OPTIMASI MODEL DENGAN ALGORITMA *SUPPORT VECTOR REGRESSOR* MENGGUNAKAN *GRID SEARCH* PADA PENILAIAN ESSAI OTOMATIS

Ahya Radiatul Kamila, Very Budiyanto
Sains Data, Fakultas Teknologi dan Desain, Universitas Bunda Mulia, Jakarta
Manajemen, Fakultas Ilmu Sosial dan Humaniora, Universitas Bunda Mulia, Jakarta
Jalan Lodan Raya No 2, RT 12 RW 02, Jakarta, Indonesia
akamila@bundamulia.ac.id

ABSTRAK

Proses evaluasi siswa adalah bagian penting dalam pendidikan untuk mengukur pemahaman dan kemajuan siswa secara menyeluruh. Evaluasi yang efektif harus bersifat komprehensif dan adil, salah satunya melalui soal esai yang memungkinkan siswa menunjukkan kemampuan berpikir kritis dan analitis. Namun, penilaian esai menghadapi tantangan seperti subjektivitas, waktu koreksi yang lebih lama, dan konsistensi penilaian. Untuk mengatasi masalah ini, digunakan penilaian esai otomatis dengan algoritma machine learning. Agar model machine learning dapat bekerja dengan baik, diperlukan data berkualitas dan pendekatan yang tepat. Penelitian ini berfokus pada optimasi algoritma Support Vector Regressor (SVR) untuk meningkatkan performa model. Optimasi dilakukan dengan penyesuaian pada tahap preprocessing dan modeling. Pada preprocessing, dibuat fitur baru, sementara pada modeling dilakukan perbandingan sebelum dan setelah tuning hyperparameter menggunakan Grid Search. Hasil penelitian menunjukkan bahwa optimasi SVR dengan Grid Search berhasil meningkatkan performa model, terbukti dari penurunan nilai MSE dari 0.0393 menjadi 0.0325. Selain itu, waktu komputasi sebelum dan setelah penggunaan Grid Search juga dibandingkan, yaitu 148ms dan 502.2ms. Meskipun Grid Search memerlukan waktu lebih lama, penurunan error yang signifikan sebanding dengan waktu tambahan tersebut.

Kata kunci : *Penilaian esai otomatis; Machine Learning; Support Vector Regressor; Grid Search;Hyperparameter Tuning*

1. PENDAHULUAN

Dalam era digital yang semakin berkembang, sistem pendidikan terus beradaptasi dengan teknologi untuk meningkatkan efisiensi dan efektivitas proses pembelajaran. Salah satu aspek penting dalam pendidikan adalah penilaian, yang memegang peranan penting dalam mengevaluasi kemampuan dan pemahaman siswa. Penilaian *essay* memiliki keunggulan tersendiri dibandingkan bentuk penilaian lainnya, seperti pilihan ganda atau jawaban singkat, karena memungkinkan siswa untuk mengungkapkan pemikiran mereka secara mendalam dan menunjukkan kemampuan analitis serta keterampilan menulis. Melalui *essay*, pendidik dapat menilai kemampuan berpikir kritis, pemahaman konsep, dan kemampuan mengorganisir serta menyampaikan ide secara jelas dan logis. Oleh karena itu, evaluasi pembelajaran dengan soal *essay* menjadi krusial dalam memberikan gambaran yang lebih komprehensif mengenai kompetensi siswa. Penilaian *essay*, yang memerlukan analisis mendalam terhadap konten tulisan, sering kali menjadi tantangan bagi pendidik karena membutuhkan waktu dan usaha dalam memberikan nilai yang adil tanpa dipengaruhi oleh faktor subjektivitas. Dalam upaya untuk mengatasi tantangan ini, penelitian di bidang pengolahan bahasa alami (*Natural Language Processing*, NLP) dengan algoritma *machine learning* telah mengarah pada pengembangan sistem penilaian otomatis yang dapat memberikan penilaian cepat dan konsisten terhadap *essay* siswa [1].

Machine learning merupakan cabang dari kecerdasan buatan yang memungkinkan mesin untuk belajar dari data historical untuk dapat memprediksi pada data baru [2].

Telah dilakukan beberapa penelitian yang bertujuan untuk menilai *essay* siswa menggunakan pendekatan *machine learning*, diantaranya adalah penelitian menggunakan algoritma *Support Vector Machine* juga diusulkan oleh untuk *essay* pendek berbahasa jepang dengan penggunaan *Term Frequency-Inverse Document Frequency*.

