

Integrasi Algoritma *Support Vector Machine* dengan Java untuk Memprediksi Kualitas Komponen Otomotif dalam Industri 4.0

Integration of Support Vector Machine Algorithm with Java to Predict the Quality of Automotive Components in Industry 4.0

Mailia Putri Utami*¹, Finna Suroso², Fifi Lailasari H.³, Febry P.J. Sibuea⁴, Kevin Chandra⁵

Politeknik STMI Jakarta, Sistem Informasi Industri Otomotif

E-mail : mailiap2206@gmail.com*¹, finna@stmi.ac.id², fifulail@stmi.ac.id³, febrysibuea@stmi.ac.id⁴, kevinchandra@stmi.ac.id⁵

Received 9 May 2025; Revised 8 July 2025; Accepted 27 July 2025

Abstrak – Industri 4.0 menandai transformasi besar dalam sektor manufaktur, termasuk industri otomotif, dengan integrasi teknologi cerdas seperti machine learning untuk meningkatkan efisiensi dan kualitas produksi. Penelitian ini mengkaji penerapan algoritma *Support Vector Machine* (SVM) yang diintegrasikan dengan bahasa pemrograman Java untuk memprediksi kualitas komponen otomotif secara akurat. SVM dikenal efektif dalam klasifikasi data yang kompleks dan sangat cocok untuk lingkungan produksi yang memerlukan ketepatan tinggi. Proses penelitian mencakup pengumpulan dan pra-pemrosesan data kualitas komponen, pelatihan model SVM, serta implementasi model dalam platform Java guna memungkinkan integrasi dengan sistem otomasi industri yang telah ada. Hasil pengujian menunjukkan bahwa model SVM yang dibangun mampu mengklasifikasikan kualitas komponen dengan akurasi yang tinggi, memberikan potensi signifikan dalam pengurangan produk cacat dan peningkatan efisiensi produksi. Integrasi dengan Java memungkinkan sistem prediksi ini mudah diimplementasikan dalam infrastruktur perangkat lunak industri yang berbasis Java. Penelitian ini menunjukkan bahwa kombinasi machine learning dan pemrograman terapan dapat menjadi solusi strategis dalam mendukung transformasi digital industri otomotif di era Industri 4.0.

Kata kunci - *Support Vector Machine* (SVM), Java, Prediksi Kualitas, Komponen Otomotif, Industri 4.0

Abstract – Industry 4.0 marks a major transformation in the manufacturing sector, including the automotive industry, through the integration of intelligent technologies such as machine learning to enhance production efficiency and quality. This study examines the implementation of the *Support Vector Machine* (SVM) algorithm integrated with the Java programming language to accurately predict the quality of automotive components. SVM is known for its effectiveness in classifying complex data and is well-suited for production environments that require high precision. The research process involves data collection and preprocessing of component quality, training of the SVM model, and implementation of the model within a Java platform to allow seamless integration with existing industrial automation systems. Testing results show that the developed SVM model can classify component quality with high accuracy, offering significant potential in reducing defective products and improving production efficiency. Integration with Java enables this predictive system to be easily implemented into industrial software infrastructures based on Java. This study highlights that the combination of machine learning and applied programming can be a strategic solution in supporting the digital transformation of the automotive industry in the Industry 4.0 era.

Keywords - *Support Vector Machine* (SVM), Java, Prediksi Kualitas, Komponen Otomotif, Industri 4.0

1. PENDAHULUAN

Perkembangan teknologi dalam era Industri 4.0 telah membawa perubahan signifikan dalam dunia manufaktur, termasuk industri otomotif. Konsep digitalisasi, otomasi, dan kecerdasan buatan (AI) menjadi komponen utama dalam peningkatan efisiensi produksi dan kualitas produk [1]. Salah satu tantangan utama dalam industri otomotif adalah menjaga kualitas komponen yang diproduksi agar memenuhi standar yang ketat. Ketidaksesuaian dalam kualitas komponen dapat menyebabkan kerugian besar, baik dari segi biaya produksi ulang maupun reputasi perusahaan. Oleh karena itu, diperlukan sistem prediksi yang andal dan efisien untuk mendeteksi potensi cacat pada komponen sejak dini [2].

Beberapa studi telah mengkaji penerapan machine learning dalam bidang kualitas manufaktur. Misalnya, Prasetyo et al. (2021) menggunakan algoritma *Decision Tree* untuk klasifikasi kualitas komponen otomotif, namun akurasi masih terbatas pada dataset berskala kecil [1]. Penelitian oleh Wijaya dan Santosa (2022) menggunakan Random Forest, tetapi implementasinya belum terintegrasi dalam platform perangkat lunak industri [2].

