

Desain dan Pengembangan *Website* untuk Mendeteksi *Malware* Menggunakan *Framework Flask* yang Diintegrasikan dengan *Machine Learning*

Ciksadan¹, Sopian Soim², Nurlita Jami³

Teknik Elektro, Politeknik Negeri Sriwijaya, Jl. Sriwijaya Negara, Bukit Lama, Kec. Ilir Barat I, Kota Palembang, Sumatera Selatan, Indonesia, 30128

e-mail: ¹ ciksadanc@gmail.com, ²sopiansoim@gmail.com, ³nurlitajami12@gmail.com

Submitted Date: July 12th, 2024

Reviewed Date: July 18th, 2024

Revised Date: July 20th, 2024

Accepted Date: July 24th, 2024

Abstract

One of the most widely used media for information dissemination is the website. A dynamic and informative website will make it easier for users to access information. Web development often requires complex technologies. One method that can simplify the development process is using the Flask framework, which offers flexibility and freedom to developers. A website must also have functionality to be useful; one current issue is the increasing number of malware file cases. Therefore, there is a need for a medium that can analyze a file. However, currently, there are limited services available for this purpose. This research aims to build a website that detects malware files using the Flask framework integrated with machine learning for malware file detection. Through this research, a website with five informative menus has been developed, featuring a dynamic and easily accessible interface with a malware file detection capability reaching 99% accuracy.

Keywords: *Website*; Flask; *malware detection*

Abstrak

Media penyebaran informasi yang paling banyak digunakan adalah *website*. Namun pengembangan *website* untuk menghasilkan tampilan yang dinamis dan informatif sering kali memerlukan teknologi yang rumit. Salah satu metode yang dapat mempermudah pengerjaan adalah menggunakan *framework* Flask yang menawarkan fleksibilitas dan kebebasan kepada *developer*. Suatu *website* juga harus memiliki fungsionalitas agar dapat bermanfaat, salah satu masalah saat ini adalah peningkatan kasus *file malware*. Untuk itu diperlukan suatu media yang dapat digunakan untuk dapat menganalisa suatu *file*. Namun saat ini media yang menyediakan layanan untuk kasus tersebut masih terbatas. Untuk itu pada penelitian ini akan membangun suatu *website* yang bertujuan untuk mendeteksi *file malware* menggunakan *framework* Flask yang diintegrasikan dengan *machine learning* sebagai pendeteksi *file malware*. Melalui penelitian ini dihasilkan suatu website dengan lima menu informatif dengan tampilan *interface* yang dinamis dan mudah diakses dengan kemampuan pendeteksi *file malware* mencapai akurasi 99%.

Kata Kunci: *Website*; Flask; *malware detection*

1 Pendahuluan

Kemunculan internet melahirkan media seperti *website* yang memiliki peran penting dalam gaya hidup modern saat ini. Berbagai aktivitas seperti *browsing*, bersosialisasi sosial hingga mengakses berbagai informasi dapat dilakukan melalui *website* (Prasetyo, Haeruddin, and Ariesryo 2024). Kelebihan utama dari

aplikasi berbasis *website* adalah dapat dijalankan secara langsung melalui *web browser* selama pengguna terhubung ke internet tanpa memerlukan versi sistem operasi tertentu (Tambuwun, Sengkey, and Rindengan 2017). Melalui perkembangan yang ada, pengunjung *website* juga mengalami penambahan yang meliputi berbagai aspek mulai dari individu

bahkan kegiatan bisnis (Budianto and Saian 2023). Dengan banyaknya pengguna yang mengakses *website* tentunya diperlukan kualitas *website* yang baik dinilai dari tampilan yang tidak rumit, konten yang mendukung, navigasi yang fungsional, dan desain yang menarik (Ayudhitama and Pujiyanto 2020).

