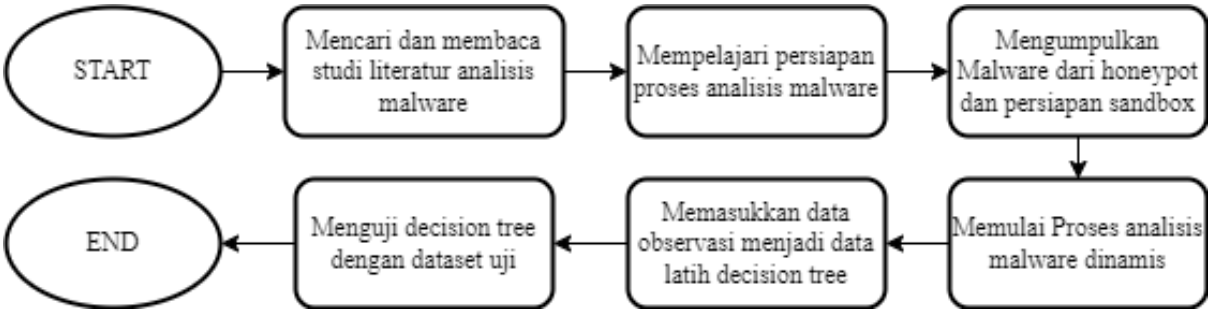
BAB III
PENELITIAN

3.1. Desain penelitian

Desain penelitian merupakan tahapan yang akan diambil oleh penulis untuk melakukan penelitian. Desain penelitian yang digunakan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

Penelitian ini berfokus pada penerapan proses *digital forensics*, khususnya melalui analisis dinamis, untuk mengidentifikasi dan mengklasifikasi *malware* yang terlibat dalam serangan *phishing* berbasis dokumen. Inti dari penelitian ini adalah bagaimana analisis dinamis dapat digunakan secara efektif untuk memahami perilaku malware dan, pada akhirnya, mengklasifikasikannya dengan bantuan *decision tree*.

Penelitian dimulai dengan mencari dan membaca studi literatur terkait analisis malware, yang memberikan dasar teori untuk proses *digital forensics* dan metode analisis yang digunakan. Literatur ini mencakup berbagai pendekatan dalam menganalisis *malware*, baik secara statis maupun dinamis, serta studi kasus tentang penggunaan analisis dinamis dalam lingkungan forensik digital.

Selanjutnya, penelitian ini masuk ke tahap persiapan proses analisis *malware*, yang mencakup pemahaman tentang bagaimana membangun lingkungan *sandbox* yang aman dan cara mengumpulkan *malware* menggunakan *honeypot*. Tahap ini sangat penting untuk memastikan bahwa *malware* dapat dianalisis dalam kondisi yang terkontrol, memungkinkan pengumpulan data yang akurat tentang perilaku *malware*.

Setelah persiapan, penelitian memasuki fase inti, yaitu memulai proses analisis *malware* dinamis. Dalam tahap ini, *malware* yang telah terkumpul dianalisis secara menyeluruh menggunakan metode analisis dinamis. Tujuannya adalah untuk memantau perilaku malware di lingkungan yang menyerupai kondisi operasional sebenarnya. Data yang diperoleh dari analisis ini meliputi berbagai indikator yang dapat digunakan untuk mengidentifikasi dan mengklasifikasi *malware* secara langsung.

Setelah proses analisis dinamis menghasilkan data observasi yang diperlukan, langkah berikutnya adalah memasukkan data tersebut ke dalam model *decision tree* sebagai data latih. Data ini kemudian digunakan untuk melatih model *decision tree*, yang akan membantu dalam proses klasifikasi *malware*. *Decision tree* berfungsi sebagai alat bantu dalam mengkategorikan malware berdasarkan karakteristik dinamis yang telah diidentifikasi selama analisis.

Akhirnya, model *decision tree* yang telah dilatih diuji dengan dataset uji untuk mengevaluasi kemampuannya dalam mengklasifikasikan *malware* dengan tepat. Hasil dari pengujian ini memberikan *insight* penting mengenai efektivitas pendekatan analisis dinamis yang diintegrasikan dengan model *decision tree*, serta kontribusinya terhadap proses digital forensics dalam konteks keamanan siber.

