PENERAPAAN RANDOM FOREST CLASSIFIER UNTUK DETEKSI PDF MALWARE PADA LAYANAN AGREGATOR GARBA RUJUKAN DIGITAL (GARUDA) KEMENDIKBUD DIKTI

SKRIPSI

OLEH:

ALFIAH NUR FATMAWATI

09011181823131

JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA

2024

Penerapan *Random Forest Classifier* untuk Deteksi PDF *Malware* pada Layanan Agregator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

OLEH
Alfiah Nur Fatmawati
09011181823131

JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA

2024

LEMBAR PENGESAHAN

Penerapan *Random Forest Classifier* untuk Deteksi PDF *Malware* pada Layanan Agregator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Satu Syarat

Memperoleh Sarjana Komputer

OLEH

Alfiah Nur Fatmawati

09011181823131

Indralaya, 27 Januari 2024

Pembimbing 1 Tugas Akhir

Prof. Deris Stiawan, M.T.,Ph.D
NIP. 197806172006041002

Pembimbing II Tugas Akhir

Nurul Afifah M, Kom.
NIP. 199211102023212049

Mengetahui ,

Ketua Jurusan Sistem Komputer

Dr.Ir. Sukemi,.M.T.
NIP.196612032006041001

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada:

Hari : **Jum'at**

Tanggal : **12 Januari 2024**

Tim Penguji :

1. **Ketua** : **Dr. Firdaus, M.Kom.**
2. **Sekretaris** : **Muhammad Ali Buchari, S.Kom., M.T.**
3. **Penguji** : **Rossi Passarella, M.Eng.**
4. **Pembimbing I** : **Prof. Deris Stiawan, M.T., Ph.D.**
5. **Pembimbing II** : **Nurul Afifah, M.Kom.**

Mengetahui,

Ketua Jurusan Sistem Komputer

Dr.Ir. Sukemi,.M.T

NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Alfiah Nur Fatmawati

NIM : 09011181823131

Judul : Penerapan *Random Forest Classifier* untuk Deteksi PDF *Malware* pada Layanan Agregator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti

Hasil Pengecekan Software *iThentiate/Turnitin* : 1%

Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila di temukan unsur penjiplakan atau plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya

Demikian Laporan ini saya buat dalam keadaan sadar dan tanpa paksaan dari siapapun.

Indralava, Januari 2024

Alfiah Nur Fatmawati

NIM. 09011181823131

HALAMAN PERSEMBAHAN

Bismillahirahmanirahim Tugas Akhir ini saya persembahkan untuk Bangsa dan Negara, kedua untuk Universitas Sriwijaya selaku rumah bagi saya dalam menimba ilmu. Saya ucapkan rasa Puji Syukur yang teramat besar kepada Allah swt, yang telah memberikan saya rezeki sehingga saya dapat berkuliah di Universitas Sriwijaya, saya bisa menjadi bagian dari salah satu manusia yang diberkati dari banyaknya orang yang ingin berkuliah disini, Ketiga saya ucapkan terimakaksih kepada kedua orang tua saya yang telah medoakan saya dan juga sabar atas segala hal yang saya lakukan sampai sejauh ini. Keluhan rasa sedih cinta dan doa semua membaur menjadi satu.

Hasbunallah Wa ni'mal Wakil

"Cukuplah Allah menjadi Penolong kami dan Allah adalah sebaik-baik Pelindung"

[Ali 'Imran, 3: 173]

"Optimis yakinkan diri pasti bisa"

(Alfiah Nur Fatmawati)

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Pertama-tama kami panjatkan rasa puja dan puji syukur Alhamdulillah penulis panjatkan atas kehadirat Allah SWT yang telah memberikan karunia dan rahmat-Nya, sehingga penulis dapat menyelesaikan penulisan Tugas Akhir ini yang berjudul **"Penerapan *Random Forest Classifier* untuk deteksi PDF *Malware* pada Layanan Agregrator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti)".**

Dalam laporan bertujuan untuk mengetahui hasil Performasi dan Evaluasi pada PDF Malware. Dalam penelitian ini kami menggunakan metode *Randome Forest Classifier* sebagai tools nya. Dimana hal ini meminimalisir terjadinya infeksi Malware. Tentang karakteristik malware berdasarkan hasil Analisa bahwa terdapat beberapa signature yang dapat di simpulkan. Penulis dengan ini sangat penuh harap agar tulisan ini dapat bermanfaat bagi banyak orang.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada beberapa pihak atas ide dan saran serta bantuannya dalam menyelesaikan penulisan Tugas Akhir ini. Oleh karena itu, penulis ingin mengucapkan rasa syukurkepada Allah SWT dan terimakasih kepada yang terhormat :

1. Allah SWT, yang telah memberikan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan penulisan Proposal Tugas Akhir ini dengan baik dan lancar.
2. Orang tua saya tercinta (Yon Sahono dan Nur Asiah) yang telah membesarkan saya dengan cinta dan kasih saying, dukungan dan doa serta keberkahan yang begitu sangat luar biasa sehingga berada di titik saat ini.
3. Kepada keluarga penulis, yang tersayang Nenek (Mujinah) dan kakek (Muhawi) saya yang telah wafat semoga allah tempatkan di sisi terbaiknya
4. Adik saya tercinta (Fachri Jamil) kebahagiaan menyertaimu terimakasih atas doa dan dukunganya

5. Prof. DR.Erwin, S.Si.,M.Si. selaku Dekan Fakultas Ilmu KomputerUniversitas Sriwijaya.
6. Bapak Dr. Ir. Sukemi, M.T., selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
7. Prof. Deris Stiawan, M.T., PH.D., IPU,.ASEAN-Eng selaku Dosen Pembimbing Tugas Akhir 1 yang telah berkenan meluangkan waktunya guna membimbing, memberikan saran dan motivasi serta bimbingan terbaik untuk penulis dalam menyelesaikan Tugas Akhir ini.
8. Mbak Nurul Afifah, M.Kom. selaku Pembimbing Tugas Akhir II saya sangat berterimakasih dan bersyukur atas segala kebaikan dan bimbingan, support dari mba nurul dan berkenan membimbing saya.
9. Bapak Rossi Passarella, M.eng. selaku Pembimbing Akademik Jurusan Sistem Komputer.
10 Kak Yopi selaku admin Jurusan Sistem Komputer yang telah membantu mengurus berkas-berkas yang di perlukan.
11. Teman saya Nata Arista yang selalu mendukung saya
12. Menyadari bahwa penulis laporan ini masih sangat jauh dari kata sempurna. Untuk itu kritik dan saran yang membangun sangatlah diharapkan penulis. Akhir kata penulis berharap, semoga proposal tugas akhir ini bermanfaat dan berguna bagikhalayak.

Wassalamu'alaikum Wr. Wb.

Indralaya, Januari 2024
Penulis,

Alfiah Nur Fatmawati
NIM. 09011181823131

PENERAPAN *RANDOM FOREST* CLASSIFIER UNTUK DETEKSI PDF *MALWARE* PADA LAYANAN AGREGATOR GARBA RUJUKAN DIGITAL (GARUDA) KEMENDIKBUD DIKTI

ALFIAH NUR FATMAWATI (09011181823131)

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email : alfiahnurfatmawati27@gmail.com

ABSTRAK

Garba Rujukan Digital (GARUDA) merupakan salah satu Penyedia layanan yang memberikan akses informasi dan sarana pengetahuan yang di hasilkan dari akademisi penelitian yang ada di Indonesia, pada saat ini Layanan GARUDA tersebut memberikan sebuah data berbentuk file *Portable Document Format* (PDF) yang mana PDF tersebut umunya di gunakan untuk layanan berbagi informasi secara cepat dan mudah, namun banyak oknum yang tidak bertanggung jawab justru memanfaatkan hal tersebut untuk melakukan kejahatan digital *Cyber Crime*. GARUDA memberikan data sebesar 10000 PDF Malware yang akan di gunakan sebagai *Dataset* dalam penelitian ini. Dari dataset yang di miliki maka di peroleh data benign 8900, data non-malpdf 196, dan data mal-pdf 6. Virus Total dan PDFiD di gunakan Pada untuk Analisis Statis pada dataset tersebut. Penelitian ini bertujuan untuk menghasilkan Performasi dan evaluasi dari kenerja Algoritma Random Forest dalam dataset Imbalance pada dataset PDF malware GARUDA. Pada proses penyeimbangan data di lakukan dengan dua pendekatan yaitu menggunakan *Over-sampling* dengan teknik SMOTE dan Under-sampling serta penggunaan *K-Fold*. Penerapan *Random Forest Classifier* di hasilkan akurasi dengan tingkat 86,22%, Precision dengan 79,55% , Recall dengan 78,98% dan F1-Score dengan 79,26%.

Kata Kunci : *Random Forest Classifier,* PDF *Malware, Virus Total,* PDFiD, *Synthetic Minority Over-sampling Technique SMOTE*

Pembimbing I

Prof. Deris Stiawan, M.T., Ph.D
NIP. 197806172006041002

Pembimbing II

Nurul Afifah, M.Kom
NIP. 199211102023212049

Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T
NIP. 196612032006041001

APPLICATION RANDOM FOREST CLASSIFIER FOR DETECTION PDF MALWARE ON AGREGATOR GARBA RUJUKAN DIGITAL SERVICE (GARUDA) KEMENDIKBUD DIKTI

ALFIAH NUR FATMAWATI (09011181823131)

Computer Engineering Department, Computer Science Faculty, Sriwijaya University

Email : alfiahnurfatmawati27@gmail.com

ABSTRACT

Garba Rujukan Digital (GARUDA) is a Service Provider whice provides information and means whice has been produced by research academics in Indonesia. Currently the service (GARUDA) Provides data in file room . Portable Document Format (PDF). Whice PDF is ussually for fast and easy information sharing service. Howaver many people are irresponsible and actually tak and advantage of this to commit digital crime Cyber Crime GARUDA have data a big and give a 10000 PDF Malware is used for dataset in research. Dataset in mine then get it data benign 8900, data non-pdf 196, and data mal-pdf 6. Virus Total and PDFiD in used for analysis statis to dataset. This research arms to produce performance and evaluation of the performance of the Random Forest algorithm in the Imbalance dataset on the GARUDA malware PDF dataset. The data Balancing process is carried out using two approaches, namely using Over-sampling with the SMOTE technique and Under-sampling as well as using K-Fold. The application of the Random Forest Classifier resulted in an accuracy rate of 86,22%, precision with 79,55%, recall with 78,98% and F1-Score with 79,26%

Key Word *: Random Forest Classifier , PDF Malware, Virus Total, PDFiD, Synthetic Minority Over-sampling Technique (SMOTE)*

First Supervisor

Prof. Deris Stiawan, M.T., Ph.D
NIP. 197806172006041002

Second Supervisor

Nurul Afifah, M.Kom
NIP. 199211102023212049

Head Of Computer Engineering Department

Dr. Ir. Sukemi, M.T
NIP. 196612032006041001

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I

PENDAHULUAN

1.1 Latar Belakang

Portable Document Format yang biasanya di kenal PDF yaitu sistem format berkas yang di buat oleh adobe sytem pada tahun 1993 yang di gunakan sebagai pertukaran dokumen digital. Format PDF di manfaatkan untuk mempresentasikan dokumen dua dimensi yang meliputi grafik vektor dua dimensi ,huruf,teks,citra. PDF di kenalkan pertama kali di publikasikan pada tahun 1993, pada masa itu penggunaan format PDF relatif masih rendah.[1] Tidak dapat dipungkiri bahwa, seiring dengan kemajuan teknologi, masyarakat kini mengandalkan teknologi untuk menjalankan tugas sehari-hari maupun untuk terlibat dalam bidang politik, sosial, dan akademik [1]. Teknologi dimanfaatkan dalam berbagai macam hal seperti pada dunia Pendidikan, teknologi digunakan sebagai media belajar, untuk memahami perkembangan dunia digital penggunaan nya dengan cara memanfaatkan *e-book* , *e-learning* maupun pemanfaatan *Website* pemanfaatan dari teknologi yang berkembang ini tentunya memberikan keuntungan bagi kita yang mana hidup pada zaman modern yang begitu banyak memanfaatkan sarana teknologi digital [2].

Malicious Software atau yang biasa di sebut *malware* yaitu perangkat lunak secara eksplisit yang di desain untuk menjalankan berbagai macam perusak maupun aktifivas yang berbahaya bagi perangkat lunak lainya seperti *spyware, trojan, eksploit maupun virus*[3].

Oleh karena itu, untuk memastikan apakah aplikasi yang terdeteksi adalah malware dan untuk mengidentifikasi jenis malware, analisis dan deteksi merupakan langkah penting dalam menentukan potensi konsekuensi dari eksekusi sistem berbahaya. Mengenai hal-hal kebenaran yang dapat kita ketahui dari Malware yang telah di jelaskan pentingnya di lakukan metode Analisa agar mengetahui jenis-jenis dari serangan malware agar kita dapat mengetahui ciri atas malware itu sendiri.[2]

Garba Rujukan Digital yang di singkat GARUDA, merupakan wadah yang di gunakan sebagai tempat berkumpulnya informasi dan sarana pengetahuan yang menaungi sumber informasi yang melingkupi banyak aspek karya ilmiah yang ada di negara Indonesia, yang mana hal tersebut bertujuan untuk mengelolah dan mengakses karya ilmiah dengan lengkap dan mudah, aspek tersebut meliputi komputer, matematika dan prilaku[3].

Dataset *Imbalance* memungkinkan akan mengalami penurunan kesetabilan. Perolehan hasil yang di dapatkan lebih dominan akan memberikan lebih banyak pada mayoritas kelas. Dalam beberapa hal seperti pada *Multiclass classification*. Imbalance data akan memberikan representasi data sehingga mengakibatkan data yang lemah akan di acuhkan. Near Miss dan SMOTE yang merupakan kepanjangan dari *Syntetic Minorty Oversampling Technique* adalah metode undersampling dan Oversampling di gunakan oleh banyak orang untuk menghadapi masalah seperti dataset imbalance.

Randome Forest Classifier[3] adalah metode asembling yang di gunakan sebagai klasifikasi dan regresi dan tugas lainya yang beroperasi dengan membangun keputusan untuk pelatih dalam klasifikasi yang beroperasi dengan membangun banyak pohon keputusan pada waktu pelatihan untuk tugas klasifikasi hasil hutan acak untuk regresi prediksi rata-rata dari masing-masing pohon atau rata-rata dari masing-masing regresi untuk regresi masing-masing pohon di kembalikan [1][3][4].

Kinerja algoritma *Random Forest* di bandingkan dengan *Naive Bayes dan K-Nerest* setelah melalukan proses klasifikasi dengan data set yang telah di kumpulkan menunjukkan bahawa *Random Forest* memiliki akurasi dan daya ingat yang lebih tinggi di antara metode yang lain. Hal ini menunjukkan lompatan akurasi dibandingkan penelitian

Penelitian dalam hal ini akan berkonsentrasi pada sejumlah contoh yang telah kami kumpulkan, seperti file malware PDF. Dari 10.000 kumpulan data yang diambil, hanya 197 yang berbahaya; fitur kumpulan data ini belum diketahui saat ini, oleh karena itu diperlukan penyelidikan lebih lanjut.

Analisis dan deteksi ini dilakukan untuk mencegah berbagai ancaman dan serangan yang dapat merugikan korbannya. Kerugian yang dapat ditimbulkan antara lain pencurian informasi secara kasar, peretasan, dan masuknya virus berbahaya ke

dalam perangkat pengguna.

Dengan memberikan konteks, menjadi jelas bahwa selain manfaat yang bisa kita rasakan, ada juga kelemahan yang harus kita waspadai. Dengan pengetahuan yang kita miliki, kita dapat menangkis serangan virus dengan lebih efektif. Oleh karena itu, kami mempunyai harapan yang besar agar penelitian ini dapat membantu masyarakat dan memberikan pengaruh yang baik.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah di jabarkan maka di perolehlah perumusan sebagai berikut:

1. Bagaimana teknik yang digunakan dalam mengekstrak Raw PDF dari PDF Malware Layanan Agregator GARUDA menjadi dataset.
2. Bagaimana cara penerapan untuk mengklasifikasi PDF Malware berupa *mal-pdf,benign, mal-html* menggunakan Algoritma *Random Forest Classifier*
3. Bagaimana pengaruh hasil performasi dan evaluasi dari kinerja algoritma *Random Forest Classifier* dalam dataset Imbalance pada dataset PDF Malware Layanan Agregator GARUDA.

