**DETEKSI DAN KLASIFIKASI SERANGAN *MALWARE* PADA DATASET *BENIGN* DAN *MALICIOUS* MENGGUNAKAN METODE *MACHINE LEARNING***

**Tegar Arsyadani1**

1tegar2000018243@webmail.uad.ac.id, Universitas Ahmad Dahlan

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **cAbstrak:** |  | Penelitian ini bertujuan untuk mengembangkan metode deteksi dan klasifikasi serangan malware menggunakan algoritma *Support Vector Machines* (SVM) pada dataset *benign* dan *malicious*. Metode SVM digunakan untuk memisahkan sampel-sampel *benign* dan *malicious* dalam ruang fitur yang kompleks. Data pra-pemrosesan, pembagian data *training* dan *testing*, serta pelatihan model SVM dilakukan. Evaluasi model melibatkan akurasi dan laporan klasifikasi. Hasil evaluasi menunjukkan akurasi sekitar 76.52%, dengan model mampu mengklasifikasikan sampel-sampel *Malware* dengan baik, tetapi mengalami kesulitan dalam mengidentifikasi sampel-sampel *Benign*. Visualisasi hasil klasifikasi menggunakan *scatter plot* membantu memvisualisasikan pola distribusi model SVM. Meskipun model SVM memiliki potensi dalam melindungi sistem komputer dari ancaman serangan malware, peningkatan lebih lanjut diperlukan untuk mengatasi kesulitan dalam mengklasifikasikan sampel-sampel *Benign*. |
| ***Key Words*:** deteksi *malware*, klasifikasi *malware*, *Support Vector Machines*, evaluasi model, *scatter plot*. | | |

## PENDAHULUAN

Perkembangan teknologi dan informasi memiliki pengaruh yang luar biasa terhadap berbagai aspek kehidupan manusia. Berbagai kegiatan manusia tidak dapat dipisahkan dari yang namanya teknologi informasi, dari mulai bekerja sampai dengan hiburan (Hasibuan, 2018). Perkembangan ini tentu saja memberikan manfaat yang luar biasa dalam memberikan kemudahan dan kepraktisan dalam membantu berbagai aktivitas manusia.

Perkembangan teknologi ini tidak dapat dilepaskan dari perkembangan di bidang komputerisasi, bahkan sampai saat ini perkembangan dalam dunia komputer masih terus terjadi (Setiawan, 2018). Komputer pada awalnya hanya digunakan dalam bidang administrasi saja, kini telah merambah ke berbagai bidang seperti pendidikan (Sawitri dkk., 2019), industri, bahkan dalam mengembangkan teknologinya itu sendiri. Oleh karena itu komputer menjadi perangkat yang sangat penting.

Namun sejalan kemudahan yang ditawarkan juga membawa ancaman yang dapat merugikan. Ancaman-ancaman tersebut seperti kejahatan dunia maya, penyadapan, virus pada komputer dan serangan pada jaringan lainnya (Hasibuan, 2018). Oleh karena itu, keamanan sistem komputer merupakan faktor penting yang harus dipastikan. Salah satu ancaman yang paling banyak terjadi adalah serangan *malware.* Serangan *malware* dapat menyebabkan kerugian finansial, kehilangan data, atau gangguan operasional yang serius (Hartono, 2023). Lebih parahnya, serangan *malware* mencuri data sensitif, kerusakan sistem, atau bahkan kehilangan data secara permanen (Mahendra dkk., 2022). Oleh karena itu dalam menghindari kerugian tersebut, *user* komputer harus memiliki langkah-langkah pencegahan dari serangan *malware*.

Serangan *malware* telah menjadi ancaman serius dalam keamanan komputer dan informasi di era digital saat ini. Jenis *malware* dapat dibedakan menjadi *benign* (tidak berbahaya) dan *malicious* (berbahaya) berdasarkan tujuan dan efeknya (Huang dkk., 2013). *Malware benign* biasanya tidak merusak atau mencuri data, tetapi mungkin memiliki efek negatif seperti menghasilkan iklan yang mengganggu atau memperlambat sistem. Contoh *malware* *benign* termasuk *adware* dan *spyware*. Di sisi lain, *malware malicious* adalah yang berbahaya dan merugikan. Jenis *malware malicious* meliputi *virus, worm, trojan, ransomware, horse, keyloggers* dan *botnet*. Tujuan dari malware malicious umumnya adalah merusak, mencuri data, atau mengambil kendali atas sistem yang terinfeksi. *Malware malicious* dapat menyebabkan kerugian finansial, kehilangan data yang signifikan, atau bahkan mengganggu kegiatan bisnis secara keseluruhan.

Deteksi dan klasifikasi serangan *malware* yang efektif menjadi krusial dalam melindungi sistem komputer dari ancaman tersebut. Metode *Support Vector Machines* (SVM) telah terbukti menjadi salah satu pendekatan yang kuat dan efisien dalam deteksi dan klasifikasi serangan *malware* (Munawar & Putri, 2020). Maka pada penelitian ini akan berfokus untuk mengembangkan metode deteksi dan klasifikasi serangan malware menggunakan metode SVM. SVM adalah sebuah algoritma *machine learning* yang mampu membangun model yang dapat memisahkan sampel-sampel *benign* (tidak berbahaya) dan *malicious* (berbahaya) dalam ruang fitur yang kompleks (Tjahjadi & Santoso, 2023).

Rumusan masalah dalam penelitian ini adalah bagaimana mengembangkan model SVM yang dapat mendeteksi dan mengklasifikasikan serangan malware dengan tingkat keakuratan yang tinggi. Selain itu, penelitian ini juga bertujuan untuk mengidentifikasi fitur-fitur yang paling relevan dalam membedakan antara sampel *benign* dan *malicious* menggunakan SVM.

Harapannya Dengan menggunakan metode SVM dapat memberikan kontribusi dalam pengembangan teknik yang lebih baik dalam membedakan sampel *benign* dan *maliciou*s. Hal ini akan membantu organisasi dan individu dalam melindungi sistem komputer mereka dari ancaman serangan *malware* yang dapat menyebabkan kerugian finansial, kehilangan data, atau gangguan operasional. Melalui pendekatan SVM, penelitian ini diharapkan dapat memberikan solusi yang handal dan efektif dalam deteksi dan klasifikasi serangan *malware*. Dengan mengintegrasikan literatur yang relevan dan terkait dengan metode SVM dalam deteksi dan klasifikasi serangan malware, penelitian ini memperkuat dasar pengetahuan yang ada dan memberikan kontribusi dalam pengembangan bidang keamanan komputer.

## METODE

Penelitian ini menggunakan metode *Support Vector Machines* (SVM) dalam mendeteksi dan klasifikasi serangan malware. Berikut adalah penjelasan mengenai metode penelitian yang digunakan:

1. Desain Penelitian: Penelitian ini menggunakan pendekatan eksperimen dengan mengimplementasikan model SVM untuk deteksi dan klasifikasi serangan malware. Pendekatan ini memungkinkan evaluasi yang sistematis terhadap performa metode SVM dalam mengklasifikasikan sampel program sebagai benign atau malicious.
2. Data Sampel: Data sampel yang digunakan dalam penelitian ini terdiri dari dataset yang terdiri dari sampel program yang telah diklasifikasikan sebelumnya sebagai benign atau malicious. Dataset ini dapat diperoleh dari sumber-sumber terpercaya seperti Kaggle atau organisasi keamanan komputer. Penting untuk mencatat jumlah sampel dan komposisi kelas (benign dan malicious) dalam dataset yang digunakan.
3. Pra-pemrosesan Data: Sebelum menggunakan dataset dalam model SVM, pra-pemrosesan data dilakukan untuk memastikan kualitas dan kecocokan data. Ini mungkin melibatkan langkah-langkah seperti pemilihan fitur, normalisasi data, atau penanganan ketidakseimbangan kelas.
4. Pembagian Dataset: Dataset yang digunakan biasanya dibagi menjadi dua subset: data pelatihan (training data) dan data pengujian (testing data). Data pelatihan digunakan untuk melatih model SVM, sedangkan data pengujian digunakan untuk menguji performa model yang telah dilatih.
5. Implementasi SVM: Model SVM diimplementasikan menggunakan algoritma SVM yang tersedia dalam pustaka atau framework machine learning seperti scikit-learn. Penjelasan tentang parameter dan konfigurasi SVM yang digunakan juga perlu disebutkan, misalnya jenis kernel yang digunakan (seperti kernel linear atau kernel RBF), parameter C, dan gamma.
6. Evaluasi Performa: Performa model SVM dievaluasi menggunakan metrik-metrik evaluasi seperti akurasi, presisi, recall, atau F1-score. Metrik-metrik ini memberikan informasi tentang sejauh mana model mampu mengklasifikasikan dengan benar sampel-sampel benign dan malicious.
7. Validasi dan Analisis: Validasi dilakukan untuk memastikan kehandalan model SVM dalam deteksi dan klasifikasi serangan malware. Selain itu, analisis yang mendalam juga dilakukan terhadap hasil dan temuan dari model SVM yang dihasilkan.
8. Peralatan dan Bahan: Dalam penelitian ini, peralatan dan bahan yang digunakan meliputi bahasa pemrograman (seperti Python), pustaka atau framework machine learning (seperti scikit-learn), dan perangkat keras atau infrastruktur komputasi yang digunakan untuk menjalankan eksperimen.

