

STUDI KOMPARASI ALGORITMA KLASIFIKASI MENTAL WORKLOAD BERDASARKAN SINYAL EEG

Dessy Kusumaningrum
Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Surabaya
dessykusumaningrum16030214016@mhs.unesa.ac.id

Elly Matul Imah
Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Surabaya
*ellymatul@unesa.ac.id

Abstrak—Kondisi psikologis dan fisik manusia dapat memengaruhi proses berpikir. Apabila kondisi individu mengalami kelelahan, maka dapat memengaruhi penurunan tingkat produktivitas maupun penurunan proses berpikir yang menyebabkan timbulnya *mental workload*. *Workload* yang dimiliki harus seimbang terhadap kemampuan dan keterbatasan yang dimiliki. *Mental workload* yang berlebih berdampak buruk bagi individu karena menimbulkan penurunan produktivitas kerja. Perangkat khusus yang dapat digunakan untuk mengetahui tingkat *mental workload* seorang individu adalah Elektroensefalogram (EEG). EEG adalah perangkat khusus yang digunakan untuk mengukur sinyal potensi listrik dari otak. Dataset yang digunakan dalam penelitian ini adalah STEW: Simultaneous Task EEG Dataset dengan 45 subjek. Dalam penelitian ini, telah dilakukan studi komparasi algoritma Random Forest, K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP), dan Support Vector Machine (SVM) untuk klasifikasi *mental workload* berdasarkan sinyal EEG. Studi dilakukan untuk menentukan algoritma terbaik dalam klasifikasi dilihat dari segi nilai akurasi dan penggunaan memori saat proses klasifikasi. Manfaat dari studi komparasi ini adalah untuk mengetahui algoritma terbaik dalam mengklasifikasikan *mental workload* dari sisi nilai akurasi, waktu pembuatan model, dan penggunaan memori. Dataset telah melalui beberapa tahapan, diantaranya pra-pemrosesan data, ekstraksi fitur, dan proses klasifikasi. Pra-pemrosesan data menerapkan pembagian data menjadi beberapa *chunk*. Untuk mendapatkan ciri dalam ekstraksi fitur, diterapkan metode Principal Component Analysis (PCA). Pada proses klasifikasi menggunakan pendekatan *10-fold cross validation*. Hasil studi penelitian ini adalah algoritma terbaik dari sisi akurasi adalah algoritma KNN, algoritma terbaik dari sisi waktu uji dan memory adalah algoritma Random Forest.

Keywords—*Mental Workload, Random Forest, K-Nearest Neighbor, Multi-Layer Perceptron, Support Vector Machine*

I. PENDAHULUAN

Kapasitas berpikir yang dimiliki manusia memiliki kuantitas dan kualitas yang berbeda-beda pada tiap individu. Kondisi psikologis dan fisik manusia dapat memengaruhi proses berpikir. Apabila kondisi individu mengalami kelelahan, maka dapat memengaruhi penurunan tingkat produktivitas maupun penurunan proses berpikir. Kelelahan yang dialami tersebut menjadi suatu penyebab adanya beban kerja mental atau *mental workload*. *Mental workload* dijelaskan sebagai kebutuhan tugas dalam hal kesulitan, kompleksitas, dan tekanan temporal, atau upaya yang dilakukan untuk memenuhi persyaratan pekerjaan [1]. Setiap *workload* yang diterima tiap manusia hendaknya setara antara kemampuan fisik, kemampuan kognitif maupun keterbatasan

manusia dalam memperoleh beban tersebut dilihat dari sudut pandang ergonomi [2]. *Mental workload* yang berlebih dapat berdampak buruk bagi individu. Semakin besar *mental workload* pada individu, semakin tinggi pula kemungkinan risiko kelelahan yang dapat berpengaruh pada penurunan produktivitas kerja [3]. Salah satu perangkat khusus yang dapat dimanfaatkan untuk mengetahui tingkat *mental workload* seorang individu adalah Elektroensefalogram (EEG).

EEG adalah perangkat khusus yang digunakan untuk mengukur potensi listrik dari otak [4] dan sindrom elektrokimia spesifik [5] dalam bentuk sinyal. Sinyal EEG juga dapat digunakan untuk klasifikasi kondisi mental. Salah satunya, riset yang dilakukan Singla, dkk. (2019) bertujuan untuk mengetahui performa manusia di bawah mental workload berdasarkan sinyal EEG dengan menggunakan beberapa fungsi kernel *Support Vector Machine* (SVM) yang berbeda. Hasil riset tersebut menunjukkan bahwa *Cubic SVM* memiliki hasil akurasi terbaik dibandingkan dengan fungsi yang lain [6]. Kemudian riset yang dilakukan Bird, dkk. (2018) bertujuan mengklasifikasikan tingkat mental berdasarkan sinyal EEG dengan algoritma Naïve Bayes, J48, Random Tree, Random Forest, MLP, dan SVM. Hasil dari penelitian tersebut menyatakan bahwa algoritma Random Forest merupakan algoritma yang paling akurat dibandingkan dengan algoritma lainnya karena memiliki akurasi prediksi terbaik [7]. Penelitian tentang klasifikasi mental juga pernah dilakukan Lobo, dkk. (2016). Dalam penelitiannya bertujuan menyelidiki metode untuk klasifikasi berbagai tingkat *cognitive workload* mulai dari EEG yang disinkronkan dan informasi pelacakan mata [8].

