BAB 3
METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Penelitian ini berfokus pada pengembangan model pembelajaran mesin untuk deteksi *phishing* berbasis URL dengan memanfaatkan karakteriktik URL sebagai dasar analisis, yang diuji melalui proses pelatihan dan pengujian data. Algoritma yang digunakan adalah *Random Forest* (RF) sebagai metode *ensemble learning* dan merupakan bagian dari *Decision Tree* sebagai fondasi *base classifier*. Implementasi metode *ensemble learning* menggunakan *Random Forest* dipilih karena *Decision Tree* tunggal sebagai fondasinya cenderung rentan terhadap *overfitting* dan memiliki keterbatasan dalam menangani dataset dengan jumlah fitur yang besar dan kompleks. Melalui pendekatan *ensemble*, *Random Forest* mampu mengombinasikan hasil dari beberapa pohon keputusan sehingga menghasilkan model yang lebih stabil, memiliki generalisasi yang lebih baik, serta meningkatkan akurasi klasifikasi.

Penelitian ini bertujuan untuk melakukan identifikasi dan klasifikasi situs website sebagai *phishing* dan *legitimate* menggunakan dataset kaggle berjudul "*Web Page Phishing Detection Dataset*" sebagai objek utama pembelajaran mesin penelitian. Proses analisis difokuskan pada pengenalan struktur dan pola karakteristik URL yang berpotensi mengindikasikan serangan *phishing*. Karakteristik yang digunakan dalam penelitian ini meliputi panjang URL dan *hostname*, jumlah dan jenis karakter khusus seperti tanda titik, tanda hubung, simbol "@", garis miring, serta simbol lainnya yang sering dimanfaatkan untuk mengaburkan struktur URL. Selain itu, penelitian ini juga mempertimbangkan rasio digit pada URL dan *hostname*, keberadaan alamat IP pada URL, jumlah subdomain, serta struktur domain yang tidak wajar dan mencurigakan, seperti penggunaan tanda hubung berlebih, pola domain acak, dan penyisipan token tertentu pada path URL. Fitur lain yang digunakan mencakup keberadaan token protokol yang menyesatkan (misalnya *https* pada path), penggunaan layanan pemendek URL (*shortening service*), serta indikasi kata-kata yang umum muncul pada URL phishing. Seluruh fitur yang digunakan berasal langsung dari analisis leksikal dan struktural URL, tanpa melibatkan informasi konten halaman web, namun masih memanfaatkan atribut domain dan reputasi ringan yang tersedia pada dataset. Pemilihan fitur

dilakukan secara selektif untuk memastikan bahwa proses pemodelan hanya bergantung pada karakteristik URL yang relevan, sehingga diharapkan mampu menghasilkan model klasifikasi yang efektif dalam membedakan URL *phishing* dan *legitimate*. Pendekatan ini memungkinkan sistem deteksi bekerja secara independen dari konten website maupun sertifikat keamanan, sehingga lebih fleksibel dan sesuai untuk implementasi deteksi phishing berbasis URL secara umum.

3.2 Metode Penelitian

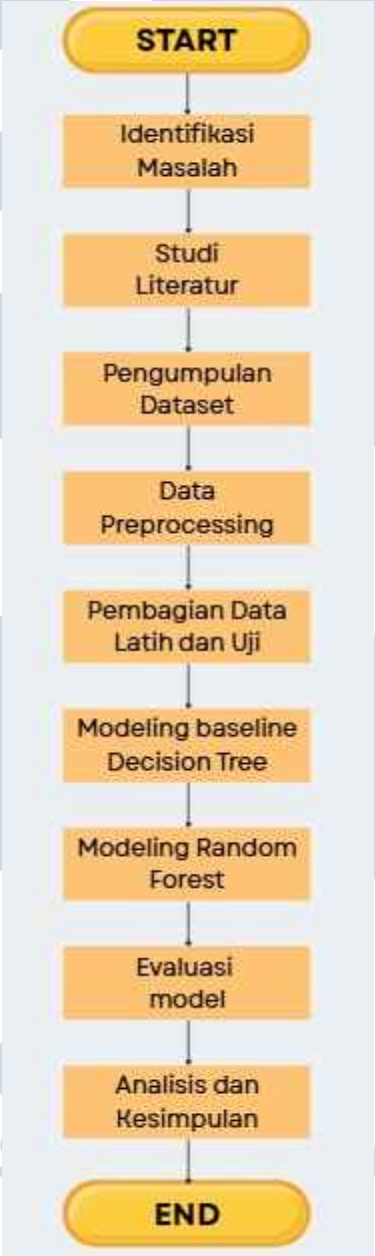
Metode penelitian yang digunakan mengadopsi kerangka kerja CRISP-DM (*Cross-Industry Standard Process for Data Mining*). Kerangka ini dipilih karena menyediakan tahapan yang terstruktur dalam pembelajaran mesin dan *data science*. Metodologi ini mencakup beberapa tahapan utama yang sistematis, dimulai dari studi literatur dan pengumpulan dataset URL. Dilanjutkan dengan *data preprocessing* yang meliputi pembersihan data (*cleaning*) dan rekayasa fitur (*feature engineering*) untuk mengekstrak fitur yang relevan terhadap karakteristik URL. Selanjutnya, dilakukan pembagian data *training* dan *testing*, hingga pelatihan model menggunakan algoritma Random Forest dan diakhiri dengan evaluasi performa model. Setiap tahapan tersebut akan diuraikan secara rinci pada sub-bab berikut.

3.2.1 Tahapan Penelitian

Tahapan penelitian ini dilakukan secara sistematis untuk mencapai tujuan penelitian dalam mendeteksi *phishing* berbasis URL. Penelitian diawali dengan identifikasi permasalahan terkait meningkatnya serangan *phishing* di Indonesia yang banyak disebarkan melalui tautan URL pada media digital seperti email dan aplikasi pesan instan. Selanjutnya, dilakukan pengunduhan data berupa dataset URL *phishing* dan *legitimate* yang diperoleh dari Kaggle dengan judul "*Web Page Phishing Detection Dataset*" oleh Shashwat Tiwari, yang memiliki berbagai atribut untuk merepresentasikan struktur dan karakteristik URL serta informasi domain yang relevan.

