Identifikasi Jenis *File* pada Artefak Digital Menggunakan Algoritma *Machine Learning*

Ihsan Fawzan
21917030

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Digital Forensik
Program Studi Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2024

Lembar Pengesahan Pembimbing

Identifikasi Jenis *File* pada Artefak Digital Menggunakan Algoritma *Machine Learning*

Ihsan Fawzan
21917030

Yogyakarta, Agustus, 2024

Pembimbing

Dr. Ahmad Luthfi, S.Kom., M.Kom.

Lembar Pengesahan Penguji

Identifikasi Jenis *File* pada Artefak Digital Menggunakan Algoritma *Machine Learning*

Ihsan Fawzan

21917030

Yogyakarta, Agustus, 2024

Tim Penguji,

Dr. Ahmad Luthfi, S.Kom., M.Kom
Ketua

Dr. Yudi Prayudi, S.Si., M.Kom.
Anggota I

Irving Vitra Paputungan, S.T., M.Sc., Ph.D
Anggota II

18/09/2024

Mengetahui,
Ketua Program Studi Informatika Program Magister
Universitas Islam Indonesia

Irving Vitra Paputungan, S.T., M.Sc., Ph.D

Abstrak

Identifikasi Jenis *File* pada Artefak Digital Menggunakan Algoritma *Machine Learning*

Sebanyak 8.831 kasus terkait kejahatan siber terjadi di Indonesia pada tahun 2022. Kejahatan terbanyak yaitu adanya manipulasi data otentik yang salah satu caranya adalah merusak barang bukti digital, sehingga akan menyulitkan ahli forensik untuk melakukan analisis terhadap artefak digital tersebut. Berdasarkan permasalahan yang telah terjadi maka penulis menemukan permasalahan yang akan diteliti yaitu bagaimana mengidentifikasi jenis *file* yang tidak terdefinisi pada artefak digital menggunakan *Machine learning*. Penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasikan jenis *file* yang tidak terdefinisi pada artefak digital menggunakan *machine learning* sehingga dapat diketahui jenis *file* apa yang akan dianalisis. Metode penelitian yang penulis gunakan yaitu metode penelitian analisis deskriptif. Analisis deskriptif adalah statistik yang digunakan untuk menganalisis data dengan cara mendeskripsikan atau menggambarkan data yang telah terkumpul dengan cara disusun, diolah, dan dianalisis untuk memberikan gambaran mengenai masalah yang ada. Ada beberapa cara untuk mengidentifikasi jenis *file* di antaranya adalah identifikasi secara *content-based* yang melakukan kalkulasi statistik terhadap isi dari *file* yang *corrupt* atau tidak terdefinisi dengan bantuan *machine learning*. *Dataset NapierOne* dipisahkan berdasarkan jenis *file* sehingga saat melakukan ekstraksi fitur jenis *file* tersebut dapat di klasifikasikan. *Byte frequency distribution* adalah persamaan aritmatika yang berurusan dengan *byte* yang berada di kisaran *0x00* dan *0xFF* dan dihitung jumlah kemunculannya. *Byte frequency distribution* digunakan untuk melakukan kalkulasi dan ekstraksi fitur yang menghasilkan 256 fitur dari sebuah *file*. *K-Nearest Neighbor* adalah suatu jenis algoritma *supervised Machine learning* yang digunakan untuk klasifikasi data ke kelas paling mirip pada *dataset*. Berdasarkan *dataset* yang telah dikumpulkan dan dikategorikan sesuai jenis *file*, maka dilakukan ekstraksi fitur menggunakan *Byte frequency distribution*. Fitur tersebut kemudian digunakan sebagai masukan ke dalam algoritma *K-Nearest Neighbor* sehingga di kelompokan berdasarkan jenis *file* yang terdefinisi. *K-Nearest Neighbor* menghasilkan sebuah model yang dapat digunakan untuk melakukan prediksi terhadap sebuah *file* yang tidak terdefinisi berdasarkan fitur-fitur yang ada pada *file* tersebut sehingga dapat diketahui jenis *file*nya berdasarkan fitur yang ada.

Kata kunci

digital forensik, file type identification, K-Nearest Neighbor, Machine learning



Abstract

Identification of File Types in Digital Artifacts Using the Machine Learning Algorithm

A total of 8,831 cases related to cyber crime occurred in Indonesia in 2022. The most common crime is manipulation of authentic data, one way of which is destroying digital evidence, making it difficult for forensic experts to carry out analysis of these digital artifacts. Based on the problems that have occurred, the author found a problem to be researched, namely how to identify undefined file types on digital artifacts using Machine learning. This research aims to identify and classify undefined file types in digital artifacts using Machine learning so that we can know what type of files will be analyzed. The research method that the author uses is the descriptive analysis research method. Descriptive analysis is statistics used to analyze data by describing or illustrating the data that has been collected by compiling, processing and analyzing to provide an overview of the existing problem. There are several ways to identify file types, including content-based identification which performs statistical calculations on the contents of corrupt or undefined files with the help of Machine learning. The NapierOne dataset is separated by file type so that when extracting features the file types can be classified. Byte Frequency Distribution is an arithmetic equation that deals with bytes that are in the range 0x00 and 0xFF and counts the number of occurrences. Byte frequency distribution is used to perform calculations and feature extraction which produces 256 features from a file. K-Nearest Neighbor is a type of supervised Machine learning algorithm that is used to classify data into the most similar classes in the dataset. Based on the dataset that has been collected and categorized according to file type, feature extraction is carried out using Byte Frequency Distribution. These features are then used as input into the K-Nearest Neighbor algorithm so that they are grouped based on defined file types. K-Nearest Neighbor produces a model that can be used to make predictions about an undefined file based on the features in the file so that the file type can be identified based on the existing features.

Keywords

digital forensics, file type identification, K-Nearest Neighbor, Machine learning

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Agustus, 2024

Ihsan Fawzan,

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

(Ihsan Fawzan, Ahmad Luthfi, 2024), IDENTIKASI JENIS *FILE* PADA ARTEFAK DIGITAL MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBOR*, Jurnal Ilmiah Penelitian dan Pembelajaran Informatika (JIPI) Vol.10 No.3 2025.

Kontributor	Jenis Kontribusi
Ihsan Fawzan	Mendesain eksperimen (70%) Menulis *paper* (80%)
Ahmad Luthfi	Mendesain eksperimen (30%) Menulis dan mengedit *paper* (20%)

Kata Pengantar

Bismillahirrahmanirrahim

Puji syukur penulis sampaikan kehadirat Allah SWT., Tuhan Yang Maha Pemurah lagi Maha Penyayang. Berkat rahmat, hidayah, dan inayah-Nya, akhirnya penulis dapat menyelesaikan laporan tesis yang berjudul "Identifikasi Jenis *File* pada Artefak Digital Menggunakan Algoritma *Machine Learning*".

Tesis ini merupakan syarat wajib yang harus ditempuh dalam mencapai Magister Strata-2 pada Program Studi Magister Informatika. Penulisan laporan tesis ini terselesaikan karena bantuan dari berbagai pihak. Untuk itu, penulis menyampaikan terima kasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan laporan tesis ini.

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., Selaku Rektor Universitas Islam Indonesia.
2. Bapak Hari Purnomo, Prof., Dr., Ir., M.T., IPU, ASEAN.Eng, Selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Irving Vitra Paputungan, S.T., M.Sc., Ph.D Selaku Ketua Program Studi Magister Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Dr. Ahmad Luthfi, S.kom., M.Kom. Selaku Dosen Pembimbing 1 yang berkenan membimbing dan mengarahkan penulis, sehingga Penelitian Tesis ini dapat selesai.
5. Semua pihak yang tidak dapat penulis sebutkan satu persatu, baik secara langsung maupun tidak langsung membantu dalam penulisan tesis ini.

Semoga atas bantuan dan kerja sama yang telah diberikan menjadi amal baik dan mendapat balasan dari Allah SWT. Penulis menyadari bahwa penelitian ini jauh dari sempurna, untuk itu diperlukan saran dan masukan demi sempurnanya penyusunan laporan tesis ini. Akhir kata, penulis berharap semoga laporan tesis ini dapat bermanfaat bagi semua yang membutuhkan.

Yogyakarta, Agustus 2024

Ihsan Fawzan

Daftar Isi

Lembar Pengesahan Pembimbing i
Lembar Pengesahan Penguji ii
Abstrak iii
Abstract v
Pernyataan Keaslian Tulisan vii
Daftar Publikasi viii
Kata Pengantar ix
Daftar Isi x
Daftar Tabel xii
Daftar Gambar xiii
BAB 1 Pendahuluan 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Tujuan Penelitian 4
1.4 Batasan Penelitian 4
1.5 Manfaat Penelitian 4
a. Manfaat secara ilmiah 4
b. Manfaat secara sosial 4
1.6 Sistematika Penulisan 5
BAB 2 Tinjauan Pustaka 7
2.1 Landasan Teori 7
2.1.1 Digital Forensik 7
2.1.2 Bukti Digital 8
2.1.3 Deteksi Jenis *File* 10
2.1.4 *K-Nearest Neighbor* 13

2.1.5 Support Vector Machine 15
2.1.6 *Byte Frequency Distribution* 17
2.1.7 *File* Corrupt 19
2.2 Tinjauan Pustaka 21
BAB 3 Metodologi 26
3.1 Metodologi Penelitian 26
3.1.1 Identifikasi Masalah 27
3.1.2 Tinjauan Pustaka 27
3.1.3 Mencari *Dataset* 27
3.1.4 Mencari Algoritma 28
3.1.5 Implementasi Algoritma 30
3.1.6 Langkah-langkah Penelitian 31
BAB 4 Hasil dan Pembahasan 33
4.1 Analisis Kebutuhan 33
4.2 Skenario Kasus 33
4.3 Persiapan Dataset 34
4.4 Ekstraksi Fitur 34
4.5 Mencari Nilai K dan C Optimum 35
4.6 Membuat Model *Machine Learning* 35
4.7 Implementasi Model *Machine Learning* 36
BAB 5 Kesimpulan dan Saran 42
5.1 Kesimpulan 42
5.2 Saran 42
Daftar Pustaka 43
Lampiran 49

Daftar Tabel

Tabel 2.1 Tinjauan Pustaka 21
Tabel 3.1 Contoh hasil ekstraksi fitur dengan *Byte Frequency Distribution* 31
Tabel 4.1 Spesifikasi hardware dan software 33

Daftar Gambar

Gambar 3.1 Metodologi penelitian 26
Gambar 3.2 *GovDocs1* Comparison 28
Gambar 3.3 Grafik Cross Validation SVM 29
Gambar 3.4 Grafik Cross Validation KNN 30
Gambar 3.5 Alur implementasi *K-Nearest Neighbor* terhadap *dataset* 30
Gambar 3.6 Sebaran data pada *file* jpg 31
Gambar 4.1 Tampilan awal aplikasi 39
Gambar 4.2 Halaman home aplikasi 39
Gambar 4.3 Halaman identifikasi *file* 40
Gambar 4.4 Halaman identifikasi *file* setelah input *file* 40
Gambar 4.5 Halaman identifikasi *file* setelah pemrosesan dengan *machine learning* 41

BAB 1
Pendahuluan

1.1 Latar Belakang

Kejahatan siber merujuk pada penggunaan teknologi komputer atau jaringan online untuk melakukan tindakan kriminal, termasuk penipuan, pencurian identitas, serta ancaman dan intimidasi (Hawdon, 2021). Di Indonesia, kejahatan siber diatur melalui Undang-Undang tentang Informasi dan Transaksi Elektronik, yaitu Undang-Undang Nomor 11 Tahun 2008 dan Undang-Undang Nomor 19 Tahun 2016.

Berdasarkan statistik (*Kejahatan Siber Di Indonesia Naik Berkali-Kali Lipat | Pusiknas Bareskrim Polri*, n.d.), sepanjang tahun 2022 terdapat 8.831 kasus terkait kejahatan siber dan menindak 8.372 orang yang menjadi terlapor. Kasus dengan jumlah penindakan terbanyak ada pada kasus manipulasi data otentik yang melanggar Undang-undang nomor 11 Tahun 2008 Pasal 35 yang berbunyi "Setiap orang dengan sengaja dan tanpa hak atau melawan hukum melakukan manipulasi, penciptaan, perubahan, penghilangan, pengrusakan informasi elektronik dan/atau dokumen elektronik dengan tujuan agar informasi elektronik dan/atau dokumen elektronik tersebut dianggap seolah-olah data yang otentik."

Serangan manipulasi data terjadi ketika penjahat tidak mengambil data, namun melakukan penyesuaian halus dan tersembunyi pada data untuk mendapatkan keuntungan atau efek tertentu. Modifikasi data ini dapat merugikan organisasi seperti halnya pelanggaran data. Manipulasi data dapat mengakibatkan distorsi persepsi karena pergeseran data yang menyebabkan kerugian finansial atau bahkan potensi hilangnya nyawa tergantung pada sistem yang dimaksud dan jenis data yang diubah (Nettles et al., 2019).

Dalam konteks digital forensik, ada beberapa penyebab terjadinya manipulasi data diantaranya adalah serangan virus, *malware*, dan malfungsi sistem sehingga data tidak terjamin keasliannya. *WhisperGate* merupakan salah satu *malware* yang dirancang untuk melakukan manipulasi data untuk merusak *file* dengan menimpa satu MB pertama dengan 0xcc dan menambahakn ekstensi secara acak (Taylor et al., 2024).

Selain *WhisperGate*, ada *malware* lain yang dapat digunakan untuk melakukan manipulasi data di antaranya adalah *PowerDuke* dan *Revil*. *PowerDuke* merupakan

backdoor yang digunakan oleh APT29 pada tahun 2016, umumnya disebarkan melalui lampiran *file* Microsoft Word atau Excel yang mengandung makro berbahaya (*PowerDuke, Software S0139 | MITRE ATT&CK®*, 2020). *PowerDuke* memiliki kemampuan untuk melakukan manipulasi data dengan perintah untuk menulis data acak di seluruh *file* dan menghapusnya. *Revil* merupakan *Ransomware-as-a-Service* (RaaS) yang menyerang sektor manufaktur, transportasi dan elektronik sejak tahun 2019 (Millington, 2024). Teknik yang digunakan untuk manipulasi data termasuk mengenkripsi *file* pada *host* target dan kemudian menjalankan *query* pada *registry* untuk menemukan ekstensi *file* acak yang akan ditambahkan ke *file* yang telah dienkripsi.

Digital forensik merupakan bagian dari ilmu forensik yang melakukan serangkaian proses untuk dapat menganalisis bukti digital maupun bukti elektronik. (Kent et al., 2006) menyebutkan ada 4 langkah utama untuk melakukan proses digital forensik yaitu *Collection, Examination, Analysis*, dan *Reporting. Examination* merupakan ranah yang memperlihatkan bahwa *tools* dan keahlian seorang analis digital forensik berperan penting untuk mengidentifikasi dan mengekstrak informasi dari data yang didapatkan serta menjaga integritas data (Kent et al., 2006). Sebelum melanjutkan dengan analisis yang lebih mendalam, ahli forensik harus terlebih dahulu mengidentifikasi jenis *file* yang akan dianalisis untuk memastikan penggunaan *tools* yang tepat dalam investigasi kejahatan siber.

Menurut (Meghanathan, 2013) dalam (Karampidis et al., 2015) salah satu cara untuk melakukan identifikasi jenis *file* adalah menggunakan analisis *content-based*. Analisis ini memanfaatkan perhitungan konten *file* untuk mengidentifikasi pola atau *pattern* secara statistik yang kemudian digunakan untuk proses klasifikasi data (Konaray et al., 2019). Analisis *content-based* dilakukan dengan memeriksa isi *file* yang telah dimanipulasi secara statistik dan membandingkannya dengan *file* asli lainnya. Proses ini memerlukan *dataset file* asli sebagai referensi untuk mengidentifikasi jenis *file* dari *file* yang telah dimodifikasi. Metode ini memungkinkan penggunaan informasi yang terdapat dalam *file* untuk menghasilkan pengelompokan yang lebih akurat berdasarkan karakteristik yang terukur. *Machine learning* dapat dimanfaatkan untuk melakukan proses analisis statistik guna memudahkan analisis lebih lanjut pada *file* yang telah dimanipulasi.

Penggunaan *machine learning* untuk melakukan klasifikasi telah dilakukan pada penelitian terdahulu di antaranya penggunaan algoritma *Convolutional Neural Network* (A. Bhat et al., 2021; Hiester, 2018; Mittal et al., 2021; Neaimi et al., 2020), *Feedforward Neural Network* (A. Bhat et al., 2021; Hiester, 2018; Vulinovic et al., 2019), *Multilayer*

Perceptron (Masoumi et al., 2021; Venkata & Green, 2019), *Support Vector Machine* (Masoumi et al., 2021; F. Wang et al., 2018; C. Zhang & Green, 2015), *Decission Tree* (Nguyen et al., 2017), dan *K-Nearest Neighbor* (Haque & Tozal, 2022; Masoumi et al., 2021; Y. Wang et al., 2018). Namun, sebagian besar penelitian yang telah dilakukan bergantung pada *dataset* lama seperti *GovDocs1* yang dikeluarkan pada tahun 2009 sehingga perlu ada penelitian baru yang menggunakan *dataset* terbaru. *NapierOne* adalah solusi keamanan siber modern yang mengelola berbagai jenis *file*, dengan salah satu fungsinya adalah untuk mendeteksi *ransomware* dan melakukan analisis forensik digital (Davies et al., 2022). *NapierOne* dikembangkan untuk mengatasi masalah dalam hal reproduktifitas dan meningkatkan konsistensi melalui penyederhanaan proses replikasi dan pengulangan penelitian serta terinspirasi oleh *GovDocs1* dan dirancang untuk berfungsi sebagai pelengkap bagi kumpulan data asli tersebut.

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja dua algoritma terkemuka, yaitu K-Nearest Neighbors (KNN) dan Support Vector Machines (SVM), dalam konteks klasifikasi jenis *file*. Algoritma KNN, yang dikenal karena kesederhanaan dan efektivitasnya, mengklasifikasikan data berdasarkan suara mayoritas dari tetangga terdekat dalam ruang fitur, menjadikannya sangat cocok untuk menangani berbagai jenis dataset dan menangkap pola lokal tanpa memerlukan struktur model yang telah ditetapkan sebelumnya. Sebaliknya, SVM menawarkan pendekatan yang kuat dengan mencari hyperplane optimal yang memaksimalkan jarak antara kelas-kelas dalam ruang berdimensi tinggi, serta menangani batas keputusan yang kompleks dan non-linear melalui fungsi kernel. Dengan menggabungkan kekuatan KNN dalam pengambilan keputusan lokal dan kemampuan SVM dalam optimasi global, penelitian ini memanfaatkan keunggulan kedua algoritma untuk meningkatkan kinerja klasifikasi dan menghasilkan hasil yang lebih andal.

Dalam studi ini, kami juga akan membandingkan efektivitas KNN dan SVM dalam mengidentifikasi dan mengklasifikasikan berbagai tipe *file* berdasarkan konten mereka. Evaluasi dilakukan menggunakan pendekatan cross-validation untuk memastikan penilaian yang objektif dan akurat terhadap performa masing-masing algoritma. Penelitian ini menghadirkan inovasi dengan penerapan dataset NapierOne yang dirilis pada tahun 2022, menggantikan dataset GovDocs1 yang lebih lama dari tahun 2009. Dengan memanfaatkan algoritma KNN yang menerapkan metode content-based melalui ekstraksi fitur menggunakan *Byte* Frequency Distribution, serta memperbarui dataset ke NapierOne yang

dirilis pada tahun 2022, penelitian ini bertujuan untuk mengembangkan model baru yang lebih komprehensif dan terpercaya dalam klasifikasi jenis *file*.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian yang telah diungkapkan, maka penulis merumuskan masalah sebagai berikut.

Bagaimana mengidentifikasi jenis *file* yang tidak terdefinisi pada artefak digital menggunakan *machine learning*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, maka tujuan penelitian ini adalah mengidentifikasi dan mengklasifikasikan jenis *file* yang tidak terdefinisi pada artefak digital menggunakan *machine learning* sehingga dapat diketahui jenis *file* apa yang akan dianalisis.

1.4 Batasan Penelitian

Batasan pada penelitian ini yaitu dataset yang digunakan berasal dari NapierOne dan menerapkan algoritma *K-Nearest Neighbor* dan *Support Vector Machine* untuk melakukan klasifikasi. Proses ini menggunakan *byte frequency distribution* sebagai metode ekstraksi fitur yang akan dihitung. Dalam analisis ini, file yang tidak terdefinisi adalah file yang telah dimanipulasi. Penelitian ini hanya mengklasifikasi jenis file tanpa melakukan pemulihan file, serta fokus pada klasifikasi file tunggal di dalam file yang tidak terdefinisi.

1.5 Manfaat Penelitian

a. Manfaat secara ilmiah

Secara teoritis penelitian ini dapat memperluas dan mengembangkan keilmuan mengenai bagaimana mengidentifikasi jenis *file* pada artefak digital menggunakan *machine learning* sehingga memudahkan saat proses analisis bukti digital secara lebih lanjut.

b. Manfaat secara sosial

Secara praktis diharapkan dapat memberikan gambaran tentang pelaksanaan proses mengidentifikasi jenis *file* pada artefak digital menggunakan *machine learning* sehingga memudahkan saat proses analisis bukti digital secara lebih lanjut.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini dijelaskan sebagai berikut.

BAB 1 PENDAHULUAN

Menjelaskan tentang latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan penelitian, dan manfaat penelitian. Bab ini memberikan konteks dan dasar yang diperlukan untuk memahami alasan di balik penelitian ini serta tujuan yang ingin dicapai. Selain itu, bab ini juga menetapkan batasan ruang lingkup penelitian untuk menjaga fokus dan relevansi.

BAB 2 TINJAUAN PUSTAKA

Menjelaskan tentang landasan teori dan kajian pustaka. Di dalam bab ini, teori-teori yang mendasari penelitian diuraikan secara mendetail, serta membahas tinjauan pustaka terkait untuk memberikan wawasan tentang perkembangan terkini dalam bidang yang diteliti. Bab ini bertujuan untuk menyusun kerangka teori yang mendukung analisis dan interpretasi data penelitian.

BAB 3 METODOLOGI

Menjelaskan tentang metodologi penelitian dan tahapan-tahapan yang dilakukan. Bab ini mencakup desain penelitian, metode pengumpulan data, serta prosedur analisis yang digunakan. Penjelasan rinci mengenai setiap tahapan penelitian bertujuan untuk memastikan transparansi dan reproduktifitas hasil.

BAB 4 HASIL DAN PEMBAHASAN

Menjelaskan tentang hasil dan pembahasan. Bab ini menyajikan temuan utama dari penelitian serta menganalisis hasil tersebut dalam konteks teori dan kajian pustaka yang telah dibahas sebelumnya. Pembahasan ini bertujuan untuk menginterpretasikan data dan menghubungkannya dengan tujuan penelitian.

BAB 5 KESIMPULAN DAN SARAN

Menjelaskan tentang kesimpulan. Bab ini menyajikan ringkasan dari hasil penelitian, memberikan kesimpulan yang diambil, serta menyarankan implikasi praktis dan arah penelitian selanjutnya. Kesimpulan ini bertujuan untuk merangkum temuan utama dan menggarisbawahi kontribusi penelitian terhadap bidang studi yang bersangkutan..

BAB 2
Tinjauan Pustaka

2.1 Landasan Teori

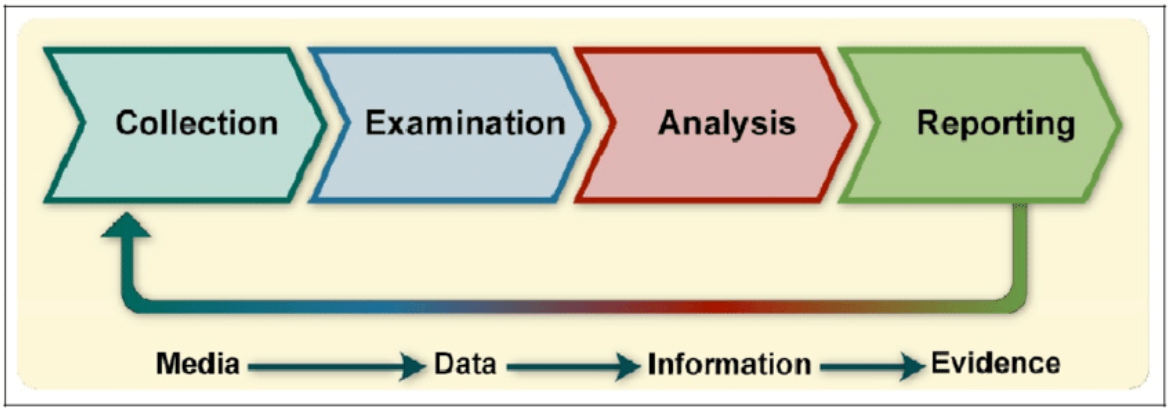
2.1.1 Digital Forensik

Digital forensik merupakan aplikasi bidang ilmu pengetahuan dan teknologi komputer untuk kepentingan pembuktian hukum (*pro justice*), yang di dalamnya berfungsi untuk membuktikan kejahatan berteknologi tinggi atau *computer crime* secara *scientific* (ilmiah) hingga bisa mendapatkan bukti-bukti digital yang berguna untuk menjerat pelaku kejahatan tersebut (Nuh Al-Azhar, 2012).

(Raharjo, 2013) berpendapat bahwa forensik digital merupakan bagian dari ilmu forensik yang melingkupi penemuan dan investigasi data yang ditemukan pada perangkat digital. Pendapat lain memaparkan bahwa digital forensik merupakan sebuah penilaian yang andal atas kredibilitas teknis bukti digital dengan kriteria berdasarkan pengetahuan ilmiah serta adanya kebutuhan yang tepat tentang tata cara penanganan data yang dapat dijadikan sebagai barang bukti dalam perkara pidana (Lewulis, 2021).

Dalam ranah digital forensik, terdapat *framework* atau kerangka kerja yang dapat dijadikan sebagai acuan untuk melakukan kegiatan-kegiatan dari mulai akuisisi hingga analisis. *National Institute of Standards and Technology (NIST)* merupakan salah satu kerangka kerja yang sering digunakan sebagai pedoman oleh ahli forensik dalam menangani dan menganalisis bukti digital dengan cara yang sistematis dan terstandarisasi (Kent et al., 2006). NIST menyediakan seperangkat prinsip dan prosedur yang memastikan bahwa setiap langkah dalam proses digital forensik, dari pengumpulan hingga analisis, dilakukan dengan cara yang dapat dipertanggungjawabkan dan konsisten.

Salah satu komponen utama dari kerangka kerja ini adalah proses identifikasi dan pengumpulan bukti digital yang memastikan bahwa data yang diperoleh tidak mengalami kerusakan atau perubahan. NIST menekankan pentingnya menjaga integritas data selama proses ini untuk memastikan bahwa hasil analisis dapat diterima di pengadilan. Dengan menggunakan teknik yang diakui secara internasional, seperti pembuatan salinan *bit-by-bit* dan penggunaan perangkat lunak forensik terstandarisasi, profesional dapat menghindari potensi kontaminasi atau kehilangan data. Gambar 2.1 menunjukan proses forensik pada framework NIST.



