PERANCANGAN APLIKASI DETEKSI *PHISHING* BERBASIS WEB MENGGUNAKAN ALGORITMA *DECISION TREE* PADA KOMUNITAS PANGLIMA DI PONDOK PESANTREN SUNAN PLUMBON

Proposal Penelitian Disusun Sebagai Persyaratan Penyusunan Skripsi Guna Memperoleh Gelar Sarjana Strata Satu (S1)

Diajukan oleh

KURNIA AINUN NAJAH

203100088

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS KOMPUTER DAN TEKNIK

UNIVERSITAS ALMA ATA

YOGYAKARTA

2024

LEMBAR PERSETUJUAN

Proposal Skripsi

PERANCANGAN APLIKASI DETEKSI *PHISHING* BERBASIS WEB MENGGUNAKAN ALGORITMA *DECISION TREE* PADA KOMUNITAS PANGLIMA DI PONDOK PESANTREN SUNAN PLUMBON

Diajukan Oleh:

KURNIA AINUN NAJAH

203100088

Telah Memenuhi Syarat dan Disetujui untuk Diseminarkan di Program Studi Sistem Informasi

Fakultas Komputer dan Teknik Universitas Alma Ata

Yogyakarta, ...

Pebimbing

Tri Rochmadi, S.Kom., M.Kom.

NIK. 12201720538

LEMBAR PENGESAHAN

Proposal Skripsi

PERANCANGAN APLIKASI DETEKSI *PHISHING* BERBASIS WEB MENGGUNAKAN ALGORITMA *DECISION TREE* PADA KOMUNITAS PANGLIMA DI PONDOK PESANTREN SUNAN PLUMBON

Yang dipersiapkan dan disusun oleh:

KURNIA AINUN NAJAH

203100088

Telah memenuhi syarat dan dinyatakan disetujui untuk dilakukan penelitian pada

Tanggal.....

Ketua Penguji
Tanggal, ...

Tri Rochmadi, S.Kom., M.Kom.

Mengetahui,
Ketua Program Studi Sistem Informasi

Tri Rochmadi, S.Kom., M.Kom.
NIK. 12201720538

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama	:	KURNIA AINUN NAJAH
NIM	:	203100088
Program Studi	:	SISTEM INFORMASI
Fakultas	:	KOMPUTER DAN TEKNIK

Dengan ini, saya menyatakan dengan sesungguhnya bahwa Skripsi yang berjudul "**Perancangan Aplikasi Deteksi *Phishing* Berbasis Web Menggunakan Algoritma *Decision Tree* Pada Komunitas Panglima Di Pondok Pesantren Sunan Plumbon**" adalah hasil karya peneliti sendiri, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain kecuali yang tertulis di dalam naskah ini dan disebutkan dalam daftar pustaka sesuai dengan kriteria etika penulisan ilmiah yang baku. Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa Skripsi ini merupakan hasil karya orang lain maka saya bersedia menerima resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, (20/Mei/2024)

Yang Membuat Pernyataan,

Materai

(Kurnia Ainun Najah)

EKSEKUTIF SUMMARY

Ancaman siber semakin berkembang seiring dengan kemajuan teknologi digital. Menyadari pentingnya keamanan dalam lingkungan internet, kebutuhan akan keamanan digital semakin meningkat, terutama di lingkungan komunitas Panglima di Pondok Pesantren Sunan Plumbon. Anggota komunitas sering kali menjadi korban penipuan *phishing* melalui pesan yang mengarah ke situs web palsu. Teknologi seperti deteksi URL *phishing* menggunakan algoritma *decision tree* dapat menjadi salah satu langkah mengurangi risiko serangan siber. Teknologi deteksi URL *phishing* akan menggunakan algoritma *decision tree* yang berbasis website dengan menggunakan metodologi *waterfall*, di mana tahap-tahapnya meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan *deployment*. Aplikasi diimplementasikan menggunakan *framework Flask* dan di-*hosting* di platform Heroku untuk kemudahan akses dan penggunaan. Melalui aplikasi ini, anggota komunitas Panglima dapat memasukkan URL yang mencurigakan, dan aplikasi akan menganalisisnya untuk menentukan apakah itu situs web *phishing* atau tidak. Hasil analisis ditampilkan kepada pengguna, agar dapat membantu menghindari ancaman *phishing*. Dengan demikian, aplikasi ini diharapkan dapat memberikan perlindungan yang lebih baik bagi anggota komunitas Panglima terhadap serangan *phishing* yang semakin canggih dan merugikan.

