BAB 3
METODOLOGI PENELITIAN

3.1 Tahap Pelaksanaan Penelitian

Dalam penelitian ini, penelitian dilakukan dengan menggunakan metode *feature selection wrapper* yaitu *Recursive Feature Elimination* dan menggunakan algoritma *Support Vector Machine (SVM)* untuk klasifikasi data. Langkah-langkah penelitian dalam analisis dataset dan pembuatan sistem dapat dipaparkan sebagai berikut.

3.1.1 Telaah Literatur

Pada penelitian ini, hal pertama yang dilakukan yaitu melakukan studi litelatur terkait masalah, model, dan algoritma yang digunakan untuk penelitian studi kasus yang digunakan. Studi literatur dilakukan dengan mencari berbagai jurnal yang berhubungan, kemudian juga mencari berbagai jurnal yang berhubungan dengan model dan algoritma yang akan digunakan. Tujuan dari tahap studi litelatur ini adalah untuk memperdalam pemahaman terkait algoritma, model, dan masalah dari penelitian kali ini yaitu, *Support Vector Machine*, *Recursive Feature Elimination*, dan *website phising*.

3.1.2 Pengumpulan Data

Pada penelitian ini, hal pertama yang dilakukan yaitu melakukan pengumpulan data. Dataset yang diambil adalah dataset dari **data mendeley** yang berjudul Web *page phishing detection* (*https://data.mendeley.com/datasets/c2gw7fy2j4/3*). Dataset ini berisi 11.430 data dengan 87 atribut. Data ini sudah diberikan label berupa kategori yaitu *phishing* dan *legitimate*. Fitur-fitur *dataset* yang digunakan pada penelitian ini hampir semua fiturnya memiliki tipe data *integer*. Hanya label saja yang merupakan tipe data *string*. Pada dataset ini juga terdapat 3 format fitur yang dimasukan, yaitu untuk format *URL-based-feature* terdapat 56 fitur, *content-based feature* terdapat 24 fitur, dan *external service feature* terdapat 7 fitur. Dataset ini juga memiliki label yang seimbang, yaitu *phishing* sebesar 50% dan *legitimate* sebesar 50%.

Pada proses pengumpulan data, data juga dibagi menjadi 2 bagian yaitu

data testing dan *data training*. Pembagian dilakukan dengan membagi sebesar 70% *data training* dan 30% *data testing*. Pembagian data sebesar 70% berbanding 30% ini dilakukan karena pada penelitian yang dilakukan oleh Rahmadan Adinugroho menyimpulkan bahwa pembagian data sebesar 70% dan 30% sangat mungkin untuk dilakukan pada data yang berjumlah 10, 1000, dan 10000. Kemudian juga pada penelitian tersebut disebutkan bahwa pembagian data sebesar 70% dan 30% memiliki nilai *error* mutlak paling rendah untuk melakukan prediksi pada suatu data [18].

Tabel 3.1 adalah fitur beserta keterangan dari *dataset* yang digunakan pada penelitian ini.

Tabel 3.1. Dataset yang digunakan

Kategori Fitur	**Fitur**	**Keterangan**
URL-based-Feature Extraction	url	Alamat domain
	length_url	Panjang alamat domain
	length_hostname	panjang hostname
	ip	Alamat numerik
	nb_dots	Jumlah ”titik alamat domain”
	nb_hyphens	Jumlah ”tanda hubung”
	nb_at	Jumlah ”@”
	nb_qm	Jumlah ”tanda tanya”
	nb_and	Jumlah ”dan”
	nb_or	Jumlah ”atau”
	nb_eq	Jumlah ”sama dengan”
	nb_underscore	Jumlah ”garis bawah”
	nb_tilde	Jumlah ”tanda hubung”
	nb_percent	Jumlah ”tanda persen”
	nb_slash	Jumlah ”garis miring”
	nb_star	Jumlah ”bintang”
	nb_colon	Jumlah ”titik dua”
	nb_comma	Jumlah ”koma”
	nb_dollar	Jumlah ”lambang dollar”
	nb_space	Jumlah spasi
	nb_www	Jumlah kata www

Lanjut pada halaman berikutnya

Tabel 3.1 Dataset yang digunakan (lanjutan)