Penelitian terdahulu mengenai estimasi *mental workload* menggunakan sinyal EEG yaitu penelitian yang dilakukan Appriou, dkk. (2018) menggunakan metode berbasis geometri *Riemannian* dan *Deep Learning*. Penelitian mereka mempelajari menggunakan dua metode untuk memperkirakan *mental workload* dari sinyal EEG [9]. Selain itu, penelitian oleh Roy, dkk. (2016) membandingkan dua tipe EEG klasik, *spectral markers* dengan *event-related potentials* (ERP) untuk estimasi *mental workload* dengan implementasi kehidupan nyata. Hasil penelitian tersebut adalah ERP memungkinkan estimasi *mental workload* yang lebih efisien daripada *spectral markers* yang mendekati implementasi kehidupan nyata [10].

Dalam penelitian ini akan dilakukan perbandingan algoritma Random Forest, *K-Nearest Neighbor* (KNN), *Multi-Layer Perceptron* (MLP), dan *Support Vector Machine* (SVM) untuk klasifikasi *mental workload* berdasarkan sinyal EEG. Alasan penggunaan algoritma Random Forest dipilih karena mampu mengklasifikasikan banyak data. Dimana Random Forest terdiri dari banyak pohon keputusan serta memiliki kesalahan klasifikasi yang rendah dibandingkan dengan algoritma tradisional lainnya [11]. Seperti metode berbasis pohon lainnya, Random Forest dapat menangani kumpulan data dalam jumlah besar dengan banyak prediktor menggunakan pemilihan faktor otomatis dan tanpa penghapusan faktor [12].

Algoritma KNN dipilih karena tergolong pada mesin pembelajaran *supervised* dalam pemecahan masalah klasifikasi. Penelitian yang telah dilakukan [13]–[15] menggunakan algoritma KNN dan menunjukkan algoritma tersebut dapat mengklasifikasi dengan baik. Selain itu, KNN dapat mempelajari tugas-tugas kompleks menggunakan prosedur sederhana dengan pendekatan lokal [16]. Sedangkan algoritma MLP merupakan model jaringan saraf tiruan *feedforward* yang dilatih dengan *stochastic gradient descent* (SGD) menggunakan backpropagation [17] yang memetakan himpunan data *input* ke dalam himpunan *output* yang sesuai [18]. Penelitian terdahulu yang dilakukan [19], [20] menunjukkan bahwa performa terbaik diberikan oleh algoritma MLP.

SVM termasuk dalam kategori algoritma populer dalam masalah pemecahan klasifikasi. Data berdimensi tinggi dapat diklasifikasikan dengan baik menggunakan algoritma ini. Percobaan mengenai SVM telah terbukti memberikan hasil prediksi dengan akurasi tertinggi [10]. SVM memiliki *hyperplane* yang memisahkan dua target dengan memaksimalkan margin [11]. Penelitian [21], [22] menggunakan algoritma SVM sebagai

classifier dan menunjukkan performa terbaik. Sehingga, algoritma SVM dipilih sebagai salah satu algoritma klasifikasi dalam penelitian ini.

Tujuan dilakukan studi komparasi ini diharapkan dapat menentukan algoritma terbaik untuk klasifikasi *mental workload* berdasarkan sinyal EEG. Komparasi dalam studi ini, membandingkan performa algoritma satu dengan lainnya dilihat dari segi nilai akurasi yang dihasilkan serta penggunaan memori pada tiap algoritma. Dalam penelitian ini menerapkan Analisis Komponen Utama (PCA) untuk ekstraksi fitur. STEW Dataset [23] dengan 45 subjek merupakan dataset utama dalam penelitian ini.

II. METODE

Sebelum melakukan penelitian, peneliti mengunduh dataset. Dataset yang telah diunduh kemudian melalui proses pra-pemrosesan data, ekstraksi fitur, dan proses klasifikasi. Dalam tahap pra-pemrosesan data dan ekstraksi fitur menggunakan *software* Matlab R2015b sedangkan dalam proses klasifikasi menggunakan *software* Weka 3.9. Proses klasifikasi menggunakan empat algoritma yaitu Random Forest, KNN, MLP, dan SVM. Selanjutnya, dilakukan pengambilan nilai akurasi dari masing-masing algoritma kemudian dilakukan perbandingan.

A. Pra-Pemrosesan Data

Pada tahap pra-pemrosesan data, 1 kanal diambil dari data awal yang mewakili beban kerja mental subjek. Rekaman sinyal EEG terdiri dari beberapa kanal yaitu AF3-AF4, F3-F4, F7-F8, FC5-FC6, O1-O2, P7-P8, T7-T8. Kanal sinyal EEG yang berada pada daerah frontal AF3 merupakan kanal yang digunakan pada penelitian ini. AF3 dipilih karena merupakan kanal yang memiliki persentase akurasi paling tinggi untuk klasifikasi *cognitive workload* atau yang biasa dikenal *mental workload* [24]. Setiap subjek memiliki dua data dalam kondisi santai dan saat melakukan tes. Dalam 1 kanal sinyal EEG terdapat 19200 data numerik. Data yang dimiliki tiap subjek berjumlah 19200 data numerik dan satu data berupa label. Data kemudian digabung antara data saat santai dan saat melakukan tes dengan subjek-subjek lain, sehingga diperoleh data sebanyak 90 data dan 19201 atribut. Setelah data baru berupa data penggabungan seluruh subjek, maka dilakukan pembagian data dengan membagi jumlah kolom dengan 300 yang berguna untuk mendapatkan akurasi sinyal yang lebih detail. Dengan membagi jumlah kolom dengan 300 sehingga didapatkan 64 *chunk* dimana matriks data setiap *chunk* adalah 90×300 . Penggabungan 64 *chunk* memperoleh hasil matriks data berjumlah 5760×300 serta satu data berupa label.