Tahap berikutnya adalah *preprocessing* data menggunakan lingkungan pemrograman Python berbasis Google Colab, yang meliputi pembersihan data, penyesuaian format dan tipe data, pengkodean label kelas dengan memetakan *phishing* sebagai nilai 1 dan *legitimate* sebagai nilai 0, penanganan *missing*

value, serta seleksi fitur dengan menghapus kolom yang tidak relevan terhadap karakteristik URL dan atribut domain. Setelah proses tersebut, dilakukan pemodelan menggunakan Decision Tree (CART) sebagai model *baseline* dan Random Forest sebagai model utama dengan pendekatan *ensemble learning*, kemudian dilakukan pengujian dan evaluasi kinerja model untuk menilai kemampuan klasifikasi dalam membedakan URL *phishing* dan *legitimate* tanpa melibatkan implementasi sistem berbasis website.



Gambar 3.1. Tahapan Penelitian.

Flowchart pada Gambar 3.1 menggambarkan tahapan penelitian yang disusun secara sistematis mulai dari identifikasi permasalahan hingga analisis hasil. Setiap tahapan saling berkaitan untuk memastikan proses penelitian berjalan terarah dan terstruktur, dimulai dari pengumpulan dataset URL *phishing* dan *legitimate*, dilanjutkan dengan pembersihan dan preprocessing data, serta seleksi fitur yang relevan terhadap karakteristik URL dan atribut domain. Selanjutnya, dilakukan pemodelan menggunakan Decision Tree (CART) sebagai model dasar (*baseline*) dan Random Forest sebagai model utama untuk proses klasifikasi. Tahap akhir penelitian meliputi evaluasi kinerja model menggunakan metrik klasifikasi guna menilai kemampuan model dalam membedakan URL *phishing* dan *legitimate*. Diagram alur ini memberikan gambaran umum mengenai alur kerja penelitian dalam mendeteksi *phishing* berbasis URL menggunakan pendekatan *machine learning*.

3.2.2 Pendekatan dan Jenis Penelitian

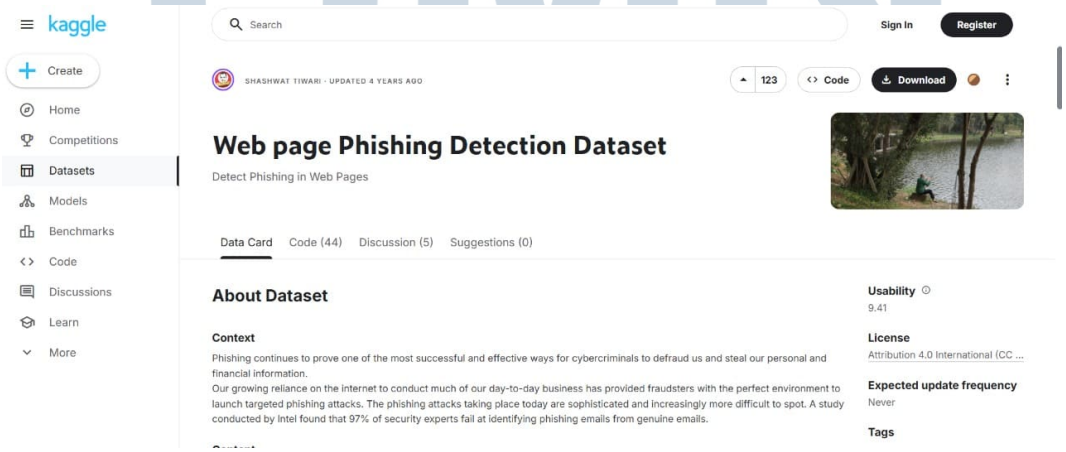
Penelitian ini merupakan jenis penelitian kuantitatif eksperimental yang berfokus pada penerapan algoritma pembelajaran mesin (*machine learning*) berbasis Random Forest (RF) untuk mendeteksi website phishing berdasarkan atribut Karakteristik URL. Algoritma Decision Tree (CART) dipertahankan dan diuji sebagai model *baseline* untuk tujuan perbandingan kinerja model terhadap dataset yang digunakan. Pendekatan kuantitatif digunakan karena penelitian ini melibatkan proses pengumpulan, pengolahan, dan analisis data numerik untuk menghasilkan model klasifikasi yang terukur dan akurat secara statistik. Pendekatan yang digunakan dalam penelitian ini adalah pendekatan machine learning berbasis data mining dengan metode klasifikasi (*classification*). Pendekatan ini dipilih karena sesuai dengan karakteristik dataset yang saya gunakan melalui Kaggle yang memiliki label target berupa dua kategori, yaitu *phishing* dan *legitimate*. Melalui proses klasifikasi, model pembelajaran mesin dilatih untuk mengenali pola tertentu dari sekumpulan fitur karakteristik URL sehingga mampu mengklasifikasikan data baru ke dalam salah satu kelas tersebut. Pemilihan algoritma Random Forest sebagai model utama didasarkan pada kemampuannya sebagai metode *ensemble learning* yang mengombinasikan sejumlah Decision Tree sebagai base classifier, sehingga mampu mengurangi kecenderungan overfitting yang umum terjadi pada Decision Tree tunggal serta meningkatkan akurasi hasil klasifikasi.

3.3 Data Penelitan

Data penelitian memegang peran krusial dalam pembangunan model deteksi phishing berbasis website ini. Data yang digunakan merupakan bahan utama untuk melatih (*training*) dan menguji (*testing*) kinerja algoritma Decision Tree (CART) dan Random Forest. Kualitas, kuantitas, dan relevansi data ini akan berpengaruh langsung terhadap akurasi model yang dihasilkan. Pada sub-bab berikutnya akan diuraikan secara detail mengenai dataset yang digunakan dan teknik pengumpulan data.

3.3.1 Dataset

Dataset yang dijadikan sebagai objek dalam penelitian ini adalah data "*Web Page Phishing Detection Dataset*" yang didapat dari sumber terbuka (*open source data*) pada platform Kaggle melalui tautan https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset?select=dataset_phishing.csv. Dataset ini berisi kumpulan situs web berjumlah 11.429 entri (*record*) dengan kolom berjumlah 89 atribut (*column*) yang merepresentasikan karakteristik dari setiap URL. Setiap sampel pada dataset memiliki berbagai jenis nilai atribut, antara lain, 74 atribut bertipe *integer*, 13 atribut bertipe desimal, 1 atribut string, dan 1 bertipe lainnya, serta pembagian kelas label berupa *phishing* (1) dan *legitimate* (0). Dataset ini disusun untuk mendukung penelitian di bidang keamanan siber, khususnya dalam pengembangan model deteksi phishing berbasis pembelajaran mesin (*machine learning*).



Gambar 3.2. Dataset Kaggle.