Gambar 2.1 *Framework* NIST

Selain itu, *framework* NIST juga memberikan panduan untuk analisis data yang mendalam dan terstruktur. Hal ini mencakup penggunaan alat dan teknik yang sudah diuji untuk mengekstrak informasi dari media digital, serta prosedur untuk memverifikasi keakuratan temuan. NIST berfokus pada metode yang dapat direplikasi dan diuji untuk memastikan bahwa setiap hasil analisis dapat dipertanggungjawabkan secara ilmiah.

Secara keseluruhan, penerapan kerangka kerja NIST dalam digital forensik membantu meningkatkan keandalan dan keabsahan proses investigasi. Dengan mengikuti standar dan praktik terbaik yang telah ditetapkan, profesional digital forensik dapat memastikan bahwa bukti yang dihasilkan adalah valid dan dapat digunakan untuk mendukung keputusan hukum dan administratif.

Sehingga yang dimaksud digital forensik dalam penelitian ini yaitu digital forensik merupakan bagian dari ilmu forensik yang berfungsi untuk membuktikan kejahatan komputer atau *cyber crime* secara ilmiah pada perangkat digital dengan memperhatikan tata cara penanganan data sehingga dapat dijadikan sebagai barang bukti untuk kepentingan pembuktian hukum.

2.1.2 Bukti Digital

Bukti digital merujuk pada informasi atau data yang disimpan dalam format elektronik dan dapat digunakan sebagai evidensi dalam investigasi hukum (Kizza & Migga Kizza, 2011). Jenis-jenis bukti digital meliputi *file* komputer, email, pesan teks, log aktivitas, dan metadata yang terkait dengan perangkat digital seperti komputer, smartphone, tablet, atau server. Dengan semakin berkembangnya teknologi, bukti digital menjadi elemen krusial dalam berbagai jenis kasus, mulai dari kejahatan siber hingga pelanggaran kontrak dan penipuan.

Pengelolaan bukti digital memerlukan prosedur yang ketat untuk memastikan integritas dan keaslian data. Proses ini mencakup langkah-langkah untuk mengumpulkan, menyimpan, dan menganalisis data tanpa merusak atau mengubah informasi asli (Kent et al., 2006). Dalam tahap pengumpulan, misalnya, penting untuk menggunakan teknik yang menjaga salinan *bit-by-bit* dari data yang diperoleh, memastikan bahwa bukti digital tetap utuh dan dapat dipertanggungjawabkan. Prosedur ini juga melibatkan pembuatan catatan rinci mengenai kondisi bukti saat pengumpulan dan penyimpanannya.

Selama proses analisis, data yang dikumpulkan harus diperiksa dengan hati-hati menggunakan alat dan teknik forensik yang terstandarisasi (Kent et al., 2006). Analisis ini bertujuan untuk mengekstrak informasi relevan yang dapat membantu dalam mengungkap fakta-fakta penting terkait kasus yang sedang ditangani. Teknik-teknik ini meliputi pemulihan data yang terhapus, analisis metadata, dan pemeriksaan jejak digital, yang semuanya bertujuan untuk menyajikan bukti yang dapat diverifikasi dan dipertanggungjawabkan.

Dalam konteks hukum, laporan tentang bukti digital harus disusun dengan jelas dan transparan untuk memastikan bahwa hasil analisis dapat dipahami dan diterima oleh pengadilan. Laporan ini mencakup deskripsi metodologi yang digunakan, temuan dari analisis, serta interpretasi dari data yang diperoleh. Dengan pelaporan yang tepat, bukti digital dapat memberikan kontribusi yang signifikan dalam proses hukum, membantu dalam penegakan hukum dan penyelesaian sengketa dengan cara yang berbasis pada fakta yang solid dan terverifikasi.

(Sidik Asyaky et al., 2018) berpendapat bahwa bukti digital adalah informasi yang tersimpan dalam bentuk digital yang bisa dijadikan bukti di ranah hukum. Menurut (Riskiyadi, 2020) adalah barang bukti berupa *file* dokumen, *file history* atau pun *log* yang diperoleh dari ekstraksi *file* pada barang bukti elektronik.

(Fanani et al., 2022) berpendapat bahwa bukti digital adalah hasil ekstraksi atau pemulihan berkas digital seperti dokumen, *file* media atau *file log*. Sehingga pada penelitian ini yang dimaksud dengan bukti digital adalah *file* dokumen atau pun *file history* yang telah diekstraksi atau melalui tahapan pemulihan yang dapat dijadikan sebagai bukti di ranah hukum.

2.1.3 Deteksi Jenis *File*

Dalam digital forensik, deteksi jenis *file* tidak hanya membantu dalam mengidentifikasi format *file* yang benar, tetapi juga dalam menentukan apakah *file* tersebut telah dimodifikasi atau disembunyikan. Dalam penyelidikan digital, *file* dapat sering kali disamarkan dengan mengubah ekstensi atau menempatkan data dalam format yang tidak biasa untuk menghindari deteksi. Oleh karena itu, menggunakan metode berbasis konten seperti analisis header atau signature *byte*s sangat penting untuk mengungkap informasi yang sebenarnya dan mendeteksi potensi upaya manipulasi atau penyembunyian (Karampidis & Papadourakis, 2017). Teknik-teknik ini memungkinkan penyidik untuk menilai keaslian dan integritas data dengan lebih akurat, yang merupakan kunci dalam mengumpulkan bukti yang sah dan dapat diterima di pengadilan.

Selain itu, deteksi jenis *file* juga berfungsi untuk memastikan bahwa alat forensik yang digunakan dapat membaca dan menganalisis *file* dengan benar. Berbagai jenis *file* dapat memiliki struktur dan format yang sangat berbeda, dan alat forensik yang tidak mampu mengidentifikasi jenis *file* secara tepat mungkin tidak dapat mengekstrak data dengan efektif. Oleh karena itu, pemilihan dan konfigurasi alat yang tepat, serta pembaruan berkala terhadap definisi jenis *file* yang didukung, merupakan bagian integral dari proses forensik. Hal ini memastikan bahwa penyidik dapat memanfaatkan alat mereka secara optimal untuk memproses dan menganalisis berbagai format *file* yang mungkin ditemukan selama penyelidikan.

Selanjutnya, deteksi jenis *file* juga memiliki implikasi penting dalam melindungi data pribadi dan menjaga privasi selama proses forensik. Misalnya, dalam kasus di mana *file* berisi informasi sensitif atau pribadi, identifikasi yang akurat dari jenis *file* memungkinkan penyidik untuk menangani data tersebut sesuai dengan protokol perlindungan data yang berlaku. Dengan memastikan bahwa *file* diidentifikasi dan diproses dengan benar, forensik digital dapat menghindari pengungkapan data yang tidak perlu dan melindungi hak-hak individu yang terlibat. Hal ini penting tidak hanya untuk kepatuhan terhadap peraturan privasi, tetapi juga untuk menjaga kepercayaan publik terhadap proses forensik dan lembaga yang melaksanakannya.

Deteksi jenis *file* juga berperan dalam proses pemulihan data, terutama ketika berhadapan dengan *file* yang rusak atau terhapus. Dalam situasi ini, teknik forensik dapat digunakan untuk menganalisis struktur *file* yang tersisa dan menentukan jenis *file* yang dapat dipulihkan. Misalnya, jika sebuah *file* gambar rusak, tetapi header *file* masih dapat diidentifikasi, forensik digital dapat memanfaatkan informasi tersebut untuk mencoba

memulihkan bagian dari gambar yang hilang atau rusak. Dengan teknik ini, penyidik dapat memaksimalkan kemungkinan pemulihan data yang berharga, yang mungkin penting untuk kasus yang sedang ditangani.

Dalam keseluruhan proses digital forensik, deteksi jenis *file* adalah langkah awal yang fundamental dan sangat penting. Keakuratan dalam mengidentifikasi dan memverifikasi jenis *file* secara langsung mempengaruhi kualitas hasil investigasi dan kekuatan bukti yang dikumpulkan. Tanpa deteksi yang tepat, seluruh proses forensik dapat terancam, dan bukti yang diperoleh mungkin tidak dapat dipertanggungjawabkan. Oleh karena itu, investasi dalam teknologi dan metode deteksi jenis *file* yang canggih merupakan hal yang sangat penting untuk memastikan bahwa proses forensik berjalan dengan efektif dan efisien.

Menurut (Meghanathan, 2013) dalam (Karampidis et al., 2015), ada 3 cara untuk mengidentifikasi jenis *file* adalah sebagai berikut.

a. Extension Based

Extension based merupakan metode yang paling cepat dan paling mudah untuk mendeteksi jenis *file* dan biasanya sebagai metode yang umum digunakan pada sistem operasi Microsoft. Semua jenis *file*, setidaknya di sistem berbasis Windows, umumnya disertai dengan ekstensi. Dalam sistem *file FAT*, nama *file* dibatasi delapan karakter dan ekstensi tiga karakter (*Understanding File Systems- Kingston Technology*, 2023). Meskipun sistem *file* yang lebih baru seperti *NTFS* mendukung lebih banyak karakter untuk identifikasi jenis *file*, untuk alasan kompatibilitas, pendekatan tiga huruf juga didukung. Pendekatan ini dapat diterapkan pada *file* biner dan teks.

Namun metode *extension based* memiliki kerentanan yang besar, sementara itu dapat dengan mudah untuk dikelabui dengan penggantian nama *file* sederhana (Dubettier et al., 2023). Dalam sistem Linux/Unix, ekstensi tidak diperlukan untuk mengidentifikasi jenis *file*.

b. Magic Number Based

Magic number based atau *file signature* merupakan nilai yang telah ditentukan sebelumnya dan dapat ditemukan di *header file* (Rizal et al., 2020). *Header file* adalah bagian pertama dari *file* komputer yang berisi metadata. Metadata dapat menyertakan informasi tentang konten, kualitas, dan kondisi *file* (Jenn, 2017). *Header file* juga berisi informasi yang diperlukan untuk aplikasi terkait untuk mengenali dan memahami isi *file*.

Memeriksa *magic byte* dari suatu *file* memang metode yang jauh lebih lambat dari pada hanya memeriksa ekstensinya karena *file* tersebut harus di buka (biasanya di *editor hex*) dan *magic byte* harus di baca dan dibandingkan dengan yang telah ditentukan sebelumnya.

Namun, metode mengidentifikasi jenis *file* ini juga memiliki kelemahan sebagai yaitu *magic byte* tidak digunakan di semua jenis *file*. Mereka hanya bekerja pada *file* biner dan bukan merupakan aspek yang dipaksakan atau diatur dari jenis *file*. Panjangnya bervariasi untuk jenis *file* yang berbeda dan tidak selalu memberikan jawaban yang sangat spesifik.

c. Content Based

Metode ketiga dari pendeteksian tipe *file* adalah dengan mengkalkulasi isi *file* menggunakan teknik pemodelan statistik. Ini adalah area penelitian baru dan merupakan satu-satunya cara untuk menentukan jenis *file* palsu (Amirani, Toorani, & Beheshti, 2008). (Masoumi et al., 2021) juga berpendapat bahwa *content-based* meliputi menyelidiki konten *file* dan menggunakan teknik statistik. (K. Bhat et al., 2018) melakukan identifikasi jenis *file* dengan menggabungkan pendekatan *content based* dengan *signature based* dan menjadikannya sebuah *framework*.

Identifikasi jenis *file* berbasis konten memainkan peran penting dalam pengelolaan dan pemrosesan data digital yang aman dan efisien. Berbeda dengan metode identifikasi berdasarkan ekstensi *file* yang dapat dimanipulasi atau disamarkan (Amirani, Toorani, Shirazi, et al., 2008; McDaniel & Heydari, 2003), pendekatan berbasis konten memeriksa informasi internal dalam *file* untuk menentukan format yang benar. Misalnya, dalam *file* dokumen, header seringkali berisi metadata yang mengidentifikasi format dokumen, seperti DOCX untuk Microsoft Word atau PDF untuk Portable Document Format. Teknik ini memungkinkan sistem untuk menghindari kesalahan atau potensi masalah yang timbul dari *file* dengan ekstensi yang salah atau dimanipulasi, sehingga meningkatkan keandalan sistem dalam membaca dan memproses data.

Selain itu, identifikasi berbasis konten sangat berguna dalam situasi di mana *file* mungkin telah dikompresi atau dienkripsi. Dalam kasus ini, ekstensi *file* sering kali tidak mencerminkan format asli atau jenis data di dalamnya. Dengan menganalisis *signature bytes* atau pola khusus dalam data yang terkompresi atau terenkripsi, sistem dapat menentukan format *file* yang benar dan memastikan bahwa data diproses dengan cara yang sesuai. Misalnya, sebuah *file* dengan ekstensi ZIP mungkin berisi berbagai jenis *file* di

dalamnya, dan identifikasi berbasis konten dapat membantu dalam mengekstrak dan mengelola *file-file* tersebut secara akurat.

Dalam konteks keamanan siber, identifikasi jenis *file* berbasis konten sangat penting untuk mendeteksi dan mencegah serangan berbahaya yang melibatkan *file* yang disamarkan (Ye et al., 2011). Misalnya, *file* dengan ekstensi .jpg yang sebenarnya adalah executable *file* yang berbahaya dapat menipu sistem jika hanya mengandalkan ekstensi *file*. Dengan menganalisis konten *file* dan mencocokkan signature *byte*s dengan pola yang dikenal, sistem dapat mengidentifikasi *file* berbahaya sebelum *file* tersebut diizinkan untuk diakses atau dieksekusi. Teknik ini membantu dalam mengurangi risiko dari malware dan meningkatkan keamanan data dengan memastikan bahwa *file* yang diizinkan benar-benar sesuai dengan format yang diharapkan.

Identifikasi berbasis konten juga sangat berguna dalam proses pemulihan data dan forensik digital. Ketika menghadapi *file* yang rusak atau tidak teridentifikasi dengan baik, pemeriksaan konten *file* dapat memberikan informasi penting untuk memahami format dan struktur data yang ada. Misalnya, dalam pemulihan data dari perangkat penyimpanan yang rusak, analisis konten *file* dapat membantu dalam mengidentifikasi jenis *file* yang mungkin telah terhapus atau terkorupsi. Dengan informasi ini, alat pemulihan data dapat bekerja lebih efektif dalam merekonstruksi *file* yang hilang atau rusak, sehingga memaksimalkan kemungkinan pemulihan data yang berharga.

Secara keseluruhan, identifikasi jenis *file* berbasis konten adalah teknik yang kuat dan krusial dalam berbagai aplikasi, dari manajemen data hingga keamanan siber dan forensik digital. Dengan fokus pada analisis data internal daripada hanya bergantung pada ekstensi *file*, pendekatan ini memastikan bahwa *file* dikelola dan diproses dengan akurat, meningkatkan keamanan dan integritas data dalam berbagai konteks penggunaan.

Pendekatan menggunakan *content based* juga dapat digunakan untuk melakukan klasifikasi jenis *malware* (Alkasassbeh & Al-Daleen, 2018) dan juga *ransomware* (Min et al., 2022). Sehingga pada penelitian ini *content-based* adalah proses menentukan jenis *file* palsu menggunakan teknik statistik.

2.1.4 *K-Nearest Neighbor*

K-Nearest Neighbor adalah algoritma berbasis instance yang tidak memerlukan proses pelatihan eksplisit, berbeda dengan banyak algoritma pembelajaran mesin lainnya. Sebagai gantinya, *K-Nearest Neighbor* menyimpan seluruh dataset pelatihan dan melakukan

komputasi jarak pada saat prediksi (Cunningham & Delany, 2021). Ketika sebuah data baru dihadapkan pada model *K-Nearest Neighbor*, algoritma ini menghitung jarak antara data tersebut dengan semua titik data dalam dataset pelatihan untuk menemukan k tetangga terdekat. Dengan pendekatan ini, *K-Nearest Neighbor* bisa menangani berbagai bentuk data dan tidak memerlukan asumsi khusus tentang distribusi data, membuatnya fleksibel dan mudah diadaptasi untuk berbagai masalah klasifikasi dan regresi.

Dalam hal pemilihan nilai k, penting untuk menemukan keseimbangan yang tepat (Abu Alfeilat et al., 2019). Nilai k yang terlalu kecil dapat membuat model sangat sensitif terhadap noise dan outlier, karena keputusan klasifikasi bisa dipengaruhi oleh titik data yang tidak representatif. Sebaliknya, nilai k yang terlalu besar dapat menyebabkan model terlalu umum, mengabaikan pola lokal dan menghasilkan keputusan yang kurang akurat. Oleh karena itu, pemilihan nilai k sering kali melibatkan teknik validasi silang untuk menentukan nilai yang memberikan performa terbaik pada data yang tidak terlihat, memastikan model *K-Nearest Neighbor* berfungsi secara optimal dalam situasi nyata.

Selain itu, pemilihan metrik jarak juga merupakan faktor kunci dalam implementasi *K-Nearest Neighbor*. Metrik yang umum digunakan termasuk *Euclidean distance* dan *Manhattan*, masing-masing dengan keunggulan dan kelemahan tergantung pada sifat data dan tujuan analisis (Z. Zhang, 2016). *Euclidean distance* sering digunakan untuk data yang terdistribusi dalam ruang yang homogen, sedangkan jarak Manhattan lebih baik untuk data dengan variabel diskret atau untuk data yang memiliki banyak dimensi. Memilih metrik jarak yang tepat dapat mempengaruhi hasil klasifikasi secara signifikan, sehingga sering kali memerlukan eksperimen dan penyesuaian.

K-Nearest Neighbor juga menghadapi tantangan ketika menangani data dalam dimensi tinggi, fenomena yang dikenal sebagai *"curse of dimensionality"*. Dalam ruang dimensi tinggi, jarak antara titik data menjadi kurang informatif, dan semua titik cenderung tampak sama jauh dari satu sama lain. Ini dapat mengurangi efektivitas *K-Nearest Neighbor* dan memperlambat proses komputasi. Untuk mengatasi masalah ini, teknik reduksi dimensi seperti *Principal Component Analysis* (PCA) atau teknik pemilihan fitur dapat diterapkan sebelum menerapkan *K-Nearest Neighbor*, untuk menyederhanakan data dan meningkatkan efisiensi serta akurasi model.

Walaupun *K-Nearest Neighbor* adalah metode yang intuitif dan mudah diimplementasikan, ia memerlukan pertimbangan yang hati-hati dalam hal kompleksitas komputasi dan kebutuhan memori. Saat dataset berkembang, waktu komputasi untuk mencari tetangga terdekat dapat meningkat secara signifikan, sehingga mempengaruhi

performa sistem secara keseluruhan. Untuk mengatasi hal ini, berbagai teknik optimisasi seperti struktur data pohon *KD-Tree* atau *Ball-Tree* dapat digunakan untuk mempercepat pencarian jarak dan meningkatkan efisiensi algoritma. Dengan pendekatan yang tepat, *K-Nearest Neighbor* dapat menjadi alat yang sangat berguna dalam analisis data dan pengembangan model prediktif.

(Z. Zhang, 2016) berpendapat bahwa *K-Nearest Neighbor* untuk mengklasifikasikan pengamatan yang tidak berlabel menugaskan mereka ke kelas berlabel paling mirip contoh atau *dataset*. Menurut (Xiong & Yao, 2021) adalah jenis algoritma *supervised machine learning* yang untuk menyelesaikan masalah *classification* dan *regression.*

(Chern et al., 2022) berpendapat bahwa *K-Nearest Neighbor* digunakan untuk mengidentifikasi data titik K terdekat dari database sejumlah titik data terbatas dalam ruang vektor. Sehingga pada penelitian ini yang dimaksud dengan *K-Nearest Neighbor* adalah suatu jenis algoritma *supervised machine learning* yang digunakan untuk klasifikasi data ke kelas paling mirip pada *dataset.*

2.1.5 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang sangat efektif untuk klasifikasi dan regresi, terutama dalam menangani data dengan dimensi tinggi dan kasus non-linear (Valkenborg et al., 2023). Prinsip utama *Support Vector Machine* adalah mencari hyperplane optimal yang memisahkan kelas-kelas data dengan margin terbesar. Dalam kasus data yang tidak dapat dipisahkan secara linier, *Support Vector Machine* menggunakan *kernel trick* untuk memetakan data ke ruang fitur yang lebih tinggi di mana pemisahan linier mungkin dilakukan. Fungsi kernel, seperti polinomial atau *Radial Basis Function (RBF)*, memungkinkan *Support Vector Machine* untuk menangani kompleksitas data non-linear dengan cara yang efisien (Montesinos López et al., 2022). Dengan fokus pada margin maksimum dan dukungan dari data yang berada di sekitar hyperplane, yang dikenal sebagai support vectors, *Support Vector Machine* tidak hanya memberikan model yang robust tetapi juga memiliki kemampuan generalisasi yang baik pada data yang tidak terlihat. Keunggulan *Support Vector Machine* dalam klasifikasi sering kali datang dengan biaya komputasi yang tinggi dan kebutuhan untuk memilih parameter kernel yang tepat, namun, ketika diatur dengan baik, *Support Vector Machine* dapat menghasilkan kinerja yang sangat kuat dan akurat.

Support Vector Machine bekerja dengan prinsip matematis yang mendalam, di mana tujuan utamanya adalah memaksimalkan margin antara dua kelas. Dengan memaksimalkan margin, *Support Vector Machine* bertujuan untuk membuat keputusan klasifikasi yang lebih stabil dan memiliki daya generalisasi yang lebih baik terhadap data yang tidak terlihat. Ini karena hyperplane yang dipilih oleh *Support Vector Machine* tidak hanya berusaha memisahkan kelas-kelas data tetapi juga melakukannya dengan margin yang paling lebar, mengurangi risiko overfitting dan meningkatkan kemampuan model untuk menangani data yang tidak terlatih.

Dalam penerapan praktis, *Support Vector Machine* menawarkan fleksibilitas tinggi berkat penggunaan fungsi kernel, yang memungkinkan penanganan data non-linear dengan memproyeksikan data ke ruang dimensi lebih tinggi. Fungsi kernel seperti *polinomial*, *Radial Basis Function (RBF)*, atau *sigmoid* memungkinkan *Support Vector Machine* untuk menemukan hyperplane pemisah yang kompleks yang tidak bisa ditemukan dengan metode linier sederhana . Pemilihan kernel yang tepat sangat penting dalam optimasi model *Support Vector Machine*, karena dapat mempengaruhi kinerja klasifikasi secara signifikan. Proses ini sering melibatkan eksperimen dan tuning parameter untuk menemukan kombinasi kernel dan parameter yang menghasilkan performa terbaik untuk dataset tertentu.

Namun, meskipun *Support Vector Machine* sangat kuat dalam banyak aplikasi, algoritma ini juga menghadapi tantangan terkait dengan skalabilitas pada dataset yang sangat besar. Proses pelatihan *Support Vector Machine* dapat memakan waktu dan sumber daya komputasi yang signifikan, terutama ketika jumlah data dan fitur meningkat. Untuk mengatasi masalah ini, teknik-teknik seperti implementasi algoritma aproksimasi, metode optimasi seeding, dan penggunaan kerangka kerja pembelajaran mesin yang terdistribusi dapat diterapkan untuk mempercepat pelatihan dan meningkatkan efisiensi. Dengan mengadopsi pendekatan ini, *Support Vector Machine* dapat tetap menjadi pilihan yang viable bahkan untuk dataset yang besar dan kompleks.

Selain itu, pemilihan parameter dalam *Support Vector Machine* seperti parameter regulasi C dan parameter *kernel* seperti *gamma* untuk fungsi *RBF*, dapat mempengaruhi kinerja model secara signifikan. Parameter C mengontrol *trade-off* antara memaksimalkan margin dan meminimalkan kesalahan klasifikasi, sedangkan *gamma* menentukan seberapa jauh pengaruh sebuah titik data pada keputusan klasifikasi. Pemilihan parameter yang optimal sering kali dilakukan melalui teknik *grid search* atau optimasi bayesian, yang bertujuan untuk menemukan kombinasi parameter yang memberikan kinerja terbaik pada

data yang diujikan. Tuning parameter yang tepat adalah langkah kritis dalam memastikan bahwa model *Support Vector Machine* memberikan hasil yang akurat dan robust.

Secara keseluruhan, *Support Vector Machine* adalah alat yang sangat kuat dan fleksibel dalam pembelajaran mesin, terutama dalam konteks klasifikasi data dengan dimensi tinggi dan kompleksitas non-linear. Keberhasilan *Support Vector Machine* dalam banyak aplikasi nyata, dari pengenalan wajah hingga analisis teks, menunjukkan kemampuannya untuk menangani berbagai jenis masalah klasifikasi dengan presisi tinggi. Namun, tantangan dalam pemilihan parameter dan skalabilitas harus diatasi dengan hati-hati untuk memaksimalkan manfaat dari algoritma ini. Dengan pendekatan yang tepat dan penerapan teknik optimasi yang efektif, *Support Vector Machine* dapat memberikan solusi yang andal dan efisien untuk berbagai masalah pembelajaran mesin.

2.1.6 *Byte Frequency Distribution*

Byte frequency distribution adalah konsep dalam analisis data dan kriptografi yang menggambarkan frekuensi kemunculan masing-masing *byte* dalam sebuah dataset atau *file* (McDaniel & Heydari, 2003). Dalam konteks ini, *byte* adalah unit data yang terdiri dari 8 bit, dan distribusi frekuensinya memberikan gambaran tentang seberapa sering setiap nilai *byte* tertentu muncul dalam data. Analisis distribusi frekuensi *byte* dapat membantu dalam berbagai aplikasi, termasuk pengenalan pola, deteksi anomali, dan kriptanalisis. Misalnya, dalam kriptografi, pola distribusi *byte* yang tidak terdistribusi secara merata dapat mengindikasikan penggunaan algoritma enkripsi yang lemah atau adanya pola tersembunyi dalam data terenkripsi. Sebaliknya, distribusi frekuensi yang merata biasanya menunjukkan bahwa data telah diproses atau dienkripsi dengan baik. Dengan memahami distribusi frekuensi *byte*, para peneliti dan profesional dapat mengevaluasi karakteristik data, memeriksa integritas, dan mengidentifikasi potensi masalah yang dapat mempengaruhi keamanan atau kualitas data.

Distribusi frekuensi *byte* memberikan wawasan berharga tentang pola dalam data digital, baik itu *file* teks, gambar, atau data biner lainnya. Dalam konteks *file* teks, distribusi ini sering kali mengungkapkan karakteristik karakter yang sering digunakan dan yang jarang muncul. Misalnya, dalam bahasa Inggris, huruf tertentu seperti 'e' dan 't' cenderung muncul lebih sering dibandingkan huruf lainnya. Dengan menganalisis distribusi frekuensi *byte* pada *file* teks, para peneliti dapat memahami struktur dan kebiasaan bahasa, yang juga dapat digunakan dalam algoritma kompresi data dan teknik analisis teks lainnya. Dalam kasus *file* biner atau gambar, distribusi frekuensi *byte* dapat

membantu mengidentifikasi jenis data atau format *file* yang digunakan, serta mendeteksi penyimpangan dari pola normal yang mungkin menunjukkan adanya kerusakan atau manipulasi.

Dalam kriptografi, distribusi frekuensi *byte* memainkan peran penting dalam kriptanalisis (Kim et al., 2022), yaitu proses menganalisis dan memecahkan cipher. Distribusi frekuensi yang tidak merata dalam data terenkripsi dapat memberikan petunjuk tentang kekuatan algoritma enkripsi yang digunakan . Sebagai contoh, algoritma enkripsi yang kuat biasanya menghasilkan distribusi frekuensi *byte* yang acak dan merata, sehingga sulit untuk mengidentifikasi pola atau informasi yang tersembunyi. Sebaliknya, jika distribusi *byte* terenkripsi menunjukkan pola yang dapat dikenali, ini bisa menjadi indikasi bahwa algoritma enkripsi tidak cukup kuat atau data telah terkompromi. Analisis distribusi frekuensi *byte* membantu dalam mengidentifikasi kelemahan dalam sistem enkripsi dan memastikan bahwa data terlindungi dengan baik.