3.2. Alat Penelitian

Berikut adalah alat-alat dan perangkat lunak yang akan digunakan untuk membuat lab analisis *malware* selama proses melakukan penelitian:

1. Komputer
    1. Processor : 12th gen Intel I7-12700H 2.3GHz
    2. Memory : 16 GB RAM
    3. Storage : 1TB SSD
    4. Network : Intel Wi-Fi 6 AX201 160MHz
2. Virtualbox
    1. Versi : 7.0.6 Edition

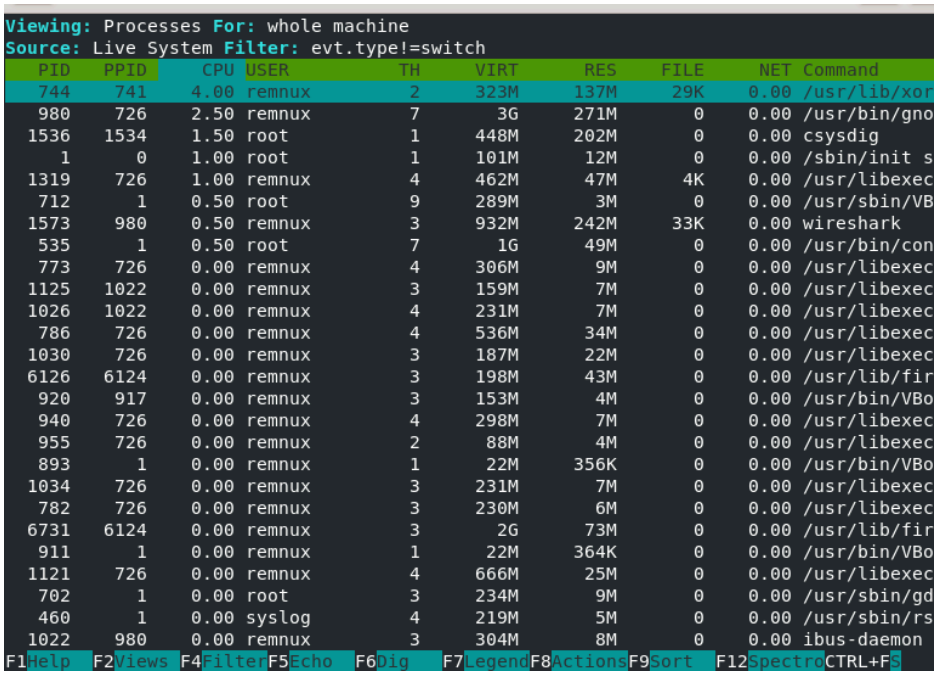
3. Remnux

    1. Versi : REMnux v7 focal

Berikut adalah perangkat lunak yang akan digunakan untuk melakukan penelitian *malware* analisis, perangkat lunak akan digunakan untuk memantau, memonitor dan mencatat cara kerja sampel yang telah dikumpulkan.

1. *Sysdig*

    Sysdig adalah sebuah perangkat lunak yang digunakan untuk penyelesaian masalah dan memonitor sistem. Perangkat lunak ini memberikan akses ke dalam aktivitas sistem dan jaringan, sehingga dapat melihat jalan pekerjaan dari *malware*. Gui sysdig dapat dilihat pada Gambar 3.2.