Algoritma SVM juga digunakan oleh [3] untuk memprediksi nilai siswa dalam penilaian *essay*. Selain itu, dilakukan pula pendekatan menggunakan algoritma SVM dan *Genetetic Algorithms* oleh untuk menilai *essay* dengan subject *muscular system*. Pendekatan lain untuk penilaian essai otomatis dilakukan oleh yang menggunakan *Two-Stage Learning Framework* berhasil meningkatkan kekuatan model.

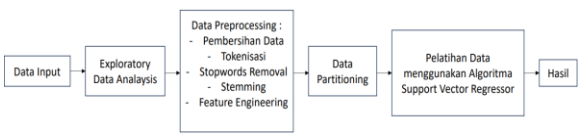
Penggunaan algoritma *pre-trained deep learning* diusulkan oleh [4] dengan 0.829 *F1-score* pada tahun 2019. Adapun sebelumnya, tahun 2018 diusulkan pula pendekatan dengan algoritma *deep learning* oleh dengan *Siamese Bidirectional Long Short Term Memory*. Dari beberapa penelitian yang pernah dilakukan sebelumnya, algoritma *Support Vector Machine* merupakan salah satu algoritma yang sering digunakan dalam pembanguanan model *automated essay grading*. Hal ini mengindikasikan bahwa *Support Vector Machine* (SVM) merupakan salah satu

teknik *machine learning* yang menunjukkan potensi besar dalam aplikasi penilaian *essay* otomatis. Adapun algoritma *Support Vector Regressor* (SVR) merupakan algoritma yang menggunakan konsep pendekatan *Support Vector Machines* (SVM) untuk permasalahan regresi. SVR dirancang untuk menyelessaikan masalah regresi dengan mencari *hyperplane* yang paling sesuai untuk memprediksi nilai dari suatu data kontinu. Dalam permasalahan *automated essay grading* dengan tujuan utamanya adalah untuk menilai skor siswa berdasarkan kecocokan jawaban dosen, dibutuhkan pendekatan berbasis regresi. Hal ini dikarenakan *variable dependent* dalam kasus ini berupa angka kontinu.

Penelitian ini bertujuan untuk mengimplementasikan algoritma *Support Vector Regressor* dalam sistem penilaian *essay* otomatis dengan pendekatan yang dapat meningkatkan akurasi penilaian. Dengan memanfaatkan berbagai teknik analisis teks, diharapkan sistem dapat memberikan penilaian yang akurat. Studi ini akan menguji efektivitas SVR dalam mengolah berbagai jenis *essay* dan mengevaluasi kinerja model berdasarkan berbagai metrik evaluasi. Implementasi sistem penilaian *essay* otomatis yang efektif dan efisien dapat memberikan berbagai manfaat, termasuk mengurangi beban kerja pendidik, meningkatkan konsistensi penilaian dengan mengurangi subjektivitas, dan memberikan umpan balik yang cepat kepada siswa. Oleh karena itu, penelitian ini tidak hanya berkontribusi pada pengembangan teknologi, tetapi juga pada peningkatan kualitas pembelajaran di berbagai tingkat pendidikan.

2. METODE PENELITIAN

Penelitian ini dirancang untuk mengembangkan dan mengimplementasikan sistem penilaian *essay* otomatis menggunakan *Natural Language Processing* (NLP) dan algoritma *Support Vector Machine* (SVM). Metodologi yang digunakan dalam penelitian ini mencakup beberapa tahapan utama, yaitu pengumpulan data, analisa data, prapemrosesan teks, pelatihan model, dan evaluasi kinerja. Hasil penelitian diharapkan dapat menunjukkan bahwa dengan menggunakan NLP dan SVM, sistem penilaian esai otomatis mampu menilai esai secara konsisten dan akurat, sekaligus memberikan manfaat efisiensi waktu dalam proses penilaian dibandingkan dengan cara manual. Penggunaan sistem ini juga memiliki potensi untuk diadaptasi pada berbagai aplikasi lainnya di bidang pendidikan maupun sektor lain yang memerlukan penilaian otomatis terhadap teks



Gambar 1. Blok Diagram

2.1. *Exploratory Data Analysis*

Pengumpulan data merupakan langkah awal yang krusial dalam penelitian ini. Data *essay* yang digunakan dikumpulkan secara eksklusif dari institusi pendidikan. Data - data ini berasal dari jawaban siswa dengan variasi yang beragam untuk memastikan model dapat menangani berbagai jenis tulisan. Setiap jawaban yang dikumpulkan dilengkapi dengan skor penilaian yang diberikan oleh pendidik atau *evaluator* manusia sebagai referensi untuk melatih model. Selanjutnya, penulis melakukan tahap *Exploratory Data Analysis* (EDA) yang merupakan langkah penting dalam memahami karakteristik dan distribusi data [5].

