PENGEMBANGAN APLIKASI DETEKSI *PHISING* BERBASIS *WEB* MENGGUNAKAN ALGORITMA *DECISION TREE*

TUGAS AKHIR

Diajukan Oleh:

MUHAMMAD RAHUL
NIM. 190705028
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
BANDA ACEH
2023/1444 H

PENGEMBANGAN APLIKASI DETEKSI *PHISING* BERBASIS WEB MENGGUNAKAN ALGORITMA *DECISION TREE*

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
dalam Ilmu/Prodi Teknologi Informasi

Oleh:
MUHAMMAD RAHUL
190705028
Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui Untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Khairan AR, M.Kom	**Malahayati, MT**
NIDN. 2004078602	**NIDN. 2027018303**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Ima Dwitawati, M.B.A
NIDN. 0113108204

PENGEMBANGAN APLIKASI DETEKSI *PHISING* BERBASIS *WEB* MENGGUNAKAN ALGORITMA *DECISION TREE*

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir/Skripsi
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Ilmu/Prodi Teknologi Informasi

Pada Hari/Tanggal: Selasa, 11 April 2023
20 Ramadhan, 1444 H

di Darussalam, Banda Aceh

Panitian Ujian Munaqasah Tugas Akhir/Skripsi:

Ketua,	Sekretaris,
Khairan AR, M.Kom	**Malahayati, MT**
NIDN. 2004078602	**NIDN. 2027018303**
Penguji I,	Penguji II,
Mulkan Fadhli, M.T.	**Rahmat Musfikar, M.Kom**
NIDN. 1328118801	**NIDN. 2013098901**

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,

Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIDN. 0002106203

ii

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Muhammad Rahul
NIM	:	190705028
Program Studi	:	Teknologi Informasi
Fakultas	:	Sains dan Teknologi
Judul	:	Pengembangan Aplikasi Deteksi *Phising* Berbasis *Web* Menggunakan Algoritma *Decision Tree*

Dengan ini menyatakan bahwa dalam penulisan tugas akhir/skripsi ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggung jawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggung jawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggung jawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 11 April 2023
Yang Menyatakan

Muhammad Rahul

ABSTRAK

Nama	:	Muhammad Rahul
NIM	:	190705028
Program Studi	:	Teknologi Informasi
Judul	:	Pengembangan Aplikasi Deteksi *Phising* Berbasis *Web* Menggunakan Algoritma *Decision Tree*
Tanggal Sidang	:	
Jumlah Halaman	:	78
Pembimbing I	:	Khairan AR, M.Kom
Pembimbing II	:	Malahayati, MT
Kata Kunci	:	Aplikasi Deteksi, *Phising*, *Decision Tree*, *Waterfall*, *Deteksi Phising*

Era Pandemi *Covid-19* yang melanda masyarakat di seluruh dunia memunculkan adanya perubahan pada kebiasaan, gaya hidup, dan tingkah laku manusia. Kebiasaan yang biasanya dilakukan luring (luar jaringan) berubah menjadi daring (dalam jaringan), pada kebiasaan baru itu maka manusia akan sering menggunakan internet, penggunaan internet tidak jauh dari menggunakan *website*. Salah satu ancaman saat menggunakan website yaitu pengguna dapat terkena penyerangan yang dapat merugikan *user* seperti *phising*, untuk menghindari *phising* ada beberapa aplikasi berbasis *website* yang menyediakan fitur deteksi *phising* tapi karena kurangnya keakuratan deteksi sehingga dapat membuat masalah pada pengguna.Maka penulis mencoba menggunakan data *decision tree* yang sudah teruji keakuratannya dan mendapatkan nilai kebenaran yang tinggi yang diambil dari *platform github*. Pengempangan aplikasi yang dilakukan pada penelitian ini bertujuan untuk menambah keakuratan deteksi *phising* demi keamanan pengguna. Pengembangan berfokus pada penerapan algoritma *decision tree* pada fitur deteksi *phising* dan pengembangan aplikasinya menggunakan metode *waterfall*. Pada fitur deteksi nantinya menggunakan *url* untuk mendeteksi *phising*.

Kata kunci: Aplikasi Deteksi, *Phising*, *Decision Tree*, *Waterfall*, Deteksi *Phising*

ABSTRACT

Name	:	Muhammad Rahul
Student ID	:	190705028
Department	:	*Information Technology*
Title	:	*Web-Based Phishing Detection Application Development Using Decision Tree Algorithm*
Date	:	
Thesis Pages	:	78
Supervisor I	:	Khairan AR, M.Kom
Supervisor II	:	Malahayati, MT
Keywords	:	*Detection Application, Phishing, Decision Tree, Waterfall, Phishing detection*

The era of the Covid-19 Pandemic that hit people all over the world gave rise to changes in human habits, lifestyles and behavior. Habits that are usually carried out enticingly turn into daring, in these new habits people will often use the internet, using the internet is not far from using the website. One of the threats when using the website is that the user can be exposed to attacks that can harm the user. To avoid phishing, there are several website-based applications that provide phishing detection features, but because of their accuracy, they can cause problems for the user. So the authors try to use a proven data decision tree. its accuracy and get high marks from github platform. The application embankment carried out in this study aims to increase the security of phishing detection for user safety. Development focuses on applying the decision tree algorithm to the phishing detection feature and developing its application using the waterfall method. The worry feature will use URLs to detect phishing.

Keywords: Detection Application, Phishing, Decision Tree, Waterfall, Phishing detection

KATA PENGANTAR

بِسْمِ اللهِ الرَّحْمَنِ الرَّحِيْمِ

Segala puji hanya milik Allah, Tuhan seluruh alam yang tiada Tuhan selain-Nya. Shalawat dan salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad Shallallahu Alaihi Wassalaam, keluarga dan sahabatnya. Alhamdulillah dengan rahmat Allah yang Maha Rahman dan yang Maha Rahim, sehingga penulis dapat menyelesaikan skripsi dengan judul "Pengembangan Aplikasi Deteksi *Phising* Berbasis *Web* Menggunakan Algoritma *Decision Tree*" ini. Skripsi ini merupakan salah satu syarat untuk menyelesaikan program studi Strata satu Teknologi Informasi pada Fakultas Sains dan Teknologi di Universitas Islam Negeri Ar-Raniry Banda Aceh.

Ucapan terima kasih penulis sampaikan kepada berbagai pihak yang menjadi sebab dari mereka penulis belajar, mendapatkan ilmu, mendapatkan dukungan, serta mendapatkan hal yang bermanfaat lainnya sehingga penulis sampai pada titik menyelesaikan skripsi ini. Terutama dalam konteks ini penulis sampaikan kepada :

- Kedua orang tua penulis yang selalu memberikan dukungan, dan doa yang tak ternilai harganya selama ini. Hanya Allah yang mampu membalas kasih sayang mereka yang tak terhingga, semoga selalu Allah limpahkan rahmat kepada mereka dan mendapatkan ridha serta cinta dari-Nya.
- Bapak Khairan AR, M.Kom. dan Ibu Malahayati, MT. selaku pembimbing yang selalu berrsedia meluangkan waktu, fikiran untuk membimbing penulis demi kesempurnaan skripsi ini. Kemudian juga kepada Bapak penguji1 dan Bapak penguji2 selaku penguji sidang proposal skripsi yang telah memberi koreksi dan saran untuk mencapai tujuan yang sama. Semoga Allah limpahkan rahmat kepada Bapak dan Ibu dan mendapatkan ridha serta cinta dari- Nya.
- Ketua dan Sekretaris Program Studi Teknologi Informasi, Ibu Ima Dwitawati, M.B.A. dan Bapak Khairan Ar, M.Kom, serta Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan dalam bidang

Teknologi Inofrmasi kepada penulis sehingga penulis mampu menyelesaikan skripsi ini.

- Pembimbing Akademik Bapak Ghufran Ibnu Yasa yang telah membimbing dan memberikan saran selama masa perkuliahan.
- Staf Prodi Ibu Cut Ida Rahmadiana S,Si. yang telah membantu membantu penulis dalam hal pengurusan administrasi dan surat-surat untuk keperluan penyelesaian skripsi.
- Orang-orang terdekat saya yang selalu mendukung saya yang tidak dapat saya sebutkan Namanya satu persatu.

Penulis berharap hasil skripsi ini dapat berguna bagi keamanan dalam bersosial media dengan adanya aplikasi deteksi *phising* yang dikembangkan. Dan dapat dijadikan sebagai referensi bagi penelitian selanjutnya ataupun sebagai referensi untuk mempelajari phsing di berbagai media.

Akhir kata, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan,oleh karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan demi mendapatkan hasil yang lebih baik lagi. Semoga perjalanan mempelajari dan berkarya pada salah satu ilmu milik-Nya ini dapat menghantarkan penulis agar dapat mengenal-Nya dan kekasih-Nya lebih banyak serta mendapatkan ridho dan cinta-Nya yang Maha Rahman dan Rahim. Shalawat dan salam kepada junjungan kita Nabi Muhammad Shallallahu Alaihi Wassalaam, keluarga dan sahabat-sahabatnya.

Banda Aceh, 11 Maret 2023

Penulis

Muhammad Rahul

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang

Era Pandemic Covid-19 yang melanda masyarakat di seluruh dunia memunculkan adanya perubahan pada kebiasaan, gaya hidup, dan tingkah laku manusia. Kebiasaan yang sebelumnya biasa dilakukan secara fisik dan luring (luar jaringan), perlahan mulai ditinggalkan dan berganti dengan kebiasaan baru yang dilakukan secara daring (dalam jaringan) melalui berbagai platform digital. Adanya internet juga turut berperan sebagai media pendukung yang tidak hanya memudahkan melainkan juga telah berevolusi menjadi media yang menyediakan berbagai fasilitas dan dukungan dalam beraktivitas secara *online*. Salah satu perubahan yang nampak dari adanya evolusi internet adalah halaman *website*. Pada awal kemunculannya, *website* hanya digunakan sebagai media untuk penyampaian informasi satu arah, akan tetapi, seiring dengan adanya perkembangan teknologi informasi dan komunikasi yang terjadi sekarang ini, membuat *website* kini mampu untuk menjadi sebuah media interaksi, media komunikasi, dan bahkan bisa juga digunakan sebagai media dan sarana untuk melakukan berbagai proses transaksi. Sayangnya, adanya perubahan yang terjadi pada *website*, justru diimbangi dengan semakin merebaknya potensi ancaman yang membahayakan dan dapat menimbulkan kerugian bagi penggunanya. Satu dari sekian banyak ancaman siber yang mengancam pengguna *website* adalah serangan *Web Phising* (Nugraha et al., 2022).

Pada kuartal ketiga tahun 2021 dilaporkan oleh Anti-*Phishing Working Group* (APWG) bahwa Serangan *phishing* masih menimbulkan risiko besar bagi pengguna Internet di seluruh dunia, dengan lebih dari 700.000 situs *web* unik dan lebih dari 45.000.000 klik pada tautan *phishing* terdeteksi oleh Kaspersky pada Q3 2021 (Drury et al., 2023). Oleh sebab itu, dengan semakin banyaknya serangan *phising* yang terjadi, perlu adanya sebuah sistem yang dapat membantu pengguna website dalam mendeteksi adanya serangan *phising* yang dapat menimbulkan berbagai kerugian yang diakibatkan dari adanya serangan *phising* tersebut.

Mengingat pentingnya menghadapi ancaman serangan *phising*, pengembangan system yang dapat mendeteksi adanya web *phising* masih terus dilakukan oleh peneliti di

berbagai sektor keilmuan, seperti pada bidang *data mining* dan *machine learning*. Banyak penelitian tentang deteksi *phising* yang telah dilakukan oleh peneliti terdahulu tapi masih sangat sedikit penelitian yang mengembangkan Algoritma *machine learning* menjadi sebuah aplikasi deteksi *phising* berbasis *website,* seperti pada penelitian sebelumnya dengan judul Deteksi *Website Phishing* menggunakan Algoritma *Machine Learning* yaitu *Decision Trees, random forests* dan *vector machine algorithms*. Hasil dari pengembangan ini adalah tingkat akurasi yang dilakukan pada pengembangan ini mencapai akurasi deteksi 97,14% menggunakan algoritma *random forest* dengan tingkat positif palsu terendah yang berbeda tipis dengan algoritma *Decision Tree* mendapatkan akurasi sebesar 97.12%. Juga hasil menunjukkan bahwa pengklasifikasi ini memberikan kinerja yang sangat baik ketika menggunakan lebih banyak data sebagai data pelatihan nya (Mahajan & Siddavatam, 2018).