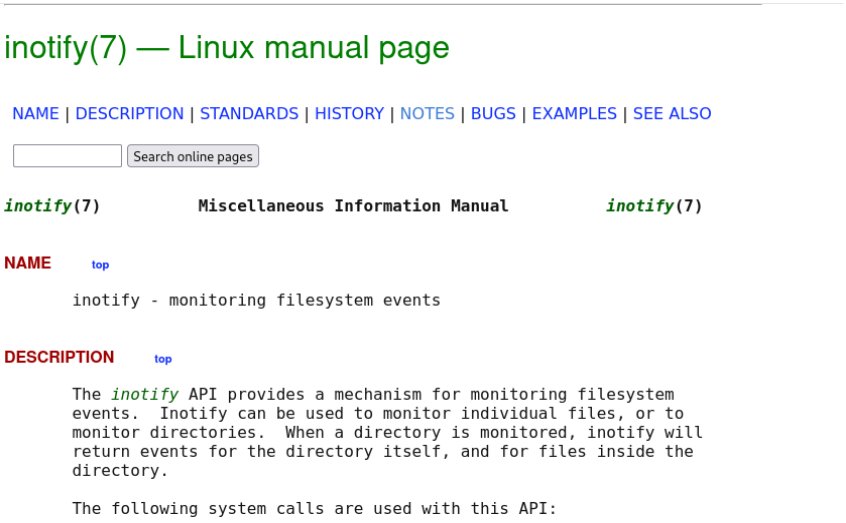


Gambar 3.2 Gui sysdig

2. *Inotify*

    inotify adalah sebuah perangkat lunak yang digunakan untuk memonitor perubahan yang dapat terjadi dalam file yang ada dalam sebuah komputer, melalui perangkat lunak ini kita dapat mengetahui apakah ada file yang dibuat,

dimodifikasi, dipindahkan atau dihapus oleh *malware*. Halaman utama dalam instalasi *inotify* dapat dilihat pada Gambar 3.3.


inotify(7) — Linux manual page

NAME | DESCRIPTION | STANDARDS | HISTORY | NOTES | BUGS | EXAMPLES | SEE ALSO

Search online pages

inotify(7) Miscellaneous Information Manual inotify(7)

NAME top

inotify - monitoring filesystem events

DESCRIPTION top

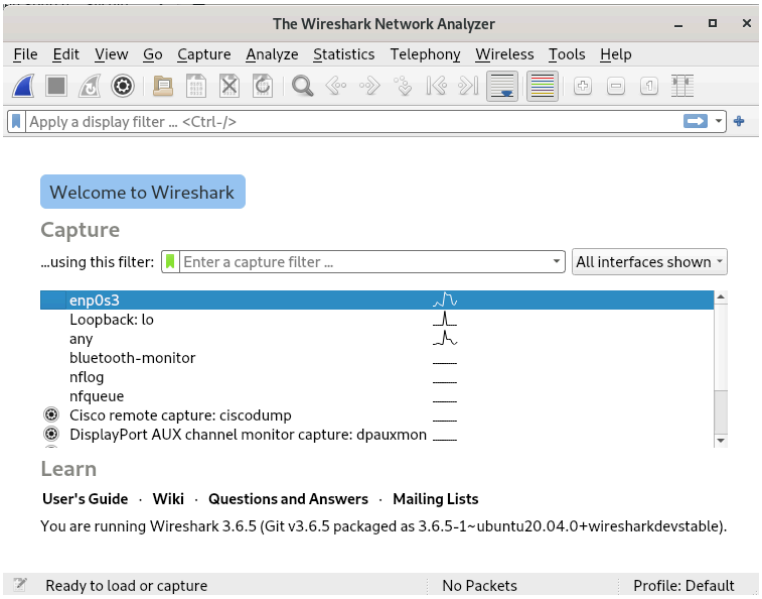
The inotify API provides a mechanism for monitoring filesystem events. Inotify can be used to monitor individual files, or to monitor directories. When a directory is monitored, inotify will return events for the directory itself, and for files inside the directory.

The following system calls are used with this API:

Gambar 3.3 Halaman manual inotify

3. *Wireshark*

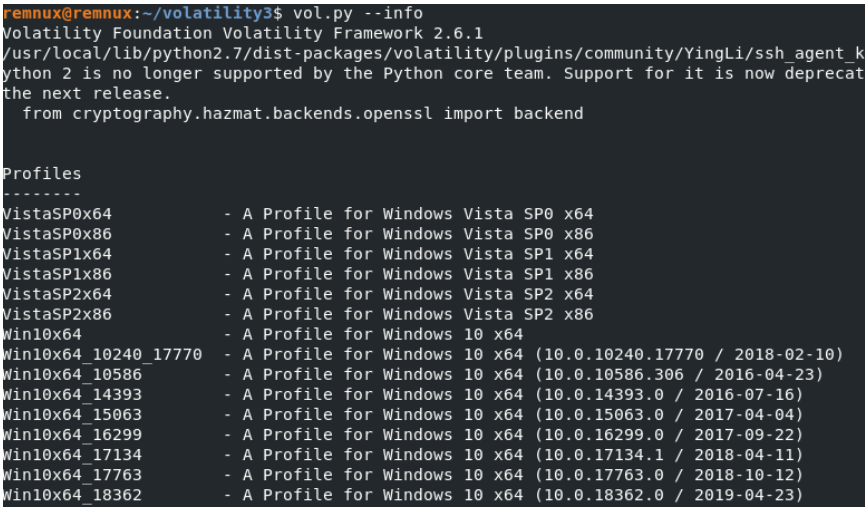
Wireshark adalah aplikasi pembantu yang digunakan untuk melakukan observasi pada jaringan dalam komputer, melalui tools ini kita dapat mendeteksi pergerakan *malware* dalam jaringan. Gui dari *wireshark* yang akan digunakan dapat dilihat pada Gambar 3.4.