B. Ekstraksi Fitur

Data yang telah melalui tahap pra-pemrosesan, selanjutnya melalui proses ekstraksi fitur menggunakan transformasi *Principal Component Analysis* (PCA). Ekstraksi fitur membentuk himpunan data baru yang merupakan hasil turunan dari himpunan data awal dengan ukuran yang lebih kecil [25]. Tujuan PCA adalah mereduksi data dengan memproyeksikan secara geometris ke dimensi yang lebih rendah yang disebut komponen utama (PC) [26], dengan mempertahankan sebanyak mungkin variasi yang ada dalam data [27]. Pada masing-masing data diambil cirinya dengan menghitung fitur komponen utama (PC). Asumsi dasar dalam penggunaan PCA adalah bahwa vektor eigen dan pemuatan yang sesuai dengan nilai eigen terbesar berisi informasi yang paling berguna terkait dengan masalah spesifik, dan yang lainnya terutama terdiri dari *noise* [28]. Oleh karena itu, vektor-vektor eigen diurutkan dari nilai eigen terbesar sampai dengan nilai eigen terkecil. Berikut ini adalah tahapan dari algoritma PCA :

- Langkah 1 : Menentukan rata-rata atau *mean* (μ) dari suatu matriks data
- Langkah 2 : Menentukan matriks kovarian (K)
- Langkah 3 : Menghitung nilai eigen (λ) dan vektor eigen (v_j) dari K
- Langkah 4 : Mengurutkan vektor eigen dengan urutan menurun yang terkait dari nilai eigen terbesar.

Pada penelitian ini, PC yang dihasilkan pada transformasi PCA digunakan sebagai fitur yang digunakan pada tahap klasifikasi. Hasil ekstraksi fitur dengan menerapkan tahapan algoritma PCA menghasilkan data sejumlah 5760 data dan 23 atribut atau secara matriks dapat dituliskan 5760×23 serta satu data berupa label. Dimana kolom pada matriks tersebut merepresentasikan fitur.

C. Proses Klasifikasi

Dalam proses klasifikasi, algoritma-algoritma yang digunakan dalam proses klasifikasi dijelaskan sebagai berikut.

1) *Random Forest*

Random Forest termasuk dalam masalah pembelajaran *supervised*. Random Forest meningkatkan keragaman antara pohon klasifikasi dengan melakukan sampling ulang data dengan penggantian dan secara acak mengubah himpunan variabel prediksi atas berbagai cara induksi pohon [12]. Algoritma ini mengambil atribut serta data secara acak sesuai ketentuan yang berlaku dalam pembangunan pohon keputusan [29]. Pohon keputusan tersebut tersusun atas beberapa simpul, yaitu simpul akar, simpul internal, dan simpul daun.

Tipe-tipe simpul dijelaskan sebagai berikut [30]:

a) *Simpul Akar*

Simpul akar tidak memiliki *input* serta mempunyai nol atau lebih *output*.

b) *Simpul Internal*

Setiap simpul internal mempunyai tepat satu *input* serta mempunyai minimal dua *output*.

c) *Simpul Daun (Simpul Terminal)*

Setiap simpul daun mempunyai tepat satu *input* serta tidak memiliki *output*. Simpul daun menyatakan label kelas.

Perhitungan nilai *entropy* untuk data latih menggunakan persamaan (1), berguna dalam menentukan node sebagai pemecah data latih berikutnya [31].

$$E(y) = - \sum_{i=1}^n p(w|y) \log_2 p(w|y) \tag{1}$$

Dimana $p(w|y)$ adalah semua data latih yang diproses di node y , sedangkan untuk memilih fitur yang digunakan sebagai pemecah node menggunakan persamaan Gain (2) [31].

$$G(y) = E(y) - \sum_{i=1}^n p(v|y) \times E(v) \tag{2}$$

Dimana $p(v|y)$ adalah seluruh nilai pada kelas simpul, $E(v)$ adalah entropi komposisi dari nilai v , dan n adalah banyak nilai berlainan dalam simpul.

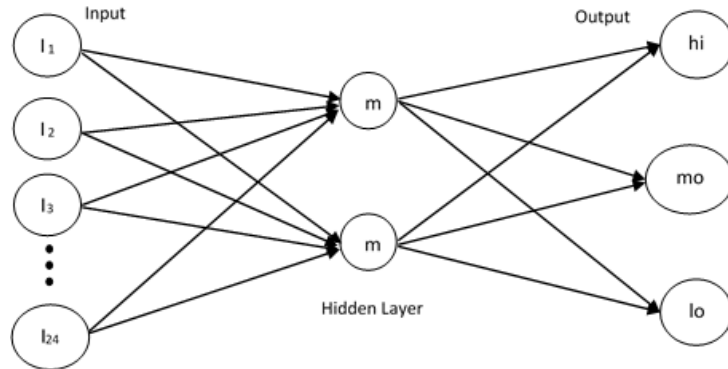
2) *K-Nearest Neighbor (KNN)*

Algoritma KNN dikenal dengan algoritma *lazy learning* karena membutuhkan waktu pelatihan yang singkat. Pola *input* algoritma ini diklasifikasikan oleh suara terbanyak dari tetangganya. Nilai dari jumlah tetangga terdekat yang disertakan dalam menentukan estimasi kelas pada data uji disimbolkan dengan K [31]. KNN menghitung jarak antara dua titik tupel. Jarak kedua tupel tersebut dapat dihitung menggunakan jarak Euclidean. Euclidean dipilih karena kesederhanaan dan interpretasi geometris yang baik [32]. Misal, $X_1 = (x_{11}, x_{12}, \dots, x_{1n})$ dan $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$ maka jarak Euclidean dapat dihitung menggunakan persamaan (3) [32].