Pada pengembangan yang berjudul Deteksi Situs *Web Phishing* Menggunakan algoritma *decision tree.* Pada pengembangan ini mereka melakukan pengembangan yaitu pengaruh *decision tree* terhadap deteksi web *phising* menggunakan 2 model yaitu *Phishing Websites Dataset* (PWD) dan PWD2. Pada pengembangan ini penulis mengahasilkan kemampuan klasifikasi pada *decision tree* yang sangat baik dan stabilitas yang tinggi tidak hanya pada PWD, tetapi juga pada PWD2 (YANG et al., 2018).

Kemudian pada penelitian Sistem Deteksi Kecanduan Pornografi Berbasis Chatbot Menggunakan Pornography Addiction Screening Tool (PAST), penelitian tersebut menggunakan metode waterfall yang dijelaskan bahwa metode tersebut sudah teruji secara best practice sebagai metode pengembangan perangkat lunak (Muhammad & Ardimansyah, 2022).

Jika didasarkan pada penelitian sebelumnya maka di dapatkan hasil yang sangat baik dan stabil pada algoritma *decision tree,* maka penulis ingin mengembangkan algoritma tersebut menjadi sebuah aplikasi *website* menggunakan metode *waterfall* yang sangat cocok digunakan pada pengembangan aplikasi seperti yang di jelaskan pada penelitian terdahulu yang kemudian diharapkan dapat mempermudah pengguna dalam mendeteksi *phising* sebagai keamanan data mereka.

1.2. Rumusan Masalah

Dari latar belakang maka dapat di simpulkan rumusan masalah yaitu :

1. Bagaimana membangun aplikasi deteksi *phising* berbasis *website*?
2. Bagaimana menggunakan algoritma *decision tree* sebagai deteksi *phising* berbasis web?

1.3. Tujuan

Setelah mendapatkan rumusan masalah maka tujuan pada pengembangan ini yaitu :

1. Merancang sebuah aplikasi deteksi *phising* berbasis website.
2. Menerapkan algoritma decision tree pada website.

1.4. Manfaat

Dari penjelasan tujuan di atas maka di dapatkan manfaat pada pengembangan ini yaitu:

1. Bagi masyarakat nantinya dapat menggunakan aplikasi pendeteksi phishing ini untuk memeriksa apakah link yang akan diakses phishing atau bukan.
2. Bagi penulis, dapat menambah pengetahuan dan sebagai referensi mengenai deteksi phishing menggunakan decision tree.

1.5. Batasan Masalah

Supaya penelitian ini tidak melebar terlalu jauh maka Batasan masalah dibatasi pada :

1. Aplikasi dibangun berbasis website.
2. Bahasa yang akan penulis gunakan dalam pengembangan aplikasi deteksi phishing adalah bahasa pemrograman front end yaitu css dan html, sedangkan untuk back end menggunakan phyton,PHP dan Laravel.

3. Hanya menguji menggunakan url.
4. Output dari aplikasi deteksi phishing nantinya berupa hasil prediksi phishing.

BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian yang relevan dibutuhkan sebagai referensi pendukung dalam proses penelitian, tujuannya memberikan wawasan dan pengetahuan untuk menyelesaikan masalah. Maka dengan penelitian terkait ini dapat menjadi acuan referensi terkait dengan penelitian yang sedang penulis lakukan.

Berdasarkan penelitian oleh (Mahajan & Siddavatam, 2018) yang berjudul "pendekatan deteksi *phising* menggunakan *machine learning"*, pada penelitian ini mengasilkan akurasi dari algoritma *random forest* yang mendapatkan nilai tertinggi yaitu 98.80% dan *decision tree* sendiri mendapatkan nilai 97.11%. Sedangkan dalam penelitian yang berjudul "Pendekatan Phishing Deteksi Menggunakan *Machine Learning*" oleh (Deepak Pathak & Sandhia, 2022), penelitian tersebut juga melakukan perbandingan nilai dari hasil data setiap algoritma machine learning yang menghasilkan akurasi tertinggi 97.11% yang di dapatkan oleh algoritma *decision tre*e.

Setelah itu pada penelitian yang berjudul "Deteksi Situs Web Phishing Menggunakan Decision Tree", penelitian ini dilakukan pengujian terhadap kemampuan klasifikasi menggunakan 2 model dataset yaitu *Phishing Websites Dataset* (PWD) dan PWD2. Dataset tersebut dibuat menggunakan beberapa algoritma dari machine learning seperti decision tree,random *forest dan Support Vector Machine* (SVM) yang mendapatkan hasil sangat baik dan stabil untuk digunakan pada deteksi *phising* (YANG et al., 2018).

Kemudian pada pengembangan aplikasi deteksi *phising* ini penulis ingin mengembangkan menggunakan metode waterfall, pada penelitian yang dilakukan oleh (Muhammad & Ardimansyah, 2022) dijelaskan bahwa metode waterfall sudah teruji secara best practice sebagai metode pengembangan perangkat lunak. Metode waterfall mudah digunakan karena metode tersebut merupakan metode pengembangan perangkat lunak yang sistematik dan sekuensial, sehingga pada penelitiannya yang berjudul "Sistem Deteksi Kecanduan *Pornografi* Berbasis *Chatbot* Menggunakan *Pornography Addiction Screening Tool* (PAST)" mereka mengunakan metode

waterfall sebagai acuan dalam pengembangan *system* deteksi kecanduan *pornografi* berbasis *chatbot*.

Phishing merupakan sebuah kejahatan *cyber* yang bertujuannya adalah menjebak orang lain. Biasanya phishing memakai alamat *url* yang kemudian mengarah pada sebuah *website* palsu dengan tujuan menjebak korban. Ph ishing sangat merugikan baik dari *privacy*, *financial* dan eksploitasi data (Irawan et al., 2021). Sedangkan dalam jurnal yang berjudul Teknik Penyerangan *Phising* Pada *Social Engineering* Menggunakan data Set Dan Pencegahannya, *phishing* merupakan sebuah serangan dengan cara menjadi orang yang bisa dipercayai dengan diwakilkan oleh orang tertentu dengan tujuan mendapatkan informasi berharga. *Phishing* biasanya dilakukan pada tahap awal serangan untuk mendapatkan kepercayaan korban (Ahmadian & Sabri, 2021).

2.2. Deteksi

Deteksi merupakan hal yang ingin diketahui oleh manusia dalam sesuatu hal tertentu sehingga dilakukan pembuatan alat yang bisa membantu memecahkan masalah manusia. Dengan adanya sebuah alat deteksi ini maka bisa mendapatkan berbagai informasi yang menjadi penyebab pada kerusakan tersebut (Syahrizal & Haryati, 2018). Deteksi ini juga dapat dikatakan sebuah proses untuk memeriksa sesuatu dengan cara tertentu yang berguna untuk membantu manusia umenyelesaikan sebuah masalah yang di hadapinnya.

2.3. Metode *Waterfall*

Metode waterfall merupakan metode yang sangat terstruktur pada setiap langkah pengembangan yang dimiliki. Pengerjaan metode waterfall ini intinya dalah pengerjaannyka dilakukan berurutan atau linear. Jadi jika langkah yang pertama belum dilakukan, maka langkah kedua juga tidak bisa dilakukan. Jika langkah kedua juga belum dilakukan maka langkah ketiga pun tidak bisa dilakukan, begitupun seterusnya. Dapat disimpulkan pada langkah ketiga bisa dilakukan apabila langkah kesatu dan kedua sudah dilakukan (Yagoyamu, 2020).

a. *Analyst*

Pengembang mengumpulkan kebutuhan-kebutuhan pada tahap ini. seperti kebutuhan fungsional dan juga non-fungsional untuk pengembangan aplikasi.

b. *Design*

Pada tahap ini, pengembang melakukan pengembangan desain aplikasi dan pemodelan untuk user story yang sedang berlangsung dalam suatu iterasi. Desain aplikasi dan pemodelan yang dibuat hanya bertujuan untuk memenuhi requirements terhadap user story yang sedang dikerjakan tanpa mencoba menebak apa yang akan dibutuhkan di masa depan.

c. *Implementation*

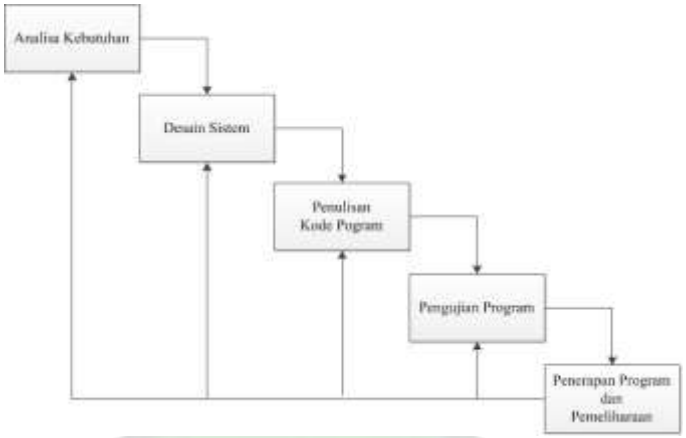
Fase ini merupakan waktu dimana kode pemrograman dibangun. Pengembang mengimplementasikan objek-objek yang terdapat pada fase design. Proses ini merupakan tahapan secara *real time* dalam merancang suatu aplikasi. Maka penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah menyelesaikan pengkodean maka akan dilakukan *testing* terhadap aplikasi yang telah dirancang. Tujuan *testing* yaitu menemukan kesalahan-kesalahan terhadap aplikasi tersebut dan kemudian bisa diperbaiki.

d. *System Testing*

System testing merupakan tahap akhir yang kemampuan dan keefektifannya di uji sehingga diketahui kekurangan dan kelemahan dari aplikasi tersebut dan setelah itu dilakukan pengkajian ulang serta perbaikan terhadap aplikasi untuk menjadi lebih baik dan sempurna.

e. *Program implementation and maintenance*

Berikut pada Gambar 2.1 dapat dilihat contoh tahapan proses metode *waterfall* yang terdiri dari analisis,desain,penulisan kode program,pengujian dan penerapan program.



Gambar 2.1 Tahapan Proses *waterfall* (Yagoyamu, 2020).

Software yang telah disampaikan kepada *client* pasti akan mengalami perubahan. Perubahan tersebut bisa mengalami kesalahan karena *software* harus beradaptasi dengan zaman (periperal atau sistem operasi baru), atau karena *client* membutuhkan perkembangan fungsional yang sama, sehingga pengerjaan pada iterasi berikutnya dapat dilakukan dengan lebih efektif. Tahap ini dapat selesai setelah melakukan perilisan aplikasi. Selanjutnya, iterasi atau iterasi berikutnya dapat dimulai kembali dari tahap*iteration inialization* atau proses pengembangan aplikasi dapat berakhir jika semua*requirements* telah terpenuhi tanpa adanya cacat yang tersisa.

2.4. Website

Website merupakan situs informasi yang dapat di gunakan banyak orang dengan cepat. *Website* ada karena perkembangan zaman dari bidang teknologi komputer. *Website* sudah menjadi sebuah media penyebaran informasi bagi perusahaan, organisasi, dan sekolah (Rahardja et al., 2018).

Pada pembuatan *website* biasanya digunakan bahasa pemograman *ccs* dan *html*, dijelaskan dalam buku Pemrograman Web : *Html Dan Css* bahwa *Hyper Text Markup Languange* (HTML) adalah bahasa pemograman yang digunakan sebagai perancangan halaman *website*. Dalam dunia pemrograman berbasis *website*, HTML disebut sebagai pondasi dasar halaman *website*. Sebuah file HTML disimpan dengan ekstensi .html

(dot html). File itu kemudian dapat di akses menggunakan *web browser*, sedangkan *css* adalah bahasa pemograman yang digunakan sebagai *web design.* Dalam men*design website*, *css* menggunakan sebuah penanda yang biasanya kita kenal dengan *id* dan juga *class* (Herho, 2018).