1.3. Batasan Masalah

Batasan masalah tersebut antara lain merupakan batasan yang penulis miliki saat ini, yang bertujuan agar pembahasan tetap pada topik.

1. Data set untuk penelitian ini berasal dari PDF Malware di agregator garba rujukan digital (GARUDA) kemdikbud Dikti berjumlah 10.000

2. Menganalisa karakteristik PDF Malware hanya menggunakan metode yang diusulkan yaitu metode *Random Forest Classifier* pada PDF malware di agregator GARUDA kemdikbud Dikti
3. Tidak membahas bagaimana cara masuk dan mencegah serangan Malware pada file PDF.

1.4 Tujuan

Berdasarkan hasil penelitian ini tujuan yang akan dicapai adalah sebagai berikut :

1. Memudahkan dalam mengekstrak Raw data pada PDF malware di agregator Garba Rujukan Digital (GARUDA) kemdikbud Dikti menjadi data yang di olah agar menjadi dataset.
2. Penerapan Random Forest Classifier untuk klasifikasi PDF malware berupa *mal-pdf,benign, mal-html.*
3. Pengaruh hasil performasi dan evaluasi dari kinerja algoritma Random Forest dalam dataset Imbalance pada dataset PDF malware GARUDA.

1.5 Manfaat

Adapun manfaatnya yang dapat di peroleh dari penelitian Skripsi ini sebagai berikut;

1. Kumpulan data Raw PDF malware yang telah di ekstraksi akan di ubah menjadi dataset
2. Memahami dan mampu mengidentifikasi malware berdasarkan kategorisasi PDF Malware Referensi Digital Kementerian Pendidikan dan Kebudayaan. Agregator GARUDA Garba menggunakan teknik Random Forest Classifier
3. Memperoleh hasil yang tepat dan optimal dalam mendekteksi PDF malware

1.6. Manfaat bagi kampus

Adapun manfaatnya yang dapat di peroleh dari penelitian Skripsi ini sebagai berikut;

1. Memahami permasalahan mengenai penyerangan Malware.
2. Menganalisa karakteristik malware yang ada pada PDF Malware di agregator GARUDA kemdikbud Dikti dengan Metode *Random Forest Classifier*
3. Menambah wawasan dan pengetahuan yang bisa digunakan sebagai acuan dalam Deteksi Malware pada suatu instansi, serta kampus yang mana bisa bermanfaat sebagai bahan ajar agar ilmu secara teori.
4. Tidak membahas tentang bagaimana mencegah serangan Malware.

1.7 Metode Penelitian

Metedologi yang akan digunakan dalam tugas akhir ini akan melewati tahapan sebagai berikut :

1. Metode studi pustaka/literatur

   Pada tahap ini melakukan literatur riview yang berkaitan tentang Malware menggunakan metode Random Forest Classifier, serta untuk menyelesaikannya dibantu melalui jurnal internasional, buku dan internet yang berkaitan dengan tugas akhir.

2. Metode Konsultasi

   Metode konsultasi pada penelitian ini melakukan konsultasi kepada Dosen pembimbing serta semua orang yang mempunyai pengetahuan dan wawasan pada saat terdapat permasalahan dalam melakukan tugas akhir.

3. Metode Pengumpulan Data

   Pada metode kali ini, database yang digunakan Dataset PDF Malware GARUDA sebanyak 10.000 pdf yang di ektraksi menggunakan Virus taotal.

4. Metode Observasi

   Pada metode kali ini melakukan pengamatan dan pencatatan dan pengumpulan terhadap data-data yang diproleh.

5. Metode Perancangana Software

Dilakukan perancangan pada tahap ini agar dapat memudahkan menganalisa serta deteksi malware pada PDF malware.

6. Metode Pengujian

Tahapan selanjutnya adalah pengujian pada Malware apakah malware tersebut berbahaya atau tidak karena pada malware yang telah di ekstraksi hanya terdapat sedikit malware yang sesungguhnya

7. Metode Analisa dan Kesimpulan

Menganalisa hasil dari pengujian metode sebelumnya telah di lakukan untuk dapat memahami karakteristik serta yang di timbulkan dari malware agar dilakukannya pengembangan untuk penelitian kedepannya..

1.8 Lingkungan perangkat keras dan perangkat lunak

Dalam tugas akhir ini perangkat lunak yang digunakan adalah Kali LINUX, Virtual Box dan Virus Total. Sedangkan perangkat keras yang digunakan adalah pc atau laptop.

1.9 Sistematika Penulisan

Sistematika pada penulisan yang digunakan dalam tugas akhir ini adalah sebagai berikut yang mana tinjauan ini di gunakan untuk mendeskripsikan subbab-bab yang tersusun. Sebagaimana berikut susunannya.

BAB I. PENDAHULUAN

Pada bab I ini akan berisi latar belakang masalah, tujuan dan manfaat serta metodologi penelitian dan sistematika penulisan mengenai Malware.

BAB II. TINJAUAN PUSTAKA

Pada bab II akan berisi dasar teori dan literatur review malware yang diteliti oleh peneliti lain dengan menggunakan metode yang beragam. Pada bab ini juga akan menjelaskan kelemahan dari metode yang digunakan pada penelitian lain.

BAB III. METODOLOGI PENELITIAN

Pada bab III akan membahas Analisis dan Deteksi Malware menggunakan metode *Random Forest Classifier*.

BAB IV. HASIL DAN ANALISA

Pada bab IV membahas proses yang serta hasil Analisa dan Pendeteksian Malware perangkat lunak menggunakan metode *Random Forest Classifier* untuk mendapatkan hasil Malware yang di butuhkan.

BAB V. KESIMPULAN DAN SARAN

Pada bab V nantinya akan berisi kesimpulan yang diambil dari bab sebelumnya mengenai hasil Analisa deteksi PDF Malware. Dan juga berisi saran yang dapat digunakan untuk perhatian pada penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Pendahuluan

Tahap selanjutnya yang di lakukan adalah mencari penelitian yang berkaitan dengan tugas akhir yang kerjakan, seoperti penelitian apa yang telah di lakukan dan apa yang di hasilkan pada penelitian sebelumnya serta dapat mengetahui metode apa saja yang di gunakan untuk menyelsaikan permasalahan yang terkait.

2.2 Penelitian Terkait

Penelitian telah memanfaatkan malware secara ekstensif. Berikut ini adalah teknik yang digunakan para peneliti:

Penelitian yang berjudul "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis" [5] Tujuan dari penelitian ini adalah untuk menentukan apakah keluaran yang dihasilkan oleh kedua metode dinamis dan statis untuk menganalisis malware Posion Ivy RAT (Remote Access Trojan) adalah sama. Kemampuan untuk mengidentifikasi malware Poison Ivy RAT merupakan fungsi dari penerapan pendekatan ini dan pemahaman metode konsep Analisis Statis [5].

Penelitian yang berjudul "Analisis malware menggunakan Metode Dynamic Analysis pada jaringan Universitas Sam Ratulangi" [2]. Tujuan dari penelitian ini adalah untuk mengetahui jenis malware yang ditemukan pada jaringan Universitas Sam Ratulangi. Cockoo Sandbox merupakan instrumen yang digunakan dalam penelitian ini, dan pendekatan yang digunakan adalah metode Analisis Dinamis. Sehingga terhindar dari serangan infeksi *malware*. Berdasarkan hasil analisa yang telah di lakukan kepada malware sehingga dapat di ketahui karakteristik *malware*, dan juga dapat di terangkan bahwasanya beberapa terdapat *string, signiture* dan perubahan *value resgistrasi* padanya. Kaitanya dalam metode ini adalah kesepahaman tentang

bagaiamana mengidentifikasi dan mengetahui karakteristik yang ada pada sebuah malware .[2]

Penelitian yang berjudul “Tools and Techniques for colletion and Analysis of Internet-of-Things malware ” DarckComet digunakan dalam penelitian ini bersama dengan pendekatan analitis dan statis. Temuan penelitian ini menunjukkan bahwa pendekatan ini dijelaskan dengan cara yang efektif dan efisien, sehingga dapat meningkatkan kinerja deteksi dan memastikan bahwa virus tersebut dihilangkan.[6]

Penelitian dengan Tema “Random Forest Classifier” [7] Pada penelitian tersebut tentang membahas Analisa malware Trojan dengan menggunakan Metode Dinamis dan Statis pada operasi sistem windows. Kaitanya dengan penelitian karena kesepahaman konsep Analysis Static untuk mengetahui karakter sebuah malware[7]

Penelitian yang berjudul “Server Analysis Malware pembangunan menggunakan Cucook Sanbox pada Sistem operasi berbasis Linux”. Penelitian ini menggunakan alat analisis yang sama Cockook Sanbox dan membahas tentang analisis malware yang memanfaatkan teknik dinamis. [8]

Beberapa Penelitian yang dilakukan oleh peneliti diatas dapat di lihat pada tabel 1. Berkaitan dengan metode mendasar yang akan diterapkan, maka temuan penelitian yang bertajuk “Penerapan Random Forest Classifier untuk Deteksi PDF Malware di Agregator Garba Rujukan Dgital (GARUDA) Kemendikbud Dikti” telah diperoleh. sebagian besar memenuhi persyaratan penelitian yang akan dilakukan.

2.3 Landasan Teori

2.3. 1 Malware

Malware, atau perangkat lunak berbahaya, adalah perangkat lunak yang sengaja dibuat dan diciptakan untuk merusak, menyusupi, atau mendapatkan akses ke sistem komputer tanpa sepengetahuan pemiliknya sehingga dapat menimbulkan risiko pada sistem dan menimbulkan dampak buruk yang berbeda-beda.[6] Malware hadir dalam berbagai bentuk, termasuk virus dan lainnya, seperti "trojan", "keylonggers", dan "spyware", yang dapat berperilaku berbahaya.[2]

2.4 PDF Malware

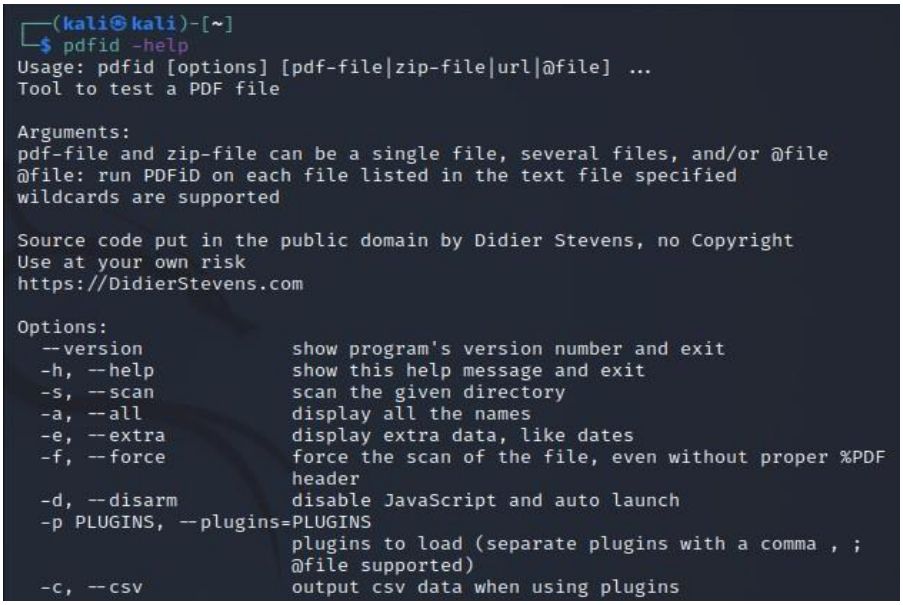
File PDF Malaware[2] adalah sebuah bentuk file yang mudah di akses dan juga di manupulasi karena di dalamnya terdapat teks dan juga hanya sedikit batasan untuk para oknum hacker untuk melakukan peretasan file tersebut. Pada adobe memberikan format yang berisikan penambahan format algoritma enkripsi, scripting, multi media serta support. [5]

Dengan adanya file PDF Malware maka dengan mudah para oknum peretas untuk menanamkan sebuah malware yang berbahaya yang tidak di ketahui para user, dalam PDF Malware tedapat beberapa tools-tools yang dapat mengidentifikasi ciri-ciri malware yang sangat bermanfaat dalam membantu untuk mendeteksi karakteristik dari malware tersebut di deteksi dengan menggunakan Metode Random Forest Classifier.[3]

2.5 PDF ID

Ekstraksi dataset file PDF Malware di lakukan dengan menggunakan PDF ID [7]sebagai alat yang di gunakan dalam menyelsaikan tugas akhir ini. PDF ID merupakan sebuah tools yang di rancang oleh seseorang Bernama Didier Stevens. Analisa secara statis di lakukan menggunakan PDF id karena PDF id adalah Script phyton.

File PDF tersebut akan di tinjau secara langsung menggunakan script yang telah di rancang. Setiap fitur melakukan perhitungan terhadap nilai yang ada. Satu fitur yang umumnya di temukan yang terasa mencurigakan dari sekian banyaknya. Nilai dan fitur yang telah di dapatkan maka akan di konversikan ke dalam bentuk CSV.[9] Gambar 2.1



Gambar 2.1 Antarmuka PDFiD

2.6 Ekstraksi Dataset

Ekstraksi dataset di lakukan untuk memperoleh hasil Analisa secara statis sebagai penunjang dalam menyelsaikan tugas Akhir secara parsing untuk semua file PDF terhadap dataset agar memperoleh hasil headeryang bermanfaat untuk fitur dataset.

2.7 Dataset PDF Malware

Untuk saat ini sudah ada sebanyak 10.000 PDF Malware yang telah kami analisa menggunakan virustotal, sejauh ini kami telah mendapat penambahan 4 malicious jenis dokumen PDF sehingga total dari malicious pdf menjadi 197. Dataset yang digunakan adalah dataset PDF Malware GARUDA.

Dataset PDF Malware GARUDA diperoleh untuk saat ini sudah ada sebanyak 10.000 PDF Malware yang telah kami analisa menggunakan virustotal, sejauh ini kami telah mendapat penambahan 4 malicious jenis dokumen pdf sehingga total dari malicious pdf menjadi 197. Dengan melihat perbandingan total pdf benign(8350) dan PDF malicious(197) yang ada.

2.8 SMOTE

Ketidak seimbangan pada pengklasifikasian suatu bahan akan melibatkan model prediktif sebagai bentuk pengembangan pada data yang mengalami ketidak seimbangan

Kumpulan data yang tidak seimbang merupakan tantangan dalam bekerja sehingga pembelajaran mesin akan sedikit di abaikan. Pendekatan yang dapat di lakukan salah satunya adalah menggunakn SMOTE[6] dengan cara mengambil sample minoritas dengan berlebiha. Duplikasi merupakan salah satu contoh pendekatan paling sederhana. Ada lanjutanya

2.9 Machine Learning

Machine learning adalah salah satu dari cabang Artificial Intelegent (Kecerdasan Buatan). Machine learning di kembangkan agar secara langsung dapat di pelajari itulah mengapa disebut mesin pembelajar, berdasarkan ini machine learning berisi ilmu-ilmu statistika, matematika dan juga lainya.[10]

2.10 Random Forest Classifier

Untuk mengumpulkan hasil tugas klasifikasi, Randome Forest Classifier[3] membuat keputusan untuk pelatih dalam klasifikasi dengan membangun banyak pohon keputusan selama pelatihan. Rata-rata prediksi setiap pohon atau rata-rata setiap regresi untuk setiap pohon regresi dikembalikan dalam hasil Random Forest untuk regresi.

2.11 Metode Malware Analisis

Pada umumnya malware ialah sebuah program yang di kelompokkan tentu berdasarkan tujuan tertentu algoritma dan logika yang di gunakan yang relavan dengannya. Hasilnya, model analitik yang digunakan untuk menyelidiki malware memiliki ikatan yang kuat dengan ilmu komputer dasar. seperti struktur data, algoritma, bahasa pemrograman, dan rekayasa perangkat lunak.

Secara umum, sebuah perangkat lunak menggunakan salah satu dari tiga bentuk analisis untuk menentukan apakah sesuatu yang terhubung dengannya adalah malware

atau bukan. Oleh karena itu, ketiga model tersebut yang masing-masing akan diberikan dan dibahas sebagai berikut merupakan strategi yang dapat diterapkan.[5]

2.12 Malware Analisis Statis

Berbeda dengan pendekatan analisis dinamis, analisis statis melibatkan mempelajari perangkat lunak tanpa menjalankannya. File malware tidak akan langsung terpicu selama analisis statis; sebaliknya, kode sumber tertulis akan diteliti dan ditelusuri, menciptakan kembali kode sumber dan algoritma yang telah dikembangkan oleh program. Data yang dikumpulkan bersifat komprehensif dan dapat memberikan gambaran yang sangat jelas tentang fungsi sistem malware secara keseluruhan. Debugger, assembler, dan program analisis semuanya dapat digunakan untuk analisis statis.

