *precision* merupakan perhitungan pada kondisi benar, yaitu kelas aktual dan kelas prediksi yang sama (positif) terhadap seluruh kondisi yang diprediksi positif. Recall adalah perhitungan pada kondisi benar yaitu, merupakan kelas data positif terhadap seluruh kondisi aktual yang bernilai positif. Sedangkan *F-Measure* adalah perhitungan yang melibatkan *precision* dan *recall* untuk dicari nilai tengah pada kedua evaluasi tersebut.

BAB 3 METODOLOGI

3.1 Metode Penelitian

Pada penelitian dilakukan tahapan-tahapan untuk menyelesaikan persamalahan dengan mengimplementasikan metode usulan yaitu *Recurrent Extreme Learning Machine Neural Network* dan *Principal Component Analysis*. Tujuan penelitian dilakukan untuk mengetahui efektifitas metode usulan pada klasifikasi *phishing email*. Alur dari metode penelitian yang diilustrasikan pada blok diagram seperti pada Gambar 3.1:



Gambar 3.1 Alur Metode Penelitian

3.2 Studi Literatur

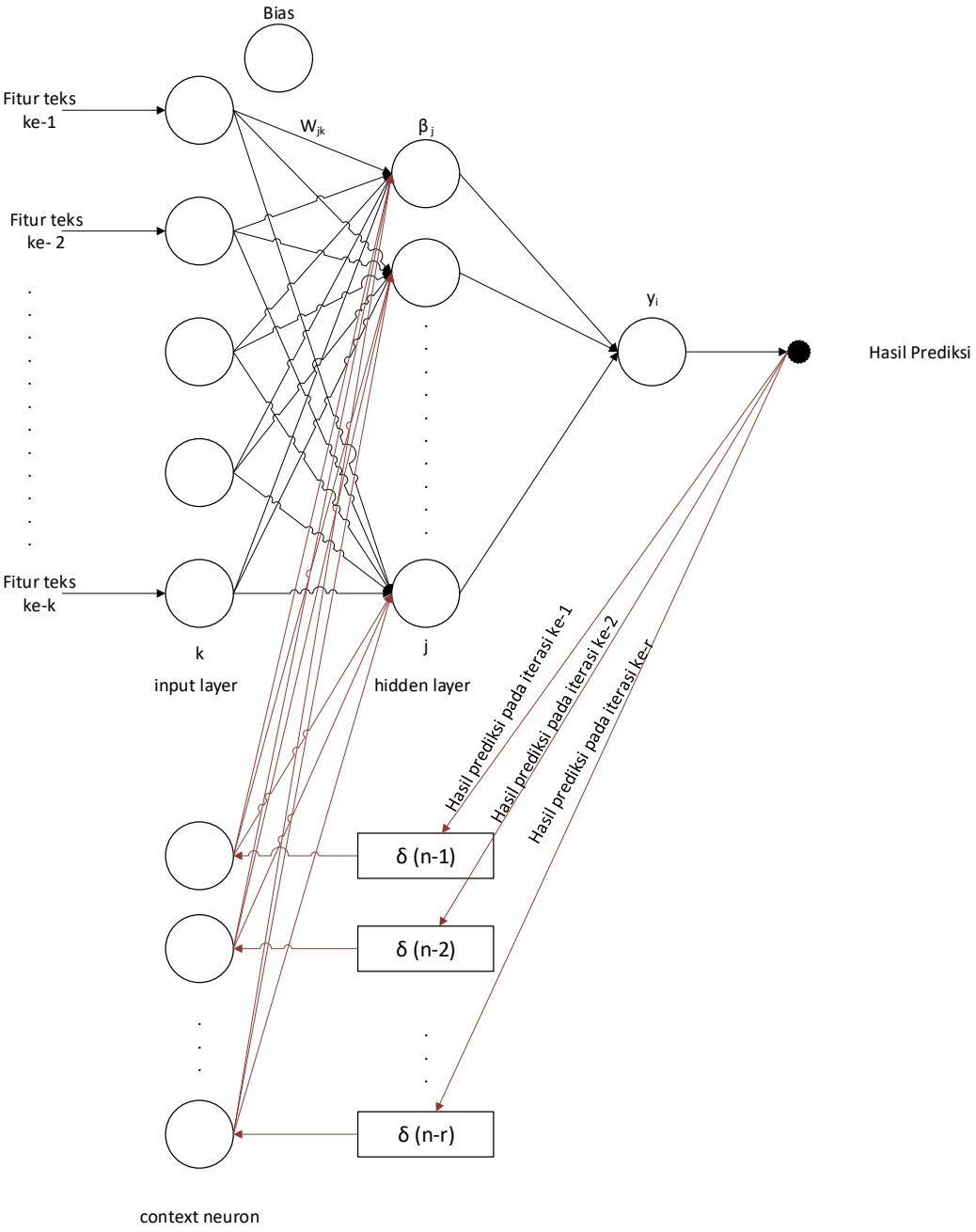
Pada studi literatur membahas mengenai teori pendukung yang digunakan sebagai acuan dalam menyelesaikan masalah. Penelitian ini memerlukan untuk mempelajari bidang ilmu berkaitan yaitu, klasifikasi *phishing email* menggunakan *Recurrent Extreme Learning Machine Neural Network*. Studi Pustaka mencakup *text mining, phishing email, recurrent extreme learning machine neural network, principal component analysis.* Literatur tersebut didapatkan dari buku, jurnal, artikel, dan dokumen *project*.