Penulis juga menggunakan *REST API* sebagai penghubung algoritma *decision tree* dengan aplikasi yang ingin dibangun. *REST API* merupakan sebuah tautan terhadap *client* dan *server* yang mana permohonan *client* yang kemudian ditanggapi oleh *server* dibangun sesuai proses *transfer* sumber daya, sebuah *REST client* mengirim permintaan melalui *HTTP Request* dan REST *server* kemudian akan merespon melalui *HTTP Response.* (Permana et al., 2019).

2.5. *Phyton*

Python merupakan suatu bahasa pemrograman *open source multiplatform* yang bisa dipakai kepada berbagai sistem operasi seperti MacOS, Windows, dan Linux. Tidak hanya itu, Bahasa pemrogaman ini bersifat fleksibel dan gampang untuk dipelajari.

Pemograman yang ditulis pada *python* biasanya lebih mudah dibaca dan lebih ringkas dibandingkan dengan bahasa C. *Python* memiliki modul standar yang menyediakan sejumlah besar algoritma dan fungsi, untuk menyelesaikan pekerjaan seperti mengurai data teks dan mengunduh data dari *web server*. Dengan menggunakan Bahasa pemograman *python*, Programmer dengan mudah menerapkan teknik komputasi tingkat lanjut, seperti pemrograman berorientasi objek (Herho, 2017).

2.6. *PHP* dan *Laravel*

Dijelaskan pada penelitian yang berjudul Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website (Endra et al., 2021) website dapat dikembangkan dengan bahasa pemrograman dinamis, salah satunya adalah bahasa pemrograman PHP (Hypertext Preprocessor) yang merupakan bahasa pemrograman open-source server side. Server Side adalah script yang dimasukkan untuk diproses oleh dan diproses di server dan PHP memiliki keunggulan bersifat open-source, yaitu pengguna bebas memodifikasi dan mengembangkan

aplikasi atau sistem sesuai keinginan. Pada penelitian ini dijelaskan juga Laravel merupakan framework yang dikembangkan oleh Taylor Otwell pada bulan Juni 2011 yang memiliki banyak Terakreditasi pengguna hingga saat ini. Pada framework Laravel terdapat fungsi-fungsi kode yang disediakan di library kemudian di install ke dalam Laravel.

2.7. *Visual Studio Code*

Visual Code Studio (VCS) merupakan *software* dengan tujuan membuat aplikasi. Selain VCS, perangkat lunak android studio juga digunakan untuk emulator sebagai testing program setelah dibuild. (Hamzan et al., 2022).

Sedangkan dalam skripsi (Suriyani, 2020) menjelaskan VSC adalah teks editor kode sumber yang cukup ringan tapi juga kuat disaat berjalan pada desktop dan tersedia pada beberapa sistem operasi seperti macOS, Windows, dan Linux.

2.8. Algoritma *Decision Tree*

Berikut pada Gambar 2.2 dapat dilihat perhitungan algoritma decision tree untuk mencari class apakah ke “virginica” atau “versicolor”.

Gambar 2.2 Contoh Perhitungan Algoritma

Decision tree merupakan salah satu algoritma yang cukup populer untuk proses klasifikasi *data mining*. Prinsip kerja algoritma *decision tree* adalah menggolongkan sebuah objek ke dalam kelas/label yang sudah tersedia (Handayani, 2020).

Terdapat tiga langkah yang menjadi dasar dari algoritma decision tree dalam permasalahan untuk deteksi *phising*

1. Persoalan deteksi *phising* ini adalah sebuah kualitatif yang di translasi ke dalam numerik
2. Datanya menjadi -1 dan 1
3. Decision tree menggunakan binary tress dalam menganalisa dan memproses data

Algoritme *decision tree* dimulai dari simpul akar dan membuat pilihan "terbaik" secara rekursif, setelah itu contoh yang tersisa dibagi menjadi simpul anak hingga tidak

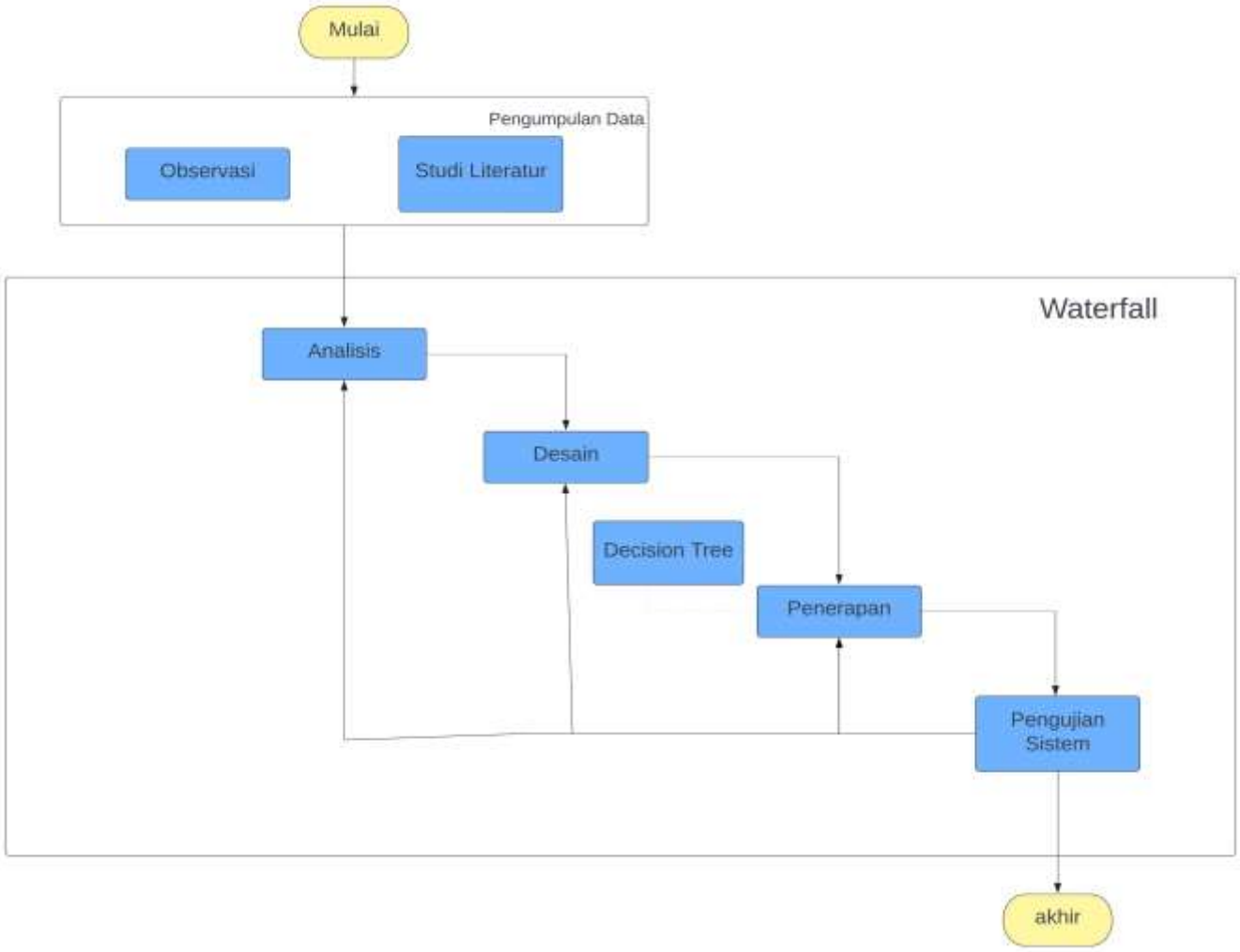
ada lagi fitur untuk dipilih atau simpul tersebut murni.

Pemilihan algoritma *decision tree* karena algoritma ini baik digunakan untuk pengembangan deteksi *phising* dan memiliki nilai yang stabil pada deteksi phising dibandingkan dengan algoritma lain seperti algoritma *logistic regression*. Pada penelitian (Irawan et al., 2021) yang berjudul "Identifikasi *Website Phishing* dengan Perbandingan Algoritma Klasifikasi" dijelaskan bahwa tingkat *error* pada *Decision Tree* lebih kecil dibandingkan algoritme SVM, sehingga penulis menggunakan algoritma decision tree untuk mendapatkan hasil yang baik dan minim terjadi error saat di masukan ke dalam pemograman.

BAB III
METODE PENELITIAN

3.1. Tahapan Penelitian

Penelitian ini dikerjakan dengan beberapa tahapan yang diadaptasi berdasarkan langkah dalam melakukan pengembangan perangkat lunak yang mengacu pada sebuah metode *decision tree.* Pemilihan metode ini dikarenakan metode decision tree bersifat *fase one by one*. Berikut tahapan penelitian yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Pada subbab berikut dijelaskan terkait diagram alur penelitian yang digambarkan pada Gambar 3.1. penulis menggunakan metode ini karena metode ini baik digunakan pada penelitian berupa perancangan dan pengembangan.

3.2. Metode Pengumpulan Data

3.2.1. Observasi

Pada tahapan ini, penulis akan melakukan pengamatan terhadap aplikasi deteksi *phising* dari hasil penelitian terdahulu dan pada website deteksi *phising* yang ada pada google saat penelitian ini dilakukan. Apakah aplikasi tersebut memberikan akurasi deteksi *phising* yang baik.

3.2.2. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan berbagai data atau informasi yang terkait dengan rancang bangun deteksi *phising*. Data atau informasi tersebut diperoleh dari berbagai sumber seperti buku, penelitian terdahulu, jurnal, artikel dan literatur lainnya.

3.3. Metode Pengembangan Aplikasi

Metode decision tree dalam prosesnya terdiri dari beberapa tahapan. Tahapan metode Waterfall tersebut bersifat fase one by one. Berikut rincian tahap-tahap pengembangan aplikasi dengan metode waterfall.

3.3.1. Analisis

Pada Tahapan pertama dalam metode waterfall adalah menentukan kebutuhan pengguna akan software yang akan dikembangkan seperti kebutuhan fungsional dan non-fungsional.

1. **Kebutuhan Fungsional**

Kebutuhan fungsional merupakan fitur-fitur yang disediakan oleh *software* kemudian bisa diakses oleh *user* secara langsung melalui *interface* yang disediakan oleh aplikasi atau system (Wijaya et al., 2018).

Dibawah ini terdapat beberapa kebutuhan fungsional pada aplikasi deteksi *phising* berbasis *website* yang dapat pengguna lihat secara langsung melalui *interface*.

a. Halaman *regristrasi* agar *user* bisa mendaftar
b. Halaman *login* agar *user* bisa masuk pada aplikasi.
c. Halaman pendeteksi *phising*.
d. Halaman hasil deteksi *phising*.

2. **Kebutuhan Non-Fungsional**

Kebutuhan non fungsional merupakan batasan dari fitur yang berikan oleh aplikasi. Batasan kebutuhan tersebut termasuk software dan hardware yang digunakan dalam pengembangan aplikasi (Sidauruk et al., 2020).

software yang dipakai oleh penulis pada pengembangan aplikasi deteksi *phising* pada pengembangan ini yaitu sebagai berikut :

a. ***Visual Studio Code***
b. ***Phyton,css dan html***
c. ***PHP dan Laravel***
d. ***Mysql***
e. ***Git & Github***
f. ***Figma***
g. ***Lucidchart***
h. ***Google Chrome***

hardware yang dipakai oleh penulis pada perancangan aplikasi deteksi *phising* ini yaitu sebagai berikut :

a. Laptop Asus A409F (2019)

b. Window 10 Home

c. RAM 8 GB

3.3.2. Desain

Pada Tahapan ini adalah Informasi tentang detail kebutuhan dari tahap analyst yang di analisa, kemudian di implementasikan pada design pengembangan. Pengembangan desain ini dilakukan untuk tujuan memberikan gambaran tentang apa yang akan dilakukan.

1. Rancangan Sistem

Metode yang penulis gunakan pada pengembangan *logic aplication* deteksi *phising* adalah *Unified Model Language* (UML). Model UML yang akan digunakan oleh penulis adalah *use case diagram*, *activity diagram*, *Squence Diagram* dan *class Diagram*.

a) Use Case Diagram

Use case diagaram menu login dapat dilihat pada gambar 3.2.