Kata Kunci: Website Deteksi, Phishing, Decision tree, Waterfall, Black box

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam era globalisasi informasi, di mana konektivitas dan akses internet telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari. Penggunaan internet ini juga dimanfaatkan dalam kegiatan bisnis secara online, termasuk Usaha Mikro, Kecil, dan, Menengah (UMKM). Berdasarkan Survei *E-commerce* yang dilakukan Badan Pusat Statistik (BPS) tahun 2021, terdapat 63,52% responden pelaku usaha menggunakan layanan internet untuk pemasaran digital, meskipun hanya 8% yang menerima pelatihan, sedangkan 55,5% menggunakan internet untuk memesan bahan mentah dan 44,3% untuk komunikasi internal perusahaan[1]. Kemudahan dalam mengakses informasi, transaksi online, hingga pemasaran digital menjadi keuntungan yang diperoleh UMKM. Komunitas Panglima di Pondok Pesantren Sunan Plumbon Temanggung, telah memanfaatkan internet untuk meningkatkan daya saing dan memperluas pasar. Namun, internet juga memberikan tantangan berupa semakin beragamnya kejahatan siber di ranah digital [2].

Saat popularitas penggunaan media sosial meningkat di seluruh dunia, pelaku kejahatan dunia siber mulai mengeksploitasi kesempatan tersebut untuk meraih keuntungan, yang umumnya melalui cara-cara lama seperti *phishing* untuk pencurian data atau informasi yang selanjutnya dapat dimanfaatkan dari informasi

yang diperoleh [3]. *Phishing* merupakan bagian dari kejahatan siber yang semakin sering terjadi dengan maraknya kegiatan kriminal melalui jaringan komputer [4]. Ketika serangan *phishing* berhasil, munculnya tanda-tanda konkrit seperti kasus kehilangan akun online, penipuan finansial, atau bahkan identitas yang dicuri [5]. Pengguna seringkali menyadari terlambat bahwa mereka telah terjebak dalam perangkap *phishing*, dan dampaknya dapat mencakup kerugian finansial, hilangnya data pribadi, dan berkurangnya kepercayaan pada keamanan daring secara keseluruhan. Karena itu, memahami gejala dan tanda serangan *phishing* menjadi kunci penting dalam melindungi diri dan organisasi dari konsekuensi yang merugikan.

Peningkatan frekuensi serangan *phishing* terhadap UMKM telah menjadi tren yang mencemaskan dalam lanskap keamanan siber. Pada studi Cisco yang berjudul "Keamanan Siber untuk UMKM: Bisnis Asia Pasifik Mempersiapkan Pertahanan Digital", yang melibatkan lebih dari 3.700 UMKM di 14 kawasan Asia-Pasifik, termasuk Indonesi, pada April hingga Juli 2021, menunjukkan 56% UMKM mengalami kejahatan siber dalam setahun[6].

Komunitas Panglima di Pondok Pesantren Sunan Plumbon merupakan sebuah komunitas yang terdiri dari sejumlah pedagang yang aktif berjualan di sekitar lingkungan pesantren tersebut. Pada masa pandemi COVID-2019, kegiatan bisnis online meningkat. Beberapa anggota komunitas telah mengalami insiden serangan *phishing* pada masa itu, di mana mereka menerima pesan yang mengandung tautan yang mengarah ke situs web palsu yang bertujuan untuk mencuri informasi pribadi atau keuangan. Salah satu insiden yang dialami adalah

ketika anggota komunitas menerima pesan yang menjanjikan bantuan dana usaha melalui pengisian data diri, namun ternyata itu adalah situs web palsu. Dengan hal ini, diperlukan adanya perlindungan digital untuk anggota komunitas Panglima di Pondok Pesantren Sunan Plumbon dari ancaman serangan *phishing*.

Penelitian ini bertujuan merancang sebuah aplikasi deteksi *phishing* berbasis web yang dapat membantu anggota komunitas dalam mengidentifikasi dan menghindari serangan *phishing*. Dengan menggunakan algoritma *Decision Tree*. Diharap aplikasi ini nantinya mampu memberi perlindungan tambahan dan meningkatkan kesadaran akan resiko *phishing* di kalangan anggota komunitas Panglima di Pondok Pesantren Sunan Plumbon.