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \tag{3}$$

3) *Multi-Layer Perceptron (MLP)*

Penggunaan algoritma *neural network* sudah sejak lama digunakan, salah satunya yaitu *Multi-Layer Perceptron (MLP)*. Struktur penyusun MLP terdiri dari tiga layer berbeda. Layer pertama adalah layer *input*. Layer tersebut berfungsi sebagai penerima vektor *input* dan mendistribusikan ke semua neuron dalam layer *hidden* [31]. Layer *hidden* merupakan layer kedua yang berjumlah satu atau lebih dari satu neuron [17]. Layer terakhir adalah layer *output* yang menerima sinyal keluaran dari layer *hidden* dan menampilkan nilai atau kelas *output* dari keseluruhan jaringan [31]. Arsitektur MLP ditampilkan pada Gambar 1.



Gambar 1. Arsitektur Multi-Layer Perceptron

Dalam penelitian ini, terdapat tiga kelas *output* dengan *input* yang digunakan berasal dari 23 atribut yang didapat dari fitur PCA serta *hidden layer* dimisalkan dengan *m* seperti pada Gambar 1. Perhitungan *output* pada tiap *node* dalam jaringan dapat dilakukan dalam dua tahap. Tahap perhitungan pertama adalah penjumlahan bobot dari *input* dapat dihitung dengan persamaan (6) di mana I_i adalah variabel *input* i , dan w_{ij} adalah bobot koneksi antara I_i dan neuron j dalam layer *hidden* [33].

$$S_j = \sum_{i=1}^n w_{ij} I_i + \beta_j \tag{4}$$

Perhitungan kedua yaitu fungsi aktivasi yang digunakan untuk memicu *output* neuron f_j berdasarkan nilai fungsi penjumlahan, *output* dari node j dalam layer *hidden* dapat ditulis dalam persamaan (7).

$$f_j(x) = \frac{1}{1 + e^{-S_j}} \tag{5}$$

Setelah perhitungan *output* pada tiap neuron dalam layer *hidden*, *output* akhir dari keseluruhan jaringan dapat dihitung dengan persamaan (8).

$$\hat{y}_k = \sum_{i=1}^m w_{kj} f_i + \beta_k \tag{6}$$

4) *Support Vector Machine (SVM)*

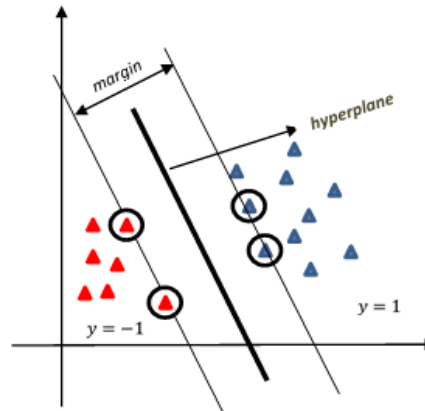
SVM termasuk algoritma *supervised learning* dalam pemecahan masalah tentang klasifikasi. Model SVM adalah alat yang berguna untuk mengidentifikasi model prediksi atau pengklasifikasi, tidak hanya karena dapat mengakomodasi data yang sedikit tetapi juga dapat mengklasifikasikan grup atau membuat aturan prediksi untuk data yang tidak diklasifikasikan secara linier [34]. Konsep SVM adalah mendapatkan *hyperplane* terbaik dengan memaksimalkan margin [31]. *Hyperplane* optimal merupakan *hyperplane* yang berada pada pertengahan dua himpunan objek dari dua kelas berbeda [35]. SVM terdiri dari *support vector*, *hyperplane*, dan margin. *Support vector* merupakan titik-titik yang letaknya berdekatan dengan *hyperplane*. Visualisasi SVM tampak pada Gambar 2. Apabila titik tersebut berada di atas *hyperplane* maka memenuhi persamaan (9) [31] :

$$w \cdot x_a + b = +1 \tag{7}$$

w dan b adalah parameter model, $w \cdot x_a$ adalah *inner-product* w dengan x_a .

Sementara, jika titik-titik berada di bawah *hyperplane* maka memenuhi persamaan (10) [31] :

$$w \cdot x_b + b = -1 \tag{8}$$



Gambar 2. SVM dengan Hyperplane Terbaik

Persamaan (11) dan persamaan (12) secara berturut-turut menyatakan bahwa data yang letaknya berada di atas *hyperplane* dikategorikan sebagai kelas +1 dan data yang letaknya berada di bawah *hyperplane* masuk dalam kelas -1. Untuk mendapatkan margin terbaik dengan cara menghitung jarak antara titik terdekat dengan *hyperplane* dapat diringkas dengan mengurangi persamaan (9) dengan persamaan (10), sehingga didapatkan persamaan (11) :

$$w(x_a - x_b) = 2 \tag{9}$$

Dengan menyederhanakan persamaan (11) maka diperoleh persamaan untuk margin optimal seperti pada persamaan (12), dengan $\|w\|$ adalah vektor bobot w . Sementara margin ialah jarak antara *hyperplane* dengan titik terdekat kelas masing-masing [31].

$$d = \frac{2}{\|w\|} \tag{10}$$

III. HASIL DAN PEMBAHASAN

Pada penelitian ini menggunakan Processor Intel(R) Core i3 GHz CPU, 4GB RAM, serta sistem operasi Microsoft Windows 7 32-bit. Pendekatan *k-fold cross validation* diterapkan dalam klasifikasi dengan memilih nilai $k = 100$ atau dapat dituliskan *100-fold cross validation*. $k = 100$ dipilih supaya memperoleh hasil yang lebih baik karena jumlah data yang digunakan dalam klasifikasi berjumlah besar.