3.3.2 Teknik Pengumpulan Data

Pada penelitian ini, proses pengumpulan data dilakukan melalui dokumentasi daring yaitu dengan mengunduh dataset secara langsung dari repositori Kaggle yang telah diverifikasi oleh komunitas. Pemilihan dataset ini didasarkan pada beberapa pertimbangan, antara lain:

a. Dataset bersifat publik dan dapat digunakan untuk penelitian akademik tanpa batasan lisensi.

b. Atribut yang tersedia pada dataset relevan dengan tujuan penelitian karena merepresentasikan karakteristik struktural URL dan atribut domain, seperti panjang URL, jumlah karakter khusus, jumlah *subdomain*, rasio digit, keberadaan alamat IP, serta atribut pendukung berupa *domain age*, *web traffic*, *google index*, dan *page rank*, yang mendukung proses deteksi phishing berbasis URL.

c. Struktur dataset tersedia dalam format standar CSV (*Comma Separated Values*) dan digunakan secara langsung dalam lingkungan pemrograman Python berbasis Google Colab. Format ini memungkinkan setiap atribut terbaca sebagai kolom terpisah sehingga data dapat diproses secara optimal menggunakan bahasa pemrograman Python dan pustaka *Scikit-learn* tanpa memerlukan penyesuaian manual tambahan.

Dataset yang digunakan dalam penelitian ini diperoleh dari repositori Kaggle dengan judul *Web Page Phishing Detection Dataset* yang dipublikasikan oleh Shashwat Work [57]. Dataset tersebut telah dilabeli ke dalam dua kelas, yaitu *phishing* dan *legitimate*, serta memuat berbagai atribut yang merepresentasikan struktur URL dan karakteristik domain sebagai indikator deteksi *phishing* berbasis URL. Sebelum dilakukan proses pemodelan, dataset diperiksa untuk memastikan tidak terdapat duplikasi data, nilai kosong, maupun ketidaksesuaian format kolom. Tahapan ini penting untuk menjamin konsistensi dan keandalan data yang akan digunakan dalam proses *data preprocessing*, *feature extraction*, serta pembangunan model klasifikasi menggunakan algoritma *Random Forest*.

3.4 Variabel Penelitian

Dalam penelitian ini, variabel penelitian merujuk pada atribut atau parameter spesifik yang diidentifikasi dari dataset URL untuk diukur dan dianalisis.

Variabel-variabel ini merupakan inti dari proses pemodelan machine learning dan dikategorikan menjadi dua jenis, yaitu variabel dependen (Y) dan variabel independen (X). Variabel dependen (Y) merepresentasikan kelas target yang akan diprediksi oleh model, yaitu kolom status yang diklasifikasikan ke dalam kategori phishing dan legitimate. Sementara itu, variabel independen (X) terdiri dari sekumpulan fitur karakteristik URL dan atribut domain yang berfungsi sebagai faktor penentu klasifikasi. Hubungan antara kedua variabel dianalisis menggunakan algoritma *machine learning* untuk menghasilkan model yang mampu mengenali pola URL phishing secara efektif.

3.4.1 Variabel Independen (X)

Variabel independen, yang dikenal sebagai variabel bebas pada penelitian ini merupakan sekumpulan fitur yang merepresentasikan karakteristik URL dan digunakan sebagai dasar proses klasifikasi untuk membedakan antara website *phishing* dan *legitimate*. Fitur dibagi menjadi dua kelompok utama, yaitu fitur external dan fitur karakteristik URL (*lexical-based features*). Fitur eksternal terdapat 3 fitur utama yang digunakan, terdiri dari `google_index`, `page_rank`, dan `domain_age`, yang merepresentasikan reputasi dan kredibilitas suatu domain berdasarkan sumber eksternal. Fitur-fitur ini memiliki pengaruh yang signifikan terhadap akurasi deteksi *phishing*, karena website *phishing* umumnya memiliki usia domain yang relatif singkat, reputasi rendah, serta belum terindeks secara optimal oleh mesin pencari dibandingkan dengan website yang *legitimate*.

Sementara itu, seperti pada tabel variabel Independen berdasarkan karakteristik URL 3.1 fitur karakteristik URL berjumlah 23 fitur yang telah ditentukan sesuai dengan relevansi karakteristik URL untuk analisis struktur internal URL tanpa bergantung pada konten halaman website. Fitur-fitur ini mencakup aspek struktur URL dan hostname, frekuensi penggunaan karakter dan simbol khusus, statistik angka dan kata pada URL dan path, serta pola-pola khas phishing seperti penggunaan alamat IP, penyisipan token keamanan, dan kata kunci mencurigakan. Karakteristik tersebut sering dimanfaatkan oleh pelaku *phishing* untuk menyamarkan URL berbahaya agar terlihat sah. Dengan mengombinasikan fitur eksternal dan fitur karakteristik URL, model pembelajaran mesin diharapkan mampu mengenali pola anomali URL secara lebih efektif, sehingga dapat meningkatkan akurasi klasifikasi dalam membedakan URL *phishing* dan *legitimate* tanpa bergantung pada analisis konten halaman website.

Tabel 3.1. Variabel Independen Berdasarkan Karakteristik URL (Bagian 1)

No	Nama Fitur	Penjelasan
1	google_index	Status indeks pada Google Search. URL yang tidak terindeks memiliki probabilitas tinggi sebagai situs phishing baru.
2	page_rank	Skor reputasi website. Situs phishing umumnya memiliki PageRank 0 atau sangat rendah dibandingkan situs resmi.
3	domain_age	Umur domain sejak didaftarkan. Domain phishing cenderung memiliki umur yang sangat muda (¡ 6 bulan).
4	length_url	Panjang total karakter URL. Phisher sering membuat URL yang sangat panjang untuk menyembunyikan domain asli.
5	length_hostname	Panjang karakter hostname. Domain palsu cenderung lebih panjang karena meniru visual domain asli.
6	nb_www	Jumlah teks "www". Phisher sering menyisipkan "www" palsu pada subdomain untuk mengelabui korban.
7	nb_subdomains	Jumlah level subdomain. Phishing sering menggunakan banyak subdomain bertingkat (misal: *paypal.secure.update.com*).
8	shortening_service	Indikasi penggunaan pemendek tautan (bit.ly, tinyurl) untuk menyembunyikan tujuan URL yang sebenarnya.
9	ip	Menunjukkan apakah URL menggunakan alamat IP mentah (misal: http://192.168.1.1) alih-alih nama domain.
10	https_token	Keberadaan token "https" di luar protokol (misal: pada subdomain), memberi kesan aman palsu.
11	nb_dots	Jumlah tanda titik (.) pada URL. Titik yang berlebihan mengindikasikan struktur subdomain yang dimanipulasi.
12	nb_hyphens	Jumlah tanda hubung (-) pada URL. Sering dipakai pada teknik *typosquatting* (misal: face-book.com).
13	nb_slash	Jumlah garis miring (/) pada URL. Menunjukkan kedalaman direktori yang seringkali lebih dalam pada phishing.
14	nb_qm	Jumlah tanda tanya (?). Digunakan untuk menyisipkan parameter query jahat atau pengalihan (redirect).
15	nb_eq	Jumlah tanda sama dengan (=). Mengindikasikan banyaknya variabel parameter yang dikirim melalui URL.