Selain itu, distribusi frekuensi *byte* juga digunakan dalam forensik digital untuk mendeteksi dan menganalisis data yang mungkin telah diubah atau dipalsukan. Dengan membandingkan distribusi frekuensi *byte* dari data yang dicurigai dengan distribusi frekuensi dari data yang sah atau data referensi, penyidik dapat mengidentifikasi perubahan yang tidak sah atau kejanggalan yang mungkin menunjukkan adanya manipulasi. Misalnya, dalam kasus dokumen digital yang diduga telah diubah, analisis distribusi *byte* dapat mengungkapkan perbedaan dalam pola distribusi yang mungkin disebabkan oleh penambahan atau penghapusan data. Teknik ini sangat berguna dalam menjaga integritas data dan mengidentifikasi jejak digital dari aktivitas yang tidak sah.

Distribusi frekuensi *byte* juga relevan dalam pengembangan algoritma kompresi data. Algoritma seperti Huffman coding menggunakan frekuensi kemunculan *byte* untuk mengoptimalkan pengkodean data, dengan memberikan kode yang lebih pendek untuk *byte* yang sering muncul dan kode yang lebih panjang untuk *byte* yang jarang muncul. Dengan memahami distribusi frekuensi *byte*, algoritma kompresi dapat memanfaatkan pola dalam data untuk mengurangi ukuran *file* secara efektif tanpa kehilangan informasi. Ini tidak hanya meningkatkan efisiensi penyimpanan data tetapi juga mempercepat proses transfer data, yang sangat penting dalam aplikasi dengan keterbatasan bandwidth atau penyimpanan.

Secara keseluruhan, distribusi frekuensi *byte* adalah alat analisis yang sangat berguna dalam berbagai bidang seperti kriptografi, forensik digital, dan kompresi data. Dengan memberikan gambaran tentang pola kemunculan *byte* dalam data, analisis ini

membantu dalam memahami struktur data, mendeteksi anomali, dan meningkatkan efisiensi proses pengolahan data. Baik dalam konteks keamanan, integritas, maupun kompresi, distribusi frekuensi *byte* memainkan peran kunci dalam memastikan bahwa data dikelola dan dilindungi dengan cara yang optimal.

(Chizari et al., 2018) berpendapat bahwa *byte frequency distribution* adalah persamaan aritmatika yang berurusan dengan *file* sebagai urutan *byte*. (Jung et al., 2020) berpendapat bahwa *byte frequency distribution* adalah nilai *byte* berada di kisaran antara 0x00 dan 0xFF, dan jumlah kemunculan setiap *byte* dihitung untuk menghitung frekuensi *byte*.

Sehingga pada penelitian ini yang dimaksud dengan *byte frequency distribution* adalah persamaan aritmatika yang berurusan dengan *byte* yang berada di kisaran 0x00 dan 0xFF dan dihitung jumlah kemunculannya.

2.1.7 *File* Corrupt

File corrupt, atau *file* yang rusak, merujuk pada *file* digital yang tidak dapat dibaca atau diakses dengan benar karena kerusakan dalam struktur atau isinya. Kerusakan ini bisa disebabkan oleh berbagai faktor, seperti gangguan saat proses penyimpanan, masalah perangkat keras, kesalahan perangkat lunak, atau infeksi virus. Ketika sebuah *file* mengalami korupsi, data di dalamnya bisa menjadi tidak utuh atau tidak dapat dipulihkan, yang dapat mengakibatkan kehilangan informasi penting atau ketidakmampuan untuk membuka *file* tersebut. Dalam beberapa kasus, *file* yang korup mungkin masih bisa dibaca sebagian, namun sebagian besar data mungkin hilang atau terdistorsi. Untuk mengatasi masalah ini, seringkali diperlukan alat pemulihan data atau teknik perbaikan *file* yang dirancang khusus untuk memperbaiki struktur *file* yang rusak. Pemulihan *file* yang berhasil dapat mengurangi dampak kerusakan dan memastikan bahwa data berharga dapat dipulihkan dan digunakan kembali. Namun, mencegah kerusakan *file* sejak awal melalui backup reguler dan pemeliharaan sistem yang baik adalah langkah terbaik untuk menghindari masalah yang lebih serius.

File corrupt dapat muncul dalam berbagai format, mulai dari dokumen teks, gambar, hingga video dan database. Setiap jenis *file* memiliki struktur dan format yang spesifik, dan kerusakan dapat terjadi pada berbagai tingkat, seperti header *file*, metadata, atau konten utama. Misalnya, dalam *file* gambar, korupsi mungkin menyebabkan gambar yang tidak dapat ditampilkan dengan benar atau menampilkan artefak visual yang tidak diinginkan. Sementara itu, dalam *file* database, kerusakan bisa mengakibatkan kehilangan

data, inkonsistensi, atau kegagalan sistem yang lebih luas. Identifikasi dan pemulihan *file* corrupt sering memerlukan pemahaman mendalam tentang format *file* tersebut dan penggunaan alat khusus yang dapat menangani berbagai jenis kerusakan.

Salah satu penyebab umum *file* corrupt adalah gangguan saat proses penulisan atau penyimpanan data. Ini bisa terjadi akibat pemadaman listrik mendadak, kegagalan perangkat keras seperti hard drive, atau kesalahan dalam perangkat lunak yang digunakan untuk menyimpan atau memproses *file*. Dalam kasus-kasus ini, *file* dapat menjadi rusak karena data yang belum sepenuhnya ditulis atau karena struktur *file* yang tidak lengkap. Untuk mengurangi risiko kerusakan semacam ini, penting untuk menggunakan perangkat keras yang handal, memastikan sistem memiliki sumber daya daya yang stabil, dan melakukan pemeriksaan rutin pada integritas *file* dan sistem penyimpanan.

Dalam situasi di mana *file* corrupt terjadi, teknik pemulihan data menjadi kunci untuk mencoba mengembalikan *file* ke kondisi yang dapat digunakan. Alat pemulihan data sering kali mencakup perangkat lunak yang dapat memperbaiki atau merekonstruksi struktur *file* yang rusak dengan cara membaca data yang masih utuh dan mengabaikan bagian yang korup. Dalam beberapa kasus, pemulihan bisa mencakup teknik manual seperti mengedit *file* dengan editor hex untuk memperbaiki header atau metadata yang rusak. Namun, keberhasilan pemulihan sering kali bergantung pada tingkat kerusakan dan ketersediaan data yang tidak terpengaruh. Oleh karena itu, penyimpanan cadangan secara teratur tetap menjadi strategi terbaik untuk melindungi data penting dari kerusakan.

Pencegahan *file* corrupt juga melibatkan penggunaan praktik terbaik dalam pengelolaan data dan sistem. Melakukan backup data secara teratur adalah langkah proaktif yang sangat penting untuk memastikan bahwa salinan terbaru dari *file* tersedia jika terjadi kerusakan. Selain itu, menjaga perangkat lunak dan sistem operasi selalu diperbarui dapat membantu mengurangi risiko kerusakan akibat bug atau kerentanan keamanan. Penggunaan sistem *file* yang mendukung integritas data, seperti sistem *file* dengan fitur journaling atau checksum, juga dapat membantu mendeteksi dan memperbaiki kesalahan sebelum *file* menjadi terlalu rusak. Dengan menerapkan langkah-langkah pencegahan ini, pengguna dapat meminimalkan risiko *file* corrupt dan memastikan data tetap aman dan dapat diakses.

Secara keseluruhan, *file* corrupt adalah masalah yang dapat mempengaruhi berbagai aspek pengelolaan data, dari *file* pribadi hingga sistem bisnis yang kritis. Memahami penyebab dan metode pemulihan untuk *file* yang rusak dapat membantu dalam mitigasi dampak kerusakan dan memastikan keberlangsungan operasional. Walaupun teknik

pemulihan dapat memperbaiki banyak kasus *file* corrupt, pencegahan melalui praktik manajemen data yang baik dan sistem yang andal tetap menjadi kunci untuk melindungi data dari kerusakan di masa depan.

(Abdillah, 2022) berpendapat bahwa *file corrupt* merupakan *file* yang tidak bisa di akses karena mengalami kerusakan. Menurut (Bairavasundaram et al., 2008; Hafner et al., 2008; Krioukov et al., 2008) dalam penelitian (Charyyev & Arslan, 2020) kerusakan data dapat terjadi pada penyimpanan sistem saat melakukan operasi membaca dan menulis.

Sehingga pada penelitian ini yang dimaksud *file corrupt* adalah *file* yang tidak bisa diakses karena mengalami kerusakan yang dapat terjadi saat melakukan operasi membaca dan menulis.

2.2 Tinjauan Pustaka

Tinjauan pustaka yang digunakan pada penelitian ini dirangkum pada Tabel 2.1.

Tabel 2.1 Tinjauan Pustaka

No	Keywords	Metode	Aspek Kritis	Pustaka
1.	*Events detection; file fragment classification; neural networks*	*Convolutional Neural Network*, *Xgboost*, *Support Vector Machine*, *K-Nearest Neighbor*	Pendekatan yang digunakan adalah membuat 3 jenis eksperimen yaitu mengkalkulasi awal *file*, tengah *file*, dan akhir *file*. Setiap eksperiman diambil ukuran sample sebanyak 64, 128, 256, 512, 1000, 2000, dan 4000 *bytes*. *Convolutional Neural Network*, *Xgboost*, *Support Vector machine*, dan *K-Nearest Neighbor* digunakan untuk melakukan identifikasi dan mengambil 20 jenis *file* dari *dataset GovDocs1* untuk melakukan identifikasi. Penelitian ini menunjukkan ada perbedaan persentasi pada *file* yang diambil dari awal, tengah, maupun akhir.	(Y. Wang et al., 2018)
2.	-	*Convolutional*	Menggunakan 13 jenis *file* dari	(A. Bhat et al.,

		Neural Network, Feed-forward Neural Network	*dataset GovDocs1* dengan masing-masing jenis berjumlah 1000 *file*. Menggabungkan *Convolutional Neural Network* dan *Feed-forward Neural Network* sebagai algoritma yang digunakan.	2021)
3.	*Digital Forensic, File Classification, Deep Learning, Hexadecimal Value*	*Convolutional Neural Network*	Menggunakan algoritma *Convolutional Neural Network* dan 8 jenis data set yaitu *avi, mp4, bmp, jpg, png, mp3, wav,* dan *pdf.*	(Neaimi et al., 2020)
4.	*Smartphone*, Multimedia, *File Signature*, Android, Metadata, HxD	HxD	Penelitian ini menggunakan tools yaitu *HxD* untuk melakukan identifikasi.	(Sourabh et al., 2021)
5.	*File-type classification, memory forensics, carv-ing, machine learning, Convolutional Neural Network*	*Convolutional Neural Network*	Menggunakan *Convolutional Neural Network* untuk melakukan kalkulasi dan menggunakan 75 tipe data yang populer.	(Mittal et al., 2021)
6.	*File type classification, Artificial Intelligence*	*Multilayer Perceptron*	Menggunakan *Multilayer Perceptron* untuk melakukan kalkulasi dan menggunakan *dataset Government Digital Service* dan *The National Archives*.	(Venkata & Green, 2019)
7.	*Multimedia forensic*	*File Signature*	Menganalisis *file signature* multimedia pada beberapa	(Hamdi et al., 2017)

	analysis, smartphone forensics, file signature analysis, multimedia file source authentication		perangkat *mobile*. Menggunakan dua perspektif yaitu menganalisis *file signature* dari *file* multimedia untuk mengetahui dari mana *file* tersebut berasal. Perpektif kedua yaitu menentukan apakah *file* tersebut masih original atau sudah dilakukan modifikasi.	
8.	-	*Byte2Vec*, *K-Nearest Neighbor*	Menggunakan fitur *Byte2Vec* dan algoritma *K-Nearest Neighbor* untuk melakukan kalkulasi.	(Haque & Tozal, 2022)
9.	-	*Feed-forward Neural Network*, *Recurrent Neural Network*, *Convolutional Neural Network*	Menggunakan algoritma *Feed-forward Neural Network*, *Recurrent Neural Network*, dan *Convolutional Neural Network* untuk melakukan kalkulasi.	(Hiester, 2018)
10.	*File fragment classification, file carving, automated feature extraction, sparse coding, dictionary learning, unsupervised learning, n-gram, support vector machine*	*Support Vector machine, Sparse Coding*	Menggunakan *sparse coding* untuk melakukan ekstraksi fitur dan menggunakan algoritma *Support Vector Machine* untuk melakukan kalkulasi.	(F. Wang et al., 2018)
11.	*Digital*		Menggunakan metode secara live	(Hegarty &

	forensics; file signatures; live investigations		dengan *USB* untuk mengidentifikasi jenis *file*.	Haggerty, 2015)
12.	*Fragment Classification; Data Type; File Type; Support Vector machine*	*Support Vector machine*	Menggunakan algoritma *Support Vector Machine* untuk melakukan kalkulasi dan menggunakan *byte frequency distribution* dan *shannon entropy* sebagai fitur.	(Zheng et al., 2015)
13.	*file fragment classification; file type detection; artificial neural network; Convolutional Neural Network; feed forward neural network; file type forensics*	*Convolutional Neural Network*, *Feed-forward Neural Network*	Menggunakan *Convolutional Neural Network* dan *Feed-forward Neural Network* untuk melakukan kalkulasi. 18 jenis data digunakan dalam penelitian ini dan juga menggunakan *GovDocs1* sebagai *dataset*.	(Vulinovic et al., 2019)
14.	*File fragment identification, image data type identification, PNG, Support Vector machine, decision tree algorithm, Shannon entropy*	*Support Vector machine, Decision Tree*	Menggunakan *Support Vector Machine* untuk melakukan kalkulasi dan menggunakan *GovDocs1* sebagai *dataset*.	(Nguyen et al., 2017)
15.	*Classification Algorithm,*	*Support Vector*	Membandingkan *Multilayer Perceptron*, *Support Vector*	(Masoumi et al., 2021)

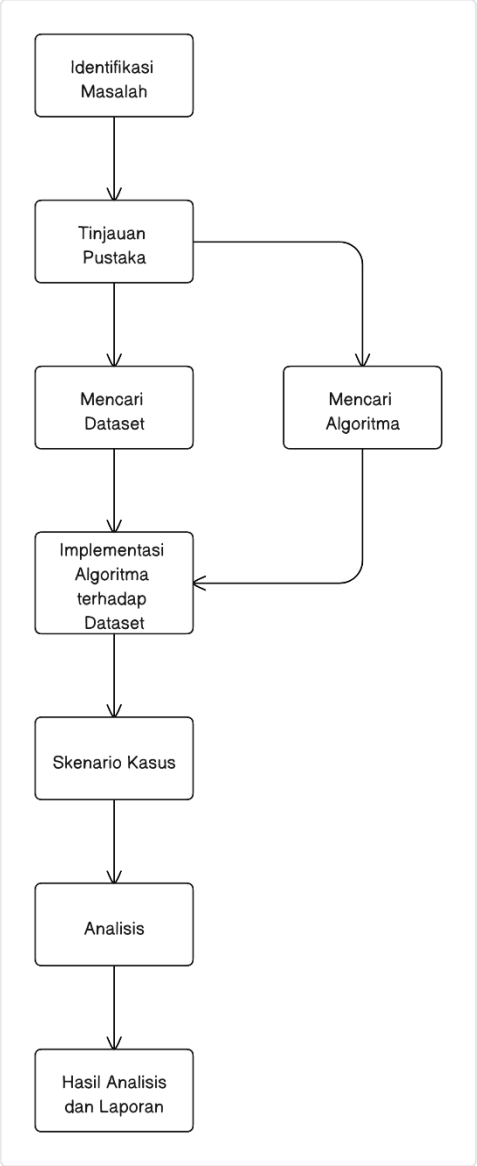
	Feature Reduction, File Fragments, File Fragment Recognition, SFS, SFFS	*machine*, *K-Nearest Neighbor*	*machine*, *K-Nearest Neighbor* untuk melakukan identifikasi *file* dan menggunakan 6 jenis *file* yaitu *pdf, txt, jpg, doc, html,* dan *exe* yang diambil dari *dataset GovDocs1*.	
16.	*Artefak Digital; file Corrupt; Forensik Digi-tal; file Type Identification; K-Nearest Neigh-bor*	*K-Nearest Neighbor, Byte Frequency Distribution*	Melakukan identifikasi jenis *file* berbasis *machine learning* dengan algoritma K-Neareset Neighbor pada *dataset NapierOne* dengan ekstraksi fitur *Byte Frequency Distribution*.	(Fawzan et al., 2024)

Berdasarkan tinjauan pustaka di atas, maka penelitian akan menggunakan algoritma *K-Nearest Neighbor* seperti penelitian (Haque & Tozal, 2022; Masoumi et al., 2021; Y. Wang et al., 2018) dan Support Vector Machine seperti penelitian (Masoumi et al., 2021; F. Wang et al., 2018; Zheng et al., 2015) untuk melakukan kalkulasi dan klasifikasi data serta akan mengubah *dataset* menggunakan *NapierOne* dan menambah jenis *file* untuk melengkapi penelitian (A. Bhat et al., 2021; Masoumi et al., 2021; Neaimi et al., 2020; Vulinovic et al., 2019; Y. Wang et al., 2018) nya menjadi 23 dan menggunakan 5000 *file*.

BAB 3
Metodologi

3.1 Metodologi Penelitian

Tahapan yang dilakukan pada penelitian ini tediri dari 8 langkah yaitu identifikasi masalah, tinjauan pustaka, mencari *dataset*, mencari algoritma, implementasi algoritma terhadap *dataset*, skenario kasus, analisis, hasil analisis dan laporan. Alur tahapan metodologi dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metodologi penelitian

3.1.1 Identifikasi Masalah

Tahap ini merupakan awal dari penelitian yaitu melakukan identifikasi masalah yang ada sehingga dapat menemukan topik untuk dilakukan penelitian. Permasalahan yang penulis temukan yaitu terdapat banyaknya kasus terkait kejahatan siber seperti 3723 orang melakukan manipulasi data otentik, 2131 orang melakukan penipuan secara daring, 1089 orang melakukan tindakan kejahan siber, 835 orang melakukan pencemaran nama baik secara daring, 358 orang mengakses sistem secara tidak sah, 64 orang melakukan judi secara daring, 145 orang melakukan tindakan pengancaman secara daring, 143 orang melakukan pornografi dan prostitusi secara daring, 59 orang melakukan tindakan penghinaan secara daring, dan 43 orang melakukan ujaran kebencian secara daring.

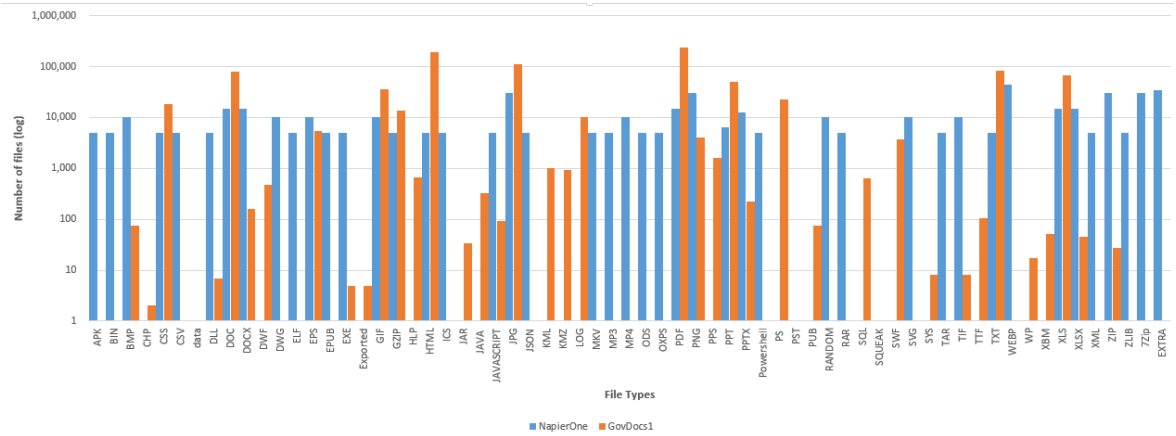
Kejahatan terbanyak yaitu adanya manipulasi data otentik yang salah satu caranya adalah merusak barang bukti digital sehingga akan menyulitkan ahli forensik untuk melakukan analisis terhadap artefak digital tersebut. Berdasarkan permasalahan yang telah terjadi maka penulis menemukan permasalahan yang akan diteliti yaitu bagaimana mengidentifikasi jenis *file* yang tidak terdefinisi pada artefak digital menggunakan *machine learning*.

3.1.2 Tinjauan Pustaka

Tahap ini penulis mengumpulkan informasi berkaitan dengan permasalahan yang akan diteliti oleh penulis. Pustaka yang dicari dan dikumpulkan bersumber dari jurnal, paper, berita, maupun artikel yang mendukung serta membahas tentang digital forensik, bukti digital, identifikasi jenis *file*, dan *machine learning* untuk mendapatkan wawasan dan hal yang belum dilakukan pada penelitian terdahulu sehingga dapat digunakan sebagai acuan pada penelitian yang akan penulis teliti.

3.1.3 Mencari *Dataset*

Tahap ini dilakukan untuk mencari *dataset* terbaru dan digunakan sebagai data acuan untuk mendukung penelitian ini. *Dataset* yang akan digunakan pada penelitian ini yaitu *NapierOne* serta menggunakan 26 jenis *file* dan menggunakan 5000 *file*.



Gambar 3.2 *GovDocs1* Comparison

Gambar 3.2 menunjukan perbandingan data antara *dataset GovDocs1* dengan *NapierOne*. *Dataset NapierOne* merupakan alternatif *dataset* dari *GovDocs1* yang memiliki 44 jenis *file* dan memiliki 5000 *file* di setiap jenisnya.

3.1.4 Mencari Algoritma

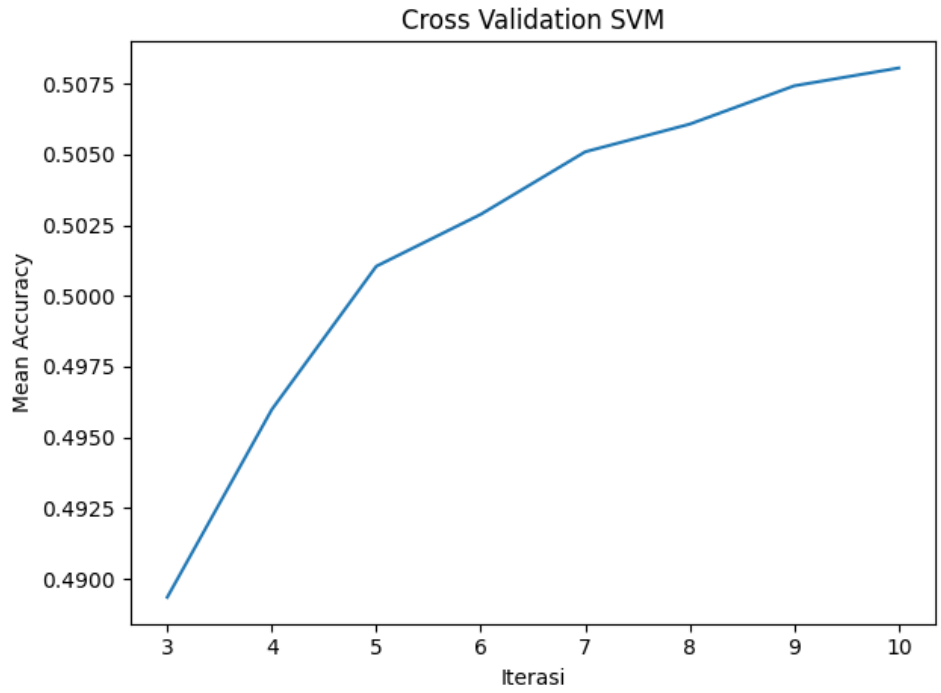
Tahap ini dilakukan untuk mencari algoritma yang cocok untuk melakukan klasifikasi terhadap *file* yang tidak terdefinisi. Berdasarkan hasil dari tahap tinjauan pustaka, maka algoritma *K-Nearest Neighbor* dan *Support Vector Machine* digunakan untuk melakukan klasifikasi.

Metode mencari algoritma yang cocok untuk melakukan klasifikasi terhadap *file* yang tidak terdefinisi yaitu dapat dilakukan menggunakan *machine learning*. *machine learning* merupakan salah satu teknik yang dapat membantu dalam melakukan klasifikasi. Menurut (Sah, 2020) ada 3 teknik *machine learning* yaitu *supervised learning, unsupervised learning*, dan *reinforcement learning*. Kegiatan untuk melakukan klasifikasi termasuk dalam *machine learning* jenis *supervised learning*. Klasifikasi dalam *machine learning* adalah suatu pengelompokan data yang di dalamnya data tersebut mempunyai kelas label atau target. Sehingga algoritma-algoritma untuk menyelesaikan masalah klasifikasi dikategorisasikan ke dalam *supervised learning* atau pembelajaran yang diawasi. Maksud dari pembelajaran yang diawasi adalah data label atau target ikut berperan sebagai *'supervisor'* atau 'guru' yang mengawasi proses pembelajaran dalam mencapai tingkat akurasi atau presisi tertentu.

(Sekeroglu et al., 2020) melakukan komparasi algoritma *K-Nearest Neighbor* dan *Support Vector Machine* untuk melakukan klasifikasi data dan mendapatkan bahwa *K-Nearest Neighbor* memiliki performa yang paling tinggi dengan akurasi 99.96%, presisi

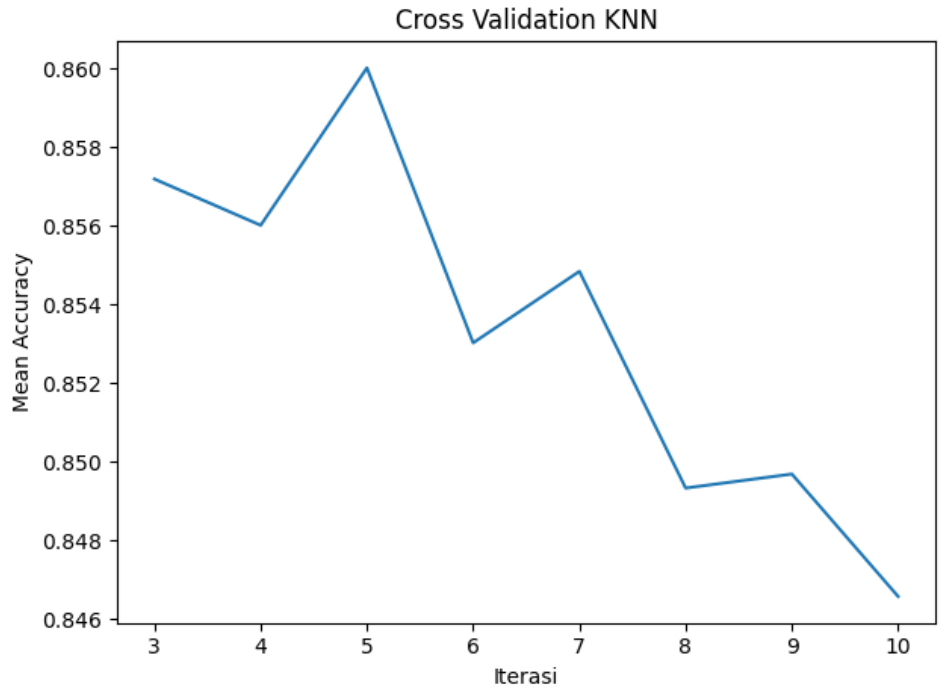
99%, dan *recall* 96.20% sedangkan *Support Vector Machine* memiliki akurasi 97.91%, presisi 93.20% dan *recall* 86.40%. (Masoumi et al., 2021) melakukan pendekatan klasifikasi jenis *file* dengan menggunakan algoritma *Multilayer Perception*, *K-Nearest Neighbor*, dan *Support Vector Machine* dengan *byte frequency distribution* sebagai ekstraksi fitur. Penelitian ini hanya memanfaatkan 1000 *byte* dan 500 *byte* fragmen *file* untuk kalkulasi jenis *file* dan menggunakan algortima *feature reduction* yaitu *Sequential Forward Selection (SFS)* dan algoritma *Sequential Floating Forward Selection (SFFS).*

Cross validation digunakan untuk mengukur performa pada model yang akan dibuat. Pada penelitian ini, dilakukan iterasi dari 3-10 untuk mengukur nilai K atau C yang paling optimal.