Meskipun telah banyak penelitian menggunakan algoritma machine learning, masih terdapat keterbatasan dalam hal integrasi langsung dengan sistem industri dan pemanfaatan algoritma yang optimal untuk data non-linear. *Support Vector Machine* (SVM) memiliki keunggulan dalam menangani data dengan dimensi tinggi dan kompleksitas tinggi, namun implementasinya dalam konteks industri otomotif khususnya menggunakan bahasa pemrograman Java masih jarang ditemui [3].

Penelitian ini menawarkan pembaruan dengan mengintegrasikan algoritma SVM ke dalam sistem berbasis Java, yang merupakan bahasa pemrograman umum dalam industri otomotif dan sistem otomasi. Hal ini memungkinkan penerapan sistem prediksi kualitas secara real-time dan terintegrasi langsung dengan infrastruktur yang telah ada.

Adapun rumusan masalah pada penelitian ini adalah, bagaimana merancang sistem prediksi kualitas komponen otomotif menggunakan algoritma *Support Vector Machine* (SVM), proses implementasi algoritma *Support Vector Machine* (SVM) dalam platform berbasis Java, mengukur tingkat akurasi model *Support Vector Machine* (SVM) dalam memprediksi kualitas komponen otomotif [4].

Tujuan penelitian ini yang selaras dengan rumusan masalah di atas yaitu, Mengembangkan model prediksi kualitas komponen otomotif menggunakan algoritma SVM. Mengimplementasikan model tersebut dalam lingkungan pemrograman Java. Mengevaluasi performa model dalam hal akurasi, presisi, dan keandalan dalam konteks industri 4.0. Dengan batasan masalah yaitu data yang digunakan terbatas pada komponen otomotif tertentu yang diperoleh dari proses produksi spesifik. Model hanya dibangun menggunakan SVM tanpa perbandingan dengan algoritma lain. Implementasi sistem dilakukan dalam skala simulasi berbasis Java, belum diujicoba secara langsung pada lini produksi nyata.

2. METODE PENELITIAN

2.1. Jenis dan Pendekatan Penelitian

Penelitian ini termasuk dalam jenis penelitian terapan dengan pendekatan kuantitatif dan eksperimen, yang bertujuan untuk mengembangkan dan menguji sistem prediksi kualitas komponen otomotif berbasis algoritma *Support Vector Machine* (SVM) dan diimplementasikan menggunakan bahasa pemrograman Java [5].

2.2. Sumber dan Pengumpulan Data

Data diperoleh dari hasil dokumentasi proses produksi salah satu pabrik otomotif, khususnya data kualitas komponen tertentu (dengan, piston, bearing, atau gear). Setiap entri data mencakup sejumlah fitur seperti dimensi fisik, toleransi, hasil uji non-destruktif, dan label kualitas (*baik* atau *cacat*). Metode pengumpulan data meliputi [6]:

- 1) Observasi langsung kebagian lini produksi komponen otomotif
- 2) Pengambilan data historis dari sistem informasi manufaktur Perusahaan komponen otomotif
- 3) Validasi data dengan wawancara teknisi kualitas komponen otomotif di Perusahaan.

Adapun berikut ini sample dari beberapa data komponen otomotif yang digunakan:

Tabel 1. Data Komponen Otomotif

ID	Panjang (mm)	Diameter (mm)	Toleransi (mm)	Kekasaran Permukaan (μm)	Hasil Uji Non-Destruktif (Skor)	Kualitas (Label)
001	0.65	0.72	0.30	0.40	0.91	Baik
002	0.47	0.68	0.50	0.60	0.82	Cacat
003	0.71	0.76	0.25	0.35	0.94	Baik
004	0.55	0.65	0.60	0.72	0.79	Cacat
005	0.69	0.70	0.40	0.38	0.89	Baik
006	0.44	0.63	0.55	0.75	0.77	Cacat
007	0.74	0.77	0.22	0.32	0.96	Baik
008	0.52	0.66	0.48	0.68	0.80	Cacat
009	0.67	0.71	0.36	0.39	0.90	Baik
010	0.49	0.64	0.53	0.70	0.78	Cacat

2.3. Pra-pemrosesan Data

Sebelum digunakan untuk training model, data melewati beberapa tahapan berikut ini [7]:

- 1) Filtering data yaitu menghapus data duplikasi, data kosong, atau anomaly data.
- 2) Normalisasi data, yaitu menggunakan min-max scaling agar semua fitur berada pada skala yang sama
- 3) Pemisahan data, yaitu data dibagi menjadi dua bagian, yakni 80% untuk training dan 20% testing (pengujian).