Tabel 3.2. Variabel Independen Berdasarkan Karakteristik URL (Bagian 2)

No	Nama Fitur	Penjelasan
16	nb_underscore	Jumlah garis bawah (_) pada URL. Sering ditemukan pada penamaan file atau folder situs phishing yang tidak baku.
17	ratio_digits_url	Rasio jumlah angka terhadap total panjang URL. URL phishing sering mengandung deretan angka acak yang tinggi.
18	ratio_digits_host	Rasio jumlah angka pada hostname. Domain phishing sering menggunakan angka untuk membedakan subdomain palsu.
19	char_repeat	Tingkat pengulangan karakter. Phishing sering menggunakan pengulangan huruf (misal: paypaaal) untuk menipu mata.
20	length_words_raw	Jumlah kata total dalam string URL. URL phishing cenderung memiliki deskripsi verbal yang berlebihan.
21	avg_words_raw	Rata-rata panjang kata dalam URL. Kata-kata acak/panjang sering muncul pada URL hasil generate mesin (DGA).
22	longest_words_raw	Panjang kata terpanjang dalam URL. Mendeteksi token enkripsi atau nama file acak yang sangat panjang.
23	avg_word_path	Rata-rata panjang kata pada path. Path phishing sering berisi direktori dengan nama acak.
24	longest_word_path	Kata terpanjang pada path. Mengidentifikasi file jahat atau token sesi yang disisipkan di URL.
25	phish_hints	Jumlah kata kunci pancingan (seperti: *login, secure, update, verify, banking*) yang terdapat pada URL.
26	shortest_word_host	Panjang kata terpendek pada hostname. Digunakan untuk mendeteksi struktur domain tidak alami, di mana domain phishing sering memiliki token singkat dan acak pada subdomain atau hostname.

Berdasarkan Tabel 3.1 dan 3.2, fitur-fitur karakteristik URL tersebut dapat dikategorikan menjadi beberapa kelompok fungsional. Pertama, fitur berbasis panjang dan kuantitas seperti length_url, nb_dots, dan nb_subdomains yang mendeteksi kompleksitas struktur URL yang tidak wajar. Kedua, fitur berbasis karakter khusus seperti nb_hyphens dan nb_underscore yang sering muncul dalam teknik *typosquatting* atau penyamaran nama domain resmi. Ketiga, fitur berbasis statistik kata dan angka, contohnya ratio_digits_url dan phish_hints, yang mengidentifikasi keberadaan istilah pancingan (seperti '*login*' atau '*verify*') serta penggunaan string acak yang dihasilkan secara otomatis oleh mesin. Terakhir, fitur

teknis seperti ip dan shortening_service digunakan untuk mengenali upaya pelaku dalam menyembunyikan identitas asli domain tujuan.

3.4.2 Variabel Dependen (Y)

Varibel dependen dalam penelitian ini merupakan hasil klasifikasi dari model yang dibangun menggunakan algoritma Random Forest dan Decision Tree (CART). Variabel ini menunjukkan status suatu website berdasarkan fitur-fitur yang dimiliki. Variabel dependen direpresentasikan oleh atribut `status` pada dataset, dengan dua kemungkinan nilai sebagai berikut:

Tabel 3.3. Keterangan Variabel Dependen (Label Keluaran)

Nilai	Keterangan
1	Website dikategorikan sebagai *Phishing*
0	Website dikategorikan sebagai *Legitimate*

Model klasifikasi dipelajari dari sekumpulan fitur (variabel independen) untuk memprediksi label keluaran suatu situs web. Secara ringkas, hubungan antara fitur dan keluaran tersebut dapat dituliskan sebagai persamaan fungsi:

$$f(X) \rightarrow Y$$

di mana X merepresentasikan sekumpulan fitur yang diekstraksi dari karakteristik URL dan reputasi domain, sedangkan Y menunjukkan hasil klasifikasi berupa *phishing* atau *legitimate*.

Dengan kata lain, model berusaha menangkap pola-pola pada fitur X, misalnya panjang URL, umur domain, atau validitas sertifikat, lalu menggunakan pola tersebut untuk menentukan apakah sebuah alamat web berpotensi berbahaya. Pendekatan ini menggunakan algoritma klasifikasi *ensemble* (*Random Forest*) untuk menghasilkan prediksi yang akurat dan stabil.

3.5 Proses Analisis dan Pembangunan Model

Proses analisis dan pembangunan model pada penelitian ini dilakukan melalui serangkaian tahapan yang terstruktur untuk memastikan bahwa data yang digunakan memiliki kualitas yang baik serta sesuai dengan tujuan penelitian. Tahapan ini mencakup pemahaman data, *preprocessing*, seleksi fitur, pemodelan

(*modeling*), serta evaluasi kinerja model pembelajaran mesin. Setiap tahapan saling berkaitan dan dirancang untuk menghasilkan model deteksi phishing berbasis karakteristik URL yang akurat, stabil, dan memiliki kemampuan generalisasi yang baik. Oleh karena itu, proses analisis diawali dengan tahap *data understanding* sebagai fondasi utama sebelum dilakukan tahapan pengolahan data dan pembangunan model klasifikasi. Bagian ini menjabarkan seluruh tahapan yang dilakukan dalam proses penelitian.

3.5.1 Data Collection

Pada tahapan *Data Collection*, penelitian ini diawali dengan pengunduhan dataset yang digunakan sebagai dasar dalam proses deteksi website *phishing*. Dataset diperoleh dari platform Kaggle yang menyediakan data *phishing* dalam format terstruktur dan telah banyak digunakan pada penelitian sebelumnya. Setelah dataset diunduh, data kemudian diunggah ke lingkungan Google Colab untuk memudahkan proses pengolahan, analisis, dan implementasi algoritma pembelajaran mesin. Selanjutnya, dilakukan pemeriksaan awal terhadap dataset untuk memastikan format file, jumlah data, serta kesesuaian atribut yang tersedia dengan kebutuhan penelitian. Tahapan ini bertujuan untuk memastikan bahwa data yang digunakan telah siap dan layak untuk diproses lebih lanjut pada tahap data *preprocessing*.