## PEMBAHASAN

Hasil pengumpulan data ini penulis menggunakan data public <https://www.kaggle.com/datasets/amauricio/pe-files-malwares> kemudian dataset tersebut akan diproses dan di klasifikasi berdasarkan pemodelan yang telah diolah oleh peneliti. Pengumpulan data berasal dari perangkat/alat dari hasil penelitian orang lain dataset public, dataset public adalah data yang sudah ada yang digunakan para peneliti sebelumnya. Malware yang menyusup di browser. Dalam satu baris ada satu serangan yang terjadi, dan setiap serangan malware pasti akan meninggalkan jejak yang disebut loc.

Pada bagian ini, akan dibahas hasil dari implementasi metode Support Vector Machines (SVM) untuk deteksi dan klasifikasi serangan malware pada dataset benign dan malicious. Implementasi ini dilakukan dengan menggunakan bahasa pemrograman Python dan pustaka scikit-learn untuk membangun dan mengevaluasi model SVM.

Dataset yang digunakan pada contoh ini adalah dataset malware, tahapan-tahapan yang akan dilakukan adalah Prepocessing data, pembagian data latih dan data uji, pemodelan dengan *Support Vektor Machines*, evaluasi model, dan Visualisasi hasil akhir.

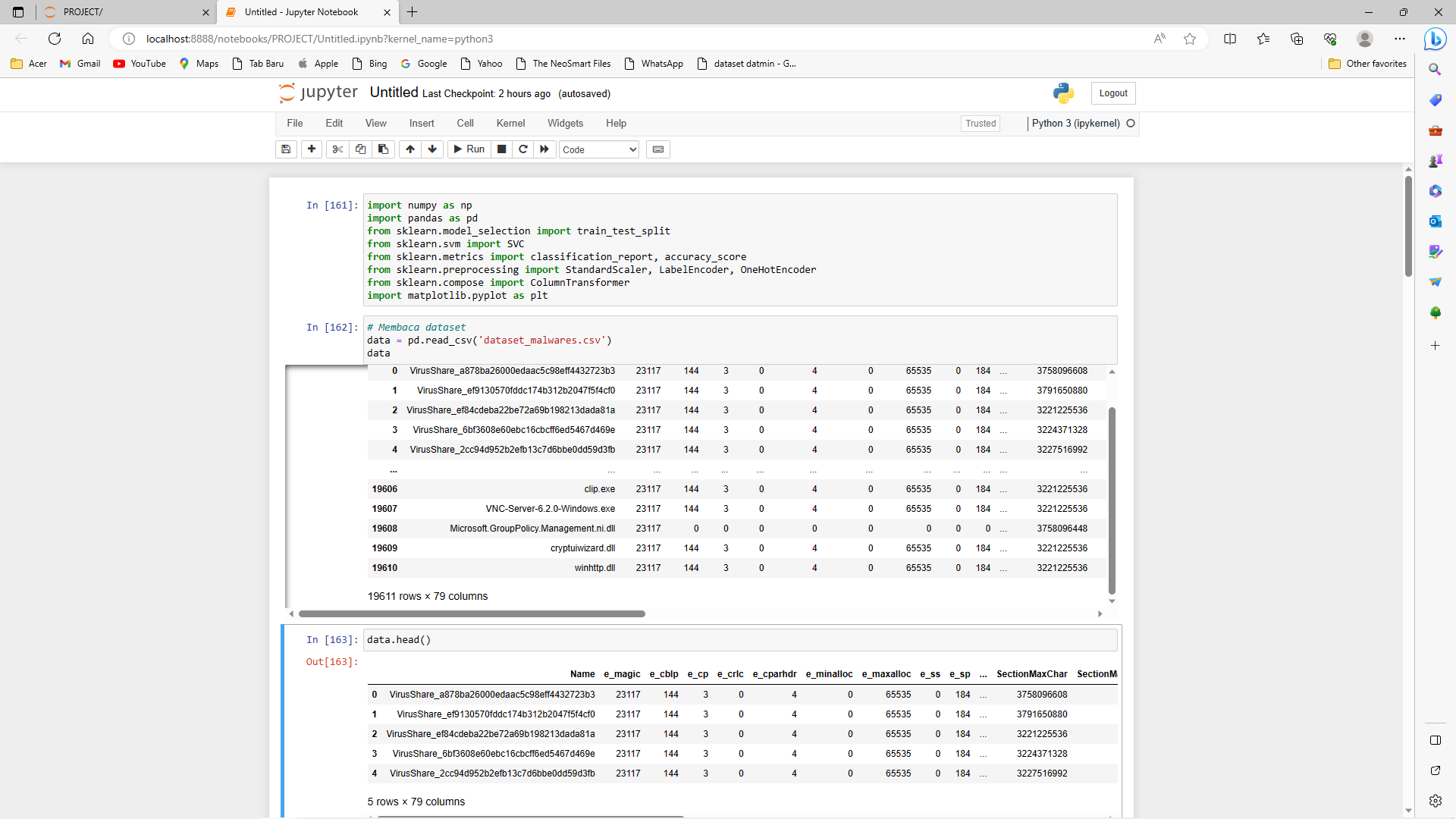
1. **Prepocessing Data**

Langkah pertama dalam penelitian ini adalah pra-pemrosesan data. Data yang digunakan diambil dari dataset eksternal yang berisi informasi tentang program-program yang telah diklasifikasikan sebagai benign atau malicious. Data ini berisi fitur-fitur penting seperti 'Machine' (mesin yang menjalankan program) dan 'TimeDateStamp' (timestamp saat program dijalankan). Proses pra-pemrosesan melibatkan pemilihan fitur-fitur yang relevan dan normalisasi data.

Head() digunakan untuk menampilkan data awal atau data teratas pada dataframe. Default-nya jika kita tidak memberikan argument di dalam tanda kurung (), data yang akan ditampilkan adalah 5 baris data teratas.