A. Dataset

Peneliti menggunakan dataset yang diunduh dari STEW: Simultaneous Task EEG Workload Dataset (<https://ieee-dataport.org/open-access/stew-simultaneous-task-eeg-workload-dataset>) yang terdiri dari 45 subjek laki-laki. Subjek diminta untuk melakukan modul uji kapasitas simultan (SIMKAP) dari Sistem Uji Vienna yang dirancang sebagai alat penilaian untuk menampilkan kemampuan *multitasking* tiap subjek di pekerjaan yang memiliki tingkat *multitasking* berat seperti manajemen lalu lintas udara [23]. STEW Dataset dipilih karena dataset ini menyajikan secara rinci data pada setiap kanal serta untuk menganalisis data jika diklasifikasikan menggunakan beberapa algoritma klasifikasi. Setiap subjek memiliki dua data dalam keadaan *mental workload* saat santai dan saat melakukan tes SIMKAP. Data EEG berupa sinyal mentah atau *raw signal* dengan frekuensi sampling 128Hz dan terdiri dari 14 kanal. Kanal tersebut adalah AF3,

F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. Dalam penelitian ini, label kelas untuk menyatakan *mental workload* rendah disimbolkan dengan 0 (lo), untuk *mental workload* moderat disimbolkan dengan 1 (mo), dan untuk menyatakan *mental workload* tinggi disimbolkan dengan 2 (hi).

B. Proses Akurasi Hasil Klasifikasi

Pada proses klasifikasi, terdapat empat algoritma klasifikasi yang digunakan di mana setiap algoritma memiliki kelebihan dan kekurangan sebagai klasifikator. Dalam klasifikasi, tidak dapat dipungkiri bahwa klasifikator tidak bekerja secara 100% benar. Oleh karena itu, perlu adanya alat ukur untuk mengukur performa klasifikasi menggunakan matriks *confusion*.

Matriks *confusion* adalah tabel dengan ukuran $m \times m$ dimana diagonal matriksnya merepresentasikan akurasi terbaik klasifikator. Nilai akurasi diperoleh dari mengetahui jumlah data yang benar terklasifikasikan. Semua algoritma klasifikasi berusaha untuk membentuk model dengan akurasi yang tinggi [31]. Perhitungan akurasi dapat menggunakan persamaan (13)

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad (11)$$

Dimana TP (Positif Benar) ialah jumlah data pada kelas A yang terprediksi secara benar pada kelas A. FP (Positif Salah) ialah jumlah data pada kelas B yang terprediksi secara benar pada kelas A. TN (Negatif Benar) ialah jumlah data pada kelas B yang terprediksi secara benar pada kelas B. Sedangkan FN (Negatif Salah) ialah jumlah data pada kelas A yang terprediksi secara benar pada kelas B.

Precision atau presisi merupakan hasil pembagian antara positif benar dengan jumlah data yang ditandai sebagai positif, sedangkan perbandingan antara data positif benar (TP) dengan jumlah data yang tergolong positif disebut dengan *recall* [31]. Persamaan (14) adalah persamaan yang digunakan untuk menghitung nilai presisi dan persamaan (15) adalah persamaan untuk menghitung *recall*.

$$Presisi = \frac{TP}{TP + FP} \quad (12)$$

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

Pengaruh dari nilai presisi dalam algoritma adalah nilai dari data yang benar pada tiap kelas dari keseluruhan data yang terprediksi pada tiap kelas. Sedangkan nilai *recall* merupakan nilai dari data yang terprediksi pada tiap kelas dibandingkan keseluruhan data yang sebenarnya dalam tiap kelas.

F-measure adalah nilai rata-rata harmonik terbobot di antara nilai *recall* dengan nilai presisi, nilai F-measure dihitung dengan persamaan (16) [31].