2.12.1 Teknik Analisis

Berikut adalah beberapa contoh metode analisis statis yang berbeda.

1. metode pendeteksian yang mengandalkan metode tanda tangan yang biasa disebut dengan metode sidik jari, masker, atau metode pencocokan string atau pola. Tanda tangan adalah serangkaian program yang dimasukkan oleh pemrogram malware ke dalam aplikasi untuk mengidentifikasi bagian malware tertentu. Pendeteksi malware mencari tanda tangan yang telah ditentukan sebelumnya dalam kode untuk mengidentifikasi konten berbahaya.
2. Teknik deteksi heuristik Nama lain metode ini adalah metode proaktif. Metode ini sebanding dengan metode yang mengandalkan tanda tangan kode tertentu; pendeteksi malware sekarang mencari perintah atau instruksi yang tidak ada dalam perangkat lunak aplikasi. Hasilnya, identifikasi jenis malware yang baru teridentifikasi menjadi lebih mudah. Berikut ini adalah banyak metode analisis heuristik.

a). File based heuristic analysis

Analisis file adalah jenis file yang mencakup analisis heuristik. Metode analisis file ini memeriksa secara menyeluruh konten file, pemrosesan, dan tujuan penggunaan, serta perintah apa pun yang mungkin disertakan untuk menghancurkan

atau merusak file lain.

b). Weight based heuristic analysis

Analisis heuristik berbasis bobot adalah metode lama. Setiap permohonan diberi bobot berdasarkan potensi bahayanya. Program diduga mempunyai kode bahaya jika nilai bobotnya lebih besar dari nilai ambang batas yang terdeteksi.

c). Rule based heuristic analysis

Dalam hal ini, analisis menghasilkan aturan yang menentukan penerapannya. Setelah kriteria tersebut digabungkan dengan aturan yang telah ditetapkan sebelumnya, program dianggap mengandung malware jika aturannya tidak sesuai.

d). Generic signature analysis

Meskipun virus yang dimaksud adalah variasi malwar, virus ini memiliki kemiripan dengan "kembar identik" dalam hal perilaku. Teknik ini mencari varian malware baru dengan memanfaatkan definisi antivirus yang sudah ada.

e).Keuntungan dari metode analisis statis.

Lebih aman dan cepat menggunakan analisis statistik untuk mengumpulkan struktur kode program untuk analisis tertentu. apakah tindakan dan perilaku tindakan keamanan di masa depan dapat dihitung menggunakan analisis statis.

f). Kerugian dari metode analisis statis

Menganalisis malware yang tidak dikenal masih diperlukan selain analisis statis. Banyak kode sumber aplikasi yang sulit ditemukan, oleh karena itu melakukan analisis statis memerlukan peneliti untuk memiliki pemahaman yang lebih baik tentang cara kerja sistem.

2.13 Malware Analisis Dinamis

Analisis dinamis adalah metode yang digunakan dalam teknik ini untuk memeriksa perilaku atau operasi yang dilakukan program saat sedang berjalan. [5] Salah satu cara untuk melakukan analisis dinamis adalah dengan mengikuti panggilan, memantau fungsi, dan mengumpulkan informasi tentang dampak malware yang teridentifikasi saat dilakukan.

Akhrinya dapat kita ketahui apa saja dan kegiatan apa yang telah malware

lakukan pada saat berhasil menginfeksi sebuah perangkat komputer. Pemerikasaan akan di lakukan secara keseluruhan pada tahapan analisis dinamis.

Untuk penelitian ini, mesin virtual atau sandbox biasanya digunakan. Aplikasi yang meragukan biasanya dijalankan di lingkungan virtual; jika aplikasi bekerja aneh, maka dianggap berbahaya atau terinfeksi malware.

2.13.1 Keuntungan dan Kekurangan Analisis Dinamis

1. Keuntungan dari Analisis Dinamis

Salah satu manfaat analisis dinamis adalah memudahkan pengguna mengidentifikasi virus yang tidak dikenal hanya dengan melihat perilaku program.

2. Kerugian dari analisis dinamis.

Pemeriksaan ini memerlukan waktu karena program mungkin berjalan lambat atau tidak aman dalam beberapa situasi.

Aplikasi yang menunjukkan variasi perubahan perilaku dengan berbagai situasi pemicu tidak tercakup dalam pemeriksaan ini. misalnya, tidak mengidentifikasi malware multipath.

2.14 Analisis Hybrid

Teknik analitik statis dan dinamis digunakan dalam penyelidikan ini. Metode. Manfaat dari dua strategi sebelumnya digabungkan dengan cara ini.

a. Hasil Studi Pustaka

Tujuan dari studi pustaka ini mengumpulkan tinjauan dari penellitian sebelumnya yang telah telah di lakukan dengan yang akan di kerjakan oleh peneliti untuk mengetahui keterkaitan dari semua penelitian yang telah di lakukan oleh sebab itu dapat di jadikan acuan untuk melakukan penelitian secara akurat dan hal tersebut terlihat pada tabel 2.1

Tabel 2.1 Perbedaan penelitian terhadulu dan penelitian penulis

Uraian	**Penelitian yang telah di lakukan**	**Penelitian yang belum di lakukan**
Dataset Malware	Malware detaction yang telah di teliti	Deteksi Malware dengan mengekstraksi keselurah data untuk mendapatkan lebih hasil performa
Dataset Malware	Malware detaction yang telah di teliti	Deteksi Malware dengan mengekstraksi keselurah data untuk mendapatkan lebih banyak hasil
	diperoleh, [5], [6], [11], [9], [11], [10], [12], [13], [7], [14],	
	[15], [16], [17], [18], [19], [4],	
Malware Analisis, Statis	[9], [20], [21], [20], [22], [23], [24], [25], [26], [27],[28],[29],[30],[1]	Deteksi Malware
Metode	1. Naïve Byes [5], [21], [21], [12], [13], [7], [16], [18], [19], [4], 2. SVM [21], [11], [21], [11], [14], [17], [27] 3. KNN [21], [21], 4. Randome Forest [21], [21],	Kinerja algoritma Random Forest di bandingkan dengan Naive Bayes dan K-Nerest setelah melalukan proses klasifikasi dengan data set yang telah di kumpulkan menunjukkan bahawa Random Forest memiliki akurasi dan daya ingat yang lebih tinggi di antara metode yang lain. Algoritma pengklasifikasi Random Forest merupakan pilihan terbaik untuk digunakan pada sistem dibandingkan dengan Naive Bayes dan K-Nearest Neighbor karena memberikan hasil yang lebih baik dibandingkan dengan dua algoritma pengklasifikasi lainnya dengan akurasi 94,31% dan rata-rata presisi 94,33%. Hal ini menunjukkan lompatan akurasi dibandingkan penelitian
Sifat	Pengujian metode serta analisis, [9], [20], [9], [11],	Pengujian data deteksi
Kelemahan	Keterbatasan waktu dan sumberdaya	Mengacu pada banyaknya tingkat kurangnya waktu dan sumberdaya Naive Bayes secara umum berkinerja lebih baik di bandingkan dengan yang lain, namun kami mengakui bahwa hasil yang di tunjukkan di atas tidak menunjukkan potensi penuh dari teknik tersebut karena data set yang di gunakan untuk model nya masih kurang, di sebabkan oleh terbatasnya waktu dan sumber daya yang tersedia.

BAB III

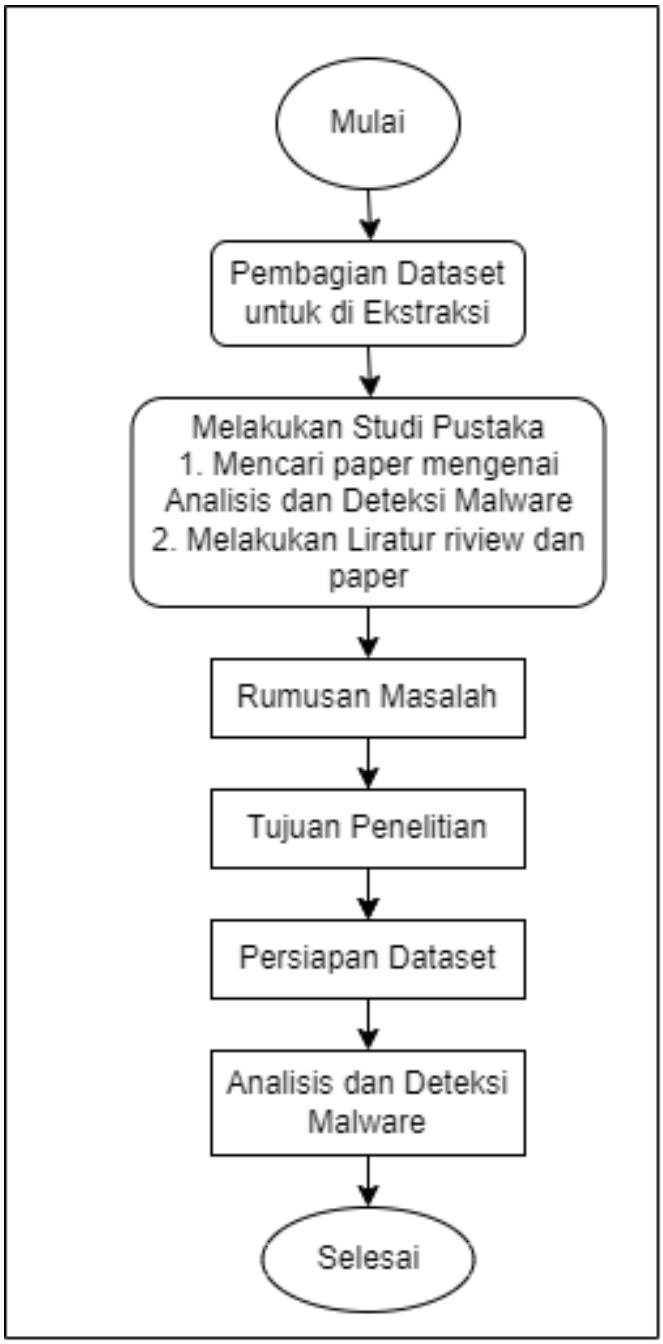
METODOLOGI PENELITIAN

3.1 Pendahuluan

Di dalam bab ini terdapat penjelasan mengenai metode penelitian yang akan di gunakan untuk penelitian tugas akhir, yang mana akan membahas tentang nalisis dan deteksi Pdf Malware pada Agregator Rujukan Digital Kemendikbud Dikti dengan metode Dynamic Analysis. Metode ini kedepannya akan di gunakan secara tersusun dan tersetruktur mengikuti dari alur kerja. Tahapan yang akan di gunakan akan di visualisasikan dalam bentuk kerangka kerja. Kerangka kerja tersebut berisi tahapan-tahapan penelitian seperti studi pustaka, data set, ektaksi data set, pengujian metode, validasi hasil, analisa dan kesimpulan.

3.2 Kerangka Kerja

Pada tahap ini akan tergambarkan setiap alur kerja yang akan di perlihatkan dari awal sampai akhir penelitian tersusun secara sistematis dan detail. Pada tahapan pertama pada penelitian adalah melakukan studi pustaka, persiapan dataset persiapan dataset yang di gunakan adalah dataset yang berasa dari PDF malware GARUDA yang telah di ekstraksi dataset merupakan hal yang sangat penting pasalnya merupakan komponen yang sangat penting untuk melakukan riset, di dalam dataset tersebut terdapat sekumpulan informasi yang dapat membantu dalam jalanya penelitian terserbut , ektraksi dataset pengujian deteksi pada tahap tersebut maka dataset yang telah di kelompokkan akan di uji untuk mengetahui nilai akurasi yang di butuhkan atau terjadinya tidak seimbangan yang terdapat pada dataset tersebut dan juga untuk mengetahui hasil yang di butuhkan , analisis dan kesimpulan yang merupakan upaya untuk mengetahui hasil yang akan di peroleh setelah melakukan analisis dan juga ektraksi yang di lakukan sebagaimana yang terihat, untuk lebih lengkapnya dapat di lihat pada gambar 2, yang akan memaparkan bagaimana cara alur kerja penelitian. Hal tersebut dapat di lihat pada gambar 3.1



Gambar 3.1 Kerangka Kerja

3.3 Kebutuhan perangkat Keras dan Perangkat Lunak

3.3.1 Perangkat Keras

Perangkat yang di perlukan pada saat melakukan penelitian ini, dapat terlihat pada tabel 3.1

Tabel 3.1 Perangkat keras yang di perlukan

No	Sistem	Spesifikasi	Keterangan
1.	Asus VivoBook 15 A516EAO, Operating System : Windows 11 Home + Office Home and Student 2021 Included	[Intel ® Core i3-1115G4, Processor 3.0 GHz (6M Cache, up to 4.1 GHz, 2 cores)/Intel®UHD Graphics/ 4GB/ 512 GB	1 Unit

3.3.2 Perangkat Lunak

Perangkat yang di perlukan pada saaat melakukan penelitian, dapat terlihat pada tabel 3.2

Tabel 3.2 Perangkat lunak yang di perlukan

No	Perangkat lunak	Tools	Keterangan
1.	Sistem Operasi	Kali Lnux	Versi 2021.3
2.	Virtual Machine	VirtualBox	Versi 6.1.32

3.4 Perancangan Sistem

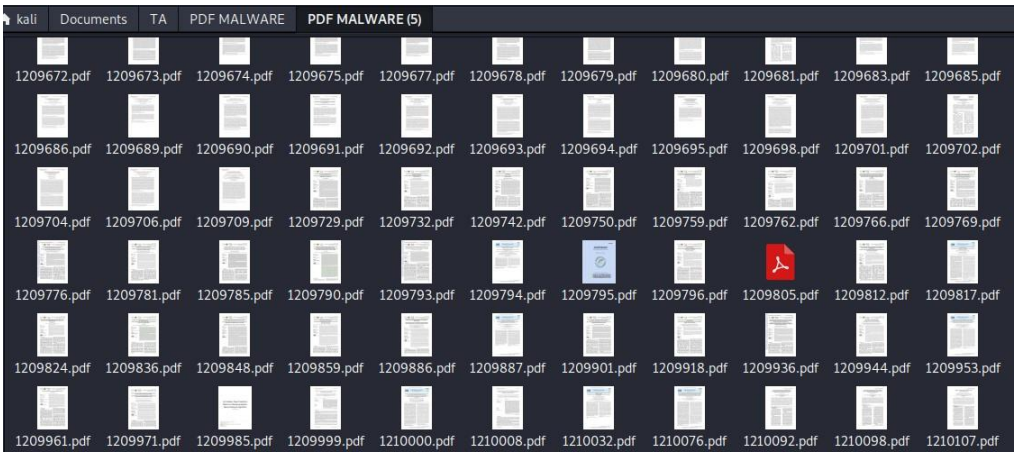
3.4.1 Membangun Virtual Lab

Pada saat menganalisa malware di butuhkan sebuah lingkungan yang aman seperti virtual lab, sehingga peneliti dapat dengan leluasa untuk melakukan aktivitas analisis tanpa merasa cemas ataupun khawatir karena malware yang menyebar sehingga dapat menimbulkan kerugian dan kerusakan pada sistem komputer.

Virtual lab dalam penelitian yang di maksud merupakan sebuah mesin yang virtual yang di dalamnya telah terinstal macam-macam tools yang di perlukan untuk di gunakan pada saat penelitian. Virtual lab yang program yang di gunakan dalam penelitian yaitu Virtualbox Pada pengaturan mesin virtual kegiatan menganalisis di lakukan meliputi operasi sistem yang di gunakan seluruhnya konfigurasi juga termasuk pertimbangan mampu terhubungan dengan jaringan serta perangkat lainya.