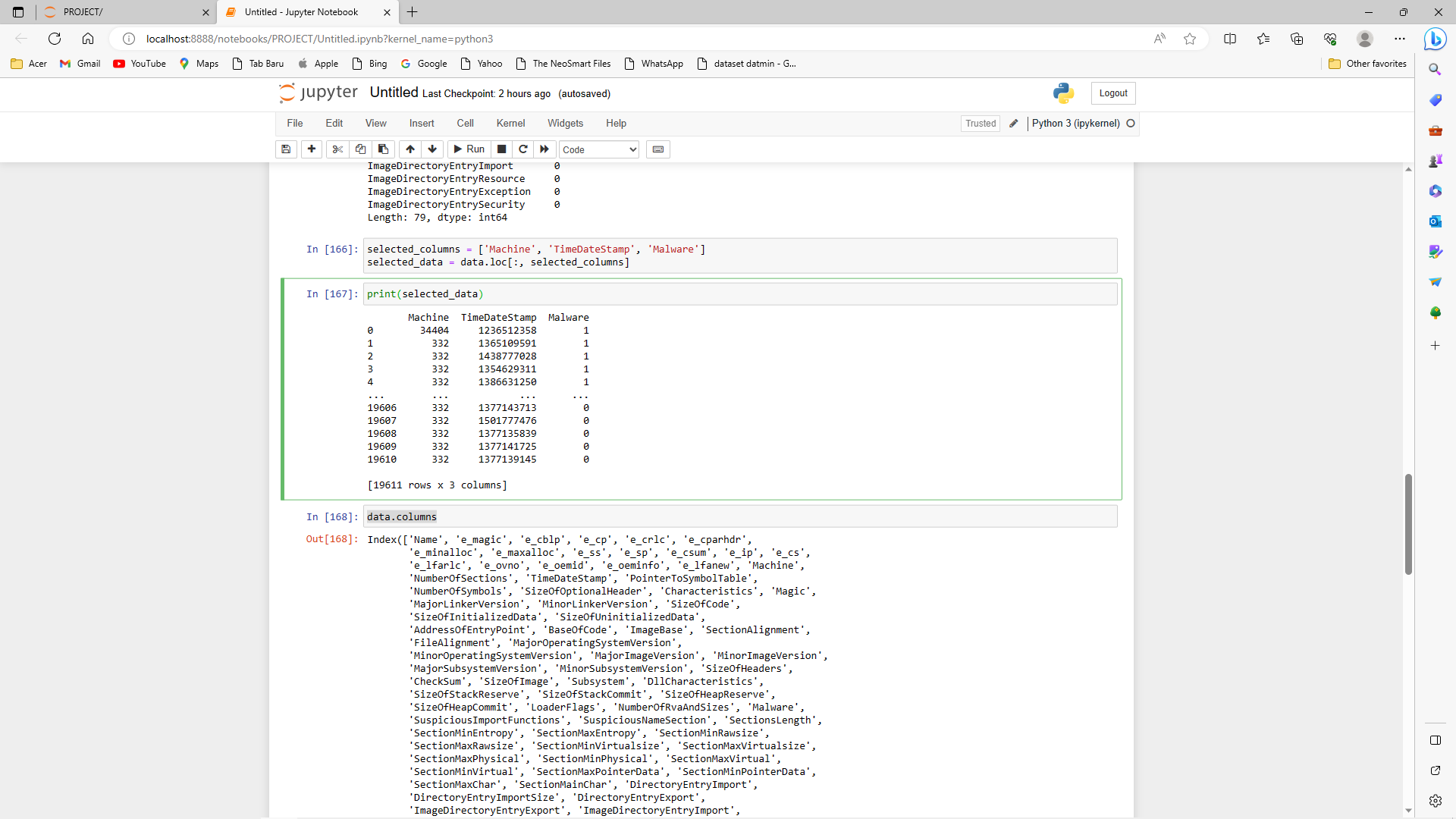
data = pd.read\_csv('dataset\_malwares.csv')

data.head()



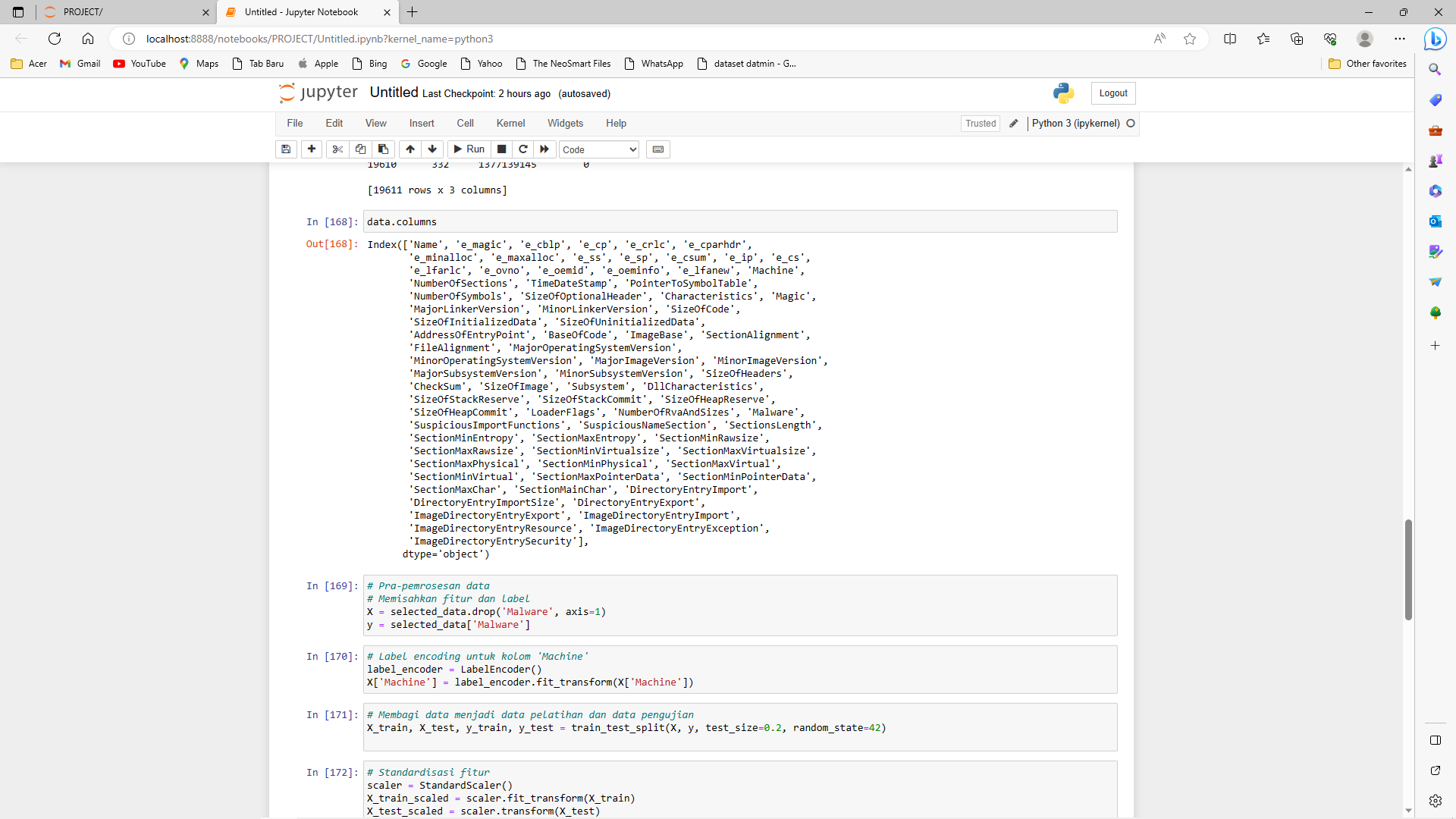
Gambar 1Hasil Eksplorasi Data head

Data.loc merupakan salah satu cara yang efektif untuk memilih baris dan kolom pada dataframe sesuai dengan nama index baris atau kolom. Seleksi kolom yang kita gunakan terdiri dari “Machine”,”TimeDateStamp”,”Malware”.