Gambar 3.4 Gui *wireshark*

4. *Volatility*

Volatility adalah sebuah perangkat lunak yang dapat digunakan untuk melihat sebuah *memory dump* dari sebuah proses yang sedang dijalankan. Perangkat lunak ini sering digunakan dalam pekerjaan forensik dalam mengambil informasi mengenai penggunaan memori komputer. Cli dari volatility dapat dilihat pada Gambar 3.5.



Gambar 3.5 Cli volatility

3.3. Bahan Penelitian

Pada bagian ini, akan dibahas mengenai bahan penelitian yang digunakan dalam penelitian ini. Bahan penelitian terdiri dari dua kategori dokumen dengan formatnya masing-masing, yaitu dokumen-dokumen biasa yang menjadi contoh representatif dari dokumen umum yang digunakan oleh organisasi, serta dokumen-dokumen yang telah terinfeksi oleh *malware* yang biasa digunakan dalam serangan *phishing*. Melalui kedua kategori dokumen ini, diharapkan dapat diperoleh pemahaman yang lebih mendalam tentang cara mengidentifikasi dan menganalisis *malware* berbentuk dokumen dalam konteks serangan *phishing*.

Bahan yang akan disimulasikan di dalam komputer dan dijadikan bahan observasi adalah sebagai berikut:

1. Dokumen yang akan menjadi representatif, yaitu dokumen yang dipakai organisasi secara umum seperti dokumen *pdf*, *docx*, *xlsx* dan *pptx*. Dokumen-dokumen ini adalah contoh representatif atas apa yang terjadi ketika proses dokumen pada umumnya itu digunakan, analisis dari proses dokumen ini akan menjadi pembanding untuk proses analisis dokumen yang berisikan *malware*.

2. Dokumen yang berisi *Malware* berbentuk dokumen yang didapat melalui serangan *phishing*, berupa bahan-bahan berikut:
    1. File *pdf*

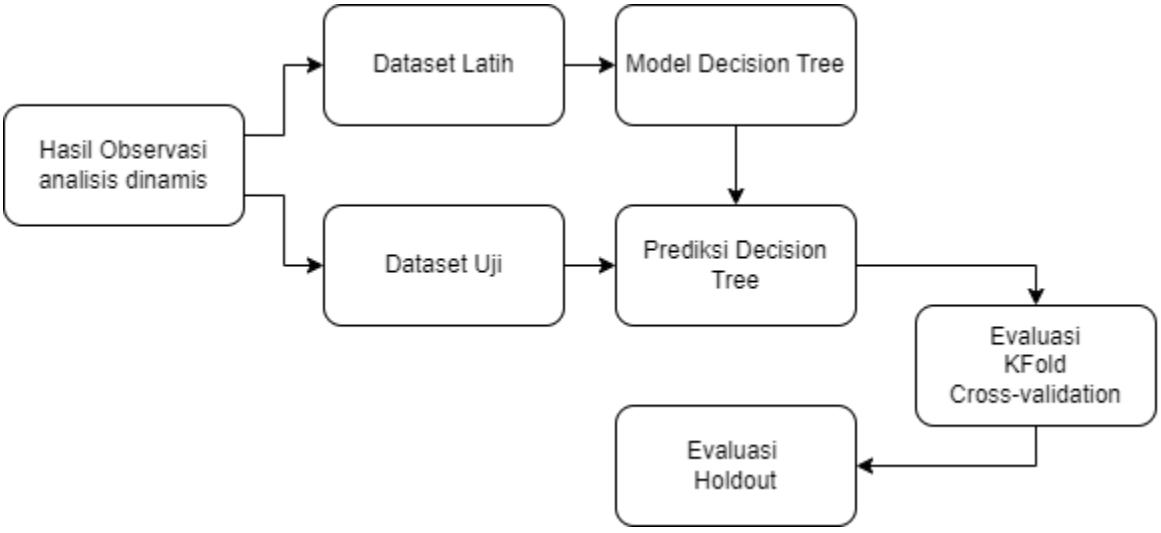
        Dalam penelitian ini, dokumen *pdf* yang digunakan adalah sebuah dokumen berisikan *malware* yang ditemukan pada tahun 2022 dari database kumpulan *malware* daring yang didapatkan oleh perusahaan.
    2. File *docx*