Gambar 3. 2 Use Case Diagram Deteksi Phising

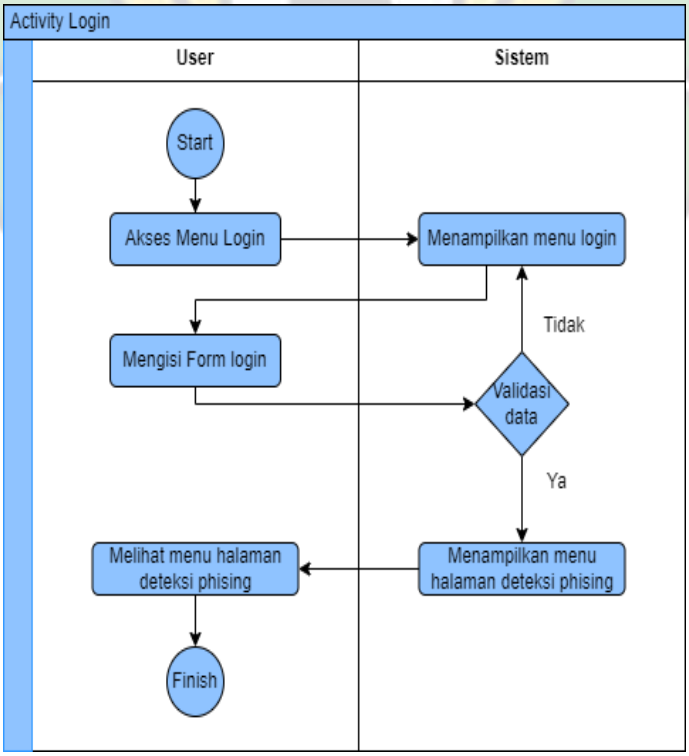
Ketika user menjalankan aplikasi untuk pertama kalinya, maka user harus melakukan login terlebih dahulu,jika user belum memiliki akun maka user harus melakukan registrasi, kemudian user baru dapat melakukan deteksi phising pada halaman menu utama. Setelah melakukan deteksi phising user dapat melihat hasil deteksi phising pada menu hasil.

b) Activity Diagram

Activity diagram dirancang berdasarkan *use case* yang telah dirancang sebelumnya untuk pengembangan aplikasi deteksi *phising* berbasis *website*. *Activity diagram* terdapat 3 bagian, yang pertama adalah *activity diagram* menu registrasi, *activity diagram* menu *login* dan *activity diagram* menu deteksi *phising.*

a. *Activity diagaram* menu *login*

Use case diagaram menu login dapat dilihat pada gambar 3.3.

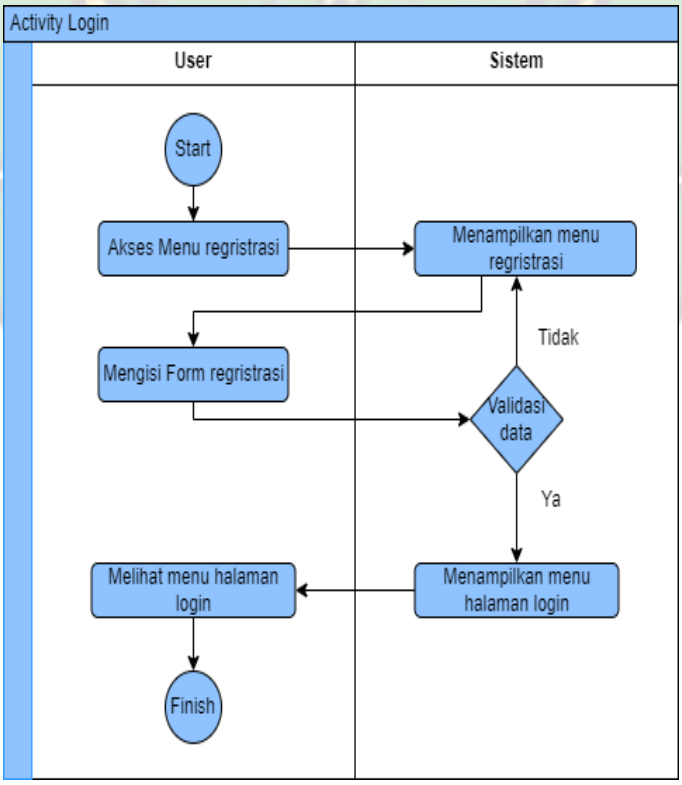


Gambar 3.3 activity menu login

Saat user menjalankan aplikasi untuk pertama kalinya, *user* akan masuk ke dalam menu *login*. Pada menu login user memasukan gmail dan password. Kemudian aplikasi melakukan pemeriksaan validasi data. Jika *user* memasukan *email* dan *password* salah maka aplikasi segera melakukan permintaan pada user agar memasukan kembali gmail dan password dengan betul. Jika *user* memasukan gmail dan password yang benar maka user akan masuk pada menu deteksi *phising* dan dapat menggunakan aplikasi.

b. *Activity diagaram* menu regristrasi

Activity Diagram Menu regristasi dapat dilihat pada gambar 3.6.

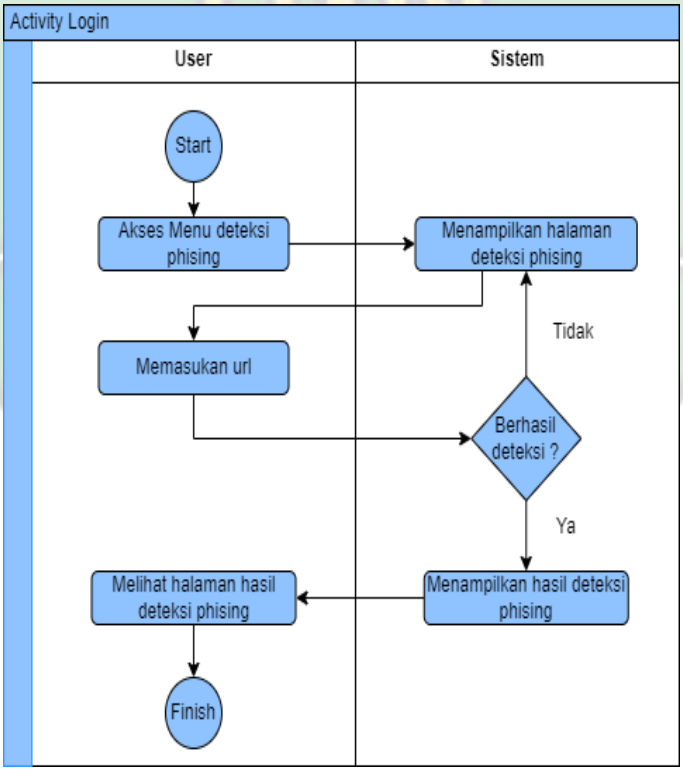


Gambar 3.4 *activity menu* regristasi

Saat user menjalankan aplikasi untuk pertama kalinya, user akan masuk ke dalam menu login. Pada menu login user memasukan gmail dan password. Kemudian aplikasi melakukan pemeriksaan validasi data. Jika user memasukan gmail dan password salah maka aplikasi segera melakukan permintaan pada user agar memasukan kembali gmail dan password dengan betul. Jika user memasukan gmail dan password yang benar maka user akan masuk pada menu deteksi phising dan dapat menggunakan aplikasi.

c. *Activity diagaram* menu deteksi *phising*

Activity Diagram Menu deteksi *phising* dapat dilihat pada gambar 3.7.



Gambar 3.5 *activity menu* regristasi

Saat user menjalankan aplikasi untuk pertama kalinya, user akan masuk ke dalam menu login. Pada menu login user memasukan

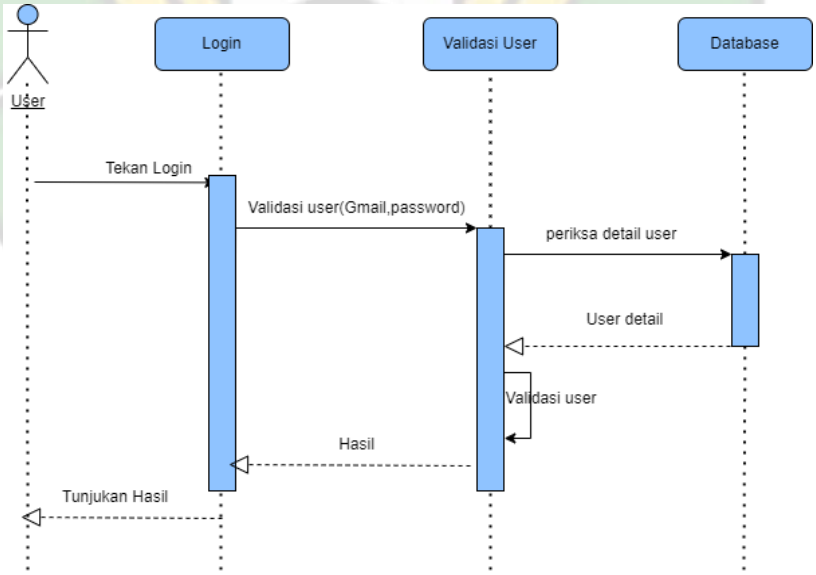
gmail dan password. Kemudian aplikasi melakukan pemeriksaan validasi data. Jika user memasukan gmail dan password salah maka aplikasi segera melakukan permintaan pada user agar memasukan kembali gmail dan password dengan betul. Jika user memasukan gmail dan password yang benar maka user akan masuk pada menu deteksi *phising* dan dapat menggunakan aplikasi.

c) Squence Diagram

Ketika sudah merancang *use case diagaram* dan *activity diagram* penulis akan merancang *Squence diagram*, tujuan dari pebuatan *sequence diagram* ini adalah menjelaskan interaksi objek**.**

a. *Squence diagram* menu *login*

Squence Diagram Menu Login dapat dilihat pada gambar 3.8.



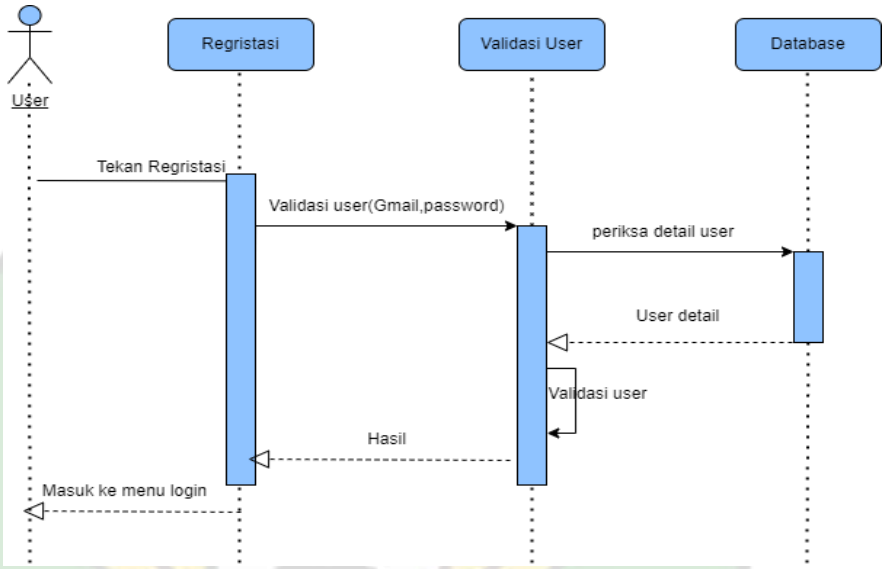
Gambar 3.6 Squence menu login

Ketika *user* ingin melakukan *login user* akan mengisi *gmail* dan *password* setelah itu aplikasi akan periksa detail *user* kemudian

mevalidasi user apakah dapat melakukan *login* atau tidak.

b. *Squence diagaram* menu regristrasi

Squence Diagram Menu Regristrasi dapat dilihat pada gambar 3.9.