3.3 Pengumpulan Data

Penelitian dilakukan dengan menggunakan dua sumber dataset umum yang terdapat pada internet. Dataset *phishing email* didapatkan pada *phishcorpus* sedangkan untuk dataset *non-phishing email* menggunakan dataset yang bersumber pada *enron email dataset*.

3.4 Perancangan Sistem

Dalam menyelesaikan kasus *phishing email* diperlukan tahapan-tahapan *Recurrent Extreme Lerning Machien Neural Network* merupakan metode yang diusulkan dengan ELM yang ditambahkan metode *recurrent*. Proses *recurrent* yang berjalan pada ELM yaitu dengan adanya *context neuron* yang berfungsi sebagai penyimpanan nilai *delayed output.* Sebelum proses klasifikasi dilakukan akan dijalankan *pre-processing* pada data teks yang kemudian dilakukan pembobotan kata dengan TF-IDF. Teks yang memiliki dimensi data yang tinggi kemudian akan direduksi menggunakan PCA sebelum masuk pada tahapan klasifikasi.



Gambar 3.2 Arsitektur Rancangan RELMNN

3.5 Skenario Pengujian

Pengujian dilakukan dengan tujuan untuk memberikan informasi akurat terkait efektifitas algoritme yang diproses oleh sistem. Skenario pengujian yang digunakan pada penelitian ini terdiri dari beberapa tahapan, diantaranya adalah pengaruh ekstraksi fitur, pengujian parameter, waktu komputasi, dan akurasi. Penjelasan terkait tahapan pengujian pada penelitian ini adalah:

1. Pengujian Pengaruh Ekstraksi Fitur

Pada tahapan pengujian ini dilakukan pengujian terhadap perbandingan sistem yang mengimplementasikan *Principal Component Analysis* dengan sistem yang tidak menggunakan.

2. Pengujian Parameter

   Pengujian parameter dilakukan dengan menguji jumlah *hidden neuron* dan jumlah *context neuron* yang digunakan oleh sistem untuk mencapai hasil maksimal dalam menyelesaikan permasalahan pada *phishing email.*

3. Pengujian Metode Pembanding

   Pada tahapan pengujian metode pembanding akan dilakukan perbadingan metode RELMNN sebagai metode usulan dengan ELM yang merupakan metode tradisional.

4. Pengujian Akurasi

   Pengujian akurasi akan dilakukan berdasarkan *confussion matrix* sehingga dapat mengetahui performa algoritme yang didapatkan untuk menyelesaikan permasalahan.

3.6 Analisa Hasil Pengujian

Setelah berkahir dilakukanya tahapan pengujian metode usulan yang telah dilakukan untuk mendektesi *phishing email*, kemudian akan dilakukan analisa hasil pengujian terhadap setiap skenario pengujian. Dari hasil analisa tersebut kemudian dapat disimpulkan apakah metode yang diusulkan sudah mencapai performa terbaik dan telah mengungguli metode pembanding.

3.7 Alat Pengolah Data

Pada penelitian ini terdapat alat pengolah data yang digunakan untuk deteksi *phishing email.* Perangkat keras dan perangkat lunak merupakan dua jenis alat yang digunakan. Rincian dari kedua perangkat tersebut ialah:

1. Perangkat keras yang digunakan pada penelitian adalah *Processor, Random Access Memory* (RAM), *Read Only Memory* (ROM). Rinciann perangkat keras ditunjukkan pada Tabel 3.1.

Tabel 3.1 Perangkat Keras

Jenis	Tipe
Processor	Intel(R) Core (TM) i5-4210U CPU @ 1.70GHz (4CPUs), ~2.4GHz
Random Access Memory (RAM)	8GB
Read Only Memory (ROM)	500GB

2. Perangkat lunak yang digunakan pada penelitian adalah Sistem Operasi, bahasa pemrograman, dan *code editor*. Rincian perangkat lunak ditunjukkan pada Tabel 3.2.