3.5 Persiapan Dataset

Saat ini sudah ada sebanyak 10.000 PDF Malware yang telah kami analisa menggunakan virustotal, sejauh ini kami telah mendapat penambahan 4 malicious jenis dokumen PDF sehingga total dari malicious pdf menjadi 197. Dataset yang digunakan adalah dataset PDF Malware GARUDA. Dataset PDF Malware GARUDA diperoleh untuk saat ini sudah ada sebanyak 10.000 PDF Malware yang telah kami analisa menggunakan virustotal, sejauh ini kami telah mendapat penambahan 4 malicious jenis dokumen pdf sehingga total dari malicious pdf menjadi 197. Dengan melihat perbandingan total pdf benign(8350) dan PDF malicious(197) yang ada pada gambar 3.2



Gambar 3.2 File PDF GARUDA

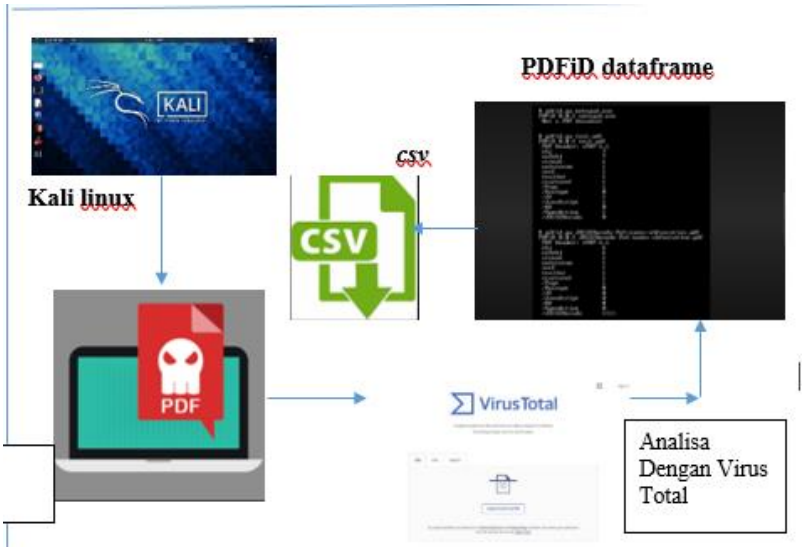
Dan jumlah dari data PDF Malware yang telah di hasilkan maka hasilnya dapat di lihat seperti pada tabel 3.3

No.	Hasil Analisa	jumlah
1	*Benign*	9800
2	*Non-pdfmal*	194
3	*Mal-pdf*	6
	Total	10.000

Tabel 3.3 Jumlah data PDF malware

3.6 Ekstraksi Dataset

Sebelum melalui tahapan Proses yang menggunakan metode Dynamic Analysis, dataset ini di Ekstraksi terlebih dahulu. Ekstrasi adalah ekstraksi adalah proses pemisahan bahan dari campuran data untuk mendapatkan hasil. Dataset yang digunakan adalah dataset PDF MALWARE Dibawah ini akan menampilkan tabel dari hasil Ektraksi 1 dataset menggunakan Virus Total. Seperti yang tertera pada gambar 3.3



Gambar 3.3 Ekstraksi Dataset

3.7 Fitur Extraksi

Obj yaitu fitur yang menunjukka nilainya sesuai dengan jumlah objek yang terdapat dalam sebuah *file* PDF. *EndObj* yaitu fitur yang mana nilainya wajib sama dengan Obj karena merupakan pasangannya. *Stream* yaitu fitur yang nilainya dihitung sesuai intruksi yang urut dan menjelaskan tampilan *file* PDF. *Endstream* merupakan fitur pasangan dari *stream*.

Nilai pada *endstream* harus sama dengan *stream*.*Xref* merupakan fitur yang nilainya dapat mengindikasi apakah *file* PDF merupakan *file* yang berbahaya atau tidak.Jika *xref* memiliki nilai atau tidak sama dengan nol maka *file* PDF tersebut tidak berbahaya. *Trailer* tersebut fitur yang juga dapat mengindikasi apakah *file* PDF merupakan *file* yang berbahaya atau tidak. Jika nilai dari fitur ini sama dengan nol,

maka *file* PDF tersebut tidak sempurna atau tidak aman. *Startxref* fitur tersebut merupakan pasangan dari *xref*. Nilai dari fitur ini harus sama dengan nilai *xref* untuk mengindikasi jika *file* PDF tidak berbahaya. Fitur ini memiliki nilai sesuai dengan jumlah halaman *file* PDF. *Encrypt* fitur tersebut yang nilainya mengindikasi adanya kode yang disematkan dalam *file* PDF.

/ObjStm Fitur tersebut dapat digunakan untuk mengaburkan atau menyamarkan sebuah objek yang dapat mengarah pada *file* berbahaya. /JS Fitur tersebut adalah tanda dari adanya *JavaScript* dalam sebuah *file* PDF. Nilai pada fitur ini dihitung dari jumlah pernyataan *JavaScript* dalam satu baris. /*JavaScript* Fitur tersebut juga merupakan tanda dari adanya *JavaScript* dalam sebuah *file* PDF.

Namun nilai pada fitur ini dihitung dari jumlah blok pernyataan *JavaScript*. /AA Nilai yang fitur ini menunjukkan jumlah tindakan yang akan dilakukan ketika *file* PDF dibuka, seperti respon *user* untuk mengakses perintah *link*. /*OpenAction* Nilai dari fitur ini juga menunjukkan jumlah tindakan yang akan dilakukan ketika *file* PDF dibuka namun tidak memerlukan respon *user* untuk menjalankannya. Artinya fitur ini dapat langsung berjalan dengan sendirinya ketika *file* PDF dibuka. /*Acroform* Nilai dari fitur ini dihitung berdasarkan jumlah konten yang terdapat dalam sebuah *file* PDF.

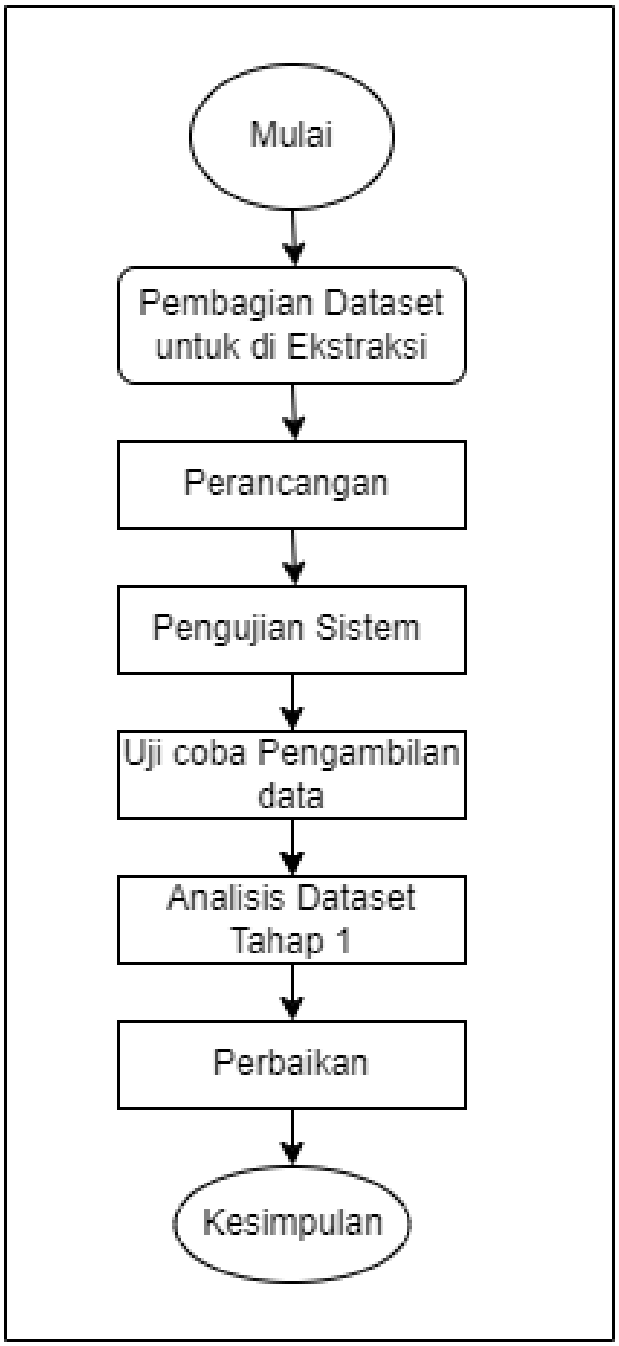
Fitur ini dapat berjalan dengan sendirinya./JBIG2Decode Nilai dari fitur ini akan menunjukkan jumlah dari penggunaan filter JBIG2Decode pada *file* PDF. /*RichMedia* Nilai yang fitur ini menunjukkan jumlah blok konten *flash* yang dapat ditemukan dalam sebuah *file* PDF. /*Launch* Nilai dari fitur ini dihitung berdasarkan jumlah tempat yang dapat digunakan untuk menjalankan program *script* pada *file* PDF.

/*EmbeddedFile* Nilai dari fitur ini dapat mengetahui apakah ada *file* yang mungkin berbahaya dalam sebuah *file* PDF. Fitur ini menunjukkan jumlah kode yang ditanamkan pada sebuah *file* PDF. /XFA Nilai yang dari fitur ini menunjukkan jumlah XML yang digunakan dalam sebuah *file* PDF. /Color > 2^24 Nilai yang fitur ini dapat mengetahui bahwa *file* PDF beresiko berbahaya jika nilainya lebih dari 2^{24}**.**

3.8 Blok Diagram Penelitian

3.8.1 Tugas Akhir I

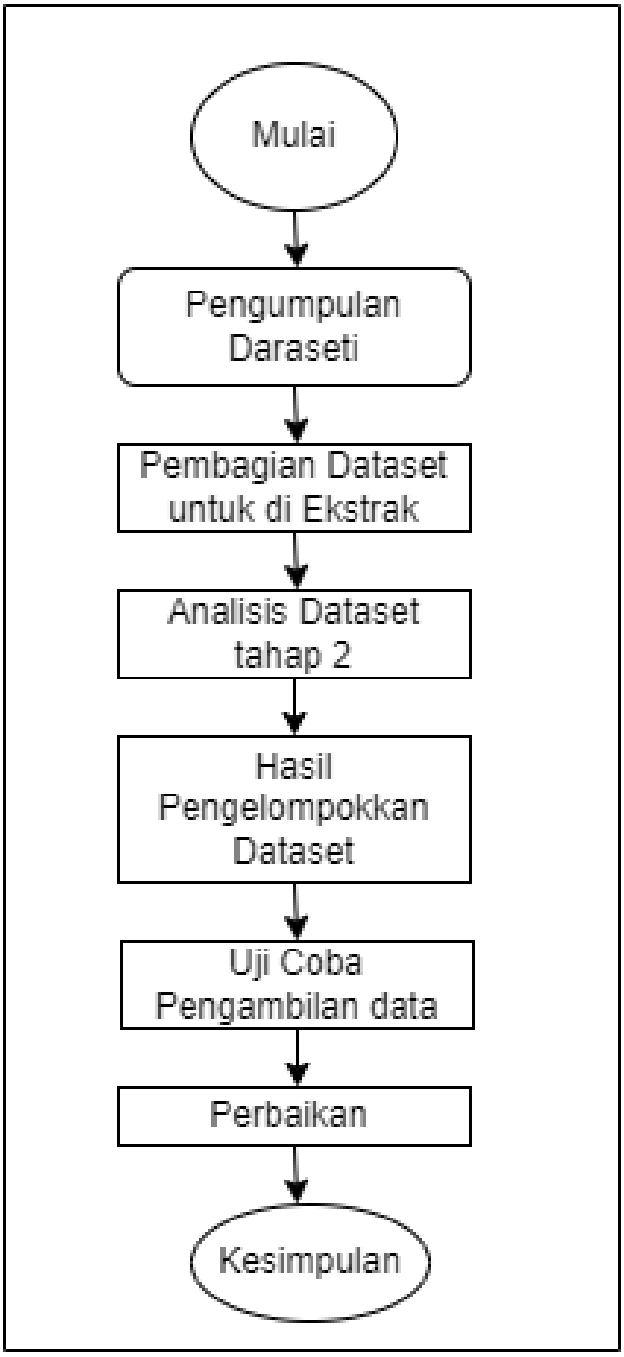
Diagram alur penelitian Tugas Akhir I yang dapat di lihat pada gambar 3.4



Gambar 3. 4 Blok Diagram TA I

3.8.2 Tugas Akhir II

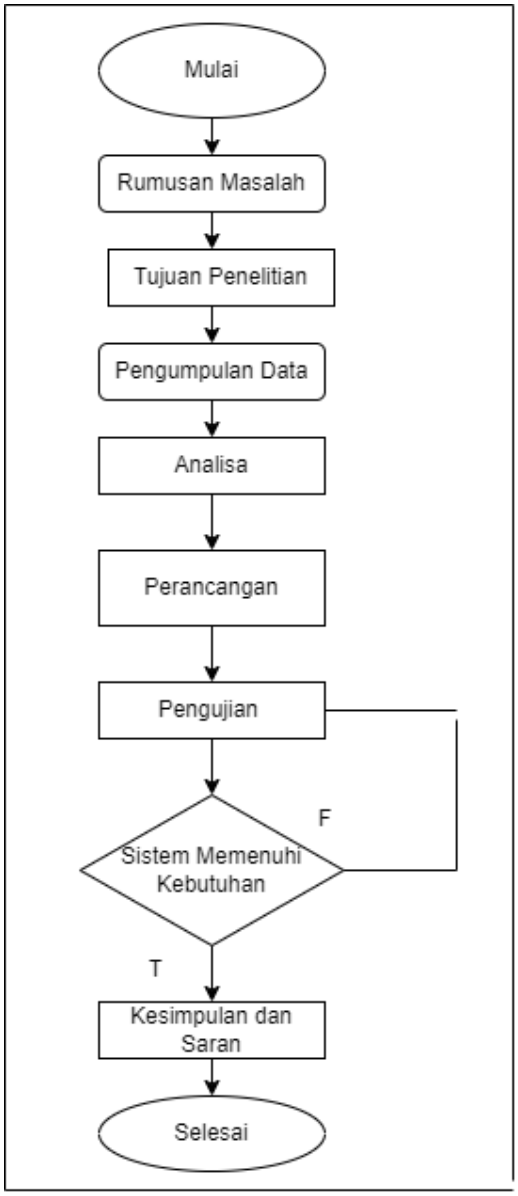
Diagram alur penelitian untuk Tugas Akhir II dapat di lihat pada gambar 3.5



Gambar 3.5 Blok Diagram TA II

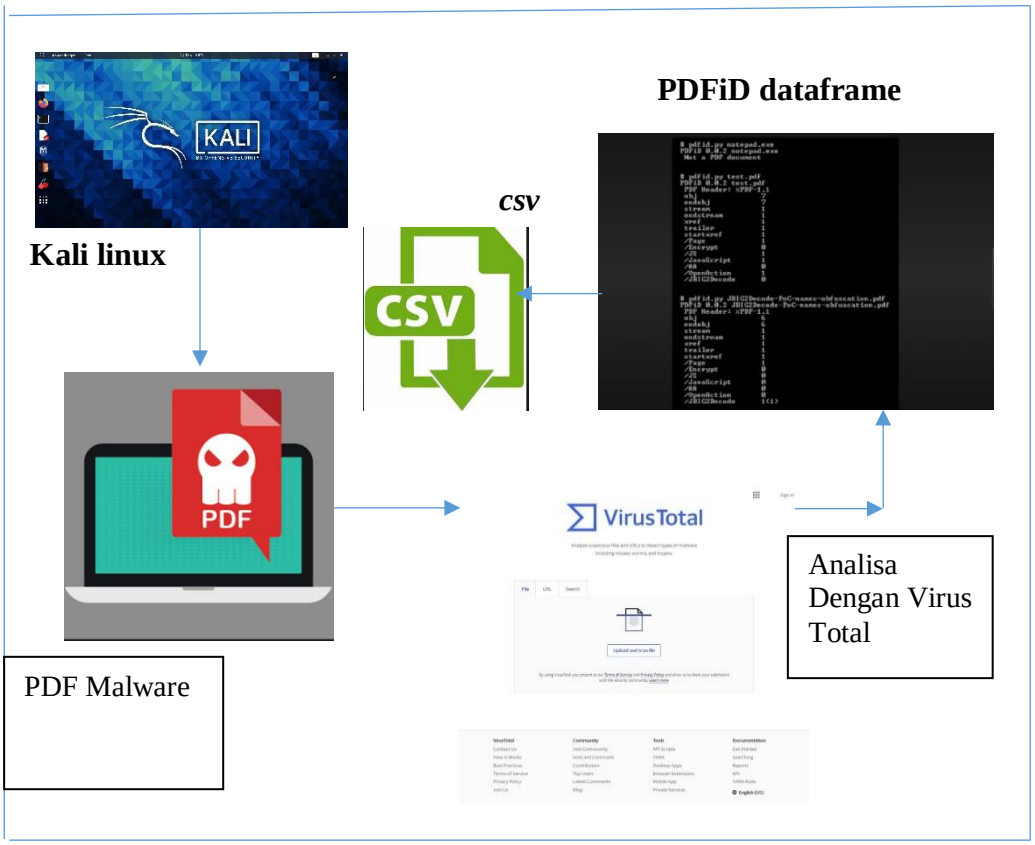
3.9 Tahapan Penelitian

Adapun tahapan Penelitian yang akan dilaksanakan pada penelitian tugas akhir ini dimulai dari perumusan masalah, tujuan, pengumpulan data, analisa, perancangan, pengujian serta tahap pendukung. Berikut adalah gambar dari tahapan penelitian tugas Akhir dapat di lihat pada gambar 3.6