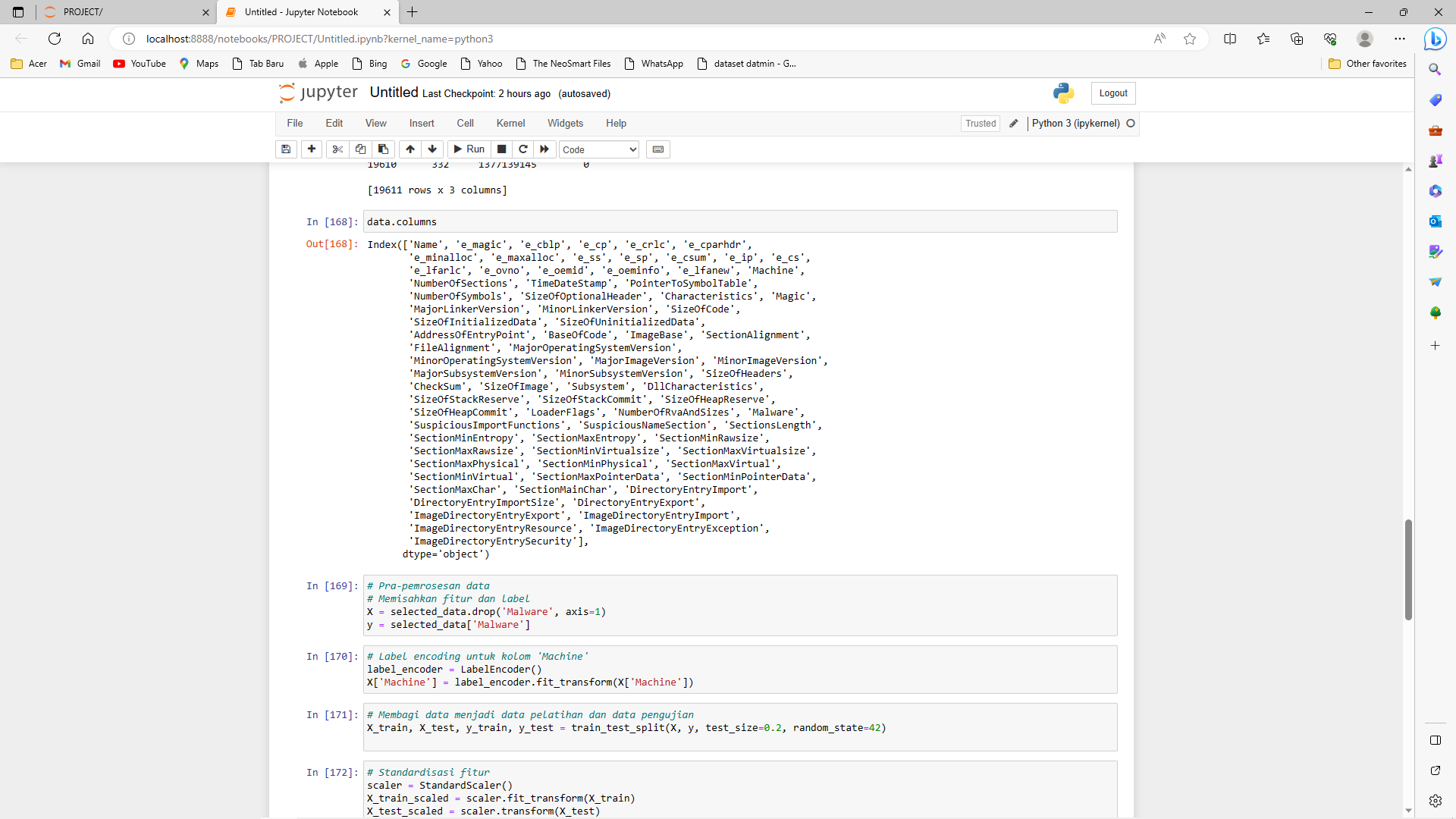
Gambar 2 Data loc

Data columns



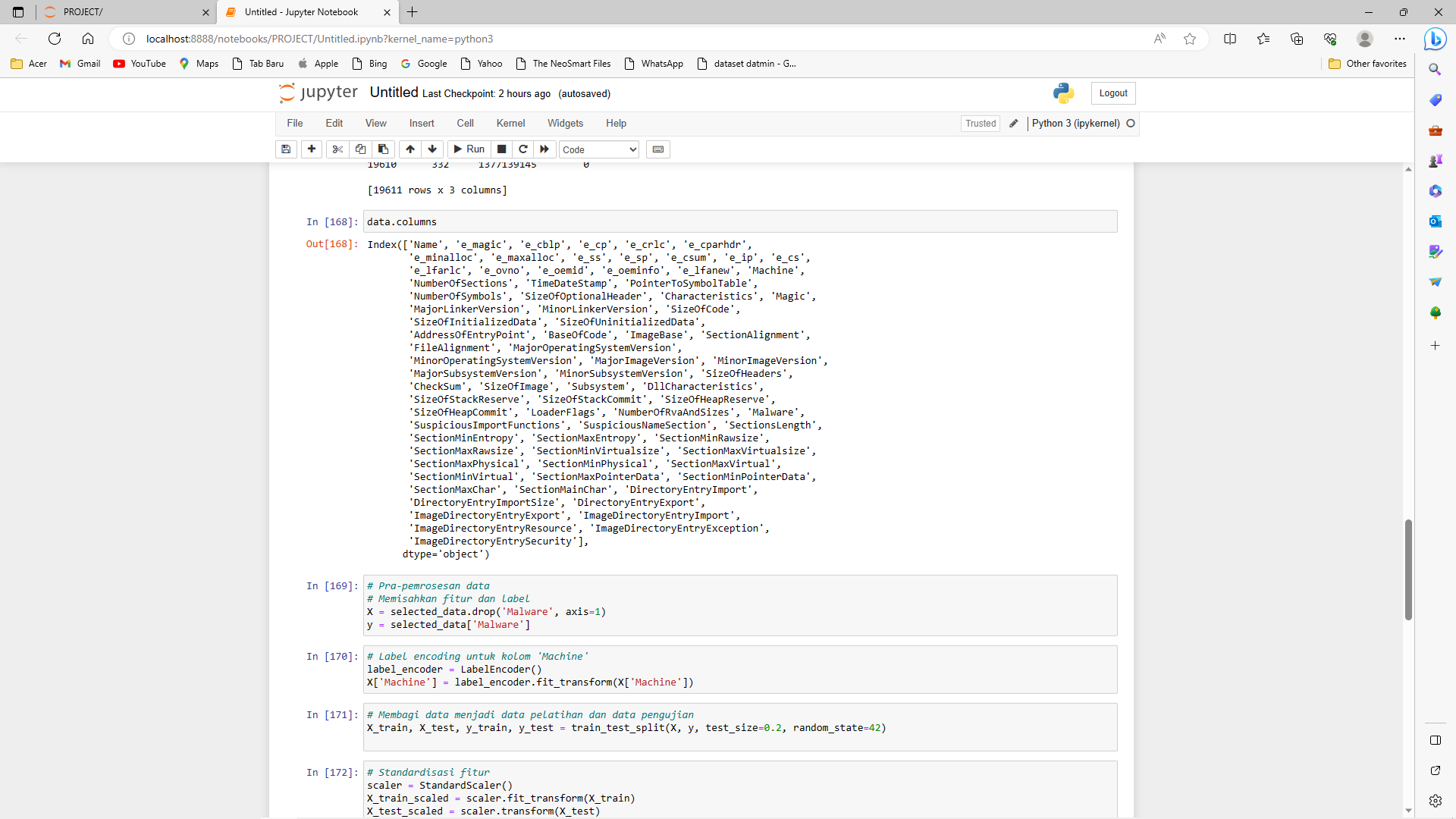
Gambar 3 Data Columns

Data yang digunakan dalam penelitian ini terdiri dari berbagai fitur (variabel) yang mungkin memiliki dampak dalam klasifikasi, serta label yang menunjukkan klasifikasi atau jenis serangan (benign atau malicious) dari sampel-sampel data. Pada kode tersebut, tahap pertama adalah memisahkan kolom 'Malware' sebagai label dari data. Kolom 'Malware' dihapus dari fitur, dan variabel X menyimpan fitur-fitur yang digunakan untuk melatih dan menguji model, sedangkan variabel y menyimpan label yang akan digunakan dalam proses pembelajaran dan evaluasi.