Gambar 3.3 Grafik Cross Validation SVM

Berdasarkan Gambar 3, maka C yang digunakan pada penelitian ini adalah 10.

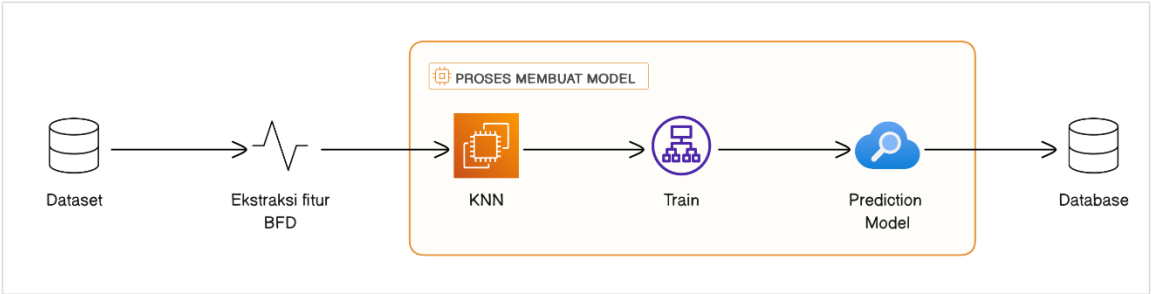


Gambar 3.4 Grafik Cross Validation KNN

Berdasarkan Gambar 4, maka K yang digunakan pada penelitian ini adalah 5.

3.1.5 Implementasi Algoritma

Tahap ini merupakan gabungan dari tahap mencari *dataset* dan tahap mencari algoritma yaitu implementasi algoritma *K-Nearest Neighbor* terhadap *dataset NapierOne* agar menghasilkan model yang dapat digunakan untuk klasifikasi jenis *file* yang tidak terdefinisi.



Gambar 3.5 Alur implementasi *K-Nearest Neighbor* terhadap *dataset*

Gambar 3.5 menjelaskan tentang alur kerja pembuatan model yang akan digunakan untuk mengklasifikasikan jenis *file*. *Dataset* yang diperoleh dari *NapierOne* dilakukan ekstraksi fitur dengan menggunakan *byte frequency distribution* kemudian hasil ekstraksi tersebut dilatih menggunakan *K-Nearest Neighbor* sehingga dapat membuat model prediksi kemudian disimpan di *database* untuk dilakukan proses selanjutnya.

3.1.6 Langkah-langkah Penelitian

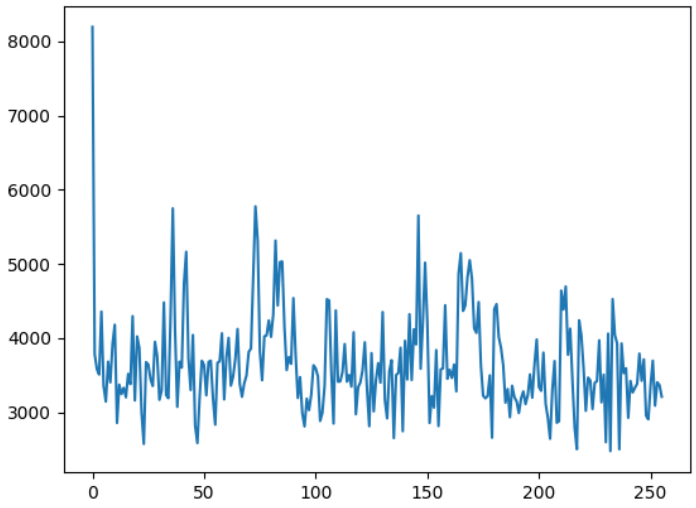
Berikut langkah-langkah yang dilakukan untuk membuat model *K-Nearest Neighbor*:

1. *Dataset NapierOne* dipisahkan berdasarkan jenis *file* sehingga saat melakukan ekstraksi fitur jenis *file* tersebut dapat di klasifikasikan.
2. Setelah dipisahkan berdasarkan jenis *file*, kemudian dilakukan proses ekstraksi fitur menggunakan teknik *byte frequency distribution* sehingga didapatkan hasil berupa 255 kolom yang berisi nilai *byte* unik dan pada kolom terakhir ditambahkan label berupa jenis *file* tersebut agar dapat dilakukan klasifikasi.

Tabel 3.1 Contoh hasil ekstraksi fitur dengan *Byte Frequency Distribution*

0	1	2	3	...	255	type
14225	2179	1728	2065		2418	apk
959	4652	3893	4138		3065	mp3

Tabel 3.1 menunjukkan contoh hasil ekstraksi fitur menggunakan *byte frequency distribution*. Hasilnya, dari sebuah *file* didapatkan total nilai unik dari setiap *byte* sehingga dapat dilakukan kalkulasi secara statistik untuk mengidentifikasi jenis *file*.



Gambar 3.6 Sebaran data pada *file* jpg

Gambar 3.6 merupakan contoh sebaran data pada *file* multimedia yaitu jpg. Sebaran data tersebut diambil dari hasil kalkulasi ekstraksi fitur menggunakan *byte frequency distribution*.

3. Setelah fitur berhasil di ekstraksi, kemudian digunakan algoritma *K-Nearest Neighbor* untuk melakukan klasifikasi berdasarkan hasil ekstraksi fitur *byte frequency*

distribution sehingga menjadi sebuah model yang dapat digunakan untuk melakukan klasifikasi pada *file* tidak terdefinisi.

4. Model yang telah terbuat kemudian disimpan didalam basis data sehingga dapat diimplementasikan menjadi sebuah aplikasi untuk memudahkan ahli forensik dalam mengidentifikasi jenis *file*.

BAB 4
Hasil dan Pembahasan

4.1 Analisis Kebutuhan

Proses penelitian ini menggunakan komponen yang terdiri dari perangkat lunak dan perangkat keras sehingga dapat dilakukan uji coba identifikasi jenis *file* menggunakan algoritma *K-Nearest Neighbor*. Tabel 4.1 menunjukkan komponen perangkat lunak dan perangkat keras yang digunakan pada penelitian ini.

Tabel 4.1 Spesifikasi hardware dan software

Hardware	
Prosesor	Intel Core i7 4750HQ (2,0 GHz/8 CPU)
RAM	12 GB
VGA	GTX 950M
Storage	Harddisk 1TB
Software	
Windows	10 version 22H2
Python	3.11.1
Scikit Learn	1.3.1
Numpy	1.26.1
Pickle	4.0
Matplotlib	3.8.0
PHP	8.2
Postgres	14.0

4.2 Skenario Kasus

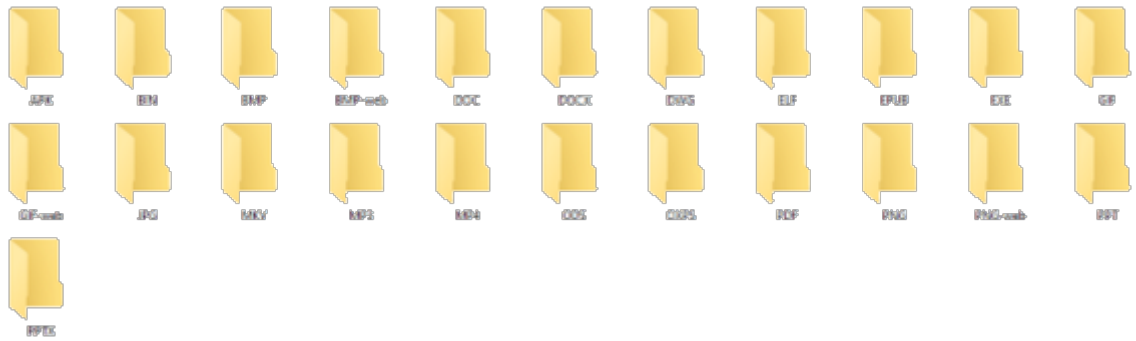
File uji merupakan *file* yang utuh sehingga dapat dilakukan komparasi terhadap hasil yang dikeluarkan oleh model *K-Nearest Neighbor*. Langkah awal sebelum melakukan pengujian terhadap model *K-Nearest Neighbor* yang telah dibuat, kita harus mempersiapkan sebuah *file* yang akan diuji terlebih dahulu.

File uji didapatkan dari situs https://*file*-examples.com/ yang diunduh pada 21 Januari 2023 yang berjumlah 44 *file* dengan berbagai ukuran dan jenis *file*. *File* tersebut kemudian dilakukan pemrosesan menggunakan model *K-Nearest Neighbor* dan *Support*

Vector Machine untuk mengidentifikasi jenis *file* dengan keadaan masih utuh. *File* tersebut juga diuji dengan menggunakan 3 cara yaitu menghapus header *file* sebanyak 10%, menghapus footer *file* sebanyak 10%, dan menghapus header dan footer *file* sebanyak masing-masing 10%.

4.3 Persiapan Dataset

Dataset NapierOne yang telah diperoleh perlu diekstraksi terlebih dahulu ke dalam folder berdasarkan jenis *file*nya. Pada setiap folder, ada 1 *file* tambahan yang berekstensi .pdf yang berisi tentang deskripsi dataset dan harus dihapus sehingga setiap folder memiliki jenis *file* yang sama. Gambar 4.1 merupakan folder-folder dengan jenis *file* yang sudah dikelompokan.



Gambar 4.1 Pengelompokan Folder Berdasarkan Jenis *File*

4.4 Ekstraksi Fitur

Setelah dataset dikelompokan berdasarkan jenis *file*, langkah selanjutnya adalah ekstraksi fitur menggunakan *byte frequency distribution*. Ekstraksi fitur ini dilakukan agar dataset memiliki panjang yang sama yaitu 256 namun memiliki frekuensi *byte* yang berbeda-beda sesuai dengan *file* tersebut sehingga dapat dilakukan kalkulasi menggunakan algoritma *K-Nearest Neighbor* dan *Support Vector Machine*.

Setiap baris data akan dilakukan penyisipan ekstensi jenis *file* sebagai *type* untuk membedakan antar jenis *file*. Gambar 4.2 merupakan contoh dataset yang telah dilakukan ekstraksi fitur menggunakan *byte frequency distribution*.

	0	1	2	3	4	5	6	7	8	9	...	247	248	249	250	251	252	253	254	255	type
0	14225	2179	1728	2065	1775	1736	1857	2242	1768	1702	...	2340	2196	2315	2192	2286	2289	2348	2377	2418	apk
1	61688	10818	8136	9698	7890	7386	7613	8175	11813	7609	...	9902	9007	9454	8960	9535	9488	9671	9646	11814	apk
2	81275	10977	7757	10518	7438	6978	7116	7718	13003	6799	...	9268	8165	8389	8218	9029	8663	8854	9046	17087	apk
3	67897	7480	4247	6132	3887	3429	3842	3749	9132	3529	...	4682	3854	4121	3975	4301	4156	4263	4379	10800	apk
4	66296	7675	4483	6610	4380	3839	4261	4232	9394	3880	...	5097	4487	4727	4489	4910	4916	4765	4972	10336	apk

[5 rows x 257 columns]

Gambar 4.2 Contoh Hasil Ekstraksi Fitur

Dapat dilihat pada Gambar 4.2, terdapat nilai *byte* unik dari 0 sampai 255 dan juga memiliki *type* sesuai dengan jenis *file*nya.

4.5 Mencari Nilai K dan C Optimum

Cross-validation adalah teknik evaluasi model dalam pembelajaran mesin yang digunakan untuk mengukur performa model secara lebih akurat dan mengurangi risiko overfitting. Proses ini melibatkan pembagian dataset menjadi beberapa subset atau lipatan, di mana model dilatih pada sebagian data dan diuji pada sisa data yang belum digunakan untuk pelatihan.

Salah satu metode cross-validation yang paling umum adalah k-fold cross-validation, di mana dataset dibagi menjadi k lipatan yang berbeda. Model dilatih k kali, masing-masing kali menggunakan k-1 lipatan untuk pelatihan dan satu lipatan yang berbeda untuk pengujian. Hasil dari setiap iterasi diuji dan kemudian digabungkan untuk memberikan estimasi kinerja model yang lebih stabil dan representatif. Teknik ini membantu dalam memastikan bahwa model tidak hanya berfungsi dengan baik pada subset data tertentu, tetapi juga dapat menggeneralisasi performanya pada data yang belum terlihat, memberikan gambaran yang lebih realistis tentang bagaimana model akan berperforma pada data baru di dunia nyata. Pada penelitian ini dilakukan iterasi dari nilai 3 sampai 10 untuk melihat berapa nilai yang paling optimum unutk dijadikan sebagai parameter pada *machine learning*.

4.6 Membuat Model *Machine Learning*

Pada penelitian ini digunakan hardware dan software untuk membantu membuat model *machine learning* seperti pada Tabel 4.1. Berikut merupakan langkah-langkah yang dilakukan untuk membuat model *machine learning* pada penelitian ini.

1. Dataset yang telah dilakukan ekstraksi fitur akan di *load* oleh pandas sehingga menghasilkan *DataFrame*.

```
csv_path = r"C:\Users\Ihsan Fawzan\Desktop\thesis\dataset.csv"
dataset = pd.read_csv(csv_path)
```

2. *DataFrame* tersebut kemudian dibagi menjadi 2 dengan variabel X dan y dimana X akan menampung nilai dari ekstraksi fitur sedangkan y akan menampung nilai jenis *file*.

```
X = dataset.drop('type', axis=1)
y = dataset['type']
```

3. Setelah itu, data X dan y akan dibagi sebanyak 30% *test size* dan 70% *train size* secara acak.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

4. Sebelum dilakukan kalkulasi model *machine learning,* dilakukan terlebih dahulu inisiasi *classifier*.

```
# Classifier pada model KNN
classifier = KNeighborsClassifier(n_neighbors=5)

# Classifier pada model SVM
classifier = svm.LinearSVC(class_weight='balanced', C=10)
```

5. Proses *fitting* digunakan untuk melakukan kalkulasi *machine learning* pada algoritma *K-Nearest Neighbor* dan *Support Vector Machine*.

```
classifier.fit(X_train.values, y_train)
```

6. Setelah proses *fitting* selesai, model selanjutnya disimpan dengan pickle sehingga dapat digunakan kembali.

```
pickle.dump(classifier, open('model.sav', 'wb'))
```

7. Proses membuat model menggunakan algoritma *K-Nearest Neighbor* dan *Support Vector Machine* menggunakan ekstraksi fitur *byte frequency distribution* sudah selesai.

4.7 Implementasi Model *Machine Learning*

Setelah model *machine learning* selesai dibuat, langkah selanjutnya adalah menguji apakah model yang sudah dibuat dapat digunakan untuk melakukan identifikasi jenis *file* berdasarkan skenario kasus pada penelitian ini.

Metadata dari *file* asli digunakan sebagai pembanding sehingga dapat terdeteksi jika ada manipulasi pada *file* tersebut. Digunakan 3 jenis *hash* yaitu *md5, sha1,* dan *sha256* dalam penelitian ini untuk mengambil *fingerprint file*. Tabel 4.2 merupakan contoh hasil hash yang telah dilakukan pada *file* asli sedangkan untuk lebih lengkapnya ada pada lampiran.

Tabel 4.2 Metadata *File* Asli

Nama *File*	Ekstensi	MD5	SHA1	SHA256
Logo-UII-Asli.png	png	ad495618f18fc70815d5bf6ba34898ff	65b26872eb36304a13759770132750cfb0dc0bea	6ac7eafc27de392831071fe05e112dea2fd681bf182f2bdb0dc5776860efca8f
file-example_PDF_1MB.pdf	Pdf	ee7ac8084eeab08035fdcb47bfa81931	cec8ad914b1e9db83626b98e8d98512616975fdf	5e4d40fcd8b22453a5da2d32533b128f2565f3fc7a4d1647a93c86cdbb4be37a
file-example_PDF_500_kB.pdf	pdf	c7d354ff10c868752f1b032e5b0e55e8	6d730748d689f1c6c71bc5c5f7a5b1a65c533242	e83014e71fc8e772b7689a3f1c8628a2ef2852a1a38e31bddaf823615570e709

Hasil kalkulasi *file* asli dengan menggunakan algoritma *K-Nearest Neighbor* dan *Support Vector Machine* untuk mengidentifikasi jenis *file* dapat dilihat Tabel 4.3 sedangkan untuk lebih lengkapnya dapat dilihat pada lampiran.

Tabel 4.3 Hasil Kalkulasi *File* Asli

Nama *File*	Metode	Hasil
Logo-UII-Asli.png	*K-Nearest Neighbor*	pngw
Logo-UII-Asli.png	*Support Vector Machine*	pngw
file-example_PDF_1MB.pdf	*K-Nearest Neighbor*	gifw
file-example_PDF_1MB.pdf	*Support Vector Machine*	pdf
file-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	ods
file-example_PDF_500_kB.pdf	*Support Vector Machine*	bmp

Dapat dilihat bahwa ada beberapa *file* yang menghasilkan keluaran yang tidak sesuai dan juga tidak dapat di proses karena kekurangan fitur sehingga tidak dapat diketahui jenis *file*nya.

Metadata dari *file* setelah dilakukan modifikasi pengubahan *file* dapat dilihat pada Tabel 4.4 dan lebih lengkapnya pada lampiran. Digunakan 3 jenis manipulasi yaitu menghapus 10 persen dari header *file*, menghapus 10 persen dari footer *file*, dan menghapus 10 persen dari header *file* dan footer *file*.

Tabel 4.4 Metadata *File* Manipulasi

Nama *File*	Ekstensi	MD5	SHA1	SHA256
CUT-akhir-10%-Logo-UII-Asli.png	png	31775bcd0fa0a6abb0c7b732f84c078c	53a0866a45c14ee94b564eeb790041a30f63a0b2	18216ba795c1ea978914381a1450872b328c0fb7232d9738f723bebd305a35a9
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	pdf	0c4b89cfb5b891b4275144ac09728c77	ae1599788d0190b2f9af124b03f32c1a1ae9954f	dd921de91700f7c2440be3abd7f991f5c5b29ca6568383fb5a0dd4650d709c3f
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	pdf	494f85160d32cffda47a1d3d5c0866fa	5475a31007d80dfd2b329f63bdc4238b56f0d468	d742eb5b3313f821280be9656285f6c3bb4b79ad421655a46630dcb8f641602c
CUT-akhir-10%-*file*-sample_100kB.doc	doc	1ccb829d101b49e4e102176011ca2374	71ee3a0b26b0956a6aafb37c9f69b65b6d3f9fc5	eaa04e5f34ea1549de04103c7d3133878c34a3793a9bd9f4659b55db7520f629

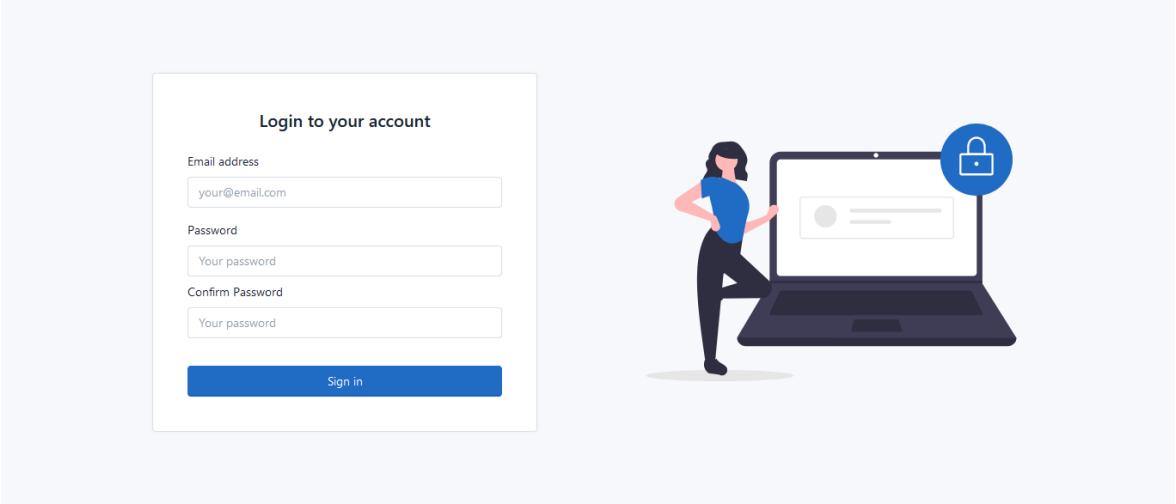
Hasil kalkulasi *file* yang telah dimanipulasi dengan menggunakan algoritma *K-Nearest Neighbor* dan *Support Vector Machine* untuk mengidentifikasi jenis *file* dapat dilihat pada Tabel 4.5 dan selengkapnya pada lampiran.

Tabel 4.5 Hasil Kalkulasi Machine Learning

Nama *File*	Metode	Hasil
CUT-akhir-10%-Logo-UII-Asli.png	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-Logo-UII-Asli.png	*Support Vector Machine*	pngw
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	*K-Nearest Neighbor*	pdf
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	*Support Vector Machine*	pdf
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	ods

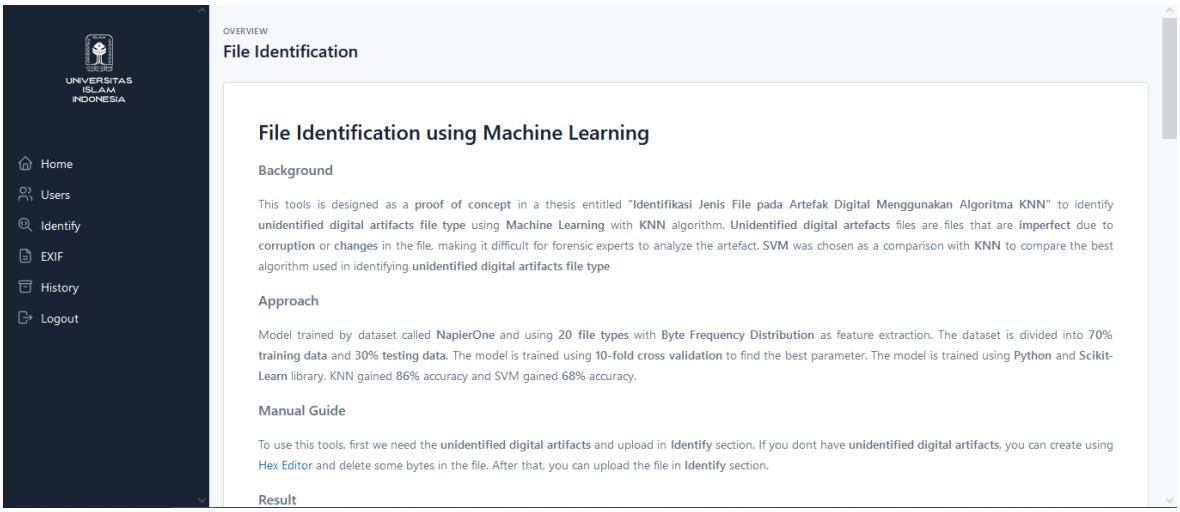
Dapat dilihat bahwa ada *file* yang menghasilkan keluaran yang sesuai serta tidak sesaui dan juga tidak dapat di proses karena kekurangan fitur sehingga tidak dapat diketahui jenis *file*nya.

Pada proses pengujian agar lebih mudah, maka dibuat sebuah aplikasi berbasis web untuk membantu mengidentifikasi *file* menggunakan *machine learning*. Berikut merupakan halaman login



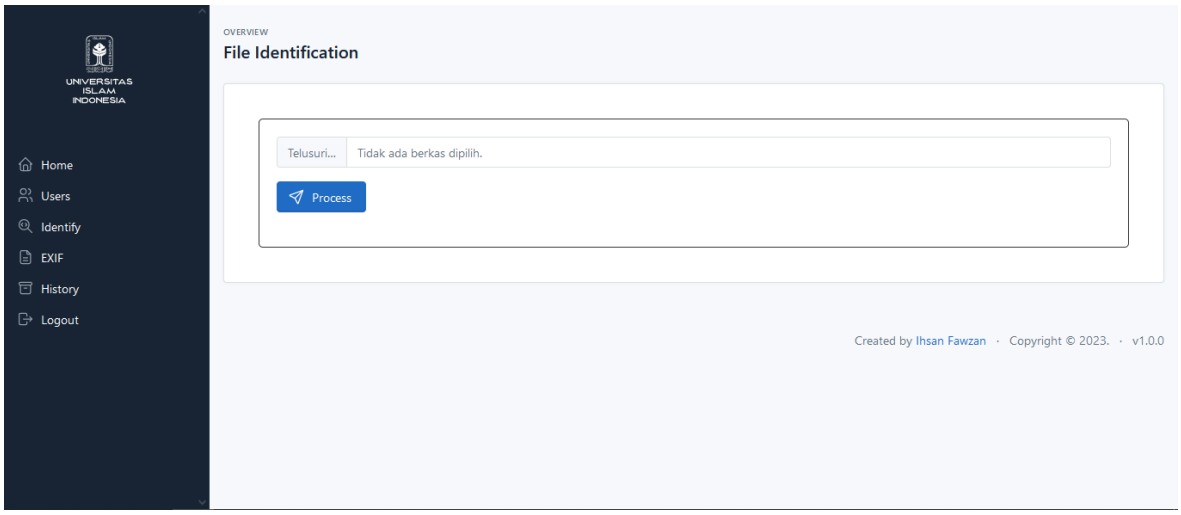
Gambar 4.3 Tampilan awal aplikasi

Sebelum masuk ke aplikasi, pengguna harus melakukan login terlebih dahulu dengan cara memasukan email dan juga password seperti pada Gambar 4.3. Jika email dan juga password cocok, maka selanjutnya akan diarahkan ke halaman home aplikasi.