Gambar 3.6 Tahapan Penelitian

Tahapan penelitian yang akan di lakukan meliputi beberapa prosedur-prosedur pengerjaan dan secara garis besar melalui beberapa tahapan (fase) yaitu; (1) Perumusan masalah, (2) pengumpulan data, (3) analisa, (4) perancangan, (5) pengujian, (6) kesimpulan dan saran. Di atas dapat di ketahui secara keseluruhan bagaimana sistem kerja keseluruhan, pengolahan data menggunakan Virtual BOX, kali LINUX sehingga dihasilkan output dari deteksi Malware. Metode yang di gunakan dalam penelitian ini meliputi studi kepustakaan, karakteristik dan pengelompokkan data sesuai jenisnya gambar dapat di lihat pada gambar 3.7

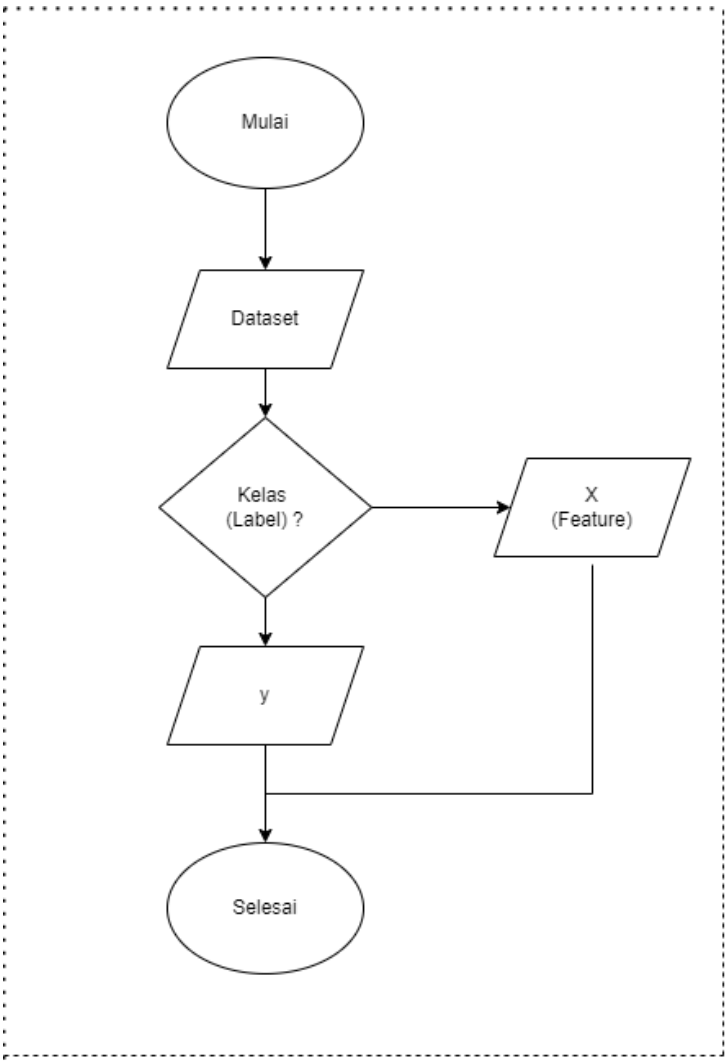


Gambar 3.7 Perancangan Sistem Penelitian

3.10 Dataset

Dalam hal ini penelitian akan berfokus karena kami telah memperoleh beberapa kasus seperti file PDF Malware yang mana dalam dataset yang di peroleh sebanyak 10.000 dataset yang telah di ekstraksi di peroleh 194 file *mal-HTML* DAN 6 File *mal-*

PDF malicious yang mana malware ini belum di ketahui karakteristiknya oleh sebab itu perlu di lakukanya analisa dan deteksi hal ini di lakukan agar terhindar dari berbagai serangan dan ancaman yang dapat merugikan bagi para korbannya, kerugian yang di peroleh antara lain seperti penyadapan serta pencurian informasi pribadi, hingga kasus perusakan sistem yang dilakukan oleh penyusup (Intruder) terhadap perangkat korban dengan berbagai alasan. Dataset ini memiliki dua puluh tiga atribut salah nya adalah class label. Atribut data yang di beri kelas dan label termasuk dalam y, dan yang bukan akan termasuk dalam x., seperti terlihat pada gambar di bawah. 3.8

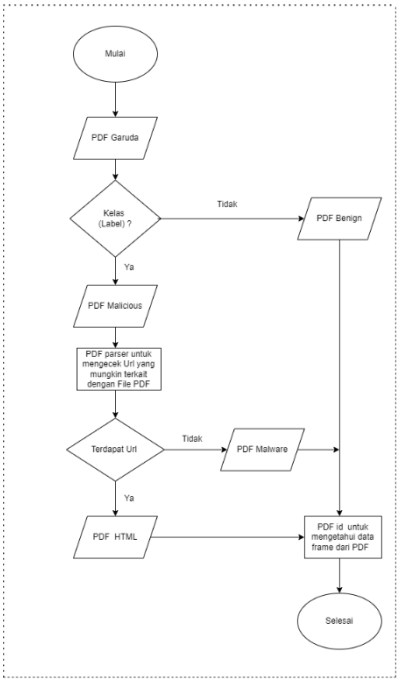


Gambar 3.8 Flow Chart Dataset

3.11 Analisa Statis Dataset PDF GARUDA

Pada tahap ini kami melakukan pembagian dataset menjadi beberapa kelompok sehingga masing-masing memperoleh bagian sendiri untuk melakukan ekstraksi, pengecekan di lakukan secara rinci dengan mengecek satu per satu tiap bagian *File* PDF. Kali Linux adalah sistem operasi yang di guanakan untuk proses analisa pada virtual Box yang sudah terinstal. Mengekstrak informasi di lakukan menggunakan analisa statis untuk mengekstraksi informasi dari malware yang tidak di jalankan melalui kode analisis dari malware itu sendiri.

Pada tahapan selanjutnya adalah dengan memanfaatkan *website* Virus Total kemudia akan muncul sebuah identifikasi mengenai jenis apakah yang terkandung di dalam file PDF tersebut, selanjutnya yaitu menggunakan teriminal sistem operasi Kali Linux untuk menganalisa. Untuk memfilter URL. http yaitu dengan melakukan PDF Parser. Tahap selanjutnya yaitu dengan mengekstraksi file menggunakan PDF id. untuk dapat mengetahui data frame pada setiap file PDF. Penggabungan antara PDF Benigh dan PDF *non-pdfmal* dan *pdf-mal* menjadi sebuah file Dataset dengan format CSV. Diagram blok dapat di lihat pada gambar 3.9



Gambar 3.9 Flow Chart Analisa Statis

3.12 Pre-processing

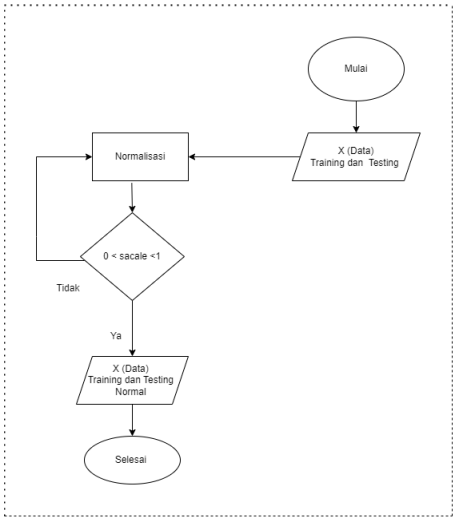
Pada kesempatan analisa di lakukan menggunakan bahasa pemograman website Google Colaboration. Pertama di awali dengan mengimport data set kedalam bentuk format. Csv ke nootbook google colaboration, pre-processing di gunakan untuk pelebelan data dan Oversampling dengan SMOTE serta Undersampling menggunakan Nearmiss

3.12.1 Pelabelan Data

Pelabelan tersebut data berfungsi untuk sebagai pengubah kata menjadi angka agar mempermudah dataset yang di labelkan yaitu, label mal-pdf di ubah menjadi 1, Benign di ubah menjadi 0 dan non pdf-mal di ubah menjadi 2.

1.13 Normalisasi

Tahap selamjutnya yaitu melakukan normalisasi ,hal tersebut di lakukan jika data yang di hasilkan kurang baik, tujuanya adalah angka yang ada di dalamnya dapat tersamarkan yang terdapat pada dataset dalam lingkup yang sama sehingga selisih tidak jauh karena nilainya di skala yang sama. Pada normalisasi di awali dengan inisiasi data training dan testing. Pada saat proses data yang mempunyai skala 0 sampai dengan 1 proses normalisasinya berhasil dan apabila tidak maka normalisasi tersebut gagal dan perlu di lakukan ulang. Hal tersebut dapat di lihat pada gambar 3.10



Gambar 3.10 Flow Chart Normalisasi

1.14 Processing

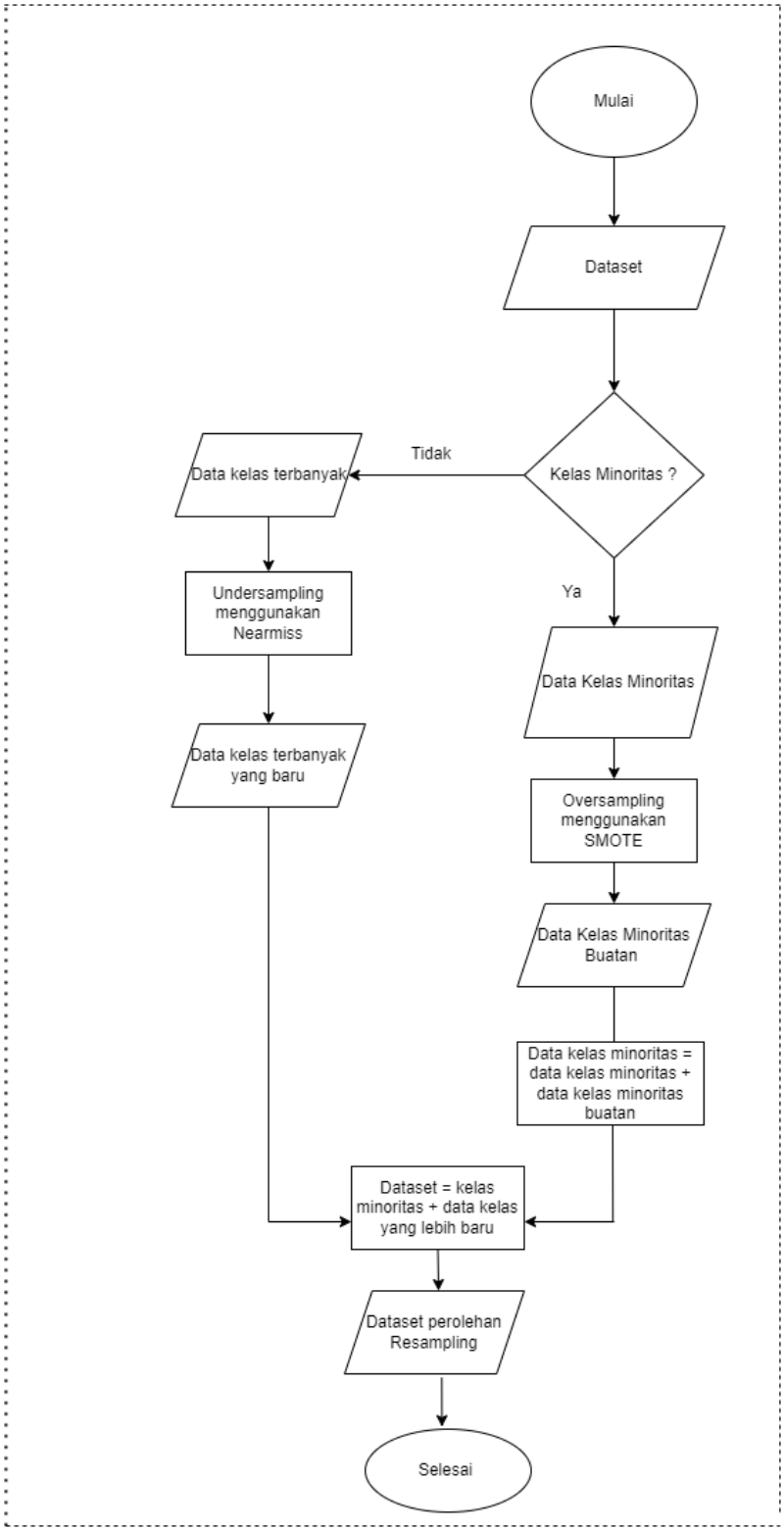
Proses yang di lakukan ini di namakan proses Resampling yang bertujuan untuk menyeimbangkan data yang tidak seimbang atau yang di sebut data Imbalance maka hal yang dapat di lakukan adalah melakukan *Oversampling* dengan menggunakan SMOTE dan juga sample minoritas pada kelas dan Nearmiss di gunakan untuk *Undersampling* yaitu menghapus sample kelas mayoritas, seperti yang terlihat pada gambar 3.3 diagram alir proses Resampling

3.14.1 Resampling

Data yang memiliki tingkat kemiringan atau perbandingan, pada penelitian yang sangat parah dalam distribusi kelas, dalam penelitian ini dataset di nyatakan imbalance karena lebih dari 10.000 data, file *mal-pdf* yang di temukan hanya berjumlah 194. file *mal-pdf* berjumlah enam dan sisanya adalah *file benign* algoritma mempengaruhi sehingga nilai yang di hasilkan menjadi tudak relavan. Terdapat 2 macam cara agar data yang dimiliki menjadi imbalance yaitu dengan cara oversampling dan undersampling. Oversampling merupakan cara menyeimbangkan dengan menduplikasi kelas minoritas. Tidak menghilangkan informasi pada keseluruhan data merupakan Oversampling,

Tetapi dapat menyebabkan Overfitting dan waktu pelatihan menjadi lama karena hasil yang di berikan menjadi lebih besar menggunakan training. Sedangkan Undersampling adalah cara untuk menyeimbangkan dengan cara menghapus beberapa kelas lebih banyak. Undersampling lebih menghemat waktu karena waktu ekstrak lebih kecil dari pada mayoritas.

Terakhir adalah pada tahap ini yaitu penggabungan dari kedua cara tersebut ialah menggunakan SMOTE *(Syntetic Minority Oversampling Technique)* untuk penggunaan Oversampling dan juga Near Miss untuk metode persatuan di antara keduanya yaitu Oversampling dan Undersampling. Manfaat dari penggabungan ini adalah sebagai metode untuk mengurangi kemungkinan kehilangan sample. Yang efisien dan juga memungkinkan pengelompokan. Hal tersebut dapat di lihat pada gambar 3.11



Gambar 3.11 Over-Undersampling

Pada tahap tersebut kita mengumpulkan data dari PDF malware yang kemudian di ekstraksi dan di jadikan sebagai dataset , lalu pada dataset akan kita cek kelas mayoritas dan juga kelas minoritas yang akan di proses menggunakan SMOTE, pada kelas mayoritas maka akan di gunakan Nearmiss untuk proses *Undersampling*, namun jika kelas pada mayoritas maka akan di gunakan Nearmiss untuk Oversampling , kemudian akan menghasilkan kelas mayoritas baru pada proses Undersampling, proses Oversampling menggunakan SMOTE maka jumlah kelas akan meningkat sama dengan kelas mayoritas, yang selanjutnya kelas mayoritas dan kelas minoritas baru yang kemudian di sebut data resampling.