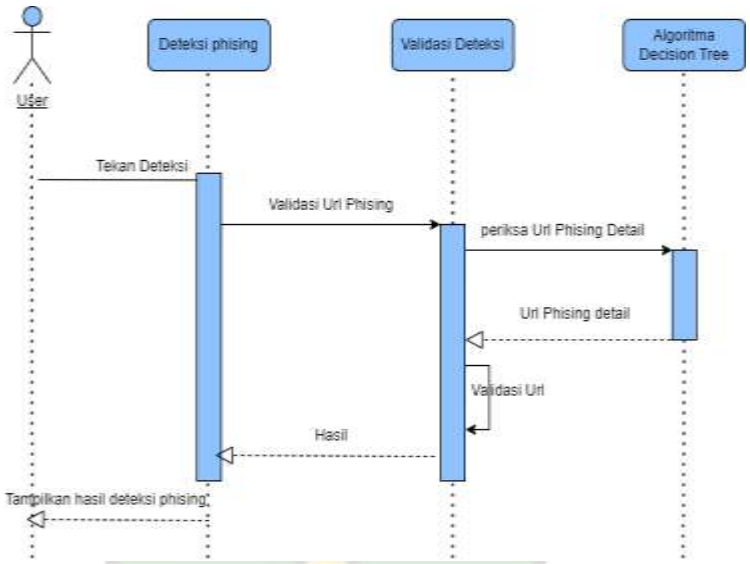


Gambar 3.7 *Squence menu* regristasi

User harus melakukan regristasi terlebih dahulu jika tidak memiliki akun, Ketika melakukan regristasi *user* harus memasukan nama, *gmail* dan *password* kemudian sistem akan periksa detail *user* untuk memvalidasi data user, jika benar *user* akan masuk pada menu *login.*

c. *Squence diagram* menu deteksi *phising*

Squence Diagram Menu Regristrasi dapat dilihat pada gambar 3.10.



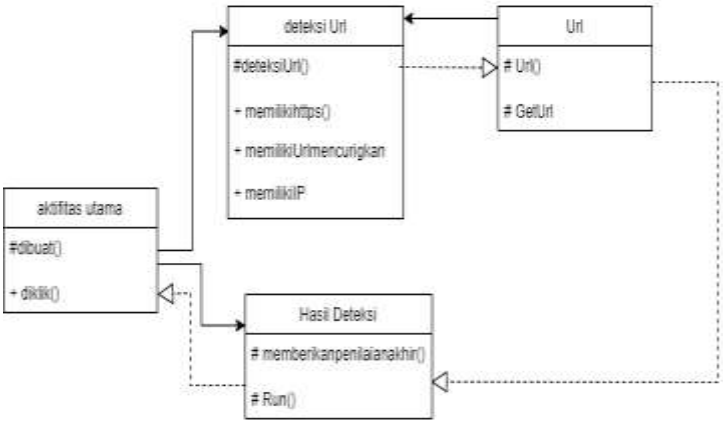
Gambar 3.8 *Squence menu* deteksi *phising*

Ketika *user* ingin melakukan deteksi *phising* user harus memasukan url, kemudian sistem akan memeriksa url tersebut menggunakan algoritma *decision tree*, setelah itu aplikasi akan melakukan validasi url dan menampikan hasil deteksi *phising* kepada user.

d) Class Diagram

Class Diagram menggambarkan hubungan antar objek-objek dan struktur yang terdapat pada aplikasi. Struktur tersebut melingkupi atribut-atribut dan metode-metode yang terdapat pada masing-masing *class*. Adapun gambaran *class diagram* dari pengembangan aplikasi deteksi *phising* berbasis *website* dapat dilihat pada gambar 3.12:

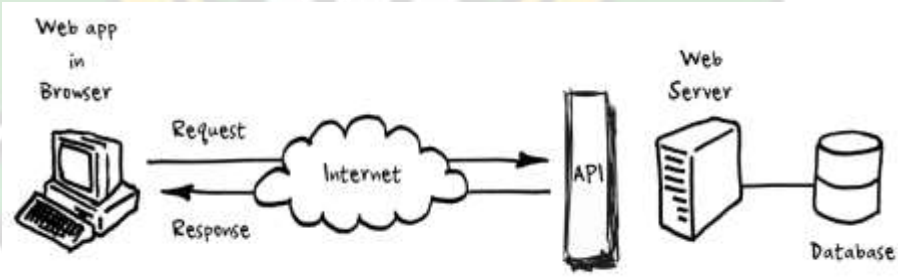
.



Gambar 3.9 *Class diagram* deteksi *phising*

2. Arsitektur Aplikasi

Pada Tahapan arsitektur aplikasi di tetapkan keputusan pengembangan aplikasi yang siap dikembangkan. Berikut pada gambar 3.2 adalah perancangan aplikasi yang siap untuk di kembangkan.



Gambar 3.10 Arsitektur Aplikasi

User melalui browser menggunakan localhost untuk dapat terhubung ke web server di saat user memakai aplikasi, server memuat interface setelah itu mengambil data yang dibutuhkan dari database. Melalui interface user bisa mengolah data sesuai yang di inginkan.

3. Rancangan User Interface

Perancangan interface merupakan proses komunikasi terhadap pengguna dengan aplikasi. Perancangan interface terdiri dari perancangan

struktur menu dan perancangan tampilan pada tampilan user (Rita Irviani., 2018).

a. *Activity Login*

Pengembangan *Activity login* terdapat tampilkan dua edit text dan dua button. Perancangan tampilan *activity login* bisa dilihat pada gambar 3.3 di bawah ini.

Gambar 3.11 Activity Login

b. *Activity Registrasi*

Activity registrasi dibangun dengan menampilkan 4 edit text yaitu nama, *email* dan *password*. 2 button untuk buat akun dan masuk, yang dapat dilihat pada gambar 3.4.

Gambar 3.12 Activity registrasi

c. *Activity Menu Utama*

Activity menu utama dirancang dengan 1 edit text yang berguna untuk menempelkan url yang ingin di deteksi. Untuk lebih jelas dapat dilihat pada gambar 3.5.

Gambar 3.13 Activity Menu Deteksi *Phising*

d. *Acivity menu hasil*

Halaman ini merupakan halaman hasil dari deteksi yang di lakukan. Pada halaman ini terdapat 3 edit teks yaitu link yang ingin

kita deteksi,hasil dan keterangan hasil deteksi aman atau tidak.untuk desain gambarnya dapat dilihat pada gambar 3.6.

Gambar 3.14 Activity menu hasil

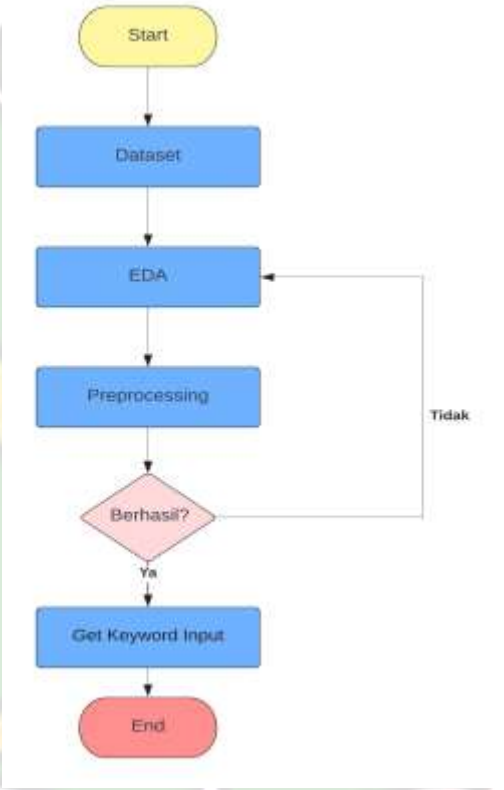
3.3.3. Penerapan

Pada Penerapan pengembang merubah desain menjadi sebuah aplikasi agar fungsi perangkat lunak bisa berjalan. Untuk merubah design menjadi aplikasi maka pengembang menggunakan visual studio code dengan pemrograman css dan html pada front end kemudian untuk back end nya menggunakan phyton,php dan laravel. Pengembangan aplikasi dikerjakan dari awal hingga aplikasi siap digunakan.

3.3.4. Algoritma *Decision Tree*

Metode rekomendasi atau algortima yang digunakan untuk fitur deteksi *phising* adalah algoritma decision tree. Algoritma decision tree digunakan karena terbukti lebih akurat dan tepat dalam menentukan class untuk menentukan link tersebut akan masuk ke class aman atau tidak aman, kemudian di jelaskan pada penelitian Data Mining Identifikasi Website *Phising* Menggunakan Algoritma decision tree penggunaan Decision Tree pada penelitian tersebut karena hasilnya sesuai dan memuaskan (Giap Yo Ceng, 2018).

Penulis menggunakan algoritma decision tree ini dalam bentuk package yang telah dibuat fungsinya dan siap untuk diimplementasikan, sehingga penulis dapat berfokus menulis code pengembangan aplikasi tanpa harus menghabiskan waktu untuk membuat fungsi algoritma decision tree sendiri. Package yang digunakan adalah decision-tree-python yang ditulis dalam bahasa pemrograman pyhton oleh Eligijus112 dan dapat diakses pada repository akun Github miliknya (Eligijus112., 2021).



Gambar 3.15 Diagram Alur Algoritma deteksi *phising*

Gambar 3.7 dijelaskan langkah-langkah yang akan diterapkan dalam pembuatan algoritma deteksi *phising*. Adapun rincian yang menjelaskan langkah-langkah yang terdapat pada Gambar 3.7 sebagai berikut.

1. ***Dataset***

Tahapan pertama pada pengembangan ini adalah menyiapkan dataset untuk deteksi *phising* yang sudah tersedia di github, data set dibuat dari kumpulan data situs web phising yang tersedia untuk umum di repositori pembelajaran mesin yang disediakan oleh UCI. UCI adalah website yang berisi arsip dengan koleksi data set yang dibuat sejak tahun 1987 oleh David Aha dan para mahasiswa pascasarjana UB Irvine. Pada github sudah disediakan dalam bentuk file data.csv yang berbentuk 1,-1 dan 0 sebagai penambah.

2. ***Exploratory Data Analisys (EDA)***

Tahapan kedua adalah Exploratory Data Analisys (EDA) dilakukan sebuah investigasi awal terhadap dataset untuk menemukan dan memahami distribusi, pola, anomali, memeriksa asumsi, serta hubungan variable.

3. ***Preprocessing***

Pada tahap ini dilakukan proses pengujian data set yang bertujuan untuk memeriksa keberhasilan data set yang ingin di gunakan, setelah berhasil dilakukan pengujian maka data set dapat digunakan pada aplikasi yang ingin di kembangkan. Pada 2000 data pengujian yang dilakukan oleh Eligijus112 mendapatkan hasil akurasi deteksi *phising* 90.5% menggunakan algoritma *decision tree.*

4. ***Get Keyword Input***

Tahapan terakhir mendapatkan input kata kunci pencarian yang dimasukkan pada parameter URL (*Uniform Resource Locator*) ketika melakukan *request* API.

3.3.5. Sistem Pengujian

Sistem pengujian adalah pengujian fungsionalitas kesuluruhan fitur hasil dari pelaksanaan yang telah terintegrasi. Sistem pengujian termasuk kategori teknik pengujian *black box.* Pengujian *software* menggunakan *black box* di uji dari aspek spesifikasi fungsional tanpa menguji design dan kode program supaya dengan mudah diketahui apakah fungsi, masukan dan keluaran dari software sesuai dengan spesifikasi yang di perlukan (Cholifah et al., 2018). Sistem pengujian bertujuan untuk menguji semua fitur yang telah diimplementasikan pada suatu iterasi dari sudut pandang pengguna secara end-to-end (Ujung ke Ujung).