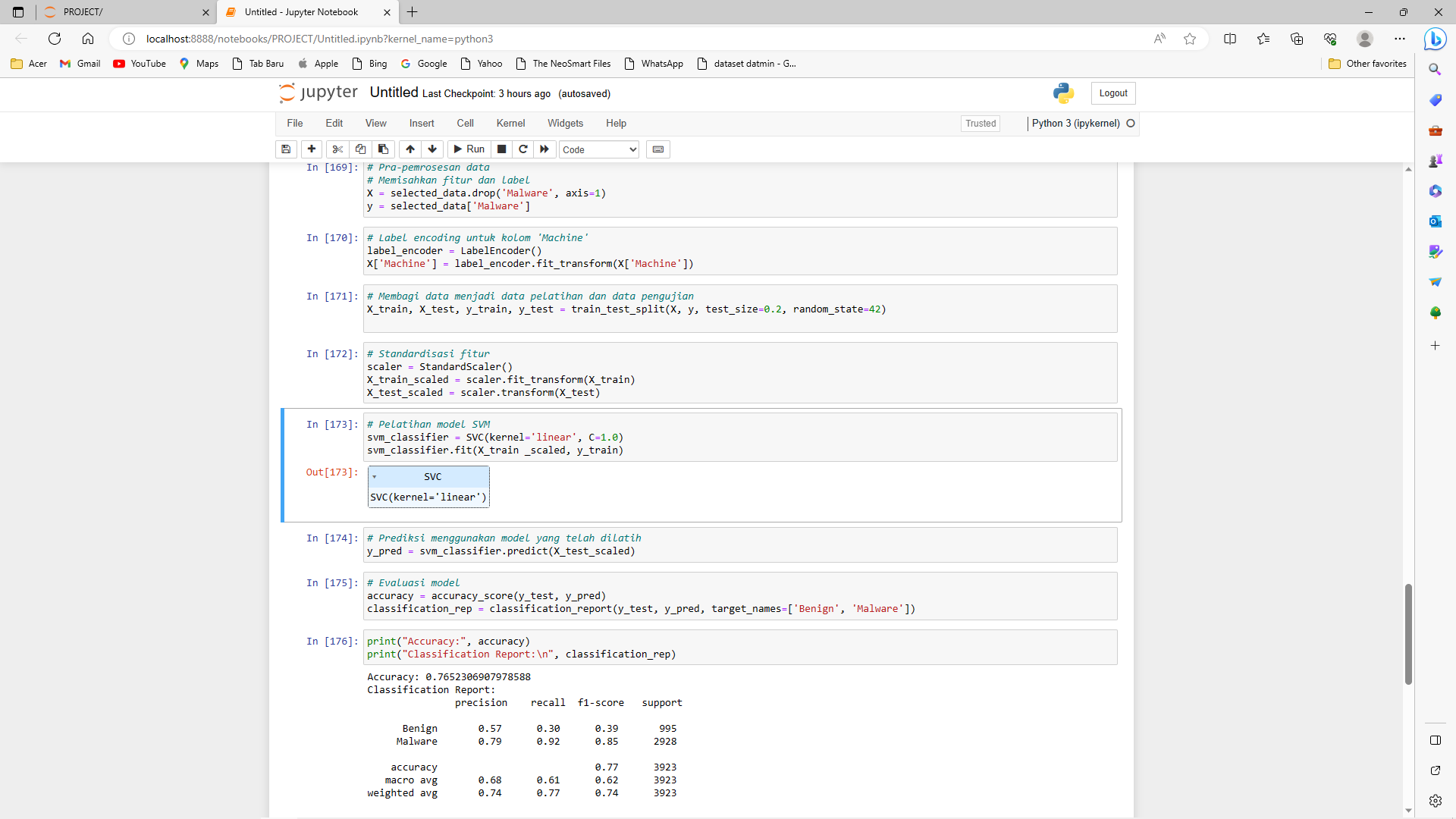
Gambar 4 Memisahkan Fitur dan Label

Data yang digunakan dalam penelitian ini terdiri dari berbagai fitur (variabel) yang mungkin memiliki dampak dalam klasifikasi, serta label yang menunjukkan klasifikasi atau jenis serangan (benign atau malicious) dari sampel-sampel data. Pada kode tersebut, tahap pertama adalah memisahkan kolom 'Malware' sebagai label dari data. Kolom 'Malware' dihapus dari fitur, dan variabel X menyimpan fitur-fitur yang digunakan untuk melatih dan menguji model, sedangkan variabel y menyimpan label yang akan digunakan dalam proses pembelajaran dan evaluasi.



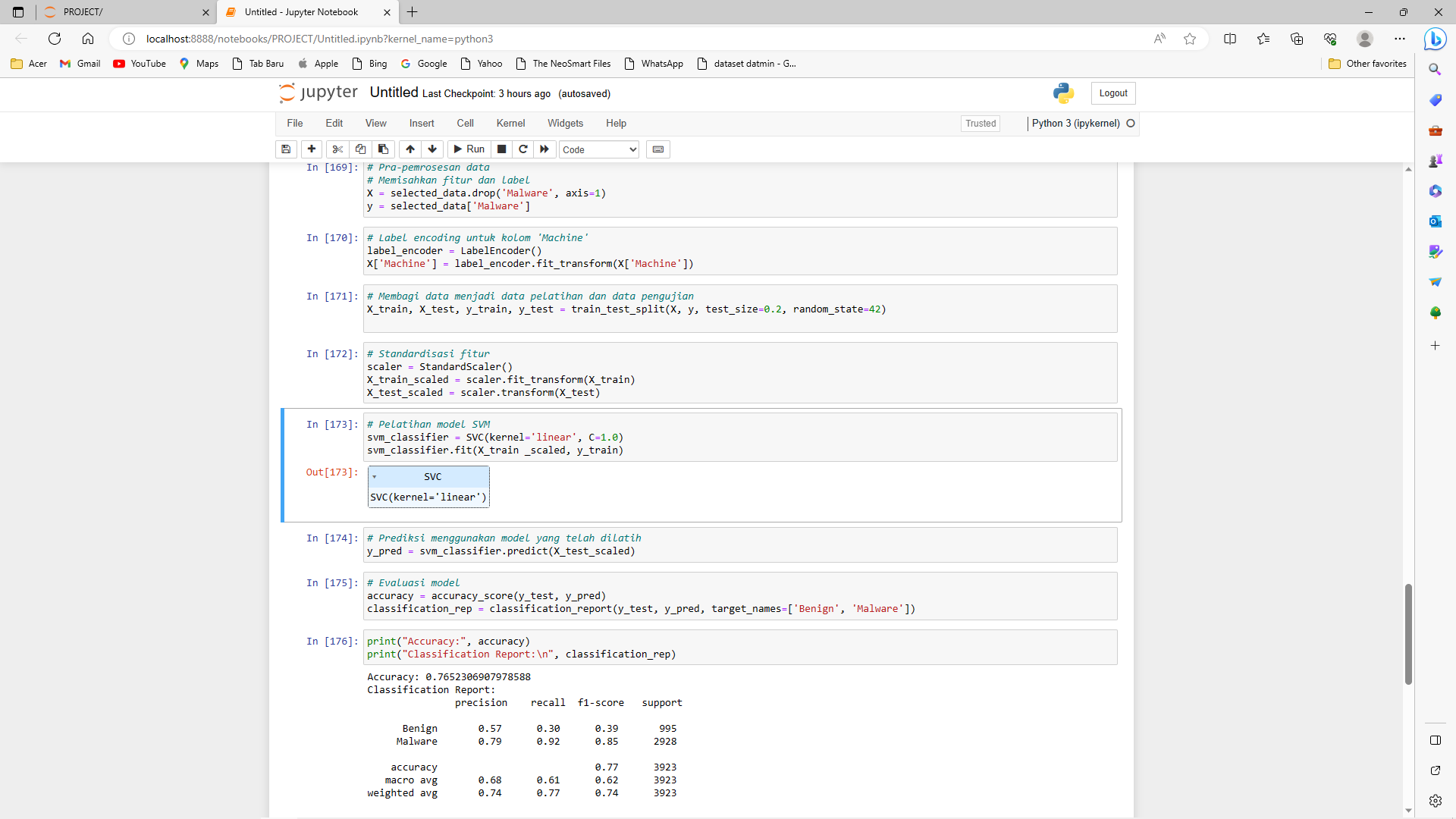
Gambar 5 Label Encoding untuk Kolom Machine

Data yang telah di-preprocess selanjutnya dibagi menjadi data pelatihan dan data pengujian. Pembagian ini bertujuan untuk menguji performa model yang telah dilatih pada data yang belum pernah dilihat sebelumnya. Pada kode tersebut, fungsi train\_test\_split digunakan untuk membagi data menjadi dua subset: X\_train dan X\_test (untuk fitur) serta y\_train dan y\_test (untuk label). Parameter test\_size menentukan ukuran data pengujian yang diinginkan, dalam hal ini 20% dari total data.



Gambar 6 Pembagian Data menjadi Data Pelatihan dan Data Pengujian

Model SVM yang digunakan mengasumsikan bahwa fitur-fitur memiliki skala yang serupa. Oleh karena itu, pada tahap ini dilakukan standardisasi fitur. Fitur-fitur dalam data pelatihan dan data pengujian diperlakukan sehingga memiliki mean (rerata) nol dan deviasi standar (standard deviation) satu. Ini memastikan bahwa pengaruh fitur-fitur dengan skala yang lebih besar tidak mendominasi dalam proses pelatihan model.