Kategori Fitur	Fitur	Keterangan
	nb_com	Jumlah kata com
	nb_dslash	Jumlah *double slash*
	http_in_path	Jumlah kata http
	https_token	Penggunaan token URL
	ratio_digits_url	rasio antara jumlah digit numerik dan total karakter
	ratio_digits_host	rasio antara jumlah digit numerik dan total karakter (host)
	punycode	Jumlah karakter non ASCII
	port	Port yang digunakan
	tld_in_path	Top Domain Level (domain)
	tld_in_subdomain	Top Domain Level (subdomain)
	abnormal_subdomain	Subdomain yang tidak biasa
	nb_subdomains	Jumah subdomain
	prefix_suffix	Sifat awal prefix suffix
	random_domain	Domain acak
	shortening_service	Layanan Penyederhanaan URL
	path_extension	Ekstensi Path
	nb_redirection	Jumlah link internal URL
	nb_external_redirection	Jumlah link eksternal URL
	length_words_raw	Total kata URL
	char_repeat	Karakter yang berulang URL
	shortest_words_raw	kata-kata terpendek URL
	shortest_word_host	kata-kata terpendek URL (host)
	shortest_word_path	Kata terpedek (path)
	longest_words_raw	kata-kata terpanjang URL
	longest_word_host	kata terpanjang URL (host)
	longest_word_path	kata terpanjang (path)
	avg_words_raw	rata-rata jumlah kata URL
	avg_word_host	rata-rata jumlah kata URL (host)
	avg_word_path	rata-rata jumlah kata (path)
	phish_hints	Pendeteksi phishing
	domain_in_brand	domain brand

Lanjut pada halaman berikutnya

Tabel 3.1 Dataset yang digunakan (lanjutan)

Kategori Fitur	**Fitur**	**Keterangan**
	brand_in_subdomain	subdomain brand 3
	brand_in_path	path brand
	suspecious_tld	Top Level Domain yang mencurigakan
	statistical_report	Jumlah laporan URL
Content-based-feature	nb hyperlinks	Baris 2, Kolom 3
	ratio_intHyperlinks	Rasio jumlah hyperlink internal
	ratio_extHyperlinks	Rasio jumlah hyperlink eksternal
	ratio_nullHyperlinks	Rasio jumlah hyperlink null
	nb_extCSS	Jumlah eksternal CSS
	ratio_intRedirection	Rasio pengalihan halaman internal
	ratio_extRedirection	Rasio pengalihan halaman eksternal
	ratio_intErrors	rasio internal error
	ratio_extErrors	rasio eksternal error
	login_form	login form
	external_favicon	Jumlah eksternal icon
	links_in_tags	Tag link
	submit_email	Jumlah email terkumpul
	ratio_intMedia	Rasio internal media
	ratio_extMedia	Rasio eksternal media
	sfh	Jumlah Server Form Handle
	iframe	Jumlah inline frame
	popup_window	Jumlah pop up
	safe_anchor	Tautan / anchor dalam HTML
	onmouseover	Jumlah mouseover pada HTML
	right_clic	Jumlah klik kanan
	empty_title	Jumlah title kosong
	domain_in_title	Jumlah domain dalam title
	domain_with_copyright	Jumlah copyright
	whois_registered_domain	Jumlah registrasi dengan protokol WHOIS
External Service	domain_registration	Baris 2, Kolom 3
		Lanjut pada halaman berikutnya

Tabel 3.1 Dataset yang digunakan (lanjutan)

Kategori Fitur	Fitur	Keterangan
	domain_age	Umur domain
	sfh	Jumlah Server Form Handler
	web_traffic	Lalu lintas web
	dns_record	Record Domain Name Server
	google_index	Halaman yang sudah diindex oleh mesin pencarian google
	page_rank	Ranking halaman
Label	status	hasil dari pengecekan website phishing

3.1.3 Implementasi Algoritma

Pada tahap ini, dilakukan proses implementasi algoritma dari telaah literatur dan dataset yang sudah didapatkan sebelumnya. Proses implementasi algoritma dilakukan sesuai dengan langkah-langkah penelitian yang sudah ditentukan sebelumnya. Pada awalnya melakukan import data yang telah dikumpulkan sebelumnya. Kemudian melakukan *preprocessing data* untuk *cleaning data*, standarisasi data, dan *split data*. Setelah *split data*, kemudian implementasi algoritma pertama dilakukan yaitu *Recursive Feature elimination* untuk menentukan fitur-fitur terbaik yang akan digunakan pada algoritma selanjutnya yaitu *Support Vector Machine* untuk menentukan kelas masing-masing dataset. Implementasi algoritma dilakukan dengan menggunakan *software jupyter notebook*. Bahasa pemrograman yang digunakan yaitu *python*.