BAB IV
HASIL DAN PEMBAHASAN

4.1. Implementasi

Hasil dari tahap implementasi dituangkan pada poin ini beserta rinciannya. Secara garis besar, proses implementasi dimulai dengan pembuatan API (Antarmuka pemrograman aplikasi) disisi *backend* menggunakan bahasa pemrograman Go dan melakukan pengujian terhadap API yang dibuat menggunakan Postman sebagai salah satu alat penguji atau simulasi API. Kemudian dilanjutkan disisi *frontend* dengan pembuatan tampilan aplikasi dan melakukan integrasi API pada aplikasi menggunakan bahasa pemrograman Dart dan Flutter. Adapun hasil implementasi pada pengembangan aplikasi kamus bahasa Aceh berbasis *mobile* dapat dilihat pada subbab berikut beserta rinciannya dalam bentuk tampilan aplikasi dan dokumentasi API. Dokumentasi API berfungsi untuk mengetahui bagaimana API dapat digunakan atau diterapkan pada aplikasi disisi *client*, hal ini mencakup cara mengirim *request* dan mendapatkan *response* yang dikembalikan dari *server*.

4.1.1. *Fitur Authentication*

Adapun hasil dari Laravel untuk fitur *Authentication* yaitu route pada web php yang dapat dilihat di bawah ini

1. Route laravel

Gambar 4.1 adalah titik akhir untuk melakukan login dan regristrasi pada aplikasi. *Route ini* ini dapat diakses melalui URL si Laravel tersebut adalah 127.0.0.1:8000/login dan 127.0.0.1:8000/regritrasi.Pada web.php ini menggunakan middleware auth yang mana nantinya si user harus melakukan login atau regristrasi terlebih dahulu baru dapat melakukan deteksi *phising* pada menu utama.

```
Route::middleware('auth')->group(function () {
    Route::get('/', [CoreController::class, 'index'])->name('home');
    Route::get('/history', [CoreController::class, 'history'])->name('history');

    Route::post('/checkDomain', [CoreController::class, 'checkDomain'])->name('checkDomain');
```

Gambar 4.1 Dokumentasi API Create Account Route laravel

Sumber: Dokumentasi Pribadi

1. Halaman *Login*

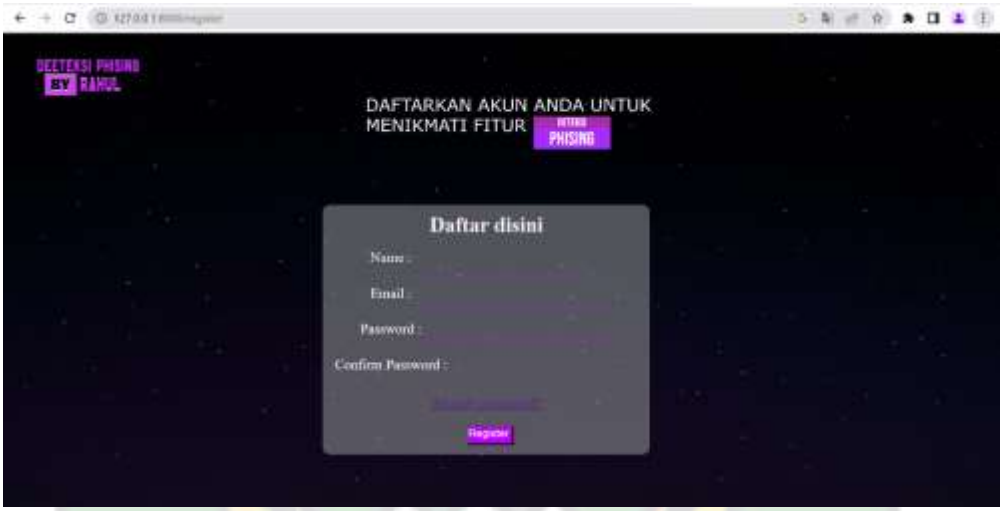
Gambar 4.2 merupakan tampilan halaman *Login*. Pada halaman ini pengguna dapat melakukan login agar dapat menikmati fitur deteksi pada aplikasi. Halaman ini merupakan bagian dari fitur atau modul *Authentication.* Untuk melakukan login user harus mengisi data seperti *email* dan *password* pada *form* yang tersedia.

Gambar 4. 2 Hasil Halaman Login

Sumber: Dokumentasi Pribadi

2. Halaman *Regristrasi*

Berikut dapat dilihat tampilan halaman *regristrasi* pada Gambar 4.11. Halaman ini tentunya berfungsi untuk proses *registrasi* pengguna pada aplikasi. Pada halaman ini, tersedia 4 data yaitu username,email,password dan konfirmasi password yang harus diisi oleh pengguna sebagai *credential* untuk kebutuhan sistem mengotentikasi pengguna yang melakukan proses login pada aplikasi.



Gambar 4. 3 Hasil Halaman Sign In

Sumber: Dokumentasi Pribadi

4.1.2. Fitur *Deteksi*

Tampilan pada Gambar 4.12 dan Gambar 4.13 adalah halaman dari hasil *user story deteksi phising* yang merupakan bagian dari fitur atau modul *Search*. Padahalaman ini memungkinkan pengguna untuk melakukan deteksi *phising* dan aplikasi akan melakukan deteksi. Deteksi yang dilakukan dicari berdasarkan perhitungan dari algoritma decision tree.

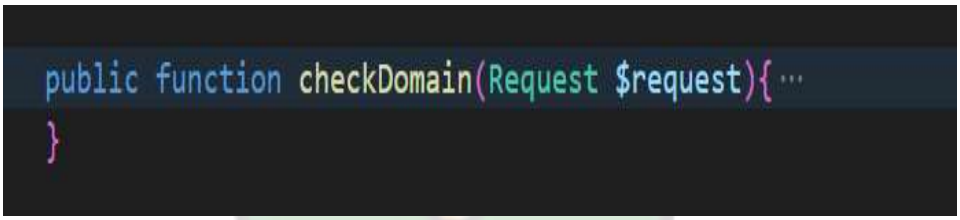
Gambar 4. 4 Halaman deteksi *phising*

Sumber: Dokumentasi Pribadi

Gambar 4. 5 Halaman hasil deteksi *phising*

Sumber: Dokumentasi Pribadi

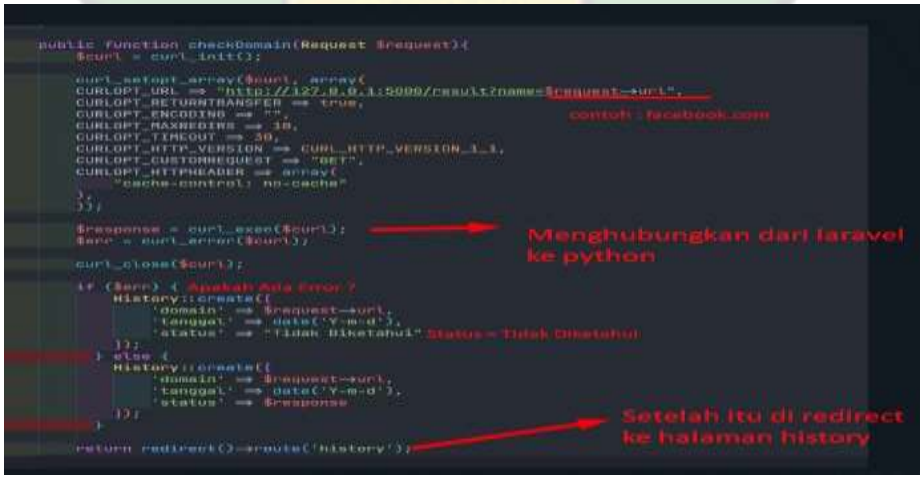
Adapun API untuk fitur deteksi phising dapat dilihat dokumentasinya pada Gambar 4.14 dibawah. Untuk fitur deteksi phising ini dapat diakses melalui URL (Uniform Resource Locator) dari yaitu 127.0.0.1:5000. URL ini harus di akses melalui phyton yang sudah di buat oleh penulis yang kemudian di hubungkan pada Laravel menggunakan request yang pada dilihat pada gambar Gambar 4.19 dibawah ini.



Gambar 4. 6 Dokumentasi API deteksi *phising*

Sumber: Dokumentasi Pribadi

Pada file core.controller.php terdapat beberapa fungsi yang penting seperti menghubungkan Laravel ke phyton yang kemudian Laravel dapat mengambil dapat pada phyton untuk melakukan deteksi phising dan kemudian memasukan hasil dari deteksi ke file history.php (menu hasil deteksi). Untuk lebih jelasnya dapat dilihat pada gambar 4.7.



Gambar 4. 7 Dokumentasi *core.controller*

Sumber: Dokumentasi Pribadi

Setelah itu masuk ke file app.py phyton untuk mengambil parameter (GetResult) yang di kirim dari phising_detection untuk kemudian di periksa yang dapat dilihat pada gambar 4.8.

Gambar 4. 8 Dokumentasi *app.py*

Sumber: Dokumentasi Pribadi

Getresult fungsinya terdapat pada phising_detection yang kemudian pada file phising_detection nanti akan membaca file data set *decision tree* dan terdapat fungsi x dan y yaitu x (mewakili fitur,yang mencakup semua kolom di dataset kecuali kolom terakhir) dan y (mewakili label yang merupakan kolom terakhir dari data set) untuk lebih jelas dapat dilihat pada gambar 4.9.

Gambar 4. 9 Dokumentasi *phising.detection*

Sumber: Dokumentasi Pribadi

Kemudian selanjutkan Memisahkan fitur dan label menjadi data latih dan data uji menggunakan metode train_test_split dari modul sklearn.model_selection. x_train dan y_train mewakili data latih, yang akan digunakan untuk melatih model. x_test dan

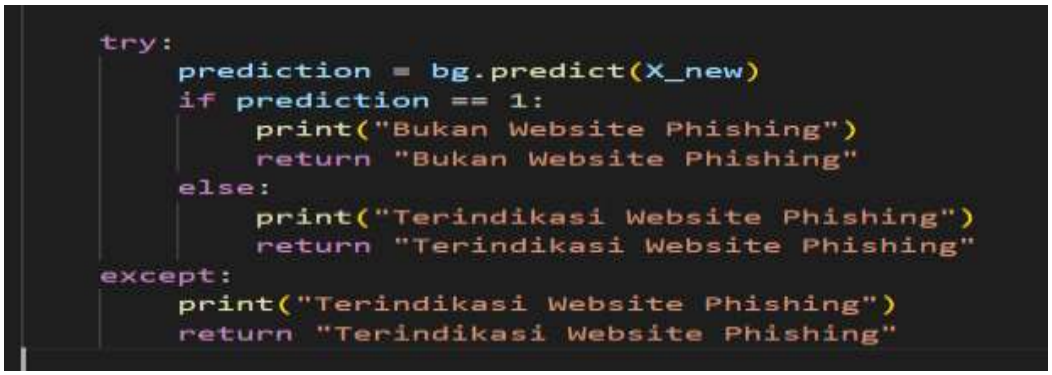
y_test mewakili data uji, yang akan digunakan untuk menguji kinerja model. test_size digunakan untuk menentukan persentase data yang akan digunakan untuk data uji. random_state digunakan untuk mengontrol pembagian data acak. Kemudian membuat instance kelas classifier dari modul sklearn.ensemble dengan Decision Tree Classifier sebagai estimator dasar, max_samples dan max_features diatur ke 1.0 untuk menggunakan seluruh sampel dan fitur di setiap estimator dan n_estimators diatur ke 100 untuk membuat 100 model pengklasifikasi. Menggunakan data latih untuk melatih model pengklasifikasi menggunakan metode fit. Setelah itu menggunakan data uji untuk menghitung akurasi model menggunakan metode score dan mencetak akurasi model. Untuk pemogramannya dapat dilihat pada gambar 4.10.