Tabel 3.2 Perangkat Lunak

Jenis	Tipe
Sistem Operasi	Windos 10 Pro 64-bit
Bahasa Pemrograman	*Python* 2.6.7
Code Editor	*PyCharm Community Edition* 2017 2.3

DAFTAR PUSTAKA

Akbari, M. I. H. A. D., Astri Novianty S.T., M., & Casi Setianingsih S.T., M. (2012). Analisis Sentimen Menggunakan Metode Learning Vector Quantization. *Telkom University*.

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*. https://doi.org/10.3389/fcomp.2021.563060

Anisatul Umah. (2021). *Kasus Phising Email yang Serang Indonesia Makin Merajalela*. CNBC Indonesia. https://www.cnbcindonesia.com/tech/20210306162132-37-228322/kasus-phising-email-yang-serang-indonesia-makin-merajalela

Asch, V. Van. (2013). *Macro- and micro-averaged evaluation measures*. 1–27.

Bagui, S., Nandi, D., Bagui, S., & White, R. J. (2019). Classifying phishing email using machine learning and deep learning. *2019 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2019*, *MI*, 1–2. https://doi.org/10.1109/CyberSecPODS.2019.8885143

Chawla, M., & Singh Chouhan, S. (2014). A Survey of Phishing Attack Techniques. *International Journal of Computer Applications*, *93*(3), 32–35. https://doi.org/10.5120/16197-5460

Cholissodin, I., Sutrisno, S., Soebroto, A. A., Hanum, L., & Caesar, C. A. (2017). Optimasi Kandungan Gizi Susu Kambing Peranakan Etawa (PE) Menggunakan ELM-PSO Di UPT Pembibitan Ternak Dan Hijauan Makanan Ternak Singosari-Malang. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *4*.

Datacom. (2020). *The 5 W's of phishing*. https://datacom.com/nz/en/discover/articles/blog-5-ws-of-phising

Ertugrul, Ö. F. (2016). Forecasting electricity load by a novel recurrent extreme learning machines approach. *International Journal of Electrical Power and Energy Systems*, *78*, 429–435. https://doi.org/10.1016/j.ijepes.2015.12.006

Ghelani, S. (2019). *Text Classification — RNN's or CNN's?* Towards Data Science. https://towardsdatascience.com/text-classification-rnns-or-cnn-s-98c86a0dd361

Huang, G. Bin, Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1–3), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126

Jiawei, H., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques Third Edition. In *MA: Morgan Kaufmann*.

KnowBe4. (2021). *What Is Phishing*. https://www.phishing.org/what-is-phishing

Kogan, J., & Berry, M. W. (2010). *Text Mining Applications and Theory* (Vol. 10).

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool.

Manning, C., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval*. Cambridge University Press.

Nugroho, G. A. P. (2016). *Analisis Sentimen Data Twitter Menggunakan K-Means Clustering*.

Parikh, R., & M.M. (2009). *Sentiment Analysis of User Generated Twitter Updates using Various Classification Techniques*.

Sanchez, F., & Duan, Z. (2012). A sender-centric approach to detecting phishing emails. *Proceedings of the 2012 ASE International Conference on Cyber Security, CyberSecurity 2012*, *SocialInformatics*, 32–39. https://doi.org/10.1109/CyberSecurity.2012.11

Sindhu, S., Patil, S. P., Sreevalsan, A., Rahman, F., & Saritha, A. N. (2020). Phishing detection using random forest, SVM and neural network with backpropagation. *Proceedings of the International Conference on Smart Technologies in Computing, Electrical and Electronics, ICSTCEE 2020*, 391–394. https://doi.org/10.1109/ICSTCEE49637.2020.9277256

Uğuz, H. (2011). A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems*, *24*(7), 1024–1032. https://doi.org/10.1016/j.knosys.2011.04.014

Walser, A. (2021). *"It was my life's savings": How hackers use email phishing scams to steal billions Cyber-crooks identified from 44 countries*. https://www.abcactionnews.com/news/local-news/i-team-investigates/it-was-my-lifes-savings-how-hackers-use-email-phishing-scams-to-steal-billions

Wang, X., & Xing, S. (2016). The research of ELM ensemble learning on multi-class resampling imbalanced data. *Proceedings of 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2015*, 455–459. https://doi.org/10.1109/IAEAC.2015.7428594

Yang, L., Zhang, J., Wang, X., Li, Z., Li, Z., & He, Y. (2021). An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. *Expert Systems with Applications*, *165*(July 2020), 113863. https://doi.org/10.1016/j.eswa.2020.113863

Zareapoor, M., & K. R, S. (2015). Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection. *International Journal of Information Engineering and Electronic Business*, *7*(2), 60–65. https://doi.org/10.5815/ijieeb.2015.02.08

Zhu, C., Idemudia, C. U., & Feng, W. (2019). Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. *Informatics in Medicine Unlocked*, *17*(April), 100179. https://doi.org/10.1016/j.imu.2019.100179

LAMPIRAN A

LAMPIRAN B