Pada tahap ini, berbagai teknik analisis statistik dan visualisasi data digunakan untuk mengeksplorasi dan menganalisis data secara mendalam.

2.2. *Preprocessing*

Proses *preprocessing* dilakukan untuk membersihkan teks *essay* dari elemen-elemen yang tidak relevan, seperti tanda baca dan karakter khusus, serta untuk mengubah teks ke dalam format yang bisa diproses lebih lanjut [6][7].

Langkah-langkah seperti tokenisasi, penghapusan *stopwords*, dan *stemming* dilakukan untuk menghasilkan representasi teks yang lebih sederhana dan konsisten. Selanjutnya, fitur-fitur linguistik seperti frekuensi kata-kata kunci diekstraksi menggunakan teknik-teknik seperti *Bag of Words* (BoW) [8].

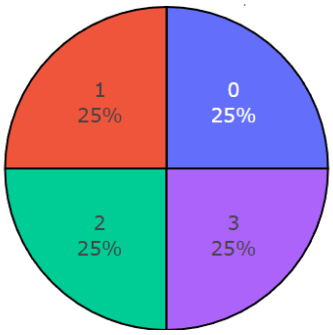
Pertama-tama, data teks diproses dengan tahap pembersihan data untuk menghilangkan karakter-karakter tidak penting seperti tanda baca, karakter khusus, dan simbol yang tidak relevan bagi analisis teks. Hal ini bertujuan untuk membersihkan teks agar lebih mudah diproses oleh algoritma. Langkah selanjutnya adalah tokenisasi, di mana teks dibagi menjadi unit-unit lebih kecil seperti kata-kata atau frasa [9].

Misalnya, kalimat "Saya suka makan nasi goreng" akan dibagi menjadi token-token seperti ["Saya", "suka", "makan", "nasi", "goreng"]. Setelah tokenisasi, langkah selanjutnya adalah menghapus kata-kata penghubung atau *stop word* dari teks. *Stop word* ini umumnya tidak memberikan kontribusi yang signifikan terhadap makna atau konteks teks secara keseluruhan [10].

Dengan menghilangkan *stop word*, analisis teks dapat lebih fokus pada kata-kata kunci yang memiliki informasi yang lebih bermakna. Kemudian, dilakukan proses *stemming*. Proses ini adalah proses untuk mengubah kata-kata menjadi bentuk dasarnya (kata dasar) [11].

Ini dilakukan dengan menghapus akhiran kata (*suffix*) sehingga kata-kata yang berbeda dengan akhiran yang sama akan diubah menjadi bentuk dasar yang sama . Misalnya, kata-kata "makanan", "makan", "makanlah" akan distem menjadi "makan". Hal ini membantu mengurangi variasi kata yang memiliki akar kata yang sama dalam analisis teks.

Penelitian ini juga menggunakan pendekatan *feature engineering* dalam tahap *preprocessing*-nya untuk meningkatkan performa model. Tahap *feature engineering* yang digunakan dalam penelitian ini adalah dengan menambahkan fitur baru yang didapatkan dari ekstraksi fitur yang sebelumnya telah terdapat pada dataset. Fitur tambahan dalam penelitian ini merupakan fitur 'Kategori Penilaian' yang dibuat dengan tujuan memudahkan model dalam mengidentifikasi dan mengelompokkan teks berdasarkan *range* kategori penilaian. Adapun kategori penilaian dengan kelas '0' merujuk pada nilai dengan rentang 0-54, kelas '1' merujuk pada nilai dengan rentang 55 – 69, kelas '2' merujuk pada nilai dengan rentang 70 – 84, dan kelas '3' meurujuk pada rentang nilai 85 – 100. Seluruh kelas pada kategori penilaian memiliki jumlah yang seimbang. Hal ini dilakukan untuk memastikan bahwa model tidak bias terhadap kelas tertentu, sehingga hasil prediksi menjadi lebih akurat. Fitur ini diharapkan dapat membantu model dalam memahami konteks data teks lebih baik dan meningkatkan akurasi prediksi.