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=7)
bg = BaggingClassifier(DecisionTreeClassifier(), max_samples=1.0, max_features=1.0, n_estimators=100)
bg.fit(x_train,y_train)
score = bg.score(x_test,y_test)

print('Akurasi : ',score*100)

X_new = []

X_input = url
X_new=feature_extraction.generate_data_set(X_input)
X_new = np.array(X_new).reshape(1,-1)
```

Gambar 4. 10 Dokumentasi *phising.detection*

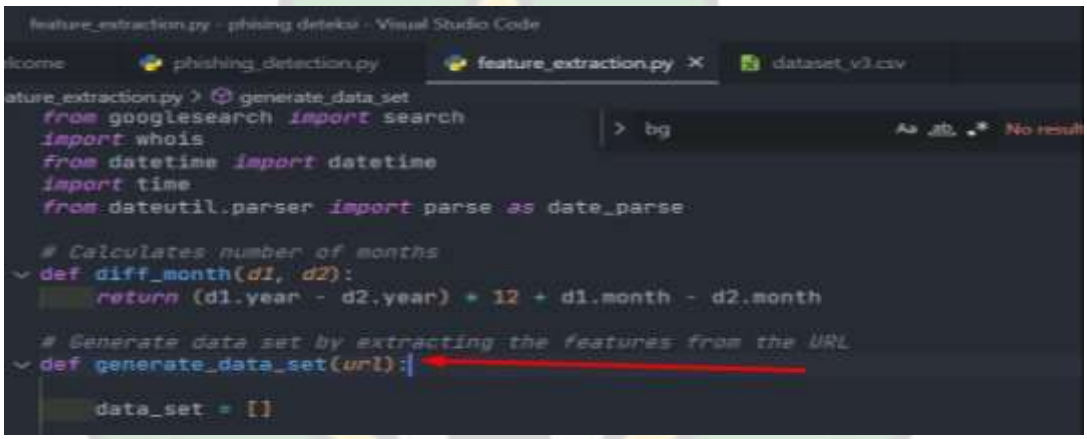
Sumber: Dokumentasi Pribadi

Kemudian setelah itu akan masuk ke dalam beberapa uji test yang terdapat index try yang berfungsi jika ada error otomatis masuk ke dalam except kemudian akan langsung memberi tahu terindikasi website phising, untuk pengecekannya menggunakan fungsi prediction = bg.predict(X_new) yang masuk ke dalam file feature_extraction ke dalam fungsi generate_data_set (url) fungsi generate_data_set berarti menerima parameter url untuk di cek.



Gambar 4. 11 Dokumentasi *prediction*

Sumber: Dokumentasi Pribadi



Gambar 4. 12 Dokumentasi *generate*

Sumber: Dokumentasi Pribadi

Gambar 4. 13 Dokumentasi pengecekan *http*

Sumber: Dokumentasi Pribadi

Gambar 4. 14 Dokumentasi pengecekan nama *domain*

Sumber: Dokumentasi Pribadi

4.2. Sistem Pengujian

Setelah pengerjaan kode program yang dilakukan pada tahapan implementasi, maka selanjutnya dilakukan tahapan pengujian sistem (*system testing*). Secara garis besar, pengujian sistem dilakukan dengan dua skenario pengujian yang diantaranya yaitu *user* yang belum melakukan pendaftaran akun dan *user* yang telah memiliki akun terdaftar. Dimana setiap skenario mencakup seluruh pengujian yang menerapkan kasus uji (*test case*) yang tertera pada Tabel 4.1 dibawah. Pengujian sistem dilakukan sebagaimana yang telah dijabarkan pada bab sebelumnya yaitu, dengan melakukan pengujian dari ujung ke ujung (*end to end*) terhadap sistem yang telah terintegrasi secara menyeluruh.

Tabel 4. 1 *Test Case*

Feature	***User story***	***Pre-Condition***	***Test Case***	***Test Steps***	***Test Data***	***Expected Result***	***Result Pass/Fail***
Authentication	*registrasi*		*registrasi with valid data*	1. Masuk pada halaman *Sign Up* 2. *Masukan username* 3. Masukkan *email* 4. Masukkan *password* 5. *Tap registrasi button*	*Username = rahul* *email =* rahul7@gmail.com *password = 12345678*	1. *User* mendapatkan pesan notifikasi berhasil buat akun 2. Masuk ke halaman *Sign In*	Berhasil
	login		*login with valid data*	1. Masuk pada halaman *login* 2. Masukkan *email* 3. Masukkan *password* 6. *Tap login button*	*email =* rahul7@gmail.com *password = 12345678*	3. *User m*asuk ke halaman utama (Deteksi *Phising*)	Berhasil

Search	*Deteck phising*	*User* sudah *login*	*Success*	1. *Tap search icon button* di *Bottom Navigation Bar* 2. Masuk Url yang ingin di deteksi 3. *Tap search icon* untuk *submit*	1. *Facebook.co* 2. *facebook.com*	3. *User* masuk ke halaman *hasil* 4. *User* dapat melihat daftar hasil dari deteksi *link url* tersebut	Berhasil

Sumber: Dokumentasi Pribadi

4.3. Hasil Penerapan Algoritma Fitur Deteksi

Pada Tabel dibawah ini dapat dilihat perbandingan pengujian 3 aplikasi yang sudah di bangun pada deteksi phising dengan menggunakan *17 url phising* dan 5 kali percobaan. Untuk lebih lengkapnya dapat pada table dibawah ini.

Tabel 4. 2 Pengujian *url phising decision tree*

No	Link *Phising*	Terdeteksi *Phising* (T)/Tidak Terdeteksi *Phising* (F)				
		Decision Tree				
1	*https://linkvip.barux.xyz/chat/?s=f52fd93b*	T	T	T	T	T
2	*ediafire.233-45-download.us/p/?s=ZDwcSye*	T	T	T	T	T
3	*http://mtw.tl/lfgmthh*	T	T	T	T	T
4	*https://mediafire.plaja.biz.id/p/?s=ABlcJ*	T	T	T	T	T
5	*Facebook.co*	T	T	T	T	T
6	*hxxps://suiprotocols.com/*	T	T	T	T	T
7	*https://rajahoki69.com/*	T	T	T	T	T
8	*http://007sites.com*	F	F	F	F	F
9	*Stanford.io/2UHIu65*	T	T	T	T	T
10	*http://file.masayip.com/WE0d*	F	F	F	F	F
11	*http://tesla-online.net*	T	T	T	T	T
12	*Http://bitly*	T	T	T	T	T
13	*https://mediafire-smkviralterbaru-downlaodfile.freq.gq/*	T	T	T	T	T
14	*https://mediafire.35-78-news.org/p/?s=SENWQUWEbUd*	T	T	T	T	T
15	*http://bancoolombia.eshost.com.ar/*	T	T	T	T	T
16	*http://www.magazineluiza.cc/index/login/*	T	T	T	T	T
17	*http://faturalulumarco.com/*	T	T	T	T	T

Sumber: Dokumentasi Pribadi

Tabel 4. 3 Pengujian *url phising check pish*

No	Link *Phising*	Terdeteksi *Phising* (T)/Tidak Terdeteksi *Phising* (F)				
		Check Phish				
1	*https://linkvip.barux.xyz/chat/?s=f52fd93b*	F	F	F	F	F

2	*ediafire.233-45-download.us/p/?s=ZDwcSye*	F	F	F	F	F
3	*http://mtw.tl/lfgmthh*	F	F	F	F	F
4	*https://mediafire.plaja.biz.id/p/?s=ABlcJ*	F	F	F	F	F
5	*Facebook.co*	T	T	T	T	T
6	*hxxps://suiprotocols.com/*	F	F	F	F	F
7	*https://rajahoki69.com/*	F	F	F	F	F
8	*http://007sites.com*	F	F	F	F	F
9	*Stanford.io/2UHIu65*	F	F	F	F	F
10	*http://file.masayip.com/WE0d*	F	F	F	F	F
11	*http://tesla-online.net*	F	F	F	F	F
12	*Http://bitly*	F	F	F	F	F
13	*https://mediafire-smkviralterbaru-downlaodfile.freq.gq/*	F	F	F	F	F
14	*https://mediafire.35-78-news.org/p/?s=SENWQUWEbUd*	F	F	F	F	F
15	*http://bancoolombia.eshost.com.ar/*	F	F	F	F	F
16	*http://www.magazineluiza.cc/index/login/*	F	F	F	F	F
17	*http://faturalulumarco.com/*	F	F	F	F	F

Sumber: Dokumentasi Pribadi

Tabel 4. 4 Pengujian *url phising decision tree*

No	Link *Phising*	Terdeteksi *Phising* (T)/Tidak Terdeteksi *Phising* (F)				
		TreatCop				
1	*https://linkvip.barux.xyz/chat/?s=f52fd93b*	F	F	F	F	F
2	*ediafire.233-45-download.us/p/?s=ZDwcSye*	F	F	F	F	F
3	*http://mtw.tl/lfgmthh*	F	F	F	F	F
4	*https://mediafire.plaja.biz.id/p/?s=ABlcJ*	F	F	F	F	F
5	*Facebook.co*	T	T	T	T	T
6	*hxxps://suiprotocols.com/*	F	F	F	F	F
7	*https://rajahoki69.com/*	F	F	F	F	F
8	*http://007sites.com*	F	F	F	F	F
9	*Stanford.io/2UHIu65*	F	F	F	F	F
10	*http://file.masayip.com/WE0d*	F	F	F	F	F
11	*http://tesla-online.net*	F	F	F	F	F
12	*Http://bitly*	F	F	F	F	F
13	*https://mediafire-smkviralterbaru-*	F	F	F	F	F

	downlaodfile.freq.gq/					
14	*https://mediafire.35-78-news.org/p/?s=SENWQUWEbUd*	F	F	F	F	F
15	*http://bancoolombia.eshost.com.ar/*	F	F	F	F	F
16	*http://www.magazineluiza.cc/index/login/*	F	F	F	F	F
17	*http://faturalulumarco.com/*	T	T	T	T	T

Sumber: Dokumentasi Pribadi

Berdasarkan deteksi telah dilakukan terhadap tiga aplikasi *deteksi phising* dengan pengujian 5 kali menggunakan 17 *url phising* dapat dilihat apllikasi menggunakan algoritma decision tree berhasil melakukan deteksi pada 15 url sedangkan aplikasi Treatcop berhasil melakukan deteksi pada 1 url phising dan Check Phish berhasil melakukan deteksi pada 2 url phising. Dari data ini dapat dilihat aplikasi yang di bangun menggunakan algoritma *decision tree* lebih baik untuk mendeteksi *url phising.* Kemudian pada tabel di bawah akan dilakukan deteksi menggunakan 17 url bukan phising untuk memeriksa apakah aplikasi *error* sehingga semua *url* di deteksi *phising.* Untuk lebih lengkapnya dapat dilihat pada tabel di bawah ini.

Tabel 4. 5 Pengujian url bukan *phising decision tree*

No	Link *Phising*	Terdeteksi *Phising* (F)/Tidak Terdeteksi *Phising* (T)				
		Decision Tree				
1	*https://www.tesla.com/*	T	T	T	T	T
2	*https://www.gojek.com/id-id/*	T	T	T	T	T
3	*https://ahrefs.com/*	T	T	T	T	T
4	*https://www.dollarshaveclub.com/shop*	T	T	T	T	T
5	*Facebook.com*	T	T	T	T	T
6	*https://www.hubspot.com/*	T	T	T	T	T
7	*https://www.jago.com/id*	T	T	T	T	T
8	*https://www.apple.com/id/*	T	T	T	T	T
9	*https://www.dfdg.com/*	T	T	T	T	T
10	*https://www.bluleadz.com/*	T	T	T	T	T
11	*https://www.paragonoak.com/*	T	T	T	T	T
12	*https://www.prco.com/*	T	T	T	T	T
13	*https://heymale.id/*	T	T	T	T	T

14	*https://majoo.id/*	T	T	T	T	T
15	*https://bro.do/*	T	T	T	T	T
16	*http://www.popolucathelabel.com/*	T	T	T	T	T
17	*https://berrybenka.com/*	T	T	T	T	T

Tabel 4. 6 Pengujian url bukan *phising decision tree*

No	Link *Phising*	Terdeteksi *Phising* (F)/Tidak Terdeteksi *Phising* (T)				
		Check Phish				
1	*https://www.tesla.com/*	T	T	T	T	T
2	*https://www.gojek.com/id-id/*	T	T	T	T	T
3	*https://ahrefs.com/*	T	T	T	T	T
4	*https://www.dollarshaveclub.com/shop*	T	T	T	T	T
5	*Facebook.com*	T	T	T	T	T
6	*https://www.hubspot.com/*	T	T	T	T	T
7	*https://www.jago.com/id*	T	T	T	T	T
8	*https://www.apple.com/id/*	T	T	T	T	T
9	*https://www.dfdg.com/*	T	T	T	T	T
10	*https://www.bluleadz.com/*	T	T	T	T	T
11	*https://www.paragonoak.com/*	T	T	T	T	T