$$F = 2 \times \frac{precision \times recall}{precision + recall} \quad (14)$$

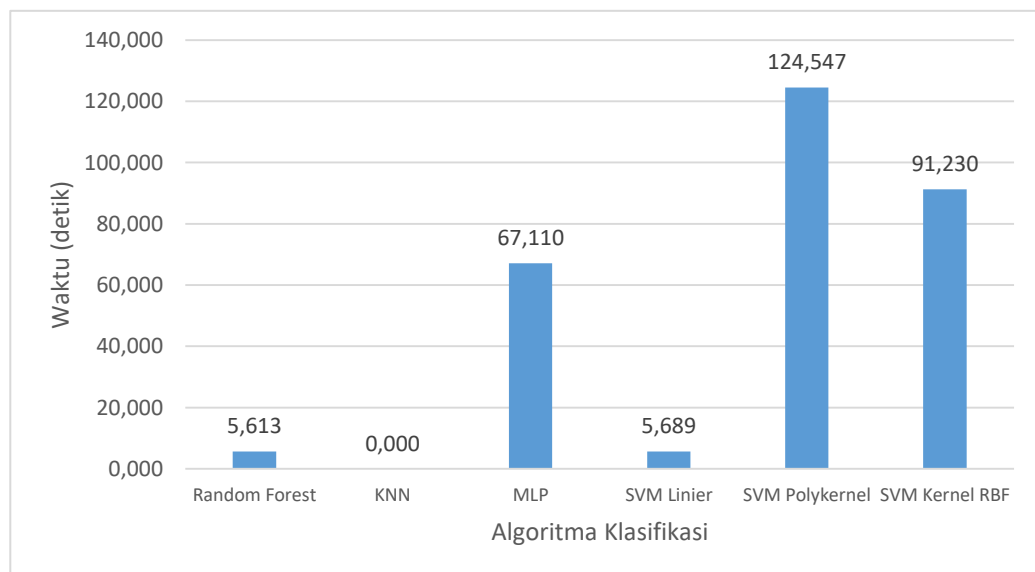
C. Hasil Akurasi Algoritma

Setiap algoritma yang digunakan terdapat beberapa parameter yang dikontrol. Pada algoritma Random Forest, parameter yang digunakan adalah *depth* 0, *depth* 5, *depth* 10. Parameter untuk algoritma KNN adalah mengubah nilai $k = 1, 5, \text{ dan } 20$. Untuk algoritma MLP, parameter yang digunakan adalah *learning rate* = 0.3, 0.6, dan 0.8 serta momentum = 0.2, 0.4, dan 0.6. Sedangkan untuk algoritma SVM Polykernel parameter yang digunakan adalah epsilon, $\epsilon = 10^{-3}, 10^{-6}, \text{ dan } 10^{-9}$ kemudian $C = 1, 2, \text{ dan } 4$ serta exponent atau derajat yang digunakan adalah 1 (Linier) dan 2. Algoritma SVM RBF menggunakan parameter Gamma, $\gamma = 0.5 \text{ dan } 1$ serta epsilon, $\epsilon = 10^{-3}, 10^{-6}, \text{ dan } 10^{-9}$. Hasil akurasi pada tiap algoritma pada Tabel 1, merupakan hasil akurasi rata-rata dari tiap-tiap parameter yang digunakan.

Tabel 1. Tabel Hasil Akurasi Algoritma

Algoritma	Nilai Rata-Rata Akurasi (%)	Nilai Rata-Rata Presisi	Nilai Rata-Rata Recall	Nilai Rata-Rata F-Measure
Random Forest	63.640 ± 12.794	0.661 ± 0.084	0.635 ± 0.128	0.579 ± 0.197
KNN	70.161 ± 9.004	0.710 ± 0.080	0.702 ± 0.090	0.704 ± 0.087
MLP	51.002 ± 1.035	0.486 ± 0.011	0.509 ± 0.010	0.485 ± 0.010
SVM Linier	46.077 ± 0.486	0.249 ± 0.189	0.460 ± 0.005	0.217 ± 0.163
SVM Polykernel	42.532 ± 3.849	0.408 ± 0.028	0.425 ± 0.038	0.355 ± 0.040
SVM Kernel RBF	45.968 ± 0.543	0.272 ± 0.211	0.460 ± 0.006	0.230 ± 0.178

Berdasarkan Tabel 1, algoritma klasifikasi yang memperoleh nilai akurasi tertinggi adalah algoritma KNN dengan nilai akurasi sebesar 70.161%. Nilai akurasi yang diperoleh algoritma ini lebih tinggi jika dibandingkan dengan algoritma klasifikasi lainnya. Sedangkan, algoritma yang memperoleh nilai akurasi terendah adalah algoritma SVM Polykernel dengan nilai akurasi sebesar 42.532%. Algoritma klasifikasi selain KNN yang memperoleh nilai akurasi lebih dari 50% adalah algoritma Random Forest dan MLP. Hal ini menunjukkan bahwa algoritma KNN memiliki performa yang terbaik dalam proses klasifikasi dari segi nilai akurasinya. Sedangkan algoritma SVM, baik SVM Linier, SVM Polykernel maupun SVM Kernel RBF memiliki nilai akurasi di bawah 50% yang dapat disimpulkan bahwa performa SVM dalam proses klasifikasi dari segi nilai akurasinya tidak baik.



Gambar 3. Grafik Waktu untuk Membangun Model

Dapat diambil kesimpulan melalui Gambar 3 bahwa algoritma yang memerlukan waktu lebih lama untuk membangun model adalah algoritma SVM Polykernel. Tetapi, algoritma KNN tidak membangun model dari data pelatihan sehingga tidak memerlukan waktu untuk membangun model yang mengakibatkan algoritma KNN membutuhkan ruang penyimpanan lebih banyak dikarenakan setiap data *input* dibandingkan dengan seluruh data latih. Dengan demikian, algoritma yang paling cepat untuk membangun model adalah algoritma Random Forest.

D. Pembahasan

Dalam penelitian ini berdasarkan ujicoba pada beberapa algoritma sebagaimana hasil pada Tabel 1 dan Gambar 3, bisa kita lihat bahwa akurasi tertinggi adalah KNN dan kedua Random Forest. Dari sisi pembuatan model KNN tidak memerlukan waktu sama sekali, karena algoritma ini tidak ada proses learning, hanya menyimpan semua sampel yang ada, kemudian saat ada data uji masuk, akan dilakukan pencocokan kemiripan dengan semua sampel yang ada. Yang paling mirip adalah kelas luarnya. Kekurangan algoritma KNN yang lain adalah meningkatkan kebutuhan ruang penyimpanan [37] serta akurasi prediksi dapat dipengaruhi kuat oleh data *noise* dan *outlier* [38].

Random Forest memerlukan waktu dalam pembuatan model, berbeda dengan KNN yang tidak memerlukan waktu sama sekali. Bisa dilihat juga pada hasil Gambar 3. Namun waktu pembuatan model Random Forest relatif lebih singkat karena algoritma ini bekerja menggunakan konsep tree, sehingga proses pencocokan lebih efisien meskipun dataset dalam jumlah besar, selama tidak terlalu banyak kelas. Alasan ini sesuai dengan [36], bahwa Random Forest algoritma yang akurat karena setiap pohon keputusan dapat membuat keputusan klasifikasi hanya berdasarkan fitur yang efektif.