Penelitian ini akan menggunakan pendekatan waterfall dengan implementasi pertama-tama, melakukan pengumpulan dataset fitur-fitur URL *phishing* dan non-*phishing* dari sumber yang terpercaya. Selanjutnya Perancangan model deteksi menggunakan *Decision Tree* dengan menggunakan Google Colab sebagai platform pengembang. Pembuatan aplikasi web menggunakan Flask sebagai framwork perancangan. Proses *deployment* aplikasi akan menggunakan layanan *hosting* Heroku setelah tahap pengujian selesai. Implementasi dan pengujian aplikasi menggunakan metode *blackbox* untuk memastikan fungsionalitas yang tepat. Melalui penelitian ini, diharap dapat menciptakan lingkungan online yang lebih aman dan terpercaya bagi anggota komunitas dalam menjalankan aktivitas bisnis mereka.

1.2 Rumusan Masalah

Pada perancangan sistem deteksi *phishing* menggunakan algoritma *decision tree* berbasis web, terdapat permasalahan yang perlu diidentifikasi dan diselesaikan. Adapun rumusan masalah yang akan dijawab dalam penelitian ini yaitu bagaimana merancang sistem deteksi *phishing* berbasis website menggunakan algoritma *decision tree*, sehingga dapat membantu anggota komunitas Panglima di Pondok Pesantren Sunan Plumbon yang dapat menghindari serangan *phishing*.

1.3 Batasan Masalah

Dalam rangka menjaga fokus dan ruang lingkup penelitian, batasan masalah pada penelitian ini mencakup hal-hal berikut:

1. Penelitian ini terbatas pada penggunaan dataset yang diperoleh dari repositori publik.
2. Aplikasi hanya akan berbasis website.
3. Proses perancangan aplikasi akan menggunakan metode *waterfall*.
4. Bahasa yang digunakan dalam perancangan adalah bahasa pemrograman HTML dan CSS untuk *frontend*, sedangkan *backend* menggunakan Python
5. Perancangan aplikasi menggunakan algoritma *decision tree*
6. Aplikasi ini akan difokuskan hanya mendeteksi URL.
7. Output aplikasi berupa hasil prediksi (*phishing* atau aman).
8. Heroku akan digunakan sebagai platform *hosting*
9. Pengujian aplikasi dilakukan dengan *blackbox*.

10. Penelitian difokuskan pada anggota komunitas Panglima yang berjualan di lingkungan Pondok Pesantren Sunan Plumbon.

1.4 Tujuan Penelitian

Tujuan utama dari penelitian ini untuk menghasilkan sebuah aplikasi deteksi *phishing* berbasis web menggunakan algoritma *decision tree* untuk memberi perlindungan tambahan kepada anggota komunitas Panglima di Pondok Pesantren Sunan Plumbon dalam menghadapi resiko serangan *phishing*, serta meningkatkan kesadaran akan pentingnya keamanan digital.

1.5 Signfikasi Penelitian

Penelitian ini memiliki beberapa signifikansi yang dapat memberikan kontribusi positif dalam bidang keamanan siber dan perlindungan pengguna daring:

1. Membantu anggota komunitas Panglima dalam menghindari serangan *phishing*
2. Meningkatkan kesadaran akan pentingnya keamanan data pribadi.
3. Mengurangi resiko pencurian informasi pribadi dan finansial.
4. Memanfaatkan algoritma *decision tree*, diharapkan dapat melakukan deteksi *phishing*.
5. Hasil penelitian dapat menjadi acuan dan sumber referensi untuk penelitian-penelitian selanjutnya.
6. Memberikan edukasi tentang penggunaan teknologi dan keamanan siber kepada komunitas Panglima di Pondok Pesantren Sunan Plumbon.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam tinjauan pustaka ini, akan dibahas beberapa penelitian terdahulu yang relevan dan dapat menjadi acuan dalam Perancangan aplikasi deteksi *phishing*.

Tabel 2.1 Penelitian Terdahulu

No	Judul	Peneliti dan Tahun Penelitian	Metode	Dataset	Hasil Penelitian
1.	Implementasi Metode Adasyn Dalam Deteksi Url Berbahaya Menggunakan *Machine Learning*: Demi Meningkatkan Keamanan Siber di Era Digital	Gilang Dwi Setyawan, Andrie Yuswanto, et al. (2023)	*Random Forest, Decision Tree, Naive Bayes*	Terdiri dari 2 fitur, yaitu "url" dan "type". Selanjutnya, diperluas menjadi 11 fitur. Nilai fitur mencakup "phishing", "benign", dan "defacement"	Pada hasil experiment menunjukkan bahwa algoritma *Random Forest mendapat akurasi 99%, Decision Tree 98%, dan Naive Bayes 96%.* [7]
2.	Deteksi Website	Vikky Aprelia	*Random Forest,*	Website UCI Machine	Naïve bayes memiliki

