

KLASIFIKASI AKTIVITAS MANUSIA MENGGUNAKAN METODE LONG SHORT-TERM MEMORY

Latansa Nury Izza Afida¹, Fitra Abdurrachman Bachtiar^{2*}, Imam Cholissodin³

^{1,2,3}Universitas Brawijaya, Malang

Email: ¹flatansa39@gmail.com, ²fitra.bachtiar@ub.ac.id, ³imamcs@ub.ac.id

*Penulis Korespondensi

(Naskah masuk: 31 Januari 2023, diterima untuk diterbitkan: 31 Maret 2023)

Abstrak

Klasifikasi aktivitas manusia merupakan salah satu topik penelitian yang penting karena dapat diterapkan pada berbagai bidang dan memiliki manfaat yang luas. Penelitian mengenai klasifikasi aktivitas manusia sebelumnya telah banyak dikembangkan dengan menerapkan dataset publik pada repositori dataset Human Activity Recognition. Namun dataset tersebut memiliki fitur yang berdimensi tinggi sehingga dataset memiliki dimensi yang tinggi pula. Pada beberapa penelitian sebelumnya menunjukkan bahwa algoritma SVM dan *Random Forest* merupakan algoritma dengan nilai akurasi yang lebih unggul dibandingkan dengan model lainnya. Akan tetapi berdasarkan penelitian tersebut model tersebut belum pernah diimplementasikan pada kasus riil yaitu pada perangkat bergerak. Penelitian ini mengusulkan model pengenalan aktivitas manusia dengan kasus riil dengan dataset primer yang dikumpulkan dengan menggunakan *smartphone*. Pengambilan dataset primer melibatkan 10 responden. Data yang terkumpul dengan *smartphone* direkam melalui sensor menghasilkan dataset berbentuk data *time series*. Dataset primer yang digunakan masih memiliki nilai yang besar dan kurangnya keseimbangan jumlah label kelas sehingga eksperimen dimulai dengan tahapan *preprocessing* yang dilakukan dengan menggunakan *moving average* untuk mereduksi data tanpa menghilangkan informasi. Selain itu juga dilakukan SMOTE untuk menyeimbangkan jumlah masing - masing kelas data. Data latih memiliki proporsi sebanyak 80%, data validasi sebanyak 10% dan data uji sebanyak 10%. Penelitian ini menggunakan LSTM untuk klasifikasi aktivitas manusia karena algoritma ini sangat baik untuk memproses data *time series* berjumlah banyak. Hasil klasifikasi kemudian dibandingkan dengan algoritma terbaik pada beberapa penelitian sebelumnya. Hasil eksperimen didapatkan bahwa model LSTM dapat mengungguli model SVM dan *Random Forest*. Hasil klasifikasi menggunakan algoritma LSTM mencapai akurasi, *Precision*, *Recall*, dan *F1-score* 95%, 96%, 95%, dan 95%, secara berurutan.

Kata kunci: Klasifikasi, LSTM, SVM, *Random Forest*, *Human Activity Recognition*

CLASSIFICATION OF HUMAN ACTIVITY USING LONG SHORT-TERM MEMORY

Abstract

Human activity classification is one of the important research topics because it can be applied to various fields and have broad benefits. Research on human activity classification has previously been developed by applying public datasets to the available Human Activity Recognition dataset repository. However, the dataset has high dimensional features so that the dataset has high dimensions as well. Previous study has shown that SVM and *Random Forest* algorithms are algorithms with superior accuracy values compared to other models. However, based on previous research, the model has never been implemented in real cases, namely on mobile devices. This research proposes a human activity recognition model in real cases situation with primary datasets collected using smartphones. The data collection for the dataset involved 10 respondents. The data collected using a smartphone recorded via sensors to produce a dataset in the form of time series data. The primary dataset used still has a large value and there is a lack of balance in the number of class labels. To this end, the experiment begins with a preprocessing stage which is carried out using a moving average to reduce the data without losing information. In addition, SMOTE was also carried out to balance the number of each data class. The proportion of training data, validation data, and testing data is 80%, 10%, and 10%, respectively. This research uses LSTM for human activity classification because this algorithm is very good for processing large amounts of time series data. The classification results were then compared with the best algorithms in several previous studies. Experimental results show that the LSTM model can outperform the SVM and *Random Forest* models. Classification results using the LSTM algorithm reached Accuracy, Precision, Recall, dan F1-score 95%, 96%, 95%, and 95%, respectively.