Algoritma MLP dan SVM membutuhkan waktu pembuatan model lebih lama dibandingkan dengan algoritma Random Forest. Sedangkan KNN tidak memerlukan waktu dalam membangun model, tetapi memerlukan waktu lebih untuk melakukan klasifikasi/ prediksi. MLP memerlukan waktu lebih lama dalam proses pelatihan. Sedangkan kelebihan dari algoritma ini mampu melakukan klasifikasi kelas jamak dengan performa yang baik, dan waktu ujicoba yang cepat. Selain itu MLP juga lebih hemat memori karena hanya menyimpan bobot saja untuk melakukan prediksi/ klasifikasi. Algoritma SVM memerlukan waktu yang lebih lama dalam pembuatan model karena harus menyimpan matriks kernel yang digunakan saat proses klasifikasi sehingga kondisi ini menimbulkan pemborosan memori untuk mengalokasikan matriks kernel tersebut. Keuntungan dari algoritma ini mampu mengklasifikasikan dataset berdimensi besar, dapat melakukan klasifikasi untuk lebih dari dua kelas, serta mampu bekerja dengan baik meskipun dataset memiliki banyak atribut.

IV. KESIMPULAN

Berdasarkan ujicoba yang dilakukan untuk klasifikasi mental workload berdasarkan data EEG menggunakan algoritma Random Forest, KNN, MLP, dan SVM dengan ekstraksi fitur PCA, serta menggunakan *10-fold cross validation* menunjukkan bahwa algoritma terbaik dari sisi akurasi adalah algoritma KNN, demikian juga dari sisi pembuatan model. Sedangkan Random Forest terbaik kedua. Namun dari segi memory, KNN memerlukan memory untuk menyimpan dataset lebih besar. Berbeda dengan Random Forest maupun MLP, dan SVM. Dari sisi akurasi dan waktu, MLP dan SVM bukan tidak menunjukkan performa yang bagus baik dari sisi waktu maupun akurasi bila dibandingkan Random Forest dan KNN. Dari semua kriteria tersebut untuk diterapkan secara real-time, maka lebih disarankan menggunakan Random Forest, karena lebih hemat memory dibandingkan dengan KNN.

UCAPAN TERIMA KASIH

Penelitian ini merupakan bagian dari penelitian payung Riset Terapan DRPM dengan nomor kontrak No.B/11606/UN38.9/LK.04.00/2020.

DAFTAR PUSTAKA

- [1] R. Kazemi, S. Karimpour, M. Shahriyari, and S. N. Hossaini, "A Survey of the Relationship between the Mental Workload and Cognitive Failure in Taxi," *J Heal. Sci Surveill. Sys*, vol. 5, no. 4, pp. 188–192, 2017.
- [2] E. Poerwanto and Gunawan, "ANALISIS BEBAN KERJA MENTAL PEKERJA

- BAGIAN GROUND HANDLING BANDARA ADISUTJIPTO UNTUK MENDUKUNG KESELAMATAN PENERBANGAN,” *J. ANGKASA*, vol. VII, no. 2, pp. 115–126, 2015.
- [3] N. Ulfah *et al.*, “Model Kuantitatif Manajemen Kelelahan dan Beban Kerja untuk Peningkatan Produktivitas Pekerja Penggilingan Padi,” *J. Kesehat. Masy.*, vol. 7, no. 10, pp. 477–480, 2013.
- [4] Y. Ahuja and S. K. Yadav, “Multiclass Classification and Support Vector Machine,” *Glob. J. Comput. Sci. Technol. Interdiscip.*, vol. 12, no. 11, pp. 15–20, 2012.
- [5] A. Narang, B. Batra, A. Ahuja, J. Yadav, and N. Pachauri, “Classification of EEG signals for epileptic seizures using Levenberg- Marquardt algorithm based Multilayer Perceptron Neural Network,” *J. Intell. Fuzzy Syst.*, vol. 34, pp. 1669–1677, 2018.
- [6] R. Singla, A. Agrawal, V. Kumar, and O. P. Verma, “Real-Time Mental Workload Detector for Estimating Human Performance Under Workload,” *Adv. Signal Process. Commun.*, 2019.
- [7] J. J. Bird, L. J. Manso, D. R. Faria, and E. P. Ribeiro, “A Study on Mental State Classification using EEG-based Brain-Machine Interface,” *2018 Int. Conf. Intell. Syst. IEEE*, 2018.
- [8] J. L. Lobo, J. Del Ser, F. De Simone, and R. Presta, “Cognitive Workload Classification using Eye-tracking and EEG Data,” *HCI-Acro’16*, 2016.
- [9] A. Appriou, A. Cichocki, and F. Lotte, “Towards Robust Neuroadaptive HCI : Exploring Modern Machine Learning Methods to Estimate Mental Workload From EEG Signals,” *CHI’18 Ext. Abstr.*, vol. April 21-2, pp. 1–6, 2018.
- [10] R. N. Roy, S. Charbonnier, A. Campagne, and S. Bonnet, “Efficient mental workload estimation using task-independent EEG features,” *J. Neural Eng.*, vol. 13, pp. 1–10, 2016.
- [11] N. Farnaaz and M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System,” *Procedia - Procedia Comput. Sci.*, vol. 89, pp. 213–217, 2016.
- [12] S. A. Naghibi, H. R. Pourghasemi, and B. Dixon, “classification and regression tree , and random forest machine learning models in Iran,” *Env. Monit Assess*, vol. 188, no. 44, 2016.
- [13] A. D. Duru, “Determination of Increased Mental Workload Condition From EEG by the Use of Classification Techniques,” *Int. J. Adv. Eng. Pure Sci.*, pp. 47–52, 2019.
- [14] T. Rahman, A. K. Ghosh, M. H. Shuvo, and M. Rahman, “Mental Stress Recognition using K-Nearest Neighbor (KNN) Classifier on EEG Signals,” *Int. Conf. Mater. Electron. Inf. Eng. ICMEIE-2015*, no. July, 2015.
- [15] S. Ibrahim, R. Djemal, and A. Alsuwailem, “Electroencephalography (EEG) signal processing for epilepsy and autism spectrum disorder diagnosis,” *Biocybern. Biomed. Eng.*, vol. 38, pp. 16–26, 2017.
- [16] R. Rajagopal and V. Ranganathan, “Design of a hybrid model for cardiac arrhythmia classification based on Daubechies wavelet transform,” *Adv Clin Exp Med.*, vol. 27, no. 6, pp. 0–7, 2018.
- [17] B. C. Yavuz, N. Yurtay, and O. Ozkan, “Prediction of Protein Secondary Structure With Clonal Selection Algorithm and Multilayer Perceptron,” *IEEE*, vol. 6, pp. 45256–45261, 2018.
- [18] S. G. Fashoto, O. Owolabi, O. Adeleye, and J. Wandera, “Hybrid Methods for Credit Card Fraud Detection Using K-means Clustering with Hidden Markov Model and Multilayer Perceptron Algorithm,” *Br. J. Appl. Sci. Technol.*, vol. 13, no. 5, pp. 1–11, 2016.
- [19] K. Sabancı and M. Koklu, “The Classification of Eye State by Using kNN and MLP Classification Models According to the EEG Signals,” *Int. J. Intell. Syst.*