3.1.4 Uji coba dan Evaluasi

Setelah implementasi algoritma selesai dilakukan, dan *training data* telah dilakukan, selanjutnya adalah melakukan uji coba dan evaluasi terhadap algoritma yang digunakan. Uji coba dilakukan dengan menggunakan bagian dari data yang telah dibagi, yaitu data testing. Data testing akan menggunakan algoritma SVM dengan RFE dan SVM tanpa RFE. Setelah uji coba dilakukan, kemudian melakukan evaluasi dengan menggunakan *confussion matrix*. *Output* yang dikeluarkan dari

proses evaluasi yaitu *precision*, *recall*, F1 *score*, dan akurasi.

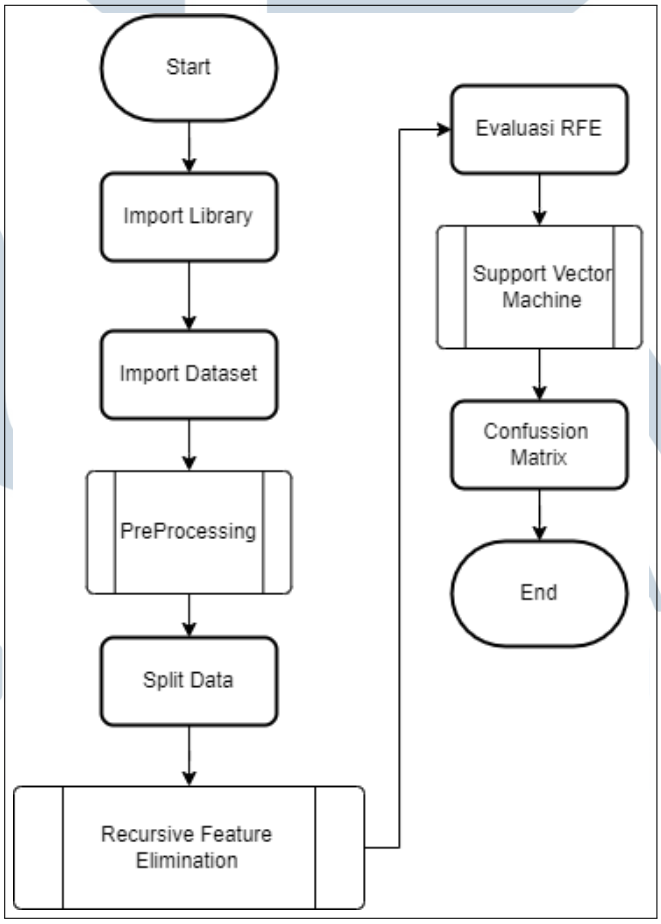
3.1.5 Konsultasi dan Penulisan Laporan

Setelah uji coba dan evaluasi dilakukan, kemudian dilakukan konsultasi untuk mendapatkan masukan dari pembimbing. Lalu melakukan evaluasi sistem untuk perbaikan sistem yang lebih baik. Setelah itu penulisan laporan dilakukan dengan tujuan mendokumentasikan segala bentuk proses penelitian serta menyimpulkan hasil akhir penelitian yang telah dilakukan.

3.2 Perancangan Sistem

Perancangan sistem dari penelitian ini memiliki *flowchart* sebagai berikut.