Gambar 3.3. Tahapan Data Collection.

3.5.2 Tahapan Pengolahan Data (Data Preprocessing)

Setelah memahami isi dataset, tahap berikutnya adalah memastikan data siap digunakan untuk proses pengolahan model penelitian. Tahapan data preprocessing merupakan tahapan untuk membersihkan data (*cleaning*) dari error, menormalkan format, dan menyiapkan struktur agar sesuai dengan algoritma Random Forest dan Decision Tree (CART). Tahapan ini dilakukan untuk memastikan data bersih, konsisten, dan memiliki format yang sesuai sehingga dapat meningkatkan kinerja model dalam mendeteksi URL phishing.

Tahapan data preprocessing yang dilakukan pada penelitian ini adalah sebagai berikut:

a. *Feature Selection*

Feature selection dilakukan dengan mempertahankan fitur-fitur yang relevan dan representatif terhadap karakteristik URL serta reputasi domain yang digunakan dalam proses klasifikasi. Pada penelitian ini, fitur yang dipilih terdiri dari dua kelompok utama, yaitu fitur eksternal dan fitur karakteristik URL. Fitur eksternal meliputi `google_index`, `page_rank`, dan `domain_age`, yang merepresentasikan reputasi, popularitas, dan usia suatu domain, serta dipertahankan karena memiliki pengaruh signifikan dalam membedakan website *phishing* dan *legitimate*. Selain itu, digunakan fitur karakteristik URL (*lexical-based features*) yang mencerminkan struktur dan pola teknis URL, meliputi struktur utama URL dan *hostname* seperti `length_url`, `length_hostname`, `nb_www`, `nb_subdomains`, dan `shortening_service`; frekuensi penggunaan karakter dan simbol khusus seperti `nb_dots`, `nb_hyphens`, `nb_slash`, `nb_qm`, `nb_eq`, dan `nb_underscore`; statistik angka dan kata pada URL dan *hostname* seperti `ratio_digits_url`, `ratio_digits_host`, `char_repeat`, `length_words_raw`, `avg_words_raw`, `longest_words_raw`, `avg_word_path`, `longest_word_path`, dan `shortest_word_host`, serta pola khas phishing seperti `phish_hints`, `ip`, dan `https_token`. Fitur-fitur lain yang tidak relevan dihapus agar model dapat fokus mempelajari pola URL yang membedakan website *phishing* dan *legitimate* secara lebih efektif dan efisien.

b. Penghapusan Kolom Duplikat

Setelah proses seleksi fitur dilakukan, tahapan selanjutnya adalah melakukan penghapusan data duplikat untuk memastikan bahwa setiap baris data bersifat

unik dan tidak terjadi pengulangan informasi pada dataset. Berdasarkan hasil pemeriksaan sebelumnya, ditemukan sebanyak 401 baris data duplikat dari total 11.430 baris data awal. Keberadaan data duplikat ini berpotensi menimbulkan bias dalam proses pelatihan model karena pola data yang sama dapat terhitung lebih dari satu kali.

Oleh karena itu, seluruh data duplikat tersebut dihapus dari dataset. Setelah proses penghapusan dilakukan, jumlah data berkurang menjadi 11.029 baris. Tahapan ini bertujuan untuk meningkatkan kualitas data, menjaga integritas dataset, serta memastikan bahwa proses pembelajaran algoritma *machine learning* berlangsung secara adil dan representatif. Dataset hasil pembersihan ini kemudian digunakan pada tahap prapemrosesan lanjutan dan pemodelan.

c. *Handle Missing Value*

Tahap pembersihan data dilakukan untuk memastikan tidak terdapat data kosong atau duplikasi yang dapat mengganggu proses pelatihan model. Fokus utama pada tahap ini adalah penanganan data yang duplikat, dengan menghapus baris data yang identik untuk mencegah kebocoran data (*data leakage*) dan bias pada model.

d. *Label Encoding*

Tahapan awal data preprocessing dilakukan transformasi terhadap variabel target status yang semula berupa data kategorikal dengan tipe data string ”*phishing*” dan ”*legitimate*” menjadi data numerik. Label phishing dikonversi menjadi nilai 1, sedangkan label legitimate dikonversi menjadi nilai 0. Proses ini diperlukan karena algoritma Random Forest dan Decision Tree (CART) hanya dapat memproses dan mengkalkulasi data dalam bentuk numerik. Selain itu, penggunaan representasi numerik juga mempermudah proses evaluasi dan perhitungan metrik kinerja model.

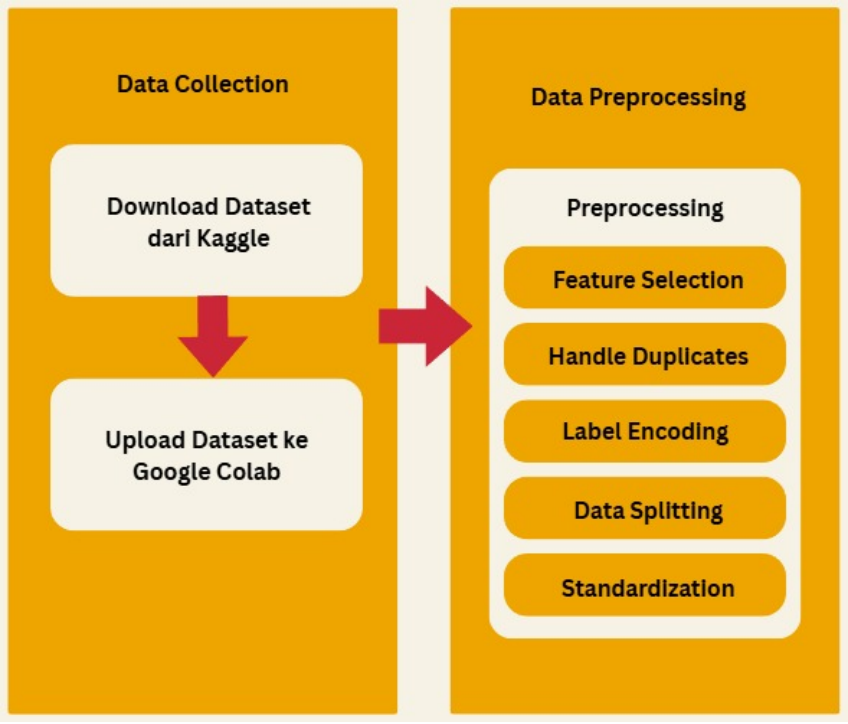
e. *Standardization* (*StandardScaler*)

