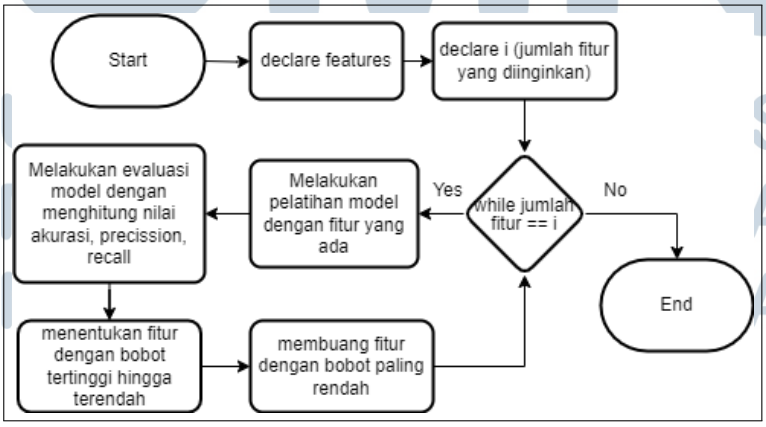
BAB 2
LANDASAN TEORI

2.1 Website Phishing

Website phishing merupakan replika dari suatu situs *website* yang resmi. Seluruh *website phishing* tidak dibangun untuk *phishing*, tetapi hanya beberapa halaman yang memang dikhususkan oleh pelaku untuk memberikan *input* atau *download*, sehingga ketika data terkirim, data tersebut dikirim ke penyerang. Terdapat beberapa metode dalam melakukan kloning *website*, yaitu bisa dengan menggunakan *software* khusus atau bisa dengan membuat secara manual [5]. Dalam *website phishing* terdapat beberapa hal yang dapat dibedakan dari situs aslinya, seperti URL, teks halaman, ikon, *hyperlink*, *domain hosting*, usia *domain*, kode sumber, sertifikat SSL, dan lainnya.

Pada zaman sekarang, *Uniform Resource Locator* (URL) yang dimiliki oleh situs web yang phising biasanya disingkat. URL *website phishing* biasanya tidak memiliki domain, subdomain, dan *Top Level Domain* (TLD). URL ini biasanya tidak menuju ke URL awal, karena tidak ada detail yang dapat ditemukan di situs web ini. Pengguna internet juga tidak begitu memperhatikan URL, sehingga penyerangan jenis URL ini sering sekali digunakan oleh pelaku. Pengguna nantinya akan memasukan *password* atau data pribadi yang dapat menyebabkan akun pengguna terancam [12].

2.2 Recursive Feature Elimination (RFE)



Gambar 2.1. *Proses Recursive Feature Elimination*

[13]

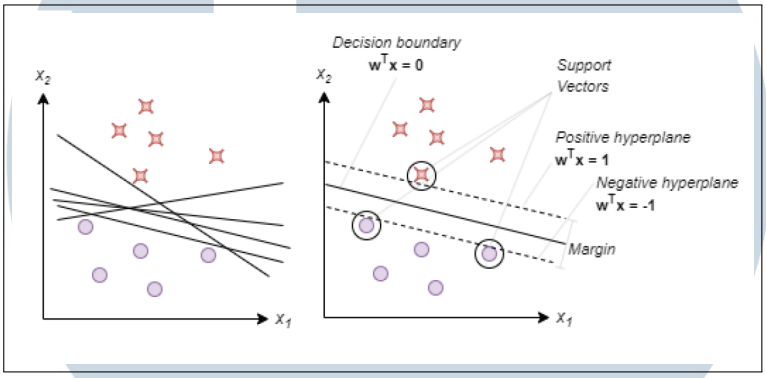
RFE adalah metode *wrapper* dalam *feature selection*. Metode ini adalah metode yang bekerja dengan menghapus fitur yang redundan dan lemah yang penghapusannya paling sedikit mempengaruhi kesalahan *training* dan menjaga fitur yang independen dan kuat untuk meningkatkan kinerja generalisasi model. metdoe RFE ini menggunakan prosedur iterasi untuk melakukan peringkat fitur yang merupakan contoh dari metode ini. Teknik ini memiliki langkah-langkah yang dilakukan seperti pada Gambar 2.1. Pada mulanya membangun model pada seluruh set fitur dan melakukan penilaian ranking fitur berdasarkan pentingnya fitur tersebut. Kemudian setelah memberi peringkat, peringkat yang paling rendah akan di hapus dan akan dibangun kembali model dan memberikan ranking ulang untuk fitur yang paling penting [13]. Berikut adalah tahapan dalam melakukan proses RFE, yaitu :