2.4. Perancangan Model

Model Klasifikasi dikembangkan menggunakan algoritma *Support Vector Machine* dengan kernel RBF (Radial Basis Function), yang cocok untuk menangani data dengan pola non-linear. Paramater utama seperti C dan gamma akan ditentukan melalui proses grid search dan validasi silang (*cross-validation*) [7].

Algoritma *Support Vector Machine* (SVM) adalah metode pembelajaran mesin yang digunakan untuk klasifikasi dan regresi [14][1]. SVM bekerja dengan mencari hyperplane optimal yang memisahkan data ke dalam kelas yang berbeda dengan margin maksimum. Teknik ini sangat efektif dalam menangani data berdimensi tinggi dan digunakan secara luas dalam berbagai bidang, seperti pengenalan pola, analisis sentimen, dan prediksi [8].

$$\text{Prediksi} = \text{sign} \left(\sum_{i=1}^m \alpha_i y_i K(x_i, x) + b \right)$$

Di mana $K(x_i, x)$ adalah fungsi kernel yang menghitung jarak antara dua titik data x_i dan x dalam ruang fitur yang diubah oleh kernel, dan α adalah koefisien dari vektor bobot.

Tujuan dari SVM adalah untuk mencari hyperlane yang memisahkan dua kelas dengan margin terbesar, yang dapat dilihat sebagai masalah optimasi sebagai berikut:

$$\text{maximize} = \frac{1}{2} \|\omega\|^2$$

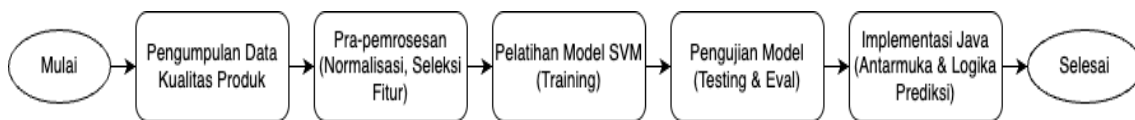
Dengan kendala:

$$Y_i(\omega^T \cdot x_i + b) \geq 1, \forall i$$

Y_i adalah label kelas untuk data x_i , dengan $y_i \in \{-1, +1\}$ dan x_i adalah data pelatihan. Kernel *Radial Basis Function* (RBF) adalah salah satu fungsi kernel yang paling umum digunakan dalam algoritma Support Vector Machine (SVM), terutama untuk menangani data yang tidak dapat dipisahkan secara linear. RBF kernel memetakan data ke dalam ruang berdimensi lebih tinggi sehingga memungkinkan SVM menemukan hyperplane pemisah yang optimal [9].

Grid Search adalah metode pencarian hiperparameter yang digunakan untuk menemukan kombinasi parameter terbaik dalam model pembelajaran mesin, seperti *Support Vector Machine* (SVM). Metode ini bekerja dengan mencoba semua kombinasi parameter yang telah ditentukan sebelumnya dan mengevaluasi kinerjanya menggunakan teknik validasi silang (cross-validation) [10].

Adapun berikut ini ilustrasi dari kerangka penelitian yang dilakukan:



Gambar 1. Kerangka Penelitian pererapan Model SVM

2.5. Implementasi Sistem

Model yang telah dilatih dan diuji akan diimplementasikan dalam bahasa pemrograman Java. Proses ini melibatkan, integrasi model SVM ke dalam Java menggunakan library seperti Weka atau libSVM, pembuatan antarmuka sederhana untuk input data dan menampilkan hasil klasifikasi, dan simulasi sistem dengan data uji untuk mengevaluasi integrasi dan respon sistem [11].

2.6. Desain Pengujian

Desain pengujian dalam penelitian ini dirancang untuk mengevaluasi performa model SVM secara objektif dan terukur. Setelah proses pelatihan model dengan data training, dilakukan pengujian terhadap model menggunakan data uji (test set) sebanyak 20% dari total dataset. Pengujian dilakukan untuk mengukur akurasi, presisi, recall, dan F1-score dari model klasifikasi.