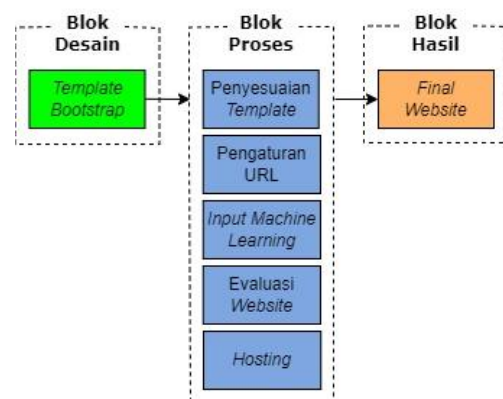
Perancangan suatu *website* akan optimal apabila dapat menampilkan suatu halaman yang dinamis dan dapat memberikan informasi dengan jelas serta tampilan *user interface* yang dapat digunakan dengan mudah (Agustin et al. n.d.). Dalam jurnal (Santoso and Saian 2023) mengatakan pembuatan *website* menggunakan Oracle Form dinilai tidak efisien karena setiap pengguna yang ingin mengaksesnya harus menggunakan versi perangkat lunak yang sesuai. Penelitian lainnya (Rico, Rudianto, and Tanaamah 2024) yang menggunakan metode *prototype* aplikasi Figma memiliki kekurangan dalam desain yang masih kurang sesuai untuk digunakan oleh masyarakat, sehingga banyak penyesuaian ulang yang perlu dilakukan mengakibatkan pembuatan *website* memerlukan waktu yang lebih lama. Pada penelitian (Pamungkas 2019) membuat *website* dengan teknik responsive web design (RWD) memerlukan penggunaan internet yang lebih besar dan jika tanpa teknik RWD *website* tidak bisa melakukan penyesuaian desain tampilan secara langsung. Untuk mengatasi masalah tersebut, penelitian ini akan menggunakan *framework* Flask. Flask adalah *web framework* berbahasa Python yang memberikan kebebasan pada *developer* untuk mengembangkan suatu *website* dan tergolong dalam *micro-framework* karena tidak memerlukan *tools* tambahan untuk menggunakannya (Ningtyas and Setiyawati 2021) Suatu *website* juga harus memiliki tujuan yang jelas agar dapat bermanfaat bagi penggunanya, untuk itu tidak hanya dari sisi tampilan namun juga dari fungsionalitas *website* juga perlu diperhatikan (Ayudhitama and Pujiyanto 2020).

Saat ini penyebaran *malware* meningkat dengan cepat yang terbukti pada pertengahan tahun 2020 terdapat 10,6 juta aplikasi *malware* yang terdeteksi (Islam et al. 2023). Permasalahan *malware* sudah banyak diteliti dan dapat dilakukan deteksi menggunakan *machine learning* untuk mengetahui apakah *file* tersebut

termasuk *malware* atau *non-malware* (Guerra-manzanares 2024), (Mobarak et al. 2023). Namun penelitian-penelitian tersebut terbatas hanya membahas pada *machine learning*, tidak diimplementasikan dalam suatu *website* yang dapat digunakan secara langsung oleh banyak orang. Permasalahan utamanya adalah bagaimana membangun suatu *website* dengan tampilan *user interface* yang dapat dengan mudah digunakan serta bagaimana *website* dapat diintegrasikan dengan *machine learning*. Berdasarkan hal tersebut penelitian ini ditujukan untuk membangun suatu *website* yang dapat mendeteksi *malware* menggunakan *machine learning*.

2 Metodologi Penelitian

Metode penelitian merupakan tahap peneliti melakukan perancangan penelitian berdasarkan tahapan yang telah dibuat agar mendapatkan hasil akhir penelitian yang akurat. Tahapan penelitian ditunjukkan pada gambar 1.



Gambar 1. Blok diagram pembuatan *website*

Langkah pertama dalam membangun situs *website* adalah memilih *template bootstrap*. *Bootstrap* adalah *template* yang terdiri dari set *file* HTML, CSS, dan JavaScript yang digunakan untuk membangun halaman *website* (Al-Fadhillah Wafiq et al. 2023). Sedangkan untuk melakukan pengolahan data memakai bahasa pemrograman Python. Setelah diperoleh *template* yang sesuai, dilakukan *initialize* Flask sebagai konfigurasi awal untuk menjalankan *website*.

```
# Initialize Flask app
app = Flask(__name__, static_folder='static', static_url_path='/static')
app.secret_key = "poltek"
```