1. Melakukan pelatihan model menggunakan semua fitur dengan *10-fold cross-validation*
2. Hitung kinerja mode untuk menentukan nilai (akurasi, presisi, dan recall) Menentukan *confusion matrix* yang akan digunakan
3. Membandingkan fitur dengan bobot tertinggi sampai terendah
4. Membuang fitur dengan bobot paling rendah
5. Melakukan iterasi untuk menghitung kinerja model sampai final :
   (a) Melakukan pelatihan dan tes model pada fitur yang terbaru
   (b) Melakukan perhitungan ulang kinerja model
   (c) Melakukan perbandingan fitur dengan bobot tertinggi hingga terendah
   (d) Membuang fitur dengan bobot terendah
6. Gunakan model optimal yang dipilih

2.3 Support Vector Machine (SVM)

Support Vectore Machine (SVM) merupakan salah satu algoritma kuat pada *machine learning*. Pada SVM, setiap *item* data dipetakan sebagai titik dalam ruang n-dimensi dan algoritma ini membangun garis pemisah untuk klasifikasi dua kelas yang dikenal dengan *hyperplane* [6]. *Hyperplane* adalah garis pemisah antar kelas dengan memaksimalkan *margin* di antara kelas-kelas tersebut. *Margin* merupakan

jarak antara *hyperplane* dengan titik dalam ruang n-dimensi terdekat pada masing-masing kelas. Algoritma SVM termasuk dalam metode *ensemble learning*. Hal ini dikarenakan sistem pembelajaran model ini menggunakan ruang hipotesis berupa fungsi-fungsi dari sebuah ruang fitur dimensi tinggi. Dalam ruang ini, terdapat banyak batas yang dapat digunakan untuk memisahkan kelas-kelas tersebut, namun hanya terdapat satu batas yang memaksimalkan *margin* [14].



Gambar 2.2. *Ilustrasi Support Vector Machine* [14]

Pada Gambar 2.2 terdapat ilustrasi algoritma SVM pada bagian kiri yang terdapat banyak *hyperplane* yang memisahkan kelas lingkaran dengan bintang. Namun *hyperplane* yang dapat memaksimalkan margin merupakan klasifikasi terbaik yang terdapat pada Gambar 2.2 bagian kanan.

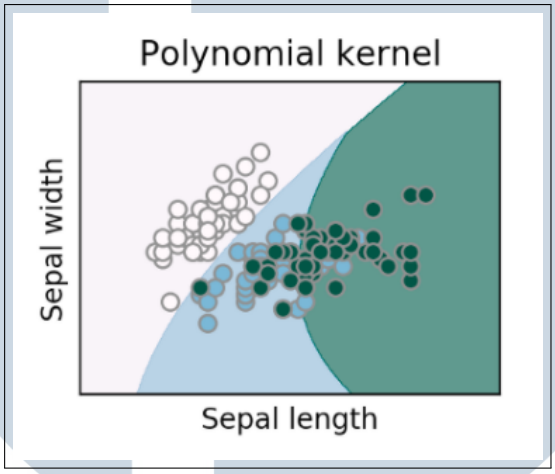
Proses pembelajaran pada algoritma SVM dalam menentukan titik-titik vektor pendukung hanya bergantung pada *dot* yang telah diubah menjadi ruang dimensi yang lebih tinggi. Pada umumnya transformasi F ini tidak diketahui dan sulit untuk dipahami dengan mudah, perhitungan hasil perkalian titik menurut teori *mercer* dapat digantikan dengan fungsi kernel yang secara implisit menentukan transformasi F yang dikenal sebagai trik *Kernel*, yang dirumuskan sebagai berikut [15]

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \tag{2.1}$$

- ϕ = fungsi keputusan dalam memberikan skor prediksi pada variable di dalamnya
- $\mathbf{x}_i$ = vektor fitur pertama
- $\mathbf{x}_j$ = vektor fitur kedua

Terdapat beberapa tipe kernel yang dapat digunakan pada algoritma SVM yaitu sebagai berikut :

1. **Polynomial**, adalah kernel yang berfungsi ketika data tidak terpisah secara linear dan cocok untuk permasalahan pada *training dataset* dinormalisasi. Contoh ruang fitur terdapat pada Gambar 2.3.