Evaluasi performa dilakukan berdasarkan tiga kombinasi parameter SVM (C dan gamma) menggunakan pendekatan grid search dengan 5-fold cross-validation [12][1]. Hal ini bertujuan untuk memastikan bahwa parameter yang dipilih tidak hanya cocok untuk data training, tetapi juga memiliki kemampuan generalisasi yang baik.

Setelah parameter optimal ditemukan, model yang telah dilatih di Python kemudian diintegrasikan ke dalam sistem berbasis Java. Pengujian dilakukan untuk memastikan bahwa model yang dilatih dapat menerima input aktual dalam bentuk fitur fisik komponen otomotif dan menghasilkan output klasifikasi dengan cepat dan akurat.

Hasil klasifikasi dari sistem simulasi ini kemudian ditampilkan melalui antarmuka pengguna berbasis Java, yang memungkinkan pemantauan secara real-time dalam konteks industri otomotif [13][2].

Pengujian akhir terhadap sistem dilakukan dalam dua tahap [15] [3]:

1. Pengujian model di lingkungan Python untuk mengidentifikasi kombinasi parameter terbaik serta mengevaluasi performa model secara statistik.
2. Simulasi integrasi sistem di Java, yaitu menguji bagaimana model yang telah dilatih dapat menerima input aktual dan menghasilkan output klasifikasi melalui antarmuka pengguna, untuk mensimulasikan penerapan sistem dalam lingkungan industri otomotif.

3. HASIL DAN PEMBAHASAN

3.1. Implementasi Sistem

Implementasi algoritma SVM tidak hanya sebatas menjalankan perangkat lunak atau library, melainkan mencakup serangkaian proses berikut:

1. Pemilihan Kernel SVM:
Kernel yang digunakan adalah Radial Basis Function (RBF) karena data memiliki pola non-linear. Kernel ini berfungsi untuk memetakan data ke ruang berdimensi lebih tinggi sehingga dapat ditemukan hyperplane pemisah yang optimal.
2. Penentuan Parameter (C dan gamma):
Parameter C berperan sebagai pengendali margin klasifikasi dan kesalahan, sedangkan gamma mengatur seberapa jauh pengaruh satu titik data terhadap garis keputusan. Kombinasi optimal (C=10, gamma=0.1) diperoleh melalui proses grid search dengan validasi silang.
3. Proses Pelatihan
Model dilatih menggunakan data training sebesar 80% dari total data. Pada tahap ini, algoritma SVM menghitung vektor support dan menyusun hyperplane pemisah optimal berdasarkan nilai kernel RBF.
4. Evaluasi Model
Model dievaluasi dengan data uji sebesar 20% untuk menghitung metrik performa (akurasi, presisi, recall, dan F1-score).
5. Integrasasi Ke Java
Model yang telah dilatih di Python kemudian diekspor dan diintegrasikan ke dalam sistem berbasis Java menggunakan LibSVM, sebuah pustaka yang mendukung pemanggilan model SVM dalam berbagai bahasa pemrograman. Pembuatan antarmuka pengguna juga dilakukan untuk memungkinkan input manual fitur dan menampilkan hasil prediksi secara langsung.

Pengujian dilakukan terhadap model klasifikasi kualitas komponen otomotif menggunakan algoritma *Support Vector Machine* (SVM) dengan kernel *Radial Basis Function* (RBF). Dataset telah melalui proses prapemrosesan berupa normalisasi, pembersihan data duplikat dan pembagian dataset menjadi 80% data latih dan 20% data uji.

Pemilihan parameter optimal dilakukan melalui Grid Search dengan cakupan nilai: C = [0.1, 1, 10]

Gamma = [0.01, 0.1, 1]

Setiap kombinasi diuji dengan menggunakan 5-fold cross-validation. Berdasarkan hasil grid search, diperoleh kombinasi parameter terbaik yaitu C = 10 dan gamma = 0.1 yang memberikan hasil klasifikasi paling optimal.

Berikut adalah hasil evaluasi dari tiga kombinasi model yang mewakili variasi parameter berbeda:

Tabel 2. Hasil evaluasi dari 3 kombinasi model

Model	C	Gamma	Akurasi (%)	Presisi (%)	Recall (%)
Model 1	1	0.01	92.5	90.2	93.0
Model 2	1	1	88.4	86.1	89.5
Model 3 (Terbaik)	10	0.1	94.1	91.8	95.0

Hasil terbaik diperoleh pada Model 3 yang menunjukkan akurasi sebesar 94.1%, presisi sebesar 91.8%, dan recall sebesar 95.0%, yang mengindikasikan model mampu melakukan prediksi yang baik terhadap kualitas komponen, dengan kesalahan klasifikasi yang minim.