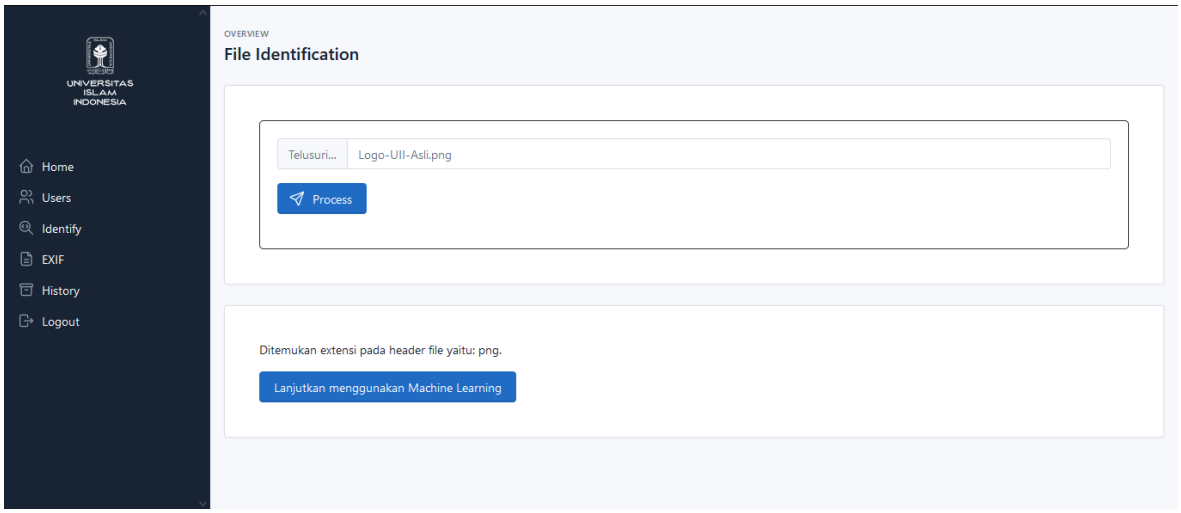
Gambar 4.4 Halaman home aplikasi

Setelah masuk ke aplikasi, maka akan ditampilkan halaman awal seperti Gambar 4.4 dan menu-menu yang dapat di akses.



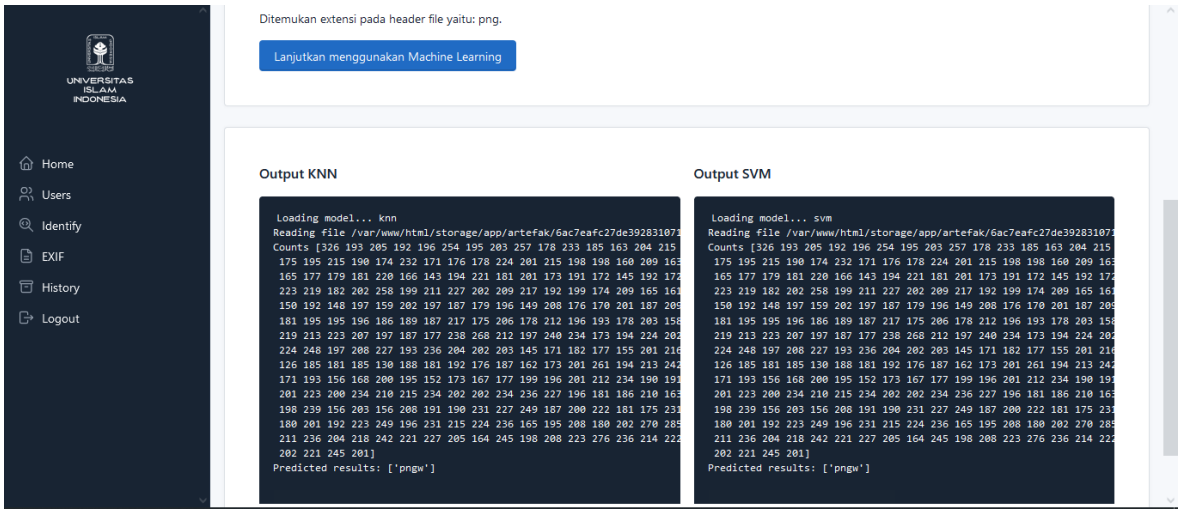
Gambar 4.5 Halaman identifikasi *file*

Gambar 4.5 merupakan tampilan untuk melakukan identifikasi *file* menggunakan *machine learning*. Langkah pertama yaitu memasukan *file* yang akan menjadi pembanding.



Gambar 4.6 Halaman identifikasi *file* setelah input *file*

Sebelum dilakukan proses identifikasi menggunakan *machine learning*, dilakukan identifikasi terlebih dahulu identifikasi menggunakan ekstensi *file* seperti pada Gambar 4.6.



Gambar 4.7 Halaman identifikasi *file* setelah pemrosesan dengan *machine learning*

Setelah menekan tombol Lanjutkan menggunakan *machine learning* seperti pada Gambar 4.7 maka proses identifikasi akan berjalan menggunakan *machine learning*. Selain menggunakan algoritma *K-Nearest Neighbor*, algoritma *Support Vector Machine* juga digunakan sebagai pembanding untuk melihat prediksi hasil *file* tidak teridentifikasi yang di inputkan.

Hasil penggunaan model *K-Nearest Neighbor* pada *dataset NapierOne* dengan menggunakan *byte frequency distribution* sebagai ekstraksi fitur berhasil mencapai akurasi 86% dapat dilihat pada lampiran.

Hasil penggunaan model *Support Vector Machine* pada *dataset NapierOne* dengan menggunakan *byte frequency distribution* sebagai ekstraksi fitur berhasil mencapai akurasi 86% dapat dilihat pada lampiran.

BAB 5
Kesimpulan dan Saran

5.1 Kesimpulan

Machine Learning khususnya algoritma *K-Nearest Neighbor* dan *Support Vector Machine* dapat digunakan untuk mengidentifikasi dan mengklasifikasikan jenis file yang tidak terdefinisi pada artefak digital. Berdasarkan penelitian yang telah dilakukan dapat dibuktikan bahwa penerapan Algoritma *K-Nearest Neighbor* dan Algoritma *Support Vector Machine* bersama dataset NapierOne serta teknik ekstraksi fitur *byte frequency distribution* untuk identifikasi jenis file dapat digunakan karena dapat mencapai tingkat akurasi sebesar 86%.

5.2 Saran

Penelitian ini masih jauh dari sempurna sehingga perlu dilakukan penelitian lebih lanjut untuk meningkatkan akurasi serta menggunakan *dataset* yang lebih besar. Berikut merupakan rekomendasi pengembangan penelitian ini pada penelitian selanjutnya.

1. Menambah dataset

   Menambahkan dataset dapat dilakukan untuk membuat model mempunyai lebih banyak referensi terhadap jenis file. Selain jenis file yang sudah digunakan pada penelitian ini, selanjutnya dapat menggunakan jenis file lainnya untuk menambah pengetahuan model sehingga memperbanyak jenis file yang dapat di identifikasi. Selain menambahkan dataset, perlu dilakukan verifikasi terlebih dahulu sehingga model yang akan di kembangkan memiliki referensi yang sesuai.

2. Menggunakan *Feature Reduction*

   Perlu melakukan *feature reduction* pada hasil ekstraksi fitur untuk mengurangi kompleksitas dan meningkatkan interpretabilitas.

3. Menggunakan *Deep Learning*

   Perlu melakukan percobaan menggunakan *deep learning* untuk melakukan kalkulasi dan membuat model. Penggunaan algoritma *Convolutional Neural Network* direkomendasikan untuk melakukan pengembangan dari algoritma yang sudah digunakan pada penelitian ini yaitu *K-Nearest Neighbor* dan *Support Vector Machine.*

Daftar Pustaka

Abdillah, M. F. (2022). *Analisis Perbandingan Data Recovery Menggunakan Tools Forensik Berbasis Open Source Pada Linux*.

Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*, *7*(4), 221–248. https://doi.org/10.1089/big.2018.0175

Alkasassbeh, M., & Al-Daleen, S. (2018). Classification of malware based on file content and characteristics. *ArXiv*.

Amirani, M. C., Toorani, M., & Beheshti, A. (2008). A new approach to content-based file type detection. *2008 IEEE Symposium on Computers and Communications*, *July 2008*, 1103–1108. https://doi.org/10.1109/ISCC.2008.4625611

Amirani, M. C., Toorani, M., Shirazi, A. A. B., & Beheshti, A. (2008). A new approach to content-based file type detection. *2008 IEEE Symposium on Computers and Communications*, *July 2008*, 1103–1108. https://doi.org/10.1109/ISCC.2008.4625611

Bairavasundaram, L. N., Goodson, G. R., Schroeder, B., Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2008). An analysis of data corruption in the storage stack. *FAST 2008 - 6th USENIX Conference on File and Storage Technologies*, 223–238.

Bhat, A., Likhite, A., Chavan, S., & Ragha, L. (2021). File Fragment Classification using Content Based Analysis. *ITM Web of Conferences*, *40*, 03025. https://doi.org/10.1051/itmconf/20214003025

Bhat, K., Lam, J. T., & Zulkernine, F. (2018). Content-based file type identification. *ICECE 2018 - 10th International Conference on Electrical and Computer Engineering*, *1*, 277–280. https://doi.org/10.1109/ICECE.2018.8636693

Charyyev, B., & Arslan, E. (2020). RIVA: Robust Integrity Verification Algorithm for High-Speed File Transfers. *IEEE Transactions on Parallel and Distributed Systems*, *31*(6), 1387–1399. https://doi.org/10.1109/TPDS.2020.2966616

Chern, F., Hechtman, B., Davis, A., Guo, R., Majnemer, D., & Kumar, S. (2022). *TPU-KNN: K Nearest Neighbor Search at Peak FLOP/s*. *NeurIPS*. http://arxiv.org/abs/2206.14286

Chizari, H., Razak, S. A., Majidi, M., & Salleh, S. bin. (2018). Computer forensic problem of sample size in file type analysis. *International Journal of Advanced Intelligence*

Paradigms, *11*(1–2), 58–74. https://doi.org/10.1504/IJAIP.2018.092951

Cunningham, P., & Delany, S. J. (2021). K-Nearest Neighbour Classifiers-A Tutorial. *ACM Computing Surveys*, *54*(6). https://doi.org/10.1145/3459665

Davies, S. R., Macfarlane, R., & Buchanan, W. J. (2022). NapierOne: A modern mixed file data set alternative to Govdocs1. *Forensic Science International: Digital Investigation*, *40*, 301330. https://doi.org/10.1016/j.fsidi.2021.301330

Dubettier, A., Gernot, T., Giguet, E., & Rosenberger, C. (2023). File type identification tools for digital investigations. *Forensic Science International: Digital Investigation*, *46*, 301574. https://doi.org/10.1016/j.fsidi.2023.301574

Fanani, G., Riadi, I., & Yudhana, A. (2022). Analisis Forensik Aplikasi Michat Menggunakan Metode Digital Forensics Research Workshop. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *6*(2), 1263. https://doi.org/10.30865/mib.v6i2.3946

Hafner, J. L., Deenadhayalan, V., Belluomini, W., & Rao, K. (2008). Undetected disk errors in RAID arrays. *IBM Journal of Research and Development*, *52*(4–5), 413–425. https://doi.org/10.1147/rd.524.0413

Hamdi, D., Iqbal, F., Baker, T., & Shah, B. (2017). Multimedia File Signature Analysis for Smartphone Forensics. *Proceedings - 2016 9th International Conference on Developments in ESystems Engineering, DeSE 2016*, 130–137. https://doi.org/10.1109/DeSE.2016.22

Haque, M. E., & Tozal, M. E. (2022). Byte embeddings for file fragment classification. *Future Generation Computer Systems*, *127*, 448–461. https://doi.org/10.1016/j.future.2021.09.019

Hawdon, J. (2021). Cybercrime: Victimization, Perpetration, and Techniques. *American Journal of Criminal Justice*, *46*(6), 837–842. https://doi.org/10.1007/s12103-021-09652-7

Hegarty, R., & Haggerty, J. (2015). SlackStick: Signature-Based File Identification for Live Digital Forensics Examinations. *2015 European Intelligence and Security Informatics Conference*, 24–29. https://doi.org/10.1109/EISIC.2015.28

Hiester, L. (2018). File Fragment Classification Using Neural Networks with Lossless Representations Networks with Lossless Representations. *Undergraduate Honors Theses*, 1–32. https://dc.etsu.edu/honors/454

Jenn, R. (2017). Understanding Metadata: What is Metadata, and What is it For? In *Cataloging & Classification Quarterly* (Vol. 55, Issues 7–8).

Jung, B. H., Bae, S. Il, Choi, C., & Im, E. G. (2020). Packer identification method based on

byte sequences. *Concurrency and Computation: Practice and Experience*, *32*(8), 1–11. https://doi.org/10.1002/cpe.5082

Karampidis, K., & Papadourakis, G. (2017). File Type Identification - Computational Intelligence for Digital Forensics. *The Journal of Digital Forensics, Security and Law*, *12*(2). https://doi.org/10.15394/jdfsl.2017.1472

Karampidis, K., Papadourakis, G., & Deligiannis, I. (2015). File Type Identification - A Literature Review. *9th International Conference on New Horizons in Industry Business and Education, NHIBE 2015*, *May 2015*, 141. http://nhibe2015.vs-net.eu/proceedings/papers/3_15_[P]0076.pdf

Kejahatan Siber di Indonesia Naik Berkali-kali Lipat | Pusiknas Bareskrim Polri. (n.d.). Retrieved June 16, 2023, from https://pusiknas.polri.go.id/detail_artikel/kejahatan_siber_di_indonesia_naik_berkali-kali_lipat

Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). Guide to Integrating Forensic Techniques into Incident Response. *The National Institute of Standards and Technology*.

Kim, G. Y., Paik, J. Y., Kim, Y., & Cho, E. S. (2022). Byte Frequency Based Indicators for Crypto-Ransomware Detection from Empirical Analysis. *Journal of Computer Science and Technology*, *37*(2), 423–442. https://doi.org/10.1007/s11390-021-0263-x

Kizza, J., & Migga Kizza, F. (2011). Digital Evidence and Computer Crime. In *Securing the Information Infrastructure*. https://doi.org/10.4018/978-1-59904-379-1.ch015

Konaray, S. K., Toprak, A., Pek, G. M., Akçekoce, H., & Deniz, K. (2019). Detecting File Types Using Machine Learning Algorithms. *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1–4.

Krioukov, A., Bairavasundaram, L. N., Goodson, G. R., Srinivasan, K., Thelen, R., Arpaci-Dusseau, A. C., & Arpaci-Dusseau, R. H. (2008). Parity lost and parity regained. *FAST 2008 - 6th USENIX Conference on File and Storage Technologies*, *May*, 127–141.

Lewulis, P. (2021). *Digital forensic standards and digital evidence in Polish criminal proceedings . An updated definition of digital evidence in forensic science . August*. https://doi.org/10.1504/IJESDF.2021.10034988

Masoumi, M., Keshavarz, A., & Fotohi, R. (2021). File fragment recognition based on content and statistical features. *Multimedia Tools and Applications*, *80*(12), 18859–18874. https://doi.org/10.1007/s11042-021-10681-x

McDaniel, M., & Heydari, M. H. (2003). Content based file type detection algorithms. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, HICSS 2003*. https://doi.org/10.1109/HICSS.2003.1174905

Meghanathan, N. (2013). Network security: Attacks and controls. In *Network Security Technologies: Design and Applications* (Issue December, pp. 174–203). https://doi.org/10.4018/978-1-4666-4789-3.ch011

Millington, E. (2024). *REvil, Software S0496 | MITRE ATT&CK®*. https://attack.mitre.org/software/S0496/

Min, D., Ko, Y., Walker, R., Lee, J., & Kim, Y. (2022). A Content-Based Ransomware Detection and Backup Solid-State Drive for Ransomware Defense. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *41*(7), 2038–2051. https://doi.org/10.1109/TCAD.2021.3099084

Mittal, G., Korus, P., & Memon, N. (2021). FiFTy: Large-Scale File Fragment Type Identification Using Convolutional Neural Networks. *IEEE Transactions on Information Forensics and Security*, *16*(1), 28–41. https://doi.org/10.1109/TIFS.2020.3004266

Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Multivariate Statistical Machine Learning Methods for Genomic Prediction. In *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. https://doi.org/10.1007/978-3-030-89010-0

Neaimi, M. Al, Hamadi, H. Al, Yeun, C. Y., & Zemerly, M. J. (2020). Digital Forensic Analysis of Files Using Deep Learning. *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)*, 1–4. https://doi.org/10.1109/ICSPIS51252.2020.9340141

Nettles, R. A., Merulla, C., & Warzala, S. (2019). *Data Manipulation: Attacks and Mitigation – CSIAC*. https://csiac.org/articles/data-manipulation-attacks-and-mitigation/

Nguyen, K., Tran, D., Ma, W., & Sharma, D. (2017). Decision tree algorithms for image data type identification. *Logic Journal of the IGPL*, *25*(1), 67–82. https://doi.org/10.1093/jigpal/jzw045

Nuh Al-Azhar, M. (2012). *Digital Forensic Practical Guidelinse for Computer Investigation*.

PowerDuke, Software S0139 | MITRE ATT&CK®. (2020). https://attack.mitre.org/software/S0139/

Raharjo, B. (2013). Sekilas Mengenai Forensik Digital. *Jurnal Sosioteknologi*, *12*(29), 384–387. https://doi.org/10.5614/sostek.itbj.2013.12.29.3

Riskiyadi, M. (2020). Investigasi Forensik Terhadap Bukti Digital Dalam Mengungkap Cybercrime. *Cyber Security Dan Forensik Digital*, *3*(2), 12–21. https://doi.org/10.14421/csecurity.2020.3.2.2144

Rizal, R., Ruuhwan, R., & Chandra, S. (2020). Signature File Analysis Using The National Institute Standard Technology Method Base on Digital Forensic Concepts. *Jurnal Informatika Universitas Pamulang*, *5*(3), 364. https://doi.org/10.32493/informatika.v5i3.6073

Sah, S. (2020). Machine Learning: A Review of Learning Types. *ResearchGate*, *July*. https://doi.org/10.20944/preprints202007.0230.v1

Sekeroglu, B., Hasan, S. S., & Abdullah, S. M. (2020). Comparison of Machine Learning Algorithms for Classification Problems. *Advances in Intelligent Systems and Computing*, *944*(2), 491–499. https://doi.org/10.1007/978-3-030-17798-0_39

Sidik Asyaky, M., Widiyasono, N., & Gunawan, R. (2018). Analisis dan Perbandingan Bukti Digital Aplikasi Instant Messenger Pada Android. *SinkrOn : Jurnal & Penelitian Teknik Informatika*, *3*(1), 220–230. https://www.jurnal.polgan.ac.id/index.php/sinkron/article/view/176

Sourabh, S., Chauhan, D., Singh, V., & Chauhan, M. (2021). Analysis of Digital Data by File Signature Method on Android Version 9. *Recent Advances in Computer Science and Communications*, *15*(8), 1074–1081. https://doi.org/10.2174/2666255813666201216114643

Taylor, P., Security, B., Brenton, M., & Security, Z. G. I. (2024). *WhisperGate, Software S0689 | MITRE ATT&CK®*. https://attack.mitre.org/software/S0689/

Understanding File Systems- Kingston Technology. (2023). https://www.kingston.com/en/blog/personal-storage/understanding-file-systems

Valkenborg, D., Rousseau, A.-J., Geubbelmans, M., & Burzykowski, T. (2023). Support vector machines. *American Journal of Orthodontics and Dentofacial Orthopedics*, *164*(5), 754–757. https://doi.org/10.1016/j.ajodo.2023.08.003

Venkata, S. K., & Green, A. (2019). *Computational Intelligence to aid Text File Format Identification*. 1–14. http://eprints.rclis.org/38969/

Vulinovic, K., Ivkovic, L., Petrovic, J., Skracic, K., Pale, P., Petrović, J., Skracic, K., & Pale, P. (2019). Neural Networks for File Fragment Classification. *2019 42nd International Convention on Information and Communication Technology,*

Electronics and Microelectronics (MIPRO), *null*, 1194–1198. https://doi.org/10.23919/MIPRO.2019.8756878

Wang, F., Quach, T. T., Wheeler, J., Aimone, J. B., & James, C. D. (2018). Sparse Coding for N-Gram Feature Extraction and Training for File Fragment Classification. *IEEE Transactions on Information Forensics and Security*, *13*(10), 2553–2562. https://doi.org/10.1109/TIFS.2018.2823697

Wang, Y., Su, Z., & Song, D. (2018). File Fragment Type Identification with Convolutional Neural Networks. *Proceedings of the 2018 International Conference on Machine Learning Technologies*, 41–47. https://doi.org/10.1145/3231884.3231889

Xiong, L., & Yao, Y. (2021). Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm. *Building and Environment*, *202*(May), 108026. https://doi.org/10.1016/j.buildenv.2021.108026

Ye, Y., Li, T., Zhu, S., Zhuang, W., Tas, E., Gupta, U., & Abdulhayoglu, M. (2011). Combining file content and file relations for cloud based malware detection. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 222–230. https://doi.org/10.1145/2020408.2020448

Zhang, C., & Green, R. (2015). Communication security in internet of thing: Preventive measure and avoid DDoS attack over IoT network. *Simulation Series*, *47*(3), 8–15.

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 1–7. https://doi.org/10.21037/atm.2016.03.37

Zheng, N., Wang, J., Wu, T., Xu, M., Wang, J., & Xu, M. (2015). A Fragment Classification Method Depending on Data Type. *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, *null*, 1948–1953. https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.288

Lampiran

Table 1 Metadata *File* Asli

Nama *File*	Ekstensi	MD5	SHA1	SHA256
Logo-UII-Asli.png	png	ad495618f18fc70815d5bf6ba34898ff	65b26872eb36304a13759770132750cfb0dc0bea	6ac7eafc27de392831071fe05e112dea2fd681bf182f2bdb0dc5776860efca8f
file-example_PDF_1MB.pdf	Pdf	ee7ac8084eeab08035fdcb47bfa81931	cec8ad914b1e9db83626b98e8d98512616975fdf	5e4d40fcd8b22453a5da2d32533b128f2565f3fc7a4d1647a93c86cdbb4be37a
file-example_PDF_500_kB.pdf	pdf	c7d354ff10c868752f1b032e5b0e55e8	6d730748d689f1c6c71bc5c5f7a5b1a65c533242	e83014e71fc8e772b7689a3f1c8628a2ef2852a1a38e31bddaf823615570e709
file-sample_100kB.doc	doc	6f7c0f42d73206b0d07ded2d2c210287	709a662583dd7ac55515de68f5b2dea424b09694	45e07cf623e634b96b3e82aeaf7ecc74efd03a05808bf1eb69ceef8a9df7d135
file-sample_100kB.docx	docx	515a1ba1611a2212b4b6d2ae121898ad	6214855065576ec84a0ff1f6fa3a4a69be20122e	332794745f5622beb843399e988a12b2d388c97c92ff1860f847b4aeadc5e0a0
file-sample_150kB.pdf	pdf	5afaf79789a776d81ec91ccbdc9fdaba	6703901978dcb3dbf2d9915e1d3e066cfe712b0a	38c9792d725c45dd431699e6a3b0f0f8e17c63c9ac7331387ee30dcc6e42a511
file-sample_1MB.doc	doc	703739859691df910c12f9c0dd7243c9	8f3498bfbe75fb34e809fc7ced3f4cc6f4acc92c	c560136e2a2b7036523f69efdb4e9cdf369abe167ba3a095e26d74e261774b20
file-sample_1MB.docx	docx	73659a2e1fe6fa69d0a250e0ae497913	688c74675f84ecae08b1f73c19875af264e34f74	27cd24f7f6e1e86449c1efc75c103acbb717733be5a36377cceb59e77be9d97c
file-sample_500kB.doc	doc	f66e2b042497390f2feed4a95ff9d613	d8e6c9825d33914e0e157c0b34a3067ab761c5a8	6cd47bd7261f1cc0c77b51d9ccb2ce89eb042e20ebbed995

				5447c929aaf6befc
file-sample_500kB.docx	docx	d4504c91b0704eaef0c8d098d4d3966c	9a0881226460bf5d0ea17b29a843c0320b9dd11e	0641b145758f807221ab814d77c9f5fccaf01a07f6f750b2864590545094becd
*file*_example_GIF_1MB.gif	gif	a952c97cd96967386b54927eb25dedfd	703464171ff52e7359e1e3dec7f8cdd5b85094ef	08cac1418c3361015211c716d4499655e780db8a1f4d7718dcd4cb882a3210f7
*file*_example_GIF_3500kB.gif	gif	bfa20cb931091581fd66d3154c417d3f	e8f22f74c70a40b34a3751b5d651ff889445ed00	0895dee19e75182d607811f50889f73df3b7aa804b83bb18cd25dd074aa497de
*file*_example_GIF_500kB.gif	gif	76faedf4c58dcadd6077c9509af2e9ad	7f58090196a9be6b29f361236e0086fff1320927	c763b9d5b7cbdf9bdcb851c8f97cfac45792cae31af35799ec6f91b900e97459
*file*_example_JPG_100kB.jpg	jpg	5aa858befc56dee1effe7eae1c0f88bb	f0fee71865babe4df97088370e44b7aa76d949d0	88aeb1f4467bd1e50cf624de972fbf3f40801632fedb64aaa7b1a8a9ef786fc6
*file*_example_JPG_1MB.jpg	jpg	38e8bedb39d8e4338c790e9af34467e4	26d4ade8d9be09ee881307a30990dab71b8b12d3	683a8528125ca09d8314435c051331de2b4c981c756721a2d12c103e8603a1d2
*file*_example_JPG_2500kB.jpg	jpg	63d148c66915aba6702a87539d37c845	9a2fc2007d9c1b42692ac5df61781da67439d824	b6630f7d0ff76f53f21b472f0d383b41a0b8a730a29282f12db435dc390dfdeb
*file*_example_JPG_500kB.jpg	jpg	1589d7f466220b7604858b1eaabdd0d2	25472d494500707b4510868df469352dc7cd1845	838e346997ab5f2dd6745e9e536de6f9cd68965088354597f2fba016ad40ab2c
*file*_example_MP3_1MG.mp3	mp3	2aded9b378dde5cfab310ec69f59fd24	e493d393c751d2bfa8807af52d306552b73ecae4	90ce3b7c9dfcce6aafcb2dcfc3fc496dab6ba8106b61531b05d2c57a4be1640e
*file*_example_MP3	mp3	61b62be9d171559	b4c1db2e6cc0c62c	0362a64f6c0dcb73