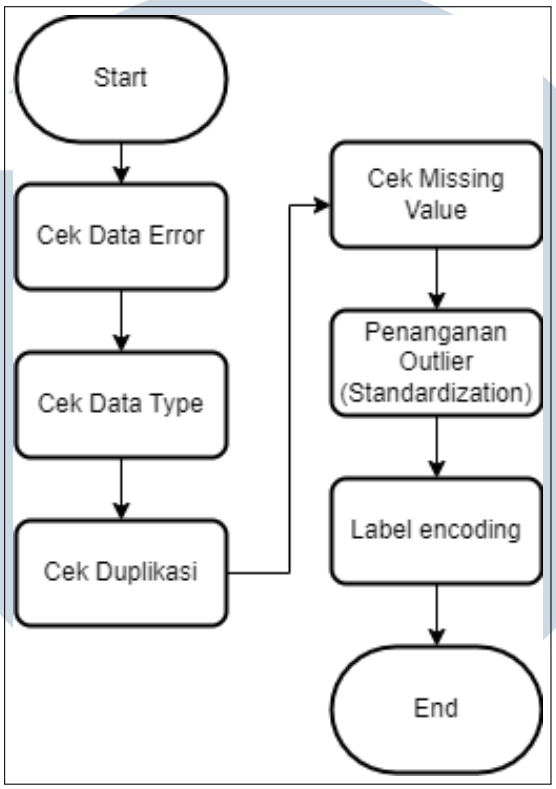
3.2.1 Flowchart Utama



Gambar 3.1. *Flowchart Utama*

Gambar 3.1 adalah alur utama dari sistem dan penelitian ini. Dalam proses perancangan sistem untuk implementasi algoritma RFE dengan SVM, proses awal yang dilakukan yaitu melakukan *import library*. *Library* yang digunakan sesuai dengan kebutuhan sistem untuk penelitian. Kemudian melakukan *import* dataset yang telah dikumpulkan sebelumnya. Dataset yang telah *diimport*, kemudian diolah menjadi data yang lebih bersih pada *preprocessing*. Setelah dataset bersih dan siap digunakan, pembagian data dilakukan. Pembagian dibagi menjadi 2 yaitu, data *training* dan data *testing*. Setelah pembagian data selesai dilakukan selanjutnya penelitian masuk ke implementasi *Recursive Feature Elimination*. Implementasi menggunakan dua model untuk nantinya dibandingkan. Proses RFE juga menggunakan *cross validation* yang digunakan untuk mendapatkan hasil akurasi yang lebih baik dengan lebih banyak pelatihan. Setelah proses RFE, kemudian melakukan evaluasi untuk melihat fitur-fitur yang dihapus, bobot tiap fitur, dan akurasinya. Selanjutnya melakukan pemodelan klasifikasi dengan menggunakan SVM. Setelah pemodelan dilakukan dan pembagian kelas sudah dilakukan, proses evaluasi model secara keseluruhan dengan menggunakan *confussion matrix* untuk melihat apakah proses klasifikasi dengan SVM menggunakan *feature selection Recursive Feature Elimination*. Proses *confussion matrix* menilai *precision, recall, F1 score*, dan akurasi dengan menggunakan rumus 2.7 untuk *precision*, rumus 2.8 untuk *recall*, rumus 2.9 untuk *F1 Score* dan rumus 2.10 untuk akurasi.

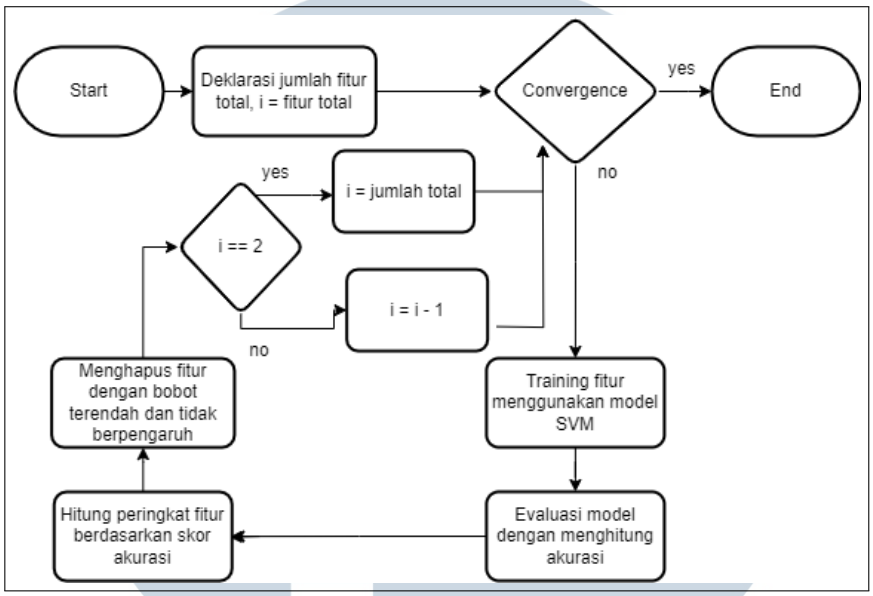
3.2.2 Flowchart Preprocessing



Gambar 3.2. *Flowchart PreProcessing*

Proses *preprocessing* terdapat pada Gambar 3.2 dan dibagi menjadi tiga bagian, yaitu *data cleaning*, *standardization*, dan *label encoding*. Proses *data cleaning* memiliki beberapa tahapan, yaitu melakukan pengecekan apakah ada data *error* dan tidak sesuai dengan tipe data fitur, kemudian melakukan pengecekan apakah ada duplikasi, serta melakukan pengecekan apakah ada data yang kosong. Jika terdapat data *error*, tipe data yang salah, data yang duplikat, dan data yang kosong, maka akan dilakukan penghapusan data-data tersebut sehingga data menjadi bersih dan bisa digunakan dengan lebih maksimal. Setelah melakukan *data cleaning*, selanjutnya, melakukan proses pengecekan *outlier*. Pada penelitian ini, terdapat *outlier* pada beberapa dataset sehingga distribusi data tidak stabil. Proses *standardization* dilakukan untuk menangani data outlier sehingga masih dalam distribusi skala yang baik. label pada dataset yang *diimport* memiliki tipe data *string*. Sedangkan model SVM memerlukan label numerik untuk dapat dilatih modelnya, sehingga perlu dilakukan proses label encoding untuk mengubah data label kategori menjadi data label numerik.