Gambar 2. Pesebaran Data pada Kelas Kategori Penilaian

Setelah itu, data *essay* dibagi menjadi dua set: data latih dan data uji, dengan proporsi yang sesuai untuk memastikan validitas hasil penelitian. Pembagian ini dilakukan secara acak untuk memastikan bahwa model tidak bias terhadap pola tertentu dalam data. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi kinerja model yang telah dilatih.

2.3. *Modeling*

Pada tahap *modeling*, SVR dipilih sebagai metode untuk membangun model prediksi skor penilaian *essay*. Model dilatih dengan menggunakan data latih yang telah melalui tahap *preprocessing*, di mana SVR mempelajari hubungan antara fitur-fitur linguistik dari *essay* dan skor penilaian manusia. Proses ini melibatkan tuning *hyperparameter* untuk meningkatkan kinerja model. *Support Vector Regressor* merupakan algoritma yang digunakan untuk menyelesaikan kasus regresi dengan prinsip kerja *Support Vector Machine* (SVM).

Algoritma ini merupakan algoritma yang dapat digunakan untuk permasalahan klasifikasi dan regresi [12] dengan mencari *hyperplane* (garis pemisah) yang optimal untuk memisahkan data ke dalam kelas-kelas yang berbeda dengan dataset {xi, yi} dengan xi adalah *variable independent* dan yi adalah *dependent variable* [13].

Adapun rumus untuk *hyperplane* dapat dilihat pada persamaan (1) dengan *variable* w adalah *perpendicular vector* dan b merepresentasikan lokasi *determinant* dari fungsi pembagian *relative*.

$$x_i . w + b = 0 \quad (1)$$

Adapun persamaan (2) & (3) menggambarkan cara lain untuk membagi satu kelas dengan kelas yang lain.

$$x_i . w + b \geq 1 \; for \; y_i = 1 \quad (2)$$
$$x_i . w + b \leq -1 \; for \; y_i = -1 \quad (3)$$

Persamaan di atas dapat digabungkan sebagaimana yang tertulis pada persamaan (4). Dimana persamaan tersebut dapat digunakan untuk memisahkan data *linear*. Sementara, untuk data yang tidak *linear (non-linear data*), digunakan *soft margin hyperplane* dengan menambahkan *variable* dimana i = 1…. L. sehingga, didapatkan persamaan (5) dapat digunakan sebagai rumus untuk memisahkan data dengan tipe *non-linear*.

$$y_i(x_i . w + b) - 1 \geq 0 \; \forall_i \quad (4)$$
$$\text{Min} \frac{1}{2} |w|^2 + C \sum_{i=1}^{L} \xi_i \; s.t \; y_i(x_i . w + b) - 1 + \xi_i \geq 0 \; \forall_i \quad (5)$$

Dari persamaan di atas, dapat diketahui bahwa algoritma SVM memiliki keunggulan dalam menangani data yang tidak *linear* melalui penggunaan *kernel trick*, yang memungkinkan data dipetakan ke dimensi yang lebih tinggi di mana data dapat dipisahkan secara *linear*. Beberapa jenis kernel yang umum digunakan antara lain *linear*, *polynomial*, *radial basis function* (RBF), dan sigmoid. Dengan kemampuan untuk menangani data yang kompleks dan performa yang baik dalam berbagai aplikasi, SVM menjadi salah satu algoritma yang populer dalam *machine learning*, baik untuk tugas klasifikasi maupun regresi.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, dilakukan implementasi algoritma *Support Vector Regressor* dengan pendekatan *Natural Language Processing.* Hasil performa model dari implementasi menggunakan metode ini dilakukan menggunakan salah satu *metrics performance model* untuk kasus regresi, yaitu *Mean Squared Error* (MSE).

$$MSE = \frac{1}{n} \sum (y - \hat{y})^2 \quad (6)$$

Data yang telah melalui proses *preprocessing* dan dianggap telah layak untuk diproses lebih lanjut, kemudian dilatih oleh algoritma *machine learning*. Sebelum proses pelatihan dilakukan, penulis membagi data menjadi dua bagian, yaitu data uji dan data latih

dengan pembagian 20%-80%. Pembagian ini bertujuan agar model dapat dievaluasi pada data yang belum pernah dilihat sebelumnya. Idealnya, model yang baik harus memiliki performa yang konsisten baik pada data latih maupun data uji. Jika model menunjukkan performa yang sangat baik pada data latih namun buruk pada data uji, maka kondisi ini dapat dianggap sebagai kondisi *overfitting*. Kondisi ini mengacu pada model yang terlalu "terkunci" pada pola spesifik dari data latih yang tidak mampu beradaptasi dengan data baru. Sebaliknya, jika model menunjukkan performa yang buruk pada data latih dan data uji, maka hal ini merupakan indikasi terjadinya kondisi *underfitting*. *Underfitting* terjadi ketika model terlalu sederhana dan tidak mampu menangkap pola-pola yang relevan dalam data. Dalam kondisi ini, model gagal belajar dari data latih yang mengakibatkan performa buruk [14].