_2MG.mp3		8003e71ec9ea52010	3d283e8f919e48e69f30bf44	47b4af8bb22828d765912d58a8becea58ff610460cd8396b
*file*_example_MP3_5MG.mp3	mp3	00ba46e6fd66e7eae9ec9c43d0828580	c9fa731192bec4756782d94a5c44ff41eb42d72e	b36fd7c68d326e2cb59bfd3546de59f8af911ee215ac81d1d8e0ed9d04d32ef0
*file*_example_MP3_700KB.mp3	mp3	cdd5a8079dda1acdda848defdd1c256a	dcaa8815859ce4c1c02c5f3b8bf01c79d0053c1f	9a2270d5964f64981fb1e91dd13e5941262817bdce873cf357c92adbef906b5d
*file*_example_MP4_1280_10MG.mp4	mp4	10c918b1d01aea85864ee65d9e0c2305	afed64ed7a90fed976cf76476432105e722049c8	8ab080c1406dff77f8897955cf977e9ad779e40ab3a07bc2f8694fbd2fc2be21
*file*_example_MP4_1920_18MG.mp4	mp4	d7a3decdb6280eb0ef3a059ac35ea0ee	a8ddbbcacc224b427f5b7ca14bdf41b692291730	d6617a009c0c6c9aebf7398d43cad6d1985ddc1b9ab0479e2ea977362b8af5b0
*file*_example_MP4_480_1_5MG.mp4	mp4	d9061d3da8601932e98f79ec8ba1c877	55508bc98a00f615dbe9bd4c84a253ba4238b021	71944d7430c461f0cd6e7fd10cee7eb72786352a3678fc7bc0ae3d410f72aece
*file*_example_MP4_640_3MG.mp4	mp4	82d9b86b01560ce17d310f3fd2a79ca9	1385f0808b7ef1600b6c39244917ef1df2ce6652	5e70b96ad27dc8581424be7069ee9de8da9388b716e6fe213d88385f19baf80a
*file*_example_ODS_10.ods	ods	39c9a4e7be91375d6eac5981f800a723	bfd608440f76d2583ef08aff84205d4d95cd46a0	be74bcfeba5fa5bb3cd400c54c6d0d9172074505b80e1a96e32ac08592f43c75
*file*_example_ODS_100.ods	ods	aa3477cb648ff8006e5fed4916387f3e	e7f9fb286e6ac8820c493fe0837ae103cae745d0	53eaf2997a587b4590ad712ff1cdafde298b32b5f49adb2f3c53e799a0e83963
*file*_example_ODS_1000.ods	ods	c5251a078ea94e3b79a99b2705bf8e9e	20209af947e0c17f3fd7adf940e9e1807dfc2d1d	a902b02c4373dad053343762dd1f6927db0a4c6355f7780363f53c0ba6c2e5f9
*file*_example_PNG	png	e6669697d9b5d7d	1dc7401f222611ac	0af3a12f59484451

_1MB.png		4370e7c20ffd6ddff	3ebe1b1ba77101299cbb0400	00da03937e9baf74c6dab503ed20afaa89b4b59a32ca5704
*file*_example_PNG_2100kB.png	png	d165d5cfdc69b71126249b2983ff53af	98f444c7e11071a12daf77b86554bfa31ceceff8	6b28dfa3d3be99363402a6358103f545807cb6ce836a8ae2d3e066ef867513f7
*file*_example_PNG_3MB.png	png	31ae6496aae1f9b1d5daca6fe81679ae	727087f19369e31ed3fc80c4f1df9e9030f89103	9a6a4c465d10968f5daf41e3f80cc0b08e1d0b8adc1ed4654d560aae87fb4886
*file*_example_PNG_500kB.png	png	1e02a85bee78984d20942bf213f4c461	31f072caa53a8657a85547f2aac842f4006e2b24	aad96d410d92b5589d41e8462507e3af57682022db3d3711a236c0245fcf296e
*file*_example_PPT_1MB.ppt	ppt	8bd09d8d6ad6a5431010a250d0fa27e4	3d30846c0c62832fd05d1c424fb650cf4da6ae23	b709debb365a5437f2472f350745ed2f8a6890d7cb3d81e6750f2d5dd44625c9
*file*_example_PPT_250kB.ppt	ppt	addf63d147411c6ec1a2b594fc36b550	cb8ea4114148bc57e7317b3d0aa72ebe53dbe386	1e4f6b07f0147358b4e6096d70ea7b1fb6d88bab18b2fb2a1b580185024ed158
*file*_example_PPT_500kB.ppt	ppt	b1474954701b41b5d0c01851ee34c93b	7e749d022bbf031b341a5880d524ba32611d1a13	5883205ac9644abfcd128f1338723decb5ae59437a8becc4f13b090f31634bf7
*file*_example_XLSX_10.xlsx	xlsx	823b3f7a9b47992e039ff459b121bc12	30d322488f8357ace0daec643311f96796c54c9f	e542d981f0d9fefff85f0f2904d598f8c1ff5053e325c5607a76c331731418c0
*file*_example_XLSX_100.xlsx	xlsx	83dac9dda2adc47c4f5fb18481f817d6	6b2be0033f57839727427f0bd6f7aaf4eb1cdae0	716fb9d3593c2b68ed2319ad10107e6783e880a3d14a136d0c912158c23bbfe6
*file*_example_XLSX_1000.xlsx	xlsx	60f55bf1cb2676af1407d74b4c2b8291	0dce77a811e9f631913c88e1b04391a868fcc93d	e865b9075f2c365ca34348753a828b45d265d64089eff36e

				e940e8450d6f9664
*file*_example_XLSX_50.xlsx	xlsx	5c256f2de57126cac5ac629ce42ca754	c94997094ad00f9f38017fd94a1d44a1241003b6	c6c3839758e6a24e6199dd663f85d10cf833005f9393ee368361cb58b858b9cc
*file*_example_XLSX_5000.xlsx	xlsx	ad24efa40026373e25c9da8360d50133	a6b1e5924e3b98ed5668fb1d0228569e3f0fefe7	678b8763910394479084f263667d45fcf2b7e345bef274ccfc1c7f8c5e36adcd
*file*_example_XLS_10.xls	xls	f7c4094f7edab169e921fa155a22fba4	e0cdb9c8a7da39f9787729345fd9bd617d02d539	771b2e98db13f1f683eaf37dd65364a9297ad32679301947f4bea68c3e625244
*file*_example_XLS_100.xls	xls	17330b6d18ddbbe04b5fa9e3c76398c6	578248d008cdcda825c0d2ef705ba03e658352a4	8347f0feb8e84572805bdcb564e573b4a670305f81d5e91b8744311ef10f10f6
*file*_example_XLS_1000.xls	xls	add49bc35ed077dad3159cf1bab13cfd	857894898dc06a2bc76a2e20562d0ad815a14f07	4e8ff8935a5acd6ade242bdcfa33319a476668c2bc9dc49b8ba73fcc7e8bf73a
*file*_example_XLS_50.xls	xls	d732eef37d941722c8b23221bf173161	c8e7174ee99d1878d033bad521ba03d0d459d96f	4383c16490d51729fddb32112c410c5da68d64b452ad69f383c0d4a5777564cb
*file*_example_XLS_5000.xls	xls	5b39869b3e945cf909959f23f83af3ed	3a0d4a8ecb77244d1580b15cb9f0766c8317b073	e9ead91638c88c48ce657dfc3bb126be62c21365a8566d68a88c5a9c43effab0

Table 2 Hasil Kalkulasi Machine Learning

Nama *File*	Metode	Hasil
Logo-UII-Asli.png	*K-Nearest Neighbor*	pngw
Logo-UII-Asli.png	*Support Vector Machine*	pngw
file-example_PDF_1MB.pdf	*K-Nearest Neighbor*	gifw
file-example_PDF_1MB.pdf	*Support Vector Machine*	pdf

file-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	ods
file-example_PDF_500_kB.pdf	*Support Vector Machine*	bmp
file-sample_100kB.doc	*K-Nearest Neighbor*	doc
file-sample_100kB.doc	*Support Vector Machine*	doc
file-sample_100kB.docx	*K-Nearest Neighbor*	epub
file-sample_100kB.docx	*Support Vector Machine*	oxps
file-sample_150kB.pdf	*K-Nearest Neighbor*	pdf
file-sample_150kB.pdf	*Support Vector Machine*	pdf
file-sample_1MB.doc	*K-Nearest Neighbor*	mp3
file-sample_1MB.doc	*Support Vector Machine*	bmp
file-sample_1MB.docx	*K-Nearest Neighbor*	pngw
file-sample_1MB.docx	*Support Vector Machine*	oxps
file-sample_500kB.doc	*K-Nearest Neighbor*	oxps
file-sample_500kB.doc	*Support Vector Machine*	pptx
file-sample_500kB.docx	*K-Nearest Neighbor*	oxps
file-sample_500kB.docx	*Support Vector Machine*	oxps
*file*_example_GIF_1MB.gif	*K-Nearest Neighbor*	gifw
*file*_example_GIF_1MB.gif	*Support Vector Machine*	bmp
*file*_example_GIF_3500kB.gif	*K-Nearest Neighbor*	pdf
*file*_example_GIF_3500kB.gif	*Support Vector Machine*	oxps
*file*_example_GIF_500kB.gif	*K-Nearest Neighbor*	pngw
*file*_example_GIF_500kB.gif	*Support Vector Machine*	bmpw
*file*_example_JPG_100kB.jpg	*K-Nearest Neighbor*	oxps
*file*_example_JPG_100kB.jpg	*Support Vector Machine*	bmpw
*file*_example_JPG_1MB.jpg	*K-Nearest Neighbor*	oxps
*file*_example_JPG_1MB.jpg	*Support Vector Machine*	bmpw
*file*_example_JPG_2500kB.jpg	*K-Nearest Neighbor*	oxps
*file*_example_JPG_2500kB.jpg	*Support Vector Machine*	bmp
*file*_example_JPG_500kB.jpg	*K-Nearest Neighbor*	oxps
*file*_example_JPG_500kB.jpg	*Support Vector Machine*	pngw
*file*_example_MP3_1MG.mp3	*K-Nearest Neighbor*	mp3
*file*_example_MP3_1MG.mp3	*Support Vector Machine*	pptx
*file*_example_MP3_2MG.mp3	*K-Nearest Neighbor*	apk
*file*_example_MP3_2MG.mp3	*Support Vector Machine*	apk
*file*_example_MP3_5MG.mp3	*K-Nearest Neighbor*	exe
*file*_example_MP3_5MG.mp3	*Support Vector Machine*	apk
*file*_example_MP3_700KB.mp3	*K-Nearest Neighbor*	mp3
*file*_example_MP3_700KB.mp3	*Support Vector Machine*	pptx
*file*_example_MP4_1280_10MG.m	*K-Nearest Neighbor*	ods

p4		
*file*_example_MP4_1280_10MG.mp4	*Support Vector Machine*	gifw
*file*_example_MP4_1920_18MG.mp4	*K-Nearest Neighbor*	oxps
*file*_example_MP4_1920_18MG.mp4	*Support Vector Machine*	ods
*file*_example_MP4_480_1_5MG.mp4	*K-Nearest Neighbor*	exe
*file*_example_MP4_480_1_5MG.mp4	*Support Vector Machine*	gifw
*file*_example_MP4_640_3MG.mp4	*K-Nearest Neighbor*	exe
*file*_example_MP4_640_3MG.mp4	*Support Vector Machine*	gifw
*file*_example_ODS_10.ods	*K-Nearest Neighbor*	ods
*file*_example_ODS_10.ods	*Support Vector Machine*	ods
*file*_example_ODS_100.ods	*K-Nearest Neighbor*	ods
*file*_example_ODS_100.ods	*Support Vector Machine*	ods
*file*_example_ODS_1000.ods	*K-Nearest Neighbor*	ods
*file*_example_ODS_1000.ods	*Support Vector Machine*	ods
*file*_example_PNG_1MB.png	*K-Nearest Neighbor*	pngw
*file*_example_PNG_1MB.png	*Support Vector Machine*	oxps
*file*_example_PNG_2100kB.png	*K-Nearest Neighbor*	png
*file*_example_PNG_2100kB.png	*Support Vector Machine*	oxps
*file*_example_PNG_3MB.png	*K-Nearest Neighbor*	png
*file*_example_PNG_3MB.png	*Support Vector Machine*	oxps
*file*_example_PNG_500kB.png	*K-Nearest Neighbor*	pngw
*file*_example_PNG_500kB.png	*Support Vector Machine*	oxps
*file*_example_PPT_1MB.ppt	*K-Nearest Neighbor*	apk
*file*_example_PPT_1MB.ppt	*Support Vector Machine*	bmp
*file*_example_PPT_250kB.ppt	*K-Nearest Neighbor*	ppt
*file*_example_PPT_250kB.ppt	*Support Vector Machine*	pptx
*file*_example_PPT_500kB.ppt	*K-Nearest Neighbor*	apk
*file*_example_PPT_500kB.ppt	*Support Vector Machine*	bmp
*file*_example_XLSX_10.xlsx	*K-Nearest Neighbor*	gifw
*file*_example_XLSX_10.xlsx	*Support Vector Machine*	docx
*file*_example_XLSX_100.xlsx	*K-Nearest Neighbor*	gifw
*file*_example_XLSX_100.xlsx	*Support Vector Machine*	docx
*file*_example_XLSX_1000.xlsx	*K-Nearest Neighbor*	gifw
*file*_example_XLSX_1000.xlsx	*Support Vector Machine*	mp3

*file*_example_XLSX_50.xlsx	*K-Nearest Neighbor*	gifw
*file*_example_XLSX_50.xlsx	*Support Vector Machine*	mp4
*file*_example_XLSX_5000.xlsx	*K-Nearest Neighbor*	gifw
*file*_example_XLSX_5000.xlsx	*Support Vector Machine*	oxps
*file*_example_XLS_10.xls	*K-Nearest Neighbor*	Does not meet 256 features: 190 found
*file*_example_XLS_10.xls	*Support Vector Machine*	Does not meet 256 features: 190 found
*file*_example_XLS_100.xls	*K-Nearest Neighbor*	Does not meet 256 features: 219 found
*file*_example_XLS_100.xls	*Support Vector Machine*	Does not meet 256 features: 219 found
*file*_example_XLS_1000.xls	*K-Nearest Neighbor*	ppt
*file*_example_XLS_1000.xls	*Support Vector Machine*	doc
*file*_example_XLS_50.xls	*K-Nearest Neighbor*	Does not meet 256 features: 214 found
*file*_example_XLS_50.xls	*Support Vector Machine*	Does not meet 256 features: 214 found
*file*_example_XLS_5000.xls	*K-Nearest Neighbor*	exe
*file*_example_XLS_5000.xls	*Support Vector Machine*	ppt

Table 3 Metadata *File* Manipulasi

Nama *File*	Ekstensi	MD5	SHA1	SHA256
CUT-akhir-10%-Logo-UII-Asli.png	png	31775bcd0fa0a6abb0c7b732f84c078c	53a0866a45c14ee94b564eeb790041a30f63a0b2	18216ba795c1ea978914381a1450872b328c0fb7232d9738f723bebd305a35a9
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	pdf	0c4b89cfb5b891b4275144ac09728c77	ae1599788d0190b2f9af124b03f32c1a1ae9954f	dd921de91700f7c2440be3abd7f991f5c5b29ca6568383fb5a0dd4650d709c3f
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	pdf	494f85160d32cffda47a1d3d5c0866fa	5475a31007d80dfd2b329f63bdc4238b56f0d468	d742eb5b3313f821280be9656285f6c3bb4b79ad421655a46630dcb8f641602c
CUT-akhir-10%-*file*-	doc	1ccb829d101b49e4e102176011ca2374	71ee3a0b26b0956a6aafb37c9f69b65b	eaa04e5f34ea1549de04103c7d313387

sample_100kB.doc			6d3f9fc5	8c34a3793a9bd9f4659b55db7520f629
CUT-akhir-10%-*file*-sample_100kB.docx	docx	f899b56682533b0b1f4cb61ecf980bef	1ea25cd22ed8051596aaa8c98bfdc840aec93c56	4a729de00ee7336bcc2daf2176e8c77bba3f590c70cd25d6cfc52bedbe50e7a5
CUT-akhir-10%-*file*-sample_150kB.pdf	pdf	cb895800795bba9efe4f652c443f1fca	f096f2c24da8a679d98ba6dcb02dfb92cd552f58	fc775ce9c8c37f5904b5f65366a5208ea9212f7ca20e879b78c38c579ebbb1e8
CUT-akhir-10%-*file*-sample_1MB.doc	doc	380ad7693ac1bba9252747cb2eb10494	6ae8635be89c98a03384496ceb054d1bb0ec1f57	e68a2339a4270007a0b776c912f8471b66015e04aee9260d0ef4a392be4e797e
CUT-akhir-10%-*file*-sample_1MB.docx	docx	e3efd3da29c7d4cbd1c8d17ea1efab2b	f2b34be5c8c381dbe95d0e49a04e8f84400cba9f	7de6217abff7c49c90cf54676cf9cc16a83354571e0d4628ba0a041bb2e5d4fc
CUT-akhir-10%-*file*-sample_500kB.doc	doc	4c3f88de664c59ba1eb08be8b474ac36	7eb2879acea8bf762d0e2469af956c3ded7c93b0	a3b02b5160626a736c1573965b48d64769f099d9882cc08df595df6c82dace1a
CUT-akhir-10%-*file*-sample_500kB.docx	docx	819cdf45ec2760d444ea75a6e17dee5b	7fe646199579ba705b1ba634da8b6d5e65ee2f10	f5910b43204583ea039bc16b301d3abe1807bc38c77d7ea1a18d1fd0fd293779
CUT-akhir-10%-*file*_example_GIF_1MB.gif	gif	0215608860c04cd500a1dc04f271d20c	f76a252df0d96ec18590e171744ef6c714293471	1fd4d704c7fea45178180916aef430e3e5a02f70692c0c351189e0021df8507c
CUT-akhir-10%-*file*_example_GIF_3500kB.gif	gif	96d20281204b77451e8f80b270432b40	0dce5d524448cc51babd8fea3c1eda745a46f554	e89dc4e97809c89e0b092e5fc4e6df6be7f9e786cfad6802d07622b1bb7f1a6e
CUT-akhir-10%-*file*_example_GIF_500kB.gif	gif	42c6938e92b53948c384f762ee3a8d76	3cecd0feb663ac7d421ef798ce5924002b5f7408	784d3e443e5b930ba9b03c721a28b654ef44477ffc61ace13dfe502d89ab9512
CUT-akhir-10%-*file*_example_JPG_	jpg	8702db631801fdb1b44cbcad8e668f95	ca1859b922731714e82d0c8def3ecdd0	c5486d95a7e148c62d298b2e99055f04

100kB.jpg			832b1a65	d321247866d6aadbf984ecf8240e6f54
CUT-akhir-10%-*file*_example_JPG_1MB.jpg	jpg	f4d5da5e04ea908d0c02dcdbc79f661d	e86d17adadda6eb542019aedacec4fec20d077ef	63bd2d0763d7ceaba0c8dafe92b88ef3dfe366abe8e8ea7dd80504291e235893
CUT-akhir-10%-*file*_example_JPG_2500kB.jpg	jpg	f9ed8568756945ad93164c11356e610a	699ad6b67e562b2fb8fb72bbfe548d8b3269df17	07cff22f005e620a7adf6b9635be98c661ce03b0674033be208b41e2c300114b
CUT-akhir-10%-*file*_example_JPG_500kB.jpg	jpg	227c057e68944e6355fd55aed009b784	9d722d9671caaee78a201071444ff752f878478b	3be00b24be48948e429030409b05503459f8abb7efc3cefcf9bb507f44f0db85
CUT-akhir-10%-*file*_example_MP3_1MG.mp3	mp3	eab6873fc4ff0a4f12af10db284c7bf4	d195001b7808e7715f8846c638caac50452dd938	4723b1909e2379ca7102a9f531c7dde106a7e0e2941b0baf6324c5e7e3c347c3
CUT-akhir-10%-*file*_example_MP3_2MG.mp3	mp3	5c6805e358c4ebb01ffcf7dfbdf94e92	3e8b00dd70dfb3df753e82834ac9cdd516609a62	94733c460ffdce38ee6c2c21125a3728e0cb142ebe7f09f0623cfe6d054cb293
CUT-akhir-10%-*file*_example_MP3_5MG.mp3	mp3	6ecc3310982ec79ef9ef22a96ec1a79e	078dc4e3f87d5d98a786455ea5c1af01c5f92055	e1b267fd9f4942daee756fed6b13372fb27b8c3f87f92d00d44a8d9ab7174fab
CUT-akhir-10%-*file*_example_MP3_700KB.mp3	mp3	ac91cb009cf5e7c48026d7d9991f2e54	75e5bec5ac2005d5717f7ad6288c857cac449bd8	a629b40d6298caf0561e73c0b1204b3a9a2d6c7a0f7af5f388722282facd7ff8
CUT-akhir-10%-*file*_example_MP4_1280_10MG.mp4	mp4	5c1d716c7e6b041537aa78c2632f4125	097b84644968348b6fb48d32170b2024833e0f2c	ff4006ae6a2a276b6c9606934204196 87d53ee2a1371abc0c2f41dbc56f25c21
CUT-akhir-10%-*file*_example_MP4_1920_18MG.mp4	mp4	964bf7af9c9c23145f4e6cdf8ebfaf55	1d5024bf1a413bbf48e16f2a48a654740a7169ef	11a6033def6020f155fcae8f53e4c492e9f4be96eb18635d2be60328c11589be
CUT-akhir-10%-	mp4	49b94b5c2a69393d	cd13ad18b28b62a2	2f0425d83d66ef1fe

*file*_example_MP4_480_1_5MG.mp4		786a0b31b94e41c1	784b432716d1e15a27c6a7f9	d64aa95359ee60929c8e48d4b65dd65decb080e5ded7207
CUT-akhir-10%-*file*_example_MP4_640_3MG.mp4	mp4	4eb36045f73d5b0826710f713d8e53bc	d92cea1c711c6148708144b0d841eaa5724067e8	fac6a94ce84c69a069566f95e9f3c9942b5a3a93cf701c9233f9bb25705940b6
CUT-akhir-10%-*file*_example_ODS_10.ods	ods	75b0919879656b37c7b0dfc34aa11998	dbf96589af25eb3b68b3cb3879cb2d2dfd33e2a5	a9485cc909a48129fcdc319ea2bcaaf1e7d6a40752ab2077ada09395940ab66f
CUT-akhir-10%-*file*_example_ODS_100.ods	ods	1d034a032570710b83ef4526aec6a99c	dd499884c51abf2889ae1f9641127ac045d3516a	c4489ae371770ac558f5afb89d215df8ecde426238b86e03b3b9d54d86152c93
CUT-akhir-10%-*file*_example_ODS_1000.ods	ods	d24c2758725a44fcadfca2db629e7be7	fe3a743eb0b4ec19e058288511e59783fec312ec	664c71bf48817bc5c5e6e7b0ed3a6b0fac03025d94e248352b95d54d5ecc6d01
CUT-akhir-10%-*file*_example_PNG_1MB.png	png	5223189ce7099a4b63ded1f0fa329b36	47c4f4c06ed42eeedbca64a1cb034b76c48fe5c2	ebe001a130d40021bc9925a651b8e4c70b635497be265cc3b50a4fd82f90a9a0
CUT-akhir-10%-*file*_example_PNG_2100kB.png	png	f8249dc8002165bec8444f724dc49637	e2031290539d76bca22c8d1ae19660d5d8df5c8b	4f77891c868c95eedd4e04129b068402fb581b91a368c92cfeffd104154acebd
CUT-akhir-10%-*file*_example_PNG_3MB.png	png	4ddf145d2ad20f5ee6f30ae9ee3e4f6f	6121ae55198684a7153acd1424d9a54abe7e4827	abc76388a00afa41cc652f40e9fe6e3aefd33adeb5852adde8ca6d45a2b495ae
CUT-akhir-10%-*file*_example_PNG_500kB.png	png	07525b42340bf02f8b305043d891afc2	71847c62452901afdb0452fd70acd87f2115db42	94ee2722ffc68a8beb24800c21bebeb835d0deec7041f1bc9ab2625e25546adb
CUT-akhir-10%-*file*_example_PPT_1MB.ppt	ppt	072cfcec017fffbf20bc72c7f3c6201b	64a238fafde85ba73e9cfd0b5482568a34cd0f7d	d5596f6e910ee28f73086cb251d0562b221471935673cda5b4a0c9bbfa88816b

CUT-akhir-10%-*file*_example_PPT_250kB.ppt	ppt	fca4772db553cb0fb4465160608c6e2b	3ee1863f2402e779ca6e625d883ef8ce4594f78b	8e7f4ec3c383263083f4bdb47d59988273102606c3b16c506568b69b23b4bf7e
CUT-akhir-10%-*file*_example_PPT_500kB.ppt	ppt	16c3a6c954434d73fc8fff74e8a9e0e1	9a35c8784fb29be1d8a3ae7ce25cc3185e5eb79d	01daa8de49b31a7ed7729e78712df895eb8f978c1736c6fe36319e03d008e7bc
CUT-akhir-10%-*file*_example_XLSX_10.xlsx	xlsx	5cb84bf46e45cc307287ee5e254d14a6	37738e3eb073d3f1d57c91729d24a4bda95780cb	20768a5281f9d8b9b1991c2d171d458ab44e5f9f3dfc17facd24ce5107916959
CUT-akhir-10%-*file*_example_XLSX_100.xlsx	xlsx	20338fc687ea73cb11a69a7746440ea8	66a343db6e99beb0c2a6a4016d6e6c61a32b0dfc	ff66c05bc082b226aa6b65d1c43396b258bc83195f5369c7bc93f2bebbc99b86
CUT-akhir-10%-*file*_example_XLSX_1000.xlsx	xlsx	a6d0f6a533e1fa784ebcc532b9c78c16	009545fecb3b63d17e1f7f311c084a51454755e2	983fac87d0b6769b2a6b975c51654829e4c58925bccd9582c09e9afee8910cd3
CUT-akhir-10%-*file*_example_XLSX_50.xlsx	xlsx	d9cdd94406d65713996fde6ac142623e	19b2b7c94ae295dbc789ecd1bf36066fc076b87b	b05fffca52c7b189ef3f293223f8709d78a145f034741234112556ad14de6f67
CUT-akhir-10%-*file*_example_XLSX_5000.xlsx	xlsx	fd7ef151faaacbd399119c96ab02d517	d0fa136154aa08d94810b5d1f7bfaff2b011f073	893207e8a426784b6ad630cfe2587daa85a181fa2f0e0bb6aff158341b8f9773
CUT-akhir-10%-*file*_example_XLS_10.xls	xls	77024d23a2e1e6e2fc10884641e42158	fa4b81fe56698d1d906dac4ee5bf87d15781c5df	cb6c4bff0f4bb61953fe116aef5455b1d38e9048a251fc605fa480ce29e1058d
CUT-akhir-10%-*file*_example_XLS_100.xls	xls	2b250c4e3ebf918a4becc22539d6808a	5b3476d46671a369038ff53f2cf9d17b8ec1cbc1	2c9666de5df3594023cb4ed2e0259c4df85697099630c89c152a5cc33c0bd65e
CUT-akhir-10%-*file*_example_XLS_1000.xls	xls	9d9ae45b4fe6cdfe40e52109cd0177af	73c1b4cc49224847e0b038fc9aa67657a3d89fef	973aa971d16c32c520418f9b89c4231890fcfb7b01fff9f23b301f4e35a3a031