No	Judul	Peneliti dan Tahun Penelitian	Metode	Dataset	Hasil Penelitian
	Phishing Menggunakan Teknik Filter Pada Model *Machine Learning*	Windarni, Anggit Ferdita Nugraha, et al. (2023)	*Decision Tree, Naive Bayes*	Learning (11055 data dengan 30 fitur)	nilai akurasi sebesar 60,4%, decision tree 94,4% dan random forest sebesar 96,3%.[8]
3.	Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing	Nabila Bianca Putri dan Arie Wahyu Wijayanto (2022)	*Random Forest, Decision Tree, Naive Bayes, Support Vector Machine*	Data website: 1.353 data dengan 702 data situs non-phishing, 103 data situs mencurigakan, dan 548 data situs phishing	*Random Forest* memiliki akurasi 90,77%, Naïve Bayes 82,31%, Decision Tree 85,77%, dan Support Vector Machine 86,25%[9]
4.	*Comparative Analysis of Phishing Website*	Muhammad Fandru Al Rifqi, Mauli	*Logistic Regression, Decission Tree,*	Source EKG Kaggle	*Logistic Regression* memiliki perfoma

No	Judul	Peneliti dan Tahun Penelitian	Metode	Dataset	Hasil Penelitian
	Prediction Classification Algorithm Using Logistic Regression, Decision Tree, and Random Forest	Dina, et al. (2022)	*Random Forest*		akurasi 92.76%, *Decission Tree* 94.57%, dan *Random Forest* 97.10% [10]
5.	Komparasi Machine Learning Memprediksi Phising Dalam Keamanan Website	Aswan Supriyadi Sunge (2022)	*Decision Tree, Naïve Bayes, Multilayer Perceptron (Neural Network), K-Nearest Neighbor, Support Vector Machines*	Data publik dengan jumlah 11.055 data, 30 atribut.	Seluruh metode yang diusulkan memiliki akurasi tinggi. Namun, *Neural Network* lebih tinggi untuk mendeteksi phishing [11]
6.	Identifikasi Website Phishing dengan	Agung Susilo Yuda Irawan,	*Support Vector Machine, Decision*	Data public dengan tiga kategori yaitu legitimate,	Seluruh algoritma yang diusulkan

No	Judul	Peneliti dan Tahun Penelitian	Metode	Dataset	Hasil Penelitian
	Perbandingan Algoritma Klasifikasi	Nono Heryana, et al. (2021)	*Tree, Random Forest, dan Multilayer Perceptron*	suspicious, dan phishing	memiliki Perfoma yang baik. Namun, *Multilayer Perceptron* mendapat nilai tertinggi dengan akurasi 93,15% dan AUC 0,976[12]
7.	Rekomendasi Jurusan Dengan Menggunakan Decision Tree Pada Sistem Penerimaan Peserta Didik Baru SMK Widya Dharma Turen	Lazuardi Noorca Rachmadi, Aji Prasetya Wibawa, Utomo Pujianto (2021)	Decision Tree dengan mengguna kan waterfall sebagai metode pengemban gan.	Nilai raport calon peserta didik baru yang masih berupa numeric.	Menggunaka n pohon keputusan dapat membantu proses penerimaan peserta didik baru pada SMK Widya Dharma Turen.

Dari tinjauan penelitian terdahulu, dapat dilihat bahwa pendekatan deteksi *phishing* menggunakan algoritma *decision tree* telah menunjukkan hasil yang baik dan stabil. Karena itu, penelitian ini bertujuan untuk memberikan kontribusi tambahan dengan mengembangkan aplikasi deteksi *phishing* berbasis web menggunakan metode *waterfall* yang diharap mampu mempermudah pengguna dalam mendeteksi *phishing*.

2.2 *Phishing*

Phishing, sebuah istilah yang berasal dari kata dalam bahasa Inggris, yang secara harfiah dapat diterjemahkan sebagai penangkapan ikan atau memancing, merujuk pada praktik penipuan daring yang bertujuan untuk memperoleh informasi sensitif dari korban tanpa sepengetahuan mereka. Kejahatan dunia maya akan terus berkembang dalam berbagai bentuk, memberikan peluang yang lebih besar bagi pelakunya [13]. Modus operandinya melibatkan upaya untuk meniru entitas resmi atau organisasi terpercaya dalam komunikasi elektronik dengan tujuan menipu pengguna untuk mengungkapkan informasi pribadi seperti nama pengguna, kata sandi, atau rincian kartu kredit [14].