Gambar 2. *Initialize* pada Flask

Penyesuaian gambar yang digunakan pada folder *static* perlu disesuaikan. Setiap *file static* perlu diberikan tambahan kode program agar *file* dapat terbaca saat *website* dijalankan.

```
<script src="{{ url_for('static', filename='lib/lightbox/js/lightbox.min.js') }}"></script>  
<script src="{{ url_for('static', filename='js/main.js') }}"></script>
```

Gambar 3. Kode program pada *file static*

Pengaturan URL juga dilakukan pada tiap *file HTML* yang akan menjadi alamat unik untuk menavigasi antarhalaman sebuah *website*.

```
<a href="index.html" class="nav-item nav-link active">Home</a>  
<a href="{{ url_for('about') }}" class="nav-item nav-link">About</a>  
<a href="{{ url_for('service') }}" class="nav-item nav-link">Method</a>  
<a href="{{ url_for('causes') }}" class="nav-item nav-link">Causes</a>
```

Gambar 4. Penyesuaian URL setiap menu

Dengan penyesuaian URL, *user* dapat langsung diarahkan ke tindakan selanjutnya saat mengklik *button* yang ada. Proses selanjutnya adalah menginput model *machine learning* yang dapat dilakukan dengan *drag and drop* kedalam folder *website*. Setelahnya model harus di *load* pada *file* yang bertugas sebagai *file* utama untuk menjalankan *website*.

```
# Fungsi untuk membuat model  
def load_model():  
    model_path = 'model.pkl'  
    model = joblib.load(model_path)  
    return model
```

Gambar 5. Fungsi *load* model *machine learning*

Berdasarkan Gambar 5 “*model.pkl*” adalah *file machine learning* berformat *pickle* yang akan di *load* dan akan menganalisa tiap fitur yang ada pada *file* yang diunggah oleh *user*. Hasil deteksi oleh *machine learning* akan ditampilkan dalam bentuk tulisan pada *website*. Setelah tahap ini, *website* perlu dievaluasi secara keseluruhan dengan mengoperasikannya melalui *local host*. Evaluasi ini dilakukan dengan mencoba *website* secara langsung untuk memastikan semua fitur yang disediakan dapat berfungsi dengan baik. Jika telah dipastikan semua fitur dapat berfungsi, proses terakhir adalah melakukan *hosting*. *Hosting* ialah layanan penyimpanan data serta *file website* di *server* sehingga dapat diakses oleh pengguna melalui internet (Tambuwun et al. 2017). *Hosting* dapat dilakukan dengan berbagai metode, pada penelitian ini *hosting* dilakukan menggunakan platform PythonAnywhere yang

disediakan oleh PythonAnywhere LLP dengan sistem berbasis *cloud*.

3 Hasil dan Pembahasan

Dihasilkan sebuah situs *website* yang menggunakan bahasa pemrograman Python dan Flask yang diintegrasikan dengan kecerdasan buatan untuk menganalisa *file malware*. Karena tujuan *website* untuk mendeteksi *file, machine learning* dapat membantu mengenali *file* berdasarkan pola dan karakteristiknya.

a. Hasil tampilan *website*

Berikut ini tampilan hasil akhir *website* yang terdiri dari lima menu utama dan informasi pendukung yang dapat digunakan oleh *user*.



Gambar 6. Menu Home *website*

Gambar 6 menunjukkan menu Home yang akan ditampilkan saat pengguna pertama kali mengakses *website*. Pada menu ini pengguna dapat melihat empat menu utama lainnya yaitu About Us, Method, Causes, dan Check. Pada menu Home menampilkan berbagai informasi mengenai deskripsi singkat terkait fungsi *website*, jenis-jenis *malware* yang umum digunakan oleh *hacker* dan jumlah kasus akibat kejahatan *malware*. Selain itu *website* juga dilengkapi dengan tampilan gambar-gambar yang relevan agar dapat meningkatkan pengalaman visual *user* dan memperjelas konten yang diberikan.