Selain itu, dihitung juga nilai F1-score untuk memberikan gambaran yang lebih seimbang antara presisi dan recall. Nilai F1-score dihitung menggunakan rumus:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Berdasarkan hasil evaluasi, F1-score untuk Model 3 adalah:

$$F1 - Score = 2 \times \frac{91.8 \times 95}{91.8 + 95} = 93.36\%$$

Nilai ini menunjukkan bahwa model tidak hanya akurat dalam prediksi, tetapi juga seimbang dalam mengklasifikasikan komponen baik maupun cacat.

3.2. Analisis Performa Model

- Akurasi Tinggi: Akurasi di atas 90% menunjukkan bahwa model memiliki performa klasifikasi yang sangat baik untuk dataset yang digunakan.
- Presisi mengukur ketepatan prediksi terhadap komponen yang dianggap berkualitas baik. Nilai presisi tinggi berarti hanya sedikit komponen cacat yang salah diklasifikasikan sebagai baik.
- Recall menunjukkan kemampuan model mendeteksi seluruh komponen cacat. Nilai recall sebesar 95.0% memperlihatkan bahwa sebagian besar komponen cacat berhasil diidentifikasi.

Dengan mempertimbangkan keseimbangan antara presisi dan recall, model menunjukkan kinerja yang stabil dan dapat diandalkan dalam konteks industri otomotif yang memerlukan deteksi dini terhadap cacat produksi.

3.3. Integrasi Model ke Dalam Platform Jawa

Model yang telah dilatih dan diuji menggunakan Python selanjutnya diintegrasikan ke dalam sistem berbasis Java menggunakan library LibSVM. Hasil integrasi memperlihatkan bahwa:

- Sistem dapat menerima input berupa fitur fisik komponen secara real-time.
- Proses klasifikasi berjalan cepat (< 1 detik per data).
- Hasil prediksi dapat ditampilkan langsung melalui antarmuka pengguna berbasis Java.

Proses ini berhasil menunjukkan bahwa algoritma SVM dengan kernel RBF dapat diterapkan secara efektif dalam sistem simulasi prediksi kualitas pada komponen otomotif dalam kerangka Industri 4.0.

3.4. Hasil dan Implikasi

- Penggunaan kernel RBF terbukti efektif dalam menangani data non-linear yang kompleks dari hasil produksi otomotif.
- Proses tuning parameter C dan gamma melalui grid search dan cross-validation menghasilkan model yang sangat akurat, tanpa overfitting terhadap data latih.
- Integrasi dengan Java memungkinkan sistem ini langsung digunakan atau diadaptasi ke dalam lingkungan industri otomasi yang umum menggunakan platform berbasis Java.
- Meskipun hasil akurat, sistem ini masih dalam skala simulasi; pengujian lebih lanjut pada data real-time dari lini produksi aktual diperlukan untuk validasi lebih kuat.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan dan mengimplementasikan model prediksi kualitas komponen otomotif berbasis algoritma Support Vector Machine (SVM) dengan kernel Radial Basis Function (RBF) yang diintegrasikan ke dalam platform berbasis Java, selaras dengan semangat transformasi digital di era Industri 4.0. Model SVM yang dikembangkan menunjukkan performa yang sangat baik, dengan nilai akurasi mencapai 94.1%, presisi 91.8%, dan recall 95.0% pada parameter optimal ($C = 10$, $\gamma = 0.1$), yang diperoleh melalui proses grid search dan validasi silang. Hasil ini menunjukkan bahwa model mampu mengklasifikasikan kualitas komponen secara akurat dan konsisten, bahkan pada data yang memiliki pola non-linear. Implementasi sistem menggunakan Java melalui integrasi dengan library LibSVM membuktikan bahwa model machine learning dapat diterapkan dalam sistem otomasi industri secara efisien. Sistem yang dikembangkan mampu melakukan prediksi secara real-time, responsif, dan dapat disesuaikan dengan infrastruktur perangkat lunak industri yang ada. Secara keseluruhan, penelitian ini menunjukkan bahwa kombinasi antara algoritma machine learning yang tepat (SVM) dan implementasi teknologi terapan (Java) dapat menjadi solusi strategis untuk meningkatkan efisiensi produksi dan deteksi dini terhadap cacat produk pada industri otomotif. Meski masih dalam skala simulasi, hasil penelitian ini memberikan dasar yang kuat untuk pengembangan lebih lanjut dan penerapan langsung dalam lini produksi nyata.