Praktik *phishing* sering kali terjadi dalam konteks layanan perbankan daring, di mana pelaku menyamar sebagai lembaga keuangan yang sah untuk mengelabui pengguna. Misalnya, pelaku dapat menciptakan formulir *login* palsu yang menyerupai tampilan resmi situs web bank. Ketika pengguna memasukkan kredensial mereka ke dalam formulir tersebut, informasi tersebut akan diserahkan kepada pelaku tanpa sepengetahuan pengguna. Namun, *phishing* tidak terbatas

pada sektor perbankan saja, ia juga menyerang melalui media sosial, email, pesan teks, dan situs web.

Phising dapat dibagi menjadi berbagai jenis berdasarkan motivasi pelaku dan sasaran yang ingin dicapainya. [14]

A. *Spear Phishing*, mengacu pada upaya *phising* yang ditujukan secara spesifik kepada target tertentu. Dalam jenis ini, pelaku memiliki peluang keberhasilan yang lebih tinggi karena sasaran mereka lebih jelas.
B. *Whaling*, mirip dengan *Spear Phishing*, namun targetnya adalah individu dengan posisi tinggi dalam organisasi, seperti pejabat atau eksekutif perusahaan. Pelaku *phishing* dalam jenis ini menggunakan media seperti subpoena untuk menakuti korban agar menghadap ke pengadilan.
C. *Clone Phishing*, merupakan jenis *phishing* konvensional di mana pelaku menggunakan email resmi untuk mengirimkan pesan yang identik dengan email asli kepada korban, tetapi dengan mengganti lampiran pesan tersebut.
D. *Covert Redirect*, merupakan teknik *phising* yang sangat halus di mana pelaku mengarahkan korban ke link yang seolah-olah resmi, tetapi sebenarnya menuju link yang dibuat oleh pelaku melalui pop-up login. Dalam teknik ini, target sulit untuk dikenali karena pelaku menggunakan link dan situs resmi dengan pop-up yang dimodifikasi, sehingga korban sulit membedakan antara form login asli dan palsu.

2.1 *Uniform Resource Locator* (URL)

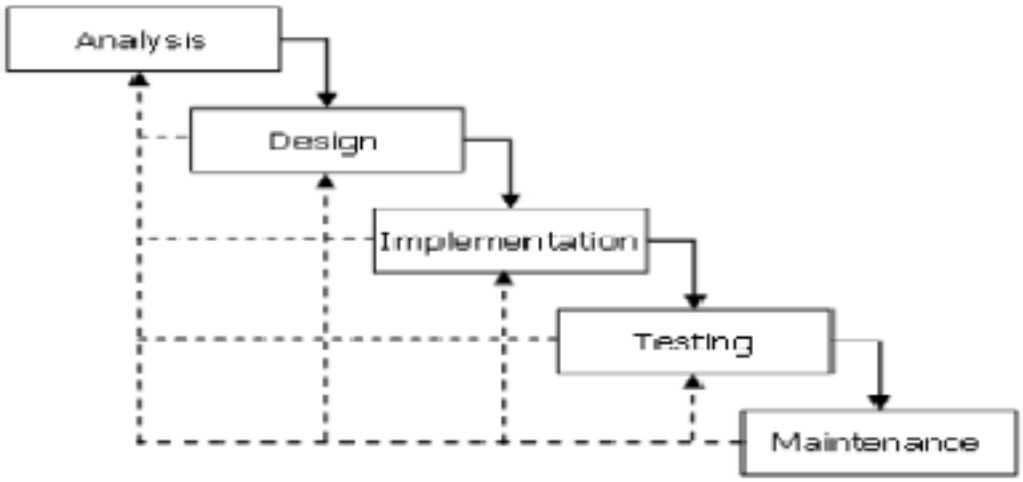
Uniform Resource Locator (URL) adalah pengidentifikasi unik yang terdiri dari protokol dan nama domain yang digunakan untuk menemukan halaman atau sumber daya di Internet. Namun, URL dapat dieksploitasi oleh penyerang untuk membuat URL berbahaya dan merugikan pengguna. URL berbahaya ini dirancang untuk melakukan serangan seperti spam, *phishing, malware*, dan *defacement*. Mereka dapat muncul di email, pesan teks, pop-up, atau iklan palsu dan dapat membahayakan komputer atau jaringan pengguna. Oleh karena itu, penting untuk mengembangkan sistem yang dapat mendeteksi URL berbahaya tersebut dan melindungi pengguna dari serangan dunia maya [15].