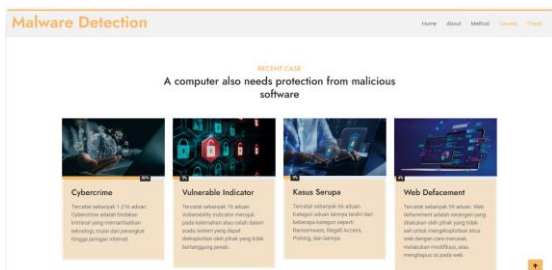
Gambar 7. Menu About Us *website*

Gambar 7 menampilkan menu About Us yang berisi tujuan utama dari *website* beserta dengan visi dan misi pembangunan *website*. Pada halaman ini, menu “About”, “Vision”, dan “Mission” memanfaatkan *navigation bar* untuk meningkatkan interaktivitas dengan *user*.



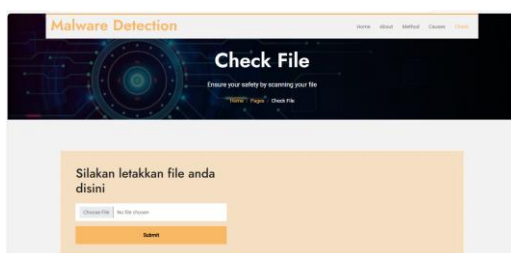
Gambar 8. Menu Method *website*

Pada gambar 8 menunjukkan menu Method yang memberikan informasi mengenai komponen pendukung untuk membangun *website*, dimana terdapat *machine learning* dengan algoritma Random Forest sebagai mesin pendeteksi yang bertugas menganalisa *file* sehingga dapat diketahui apakah *file* yang dideteksi termasuk dalam jenis *malware* atau *file non-malware*.



Gambar 9. Menu Causes *website*

Menu Causes memberikan informasi mengenai serangan siber yang telah terjadi di Indonesia yang dihimpun melalui laporan Badan Keamanan Siber Indonesia pada tahun 2023.



Gambar 10. Menu Check *website*

Menu Check merupakan menu dengan fungsi utama dimana pengguna dapat mengetahui apakah *file* yang dimilikinya terdeteksi sebagai *malware* atau *file* yang aman. Pengguna diminta untuk mengupload *file* melalui bar yang telah disediakan. Selanjutnya *file* akan diproses oleh *machine learning* untuk di analisa isi *file* tersebut. Hasil analisa akan ditampilkan dalam bentuk tulisan dalam menu Check.

b. Pengujian Sistem

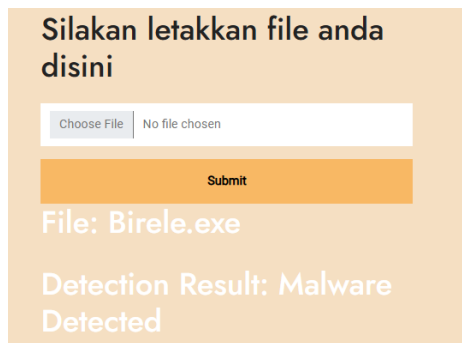
Pengujian sistem dilakukan dengan melakukan akses secara langsung oleh pengguna sehingga diketahui tingkat fungsionalitas sistem. Percobaan dilakukan sebanyak 20 kali percobaan pada *website* yang juga melakukan pengujian menggunakan *file malware* dan *non-malware* secara bergantian. Metode *Black Box Testing* yang berfokus pada *input* dan *output* yang diberikan, digunakan untuk menganalisa parameter penilaian *website*. Berikut ini tabel hasil uji coba *website* dalam 20 kali percobaan.

Tabel 1. Tabel Hasil Pengujian

Proses Uji Coba	Hasil yang Diharapkan	Hasil Uji Coba
Pengguna dapat mengakses seluruh menu dalam <i>website</i>	Pengaturan URL berjalan dengan baik	Berhasil
Informasi yang menggunakan	Informasi akan terlihat jika pengguna menekan tombol	Berhasil
Seluruh animasi dapat terlihat	Gambar dapat ditampilkan dengan sempurna	Berhasil
<i>Machine learning</i> terintegrasi dengan <i>website</i>	<i>Machine learning</i> mampu mendeteksi <i>file</i> yang dimasukkan oleh pengguna	Berhasil
<i>Website</i> menampilkan hasil deteksi <i>file</i>	Menampilkan “No Malware Detected” untuk <i>file non-malware</i> atau “Malware Detected” untuk <i>file malware</i>	Berhasil