Untuk mengontrol nilai agar mendapat nilai yang sama setiap kodenya yaitu menggunakan *Random-state.* Parameter yang di gunakan untuk proses Resampling di gunakan nilai validasi 123 karena parameter *Random-state* bisa menggunakan bilangan bulat lainya seperti yang banyak di pakai yaitu 1 atau 24. 123 jika di gunakan maka output yang akan di hasilkan akan sama setiap kali di eksekusi kodenya pada tabel 3.4

Tabel 3.4 Parameter SMOTE

Parameter	Validasi
Random-state	123

Pada proses Resampling akan di gunakan Pseudocode dengan teknik SMOTE dan *Nearmiss* untuk Undersampling dapat di lihat pada gambar 3.12

```
Program     : Proses Resampling
Deklarasi   : X_mm, X_smote : float
              Y, y_smote : integer

Deskripsi
1. START
2. CALL make_pipline from imlearn.pipline
3. CALL SMOTE from imblearn. over_sampling
4. CALL NearMiss from imblearn. under_sampling
5. Balance number - 3000
6. pipe - make_pipeline
          NearMiss (sampling_Strategy={0: balance_number}),
          SMOTE(SAMPLING_Strategy={1: balace_number,  2:
balance_number, random_state=123)
  )
7. X_smote, y_smote-pipe.fit_resample(X_mm, y)
8. END
```

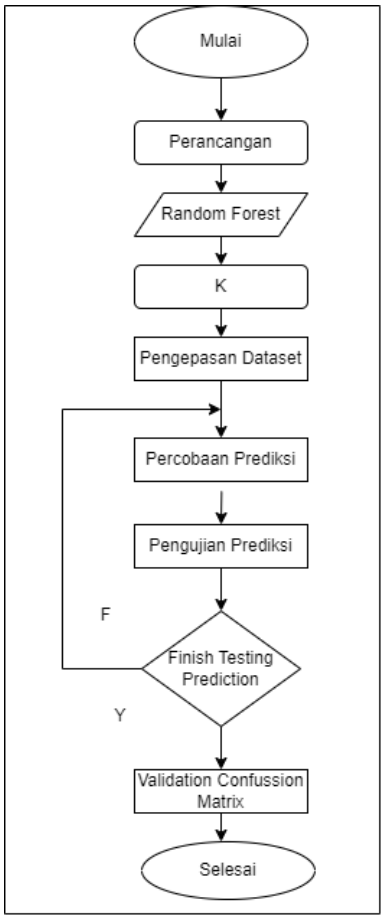
Gambar 3.12 Pseudocode untuk Proses Resampling

3.15 Processing

Processing pada tahap ini adalah proses untuk melakukan beberapa tahapan yang akan di lakukan yaitu proses Klasifikasi menggunakan *Random Forest*, dan hasil performa *validasi* menggunakan *Stratified k-fold Cross Validation*.

3.15.1 Penerapan *Random Forest*

Penggunaan *Random Forest* berdasarkan *majority* merupakan pilihan suara terbanyak data di bagi menjadi data testing dan juga training yang mana 80& untuk data *training* dan juga 20% untuk data *testing.* Penerapan random forest untuk klasifikasi dan pendeteksian menggunakan parameter yang akan di gunakan. Spesifikasi Parameter dapat di lihat Pada penelitian ini penggunaan algoritma Random Forest akan di dapat di lihat pada tahapan-tahapan untuk klasifikasi PDF malware yang terdapat pada gambar 3.13



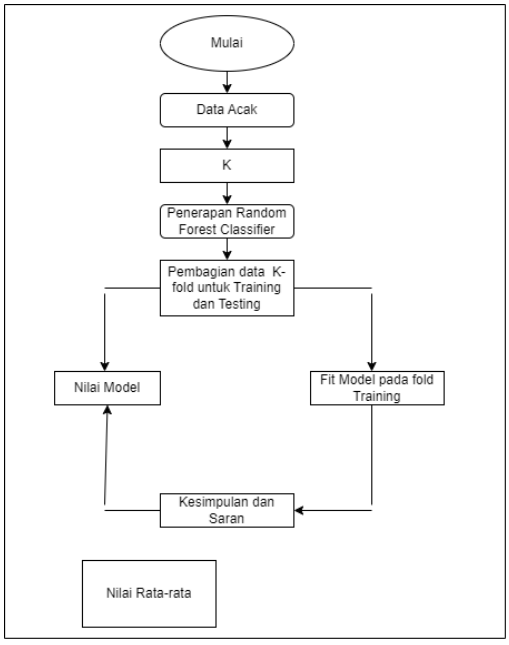
Gambar 3.13 Flowchart Random Forest

3.15.2 Validasi

Untuk mengetahui hasil kinerja dan juga performasi dari proses deteksi dan klasifikasi PDF malware maka di gunakan metode Random Forest Classifier untuk memvalidasi. Pada setiap percobaan maka akan di lakukan perbandingan hasil, itulah proses validasi.

3.15.3 Stratified Cross Validation

Pada tahap ini proses *Stratified K-fold Cross Validation* di gunakan untuk memvalidasi performa hasil dengan cara membagi dua yaitu data testing dan juga data training yang di lakukan untuk beberapa kali percobaan sebanyak nilai K yang di butuhkan. Yang pertama adalah melakukan pengecekan dari dataset, menginisialisasi nilai n_split, membagi K menjadi jumlah lipatan (fold) menyisipkan satu lipatan untuk testing dan untuk training menggunakan sisa fold, fit model untuk training dan evaluasi model pada training, semua proses dapat di ulangi untuk semua fold, fold terpisah setiap kali sebagai data uji, pada kumpulan data, maka setiap iterasi model akan di latih dan di uji, dari seiap split akan mendapatkan nilai rata-rata lalu nilai akan di jumlahkan terlihat pada gambar 3.14



Gambar 3.14 *Proses K-Fold*

Pada proses K-Fold selanjutnya dapat di lihat yaitu pseudocode yang telah di buat, hal tersebut dapat di lihat pada gambar 3.15

```
Program     : Stratified K-FOLD Cross Validation
Deklarasi   : X_smote : float
              Y_smote, train_index, test_index : integer

Deskripsi
1. START
2. CALL  cross_val_score && StratifiedKFold Library
3. skf - StratifiedKfold(n_splits=7)
4. Get n-splits
5. FOR train_index, test_index in n_split DO
    Write(train_index, test_index)
6. END FOR
7. Score - cross_val_score(model,X_smote, y_smote,cv-skf)
8. Write (score)
9. Write (score.mean)
10. END
```

Gambar 3.15 Pseudocode untuk K-Fold

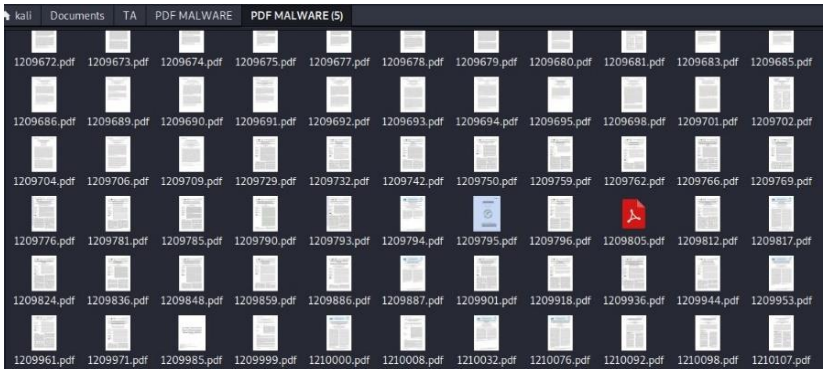
BAB IV
HASIL DAN ANALISA

4.1 Pendahuluan

Pada bab 4 ini akan membahas tentang tahapan-tahapan pengujian yang akan di lakukan dalam penelitian sesuai dengan prosedur perancangan sistem berupa *pre-processing* dan *processing*, dan selanjutnya akan di lakukan pelabelan data lalu melakukan *Oversampling* di karenakan dataset pada penelitian ini data imbalance. Pengujian data pada PDF malware dengan menggunakan *Random Forest* pada tahap *processing* lalu di lakukan SMOTE yang di gunakan untuk menyeimbangkan data yang imbalance dan proses *K-fold cross validation* pada tahap processing klasifikasi PDF malware di lakukan dengan *Random Forest* dan melakukan validasi kepada data pengujian.

4.2 Dataset

Dataset yang telah di olah berasal dari GARUDA *repository* yang telah di ekstraksi. Dalam data tersebut terdiri dari 3 kelompok yaitu kelas *benign, mal-pdf, dan non-pdfmal* dataset ini berjumlah 10.000 yang mana data tesebut terdiri dari 9.800 data benign, 194 data non-malpdf dan 6 data mal-pdf. Dataset tersebut memiliki 21 atribut dan 1 class label. Dapat dilihat dataset PDF Malware GARUDA pada gambar 4.1



Gambar 4.1 File PDF Malware

Untuk Dataset sebelum di Normalisasi yang dapat di lihat pada tabel 4.1

Tabel 4.1 Tabel Dataset GARUDA

Nama PDF	obj	endobj	stream	endstream	xref	trailer	startxref	/Page	/Encrypt	/ObjStm	/JS	/JavaScript	/AA	/OpenAction	/AcroForm	/JBIG2Decode	/RichMedia	/Launch	/EmbeddedFile	/XFA	/Colors>2^24	label
603616.pdf	160	160	128	128	1	1	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603629.pdf	41	41	30	30	1	1	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603639.pdf	235	235	204	204	1	1	1	14	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603645.pdf	153	153	66	66	1	1	1	21	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603652.pdf	170	170	129	129	1	1	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603660.pdf	5058	5056	223	223	1	1	1	156	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
594487.pdf	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	mal-html
603669.pdf	94	94	26	26	1	1	1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603680.pdf	92	92	31	31	1	1	1	11	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603687.pdf	145	145	126	126	1	1	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603694.pdf	134	134	75	75	1	1	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603701.pdf	76	76	21	21	1	1	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603723.pdf	84	84	25	25	2	2	2	12	0	4	0	0	0	0	0	0	0	0	0	0	0	Benign
813661.pdf	520	520	21	21	1	1	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	mal-pdf
603738.pdf	241	241	212	212	1	1	1	13	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603749.pdf	205	205	146	146	1	1	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603760.pdf	78	78	67	67	1	1	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603772.pdf	101	101	42	42	1	1	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603787.pdf	80	80	24	24	2	2	2	15	0	3	0	0	0	0	0	0	0	0	0	0	0	Benign
603805.pdf	131	131	102	102	1	1	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603814.pdf	353	353	312	312	1	1	1	19	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign
603828.pdf	149	149	112	112	1	1	1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	Benign

4.2.1 Pelabelan Data

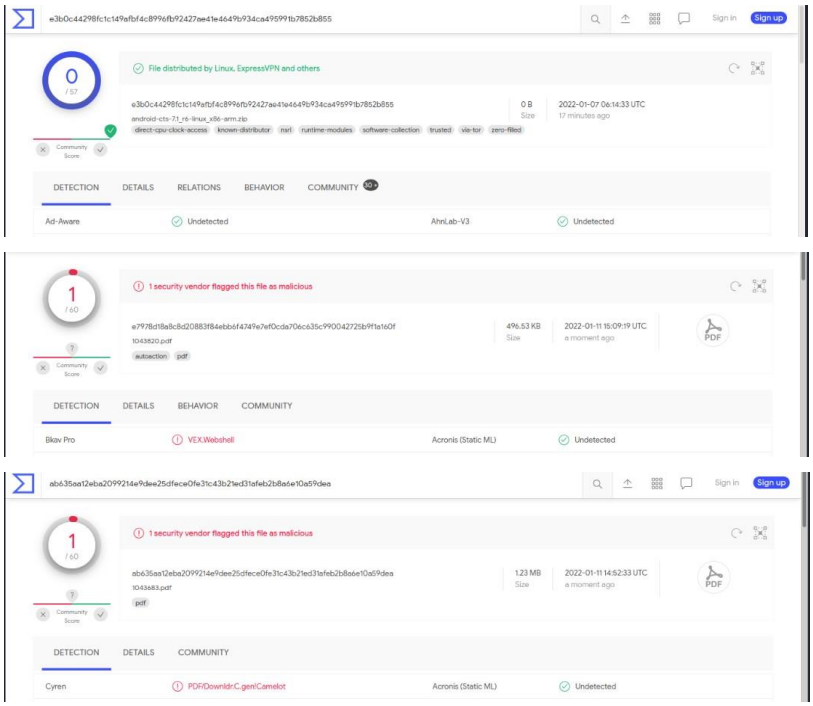
Agar mudah di proses maka label sebelumnya di lakukan perubahan menjadi 0 untuk benign, 1 untuk non-malpdf dan 2 untuk mal-df hal tersebut dapat di lihat pada gambar 4.2

Gambar 4.2 Pelabelan Data

4.2.2 Analisa Statis PDF GARUDA

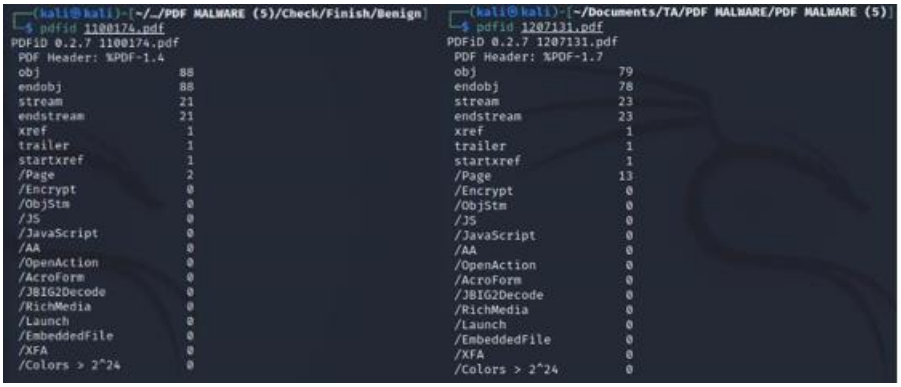
Pada tahap ini di lakukan untuk langkah awal tugas akhir. Dataset berupa file PDF akan di ekstraksi menggunakan sistem operasi Kali Linux yang telah di instal pada VirtualBox. Pada tahap ini di lakukan secara Statis di lakukan pengecekan secara manual satu persatu pada setiap file PDF pada dataset. Pada prosesnya di bagi menjadi dua macam yaitu;

1. Analisa menggunakan *Website* VirusTotal : Pada tahap ini yaitu kami memiliki file PDF malware berjumlah 10.000 file, yang akan di ekstraksi menggunakan virus total untuk di analisis dan cek secara manual satu persatu pada *website*, selanjutnya VirusTotal akan secara otomatis menjalankan sistem untuk medeteksi dan mengidentifikasi apakah file tersebut menyisipkan malware atau tidak. Hasil dari 10.000 file PDF yang telah di ekstraksi adalah 9800 benign, 200 file pdf malware. Analisa tersebut menggunakan Virus Total tersebut dapat di lihat pada gambar 4.3



Gambar 4.3 Analisa Statis VirusTotal

2. Analisa Menggunakan Terminal Linux : Proses analisis ini menggunakan PDFiD setelah melakukan proses tersebut dapat di peroleh hasil dari 200 file PDF malware yang telah di analisa di hasilkan 194 file mal-html dan 6 file pdf-mal setelah di lakukan analisa menggunakan PDFiD menghasilkan nilai dari 21 atribut dataframe yang di miliki, berikut gambar nya setelah di lakukan normalisasi terlihat pada gambar 4.4



Gambar 4.4 Analisa PDFiD

Hasil dari analisa menggunakan PDFiD dapat di ketahui file PDF malware yang berjumlah 10.000 dan telah di analisa menghasilkan 9800 file *benign,* 194 file *non-pdfmal* dan enam *pdf-mal*. Sehingga hasil yang di dapatkan adalah kumpulan-kumpulan dataframe dari nilai tersebut yang selanjutnya akan di ubah menjadi *dataset* dalam bentuk *.csv (Comma Separated Value)*.

3. Fitur Atribut Dataframe

Beberapa fitur Dataframe yang di hasilkan sebagai berikut;

1) Obj

Yaitu fitur yang berfungsi untuk menunjukkan nilai yang sesuai dengan jumlah objek yang terdapat pada file

2) EndObj

Yaitu sebuah fitur yang mana nilainya wajib sama dengan Obj karena mereka berpasangan

3) Stream

Yaitu fitur yang nilainya dapat di hitung sesuai dengan instruksi yang urut dan menjelaskan tampilanya.