2.2 Website

Situs web merupakan sebuah koleksi halaman yang tergabung dalam satu domain, menyediakan beragam informasi yang dapat diakses oleh pengguna internet melalui mesin pencari. Konten yang disajikan dalam situs web mencakup gambar, ilustrasi, video, serta teks yang dirancang untuk memenuhi berbagai keperluan pengguna. Situs web menjadi sarana bagi pengguna untuk menjelajahi informasi, mengakses berita, mencari referensi, atau bahkan berkomunikasi secara daring. Melalui situs web, pengguna dapat memperoleh pengetahuan, mengeksplorasi produk dan layanan, atau bahkan terlibat dalam komunitas daring. Dengan demikian, situs web menjadi jendela virtual yang membuka akses ke berbagai jenis informasi dan interaksi di era digital [16].

2.3 Metode *Waterfall*

Siklus Hidup Pengembangan Perangkat Lunak atau *Software development* life cycle (SDLC) adalah metode yang digunakan dalam industri untuk merencanakan, membangun dan memelihara sistem informasi. Salah satu model SDLC yang paling umum adalah model Waterfall, yang terdiri dari lima tahapan yang harus diselesaikan secara berurutan. Model *Waterfall* adalah proses pengembangan perangkat lunak yang berurutan di mana kemajuan dipandang mengalir ke bawah seperti air terjun melalui tahapan-tahapan yang harus diselesaikan secara berurutan. Model ini pertama kali diusulkan oleh Winston W. Royce pada tahun 1970 untuk menggambarkan praktik perangkat lunak. Seluruh tahapan model *Waterfall* harus diselesaikan sebelum melanjutkan ke tahap berikutnya dan dapat diulangi hingga tahap tersebut selesai sepenuhnya. Secara umum model *Waterfall* terdiri dari analisis, desain, implementasi, pengujian dan pemeliharaan. Penjelasan mengenai metode *Waterfall* dapat dilihat pada Gambar 2.1 Metode Waterfall [17].



Gambar 2.1 Metode Waterfall [17]

Model SDLC Waterfall terdiri dari lima fase:

a) *Analysis*: Menentukan kebutuhan perangkat lunak secara lengkap dan komprehensif, melibatkan analis untuk mendefinisikan kebutuhan fungsional dan non-fungsional.
b) *Design*: Merencanakan solusi perangkat lunak, termasuk desain algoritma, arsitektur perangkat lunak, dan antarmuka pengguna.
c) *Implementation*: Merealisasikan kebutuhan bisnis dan desain menjadi program yang dapat dieksekusi melalui pemrograman dan deployment.
d) *Testing*: Memverifikasi dan memvalidasi perangkat lunak untuk memastikan sesuai dengan spesifikasi dan tujuan, serta melakukan debugging.
e) *Maintenance*: Memodifikasi perangkat lunak setelah pengiriman untuk memperbaiki kesalahan, meningkatkan kinerja, dan menyesuaikan dengan kebutuhan baru.

Setiap fase dalam model *Waterfall* harus diselesaikan sebelum melanjutkan ke fase berikutnya, dan dapat diulang hingga sempurna.

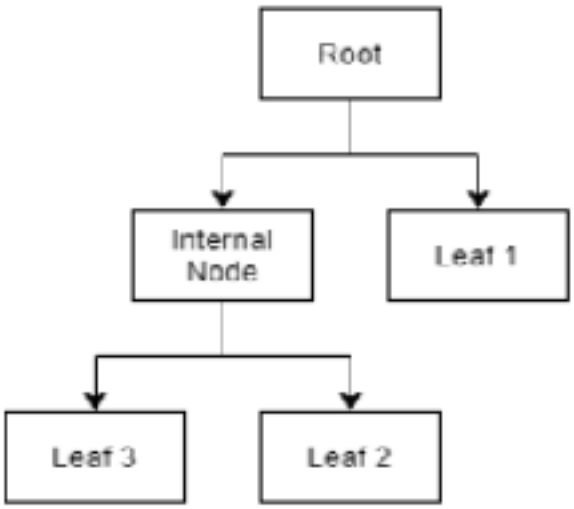
2.4 Data Sekunder

Data sekunder mengacu pada data yang sudah ada dan tersedia untuk dapat digunakan oleh peneliti lain. Dengan kata lain, data sekunder adalah data historis yang dikumpulkan pada masa lalu untuk tujuan tertentu. Misalnya, peneliti mengumpulkan data dan kemudian membagikannya kepada orang lain untuk digunakan dalam penelitian mereka sendiri. Data sekunder dapat berasal dari

berbagai sumber. Penting untuk dicatat bahwa data yang semula digunakan sebagai data primer dalam suatu penelitian, dapat menjadi data sekunder dalam penelitian lain jika data tersebut digunakan kembali oleh peneliti lain. Hal ini terjadi ketika data yang sama digunakan untuk tujuan penelitian yang berbeda [18].