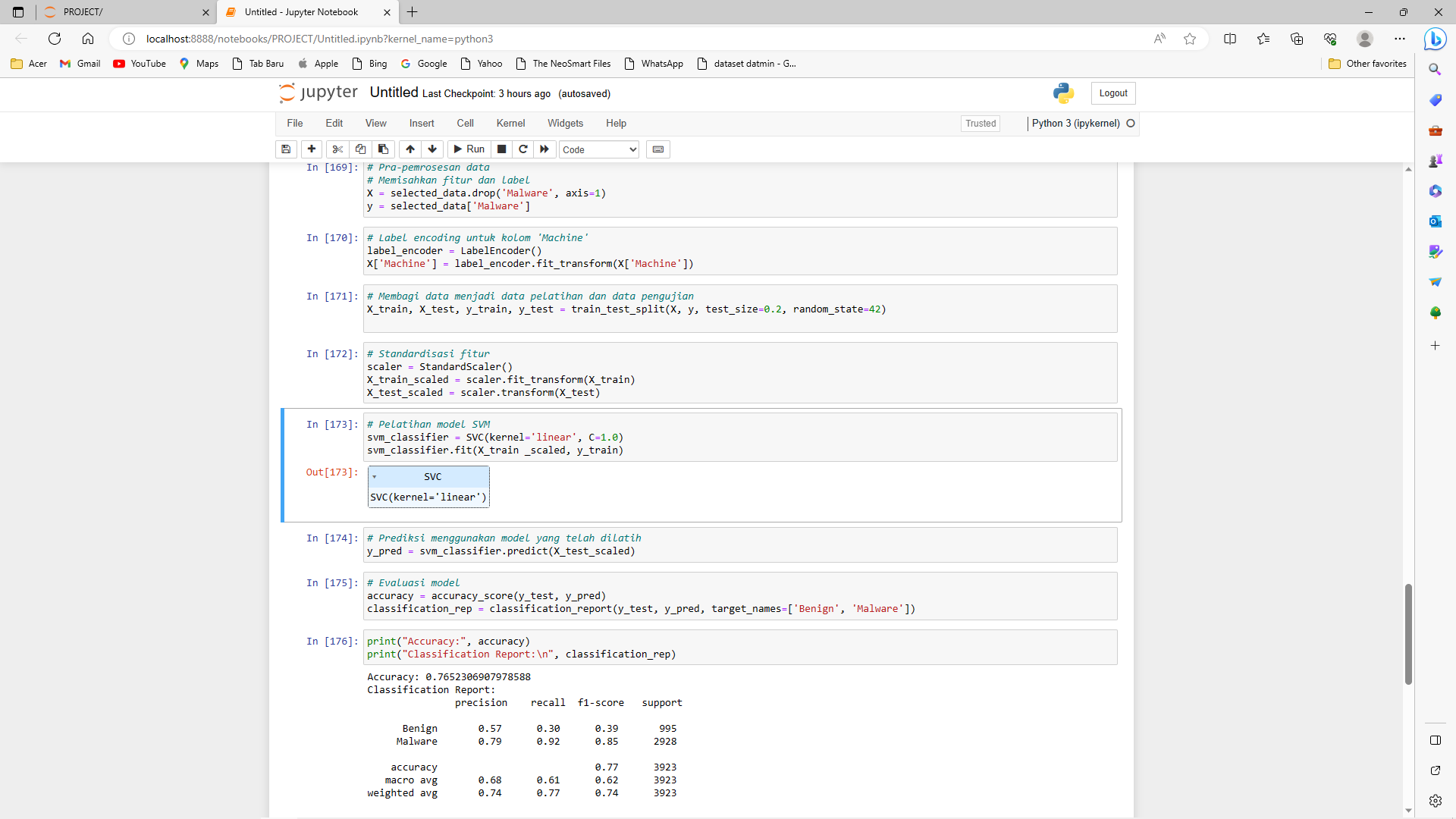
Gambar 7 Standardisasi Fitur

1. **Pemodelan dengan Support Vector Machines (SVM)**

Metode SVM diimplementasikan menggunakan pustaka scikit-learn. SVM merupakan algoritma yang digunakan untuk memisahkan dua kelas dengan mencari hyperplane terbaik yang memaksimalkan jarak antara kelas-kelas tersebut. Pada kasus ini, kelas yang akan dipisahkan adalah benign dan malicious. Pada kode yang diberikan, SVM dengan kernel linear digunakan untuk membangun model.

Pada langkah ini, model SVM akan dilatih menggunakan data pelatihan yang telah dipreprocess dan di-standardisasi sebelumnya. Terdapat dua tahap yaitu :

1. Inisialisasi Model SVM: Sebelum pelatihan dimulai, objek SVC (Support Vector Classifier) diinisialisasi dengan parameter yang ditentukan. Dalam kasus ini, parameter yang diatur adalah kernel dan C. kernel='linear' menunjukkan bahwa model SVM akan menggunakan kernel linear, yang cocok untuk masalah klasifikasi dengan fitur yang di-standardisasi. Nilai C mengontrol trade-off antara margin dan penalti terhadap kesalahan klasifikasi. Nilai yang lebih besar pada C memberikan penalti yang lebih besar terhadap kesalahan klasifikasi pada data pelatihan.
2. Pelatihan Model: Setelah objek model svm\_classifier diinisialisasi, langkah berikutnya adalah melatih model dengan data pelatihan yang telah di-standardisasi (X\_train\_scaled dan y\_train). Fungsi .fit() digunakan untuk melakukan pelatihan. Model SVM akan mempelajari hubungan antara fitur-fitur pada data pelatihan dan label yang sesuai (benign atau malicious). Tujuan dari pelatihan ini adalah untuk menemukan hyperplane yang dapat memisahkan dengan baik antara kelas-kelas yang ada dalam data pelatihan.



Gambar 8 Pelatihan Model SVM

Dengan melalui langkah pelatihan model ini, model SVM akan mempelajari pola-pola yang ada dalam data pelatihan yang telah di-standardisasi. Model ini akan menjadi representasi matematis dari batas keputusan yang memisahkan antara sampel-sampel benign dan malicious di dalam ruang fitur yang telah di-transformasi dan di-standardisasi. Model yang telah dilatih nantinya akan digunakan untuk melakukan klasifikasi pada data pengujian.

Langkah pelatihan model SVM ini merupakan tahap yang sangat penting dalam proses deteksi dan klasifikasi serangan malware, karena model yang dihasilkan dari tahap ini akan menjadi dasar bagi kemampuan model dalam mengenali dan membedakan antara sampel benign dan malicious pada data yang belum pernah dilihat sebelumnya.

1. **Evaluasi Model**

Setelah model Support Vector Machines (SVM) dilatih dan digunakan untuk melakukan prediksi pada data pengujian, tahap selanjutnya adalah melakukan evaluasi terhadap performa model tersebut. Evaluasi dilakukan untuk mengukur sejauh mana model mampu mengklasifikasikan sampel-sampel benign dan malicious dengan benar, serta untuk mendapatkan pemahaman lebih lanjut tentang kinerja model.

1. Akurasi (Accuracy)

Akurasi adalah salah satu metrik evaluasi yang umum digunakan untuk mengukur performa model klasifikasi. Pada langkah ini, kita menggunakan fungsi accuracy\_score(y\_test, y\_pred) untuk menghitung akurasi model, yaitu seberapa besar proporsi prediksi yang benar dibandingkan dengan keseluruhan data pengujian.