Keywords: *Classification, LSTM, SVM, Random Forest, Human Activity Recognition*

1. PENDAHULUAN

Aktivitas manusia sehari-hari dapat diketahui berapa banyak waktu yang dibutuhkan untuk melakukan aktivitas tersebut. Berapa lama kita berjalan atau berapa lama duduk dalam sehari. Ilmu yang mempelajari tentang hal tersebut disebut pengenalan aktivitas manusia. (Bachtiar dkk, 2021) Pengenalan aktivitas manusia merupakan salah satu topik penelitian yang penting karena memiliki potensi untuk dapat diterapkan pada berbagai bidang seperti pengawasan, kesehatan, dan kecerdasan buatan (Khan dkk, 2020). Salah satu cara dalam melakukan pengenalan aktivitas manusia yaitu menggunakan sensor. Pengenalan aktivitas manusia juga dilakukan menggunakan visi komputer, namun secara umum penggunaan sensor akan lebih unggul karena tidak sensitif terhadap *noise*, murah, dan konsumsi daya yang lebih efisien (Bevilacqua dkk, 2019). Penelitian mengenai pengenalan aktivitas manusia menggunakan sensor *smartphone* telah sering dilakukan dengan memanfaatkan sensor *gyroscope* dan *accelerometer* pada *smartphone* sehingga pengguna tidak lagi perlu menambahkan sensor tambahan karena sudah terintegrasi pada *smartphone* (Reiss dkk, 2013).

Penelitian mengenai klasifikasi aktivitas manusia sebelumnya telah banyak dikembangkan dengan menerapkan dataset publik yang telah tersedia. Namun dataset tersebut memiliki dimensi yang tinggi dikarenakan dicantumkan merupakan hasil ekstraksi fitur sebenarnya. Hal tersebut akan membuat proses komputasi menjadi lebih berat (Anguita dkk, 2013). Dimensi yang tinggi akan memungkinkan dataset berisi banyak fitur yang tidak memiliki relevansi. Tingginya dimensi juga menyebabkan model bingung dalam proses klasifikasinya sehingga membutuhkan peningkatan kinerja untuk mendapatkan hasil klasifikasi yang baik (Zhang dkk, 2011). Dimensi yang besar serta kompleksnya model yang digunakan akan menghasilkan model yang kurang efektif bila diterapkan pada kasus nyata. Pada penelitian yang dilakukan Ghate & Hemalatha (2023) klasifikasi aktivitas manusia yang digunakan dengan data publik UCI HAR menggunakan beberapa *machine learning* yang memiliki label output enam aktivitas. Penelitian tersebut mendapatkan metode yang terbaik yaitu SVM dengan akurasi sebesar 96,4 %. Penelitian lain oleh Rabbi dkk. (2021) dengan menggunakan dataset yang sama yaitu UCI HAR dan akurasi terbaik juga dihasilkan pada metode SVM yaitu sebesar 96,33%. SVM merupakan algoritma *supervised learning* yang sangat cocok untuk proses klasifikasi pada data yang telah memiliki label (Stitson, 1996). Dataset publik UCI HAR merupakan dataset dengan dimensi tinggi, sehingga dataset ini akan sangat cocok diproses menggunakan algoritma SVM yang mampu dengan

sangat baik memproses dataset berdimensi tinggi (Yulianti dkk, 2020).

Penelitian sebelumnya dengan menggunakan dataset publik yang telah disediakan pada repositori UCI akurasi terbaik dihasilkan oleh algoritma SVM. Proses pengembangan model yang disiapkan untuk implementasi kasus riil diperlukan sampel data yang lebih banyak dan pada sumber daya yang terbatas. Oleh karena itu, klasifikasi aktivitas manusia dengan menggunakan data primer perlu dilakukan. Data primer dalam perekaman aktivitas manusia merupakan data yang dikumpulkan langsung dari nilai sensor *accelerometer* dan *gyroscope* pada *smartphone* yang biasanya bersifat *timeseries*. Namun SVM memiliki keterbatasan pada komputasi dan penggunaan memori pada data dalam jumlah besar. Menurut Khalilzadeh & Tasci (2017), data yang besar merupakan data yang memiliki lebih dari 10.000 data. sehingga dari permasalahan tersebut dibutuhkan algoritma *deep learning* untuk dapat menangani besarnya data primer yang tidak kalah penting untuk dilakukan penelitian terkait.

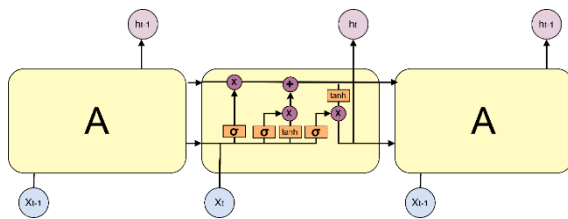
Penelitian mengenai klasifikasi aktivitas manusia menggunakan dataset UCI - HAR yang menerapkan algoritma mendalam juga telah banyak dilakukan. seperti pada penelitian yang dilakukan Ullah (2019) penelitian ini menggunakan stacked 5 LSTM dengan akurasi 93%. Penelitian lain dilakukan Mutageki (2020) menggunakan CNN - LSTM mendapatkan akurasi 92%. Penelitian terkait juga dilakukan Yu & Qin (2018) menggunakan algoritma BiLSTM mendapatkan akurasi sebesar 93,79%. Berdasarkan penelitian sebelumnya, penggunaan algoritma mendalam pada repositori dataset HAR UCI mampu memberikan akurasi yang tinggi. Namun beberapa algoritma *deep learning* yang digunakan tersebut akan membutuhkan banyak waktu untuk proses pembelajarannya.