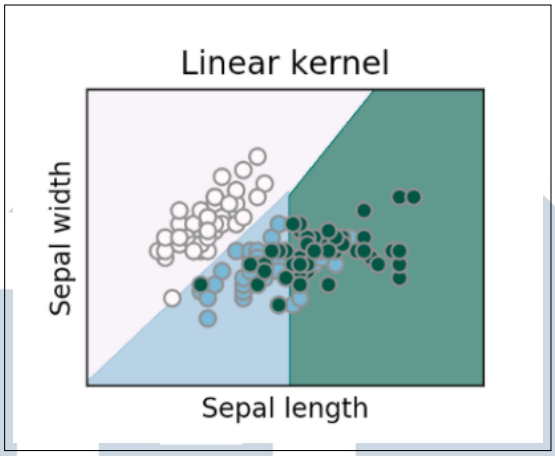


Gambar 2.3. *Ilustrasi Polynomial Kernel*
[14]

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i, \mathbf{x}_j + 1)^p \tag{2.2}$$

- ^{p} = parameter penentu derajat polynomial

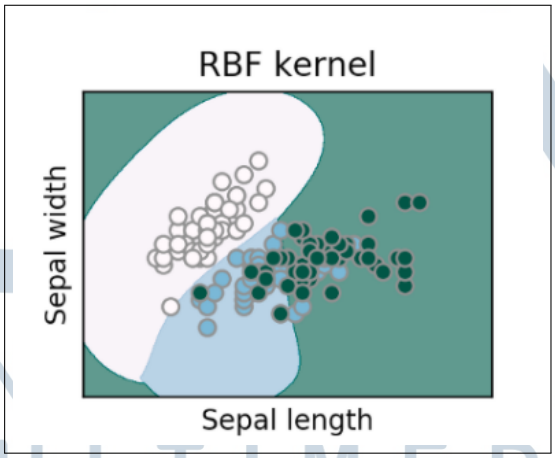
2. **Linear**, adalah kernel yang berfungsi ketika data yang dianalisis sudah terpisah secara linear. Kernel ini cocok ketika terdapat banyak fitur yang tidak dapat meningkatkan kinerja pada klasifikasi teks. Contoh ruang fitur terdapat pada Gambar 2.4.



Gambar 2.4. *Ilustrasi Linear Kernel*
[14]

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.3}$$

3. **Gaussian Radial Basis Function**, adalah kernel yang berfungsi dalam analisis ketika data tidak terpisah secara linear. Dalam melakukan analisis dengan fungsi RBF kernel, dilakukan optimasi parameter *Cost* (C) dan *Gamma* (y). Contoh ruang fitur terdapat pada gambar 2.5.



Gambar 2.5. *Ilustrasi RBF Kernel*
[14]

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{2.4}$$

- exp = fungsi eksponensial, menghasilkan nilai eksponensial dari argumennya
- $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ = jarak Euclidean kuadrat antara vektor fitur x1 dan x2
- σ = parameter *bandwidth* dari *kernel Gauss*, yang mengontrol seberapa jauh pengaruh titik data tersebar dalam ruang fitur.

Selanjutnya hasil klasifikasi dari data diperoleh dari persamaan 2.5 berikut

$$f(\phi(\mathbf{x})) = \sum a_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{2.5}$$

- $f(\phi(\mathbf{x}))$ = nilai yang diprediksi oleh model SVM untuk vektor x
- a_i = koefisien dari vektor dukungan yang dihitung
- y_i = label kelas dari sampel pelatihan xi
- b = pergeseran (bias) yang dihitung selama proses pelatihan SVM.