Dengan demikian, penulis dapat mengukur seberapa baik model dapat menggeneralisasi pola dari data latih ke data baru. Untuk dapat menghindari kedua kondisi di atas (*overfitting* atau *underfitting*) dilakukan beberapa pendekatan yang disesuaikan dengan karakteristik data. Pada penelitian ini penghindaran pada kondisi *underfitting* dilakukan pada tahap *preprocessing* dan pada tahap *modeling*. Pada tahap *preprocessing* dilakukan beberapa tahap seperti pembuatan fitur baru dan beberapa penyesuaian lainnya. Adapun pada tahap *modeling* penulis melakukan *tuning* pada *hyperparameter* algoritma *support vector regressor*. *Tuning hyperparameter* dilakukan dengan menggunakan teknik *Grid Search* untuk meningkatkan performa model. *Grid Search* adalah metode pencarian yang sistematis dan menyeluruh untuk menentukan kombinasi terbaik dari parameter model yang memberikan hasil performa optimal [15].

Pada tahap ini, beberapa parameter penting seperti C, epsilon, dan kernel divariasikan dalam rentang nilai tertentu untuk menemukan kombinasi yang menghasilkan model dengan performa terbaik. *Grid Search* menggunakan *cross-validation* untuk mengevaluasi setiap kombinasi parameter dan memilih yang terbaik berdasarkan metrik evaluasi [16].

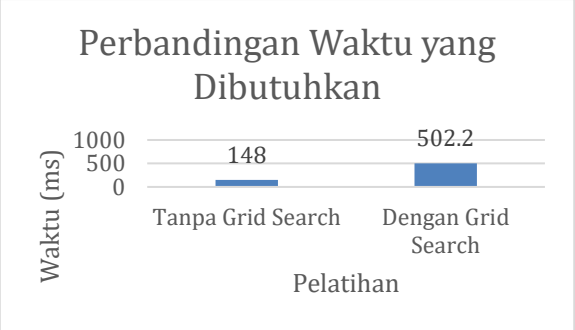
Untuk memastikan seberapa berpengaruh algoritma *Grid Search* dalam *tuning hyperparameter*, penulis melakukan perbandingan pada performa model sebelum dan setelah menggunakan algoritma ini. Penulis pertama – tama melatih model menggunakan *hyperparameter default* yang kemudian dilanjutkan dengan pelatihan menggunakan algoritma *Grid Search* untuk menentukan *hyperparameter* optimal. Setelah itu, penulis mencatat hasil performa model dari kedua pelatihan tersebut dan membandingkannya untuk mengidentifikasi peningkatan yang terjadi. Perbandingan ini memberikan wawasan mengenai seberapa besar pengatuh *tuning* terhadap kemampuan model dalam menangani data. Selain itu, analisis juga dilakukan untuk menilai *trade-off* antara waktu komputasi yang dibutuhkan oleh *Grid Search* dengan meningkatan performa yang diperoleh, sehingga dapat dievaluasi apakah metode *tuning* ini layak diterapkan.

Hasil yang ditampilkan menunjukkan perbedaan antara sebelum dan setelah menggunakan *Grid Search* sebagai algoritma *tuning hyperparameter*. Dari table 1, dapat diketahui bahwa sebelum menggunakan *Grid Search*, rata -rata *mean squared error* yang tercatat adalah 0.0393. Sementara, setelah menerapkan *Grid Search* dalam menentukan konfigurasi *hyperparameter* terbaik, rata – rata MSE menurun menjadi 0.0325. Penurunan ini mengindikasikan bahwa model dengan *tuning hyperparameter* menggunakan *grid search* lebih akurat dalam memprediksi nilai target, sehingga kesalahan prediksi berkurang secara signifikan.

Selain itu, dilakukan pula analisa waktu pelatihan data untuk memahami pengaruh penggunaan *grid search* terhadap efisiensi proses pelatihan model. Gambar 3 merupakan visualisasi dari perbandingan waktu pelatihan antara model tanpa menggunakan *grid search* dan model yang menggunakan *grid search* untuk *tuning hyperparameter.* Waktu pelatihan tanpa *grid search* tercatat sebesar 148 ms. Sementara, untuk pelatihan menggunakan *grid search* membutuhkan waktu 502.2ms. Peningkatan waktu ini disebabkan oleh proses pencarian *hyperparameter* yang melibatkan evaluasi berbagai kombinasi parameter untuk menemukan konfigurasi yang terbaik.