CUT-akhir-10%-*file*_example_XLS_50.xls	xls	f284cd10e3644a094faf279771c568f6	ce05e6aa0a725899a84e4f3852677ea7527e8b28	5f6c3d6a08ca6982808a10dc5d78b2b7c2dc273f0ad0229ebaa7614aee85bf15
CUT-akhir-10%-*file*_example_XLS_5000.xls	xls	7f095f9e8407d6b0de96ab708e750f94	a090247ef5d64c57f8b92651f20b755804bea801	0ffb97d0119a6dc57e03e1e278c7b7cecec2e64a8f17807bd3ce0b2c76bed766
CUT-awal-10%-Logo-UII-Asli.png	png	4f5bf5096eeafea231a930e6de134fe9	32d8619a0bcbe44e447eaccf5ec75393283f5a3d	c82330789785839696caa41a20e03f97dc80c07d4aeb2e130217106a8ce8129c
CUT-awal-10%-*file*-example_PDF_1MB.pdf	pdf	10f5b75e3f807437e626cb2f728f6039	b2f09de9a42ed3bf689b2d43dc88768fd05ba1d7	062234ae743bfcaf3ff384c9bfcf264f1f2604deefbc2ea19ec013e37ec3791a
CUT-awal-10%-*file*-example_PDF_500_kB.pdf	pdf	e00eba559be06c34d4d0667e700b8959	11c891c81e3d3efe56a1783ee0d037da747261b4	42168b3d238cf4e04fcd5e27a6ef712a94a1f0a609da3bfd35c0f1eb9bffa781
CUT-awal-10%-*file*-sample_100kB.doc	doc	a05d8cbad77d01834847ea1f81523b45	eb40a55e132b9546ce581fcabc4184586de9b587	f9baf626b9319ce907004d992b520bc4f2a44ee1829b238ab757ed5527b8701e
CUT-awal-10%-*file*-sample_100kB.docx	docx	883f4c632ce4fd8ee7b1f216716c9d15	8ca73560440c64ca71c4f81b90f63344685e3e01	2116253477789f1ad7e43c8019d4b1eee92084492e3fb8167f28d85d502c89be
CUT-awal-10%-*file*-sample_150kB.pdf	pdf	6dec5e4789e7345912748f772ddb3949	abe82f271be5097c8712ba0aa5c6797e8bb48bca	2f839be763c28817507519236fdfa27faa1b25df7d28df9a13250ed317bbb79f
CUT-awal-10%-*file*-sample_1MB.doc	doc	c49b942c47fac43ccb07abfbe3095a29	e070ea0b017b565bfa3c331213812282b47a8a62	6055830cc64f55cbe690694d43135c930730bd8efe79db08729c7b3c73cff3c7
CUT-awal-10%-*file*-	docx	23dd3daa1f19d692b293766808d9808	74ec8b5c437016ca70918fc5a045bfa0	9c3b251a31e181204cadc798ca8ed05e

sample_1MB.docx		0	20ee308b	7aa79b47ae7b941fcb724826cd74b4a5
CUT-awal-10%-*file*-sample_500kB.doc	doc	1edb721e8c388f1d6e5fb7b7f4b2eb0f	eb823a692d267d378fe3a7cdf03e06897752a6a6	726e0038de1c70c959c4cc60a18067442ad788913b73340255dd3c1bb2260dba
CUT-awal-10%-*file*-sample_500kB.docx	docx	15543f651ba97e115990b91c0a456442	592116639b0218b810609e2f3613af6822112532	258a6705d963f7b8ed7727125e0e7bb0720e80073583d32cd814b1cd5d8cdf55
CUT-awal-10%-*file*_example_GIF_1MB.gif	gif	a27bcec46bf551dcf63a787970af0eac	a8fdc0d0d822d51a16548c84a37a85f73a3a7ea7	44dc2888db57238fc2a4af2e68747ac3b2582ded0a0ceeb4e2f2b3fce4a9c521
CUT-awal-10%-*file*_example_GIF_3500kB.gif	gif	f264c31371f371f73b5ab3e01b969617	87eb76b10815fc8a479c451c2300abda6463c0cd	c6bc86bcda97853afeadf2301c9c27aa1a2521b70824fb3b8fe887f297e7ae16
CUT-awal-10%-*file*_example_GIF_500kB.gif	gif	3e4515ca5d0ed001f0a11998b3bd15dc	fac4c2cd05d9c059ef0aff10a36a22b16e7c8913	71736b8e8fd26c7b161acd3e6875a874036a4b845944eb0ed5cf3da9299d488e
CUT-awal-10%-*file*_example_JPG_100kB.jpg	jpg	335dbf1e8c050d1ac50a9fd42c1ecd8e	af64c8f97d1c84204e653e521acb8836e55fa91b	59137b39978ad05371f26700965d1bbbfe6142f6fedb81fb6c29c187b5beca36
CUT-awal-10%-*file*_example_JPG_1MB.jpg	jpg	4b45cce3c6db3bb46461141cd4622270	9b8b0d9c1cae0e16c807e668750ce8e758799ac3	a2510db22b0f9a71e61542bfaca37358bca054ef408ae64042567e53a73d7fd6
CUT-awal-10%-*file*_example_JPG_2500kB.jpg	jpg	409c6c4f77f16633390480cb3ae28ba1	073404452196f4f64b918bb945669105cc9121cb	45e3459235de76734f18ef213fb32612caad1f5640c6654bd859d469508cbbba
CUT-awal-10%-*file*_example_JPG_500kB.jpg	jpg	c289e1b9ffcc230e32620063ec29196b	fb44c26eec78889808804354284116d8b956ae30	bdad749f59c58db08fa5819de6f44fef2819ad4a553d7ed30583c930e38b84a2
CUT-awal-10%-	mp3	19ae80696204ea9b	7a27de80b348baa2	4616fa56cfbf33cd3

*file*_example_MP3_1MG.mp3		5d68cf61efe86ffb	b4a3c8d4a9d2b1871518c0aa	a3eb1b330c5f6887623c6aed8c2f5ad49d77955ab23c1ab
CUT-awal-10%-*file*_example_MP3_2MG.mp3	mp3	f8f2405c22b8f0d9b9a2a876d91c7252	64c6010f8070c563844067408734efdb3bf3605e	14656d1d32a54c518974036d16ac5e854836984f8520e539b9c442b3f8c09e1d
CUT-awal-10%-*file*_example_MP3_5MG.mp3	mp3	dd80d2db4f712a8e01fcb6470adf6fa7	98f17cbb70ba48b509e41d6dbd75950375557d82	7a7c8af1749cb13b9157a4ca499f9fa5c44588b252392785eb78dcb8d19b7333
CUT-awal-10%-*file*_example_MP3_700KB.mp3	mp3	ae5d8e868504f28efeef2496e53113a3	64a52c5e5b693768b8ccdde162504582982b41f5	cf1b2f29a08ac5900a3658791f84585101d78784c5fb2444a89f345b6998c19d
CUT-awal-10%-*file*_example_MP4_1280_10MG.mp4	mp4	4860b32f3813420ff4eec101edacf0e2	39e79dac608fbe359d94f4749f2396eb62493917	6d8f7481311c72ea4259f2c819124550108f761c97714833486633f514eebd62
CUT-awal-10%-*file*_example_MP4_1920_18MG.mp4	mp4	68024f71e13d4ffb01d378d208efa0a0	1534cf77c507d628a6e1a4e2d921798b2fd572cb	9639f9718b048794907bf1f0076ae41531a250a665cb27a73254b3c4e714abd7
CUT-awal-10%-*file*_example_MP4_480_1_5MG.mp4	mp4	b6b47583dcf2f48b63dedb1bcca265ed	5cc7ee005a568cf1e00a83cbabe6b65f8bfb26c3	65b0688bb19699ee3a915f4a04ea424e9ec68b9a175e796cf457287bd739c41a
CUT-awal-10%-*file*_example_MP4_640_3MG.mp4	mp4	2327082c2666fa6c7c0c7e1a03c6e308	0e8f38bbec77ad3608ffad354c9094179d2c6495	b8474201757397c833a5538154b1ad50411b023a320bcfec11a7acc2650f7d9b
CUT-awal-10%-*file*_example_ODS_10.ods	ods	27adf0353d77586db67fcb7c6f3f72de	5aadd3509e0e5263c0597df0130c03b657a6c94d	c6346436277aa5c7bb7c6518e7b9b1c71d397ccc923db57edad449291b5defc9
CUT-awal-10%-*file*_example_ODS_100.ods	ods	8cc9487fa1b2bcdded1f531283e6907f	8d69fa312b19b3aa5fb2d80bdfaaf2b24e7f4143	e2ca513b3e56ef0791c7e3a8a9605220f9cc5b574267f71dcd48628939dd67b

				9
CUT-awal-10%-*file*_example_ODS_1000.ods	ods	008f469f44a602585cc901eb5230d668	e347c4536aa76a047060de724f94f24edb102c30	3d29b5111db81edceb2d3ccb10920d020271542a5ca37cc3cb83a378bedddab4
CUT-awal-10%-*file*_example_PNG_1MB.png	png	56c052668c5dbbdf2516a5a25c8090f7	211e0a43dd78094de0ce2fa810212b0f0d4f5f2a	0e36d9de01a7d0d5a0b760d9eeff099816b47441762e299b2f6773d8031b861f
CUT-awal-10%-*file*_example_PNG_2100kB.png	png	69eb4fd4af26dc43b57616bcb39fe7e3	739e1bbd349bac0a012c4abc7731ae2d57b6e5be	a02cad95e88407a5fbc51d84b7b060bebb62b09110f5b8707718a037b16009a2
CUT-awal-10%-*file*_example_PNG_3MB.png	png	ac91fd8cad9449635c4623286ad7402e	e9f772b64d50b4274516fabdeef37e3e6d2f1450	2ccdf0c40dc78ab6720351af233e69e876315ba66d4350ebeb3aa82dc23c1c52
CUT-awal-10%-*file*_example_PNG_500kB.png	png	307ad9c92611d08d8d6d72613507d818	674a44945674288b132f84f9bbf8925ea2dd4309	c93da85a3be70a21063c90deb5c02c8734a6739cde2c4c351044d8c70786b016
CUT-awal-10%-*file*_example_PPT_1MB.ppt	ppt	772bb4613d54d5c69022651d7f9a4b09	760a78445068d085021c68fe8c3500b67ee3c15f	752519b385145a86758485d497b3c13fc694af135466c21389ee77ef5c514567
CUT-awal-10%-*file*_example_PPT_250kB.ppt	ppt	9f449aeaa09f8c763f47ff64be966232	46cb6ad1dfaea12e84dbf7301f10e245dc6659bd	15399e563b0303db721f1fe3994f6cf597680e6c9e575e1c3f2960d29d31eeb4
CUT-awal-10%-*file*_example_PPT_500kB.ppt	ppt	689d27032d65a80d4ede39db8b0817ef	72699adb6c2b11a292d52ec913a1f8e19fbc1241	e2da21e5659512fe5d777dab59c6979b74a02ad4d331c292a9bd8244607d4747
CUT-awal-10%-*file*_example_XLSX_10.xlsx	xlsx	dc26e5f61bea0574be6fe143e175c573	341923910a64b763e64f756954acbbfced502c22	970a96a448fef51be5560e6145782b967cacfcecbc50b87d

				62963a5c6c9a3f39
CUT-awal-10%-*file*_example_XLSX_100.xlsx	xlsx	f8a1f4d52bfcd6e2ddf2f762e01da282	4930f5833f3a3efcc1fde53151efe5a5fc9831df	3151cacc50b9cf66663ef814d62e859400cae91387596d06d31009d3062da736
CUT-awal-10%-*file*_example_XLSX_1000.xlsx	xlsx	09a50537eb15d48130c4bf6fb798873a	2c81c7f0a8acbda19c521a67bc5f1f75c4c00dc8	1d1475b40153b9338dce56f412040ad146bcf3d2cdb8cc466eb3b57cd1394e0f
CUT-awal-10%-*file*_example_XLSX_50.xlsx	xlsx	b4671c5ee8761d398e2532542e60e40f	210231e64027db0adbeb18b6990e3925fcce8c48	a0f3d8dd787a90d5d8a4001c398ef5b1aac9e6ddcd16a48e9404bb7b6ad92844
CUT-awal-10%-*file*_example_XLSX_5000.xlsx	xlsx	cbc9231624954d8bb4c8a67d1573f321	1bd5decc813c89e3d20daed4dad9174fd63eae93	6a4d6ee67b85e93177d5b43646e719c28c3b0f6bfc7143960231d898d80823fa
CUT-awal-10%-*file*_example_XLS_10.xls	xls	0f9d9728c31cdb5407504b609557acca	d4b0cbe9a31aebb4182fd98143344d270c2af7c5	dba2fe3a410ccf3954303442e55e12afe33d3d18cad2d4763f6c8980630b8a85
CUT-awal-10%-*file*_example_XLS_100.xls	xls	48003e97094680159a14092d9e58f7e0	1b37e20f9d66e77e31a1ee51568cfdaba20d2607	bde354168f4168c315ccc0a5ab2b92c6ed53959d40539a3dd20ed3afa7ee454a
CUT-awal-10%-*file*_example_XLS_1000.xls	xls	26156b28fee81a7968015100820ee59b	23b97fa1c482e5ffd611ad3c6cadf99f202acf87	27f23ef7edb66339bc3e4dc480c78745b8ad558ab4f8cbaea9fc4b8175eaf6df
CUT-awal-10%-*file*_example_XLS_50.xls	xls	6bab8fe045deca76f24999c3bef5c62a	9cd8a3bda100dca904ecd96033e4c59e99fd4eb7	1a08ae8b6856b089c13670bf79f351885f9ce84ef73a4f7666b1bdf04cfc71a5
CUT-awal-10%-*file*_example_XLS_5000.xls	xls	c323bb8a8b280f10a8277a8336bc8fc5	9b54e8db964f5bdf878e9c6184d512c460e6f89b	d5a19af53e9897a8ff100d3888d6cd1e2bdb413910fdf7f8b3febf2ba3a0c416

CUT-keduanya-10%-Logo-UII-Asli.png	png	31775bcd0fa0a6abb0c7b732f84c078c	53a0866a45c14ee94b564eeb790041a30f63a0b2	18216ba795c1ea978914381a1450872b328c0fb7232d9738f723bebd305a35a9
CUT-keduanya-10%-*file*-example_PDF_1MB.pdf	pdf	0c4b89cfb5b891b4275144ac09728c77	ae1599788d0190b2f9af124b03f32c1a1ae9954f	dd921de91700f7c2440be3abd7f991f5c5b29ca6568383fb5a0dd4650d709c3f
CUT-keduanya-10%-*file*-example_PDF_500_kB.pdf	pdf	494f85160d32cffda47a1d3d5c0866fa	5475a31007d80dfd2b329f63bdc4238b56f0d468	d742eb5b3313f821280be9656285f6c3bb4b79ad421655a46630dcb8f641602c
CUT-keduanya-10%-*file*-sample_100kB.doc	doc	1ccb829d101b49e4e102176011ca2374	71ee3a0b26b0956a6aafb37c9f69b65b6d3f9fc5	eaa04e5f34ea1549de04103c7d3133878c34a3793a9bd9f4659b55db7520f629
CUT-keduanya-10%-*file*-sample_100kB.docx	docx	f899b56682533b0b1f4cb61ecf980bef	1ea25cd22ed8051596aaa8c98bfdc840aec93c56	4a729de00ee7336bcc2daf2176e8c77bba3f590c70cd25d6cfc52bedbe50e7a5
CUT-keduanya-10%-*file*-sample_150kB.pdf	pdf	cb895800795bba9efe4f652c443f1fca	f096f2c24da8a679d98ba6dcb02dfb92cd552f58	fc775ce9c8c37f5904b5f65366a5208ea9212f7ca20e879b78c38c579ebbb1e8
CUT-keduanya-10%-*file*-sample_1MB.doc	doc	380ad7693ac1bba9252747cb2eb10494	6ae8635be89c98a03384496ceb054d1bb0ec1f57	e68a2339a4270007a0b776c912f8471b66015e04aee9260d0ef4a392be4e797e
CUT-keduanya-10%-*file*-sample_1MB.docx	docx	e3efd3da29c7d4cbd1c8d17ea1efab2b	f2b34be5c8c381dbe95d0e49a04e8f84400cba9f	7de6217abff7c49c90cf54676cf9cc16a83354571e0d4628ba0a041bb2e5d4fc
CUT-keduanya-10%-*file*-sample_500kB.doc	doc	4c3f88de664c59ba1eb08be8b474ac36	7eb2879acea8bf762d0e2469af956c3ded7c93b0	a3b02b5160626a736c1573965b48d64769f099d9882cc08df595df6c82dace1a
CUT-keduanya-10%-*file*-sample_500kB.doc	docx	819cdf45ec2760d444ea75a6e17dee5b	7fe646199579ba705b1ba634da8b6d5e65ee2f10	f5910b43204583ea039bc16b301d3abe1807bc38c77d7ea1

x				a18d1fd0fd293779
CUT-keduanya-10%-*file*_example_GIF_1MB.gif	gif	0215608860c04cd500a1dc04f271d20c	f76a252df0d96ec18590e171744ef6c714293471	1fd4d704c7fea45178180916aef430e3e5a02f70692c0c351189e0021df8507c
CUT-keduanya-10%-*file*_example_GIF_3500kB.gif	gif	96d20281204b77451e8f80b270432b40	0dce5d524448cc51babd8fea3c1eda745a46f554	e89dc4e97809c89e0b092e5fc4e6df6be7f9e786cfad6802d07622b1bb7f1a6e
CUT-keduanya-10%-*file*_example_GIF_500kB.gif	gif	42c6938e92b53948c384f762ee3a8d76	3cecd0feb663ac7d421ef798ce5924002b5f7408	784d3e443e5b930ba9b03c721a28b654ef44477ffc61ace13dfe502d89ab9512
CUT-keduanya-10%-*file*_example_JPG_100kB.jpg	jpg	8702db631801fdb1b44cbcad8e668f95	ca1859b922731714e82d0c8def3ecdd0832b1a65	c5486d95a7e148c62d298b2e99055f04d321247866d6aadbf984ecf8240e6f54
CUT-keduanya-10%-*file*_example_JPG_1MB.jpg	jpg	f4d5da5e04ea908d0c02dcdbc79f661d	e86d17adadda6eb542019aedacec4fec20d077ef	63bd2d0763d7ceaba0c8dafe92b88ef3dfe366abe8e8ea7dd80504291e235893
CUT-keduanya-10%-*file*_example_JPG_2500kB.jpg	jpg	f9ed8568756945ad93164c11356e610a	699ad6b67e562b2fb8fb72bbfe548d8b3269df17	07cff22f005e620a7adf6b9635be98c661ce03b0674033be208b41e2c300114b
CUT-keduanya-10%-*file*_example_JPG_500kB.jpg	jpg	227c057e68944e6355fd55aed009b784	9d722d9671caaee78a201071444ff752f878478b	3be00b24be48948e429030409b05503459f8abb7efc3cefcf9bb507f44f0db85
CUT-keduanya-10%-*file*_example_MP3_1MG.mp3	mp3	eab6873fc4ff0a4f12af10db284c7bf4	d195001b7808e7715f8846c638caac50452dd938	4723b1909e2379ca7102a9f531c7dde106a7e0e2941b0baf6324c5e7e3c347c3
CUT-keduanya-10%-*file*_example_MP3_2MG.mp3	mp3	5c6805e358c4ebb01ffcf7dfbdf94e92	3e8b00dd70dfb3df753e82834ac9cdd516609a62	94733c460ffdce38ee6c2c21125a3728e0cb142ebe7f09f0623cfe6d054cb293
CUT-keduanya-10%-*file*_example_MP3	mp3	6ecc3310982ec79ef9ef22a96ec1a79e	078dc4e3f87d5d98a786455ea5c1af01c5f92055	e1b267fd9f4942daee756fed6b13372fb27b8c3f87f92d00

_5MG.mp3				d44a8d9ab7174fab
CUT-keduanya-10%-*file*_example_MP3_700KB.mp3	mp3	ac91cb009cf5e7c48026d7d9991f2e54	75e5bec5ac2005d5717f7ad6288c857cac449bd8	a629b40d6298caf0561e73c0b1204b3a9a2d6c7a0f7af5f388722282facd7ff8
CUT-keduanya-10%-*file*_example_MP4_1280_10MG.mp4	mp4	5c1d716c7e6b041537aa78c2632f4125	097b84644968348b6fb48d32170b2024833e0f2c	ff4006ae6a2a276b6c960693420419687d53ee2a1371abc0c2f41dbc56f25c21
CUT-keduanya-10%-*file*_example_MP4_1920_18MG.mp4	mp4	964bf7af9c9c23145f4e6cdf8ebfaf55	1d5024bf1a413bbf48e16f2a48a654740a7169ef	11a6033def6020f155fcae8f53e4c492e9f4be96eb18635d2be60328c11589be
CUT-keduanya-10%-*file*_example_MP4_480_1_5MG.mp4	mp4	49b94b5c2a69393d786a0b31b94e41c1	cd13ad18b28b62a2784b432716d1e15a27c6a7f9	2f0425d83d66ef1fed64aa95359ee60929c8e48d4b65dd65decb080e5ded7207
CUT-keduanya-10%-*file*_example_MP4_640_3MG.mp4	mp4	4eb36045f73d5b0826710f713d8e53bc	d92cea1c711c6148708144b0d841eaa5724067e8	fac6a94ce84c69a069566f95e9f3c9942b5a3a93cf701c9233f9bb25705940b6
CUT-keduanya-10%-*file*_example_ODS_10.ods	ods	75b0919879656b37c7b0dfc34aa11998	dbf96589af25eb3b68b3cb3879cb2d2dfd33e2a5	a9485cc909a48129fcdc319ea2bcaaf1e7d6a40752ab2077ada09395940ab66f
CUT-keduanya-10%-*file*_example_ODS_100.ods	ods	1d034a032570710b83ef4526aec6a99c	dd499884c51abf2889ae1f9641127ac045d3516a	c4489ae371770ac558f5afb89d215df8ecde426238b86e03b3b9d54d86152c93
CUT-keduanya-10%-*file*_example_ODS_1000.ods	ods	d24c2758725a44fcadfca2db629e7be7	fe3a743eb0b4ec19e058288511e59783fec312ec	664c71bf48817bc5c5e6e7b0ed3a6b0fac03025d94e248352b95d54d5ecc6d01
CUT-keduanya-10%-*file*_example_PNG_1MB.png	png	5223189ce7099a4b63ded1f0fa329b36	47c4f4c06ed42eeedbca64a1cb034b76c48fe5c2	ebe001a130d40021bc9925a651b8e4c70b635497be265cc3b50a4fd82f90a9a0
CUT-keduanya-10%-	png	f8249dc8002165bec8444f724dc49637	e2031290539d76bca22c8d1ae19660d5	4f77891c868c95eedd4e04129b06840

*file*_example_PNG_2100kB.png			d8df5c8b	2fb581b91a368c92cfeffd104154acebd
CUT-keduanya-10%-*file*_example_PNG_3MB.png	png	4ddf145d2ad20f5ee6f30ae9ee3e4f6f	6121ae55198684a7153acd1424d9a54abe7e4827	abc76388a00afa41cc652f40e9fe6e3aefd33adeb5852adde8ca6d45a2b495ae
CUT-keduanya-10%-*file*_example_PNG_500kB.png	png	07525b42340bf02f8b305043d891afc2	71847c62452901afdb0452fd70acd87f2115db42	94ee2722ffc68a8beb24800c21bebeb835d0deec7041f1bc9ab2625e25546adb
CUT-keduanya-10%-*file*_example_PPT_1MB.ppt	ppt	072cfcec017fffbf20bc72c7f3c6201b	64a238fafde85ba73e9cfd0b5482568a34cd0f7d	d5596f6e910ee28f73086cb251d0562b221471935673cda5b4a0c9bbfa88816b
CUT-keduanya-10%-*file*_example_PPT_250kB.ppt	ppt	fca4772db553cb0fb4465160608c6e2b	3ee1863f2402e779ca6e625d883ef8ce4594f78b	8e7f4ec3c383263083f4bdb47d59988273102606c3b16c506568b69b23b4bf7e
CUT-keduanya-10%-*file*_example_PPT_500kB.ppt	ppt	16c3a6c954434d73fc8fff74e8a9e0e1	9a35c8784fb29be1d8a3ae7ce25cc3185e5eb79d	01daa8de49b31a7ed7729e78712df895eb8f978c1736c6fe36319e03d008e7bc
CUT-keduanya-10%-*file*_example_XLSX_10.xlsx	xlsx	5cb84bf46e45cc307287ee5e254d14a6	37738e3eb073d3f1d57c91729d24a4bda95780cb	20768a5281f9d8b9b1991c2d171d458ab44e5f9f3dfc17facd24ce5107916959
CUT-keduanya-10%-*file*_example_XLSX_100.xlsx	xlsx	20338fc687ea73cb11a69a7746440ea8	66a343db6e99beb0c2a6a4016d6e6c61a32b0dfc	ff66c05bc082b226aa6b65d1c43396b258bc83195f5369c7bc93f2bebbc99b86
CUT-keduanya-10%-*file*_example_XLSX_1000.xlsx	xlsx	a6d0f6a533e1fa784ebcc532b9c78c16	009545fecb3b63d17e1f7f311c084a51454755e2	983fac87d0b6769b2a6b975c51654829e4c58925bccd9582c09e9afee8910cd3
CUT-keduanya-10%-*file*_example_XLSX_50.xlsx	xlsx	d9cdd94406d65713996fde6ac142623e	19b2b7c94ae295dbc789ecd1bf36066fc076b87b	b05fffca52c7b189ef3f293223f8709d78a145f034741234112556ad14de6f67
CUT-keduanya-	xlsx	fd7ef151faaacbd39	d0fa136154aa08d9	893207e8a426784b

10%-*file*_example_XLSX_5000.xlsx		9119c96ab02d517	4810b5d1f7bfaff2b011f073	6ad630cfe2587daa85a181fa2f0e0bb6aff158341b8f9773
CUT-keduanya-10%-*file*_example_XLS_10.xls	xls	77024d23a2e1e6e2fc10884641e42158	fa4b81fe56698d1d906dac4ee5bf87d15781c5df	cb6c4bff0f4bb61953fe116aef5455b1d38e9048a251fc605fa480ce29e1058d
CUT-keduanya-10%-*file*_example_XLS_100.xls	xls	2b250c4e3ebf918a4becc22539d6808a	5b3476d46671a369038ff53f2cf9d17b8ec1cbc1	2c9666de5df3594023cb4ed2e0259c4df85697099630c89c152a5cc33c0bd65e
CUT-keduanya-10%-*file*_example_XLS_1000.xls	xls	9d9ae45b4fe6cdfe40e52109cd0177af	73c1b4cc49224847e0b038fc9aa67657a3d89fef	973aa971d16c32c520418f9b89c4231890fcfb7b01fff9f23b301f4e35a3a031
CUT-keduanya-10%-*file*_example_XLS_50.xls	xls	f284cd10e3644a094faf279771c568f6	ce05e6aa0a725899a84e4f3852677ea7527e8b28	5f6c3d6a08ca6982808a10dc5d78b2b7c2dc273f0ad0229ebaa7614aee85bf15
CUT-keduanya-10%-*file*_example_XLS_5000.xls	xls	7f095f9e8407d6b0de96ab708e750f94	a090247ef5d64c57f8b92651f20b755804bea801	0ffb97d0119a6dc57e03e1e278c7b7cecec2e64a8f17807bd3ce0b2c76bed766

Table 4 Hasil Kalkulasi Machine Learning

Nama *File*	Metode	Hasil
CUT-akhir-10%-Logo-UII-Asli.png	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-Logo-UII-Asli.png	*Support Vector Machine*	pngw
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	*K-Nearest Neighbor*	pdf
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	*Support Vector Machine*	pdf
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	ods
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*-sample_100kB.doc	*K-Nearest Neighbor*	docx
CUT-akhir-10%-*file*-sample_100kB.doc	*Support Vector Machine*	pptx