2.4 Standardization

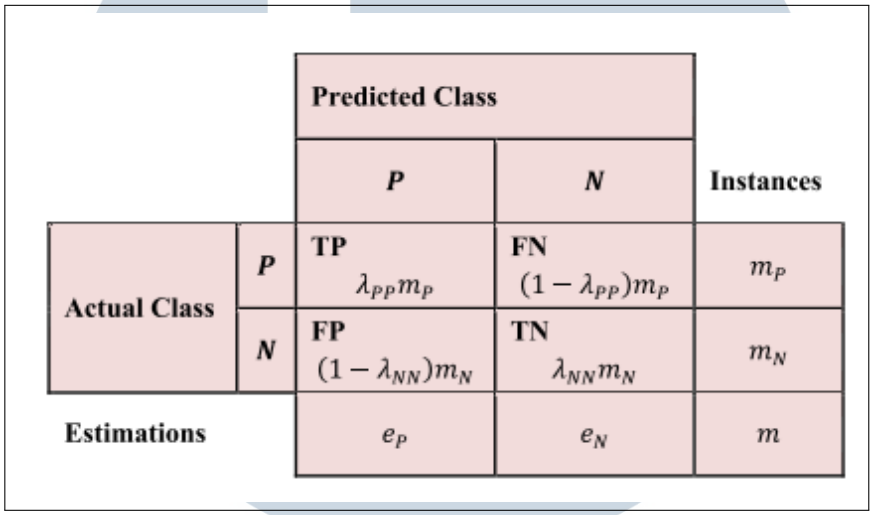
Standarisasi data merupakan langkah *preprocessing* sentral dalam *data mining*. Nilai-nilai fitur atau atribut dari berbagai *range* yang berbeda dan dinamis dibuat menjadi *range* tertentu, yaitu nilai rata-rata yang diharapkan yaitu nol dan deviasi standar satu [14]. Penggunaan metode standarisasi digunakan karena terdapat data *outlier* pada dataset yang digunakan. *Outlier* merupakan nilai yang secara signifikan berbeda dari nilai-nilai lain dalam dataset. Standarisasi menggunakan metode standarisasi *Z-score*. Standarisasi data yang digunakan memiliki formula pada rumus 2.6.

$$Z = \frac{x - \mu}{\sigma} \tag{2.6}$$

- X = nilai hasil
- μ = rata-rata seluruh data
- σ = standar deviasi dari seluruh data

2.5 Confusion Matrix

Confusion matrix merupakan salah satu metode yang paling umum untuk menyajikan hasil yang diperoleh pada saat melakukan klasifikasi. *Confusion matrix* terdapat tabel yang menjadi dasar pada perhitungan evaluasi [16]. Gambar 2.6 adalah tabel *confussion matrix*.



		Predicted Class		
		P	***N***	**Instances**
Actual Class	***P***	**TP** $\lambda_{PP}m_P$	**FN** $(1-\lambda_{PP})m_P$	m_P
	N	**FP** $(1-\lambda_{NN})m_N$	**TN** $\lambda_{NN}m_N$	m_N
Estimations		e_P	e_N	m

Gambar 2.6. *Classification Rate Table*

[16]

Confusion matrix memiliki empat istilah :

1. **True Negative (TN)** : Model memprediksi data ada di kelas **Negatif** dan yang sebenarnya data memang ada di kelas **Negatif**
2. **True Positive (TP)** : Model memprediksi data ada di kelas **Positif** dan yang sebenarnya data memang ada di kelas **Positif**
3. **False Negative (FN)** : Model memprediksi data ada di kelas **Negatif** dan yang sebenarnya data memang ada di kelas **Positif**
4. **False Positive (Fp)** : Model memprediksi data ada di kelas **Positif** dan yang sebenarnya data memang ada di kelas **Negatif**

Confusion matrix adalah salah satu evaluasi yang dilakukan dengan tujuan untuk mengetahui nilai performa dari sebuah sistem yang melakukan perhitungan *precision*, *recall*, F1 *score*, dan akurasi.

Precision digunakan untuk mengukur ketepatan hasil *classifier*, yang dapat dihitung dengan TP dibagi dengan total dari TP dan FP. Persamaan *precision*

terdapat pada rumus 2.7. [17].

$$Precision = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Positive (FP)}} \tag{2.7}$$

recall digunakan untuk mengukur kelengkapan hasil *classifier* yang diukur dengan perhitungan TP dibagi dengan jumlah TP dan FN. Persamaan *recall* terdapat pada rumus 2.8. [17].

$$Recall = \frac{\text{TP}}{\text{TP + FN}} \tag{2.8}$$

F1 *score* digunakan untuk menghitung rata-rata dari *precision* dan *recall*. Persamaannya terdapat pada rumus 2.9. [17].

$$F1Score = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{2.9}$$

Akurasi digunakan untuk menentukan kedekatan nilai sesungguhnya dengan nilai hasil prediksi atau penelitian. Persamaannya terdapat pada rumus 2.10.

$$Akurasi = \frac{(TP + TN)}{TP + FP + FN + TN} \tag{2.10}$$