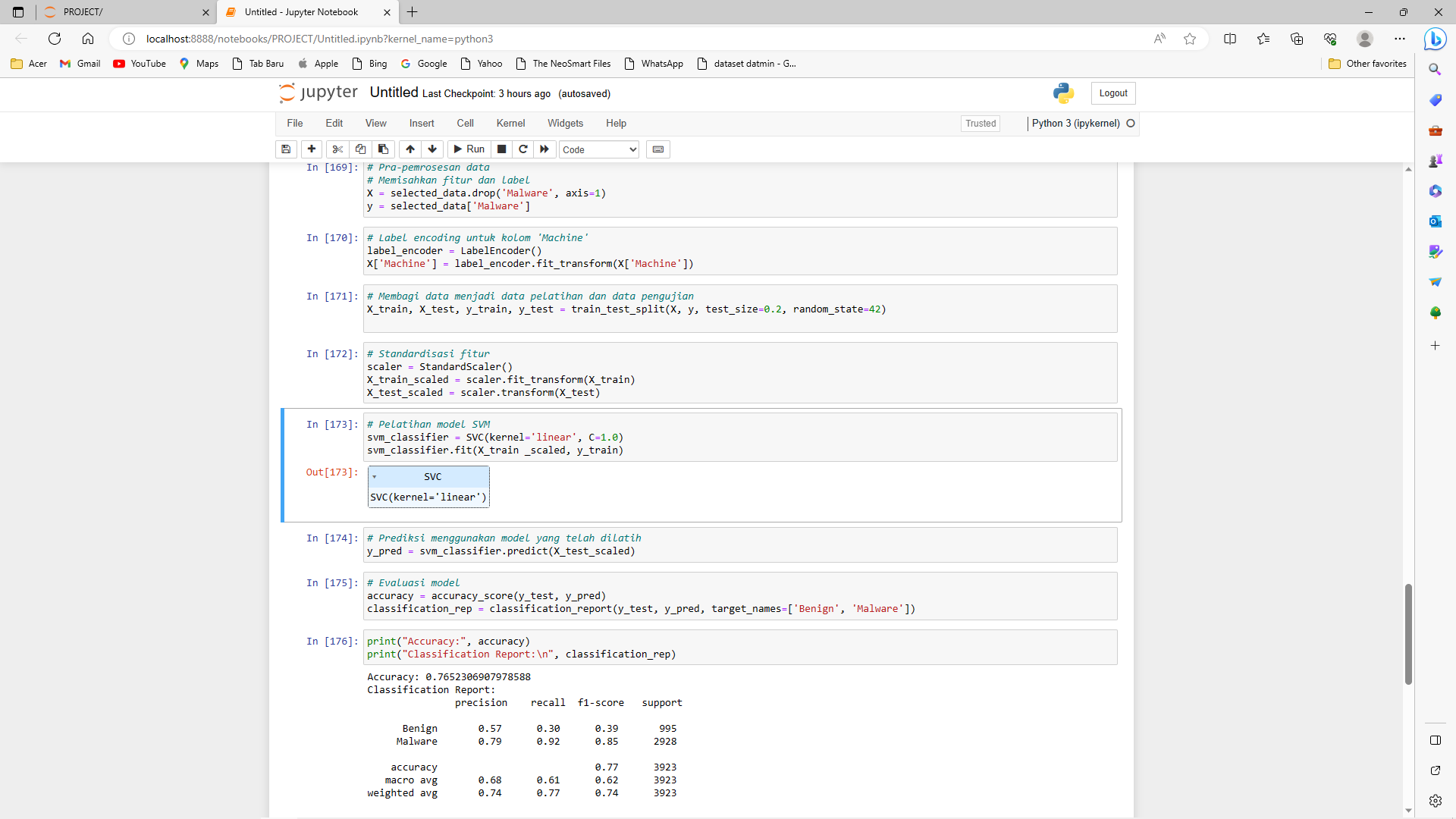
Akurasi memberikan gambaran umum tentang seberapa baik model dalam mengklasifikasikan seluruh sampel data pengujian dengan benar. Namun, perlu diingat bahwa akurasi dapat memberikan hasil yang bias tergantung pada keseimbangan kelas dalam dataset. Jika distribusi kelas tidak seimbang, akurasi mungkin tidak memberikan gambaran yang lengkap tentang kinerja model.

1. Laporan Klasifikasi (Classification Report)

Laporan klasifikasi (classification\_report) merupakan ringkasan yang lebih rinci tentang kinerja model dalam mengklasifikasikan tiap kelas secara terpisah. Pada langkah ini, menggunakan fungsi classification\_report(y\_test, y\_pred, target\_names=['Benign', 'Malware']) untuk menghasilkan laporan klasifikasi.

Laporan klasifikasi menyajikan beberapa metrik evaluasi untuk setiap kelas, termasuk:

1. Presisi (Precision): Seberapa banyak prediksi positif yang benar dibandingkan dengan total prediksi positif.
2. Recall (Recall atau Sensitivity): Seberapa banyak prediksi positif yang benar dibandingkan dengan total data sebenarnya yang seharusnya positif.
3. F1-Score: Harmonic mean dari presisi dan recall, memberikan gambaran keseluruhan tentang kinerja model.
4. Support: Jumlah sampel dalam setiap kelas.
5. Cvb bcfxb



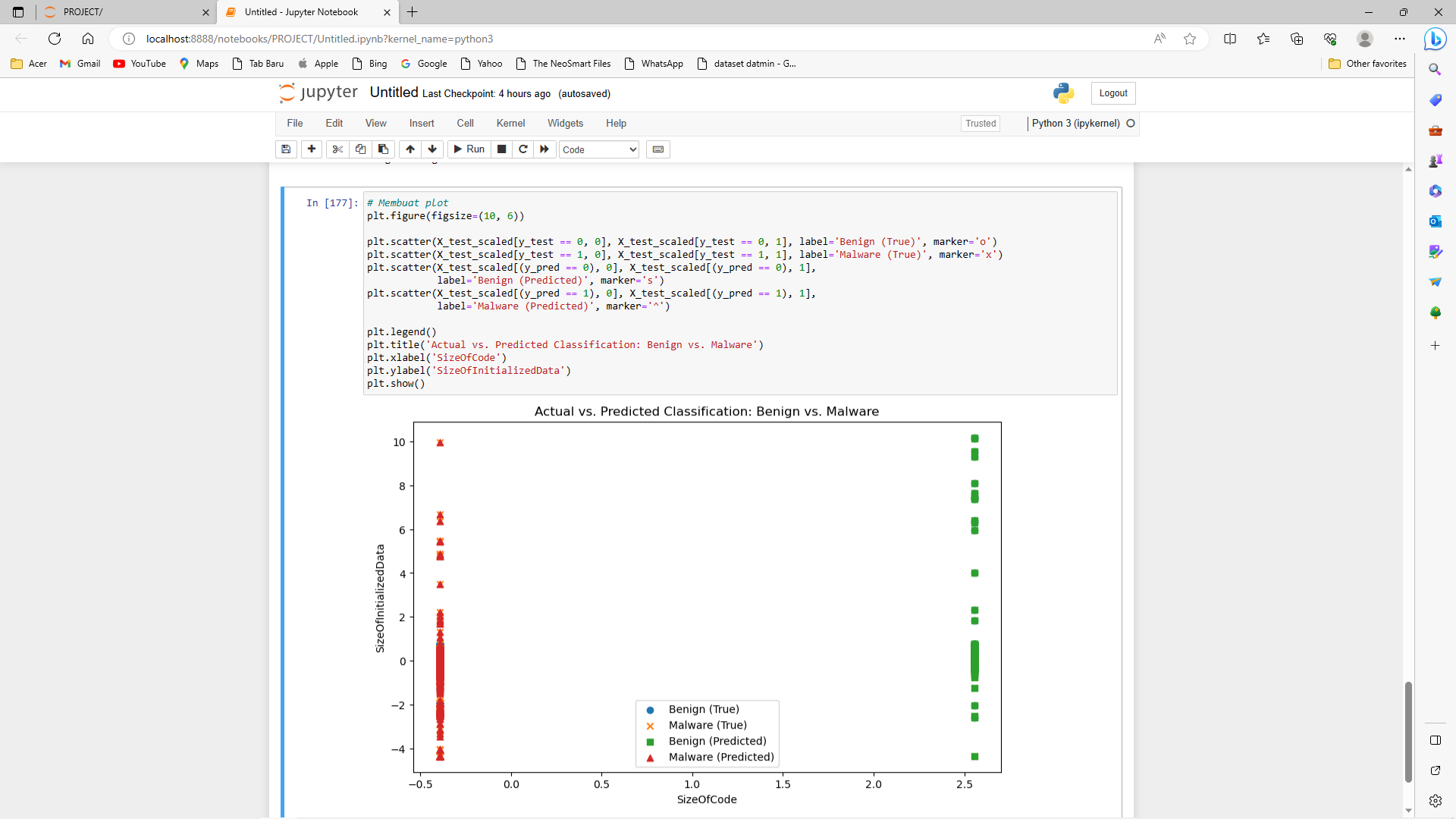
Gambar 9 Evaluasi Model

Akurasi model yang dihasilkan adalah sekitar 76.52%. Akurasi ini mengukur sejauh mana model mampu mengklasifikasikan seluruh sampel data pengujian dengan benar. Meskipun akurasi ini memberikan gambaran umum tentang kinerja model, kita perlu mempertimbangkan distribusi kelas dalam dataset. Akurasi yang tinggi mungkin disebabkan oleh kelas mayoritas (Malware) yang lebih banyak daripada kelas minoritas (Benign).