CUT-akhir-10%-*file*-sample_100kB.docx	*K-Nearest Neighbor*	epub
CUT-akhir-10%-*file*-sample_100kB.docx	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*-sample_150kB.pdf	*K-Nearest Neighbor*	pdf
CUT-akhir-10%-*file*-sample_150kB.pdf	*Support Vector Machine*	pdf
CUT-akhir-10%-*file*-sample_1MB.doc	*K-Nearest Neighbor*	mp3
CUT-akhir-10%-*file*-sample_1MB.doc	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*-sample_1MB.docx	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-*file*-sample_1MB.docx	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*-sample_500kB.doc	*K-Nearest Neighbor*	pptx
CUT-akhir-10%-*file*-sample_500kB.doc	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*-sample_500kB.docx	*K-Nearest Neighbor*	exe
CUT-akhir-10%-*file*-sample_500kB.docx	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*_example_GIF_1MB.gif	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-*file*_example_GIF_1MB.gif	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*_example_GIF_3500kB.gif	*K-Nearest Neighbor*	pdf
CUT-akhir-10%-*file*_example_GIF_3500kB.gif	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*_example_GIF_500kB.gif	*K-Nearest Neighbor*	gif
CUT-akhir-10%-*file*_example_GIF_500kB.gif	*Support Vector Machine*	bmpw
CUT-akhir-10%-*file*_example_JPG_100kB.jpg	*K-Nearest Neighbor*	gifw
CUT-akhir-10%-*file*_example_JPG_100kB.jpg	*Support Vector Machine*	bmpw
CUT-akhir-10%-*file*_example_JPG_1MB.jpg	*K-Nearest Neighbor*	ods
CUT-akhir-10%-*file*_example_JPG_1MB.jpg	*Support Vector Machine*	bmpw
CUT-akhir-10%-*file*_example_JPG_2500kB.jpg	*K-Nearest Neighbor*	oxps
CUT-akhir-10%-*file*_example_JPG_2500kB.jpg	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*_example_JPG_500kB.jpg	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-*file*_example_JPG_500kB.jpg	*Support Vector Machine*	oxps

CUT-akhir-10%-*file*_example_MP3_1MG.mp3	*K-Nearest Neighbor*	mp3
CUT-akhir-10%-*file*_example_MP3_1MG.mp3	*Support Vector Machine*	apk
CUT-akhir-10%-*file*_example_MP3_2MG.mp3	*K-Nearest Neighbor*	mkv
CUT-akhir-10%-*file*_example_MP3_2MG.mp3	*Support Vector Machine*	apk
CUT-akhir-10%-*file*_example_MP3_5MG.mp3	*K-Nearest Neighbor*	exe
CUT-akhir-10%-*file*_example_MP3_5MG.mp3	*Support Vector Machine*	apk
CUT-akhir-10%-*file*_example_MP3_700KB.mp3	*K-Nearest Neighbor*	mp3
CUT-akhir-10%-*file*_example_MP3_700KB.mp3	*Support Vector Machine*	pptx
CUT-akhir-10%-*file*_example_MP4_1280_10MG.mp4	*K-Nearest Neighbor*	docx
CUT-akhir-10%-*file*_example_MP4_1280_10MG.mp4	*Support Vector Machine*	gifw
CUT-akhir-10%-*file*_example_MP4_1920_18MG.mp4	*K-Nearest Neighbor*	oxps
CUT-akhir-10%-*file*_example_MP4_1920_18MG.mp4	*Support Vector Machine*	ods
CUT-akhir-10%-*file*_example_MP4_480_1_5MG.mp4	*K-Nearest Neighbor*	exe
CUT-akhir-10%-*file*_example_MP4_480_1_5MG.mp4	*Support Vector Machine*	gifw
CUT-akhir-10%-*file*_example_MP4_640_3MG.mp4	*K-Nearest Neighbor*	exe
CUT-akhir-10%-*file*_example_MP4_640_3MG.mp4	*Support Vector Machine*	gifw
CUT-akhir-10%-*file*_example_ODS_10.ods	*K-Nearest Neighbor*	ods
CUT-akhir-10%-*file*_example_ODS_10.ods	*Support Vector Machine*	ods
CUT-akhir-10%-*file*_example_ODS_100.ods	*K-Nearest Neighbor*	ods
CUT-akhir-10%-*file*_example_ODS_100.ods	*Support Vector Machine*	ods
CUT-akhir-10%-*file*_example_ODS_1000.ods	*K-Nearest Neighbor*	ods
CUT-akhir-10%-*file*_example_ODS_1000.ods	*Support Vector Machine*	ods
CUT-akhir-10%-*file*_example_PNG_1MB.png	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-*file*_example_PNG_1MB.png	*Support Vector*	oxps

	Machine	
CUT-akhir-10%-*file*_example_PNG_2100kB.png	*K-Nearest Neighbor*	png
CUT-akhir-10%-*file*_example_PNG_2100kB.png	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*_example_PNG_3MB.png	*K-Nearest Neighbor*	png
CUT-akhir-10%-*file*_example_PNG_3MB.png	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*_example_PNG_500kB.png	*K-Nearest Neighbor*	epub
CUT-akhir-10%-*file*_example_PNG_500kB.png	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*_example_PPT_1MB.ppt	*K-Nearest Neighbor*	oxps
CUT-akhir-10%-*file*_example_PPT_1MB.ppt	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*_example_PPT_250kB.ppt	*K-Nearest Neighbor*	doc
CUT-akhir-10%-*file*_example_PPT_250kB.ppt	*Support Vector Machine*	pptx
CUT-akhir-10%-*file*_example_PPT_500kB.ppt	*K-Nearest Neighbor*	apk
CUT-akhir-10%-*file*_example_PPT_500kB.ppt	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*_example_XLSX_10.xlsx	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-*file*_example_XLSX_10.xlsx	*Support Vector Machine*	docx
CUT-akhir-10%-*file*_example_XLSX_100.xlsx	*K-Nearest Neighbor*	gifw
CUT-akhir-10%-*file*_example_XLSX_100.xlsx	*Support Vector Machine*	epub
CUT-akhir-10%-*file*_example_XLSX_1000.xlsx	*K-Nearest Neighbor*	gifw
CUT-akhir-10%-*file*_example_XLSX_1000.xlsx	*Support Vector Machine*	mp3
CUT-akhir-10%-*file*_example_XLSX_50.xlsx	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-*file*_example_XLSX_50.xlsx	*Support Vector Machine*	mp4
CUT-akhir-10%-*file*_example_XLSX_5000.xlsx	*K-Nearest Neighbor*	gifw
CUT-akhir-10%-*file*_example_XLSX_5000.xlsx	*Support Vector Machine*	oxps
CUT-akhir-10%-*file*_example_XLS_10.xls	*K-Nearest Neighbor*	Does not meet 256 features: 186 found
CUT-akhir-10%-*file*_example_XLS_10.xls	*Support Vector Machine*	Does not meet 256 features: 186 found
CUT-akhir-10%-*file*_example_XLS_100.xls	*K-Nearest Neighbor*	Does not meet 256 features: 210

		found
CUT-akhir-10%-*file*_example_XLS_100.xls	*Support Vector Machine*	Does not meet 256 features: 210 found
CUT-akhir-10%-*file*_example_XLS_1000.xls	*K-Nearest Neighbor*	ppt
CUT-akhir-10%-*file*_example_XLS_1000.xls	*Support Vector Machine*	doc
CUT-akhir-10%-*file*_example_XLS_50.xls	*K-Nearest Neighbor*	Does not meet 256 features: 211 found
CUT-akhir-10%-*file*_example_XLS_50.xls	*Support Vector Machine*	Does not meet 256 features: 211 found
CUT-akhir-10%-*file*_example_XLS_5000.xls	*K-Nearest Neighbor*	exe
CUT-akhir-10%-*file*_example_XLS_5000.xls	*Support Vector Machine*	ppt
CUT-awal-10%-Logo-UII-Asli.png	*K-Nearest Neighbor*	pngw
CUT-awal-10%-Logo-UII-Asli.png	*Support Vector Machine*	pngw
CUT-awal-10%-*file*-example_PDF_1MB.pdf	*K-Nearest Neighbor*	pdf
CUT-awal-10%-*file*-example_PDF_1MB.pdf	*Support Vector Machine*	pdf
CUT-awal-10%-*file*-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	pdf
CUT-awal-10%-*file*-example_PDF_500_kB.pdf	*Support Vector Machine*	pdf
CUT-awal-10%-*file*-sample_100kB.doc	*K-Nearest Neighbor*	docx
CUT-awal-10%-*file*-sample_100kB.doc	*Support Vector Machine*	docx
CUT-awal-10%-*file*-sample_100kB.docx	*K-Nearest Neighbor*	gifw
CUT-awal-10%-*file*-sample_100kB.docx	*Support Vector Machine*	docx
CUT-awal-10%-*file*-sample_150kB.pdf	*K-Nearest Neighbor*	pdf
CUT-awal-10%-*file*-sample_150kB.pdf	*Support Vector Machine*	pdf
CUT-awal-10%-*file*-sample_1MB.doc	*K-Nearest Neighbor*	docx
CUT-awal-10%-*file*-sample_1MB.doc	*Support Vector Machine*	bmp
CUT-awal-10%-*file*-sample_1MB.docx	*K-Nearest Neighbor*	epub
CUT-awal-10%-*file*-sample_1MB.docx	*Support Vector Machine*	oxps
CUT-awal-10%-*file*-sample_500kB.doc	*K-Nearest Neighbor*	docx
CUT-awal-10%-*file*-sample_500kB.doc	*Support Vector*	bmp

	Machine	
CUT-awal-10%-*file*-sample_500kB.docx	*K-Nearest Neighbor*	epub
CUT-awal-10%-*file*-sample_500kB.docx	*Support Vector Machine*	docx
CUT-awal-10%-*file*_example_GIF_1MB.gif	*K-Nearest Neighbor*	gifw
CUT-awal-10%-*file*_example_GIF_1MB.gif	*Support Vector Machine*	oxps
CUT-awal-10%-*file*_example_GIF_3500kB.gif	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_GIF_3500kB.gif	*Support Vector Machine*	oxps
CUT-awal-10%-*file*_example_GIF_500kB.gif	*K-Nearest Neighbor*	gifw
CUT-awal-10%-*file*_example_GIF_500kB.gif	*Support Vector Machine*	bmp
CUT-awal-10%-*file*_example_JPG_100kB.jpg	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_JPG_100kB.jpg	*Support Vector Machine*	pngw
CUT-awal-10%-*file*_example_JPG_1MB.jpg	*K-Nearest Neighbor*	pdf
CUT-awal-10%-*file*_example_JPG_1MB.jpg	*Support Vector Machine*	jpg
CUT-awal-10%-*file*_example_JPG_2500kB.jpg	*K-Nearest Neighbor*	ods
CUT-awal-10%-*file*_example_JPG_2500kB.jpg	*Support Vector Machine*	bmp
CUT-awal-10%-*file*_example_JPG_500kB.jpg	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_JPG_500kB.jpg	*Support Vector Machine*	jpg
CUT-awal-10%-*file*_example_MP3_1MG.mp3	*K-Nearest Neighbor*	gifw
CUT-awal-10%-*file*_example_MP3_1MG.mp3	*Support Vector Machine*	pptx
CUT-awal-10%-*file*_example_MP3_2MG.mp3	*K-Nearest Neighbor*	gifw
CUT-awal-10%-*file*_example_MP3_2MG.mp3	*Support Vector Machine*	exe
CUT-awal-10%-*file*_example_MP3_5MG.mp3	*K-Nearest Neighbor*	mp3
CUT-awal-10%-*file*_example_MP3_5MG.mp3	*Support Vector Machine*	mp3
CUT-awal-10%-*file*_example_MP3_700KB.mp3	*K-Nearest Neighbor*	ods
CUT-awal-10%-*file*_example_MP3_700KB.mp3	*Support Vector Machine*	mp3
CUT-awal-10%-*file*_example_MP4_1280_10MG.mp4	*K-Nearest Neighbor*	jpg

CUT-awal-10%-*file*_example_MP4_1280_10MG.mp4	*Support Vector Machine*	gifw
CUT-awal-10%-*file*_example_MP4_1920_18MG.mp4	*K-Nearest Neighbor*	png
CUT-awal-10%-*file*_example_MP4_1920_18MG.mp4	*Support Vector Machine*	gifw
CUT-awal-10%-*file*_example_MP4_480_1_5MG.mp4	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_MP4_480_1_5MG.mp4	*Support Vector Machine*	gifw
CUT-awal-10%-*file*_example_MP4_640_3MG.mp4	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_MP4_640_3MG.mp4	*Support Vector Machine*	gifw
CUT-awal-10%-*file*_example_ODS_10.ods	*K-Nearest Neighbor*	Does not meet 256 features: 253 found
CUT-awal-10%-*file*_example_ODS_10.ods	*Support Vector Machine*	Does not meet 256 features: 253 found
CUT-awal-10%-*file*_example_ODS_100.ods	*K-Nearest Neighbor*	ods
CUT-awal-10%-*file*_example_ODS_100.ods	*Support Vector Machine*	ods
CUT-awal-10%-*file*_example_ODS_1000.ods	*K-Nearest Neighbor*	ods
CUT-awal-10%-*file*_example_ODS_1000.ods	*Support Vector Machine*	ods
CUT-awal-10%-*file*_example_PNG_1MB.png	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_PNG_1MB.png	*Support Vector Machine*	pngw
CUT-awal-10%-*file*_example_PNG_2100kB.png	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_PNG_2100kB.png	*Support Vector Machine*	oxps
CUT-awal-10%-*file*_example_PNG_3MB.png	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_PNG_3MB.png	*Support Vector Machine*	bmp
CUT-awal-10%-*file*_example_PNG_500kB.png	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_PNG_500kB.png	*Support Vector Machine*	bmpw
CUT-awal-10%-*file*_example_PPT_1MB.ppt	*K-Nearest Neighbor*	doc
CUT-awal-10%-*file*_example_PPT_1MB.ppt	*Support Vector Machine*	pngw

CUT-awal-10%-*file*_example_PPT_250kB.ppt	*K-Nearest Neighbor*	Does not meet 256 features: 235 found
CUT-awal-10%-*file*_example_PPT_250kB.ppt	*Support Vector Machine*	Does not meet 256 features: 235 found
CUT-awal-10%-*file*_example_PPT_500kB.ppt	*K-Nearest Neighbor*	Does not meet 256 features: 238 found
CUT-awal-10%-*file*_example_PPT_500kB.ppt	*Support Vector Machine*	Does not meet 256 features: 238 found
CUT-awal-10%-*file*_example_XLSX_10.xlsx	*K-Nearest Neighbor*	Does not meet 256 features: 97 found
CUT-awal-10%-*file*_example_XLSX_10.xlsx	*Support Vector Machine*	Does not meet 256 features: 97 found
CUT-awal-10%-*file*_example_XLSX_100.xlsx	*K-Nearest Neighbor*	Does not meet 256 features: 211 found
CUT-awal-10%-*file*_example_XLSX_100.xlsx	*Support Vector Machine*	Does not meet 256 features: 211 found
CUT-awal-10%-*file*_example_XLSX_1000.xlsx	*K-Nearest Neighbor*	pngw
CUT-awal-10%-*file*_example_XLSX_1000.xlsx	*Support Vector Machine*	mp4
CUT-awal-10%-*file*_example_XLSX_50.xlsx	*K-Nearest Neighbor*	Does not meet 256 features: 158 found
CUT-awal-10%-*file*_example_XLSX_50.xlsx	*Support Vector Machine*	Does not meet 256 features: 158 found
CUT-awal-10%-*file*_example_XLSX_5000.xlsx	*K-Nearest Neighbor*	gifw
CUT-awal-10%-*file*_example_XLSX_5000.xlsx	*Support Vector Machine*	jpg
CUT-awal-10%-*file*_example_XLS_10.xls	*K-Nearest Neighbor*	Does not meet 256 features: 235 found
CUT-awal-10%-*file*_example_XLS_1000.xls	*Support Vector Machine*	Does not meet 256 features: 235 found
CUT-awal-10%-*file*_example_XLS_50.xls	*K-Nearest Neighbor*	
CUT-keduanya-10%-*file*-example_PDF_1MB.pdf	*K-Nearest Neighbor*	pdf
CUT-keduanya-10%-*file*-example_PDF_1MB.pdf	*Support Vector Machine*	pdf
CUT-keduanya-10%-*file*-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	ods
CUT-keduanya-10%-*file*-example_PDF_500_kB.pdf	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*-sample_100kB.doc	*K-Nearest Neighbor*	docx

CUT-keduanya-10%-*file*-sample_100kB.doc	*Support Vector Machine*	pptx
CUT-keduanya-10%-*file*-sample_100kB.docx	*K-Nearest Neighbor*	epub
CUT-keduanya-10%-*file*-sample_100kB.docx	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*-sample_150kB.pdf	*K-Nearest Neighbor*	pdf
CUT-keduanya-10%-*file*-sample_150kB.pdf	*Support Vector Machine*	pdf
CUT-keduanya-10%-*file*-sample_1MB.doc	*K-Nearest Neighbor*	mp3
CUT-keduanya-10%-*file*-sample_1MB.doc	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*-sample_1MB.docx	*K-Nearest Neighbor*	pngw
CUT-keduanya-10%-*file*-sample_1MB.docx	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*-sample_500kB.doc	*K-Nearest Neighbor*	pptx
CUT-keduanya-10%-*file*-sample_500kB.doc	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*-sample_500kB.docx	*K-Nearest Neighbor*	exe
CUT-keduanya-10%-*file*-sample_500kB.docx	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_GIF_1MB.gif	*K-Nearest Neighbor*	pngw
CUT-keduanya-10%-*file*_example_GIF_1MB.gif	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*_example_GIF_3500kB.gif	*K-Nearest Neighbor*	pdf
CUT-keduanya-10%-*file*_example_GIF_3500kB.gif	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_GIF_500kB.gif	*K-Nearest Neighbor*	gif
CUT-keduanya-10%-*file*_example_GIF_500kB.gif	*Support Vector Machine*	bmpw
CUT-keduanya-10%-*file*_example_JPG_100kB.jpg	*K-Nearest Neighbor*	gifw
CUT-keduanya-10%-*file*_example_JPG_100kB.jpg	*Support Vector Machine*	bmpw
CUT-keduanya-10%-*file*_example_JPG_1MB.jpg	*K-Nearest Neighbor*	ods
CUT-keduanya-10%-*file*_example_JPG_1MB.jpg	*Support Vector Machine*	bmpw
CUT-keduanya-10%-	*K-Nearest Neighbor*	oxps

*file*_example_JPG_2500kB.jpg		
CUT-keduanya-10%-*file*_example_JPG_2500kB.jpg	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*_example_JPG_500kB.jpg	*K-Nearest Neighbor*	pngw
CUT-keduanya-10%-*file*_example_JPG_500kB.jpg	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_MP3_1MG.mp3	*K-Nearest Neighbor*	mp3
CUT-keduanya-10%-*file*_example_MP3_1MG.mp3	*Support Vector Machine*	apk
CUT-keduanya-10%-*file*_example_MP3_2MG.mp3	*K-Nearest Neighbor*	mkv
CUT-keduanya-10%-*file*_example_MP3_2MG.mp3	*Support Vector Machine*	apk
CUT-keduanya-10%-*file*_example_MP3_5MG.mp3	*K-Nearest Neighbor*	exe
CUT-keduanya-10%-*file*_example_MP3_5MG.mp3	*Support Vector Machine*	apk
CUT-keduanya-10%-*file*_example_MP3_700KB.mp3	*K-Nearest Neighbor*	mp3
CUT-keduanya-10%-*file*_example_MP3_700KB.mp3	*Support Vector Machine*	pptx
CUT-keduanya-10%-*file*_example_MP4_1280_10MG.mp4	*K-Nearest Neighbor*	docx
CUT-keduanya-10%-*file*_example_MP4_1280_10MG.mp4	*Support Vector Machine*	gifw
CUT-keduanya-10%-*file*_example_MP4_1920_18MG.mp4	*K-Nearest Neighbor*	oxps
CUT-keduanya-10%-*file*_example_MP4_1920_18MG.mp4	*Support Vector Machine*	ods
CUT-keduanya-10%-*file*_example_MP4_480_1_5MG.mp4	*K-Nearest Neighbor*	exe
CUT-keduanya-10%-*file*_example_MP4_480_1_5MG.mp4	*Support Vector Machine*	gifw
CUT-keduanya-10%-*file*_example_MP4_640_3MG.mp4	*K-Nearest Neighbor*	exe
CUT-keduanya-10%-*file*_example_MP4_640_3MG.mp4	*Support Vector Machine*	gifw

CUT-keduanya-10%-*file*_example_ODS_10.ods	*K-Nearest Neighbor*	ods
CUT-keduanya-10%-*file*_example_ODS_10.ods	*Support Vector Machine*	ods
CUT-keduanya-10%-*file*_example_ODS_100.ods	*K-Nearest Neighbor*	ods
CUT-keduanya-10%-*file*_example_ODS_100.ods	*Support Vector Machine*	ods
CUT-keduanya-10%-*file*_example_ODS_1000.ods	*K-Nearest Neighbor*	ods
CUT-keduanya-10%-*file*_example_ODS_1000.ods	*Support Vector Machine*	ods
CUT-keduanya-10%-*file*_example_PNG_1MB.png	*K-Nearest Neighbor*	pngw
CUT-keduanya-10%-*file*_example_PNG_1MB.png	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_PNG_2100kB.png	*K-Nearest Neighbor*	png
CUT-keduanya-10%-*file*_example_PNG_2100kB.png	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_PNG_3MB.png	*K-Nearest Neighbor*	png
CUT-keduanya-10%-*file*_example_PNG_3MB.png	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_PNG_500kB.png	*K-Nearest Neighbor*	epub
CUT-keduanya-10%-*file*_example_PNG_500kB.png	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_PPT_1MB.ppt	*K-Nearest Neighbor*	oxps
CUT-keduanya-10%-*file*_example_PPT_1MB.ppt	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*_example_PPT_250kB.ppt	*K-Nearest Neighbor*	doc
CUT-keduanya-10%-*file*_example_PPT_250kB.ppt	*Support Vector Machine*	pptx
CUT-keduanya-10%-*file*_example_PPT_500kB.ppt	*K-Nearest Neighbor*	apk
CUT-keduanya-10%-*file*_example_PPT_500kB.ppt	*Support Vector Machine*	bmp
CUT-keduanya-10%-*file*_example_XLSX_10.xlsx	*K-Nearest Neighbor*	pngw

CUT-keduanya-10%-*file*_example_XLSX_10.xlsx	*Support Vector Machine*	docx
CUT-keduanya-10%-*file*_example_XLSX_100.xlsx	*K-Nearest Neighbor*	gifw
CUT-keduanya-10%-*file*_example_XLSX_100.xlsx	*Support Vector Machine*	epub
CUT-keduanya-10%-*file*_example_XLSX_1000.xlsx	*K-Nearest Neighbor*	gifw
CUT-keduanya-10%-*file*_example_XLSX_1000.xlsx	*Support Vector Machine*	mp3
CUT-keduanya-10%-*file*_example_XLSX_50.xlsx	*K-Nearest Neighbor*	pngw
CUT-keduanya-10%-*file*_example_XLSX_50.xlsx	*Support Vector Machine*	mp4
CUT-keduanya-10%-*file*_example_XLSX_5000.xlsx	*K-Nearest Neighbor*	gifw
CUT-keduanya-10%-*file*_example_XLSX_5000.xlsx	*Support Vector Machine*	oxps
CUT-keduanya-10%-*file*_example_XLS_10.xls	*K-Nearest Neighbor*	Does not meet 256 features: 186 found
CUT-keduanya-10%-*file*_example_XLS_10.xls	*Support Vector Machine*	Does not meet 256 features: 186 found
CUT-keduanya-10%-*file*_example_XLS_100.xls	*K-Nearest Neighbor*	Does not meet 256 features: 210 found
CUT-keduanya-10%-*file*_example_XLS_100.xls	*Support Vector Machine*	Does not meet 256 features: 210 found
CUT-keduanya-10%-*file*_example_XLS_1000.xls	*K-Nearest Neighbor*	ppt
CUT-keduanya-10%-*file*_example_XLS_1000.xls	*Support Vector Machine*	doc
CUT-keduanya-10%-*file*_example_XLS_50.xls	*K-Nearest Neighbor*	Does not meet 256 features: 211 found
CUT-keduanya-10%-*file*_example_XLS_50.xls	*Support Vector Machine*	Does not meet 256 features: 211 found
CUT-keduanya-10%-*file*_example_XLS_5000.xls	*K-Nearest Neighbor*	exe
CUT-keduanya-10%-*file*_example_XLS_5000.xls	*Support Vector Machine*	ppt
CUT-akhir-10%-Logo-UII-Asli.png	*K-Nearest Neighbor*	pngw
CUT-akhir-10%-Logo-UII-Asli.png	*Support Vector Machine*	pngw

CUT-akhir-10%-*file*-example_PDF_1MB.pdf	*K-Nearest Neighbor*	pdf
CUT-akhir-10%-*file*-example_PDF_1MB.pdf	*Support Vector Machine*	pdf
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	*K-Nearest Neighbor*	ods
CUT-akhir-10%-*file*-example_PDF_500_kB.pdf	*Support Vector Machine*	bmp
CUT-akhir-10%-*file*-sample_100kB.doc	*K-Nearest Neighbor*	docx
CUT-akhir-10%-*file*-sample_100kB.doc	*Support Vector Machine*	pptx
CUT-akhir-10%-*file*-sample_100kB.docx	*K-Nearest Neighbor*	epub

Table 5 Hasil K-Nearest Neighbor

Type	**Precision**	**Recall**	**F1-score**	**Support**
apk	0.84	0.89	0.87	1491
bin	0.80	0.89	0.84	1442
bmp	0.99	0.99	0.99	1396
bmpw	0.95	0.85	0.90	839
doc	0.88	0.93	0.90	1376
docx	0.84	0.88	0.86	1495
dwg	1.00	1.00	1.00	1537
elf	0.84	0.76	0.80	1319
epub	0.93	0.96	0.95	1481
exe	0.88	0.86	0.87	1439
gif	0.97	0.99	0.98	1487
gifw	0.92	0.93	0.92	1308
jpg	1.00	1.00	1.00	1510
mkv	0.39	0.39	0.39	1431
mp3	0.98	1.00	0.99	1507
mp4	0.41	0.41	0.41	1499
ods	0.92	0.87	0.90	1481
oxps	0.81	0.86	0.83	1530
pdf	0.95	0.91	0.93	1544
png	0.98	0.98	0.98	1505
pngw	0.94	0.79	0.86	1461

ppt	0.74	0.70	0.72	643
pptx	0.73	0.72	0.73	1275
Accuracy			0.86	31996
Macro avg	0.86	0.85	0.85	31996
Weighted avg	0.86	0.86	0.86	31996

Table 6 Hasil Support Vector Machine

Type	**Precision**	**Recall**	**F1-score**	**Support**
apk	0.90	0.92	0.91	1491
bin	0.71	0.95	0.81	1442
bmp	0.72	0.70	0.71	1396
bmpw	0.57	0.32	0.41	839
doc	0.93	0.95	0.94	1376
docx	0.89	0.79	0.84	1495
dwg	1.00	1.00	1.00	1537
elf	0.85	0.56	0.68	1319
epub	0.83	0.96	0.89	1481
exe	0.84	0.83	0.84	1439
gif	0.95	1.00	0.97	1487
gifw	0.95	0.69	0.80	1308
jpg	0.98	0.99	0.98	1510
mkv	0.97	0.99	0.98	1431
mp3	0.98	0.99	0.99	1507
mp4	0.95	0.99	0.97	1499
ods	0.85	0.87	0.86	1481
oxps	0.80	0.95	0.87	1530
pdf	0.90	0.97	0.94	1544
png	0.81	0.91	0.86	1505
pngw	0.74	0.37	0.49	1461
ppt	0.70	0.87	0.78	643
pptx	0.74	0.92	0.82	1275

Accuracy			0.86	31996
Macro avg	0.85	0.85	0.84	31996
Weighted avg	0.86	0.86	0.85	31996