Tabel 1. Perbandingan Hasil Sebelum dan Setelah menggunakan *Grid Search*

y	0.5	0.8	1	1	0.4	1	0.8	1	1	0.8	1	1	1	0.1	0.9	0.3	0	0.9	0.8	0.1	1	1	1	0.3	0.8	0.4	0.7	0.9	0.8	1	1	0.9	0.5	1	0.6	1	0	0.9	1	0.1	1	0.6	0.8	0.8
ŷ	0.4	0.7	1	1	0.5	0.8	0.7	1	1	0.7	0.6	0.8	0.7	0.4	0.8	0.4	0	0.6	0.7	0.7	0.7	1	0.8	0.5	0.7	0.5	0.6	0.8	0.7	1	1	0.6	0.7	0.8	0.5	1	0.4	0.7	1	0.4	1	0.7	0.6	0.6
e	0	0	0	0	0	0	0	0	0	0	0.2	0	0.1	0.1	0	0	0	0.1	0	0.4	0.1	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0.2	0	0	0.1	0	0	0	0
Rata - Rata MSE Sebelum Menggunakan Grid Search : 0.039318182																																												
y	0.5	0.8	1	1	0.4	1	0.8	1	1	0.8	1	1	1	0.1	0.9	0.3	0	0.9	0.8	0.1	1	1	1	0.3	0.8	0.4	0.7	0.9	0.8	1	1	0.9	0.5	1	0.6	1	0	0.9	1	0.1	1	0.6	0.8	0.8
ŷ	0.4	0.8	1	1	0.5	0.9	0.7	1	1	0.8	0.6	0.9	0.8	0.4	0.9	0.4	0	0.6	0.8	0.7	0.8	1	0.9	0.4	0.7	0.5	0.6	0.9	0.8	1	1	0.7	0.7	0.8	0.5	1	0.4	0.8	1	0.4	1	0.7	0.5	0.7
e	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.1	0	0	0	0.1	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.1	0	0	0.1	0
Rata - Rata MSE Setelah Menggunakan Grid Search : 0.0325																																												

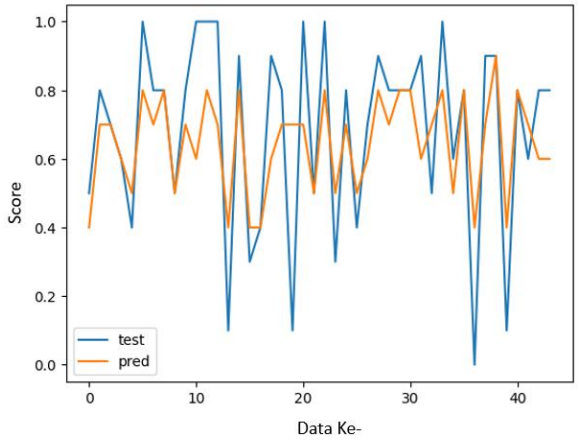


Gambar 3. Visualisasi Perbandingan Waktu Pelatihan Tanpa dan Dengan *Grid Search*

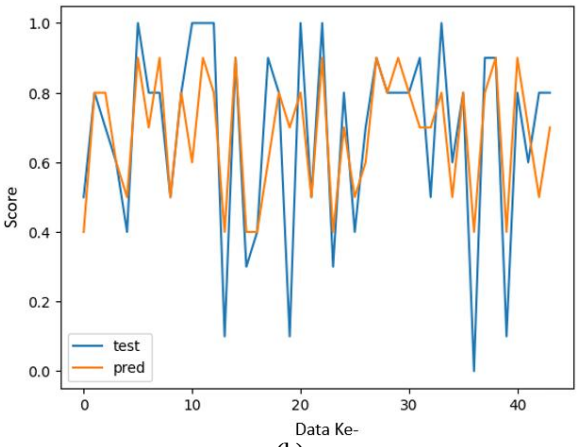
Penurunan MSE pada tabel 1 menunjukkan peningkatan performa model dalam memprediksi nilai target dengan tingkat kesalahan yang lebih kecil. Hal ini juga dapat dilihat dari Gambar 4a dan 4b yang merupakan visualisasi perbandingan dari nilai aktual dan nilai hasil prediksi yang dihasilkan model sebelum dan setelah menggunakan grid search. Gambar 4b memiliki jarak garis orange dan biru yang lebih rapat jika dibandingkan dengan gambar 4a, di mana garis warna orange merepresentasikan nilai hasil prediksi model dan garis biru merupakan representasi dari nilai aktual. Penurunan jarak ini menunjukkan bahwa model yang telah dioptimalkan dengan grid search lebih mampu menghasilkan prediksi yang mendekati nilai aktual, mengkonfirmasi efektivitas grid search dalam meningkatkan akurasi model. Perubahan visual ini juga memperjelas dampak positif dari tuning hyperparameter terhadap performa model.