4) EndStream

Yaitu fitur yang merupakan pasangan dari Stream karena pada EndStream nilainya harus sama dengan Stream

5) Xref

Yaitu fitur yang mana nilainya dapat mengetahui apakah file PDF merupakan file yang berbahaya atau tidak. Jika nilai yang di hasilkan dari fitur ini sama dengan 0, maka tidak sempurna file PDF tersebut dengan kata lain tidak aman

6) Trailer

Yaitu fitur yang mana dapat mengetahui apakah file PDF tersebut adalah file yang membahayakan atau tidak, jika nilai dari fitur tersebut sama dengan 0, maka file tersebut tidak aman

7) Starxref

Yaitu fitur yang merupakan pasangan dari Xref, yang mana nilainya wajib sama dengan fitur dari Xref untuk dapat mengetahui apakah file PDF berbahaya atau tidak

8) */Page*

Yaitu fitur yang memiliki nilai yang sama dengan jumlah yang ada di file PDF

9) */Encrypt*

Yaitu fitur yang nilainya dapat mengetahui adanya kode yang di pin di dalam file PDF

10) */ObjStm*

Fitur tersebut dapat di gunakan untuk menyamarkan sebuah objek yang dapat menuju pada file PDF berbahaya

11) */JS*

Fitur tersebut adalah tanda terdapatnya JavaScript yang terdapat pada file PDF nilai pada fitur tersebut di hitung dari jumlah pernyataan pada blok JavaScript

12) */JavaScript*

Fitur tersebut juga salah satu tanda adanya JavaScript yang terdapat pada sebuah File PDF, tetapi nilai dalam fitur ini di hitung dari jumlah blok pada pernyataan JavaScript

13) */AA*

Nilai yang fitur ini tunjukkan yaitu jumlah tindakan yang akan di lakukan ketika file di akses, menunjukkan respon

14) */OpenAction*

Nilai dari fitur ini menunjukkan banyaknya tindakan yang akan di lakukan ketika file PDF di akses tetapi tidak membutuhkan respon dari pengguna untuk menjalankanya, karena otomatis

15) */ Acroform*

Nilai fitur tersebut di hitung dari jumlah konten yang ada pada sebuah file PDF dan berjalan secara otomatis

16) /JBIG2Decode

Menunjukkan berapa jumlah pengguna yang mengakses filter JBIG2Decode pada file PDF

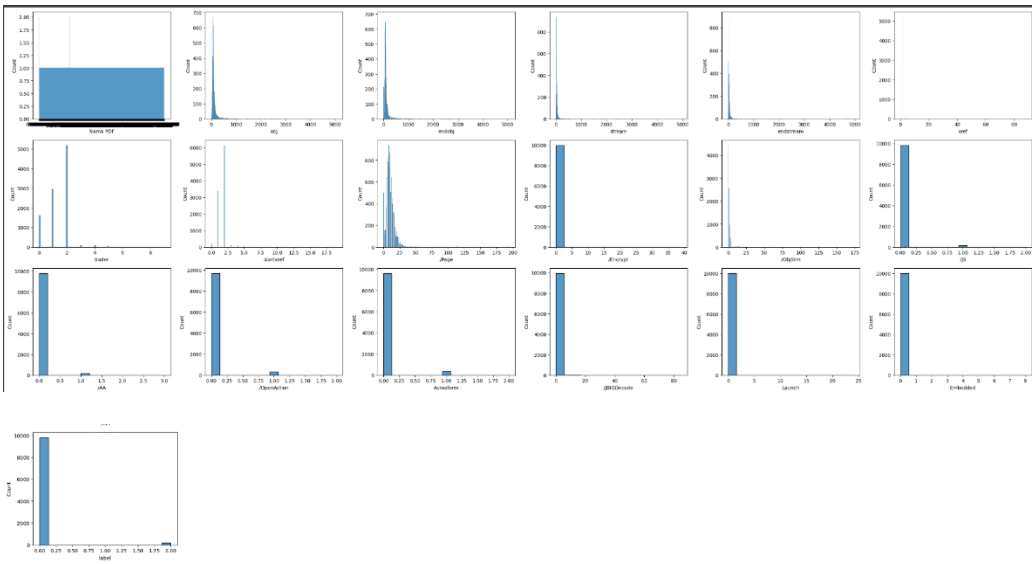
17) /Launch

Nilai fitur tersebut di hitung dari jumlah tempat yang dapat digunakan untuk mendemokan program Script pada file PDF.

18) /Embeddedfile

Nilai pada Fitur tersebut dapat berfungsi untuk mengetahui apakah adanya file yang dapat membahayakan di dalam PDF. Fitur ini akan menginformasikan jumlah kode yang di tanam pada file PDF.

Hasil dari Dataframe yang di hasilkan menggunakan PDFiD telah di ketahui seperti pada gambar 4.5



Gambar 4.5 Atribut Dataframe

Setelah dilakukanya analisis menggunakan PDFiD dapat di ketahui jumlah PDF yang yaitu 10.000 yang kemudian menghasilkan *9800 file benign, 194 file non-mal* dan *6 pdf-mal* dalam bentuk dataframe yang kemudian dari nilai tersebut di ubah ke dalam bentuk dataset csv seperti terlihat pada Tabel 4.2

Tabel 4.2 Hasil Ekstraksi Normalisasi

obj	endobj	stream	endstream	xref	trailer	startxref	/Page	/Encrypt	/ObjStm	...	/AA	/OpenAction	/AcroForm	/JBIG2Decode	/RichMedia	/Launch	/EmbeddedFile	/XFA	/Colors>2^24
0.017200	0.017207	0.006888	0.006888	0.011364	0.111111	0.052632	0.061856	0.0	0.00000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.012060	0.012065	0.004457	0.004457	0.022727	0.222222	0.105263	0.036082	0.0	0.00578	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.011269	0.011274	0.003444	0.003444	0.022727	0.222222	0.105263	0.025773	0.0	0.00578	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.014630	0.014636	0.003849	0.003849	0.022727	0.222222	0.105263	0.041237	0.0	0.00578	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.012060	0.012065	0.003444	0.003444	0.022727	0.222222	0.105263	0.025773	0.0	0.00578	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.028667	0.028679	0.010940	0.010940	0.011364	0.111111	0.052632	0.072165	0.0	0.00000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.012060	0.012065	0.003849	0.003849	0.011364	0.111111	0.052632	0.036082	0.0	0.00000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.009885	0.009889	0.002634	0.002634	0.022727	0.222222	0.105263	0.030928	0.0	0.00000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.017991	0.017998	0.005065	0.005065	0.011364	0.111111	0.052632	0.067010	0.0	0.00000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.016212	0.016218	0.005673	0.005673	0.011364	0.111111	0.052632	0.041237	0.0	0.00000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

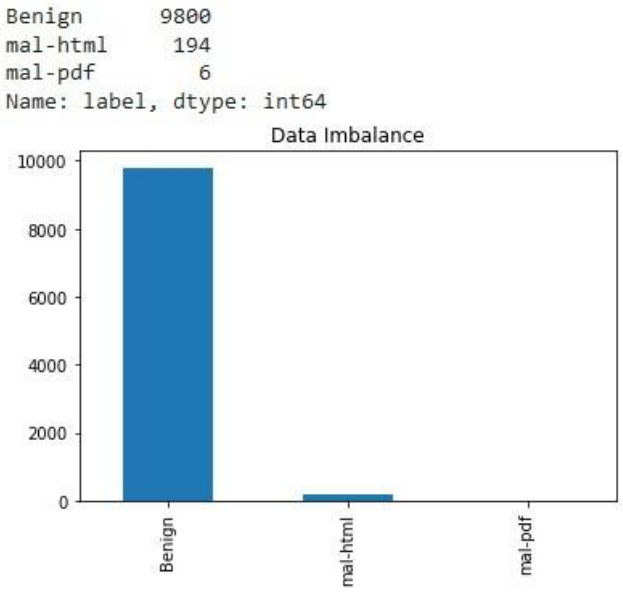
4.3 Pre-Processing

4.3.1 Analisa Dataset

Tahap ini di lakukan untuk analisa dataset PDF Malware untuk mengetahui informasi mengenai dataset tersebut

1. Dataset PDF malware terdiri dari 10.000 file PDF
2. Dataset memiliki dua tipe yaitu data objek dan data integer
3. Dataset memiliki 3 kelas yaitu kelas *benign, mal-pdf, dan non-malpdf*
4. Dataset tidak memiliki ruang kosong
5. Dataset termasuk kedalam golongan kelas Imbalance karena data tersebut berjumlah 9800 untuk *file benign*, 194 untuk *file non-pdfmal* dan 6 untuk *file mal-pdf*

Seperti yang terlihat di bawah ini merupakan hasil dari dataset dengan imbalance pada gambar 4.6



Gambar 4.6 *Dataset Imbalance*

4.3.1 Normalisasi

Proses Normalisasi bertujuan untuk menyamakan nilai angka yang terdapat pada dataset dalam skala yang sama sehingga angka-angka tersebut memiliki selisih minimum. Nilai dari dataset akan di ubah dari skala 0 menjadi skala 1. Hal tersebut menjadi acuan yang baik karena data yang di hasilkan dapat di seimbangkan.

4.4 Processing

4.4.1 *Resampling*

Proses *resampling* di bagi menjadi 2 tahap yang berbeda dalam penelitian ini, yang pertama yaitu penerapan *Oversampling* dan yang kedua yaitu penggabungan *Oversampling dan undersampling* dalam hal ini pembagian tersebut bertujuan untuk mengetahui perbedaan yang di hasilkan dari masiing-masing teknik yang selanjutnya akan di bandingkan untuk memperoleh hasil terbaik Proses *resampling* yang akan di gunakan adalah sebagai berikut;

Resampling dengan SMOTE yang di gunakan pada metode untuk resampling dengan SMOTE ini adalah penerapan dari Oversampling, proses SMOTE ini akan menghasilkan data baru secara sintetis. Data yang akan dibuat bisa mirip dengan data yang asli pada kelas minoritas.

Proses dari Oversampling ini akan menghasilkan jumlah data pada setiap kelasnya dengan jumlah yang sama data mayoritas yaitu sebanyak 9800 pada setiap kelasnya. Hasil dari proses *Resampling* menggunakan SMOTE dan undersampling menggunakan *NearMiss* bahwa jumlah data yang di hasilkan benign, mal-pdf, dan non-pdfmal. Yang telah terdistribusi telah *balance* dengan jumlah data 3000 dari hasil resampling dari masing-masing teknik dan pada masing-masing kelas. Hasil dapat di lihat pada gambar yang ada di bawah yaitu terlihat pada gambar 4.7

Gambar 4.7 *Dataset Balance*

4.4.2 Split Data

Split data training dan juga data testing merupakan proses yang di lakukan untuk melakukan pengujian model pada penelitian. Yang di lakukan yaitu proses pembagian data pada penelitian yaitu 20% data testing dan 80% data training.

4.5 Processing

Proses ini merupakan tahap yang di lakukan untuk klasifikasi PDF Malware dengan menggunakan Random Forest.

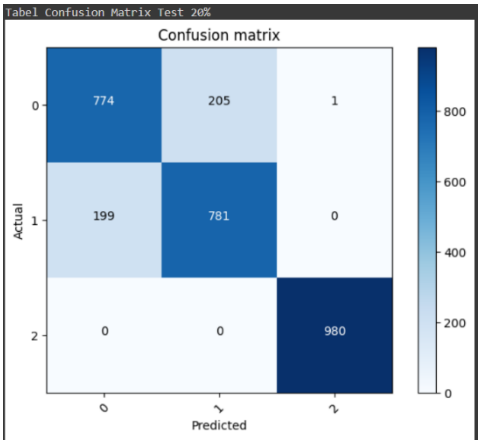
4.6 Hasil Percobaan pada klasifikasi Random Forest

Proses klasifikasi pada percobaan ini menggunakan metode Random Forest Classifier dan menggunakan GridSearchCV, dari hal tersebut dapat di ketahui hasil confussion matrix terlihat pada tabel 4.3

Tabel 4.3 Perbandingan Hasil Performa

n-estimator	Accuracy	Precision	Recall	F1-Score
20	86,22%	79,55%	78,98%	79,26%
40	86,22%	79,21%	79,45%	79,45%

Dari hasil percobaan maka di peroleh hasil akurasi yang terbaik ketika jumlah pada pohon adalah 40% yaitu menunjukkan sebesar 86,22%, kemudian max_dipth yang di gunakan yaitu 10, dan min_samples_split yaitu 3. Akurasi yang dapat di lihat pada gambar 4.8



Gambar 4.8 Confussion Matrix

Dapat kita ketahui hasil dari confussion matrix yaitu hasil performa yang di hasilkan dari presisi, recall, dan F1-score terlihat pada tabel 4.4

Tabel 4.4 performa confussion Matrix

	Presisi	Recall	F1-score
0	79,55%	78,98%	79,26%
1	79,21%	79,69%	79,45%
2	99,90%	100%	99,95%
Akurasi=86,22%			

4.7 Hasil Validasi dengan Stratified Kfold Cross Validation

Berikut adalah hasil yang telah di keluarkan menggunakan Stratified Kfold Cross Validation dengan 4 kali percobaan dengan hasil fold yaitu 3 fold, 5 fold, dan juga 7 fold dan 10 fold. Pada percobaan ini performa yang yang lebih baik yaitu dengan menggunakan fold 7.

1. Berikut merupakan hasil dari percobaan dengan split 3 kali dapat di lihat pada gambar 4.9

```
StratifiedKFold(n_splits=3, random_state=None, shuffle=False)
TRAIN: [ 3294  3295  3296 ... 10062 10063 10064] TEST: [   0    1    2 ... 7947 7948 7949]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [3294 3295 3296 ... 8006 8007 8008]
TRAIN: [   0    1    2 ... 8006 8007 8008] TEST: [ 6588  6589  6590 ... 10062 10063 10064]
```

Gambar 4.9 fold 3 kali

2. Berikut merupakan hasil yang di peroleh dari percobaan menngunakan split 5 kali seperti yang terlihat pada gambar 4.10

```
StratifiedKFold(n_splits=5, random_state=None, shuffle=False)
TRAIN: [ 1977  1978  1979 ... 10062 10063 10064] TEST: [   0    1    2 ... 7923 7924 7925]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [1977 1978 1979 ... 7958 7959 7960]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [3953 3954 3955 ... 7994 7995 7996]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [5929 5930 5931 ... 8086 8087 8088]
TRAIN: [   0    1    2 ... 8086 8087 8088] TEST: [ 7890  8033  8034 ... 10062 10063 10064]
```

Gambar 4.10 fold 5 kali

4. Berikut merupakan hasil yang di peroleh dari percobaan menggunakan split 7 kali seperti yang terlihat pada gambar 4.11

```
StratifiedKFold(n_splits=7, random_state=None, shuffle=False)
TRAIN: [ 1412  1413  1414 ... 10062 10063 10064] TEST: [   0    1    2 ... 7913 7914 7915]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [1412 1413 1414 ... 7938 7939 7940]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [2824 2825 2826 ... 7963 7964 7965]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [4236 4237 4238 ... 7989 7990 7991]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [5648 5649 5650 ... 8015 8016 8017]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [7059 7060 7061 ... 8651 8652 8653]
TRAIN: [   0    1    2 ... 8651 8652 8653] TEST: [ 7890  8044  8045 ... 10062 10063 10064]
```

Gambar 4.11 fold 7

5. Berikut merupakan hasil yang di peroleh dari percobaan split 10 kali seperti yang dapat di lihat pada gambar 4.12

```
StratifiedKFold(n_splits=10, random_state=None, shuffle=False)
TRAIN: [  989   990   991 ... 10062 10063 10064] TEST: [   0    1    2 ... 7906 7907 7908]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [ 989  990  991 ... 7924 7925 7926]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [1977 1978 1979 ... 7942 7943 7944]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [2965 2966 2967 ... 7960 7961 7962]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [3953 3954 3955 ... 7978 7979 7980]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [4941 4942 4943 ... 7995 7996 7997]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [5929 5930 5931 ... 8012 8013 8014]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [6917 6918 6919 ... 8086 8087 8088]
TRAIN: [    0     1     2 ... 10062 10063 10064] TEST: [8033 8034 8035 ... 9074 9075 9076]
TRAIN: [   0    1    2 ... 9074 9075 9076] TEST: [ 8051  8052  8053 ... 10062 10063 10064]
```

Gambar 4.12 fold 10

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah di lakukan dapat di simpulkan sebagai berikut;

1. Dataset yang di miliki berjumlah 10.000 dataset dengan perolehan 9800 file *benign*, 194 *file non-pdfmal* dan 6 *file pdf-mal*. Dari proses ektraksi yang telah di lakukan, menggunakan tools dari PDFiD menggunakan sistem operasi kali linux. Yang selanjutnya data yang telah di ekstraksi di ubah kedalam bentuk dataset .*cvs (Comma Separated Value)*. Dataset yang di miliki merupakan kelas imbalance atau data tidak seimbang hal tersebut merupakan permasalahan data maining yang perlu di lakukanya proseses keseimbangan menggunakan proses *Resampling* dengan penerapan *Oversampling* SMOTE dan *Undersampling*
2. Proses di lakukan untuk menghasilkan Performasi dan evaluasi pada penelitian ini menggunakan metode *Random Forest Classifier*.. Pada penelitian ini dapat di peroleh akurasi 86,22%, Precision 79,55%, Recall 78,98% dan F1-Score 79,26
3. Perolehan Hasil akurasi sebelum Resampling adalah sebesar 100% setelah di lakukan Proses Resampling dengan Oversampling maka nilai akurasi 86,22%

5.2 Saran

1. Dapat menerapkan teknik lainnya untuk mengatasi *imbalanced data*.
2. Untuk selanjutnya dapat mencoba menggunakan seleksi fitur lain agar mendapat performasi dan evaluasi hasil yang lebih baik.
3. Diharapkan untuk kasus selanjutnya jumlah *Dataset Malware* ditambah lagi agar perbandingan dengan *benign* tidak jauh berbeda. Kelas *imbalance*

DAFTAR PUSTAKA

[1] W. Deng, Z. Huang, J. Zhang, and J. Xu, "A Data Mining Based System for Transaction Fraud Detection," *2021 IEEE Int. Conf. Consum. Electron. Comput. Eng. ICCECE 2021*, pp. 542–545, 2021, doi: 10.1109/ICCECE51280.2021.9342376.