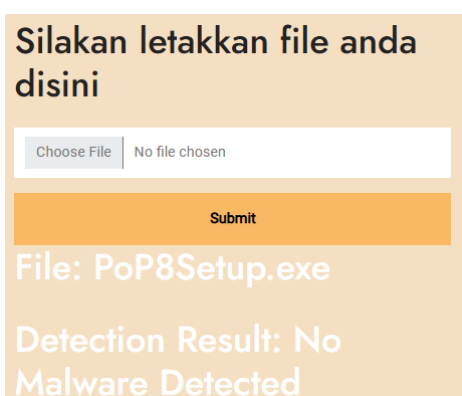
Dasil hasil pengujian menunjukkan bahwa *website* sudah dapat berjalan sesuai harapan dengan semua fungsi dan *navigation bar* berjalan

dengan baik. Berdasarkan pengujian yang dilakukan terdapat dua *output* yang ditampilkan pada Gambar 11 dan 12.



Gambar 11. Hasil deteksi *file malware*

File Birele.exe diunggah melalui *website* untuk dianalisa lebih mendalam meliputi analisa fitur dan pola distribusi. Setelah proses analisa dilakukan, hasilnya menunjukkan bahwa *file* Birele.exe memiliki karakteristik yang mengindikasikan *file* berbahaya. Dengan demikian *file* tersebut diklasifikasikan oleh *machine learning* sebagai *malware* yang berpotensi menimbulkan ancaman terhadap keamanan sistem komputer. Sama seperti *file* Birele.exe, *file* yang diunggah ke *website* akan melalui proses analisis yang sama namun hasil analisa bergantung pada karakteristik yang dimiliki oleh tiap *file*. Dilakukan percobaan menggunakan *file* PoP8Setup.exe dan hasil analisis mengindikasikan bahwa *file* tidak termasuk dalam kategori berbahaya.



Gambar 12. Hasil deteksi *file non-malware*

Analisis berdasarkan fitur yang dievaluasi oleh *machine learning* terhadap kedua *file* ditunjukkan oleh Tabel 2 berikut.

Tabel 2. Hasil analisa karakteristik *file*

Nama Fitur	Malware	Non-Malware
Machine	332	332
SizeOfOptionalHeader	224	224
Characteristics	271	258
MajorLinkerVersion	5	8
MinorLinkerVersion	5	0
SizeOfCode	118784	8192
SizeOfInitializedData	4096	2743296
SizeOfUninitializedData	102400	0
AddressOfEntryPoint	222352	5549
BaseOfCode	106496	4096
BaseOfData	225280	12288
ImageBase	4194304	4194304
SectionAlignment	4096	4096
FileAlignment	512	512
MajorOperatingSystemVersion	10	4

Karakteristik *malware* ditandai dengan nilai pada fitur "SizeOfCode", "SizeOfUninitializedData", "AddressOfEntryPoint", "BaseOfCode", dan "BaseOfData" jauh lebih tinggi dari *file non-malware*. Sebaliknya, *file non-malware* memiliki nilai lebih tinggi pada fitur "SizeOfInitializedData" dibandingkan *file malware*. Fitur-fitur lainnya tidak menunjukkan perbedaan yang signifikan.

4 Kesimpulan

Website yang dibangun menggunakan *framework* Flask dapat dibuat dengan mudah, menghasilkan website yang dinamis dan interaktif. Navigasi berfungsi dengan baik, memungkinkan pengguna diarahkan ke halaman yang sesuai dengan aktivitas mereka. Integrasi Flask dengan *machine learning* untuk analisis file juga berjalan lancar, terbukti dengan kemampuannya mendeteksi file *malware* secara akurat. Analisis menunjukkan suatu *file* termasuk berbahaya dan tidak berbahaya dianalisa melalui fitur 'SizeOfCode', 'SizeOfUninitializedData', 'BaseOfCode', 'AddressOfEntryPoint', 'BaseOfData', dan 'SizeOfInitializedData'. Website ini diharapkan dapat bermanfaat bagi

banyak pengguna untuk menghindari kejahatan siber di masa depan.