12	*https://www.prco.com/*	T	T	T	T	T
13	*https://heymale.id/*	T	T	T	T	T
14	*https://majoo.id/*	T	T	T	T	T
15	*https://bro.do/*	T	T	T	T	T
16	*http://www.popolucathelabel.com/*	T	T	T	T	T
17	*https://berrybenka.com/*	T	T	T	T	T

Tabel 4. 7 Pengujian *url* bukan *phising decision tree*

No	Link *Phising*	Terdeteksi *Phising* (F)/Tidak Terdeteksi *Phising* (T)				
		TreatCop				
1	*https://www.tesla.com/*	T	T	T	T	T
2	*https://www.gojek.com/id-id/*	T	T	T	T	T
3	*https://ahrefs.com/*	T	T	T	T	T

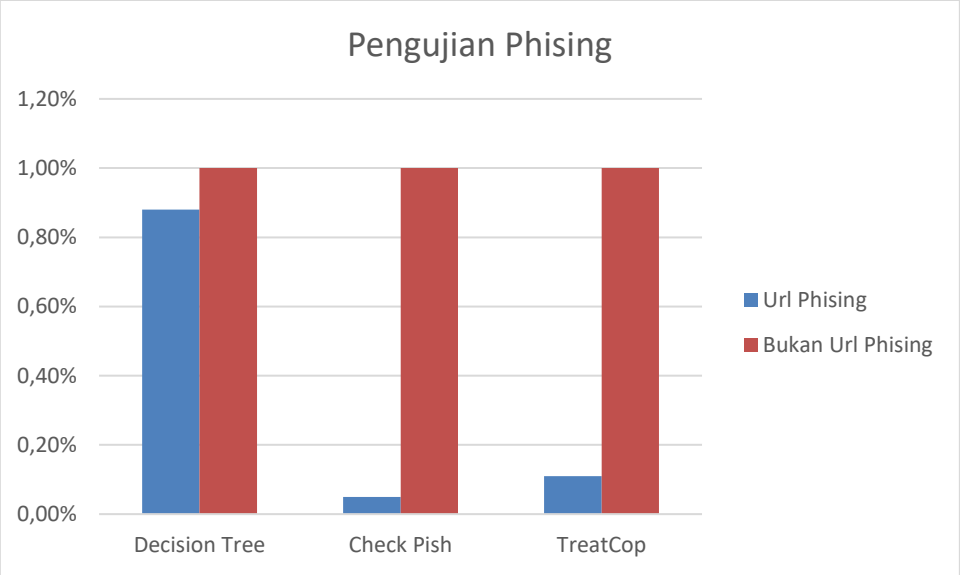
4	*https://www.dollarshaveclub.com/shop*	T	T	T	T	T
5	*Facebook.com*	T	T	T	T	T
6	*https://www.hubspot.com/*	T	T	T	T	T
7	*https://www.jago.com/id*	T	T	T	T	T
8	*https://www.apple.com/id/*	T	T	T	T	T
9	*https://www.dfdg.com/*	T	T	T	T	T
10	*https://www.bluleadz.com/*	T	T	T	T	T
11	*https://www.paragonoak.com/*	T	T	T	T	T
12	*https://www.prco.com/*	T	T	T	T	T
13	*https://heymale.id/*	T	T	T	T	T
14	*https://majoo.id/*	T	T	T	T	T
15	*https://bro.do/*	T	T	T	T	T
16	*http://www.popolucathelabel.com/*	T	T	T	T	T
17	*https://berrybenka.com/*	T	T	T	T	T

Berdasarkan deteksi telah dilakukan terhadap tiga aplikasi *deteksi phising* dengan pengujian 5 kali menggunakan 17 *url* bukan *phising* dapat dilihat ketiga apllikasi tidak mendeteksi phising pada 17 url bukan phising dan ini menunjukan aplikasi tidak error untuk melakukan deteksi *phising.*

$$\text{Persen (\%)} = \frac{\text{Jumlah Bagian}}{\text{Jumlah Seluruh}} \times 100\%$$

Gambar 4. 15 Perhitungan persen

Dari tabel dengan 17 url phising dengan 5 kali pengujian, aplikasi *decision tree* mendapatkan nilai 0,88%, sedangkan aplikasi *check pish* mendapatkan nilai 0,05% dan aplikasi *treatcop* mendapatkan nilai 0,11%. Pada tabel 17 url bukan phising dengan 5 kali pengujian, apliaksi *decision tree*, *checkpish* dan *treatcop* mendapatkan nilai 1%.untuk lebih jelasnya dapat dilihat pada diagram batang di bawah ini.



Gambar 4. 16 Diagram batang pengujian *phising*

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah dilakukan penelitian dan evaluasi, dapat disimpulkan bahwa:

1. Aplikasi deteksi *phising* berbasis *website* yang dikembangkan dapat memenuhi tujuan penelitian yaitu mengembangkan aplikasi deteksi *phising* berbasis *website* dan membangun fitur *deteksi phising* menggunakan algoritma *decision tree.*
2. Aplikasi ini juga dapat menutup kekurangan dari aplikasi deteksi *phising* berbasis web yang sudah ada sebelumnya, yaitu dari segi keberhasilan deteksi *url* yang mendapatkan nilai keberhasilan lebih baik
3. Algoritma *decision tree* baik untuk digunakan dalam fitur *deteksi phising* pada aplikasi deteksi *phising* dalam kasus dimana url *phising* yang dapat mencuri data pengguna.

5.2. Saran

Setelah melakukan penelitian ini, terdapat beberapa saran untuk penelitian selanjutnya yang dapat dilakukan agar dapat meningkatkan manfaat dan kegunaan aplikasi deteksi *phising* bagi penggunanya. Adapun saran yang dapat diberikan untuk penelitian selanjutnya adalah:

1. Dapat membuat aplikasi bisa di gunakan dalam berbagai *platform* seperti *mobile,web,desktop* dll.
2. Dapat menambah menu hasil yang lebih lengkap seperti *IP*,lokasi,tujuan pencurian data dan lain sebagainya.

3. Menambahkan fitur pembelajaran deteksi *phising*, dengan menambahkan modul-modul pelajaran yang dapat membantu pengguna dalam memahami *phising*
4. Dapat mengembangkan fitur bahasa, sehingga aplikasi dapat digunakan dalam beberapa bahasa.
5. Dapat menggabungkan beberapa algoritma untuk mendapatkan keberhasilan deteksi yang lebih baik seperti *decision tree* di tambah dengan algoritma svm dan random forest.

DAFTAR PUSTAKA

Ahmadian, H., & Sabri, A. (2021). Teknik Penyerangan Phishing Pada Social Engineering Menggunakan Set Dan Pencegahannya. *Djtechno Jurnal Teknologi Informasi*, *2*(1), 13–20. https://doi.org/10.46576/djtechno.v2i1.1251

Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *3*(2), 206. https://doi.org/10.30998/string.v3i2.3048

Deepak Pathak, M. R., & Sandhia, D. G. K. (2022). Phishing Detection using Machine Learning. *Interantional Journal of Scientific Research in Engineering and Management*, *06*(04), 4233–4238. https://doi.org/10.55041/ijsrem12161

Drury, V., Roepke, R., Schroeder, U., & Meyer, U. (2023). *Analyzing and Creating Malicious URLs: A Comparative Study on Anti-Phishing Learning Games*. *April*. https://doi.org/10.14722/usec.2022.23085

Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhan, W. (2021). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, *11*(1), 48. https://doi.org/10.36448/expert.v11i1.2012

Giap Yo Ceng, S. T. (2017). DATA MINING IDENTIFIKASI WEBSITE PHISING MENGGUNAKAN ALGORITMA C4.5 Tomy Salim 1) Yo Ceng Giap 2). *Technology Acceptance Model*, *8*, 130–135.

Hamzan, M. A., Nrartha, I. M. A., & Wiryajati, I. K. (2022). Rancang Bangun Sistem Pemantauan Daya Listrik Berbasis Android Menggunakan Teknologi React Native. *Dielektrika*, *9*(1), 42–50. http://www.dielektrika.unram.ac.id/index.php/dielektrika/article/view/292

Handayani, P. K. (2020). Penerapan Principal Component Analysis untuk Peningkatan Kinerja Algoritma Decision Tree pada Iris Dataset. *Indonesian Journal of Technology, Informatics and Science (IJTIS)*, *1*(2), 55–58. https://doi.org/10.24176/ijtis.v1i2.4939

Herho, S. H. S. (2017). Tutorial Pemrograman Python 2 Untuk Pemula. *WCPL Press*, 1–140.

Irawan, A. S. Y., Heryana, N., Hopipah, H. S., & Rahma, D. (2021). Identifikasi Website Phishing dengan Perbandingan Algoritma Klasifikasi. *Syntax : Jurnal Informatika*, *10*(01), 57–67. https://doi.org/10.35706/syji.v10i01.5292

Mahajan, R., & Siddavatam, I. (2018). Phishing Website Detection using Machine Learning Algorithms. *International Journal of Computer Applications*, *181*(23), 45–47. https://doi.org/10.5120/ijca2018918026

Muhammad, R., & Ardimansyah, M. I. (2022). *Sistem Deteksi Kecanduan Pornografi Berbasis Chatbot Menggunakan Pornography Addiction Screening Tool (PAST)*. *4*(3), 1616–1624. https://doi.org/10.47065/bits.v4i3.2660

Nugraha, A. F., Aziza, R. F. A., & Pristiyanto, Y. (2022). Penerapan metode stacking dan random forest untuk meningkatkan kinerja klasifikasi pada proses deteksi web phishing. *Jurnal Infomedia*, *7*(1), 39–44. http://e-jurnal.pnl.ac.id/infomedia/article/view/2959%0Ahttp://e-jurnal.pnl.ac.id/infomedia/article/viewFile/2959/2506

Permana, I. G. T., Rusdianto, D. S., & Fanani, L. (2019). Pengembangan Sistem Presensi berbasis Lokasi menggunakan Geofence WiFi dan REST API pada Fakultas Ilmu Komputer Universitas Brawijaya. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(9), 9305–9313. http://j-ptiik.ub.ac.id

Rahardja, U., Lutfiani, N., & Rahmawati, R. (2018). APTISI Student Perception to the News on The APTISI Website. *Jurnal Ilmiah SISFOTENIKA*, *8*(2), 117–127.

Sidauruk, C. N., Purnama, A., Zani, T., Ilmu, F., & Universitas, T. (2020). Pembangunan Aplikasi Augmented Reality Dan Implementasi Video Alat Musik Tradisional Jawa Barat. *E-Proceeding of Applied Science*, *6*(2), 4161–4170.

Suriyani, I. (2020). Sistem Informasi Pembayaran Rekening Air Berbasis Web Pada Pamsimas Jorong Panyalai. *Indonesian Journal of Technology, Informatics and Science (IJTIS)*, *1*(2), 21–26. https://doi.org/10.24176/ijtis.v1i2.4833

Syahrizal, M., & Haryati, H. (2018). Perancangan Aplikasi Sistem Pakar Deteksi Kerusakan Mesin Alat Berat (Beko) Dengan Menerapkan Metode Teorema Bayes. *Jurnal Media Informatika Budidarma*, *2*(2), 23–33. https://doi.org/10.30865/mib.v2i2.596

Wijaya, W., Tolle, H., & Kharisma, A. P. (2018). Rancang Bangun Aplikasi Geotagging Social Report Bencana Banjir. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *II*(7), 2817–2824.

Yagoyamu, T. (2020). *Pengembangan sistem informasi berbasis web menggunakan waterfall method untuk memperkenalkan kebudayaan dan pariwisata suku asmat*.

YANG, X., YAN, L., YANG, B., & LI, Y. (2017). Phishing Website Detection Using C4.5 Decision Tree. *DEStech Transactions on Computer Science and Engineering*, *itme*, 119–124. https://doi.org/10.12783/dtcse/itme2017/7975

LAMPIRAN

Adapun Lampiran dari penelitian Pengembangan Aplikasi Deteksi Phising Berbasis Web Menggunakan Algoritma Decision Tree dapat diakses pada link berikut:

https://1drv.ms/f/s!AgPrs2t6jXRCaaB0w5Xiyd_43Ls?e=uIQXuK