Gambar 3 menunjukkan adanya peningkatkan penggunaan waktu pelatihan setelah menggunakan grid search sebanyak kurang lebih tiga kali lipat. Namun, meskipun demikian, hasilnya sepadan karena model menjadi lebih akurat dan andal. Hal ini juga didukung dengan peningkatan waktu komputasi yang masih dapat dianggap wajar, meskipun durasi pelatihan meningkat hingga tiga kali lipat. Waktu pelatihan yang dibutuhkan sebenarnya hanya sekitar 8 detik, yang tetap tergolong cepat dan efisien dalam konteks pelatihan model menggunakan algoritma machine learning. Angka ini menunjukkan bahwa meskipun terjadi peningkatan durasi pelatihan, proses tersebut masih dalam batas yang dapat diterima untuk memastikan optimasi performa model. Dalam konteks aplikasi yang membutuhkan prediksi dengan kualitas tinggi seperti pada penelitian ini, tambahan waktu pelatihan dengan peningkatan performa model merupakan usaha yang layak dilakukan. Sehingganya dapat dilakukan penilaian essai otomatis dengan nilai yang mendekati nilai aktual. Di sisi lain, waktu pelatihan yang relatif singkat menunjukkan efisiensi model dalam hal komputasi yang menjadikannya cocok diaplikasikan untuk memberikan penilaian yang konsisten dan objektif yang dapat meningkatkan efisiensi proses evaluasi.

Namun, masih tetap dibutuhkan evaluasi lebih lanjut untuk memastikan bahwa model ini berfungsi dengan baik pada berbagai jenis essay dan konteks penilaian yang berbeda. Selain itu, perlu juga dilakukan pemantauan untuk menjaga akurasi dan relevansi dalam aplikasi yang lebih luas. Penelitian lebih lanjut bisa berfokus pada eksplorasi lebih lanjut dapat berfokus pada pengembangan metode evaluasi yang dapat mengadaptasi berbagai jenis essay dan konteks penilaian, serta menilai dampak dari penggunaan model ini dalam situasi yang bervariasi.



(a)



(b)

Gambar 4. Perbandingan Hasil Prediksi dari (a) Pelatihan Tanpa *Grid Search* dan (b) Dengan *Grid Search*

4. KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa penggunaan algoritma support vector regressor dengan grid search untuk penilaian essay otomatis dapat meningkatkan akurasi prediksi dengan metric MSE 0.0325 dengan waktu pelatihan 502.2 ms. Hasil ini membuktikan bahwa algoritma SVR dengan tuning hyperparameter dengan grid search merupakan metode yang efektif dalam memberikan penilaian essay yang mendekati nilai aktualnya. Sehingganya, penelitian ini secara keseluruhan dapat memberikan kontribusi dalam bidang penialaian otomatis dengan solusi yang efektif dan efisien untuk menurunkan tingkat kesalahan dalam penialaian essay otomatis. Penelitian ini membuktikan bahwa penggunaan algoritma Support Vector Regressor (SVR) dengan tuning hyperparameter menggunakan Grid Search dapat

meningkatkan akurasi prediksi dalam sistem penilaian esai otomatis. Hal ini ditunjukkan dengan penurunan nilai Mean Squared Error (MSE) dari 0.0393 menjadi 0.0325. Meskipun waktu pelatihan meningkat dari 148 ms menjadi 502.2 ms, peningkatan akurasi yang diperoleh menunjukkan bahwa metode ini efektif dan sebanding dengan tambahan waktu komputasi yang dibutuhkan. Hasil penelitian ini memberikan kontribusi pada pengembangan sistem penilaian otomatis yang lebih akurat dan efisien, sehingga dapat mengurangi subjektivitas dan mempercepat proses evaluasi esai. Namun, penelitian ini masih memiliki keterbatasan dalam hal generalisasi model terhadap berbagai jenis esai. Oleh karena itu, penelitian lanjutan dapat difokuskan pada evaluasi model terhadap dataset yang lebih beragam serta eksplorasi metode optimasi tambahan guna meningkatkan performa sistem lebih lanjut.