- Appl. Eng.*, vol. 3, no. 4, pp. 127–130, 2015.
- [20] X. Zhai, A. Ait, S. I. Ali, and S. Member, “MLP Neural Network Based Gas Classification System on Zynq SoC,” *IEEE Access*, vol. 4, pp. 8138–8146, 2016.
- [21] P. T. Noi and M. Kappas, “Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery,” *Sensors*, vol. 18, no. 18, pp. 1–20, 2018.
- [22] Z. Wei, C. Wu, X. Wang, A. Supratak, P. Wang, and Y. Guo, “Using Support Vector Machine on EEG for Advertisement Impact Assessment,” *Front. Neurosci*, vol. 12, no. 76, 2018.
- [23] W. L. Lim, O. Sourina, and L. P. Wang, “STEW: Simultaneous Task EEG Workload Dataset,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, pp. 1–8, 2018.
- [24] S. Chandra, K. L. Verma, G. Sharma, A. Mittal, and D. Jha, “EEG BASED COGNITIVE WORKLOAD CLASSIFICATION DURING NASA MATB-II MULTITASKING,” *Int. J. Cogn. Res. Sci. Eng. Educ.*, vol. 3, no. 1, pp. 35–42, 2015.
- [25] B. Purnama, *Pengantar Machine Learning: Konsep dan Praktikum dengan Contoh Latihan Berbasis R dan Phyton*. Bandung: Informatika, 2019.
- [26] J. Lever, M. Krzywinski, and N. Altman, “Principal component analysis,” *Nat. Publ. Gr.*, vol. 14, no. 7, pp. 641–642, 2017.
- [27] S. Mishra et al., “Principal Component Analysis,” no. January, 2017.
- [28] S. Wold, K. I. M. Esbensen, and P. Geladi, “Principal Component Analysis,” vol. 2, pp. 37–52, 1987.
- [29] Y. S. Nugroho and N. Emiliyawati, “Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest,” *J. Tek. Elektro*, vol. 9, no. 1, pp. 24–29, 2017.
- [30] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to Data Mining*, 2nd Editio. Pearson, 2018.
- [31] E. Prasetyo, *DATA MINING-Mengolah Data menjadi Informasi Menggunakan Matlab*, Edisi I. Yogyakarta: ANDI, 2014.
- [32] L. Cai, N. F. Thornhill, S. Kuenzel, and B. C. Pal, “Real-Time Detection of Power System Disturbances Based on k -Nearest Neighbor Analysis,” *IEEE Access*, vol. 5, pp. 5631–5639, 2017.
- [33] I. Aljarah, H. Faris, and S. Mirjalili, “Optimizing connection weights in neural networks using the whale optimization algorithm,” *Soft Comput.*, 2016.
- [34] H. Sanz, C. Valim, E. Vegas, J. M. Oller, and F. Reverter, “SVM-RFE : selection and visualization of the most relevant features through non-linear kernels,” *BMC Bioinformatics*, vol. 19, no. 432, pp. 1–18, 2018.
- [35] B. Santosa, *Data Mining Teori & Aplikasi*. Graha Ilmu, 2007.
- [36] R. Xinjiang, S. Tian, X. Zhang, J. Tian, and Q. Sun, “Random Forest Classification of Wetland Landcovers from Multi-Sensor Data in the Arid,” *Remote Sens*, vol. 8, no. 954, pp. 1–14, 2016.
- [37] A. E. Mohamed, “Comparative Study of Four Supervised Machine Learning Techniques for Classification,” *Int. J. Appl. Sci. Technol.*, vol. 7, no. 2, pp. 5–18, 2017.
- [38] Y. Agarwal, A. Kumar, and R. Bhatnagar, “Identifying Non-pulsar radiation and Predicting Chess Endgame Result Using ARSkNN,” *Smart Syst. IoT Innov. Comput. Proceeding SSIC 2019*, 2020.