DAFTAR PUSTAKA

- [1] A. Prasetyo, R. Hidayat, dan M. F. Ardiansyah, "Klasifikasi Kualitas Komponen Otomotif Menggunakan Algoritma Decision Tree," *Jurnal Teknik Mesin*, vol. 9, no. 2, pp. 101–108, 2021.
- [2] A. Wijaya dan R. Santosa, "Penerapan Algoritma Random Forest untuk Prediksi Kualitas Komponen Manufaktur Otomotif," *Jurnal Sistem dan Teknologi Informasi*, vol. 10, no. 1, pp. 45–52, 2022.
- [3] H. Susilawati, et al., "Sistem Pendeteksian Kerusakan Mesin Sepeda Motor 4-Langkah Berbasis Suara Menggunakan Support Vector Machine (SVM)," *Semesta Teknika*, vol. 14, no. 1, pp. 1–10, 2015.
- [4] R. Rasyid, "Predictive Maintenance on Dry 8 Production Machine Line Using Support Vector Machine (SVM)," *Jurnal Teknik Industri*, vol. 23, no. 1, pp. 153–162, 2023.

- [5] M. Ichwan, I. A. Dewi, dan Z. M. S, "Klasifikasi Support Vector Machine (SVM) Untuk Menentukan Tingkat Kemanisan Manga Berdasarkan Fitur Warna," *MIND Journal*, vol. 3, no. 2, pp. 16–23, 2021
- [6] N. Sulistianingsih, F. Astutik, dan A. Rahman, "Optimasi Seleksi Fitur Untuk Perbaikan Akurasi Support Vector Machine Classifier Pada Klasifikasi Citra Tanaman Rimpang," *Jurnal Fasilkom*, vol. 14, no. 2, pp. 1–10, 2023.
- [7] D. H. Putri, M. Rizka, dan Z. K. Simbolon, "Rancang Bangun Sistem Identifikasi Kesegaran Ikan Berdasarkan Citra Mata Menggunakan Support Vector Machine," *Jurnal Teknologi Rekayasa Informasi dan Komputer*, vol. 8, no. 2, pp. 1–10, 2022.
- [8] R. A. Putri dan N. Rochmawati, "Penerapan Algoritma Support Vector Machine untuk Klasifikasi Motif Citra Batik Solo Berdasarkan Fitur Multi-Autoencoders," *Journal of Informatics and Computer Science (JINACS)*, vol. 1, no. 1, pp. 56–63, 2019.
- [9] R. D. Satriya, I. M. A. D. Suarjaya, dan I. P. A. E. Pratama, "Sentiment Analisis Antusias Masyarakat Terhadap Sampah Plastik Dengan Menggunakan Metode Support Vector Machine (SVM)," *JITTER: Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, no. 1, pp. 880–893, 2022.
- [10] Nugraha, "Sentimen Analisis Penerapan Social Distancing Menggunakan Feature Selection Pada Algoritma Support Vector Machine," *Jurnal Teknik Komputer*, vol. 8, no. 2, pp. 1–10, 2021.
- [11] Abu Alhija, M. Azzeh, dan F. Almasalha, "Software Defect Prediction Using Support Vector Machine," *arXiv preprint arXiv:2209.14299*, 2022.
- [12] D. S. T. Muhammad Yusuf Ramadan, "Implementasi Metode Klasifikasi Support Vector Machine (SVM) Terhadap Pemakaian Minyak Goreng," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3 No.2, pp. 1669-1677, 2019.
- [13] S. S. Sanjay Yadav, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," *IEEE*, pp. 78-83, 2016.
- [14] A. W. Didin Muhidin, "Perbandingan Kinerja Algoritma Support Vector Machine Dan K-Nearest Neighbor Terhadap Analisis Sentimen Kebijakan New Normal," *String (Satuan Tulisan Riset dan Inovasi Teknologi)*, pp. 153-159, 2020.
- [15] S. E. Risa Wati, "Analisis Sentimen Persepsi Publik Mengenai PPKM Pada Twitter Berbasis SVM Menggunakan Python," *Jurnal Teknik Informatika*, vol. 06 Nomor 02, pp. 240-247, 2021.