2.5 Algoritma *Decision Tree*

Decision tree diperkenalkan pada tahun 1960-an oleh Fredkin. Singkatnya, *Decision tree* merupakan metode yang khusus dalam proses pengklasifikasian pada Text Mining [19]. *Decision tree* adalah sebuah struktur pohon, dimana setiap node internal (non-leaf) merepresentasikan pengujian atribut, setiap cabang merupakan suatu pembagian hasil uji, dan node daun (leaf) merepresentasikan kelompok kelas tertentu [9]. Model pohon keputusan banyak digunakan untuk pendeteksian web phishing [11]. Algoritma ini memiliki fungsi untuk mengeksplorasi data serta menemukan hubungan dari beberapa data yang ada di dalam dataset [8]. Metode pohon keputusan mengubah data menjadi bentuk pohon keputusan, kemudian mengubah pohon tersebut menjadi bentuk aturan, dan menyederhanakan aturan-aturan tersebut. Hasil pengujian ditetapkan pada setiap node internal, dimana setiap node internal mengembalikan hasil pengujian pada atribut [10].



Gambar 2.2 Decision Tree [9]

Root diambil dari dataset pada atribut yang paling berpengaruh atau memiliki kemampuan terbaik untuk memisahkan data menjadi kelompok yang berbeda. Pemilihan atribut untuk menjadi root biasanya dilakukan berdasarkan kriteria yang menghasilkan pemisahan yang paling baik antara kelas atau label data. Sedangkan internal root merupakan node internal (non-leaf) yang berada di bawah root. Node ini merepresentasikan pengujian atribut yang mengarahkan ke node-node cabang. Leaf 1, Leaf 2, Leaf 3 merupakan node daun (leaf) dalam struktur pohon keputusan. Node daun ini merupakan hasil dari pengujian atribut pada node internal, yang menunjukkan kelas atau label yang diprediksi untuk sampel data tertentu.

Algoritma decision tree sering digunakan untuk deteksi phishing karena bisa mengubah masalah kualitatif menjadi numerik. Data biasanya diubah menjadi nilai -1 dan 1, yang mewakili ketiadaan atau kehadiran fitur tertentu. Decision tree menggunakan struktur pohon biner untuk menganalisis data ini. Setiap langkah dalam pohon membagi data berdasarkan fitur-fitur, dan proses ini terus berlanjut

hingga prediksi akhir tercapai. Dengan cara ini, *decision tree* membantu mengidentifikasi pola dalam data *phishing* dengan mudah dan menghasilkan prediksi yang tepat.

2.6 SQLite

SQLite adalah sebuah library yang menerapkan mesin database self-contained, serverless, zero-configuration, dan transactional. Self-contained berarti SQLite dapat beroperasi tanpa memerlukan dukungan tambahan dari library eksternal atau sistem operasi. Serverless artinya akses ke database, baik itu untuk membaca maupun menulis, dapat dilakukan langsung dari file database tanpa perlu melalui proses server, dan tidak mendukung akses remote. Zero-configuration berarti SQLite dapat digunakan tanpa perlu proses instalasi sebelumnya. Transactional berarti SQLite adalah database transaksional yang menerapkan prinsip Atomic, Consistent, Isolated, dan Durable (ACID). Aplikasi database yang menggunakan SQL server memiliki kekurangan seperti memerlukan instalasi sebelum penggunaan, memerlukan server untuk memproses file database, dan membutuhkan ukuran memori yang lebih besar dibandingkan dengan penggunaan SQLite [20].

2.7 Python

Pencipta dari Python adalah Guido Van Rossum di Belanda pada tahun1990 dan dinamai berdasarkan acara televisi favoritnya "*Monty Python's Flying Circus*".

Pada awalnya Perancangan Python merupakan sebuah hobi, namun Python kemudian menjadi Bahasa pemrograman yang banyak digunakan karena kesederhanaannya, singkat, sintaksnya yang intuitif, dan Pustaka yang luas. Secara umum, Python mendukung pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Kelebihan Python adalah [21]:

a) Banyak koleksi perpustakaan dengan modul siap pakai untuk berbagai keperluan.
b) Struktur bahasanya jelas, sederhana dan mudah dipelajari.
c) Objek.
d) Sistem memori otomatis.
e) Bersifat modular.

2.8 *HyperText Markup Language* (HTML)

HTML (*Hyper Text Markup Language*) adalah bahasa yang digunakan untuk mengatur tata letak halaman web. Saat membuat website, HTML sangatlah penting karena merupakan fondasi dari website. File HTML disimpan dengan ekstensi .html dan dapat dibuka dengan browser web. HTML adalah dasar untuk membuat halaman web, di mana dapat mengatur elemen halaman dan membuat judul [22].