        Dalam penelitian ini, dokumen *docx* yang digunakan adalah sebuah dokumen berisikan *malware* yang ditemukan pada tahun 2022 dari database kumpulan *malware* daring yang didapatkan oleh perusahaan.
    3. File *xlsx*

        Dalam penelitian ini, dokumen *xlsx* yang digunakan adalah sebuah dokumen berisikan *malware* yang ditemukan pada tahun 2022 dari database kumpulan *malware* daring yang didapatkan oleh perusahaan.
    4. File *pptx*

        Dalam penelitian ini, dokumen *pptx* yang digunakan adalah sebuah dokumen berisikan *malware* yang ditemukan pada tahun 2022 dari database kumpulan *malware* daring yang didapatkan oleh perusahaan.

3.4. Metodologi Decision Tree



Gambar 3.6 Metodologi Decision Tree

Gambar 3.6 mencantumkan metodologi pada *decision tree* yang akan digunakan dalam penelitian ini, metodologi ini terdiri dari beberapa tahapan utama.

Tahapan pertama adalah pengumpulan data hasil observasi dari analisis dinamis malware. Data yang terkumpul dari proses ini mencakup berbagai informasi yang diperoleh dari aktivitas malware dalam lingkungan terisolasi. Data ini kemudian diproses dan dibagi menjadi dua bagian, yaitu dataset latih dan dataset uji, yang masing-masing akan digunakan untuk membangun dan menguji model.

Pada tahap berikutnya, dataset latih digunakan untuk membangun model *decision tree*. Model ini dilatih dengan menggunakan data yang telah diolah, dengan tujuan untuk mengenali pola dan fitur yang ada dalam dataset tersebut. Model *decision tree* yang terbentuk kemudian digunakan untuk memprediksi kelas dari sampel malware berdasarkan fitur-fitur yang telah diidentifikasi sebelumnya.

Setelah model terbentuk, tahap pengujian dilakukan dengan menggunakan dataset uji. Dataset uji ini terdiri dari data yang belum pernah dilihat oleh model, sehingga memungkinkan untuk menguji performa model dalam memprediksi data baru. Hasil

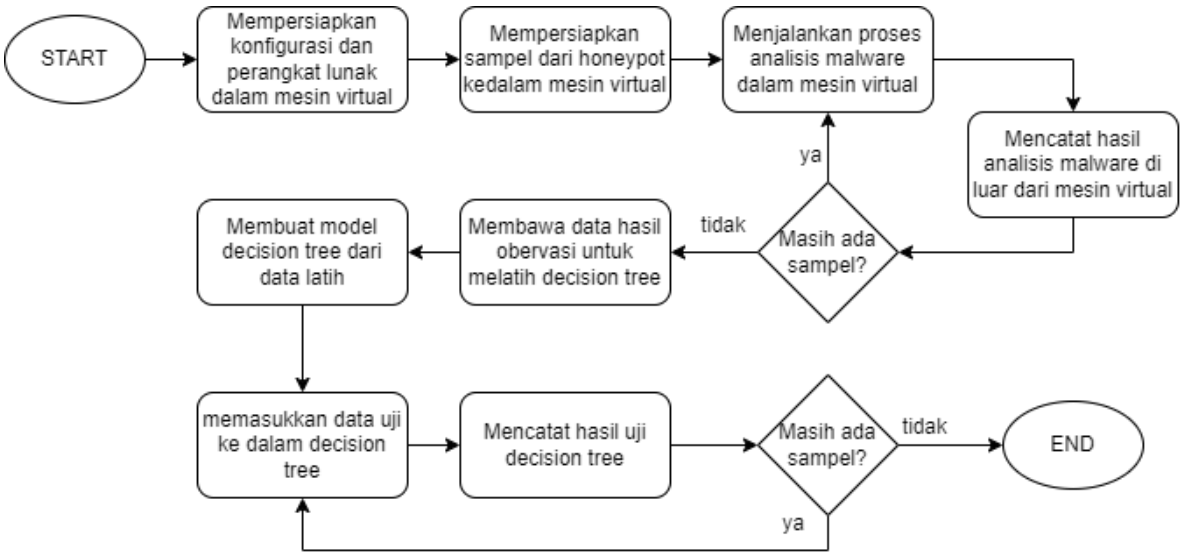
prediksi yang dihasilkan oleh model kemudian dievaluasi untuk menilai akurasi dan kinerjanya.