References

- Agustin, Wirta, Unang Rio, Rometdo Muzawi, Torkis Nasution, and Dwi Haryono. n.d. "Penguatan Pengelolaan Website Desa Untuk Meningkatkan Layanan Administrasi Kependudukan Di Desa Pasir Baru Rokan Hulu."
- Al-Fadhillah Wafiq, Muhammad, Terra Purnama Putra, Renaldy Permana Sundawa, Meakhel Gunawan, and Ronggo Alit. 2023. "Redesain Website SMKN 2 Surabaya Untuk Meningkatkan Tampilan Dan Fungsionalitas Dengan Bootstrap." *Jurnal Ilmu Teknik* 1(2):19–29.
- Ayudhitama, Annisa Putri, and Utomo Pujiyanto. 2020. "Analisa Kualitas Dan Usability Berdasarkan Persepsi Pada Website Shopee." *Jurnal Informatika Polinema* 6(1):61–70. doi: 10.33795/jip.v6i1.275.
- Budianto, Andre Josiah, and Pratyaksa Ocsa Nugraha Saian. 2023. "Pengembangan Modul Inventory Management Pada Aplikasi Master Distribution Centre System Menggunakan Framework Flask Di PT XYZ." *Jurnal JTik (Jurnal Teknologi Informasi Dan Komunikasi)* 7(2):201–7. doi: 10.35870/jtik.v7i2.714.
- Guerra-manzanares, Alejandro. 2024. "Computers & Security Machine Learning for Android Malware Detection : Mission Accomplished ? A Comprehensive Review of Open Challenges and Future Perspectives." *Computers & Security* 138(December 2023):103654. doi: 10.1016/j.cose.2023.103654.
- Islam, Rejwana, Moinul Islam Sayed, Sajal Saha, Mohammad Jamal Hossain, and Md Abdul Masud. 2023. "Android Malware Classification Using Optimum Feature Selection and Ensemble Machine Learning." *Internet of Things and Cyber-Physical Systems* 3(October 2022):100–111. doi: 10.1016/j.iotcps.2023.03.001.
- Mobarak, Md Hosne, Mariam Akter Mimona, Md Aminul Islam, Nayem Hossain, Fatema Tuz Zohura, Ibnul Imtiaz, and Md Israfil Hossain Rimon. 2023. "Scope of Machine Learning in Materials Research—A Review." *Applied Surface Science Advances* 18(August):100523. doi: 10.1016/j.apsadv.2023.100523.
- Ningtyas, Dinda Fitri, and Nina Setiyawati. 2021. "Implementasi Flask Framework Pada Pembangunan Aplikasi Purchasing Approval Request." *Jurnal Janitra Informatika Dan Sistem Informasi* 1(1):19–34. doi: 10.25008/janitra.v1i1.120.
- Pamungkas, Ridho. 2019. "Analisis Pengaruh Teknik Responsive Web Design (RWD) Terhadap Kualitas Website Dengan Metode PIECES." *Seminar Nasional Teknologi Informasi Dan Komunikasi* 149–54.
- Prasetyo, Stefanus Eko, Haeruddin Haeruddin, and Kelvin Ariesryo. 2024. "Website Security System from Denial of Service Attacks, SQL Injection, Cross Site Scripting Using Web Application Firewall." *Antivirus : Jurnal Ilmiah Teknik Informatika* 18(1):27–36. doi: 10.35457/antivirus.v18i1.3339.
- Rico, Valentino, Christ Rudianto, and Rocky Tanaamah. 2024. "Perancangan Website Kembar Jaya Steel Dengan Menggunakan Metode Prototype." *Jurnal Ilmiah Komputasi* 23(1):139–46. doi: 10.32409/jikstik.23.1.3575.
- Santoso, Bartolomeus Bima, and Pratyaksa Ocsa Nugraha Saian. 2023. "Implementasi Flask Framework Pada Development Modul Reporting Aplikasi Sistem Informasi Helpdesk Di PT.XYZ." *Jurnal JTik (Jurnal Teknologi Informasi Dan Komunikasi)* 7(2):217–26. doi: 10.35870/jtik.v7i2.718.
- Tambuwun, T. F., R. Sengkey, and Y. D. Y. Rindengan. 2017. "Perancangan Aplikasi Web Berbasis Usabilitas."