DAFTAR PUSTAKA

[1] D. Ramesh and S. K. Sanampudi, "An automated essay scoring systems: a systematic literature review," *Artif Intell Rev*, vol. 55, no. 3, pp. 2495–2527, Mar. 2022, doi: 10.1007/s10462-021-10068-2.

[2] C. T Angel, V. H. Pranatawijaya, and Widiatry, "Analisis Sentimen dan Emosi dari Ulasan Google Maps untuk Layanan Rumah Sakit di Palangka Raya Menggunakan Machine Learning," *Jurnal KONSTELASI : Konversi Teknologi dan Sistem Informasi*, vol.4, no.1, 2024.

[3] Q. I. Susilowati, and Z. Y. Acar, "Essay Scoring for Essay Evaluation Using Support Vector Machine in predicting Youth Break the Boundaries Scholarship Applicants", *Journal Elektronik Sistem Informasi (JESII) Universitas Kebangsaan Republik Indonesia (UKRI)*, vol.1, no.2, 2023, doi: 10.31848/jesii.xxxx.xxxx.

[4] R. A. Rajagede, "Improving Automatic Essay Scoring for Indonesian Language using Simpler Model and Richer Feature," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 11–18, Feb. 2021, doi: 10.22219/kinetik.v6i1.1196.

[5] C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi," *Technologia : Jurnal Ilmiah*, vol. 15, no. 1, p. 93, Jan. 2024, doi: 10.31602/tji.v15i1.13457.

[6] Sembiring, A. Iriani, J. Veronika Br Ginting, and J. A. Ginting, "A Novel Approach to Network Forensic Analysis: Combining Packet Capture Data and Social Network Analysis.", *International Journal of Advanced Computer Science and Aplications*, vol. 14, n0.3, 2023, [Online]. Available: www.ijacsa.thesai.org

[7] N. Azizah and A. F. Rozi, "Sistem Rekomendasi Produk Somethinc Menggunakan Metode Content-based Filtering," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 3, pp. 461–468, Jul. 2024, doi: 10.47233/jteksis.v6i3.1411.

[8] R. A. Rahman, V. H. Pranatawijaya, and N. N. K. Sari, "Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Gojek" , *Jurnal KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 4, no.1, 2024.

[9] R. A. Fitrianto and A. S. Editya, "Klasifikasi Tweet Sarkasme Pada Platform X Menggunakan Bidirectional Encoder Representations from Transformers," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 3, pp. 366–371, Jul. 2024

[10] D. J. Ladani and N. P. Desai, "Stopword Identification and Removal Techniques on TC and IR Applications: A Survey", *International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2020.

[11] R. Saputra and F. N. Hasan, "Analisis Sentimen Terhadap Program Makan Siang & Susu Gratis Menggunakan Algoritma Naive Bayes," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 3, pp. 411–419, Jul. 2024, doi: 10.47233/jteksis.v6i3.1378.

[12] F. Sesilia, V. H. Pranatawijaya, and R. Priskila, "Machine Learning untuk Memprediksi Jumlah Penjualan, Stok dan Jumlah Tanam Hasil Pertanian Hidroponik", *Jurnal KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 4, no.1, 2024.

[13] F. Feiters Tampinongkol, C. Herdian, H. Basri, and L. Halim, "Identifikasi Penyakit Daun Tomat Menggunakan Gray Level Co-occurrence Matrix (GLCM) dan Support Vector Machine (SVM).", *Techno Xplore Jurnal Ilmu Komputer dan Teknologi Informasi*, 2023, [Online]. Available: https://www.kaggle.com/

[14] W. A. Firmansyach, U. Hayati, and Y. A. Wijaya, "Analisa Terjadinya Overfitting Dan Underfitting Pada Algoritma Naive Bayes Dan Decision Tree Dengan Teknik Cross Validation,", *Jurnal Mahasiswa Teknik Informatika*, vol.7, no.1, 2023.

[15] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *International Journal of Computers and Applications*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.

[16] W. Nugraha and A. Sasongko, "Hyperparameter Tuning on Classification Algorithm with Grid Search," *SISTEMASI*, vol. 11, no. 2, p. 391, May 2022, doi: 10.32520/stmsi.v11i2.1750.