Tahap standarisasi dilakukan untuk menyeragamkan skala nilai pada seluruh fitur numerik menggunakan fungsi *StandardScaler* karena dataset yang digunakan memiliki range nilai yang beragam dan sangat jauh perbedaannya. Beberapa fitur seperti `domain_age` memiliki nilai yang relatif besar, sementara fitur rasio seperti `ratio_digits_url` berada pada rentang yang lebih kecil. Tanpa standardisasi, fitur dengan nilai yang lebih besar berpotensi

memberikan pengaruh yang lebih dominan dalam proses pembelajaran model. Pada penelitian ini, standarisasi diintegrasikan langsung ke dalam *pipeline* pemodelan dan tetap diterapkan meskipun algoritma *Random Forest* relatif tahan terhadap perbedaan skala data. Selain itu, standarisasi dilakukan untuk menjaga konsistensi pipeline serta memungkinkan replikasi penelitian dengan algoritma lain yang sensitif terhadap skala data. Hal ini dilakukan sebagai upaya mengikuti praktik terbaik dalam *machine learning* serta memastikan kontribusi setiap fitur tetap seimbang dalam proses pelatihan dan evaluasi model.

f. *Stratified Train–Test Split*

Setelah seluruh tahapan data preprocessing selesai, dataset dibagi menjadi data latih dan data uji dengan rasio 80% data latih dan 20% data uji. Pembagian data dilakukan menggunakan metode *stratified train–test split* untuk memastikan proporsi kelas phishing dan legitimate tetap seimbang pada kedua subset data. Pendekatan ini bertujuan untuk menghindari bias kelas selama proses pelatihan dan evaluasi model, sehingga hasil pengujian dapat merepresentasikan performa model secara lebih objektif dan adil.



Gambar 3.4. Tahapan *Data Preprocessing*.

3.5.3 Penerapan Model Klasifikasi (Decision Tree dan Random Forest)

Pada tahapan ini dilakukan modeling algoritma Decision Tree (DT) dengan metode *Classification and Regression Trees* (CART) sebagai model baseline dan Random Forest (RF) sebagai model utama yang dioptimasi untuk mendapatkan akurasi terbaik. Model DT diimplementasikan terlebih dahulu menggunakan metode *Classification and Regression Trees* (CART). Pemilihan DT sebagai baseline didasarkan pada keunggulannya dalam interpretasi (*interpretability*) dan kemampuannya menangani masalah klasifikasi biner, seperti deteksi phishing. Algoritma CART membangun struktur pohon keputusan yang terdiri dari simpul akar (*root node*), cabang (*branches*), dan daun (*leaf node*). Proses pembentukan pohon dilakukan secara rekursif menggunakan prinsip *Divide and Conquer*, di mana *Gini Index* digunakan sebagai pengukuran ketidakmurnian (*impurity*) untuk menentukan pemisahan data terbaik. Meskipun Decision Tree mampu menghasilkan model yang efektif untuk menganalisis pola *phishing*, model tunggal ini memiliki risiko variansi tinggi dan *overfitting*. Untuk mengatasi kelemahan tersebut dan mencapai kinerja klasifikasi yang optimal, kami mengimplementasikan Random Forest (RF). Secara umum, Random Forest adalah teknik *ensemble* yang memanfaatkan sekumpulan besar *Decision Tree* (dibangun dengan CART). Meskipun metode konvensional menggunakan *Bagging*, penelitian ini memodifikasi pendekatan tersebut dengan menggunakan distribusi data penuh, sehingga keberagaman model sepenuhnya bergantung pada penerapan *Random Feature Selection* pada setiap pemisahan *node*. Untuk mengoptimasi kinerja kedua model, dilakukan *Hyperparameter Tuning* secara ekstensif guna menemukan konfigurasi parameter terbaik.

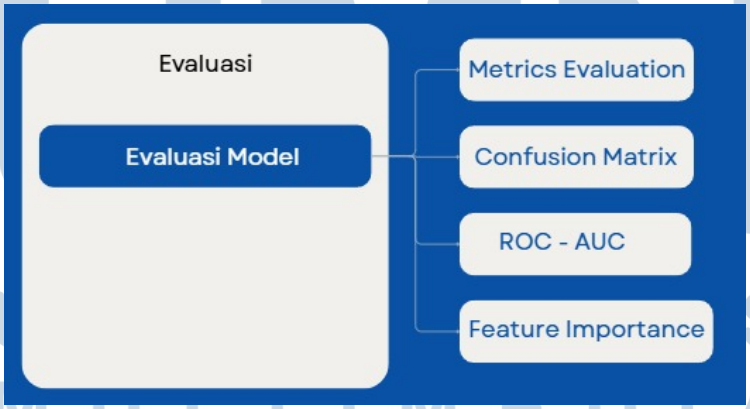
Gambar 3.5. Tahapan *Modeling*.

Berdasarkan alur tahapan modeling pada gambar 3.5 Proses pengembangan sistem deteksi dalam penelitian ini mengikuti alur sistematis yang digambarkan dalam skema Tahapan *Modeling*. Proses dimulai dengan pembentukan *Pipeline* yang mencakup tahap preprocessing menggunakan *Imputer* untuk menangani data yang hilang dan *Scaler* untuk standarisasi fitur agar model memiliki performa yang optimal. Tahap selanjutnya adalah implementasi dua algoritma utama, yaitu Decision Tree (DT) sebagai model *baseline* dan Random Forest (RF) sebagai model utama berbasis ensemble learning. Untuk mendapatkan hasil terbaik, dilakukan proses *Hyperparameter Tuning* guna mencari parameter optimal bagi model. Tahapan ini diakhiri dengan evaluasi model *Random Forest* serta analisis perbandingan kinerja antara Decision Tree dan Random Forest melalui metrik evaluasi seperti akurasi, *precision*, *recall*, dan *F1-score*. Alur ini memastikan bahwa model yang dihasilkan tidak hanya akurat, tetapi juga memiliki kemampuan generalisasi yang baik dalam mendeteksi ancaman phishing yang sebenarnya.