[2] S. D. S. K. Virgiawan A. Manoppo, Arie S. M. Lumenta, "Analisa Malware Menggunakan Metode Dynamic Analysis Pada Jaringan Universitas Sam Ratulangi," *J. Tek. Elektro Dan Komput.*, vol. 9, no. 3, pp. 181–188, 2020.

[3] F. C. C. Garcia, "Random Forest for Malware Classification," pp. 1–4.

[4] N. Afifah, D. Stiawan, and S. Nurmaini, "The Implementation of Deep Neural Networks Algorithm for Malware Classification," *Comput. Eng. Appl.*, vol. 8, no. 3, pp. 189–202, 2019.

[5] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Justindo, J. Sist. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 19–30, 2017, [Online]. Available: http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/1037

[6] S. Madan, S. Sofat, and D. Bansal, "Tools and Techniques for Collection and Analysis of Internet-of-Things malware: A systematic state-of-art review," *J. King Saud Univ. - Comput. Inf. Sci.*, no. 2022, doi: 10.1016/j.jksuci.2021.12.016.

[7] A. Beaudet, C. Escudero, and É. Zamaï, "Malicious anomaly detection approaches robustness in manufacturing ICSs," *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 146–151, 2021, doi: 10.1016/j.ifacol.2021.08.016.

[8] A. S. Rusdi, N. Widiyasono, and H. Sulastri, "Analisis Infeksi Malware Pada Perangkat Android Dengan Metode Hybrid Analysis," *Jl. Siliwangi No*, vol. 46115, no. 24, 2019.

[9] Z. Shen, M. U. Rehman, W. Chen, Y. Liu, J. Liu, and T. Zhong, "A Method

based on Modified PageRank-Algorithm for Measuring and Rating Android Malwares," *Procedia Comput. Sci.*, vol. 174, pp. 252–255, 2020, doi: 10.1016/j.procs.2020.06.081.

[10] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, and M. R. Sillero-Denamiel, "Variable selection for Naïve Bayes classification," *Comput. Oper. Res.*, vol. 135, p. 105456, 2021, doi: 10.1016/j.cor.2021.105456.

[11] N. Bala, A. Ahmar, W. Li, F. Tovar, A. Battu, and P. Bambarkar, "DroidEnemy: battling adversarial example attacks for Android malware detection," *Digit. Commun. Networks*, 2021, doi: 10.1016/j.dcan.2021.11.001.

[12] T. Olsson, M. Ericsson, and A. Wingkvist, "To automatically map source code entities to architectural modules with Naive Bayes," *J. Syst. Softw.*, vol. 183, p. 111095, 2022, doi: 10.1016/j.jss.2021.111095.

[13] M. H. Junejo, A. A. H. Ab Rahman, R. A. Shaikh, K. M. Yusof, D. Kumar, and I. Memon, "Lightweight Trust Model with Machine Learning scheme for secure privacy in VANET," *Procedia Comput. Sci.*, vol. 194, pp. 45–59, 2021, doi: 10.1016/j.procs.2021.10.058.

[14] R. Upadhyay, U. R. Bhatt, and H. Tripathi, "DDOS Attack Aware DSR Routing Protocol in WSN," *Phys. Procedia*, vol. 78, no. December 2015, pp. 68–74, 2016, doi: 10.1016/j.procs.2016.02.012.

[15] J. Kim and P. R. Kumar, "Security of control systems with erroneous observations," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2225–2230, 2020, doi: 10.1016/j.ifacol.2020.12.008.

[16] X. Xie, Q. Lu, A. K. Parlikad, and J. M. Schooling, "Digital twin enabled asset anomaly detection for building facility management," *IFAC-PapersOnLine*, vol. 53, no. 3, pp. 380–385, 2020, doi: 10.1016/j.ifacol.2020.11.061.

[17] V. Syrris and D. Geneiatakis, "On machine learning effectiveness for malware detection in Android OS using static analysis data," *J. Inf. Secur. Appl.*, vol. 59, no. May, p. 102794, 2021, doi: 10.1016/j.jisa.2021.102794.

[18] S. Farhana, "Classification of Academic Performance for University Research Evaluation by Implementing Modified Naive Bayes Algorithm," *Procedia*

Comput. Sci., vol. 194, pp. 224–228, 2021, doi: 10.1016/j.procs.2021.10.077.

[19] S. R. T. Mat, M. F. A. Razak, M. N. M. Kahar, J. M. Arif, and A. Firdaus, "A Bayesian probability model for Android malware detection," *ICT Express*, no. xxxx, 2022, doi: 10.1016/j.icte.2021.09.003.

[20] M. K. Ishak and F. K. Khan, "Unique message authentication security approach based controller area network (can) for anti-lock braking system (abs) in vehicle network," *Procedia Comput. Sci.*, vol. 160, pp. 93–100, 2019, doi: 10.1016/j.procs.2019.09.448.

[21] Hubert, P. Phoenix, R. Sudaryono, and D. Suhartono, "Classifying Promotion Images Using Optical Character Recognition and Naïve Bayes Classifier," *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 498–506, 2021, doi: 10.1016/j.procs.2021.01.033.

[22] J. Yuste, E. G. Pardo, and J. Tapiador, "Optimization of code caves in malware binaries to evade Machine Learning detectors," *Comput. Secur.*, p. 102643, 2022, doi: 10.1016/j.cose.2022.102643.

[23] N. Zakeya, K. Ségla, T. Chamseddine, and B. B. Alvine, "Probing AndroVul dataset for studies on Android malware classification," *J. King Saud Univ. - Comput. Inf. Sci.*, no. 2021, doi: 10.1016/j.jksuci.2021.08.033.

[24] W. Casbolt, I. Esnaola, and B. Jones, "Denial of service attacks on control systems with packet loss," *IFAC-PapersOnLine*, vol. 53, pp. 3488–3495, 2020, doi: 10.1016/j.ifacol.2020.12.1699.

[25] N. Al Sarah, F. Y. Rifat, M. S. Hossain, and H. S. Narman, "An Efficient Android Malware Prediction Using Ensemble machine learning algorithms," *Procedia Comput. Sci.*, vol. 191, no. 2019, pp. 184–191, 2021, doi: 10.1016/j.procs.2021.07.023.

[26] A. Yudhana, D. Sulistyo, and I. Mufandi, "GIS-based and Naïve Bayes for nitrogen soil mapping in Lendah, Indonesia," *Sens. Bio-Sensing Res.*, vol. 33, p. 100435, 2021, doi: 10.1016/j.sbsr.2021.100435.

[27] P. Bhat and K. Dutta, "A multi-tiered feature selection model for android malware detection based on Feature discrimination and Information Gain," *J.*

King Saud Univ. - Comput. Inf. Sci., 2021, doi: 10.1016/j.jksuci.2021.11.004.

[28] Y. Cahyaningrum, I. R. Widiasari, and J. O. Notohamidjojo, "Analisis Performa Container Berplatform Docker atas Serangan Malicious Software (Malware)," pp. 47–54, 2020.

[29] A. Saleh, "Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga," *Creat. Inf. Technol. J.*, vol. 2, no. 3, pp. 207–217, 2015.

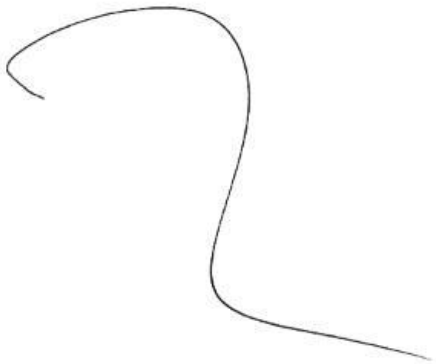
[30] G. Fiore, E. De Santis, and M. D. Di Benedetto, "Secure Mode Distinguishability for Switching Systems Subject to Sparse Attacks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9361–9366, 2017, doi: 10.1016/j.ifacol.2017.08.1442.

LAMPIRAN

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS SRIWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM KOMPUTER
Jalan Palembang – Prabumulih Km. 32 Indralaya Kabupaten Ogan Ilir Kode Pos 30662
Telepon (0711)7072729, 379249, 581700 Faksimili (0711) 379248, 581710
Pos-el : info@ilkom.unsri.ac.id

FORM PERBAIKAN UJIAN SKRIPSI (TUGAS AKHIR II)

Nama Mahasiswa	: Alfiah Nur Fatmawati
NIM	: 09011181823131.
Jurusan	: Sistem Komputer
Hari / Tanggal	: Jum'at / 12 Januari 2024
Waktu	: 14:30 s.d 15:00 WIB
Judul Tugas Akhir	: Penerapan Random Forest Classifier untuk Deteksi PDF Malware pada Layanan Agregator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti
Pembimbing	: Prof. Deris Stiawan, M.T., Ph.D. Nurul Afifah, M.Kom
Perbaikan/Saran	:



Jangka Waktu Perbaikan : 2 minggu

Telah diperbaiki sesuai dengan saran dan koreksi tim penguji ujian komprehensif.

No.	Nama Penguji	Status Penguji	Tanda Tangan
1.	Prof. Deris Stiawan, M.T., Ph.D.	Pembimbing I	

Palembang, 12 Januari 2024
Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.
NIP 196612032006041001

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS SRIWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM KOMPUTER
Jalan Palembang – Prabumulih Km. 32 Indralaya Kabupaten Ogan Ilir Kode Pos 30662
Telepon (0711)7072729, 379249, 581700 Faksimili (0711) 379248, 581710
Pos-el : info@ilkom.unsri.ac.id

FORM PERBAIKAN UJIAN SKRIPSI (TUGAS AKHIR II)

Nama Mahasiswa	: Alfiah Nur Fatmawati
NIM	: 09011181823131.
Jurusan	: Sistem Komputer
Hari / Tanggal	: Jum'at / 12 Januari 2024
Waktu	: 14:30 s.d 15:00 WIB
Judul Tugas Akhir	: Penerapan Random Forest Classifier untuk Deteksi PDF Malware pada Layanan Agregator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti
Pembimbing	: Prof. Deris Stiawan, M.T., Ph.D. Nurul Afifah, M.Kom
Perbaikan/Saran	:

- Selesaikan revisi penguji

Jangka Waktu Perbaikan : 2 minggu

Telah diperbaiki sesuai dengan saran dan koreksi tim penguji ujian komprehensif.

No.	Nama Penguji	Status Penguji	Tanda Tangan
1.	Nurul Afifah, M.Kom	Pembimbing II	

Palembang, 12 Januari 2024
Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.
NIP 196612032006041001

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS SRIWIJAYA
FAKULTAS ILMU KOMPUTER
JURUSAN SISTEM KOMPUTER
Jalan Palembang – Prabumulih Km. 32 Indralaya Kabupaten Ogan Ilir Kode Pos 30662
Telepon (0711)7072729, 379249, 581700 Faksimili (0711) 379248, 581710
Pos-el : info@ilkom.unsri.ac.id

FORM PERBAIKAN UJIAN SKRIPSI (TUGAS AKHIR II)

Nama Mahasiswa	: Alfiah Nur Fatmawati
NIM	: 09011181823131.
Jurusan	: Sistem Komputer
Hari / Tanggal	: Jum'at / 12 Januari 2024
Waktu	: 14:30 s.d 15:00 WIB
Judul Tugas Akhir	: Penerapan Random Forest Classifier untuk Deteksi PDF Malware pada Layanan Agregator Garba Rujukan Digital (GARUDA) Kemendikbud Dikti
Pembimbing	: Prof. Deris Stiawan, M.T., Ph.D. Nurul Afifah, M.Kom
Perbaikan/Saran	:

- Perbaiki penulisan.
- tambahkan nilai skewness, untuk beberapa variabel
- ~~masih~~ K-Fold dimasukkan dlm kodingan

Jangka Waktu Perbaikan : 2 minggu

Telah diperbaiki sesuai dengan saran dan koreksi tim penguji ujian komprehensif.

No.	Nama Penguji	Status Penguji	Tanda Tangan
1.	Rossi Passarella, M.Eng	Penguji	

Palembang, ~~12 Januari~~ 2024
Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.
NIP 196612032006041001

PENERAPAN RANDOM FOREST CLASSIFIER UNTUK DETEKSI PDF MALWARE PADA LAYANAN AGREGATOR GARBA RUJUKAN DIGITAL (GARUDA) KEMENDIKBUD DIKTI

by 09011181823131 ALFIAH NUR FATMAWATI

Submission date: 19-Jan-2024 06:38PM (UTC+0700)
Submission ID: 2269742312
File name: kan_Digital_GARUDA_Kemendikbud_Dikti_-_alfiah_nur_fatmawati.docx (75.09K)
Word count: 4148
Character count: 25687

PENERAPAN RANDOM FOREST CLASSIFIER UNTUK DETEKSI PDF MALWARE PADA LAYANAN AGREGATOR GARBA RUJUKAN DIGITAL (GARUDA) KEMENDIKBUD DIKTI

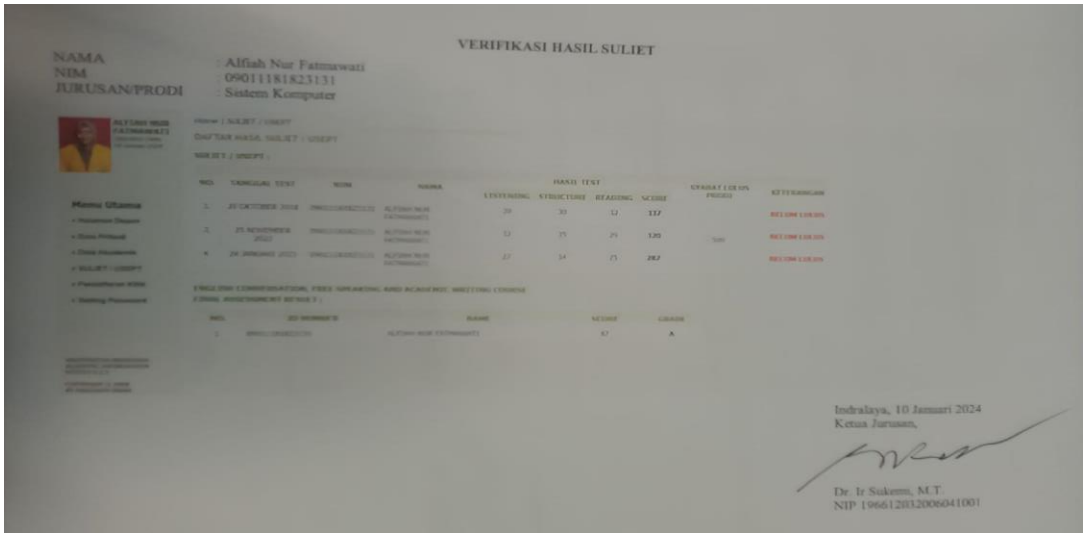
ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	jurnal.sar.ac.id Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		



VERIFIKASI HASIL SULIET

NAMA : Alfiah Nur Fatmawati
NIM : 09011181823131
JURUSAN/PRODI : Sistem Komputer

Indralaya, 10 Januari 2024
Ketua Jurusan,

Dr. Ir Sukemi, M.T.
NIP 196612032006041001