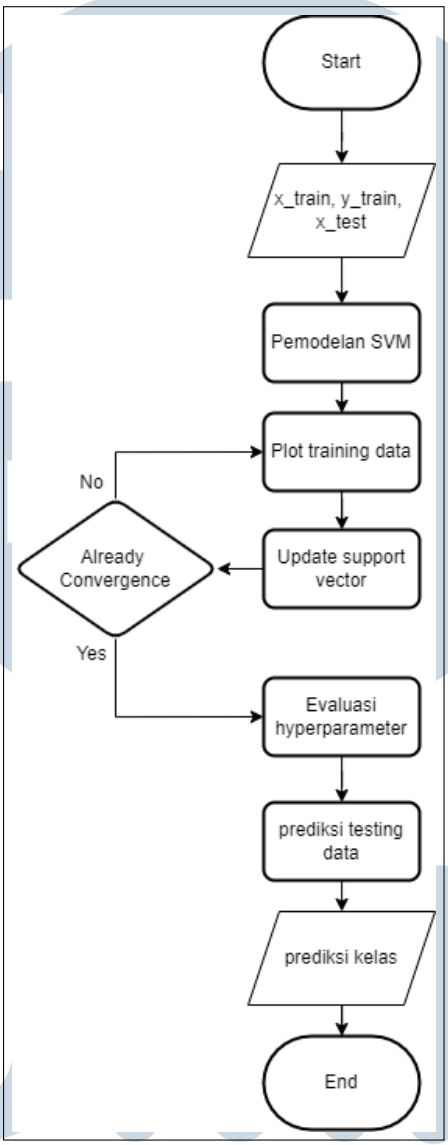
3.2.3 Flowchart Recursive Feature Elimination



Gambar 3.3. *Flowchart Recursive Feature Elimination*

Proses implementasi *Recursive Feature Elimination* diawali dengan tahapan-tahapan seperti Gambar 3.3. Proses dimulai dengan melakukan deklarasi fitur dan *index*. Kemudian melakukan iterasi terhadap proses *training* tiap fitur menggunakan model SVM dan RFE. Kemudian menggunakan validasi silang 10 untuk membuat akurasi lebih akurat dan lebih kompleks dalam pelatihannya. Lalu menghitung peringkat fitur berdasarkan skor akurasi. Setelah fitur selesai di urutkan, kemudian menghapus fitur yang memiliki bobot paling rendah, lalu iterasi terus dilakukan sampai dengan fitur mencapai *convergence*. *Convergence* merupakan kondisi pada saaat Fitur-fitur yang terpilih telah menemukan optimasi yang paling optimal dan kriteria-kriterianya telah terpenuhi. Kriteria yang dimaksudkan adalah jumlah fitur yang telah ditentukan.

3.2.4 Flowchart Support Vector Machine



Gambar 3.4. *Flowchart Support Vector Machine*

Proses implementasi *Support Vector Machine* pada Gambar 3.4 diawali dengan memasukan *x_train, y_train* dan *x_test*. Kemudian melakukan pemodelan SVM dengan menggunakan parameter yang sebelumnya didapatkan melalui proses *gridsearchCV*. Setelah itu melakukan proses *training* dengan model yang SVM dengan mamasukan *x_train* dan *y_train*. Proses training data dilakukan dengan melakukan pengecekan terhadap konvergensi. Konvergensi dalam SVM merujuk pada pelaihan SVM untuk mencapai solusi yang paling optimal atau mendeketi solusi yang optimal. Jika konvergensi belum mencapai titik optimal maka, training

akan dilakukan lagi dengan titik yang berbeda dan akan diupdate titiknya vektornya. Namun, jika sudah konvergensi, maka akan dilakukan pengecekan akurasi dari *hyperparameter*. Setelah data berhasil di-training selanjutnya melakukan prediksi terhadap data *x_test* dengan menggunakan prediksi training yang sebelumnya sudah dilakukan. Hasil prediksi nantinya akan digunakan untuk evaluasi sistem dan menentukan kinerja algoritma RFE dan SVM.