Evaluasi model dilakukan dengan dua metode utama, yaitu metode *holdout* dan *KFold cross-validation*. Metode *holdout* bertujuan untuk mengukur akurasi dan *AUC* (*Area Under Curve*) dari model dengan membandingkan hasil prediksi terhadap data aktual dalam dataset uji. Sementara itu, metode *KFold cross-validation* digunakan untuk memastikan stabilitas dan kemampuan generalisasi model. *KFold cross-validation* membagi data menjadi beberapa subset dan melakukan pengujian secara berulang, yang membantu dalam mengidentifikasi apakah model dapat memberikan performa yang konsisten di berbagai pembagian data.

3.5. Diagram Alir

Bagian ini akan menjelaskan diagram alir yang digunakan dalam proses analisis *malware* berbentuk dokumen dalam konteks serangan *phishing*. Diagram alir ini dirancang untuk menggambarkan langkah-langkah metodologi yang digunakan dalam analisis.

Dengan menggunakan diagram alir pada Gambar 3.7, diharapkan pembaca dapat memahami secara visual bagaimana penelitian ini dilaksanakan dan bagaimana setiap langkah berkontribusi terhadap tujuan penelitian yang telah ditetapkan. Diagram alir juga membantu dalam memberikan gambaran yang jelas tentang alur kerja analisis *malware* berbentuk dokumen yang dilakukan dalam penelitian ini.



Gambar 3.7 Diagram Alir Penelitian

Penelitian ini dimulai dengan tahap persiapan, di mana konfigurasi dan perangkat lunak yang diperlukan diatur dalam sebuah mesin virtual. Lingkungan ini dipersiapkan secara hati-hati untuk memastikan bahwa analisis dapat dilakukan dengan aman dan terkendali. Setelah lingkungan siap, sampel malware yang telah dikumpulkan dari honeypot dimasukkan ke dalam mesin virtual. Langkah ini memastikan bahwa setiap analisis dilakukan dalam kondisi yang terisolasi, mengurangi risiko penyebaran malware di luar lingkungan uji.

Proses analisis malware dilakukan secara dinamis di dalam mesin virtual. Dalam tahap ini, perilaku malware dipantau dan diamati secara teliti untuk mengumpulkan data penting yang akan digunakan dalam langkah-langkah selanjutnya. Data yang diperoleh dari proses ini kemudian dicatat secara sistematis di luar mesin virtual, menyediakan basis data yang diperlukan untuk membangun model decision tree.

Tahap berikutnya adalah pembuatan model decision tree berdasarkan data latih yang telah dikumpulkan dari analisis sebelumnya. Decision tree ini bertujuan untuk mengklasifikasikan jenis-jenis malware berdasarkan perilaku yang telah diamati selama proses analisis dinamis. Setelah model decision tree dibentuk, data hasil observasi

digunakan untuk melatih model tersebut, sehingga decision tree mampu melakukan klasifikasi dengan akurat.

Setelah model decision tree dilatih, langkah selanjutnya adalah memasukkan data uji ke dalam model untuk mengevaluasi performanya. Hasil pengujian decision tree ini dicatat dengan tujuan untuk menganalisis tingkat keakuratan model dalam mengklasifikasikan jenis malware. Jika masih ada sampel malware lainnya yang belum dianalisis, maka proses ini diulangi mulai dari tahap analisis malware hingga pencatatan hasil uji decision tree.