Dari hasil evaluasi yang telah dilakukan, terdapat beberapa hal yang perlu diperhatikan :

1. Kelas "Benign" memiliki nilai recall yang rendah (0.30), yang mengindikasikan bahwa model cenderung mengabaikan sejumlah besar sampel "Benign" yang seharusnya diidentifikasi. Meskipun presisi relatif lebih baik (0.57), f1-score yang rendah (0.39) menunjukkan bahwa terdapat kesulitan dalam menemukan keseimbangan antara presisi dan recall.
2. Kelas "Malware" memiliki nilai recall yang tinggi (0.92) dan presisi yang juga cukup baik (0.79), serta f1-score yang tinggi (0.85). Ini menunjukkan bahwa model mampu dengan baik dalam mengklasifikasikan sampel "Malware".
3. Nilai akurasi sebesar 76.52% mencerminkan sejauh mana model berhasil dalam mengklasifikasikan keseluruhan sampel, namun perlu diperhatikan bahwa distribusi kelas yang tidak seimbang dapat mempengaruhi interpretasi akurasi.
4. Nilai macro average (rata-rata dari precision, recall, dan f1-score) untuk kelas "Benign" lebih rendah (0.61) daripada kelas "Malware" (0.79), menunjukkan kesulitan dalam klasifikasi sampel "Benign".
5. Nilai weighted average untuk precision, recall, dan f1-score adalah 0.74, 0.77, dan 0.74 secara berurutan, yang mencerminkan rata-rata tertimbang dari kinerja model di semua kelas.
6. **Visualisasi Hasil**

Salah satu aspek menarik dari implementasi ini adalah visualisasi hasil klasifikasi menggunakan grafik scatter plot. Scatter plot tersebut menggambarkan sampel-sampel data pengujian yang telah diklasifikasikan oleh model SVM. Dalam grafik tersebut, sampel benign dan malicious yang telah diklasifikasikan dengan benar maupun yang salah dipisahkan menggunakan simbol-simbol yang berbeda. Visualisasi ini membantu memahami sejauh mana model dapat membedakan antara sampel-sampel benign dan malicious.



Gambar 10 Visualisasi Hasil Klasifikasi

## KESIMPULAN

Dalam penelitian ini, peneliti melakukan deteksi dan klasifikasi serangan malware pada dataset benign dan malicious menggunakan metode *Support Vector Machines* (SVM). Implementasi ini melibatkan beberapa tahap penting, termasuk pra-pemrosesan data, pembagian data training dan data testing, pemodelan dengan SVM, evaluasi model, dan visualisasi hasil akhir.

Hasil pra-pemrosesan data melibatkan pemilihan fitur-fitur yang relevan dan normalisasi data untuk memastikan kualitas dan kesesuaian. Data kemudian dipisahkan menjadi data training dan data testing untuk menguji performa model yang akan dibangun. Proses pelatihan model SVM melibatkan inisialisasi model dan pelatihan model dengan data pelatihan yang telah di-standardisasi.

Dalam evaluasi model, kami menggunakan akurasi sebagai ukuran umum performa model dan laporan klasifikasi (*classification report*) untuk mendapatkan pemahaman lebih rinci tentang kinerja model dalam mengklasifikasikan kedua kelas (*benign* dan *malicious*). Evaluasi ini memberikan pandangan yang lebih mendalam tentang kemampuan model dalam mengenali dan membedakan antara sampel-sampel benign dan malicious.

Hasil evaluasi menunjukkan bahwa model SVM memiliki akurasi sekitar 76.52%, namun distribusi kelas yang tidak seimbang mempengaruhi interpretasi akurasi. Model mampu dengan baik dalam mengklasifikasikan sampel-sampel "Malware", tetapi mengalami kesulitan dalam mengidentifikasi sampel-sampel "Benign". Nilai presisi, recall, dan f1-score juga memberikan gambaran tentang kinerja model dalam tiap kelas.

Terakhir, visualisasi hasil klasifikasi menggunakan *scatter plot* membantu memvisualisasikan sejauh mana model dapat membedakan antara sampel-sampel *benign* dan *malicious*. *Scatter plot* menggambarkan sampel-sampel yang sesuai dengan kelas aktual dan prediksi, membantu memahami pola distribusi yang dihasilkan oleh model SVM.

Secara keseluruhan, implementasi metode SVM dalam deteksi dan klasifikasi serangan *malware* pada dataset *benign* dan *malicious* memberikan pandangan tentang potensi dan batasan model ini. Penggunaan SVM dapat menjadi alat yang berguna dalam upaya melindungi sistem komputer dari ancaman serangan *malware*, namun evaluasi dan peningkatan lebih lanjut masih diperlukan untuk mengatasi tantangan dalam mengklasifikasikan sampel-sampel "Benign" yang lebih sulit dikenali.

## UCAPAN TERIMA KASIH

Saya ingin mengucapkan terima kasih yang tulus kepada semua pihak yang telah berkontribusi dan mendukung dalam penelitian ini. Tanpa bantuan dan dukungan dari berbagai pihak, penelitian ini tidak akan mungkin terlaksana.

Dengan penuh rasa syukur, saya berharap bahwa hasil dari penelitian ini dapat memberikan manfaat dan kontribusi positif dalam pengembangan bidang keamanan komputer dan ilmu pengetahuan secara lebih luas. Terima kasih sekali lagi kepada semua yang telah berperan dalam keberhasilan penelitian ini.

## DAFTAR PUSTAKA

Hartono, B. (2023). Ransomware: Memahami Ancaman Keamanan Digital. *Bincang Sains dan Teknologi*, *2*(02), 55-62.

Hasibuan, M. S. (2018). Keylogger pada Aspek Keamanan Komputer. *Jurnal Teknovasi: Jurnal Teknik dan Inovasi Mesin Otomotif, Komputer, Industri dan Elektronika*, *3*(1), 8-15.

Huang, C. Y., Tsai, Y. T., & Hsu, C. H. (2013). Performance evaluation on permission-based detection for android malware. In *Advances in Intelligent Systems and Applications-Volume 2: Proceedings of the International Computer Symposium ICS 2012 Held at Hualien, Taiwan, December 12–14, 2012* (pp. 111-120). Springer Berlin Heidelberg.

Mahendra, G. S., Wali, M., Idwan, H., Listartha, I. M. E., Yuliastuti, G. E., Sasongko, D., & Saskara, G. A. J. (2022). Keamanan Komputer'. *Galiono Digdaya*.

Munawar, Z., & Putri, N. I. (2020). Keamanan IoT Dengan Deep Learning dan Teknologi Big Data. *TEMATIK*, *7*(2), 161-185.

Sawitri, E., Astiti, M. S., & Fitriani, Y. (2019, July). Hambatan dan tantangan pembelajaran berbasis teknologi informasi dan komunikasi. In *Prosiding Seminar Nasional Program Pascasarjana Universitas Pgri Palembang*.

Setiawan, D. (2018). Dampak perkembangan teknologi informasi dan komunikasi terhadap budaya. *JURNAL SIMBOLIKA Research and Learning in Communication Study*, *4*(1), 62-72.

Tjahjadi, E. V., & Santoso, B. (2023). Klasifikasi Malware Menggunakan Teknik Machine Learning. *Jurnal Ilmiah Ilmu Komputer Banthayo Lo Komputer*, *2*(1), 60-70.