3.5.4 Evaluasi Model

Tahapan evaluasi model bertujuan untuk menilai kinerja dan efektivitas kedua model klasifikasi yaitu *Decision Tree* dan *Random Forest* yang telah dibangun

pada tahap *modeling*. Pada tahap ini, model yang telah dilatih akan diuji menggunakan data uji yang belum pernah digunakan selama proses penelitian. Evaluasi dilakukan dengan beberapa indikator yang saling melengkapi untuk menghasilkan kinerja yang baik. *Confusion Matrix* memberikan gambaran jumlah prediksi yang benar dan salah untuk masing-masing kelas. Dari proses inilah akan didapatkan perhitungan ukuran metrik yang lebih spesifik terdiri dari, *Accuracy* sebagai ukuran kebenaran prediksi, *Precision* sebagai penunjuk jumlah prediksi phishing yang benar-benar berbahaya, *Recall* mengukur kemampuan model menangkap seluruh kasus phishing yang nyata, dan *F1 Score* yang menggabungkan *precision* dan *recall* menjadi satu nilai seimbang. Melalui kombinasi metrik tersebut, evaluasi tidak hanya menilai performa model secara keseluruhan, namun memudahkan dalam mengidentifikasi potensi kesalahan seperti *False Positive* (situs sah yang salah terdeteksi sebagai phishing) dan *False Negative* (situs phishing yang lolos sebagai legitimate). Dengan demikian, hasil evaluasi memberikan dasar yang kuat untuk membandingkan kinerja antara algoritma Decision Tree (*Model Baseline*) dan *Random Forest* (Model Utama). Perbandingan ini akan membenarkan pemilihan algoritma Random Forest sebagai model terbaik dalam konteks keamanan siber, serta memastikan model yang dihasilkan memiliki tingkat reliabilitas tinggi untuk digunakan pada sistem deteksi *phishing* di dunia nyata. Berikut adalah konsep tabel *confusion matrix* yang digunakan pada model klasifikasi:



Gambar 3.6. Tahapan *Evaluation*.

Tabel 3.4. Tabel Konsep *Confusion Matrix* pada Model Klasifikasi

Kelas Aktual / Prediksi	**Legitimate (0)**	**Phishing (1)**
Legitimate (0)	True Negative (TN)	False Positive (FP)
Phishing (1)	False Negative (FN)	True Positive (TP)

a. **True Positive (TP)**: Jumlah situs *phishing* yang berhasil diklasifikasikan dengan benar sebagai *phishing*.

b. **True Negative (TN)**: Jumlah situs *legitimate* yang berhasil diklasifikasikan dengan benar sebagai *legitimate*.

c. **False Positive (FP)**: Situs *legitimate* yang salah diklasifikasikan sebagai *phishing*.

d. **False Negative (FN)**: Situs *phishing* yang salah diklasifikasikan sebagai *legitimate*.

Rumus-rumus Pengukuran

A Akurasi (*Accuracy*)

Akurasi menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar dari seluruh data uji. Nilai akurasi yang tinggi mengindikasikan bahwa model mampu mengenali sebagian besar sampel dengan benar.

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ &= \frac{1043+1062}{1043+1062+51+50} \\ &= \frac{2105}{2206} \\ &= 0.9542 \end{aligned} \tag{3.1}$$

B Presisi (*Precision*)

Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Dalam konteks ini, presisi mengukur berapa banyak situs yang diklasifikasikan sebagai phishing dan benar-

benar merupakan phishing.

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ &= \frac{1043}{1043+51} \\ &= \frac{1043}{1094} \\ &= 0.9534 \end{aligned} \tag{3.2}$$

C *Recall* atau *Sensitivity*

Recall atau sensitivitas menunjukkan kemampuan model untuk menemukan seluruh kasus positif yang sebenarnya ada. Dalam konteks ini, recall menunjukkan seberapa banyak situs phishing yang berhasil terdeteksi oleh model.

$$\begin{aligned} Recall &= \frac{TP}{TP+FN} \\ &= \frac{1043}{1043+50} \\ &= \frac{1043}{1093} \\ &= 0.9543 \end{aligned} \tag{3.3}$$

D *F1-Score*

F1-Score merupakan rata-rata harmonik antara nilai Precision dan Recall. Metrik ini digunakan untuk menyeimbangkan kedua ukuran tersebut, terutama ketika dataset tidak seimbang (jumlah phishing dan legitimate berbeda jauh).

$$\begin{aligned} F1\text{-}Score &= \frac{2 \times Precision \times Recall}{Precision + Recall} \\ &= \frac{2 \times 0.9534 \times 0.9543}{0.9534 + 0.9543} \\ &= 0.9538 \end{aligned} \tag{3.4}$$

3.5.5 Proses Pembentukan Pohon Keputusan

Pada proses penelitian menggunakan algoritma Decision Tree dan Random Forest melalui proses pembentukan pohon keputusan. Hanya saja pembedanya pada Decision Tree hanya pohon keputusan tunggal, sedangkan Random Forest yang merupakan metode ensemble terdiri dari beberapa Decision Tree atau pohon keputusan yang kemudian diambil kesimpulan berupa voting. Pada pembangunan pohon menggunakan kriteria pemisahan (splitting criterion) yang disebut Gini Impurity. Gini impurity adalah ukuran ketidakmurnian atau ketidakteraturan dalam sekumpulan data. Gini Impurity dipilih karena komputasinya yang cenderung lebih cepat dibandingkan Entropy. Nilai Gini Impurity mengukur tingkat ketidakmurnian dari sebuah dataset. Semakin kecil nilai Gini, semakin murni klasifikasi pada node tersebut. Perhitungan Gini Impurity dilakukan menggunakan persamaan berikut.

$$Gini(D) = 1 - \sum_{i=1}^{c} (p_i)^2 \tag{3.5}$$

Keterangan:

G : Nilai *Gini Impurity*

c : Jumlah kelas (misalnya: *Phishing* dan Aman)

p_i : Peluang atau proporsi sampel yang masuk ke dalam kelas i pada *node* tersebut.

Proses algoritma dalam membentuk pohon keputusan adalah sebagai berikut:

a. Inisialisasi (Root Node)
   Algoritma memulai dengan menempatkan seluruh data latih (D) ke dalam node akar. Nilai *Gini Impurity* awal dihitung untuk mengetahui tingkat ketidakmurnian data sebelum dilakukan pemecahan.