2.9 *Cascading Style Sheets* (CSS)

Cascading Style Sheets atau CSS, adalah bahasa pemrograman yang digunakan untuk mendesain halaman web. Dalam proses desain web, CSS digunakan untuk menambahkan gaya dan tampilan pada elemen halaman menggunakan tag seperti ID dan kelas. CSS direkomendasikan oleh World Wide Web Consortium (W3C) pada tahun 1996, dan browser seperti Internet Explorer dan Netscape telah merilis versi terbaru yang mendukung standar CSS. Terdapat tiga versi utama CSS: CSS1, CSS2, dan CSS3. Selain digunakan pada HTML dan XHTML, CSS juga dapat digunakan untuk mendesain tampilan aplikasi Android. CSS memungkinkan untuk mengonfigurasi dan memilih tampilan elemen dalam dokumen web [22].

2.10 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk menjelaskan dan menggambar rencana serta desain sistem secara visual. Dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson, UML memberikan cara yang jelas untuk berkomunikasi antara pengguna dan pengembang, serta antara tim pengembangan perangkat lunak. UML memungkinkan untuk melihat struktur dan interaksi sistem dengan lebih mudah, dan memastikan keselarasan antara ide desain dan pelaksanaannya. Meskipun UML tidak memberikan langkah-langkah spesifik untuk analisis dan desain [23]. Berikut notasi dan artifak penting dalam UML:

A. *Use Case* Diagram

Diagram *use case* (UCD) menggambarkan apa yang dilakukan sistem yang sedang dibangun dan siapa yang berinteraksi dengan sistem. UCD bertindak sebagai dokumen kontrak antara pelanggan, pengguna, dan pengembang. Pengguna menggunakan UCD untuk memahami sistem dan memastikan bahwa sistem benar-benar menyelesaikan masalah yang telah atau sedang terjadi. Pengembang menggunakan UCD sebagai acuan tepat dalam pengembangan sistem. Secara umum diagram use case terdiri dari elemen-elemen seperti aktor, use case, dependensi, generalisasi, dan hubungan. UCD ini memberikan pandangan statis dari sistem yang sedang dibangun dan merupakan artefak dari proses analisis [23]. Notasi pada use case dapat dilihat pada tabel 2.2 Notasi Diagram *Use Case*.

Tabel 2.2 Notasi Diagram *Use Case* [24]

No.	Nama	Notasi	Keterangan
1.	*Actor*		Segala hal yang berinteraksi dengan sistem aplikasi komputer
2.	*Ekstends*		Relasi untuk menambahkan fitur tambahan walau tanpa usecase utama
3.	*Include*		Relasi untuh menambahkan fitur tambah dan hanya dapat dijalankan ketika usecase utama terpenuhi.
4.	*Use Case*		Tindakan aktor dalam sistem untuk mencapai tujuan tertentu
5.	*Association*		Relasi yang menghubungkan satu objek dengan objek lainnya
6.	*System*		Menampilkan atau menggunakan data paket untuk menyajikan

No.	Nama	Notasi	Keterangan
			informasi atau fitur tertentu kepada pengguna.

B. Activity Diagram

Diagram aktivity disebut juga diagram fungsional merupakan representasi visual dari aliran kerja atau aktivitas dalam suatu sistem atau bisnis. Diagram ini sering digunakan untuk berbagai tujuan, termasuk desain proses bisnis [24]. Notasi pada activity diagram dapat dilihat pada tabel 2.3 Notasi *Activity Diagram.*

Tabel 2.3 Notasi *Activity* Diagram [24]

No.	Nama	Notasi	Keterangan
1.	*Actifity*		Menunjukkan interaksi antar kelas antarmuka
2.	*Action*		Kondisi sistem saat melakukan aksi tertentu
3.	*Initial Node*		Menandai awal pembentukan atau proses suatu objek
4.	*Actifity Final Node*		Menunjukkan akhir dari pembentukan objek.
5.	*Fork Node*		Aliran sistemyang pada tahp tertentu bercabang menjadi beberapa aliran.

C. Sequence Diagram

Sequence diagram menunjukkan urutan proses sistem untuk mencapai tujuan tertentu, seperti interaksi antar kelas, aktivitas yang terlibat, urutan aktivitas, dan informasi yang diperlukan. Membuat diagram sequence sangat penting dalam perancangan karena memandu program dan menunjukkan aliran kendali program. Desain urutan biasanya terdiri dari elemen objek,