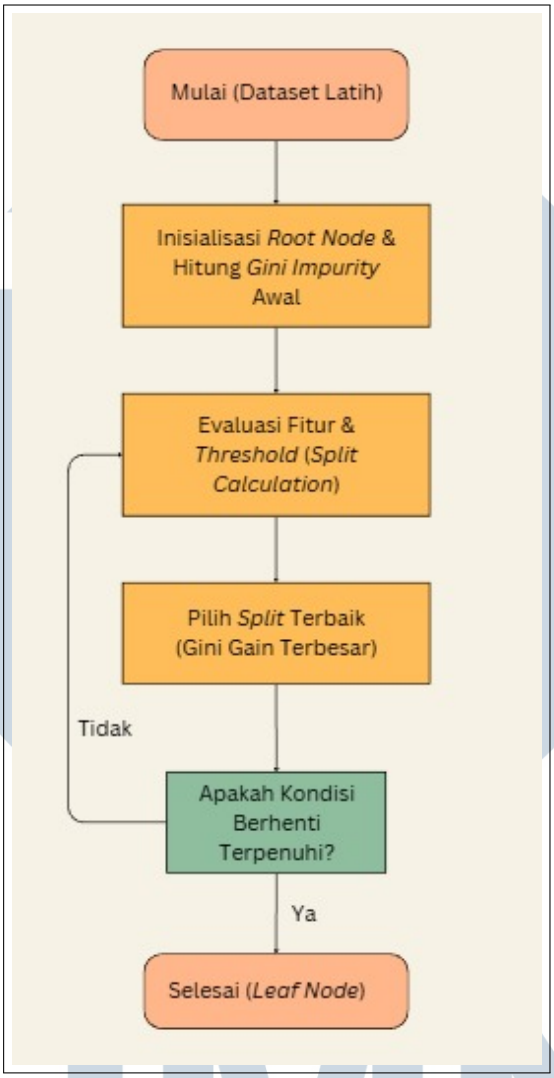
b. Evaluasi Fitur (Feature Evaluation)
   Algoritma menguji seluruh fitur seperti *google_index, page_rank* untuk dievaluasi pada setiap calon pemecahan. Ini dilakukan untuk mengurangi korelasi antar pohon.

c. *Split Calculation*
   Untuk setiap fitur kandidat, algoritma menghitung *Weighted Gini Impurity*

dari node anak yang akan terbentuk jika data dipecah menggunakan fitur tersebut.

d. Memilih Split Terbaik
   Algoritma menentukan fitur dan nilai ambang batas (*threshold*) yang menurunkan Gini Impurity (*Gini Gain*) terbesar. Seperti pada node akar, fitur *google_index* dengan *threshold* ≤ -0.062 dipilih karena menghasilkan pemisahan kelas *Phishing* dan *Legitimate* yang paling murni dibanding fitur lain.

e. Iterasi (*Recursive Partitioning*)
   Diulang seluruh proses diatas pada node *child* hingga kondisi berhenti terpenuhi (misalnya node sudah murni berisi hanya satu kelas URL, atau kedalaman pohon mencapai maksimum).



Gambar 3.7. Proses Pembentukan Pohon Keputusan.

Berdasarkan alur proses pembentukan pohon keputusan pada gambar 3.7, dimulai dengan inisialisasi dataset latih untuk menentukan *Root Node* dan menghitung nilai *Gini Impurity* awal. Selanjutnya, dilakukan evaluasi fitur dan threshold melalui *Split Calculation* untuk mencari pembagian data paling optimal. Algoritma kemudian memilih *split* dengan *Gini Gain* terbesar agar setiap cabang menjadi lebih homogen. Tahapan ini dilakukan secara iteratif hingga kondisi berhenti terpenuhi, yang kemudian berakhir pada Leaf Node sebagai hasil klasifikasi akhir.

3.5.6 Ekstraksi Fitur (Feature Extraction)

Tahapan ekstraksi fitur merupakan proses krusial untuk mentransformasi URL mentah (raw URL) yang bersifat tekstual menjadi representasi numerik yang dapat dipahami oleh algoritma *Random Forest*. Secara teknis, algoritma *machine learning* tidak dapat mengolah data teks secara langsung. Diperlukan proses pemecahan defenisi setiap komponen pada URL menjadi sekumpulan fitur karakteristik atau vektor fitur yang merepresentasikan pola keamanan sebuah situs. Dalam penelitian ini, dilakukan ekstraksi terhadap 26 fitur kunci yang mencakup aspek leksikal dan reputasi domain. Proses ini bertujuan untuk memberikan nilai kuantitatif pada setiap indikator yang dicurigai sebagai taktik phishing, seperti penggunaan karakter spesial yang berlebihan, panjang URL yang tidak wajar, hingga keberadaan kata kunci sensitif dalam struktur alamat tersebut. Dengan membedah satu struktur URL menjadi komponen-komponen diskrit seperti bagian domain name, hostname, path, query, sehingga algoritma dapat mengidentifikasi korelasi antar fitur secara terkalkulasi melalui nilai confidence atau kepercayaan. Nilai Confidence atau tingkat keyakinan dalam sistem deteksi phishing ini direpresentasikan melalui nilai probabilitas yang dihasilkan oleh fungsi matematis pada algoritma Random Forest.

Hasil dari tahap ekstraksi ini adalah sebuah vektor data yang memungkinkan algoritma Random Forest melakukan kalkulasi probabilitas melalui mekanisme voting pada setiap pohon keputusan. Melalui representasi numerik ini, model tidak hanya mengenali URL secara harfiah, tetapi mampu memahami pola struktural yang membedakan antara situs legitimate yang sah dengan situs phishing yang mencoba mengelabui pengguna. Efektivitas deteksi sangat bergantung pada kualitas ekstraksi ini, karena fitur-fitur inilah yang menjadi basis bagi model dalam menentukan ambang batas keputusan (decision *threshold*) pada tahap pengujian data baru. Ambang batas 50% atau threshold 0,5 adalah ”titik tengah” yang digunakan model untuk mengambil keputusan final. Karena algoritma Random Forest bekerja dengan cara mengumpulkan suara (*voting*) dari banyak pohon keputusan, angka 0,5 ini menjadi penentu: jika lebih dari 50% pohon mendeteksi ciri phishing, maka URL tersebut langsung dicap sebagai bahaya. Secara sederhana, ini adalah batas netral untuk menentukan mana yang masuk kategori aman dan mana yang tidak. Jika nilai *confidence* berada sangat dekat di angka ini (seperti kasus Bitly tadi), itu tandanya model sedang ”ragu-ragu” karena URL tersebut memiliki campuran ciri-ciri baik dan buruk yang hampir seimbang.