Berdasarkan beberapa uraian sebelumnya diperlukan model algoritma mendalam yang ringan digunakan sehingga dapat menghasilkan akurasi yang tinggi namun tetap efisien terhadap waktu komputasi. Penggunaan dataset HAR pada UCI *Repository* yang telah banyak dilakukan juga memiliki kekurangan apabila model akan diimplementasikan pada kasus nyata (Zhang dkk, 2011). Untuk itu pada penelitian ini diusulkan penggunaan dataset primer yang memiliki jumlah besar. Penggunaan dataset primer yang bersifat *time series* akan memiliki hasil yang baik bila diproses dengan menggunakan algoritma *deep learning* LSTM. Penerapan dataset primer dengan algoritma mendalam LSTM pada penelitian ini diharapkan mampu melakukan pengenalan aktivitas manusia dengan algoritma yang ringan pada proses komputasinya sehingga model dapat dikembangkan pada implementasi perangkat *smartphone*.

2. LANDASAN PUSTAKA

2.1. Long Short-Term Memory (LSTM)

LSTM merupakan salah satu metode yang mengimplementasikan arsitektur dari Recurrent Neural Network (RNN). LSTM dikenal sebagai metode yang dapat menyelesaikan masalah *vanishing gradient* yang masih belum terselesaikan pada RNN. *Vanishing gradient* pada RNN terjadi ketika harus menyimpan memori informasi pada jangka panjang. Terselesaikannya *vanishing gradient* pada LSTM dikarenakan memiliki unit yang mampu mempelajari kapan waktu untuk membuka dan menutup pintu pada jaringan sehingga dapat mengatur pergerakan *error* yang konstan pada jaringan (Sagheer dkk, 2019). Pada arsitektur jaringan RNN hanya menggunakan satu layer sederhana pada iterasinya, sedangkan pada LSTM terdapat 4 layer dalam iterasinya (Cholissodin dkk, 2021). Arsitektur LSTM ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur LSTM
(Sumber : Abadi, 2021)

Proses perhitungan pada sel LSTM dimulai pada perhitungan *forget gate* menggunakan Persamaan 1. Langkah selanjutnya merupakan perhitungan yang dilakukan pada *input gate* dan juga perhitungan *candidate state* menggunakan Persamaan 2 dan Persamaan 3. Setelah nilai pada *input gate* didapat, selanjutnya melakukan perhitungan nilai *cell state* menggunakan Persamaan 4. Langkah terakhir yaitu menghitung nilai *output gate* dan *hidden state* menggunakan Persamaan 5 dan Persamaan 6.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

Pada persamaan tersebut, f_t merupakan *forget gate*. Sedangkan σ merupakan fungsi *sigmoid*. W_f Adalah bobot pada *forget gate*. h_{t-1} merupakan hasil output pada *time step* $t - 1$. x_t bobot pada *forget gate*. b_f bias pada *forget gate*. Gate pertama yang dijalankan dalam sel LSTM adalah *forget gate*. *Forget gate* akan menentukan informasi yang akan disimpan maupun informasi yang harus dibuang dari *cell state*. h_{t-1} merupakan keluaran dari proses LSTM pada *time step* $t - 1$ dan x_t merupakan *input* pada *time step* t . Output dari *forget gate* merupakan nilai dengan rentang 0 hingga 1.

$$I_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Pada kedua persamaan tersebut, I_t adalah *input gate*. W_i bobot pada *input gate*. b_i bias pada *input gate*. C'_t merupakan kandidat *cell state* baru pada C_{t-1} . \tanh adalah fungsi *tanh*. W_i merupakan bobot pada *cell state* baru. h_{t-1} merupakan *output* pada *time step* $t - 1$. x_t merupakan *input* pada *time step* t . Sedangkan b_c merupakan bias pada *cell state* baru. Setelah didapatkan hasil keluaran dari *forget gate* maka operasi selanjutnya yaitu *input gate*. *Input gate* memiliki dua bagian, bagian pertama menerapkan fungsi *sigmoid* yang mampu menentukan informasi yang akan diperbarui. Sedangkan bagian lainnya menerapkan fungsi *tanh* sehingga mampu menentukan vektor yang ditambahkan pada *cell state*. Kedua bagian tersebut akan menentukan informasi baru mana yang akan disimpan pada *cell state*.

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (4)$$

Dalam persamaan tersebut, C_t merupakan *cell state* pada *time step* t . f_t merupakan *forget gate*. C_{t-1} adalah *cell state* pada *time step* $t - 1$. Sedangkan C'_t merupakan nilai *cell state* baru hasil *input gate*. Pada tahap ini, akan dihitung nilai *cell state* yang baru. Hasil dari *forget gate* dan *input gate* akan dioperasikan supaya dapat memperbarui nilai *cell state* C_{t-1} . Selanjutnya akan dilakukan perkalian dari *cell state* C_{t-1} dengan hasil dari *forget gate*, setelah dilakukan perkalian hasilnya akan ditambah dengan hasil operasi $i_t * C'_t$.

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

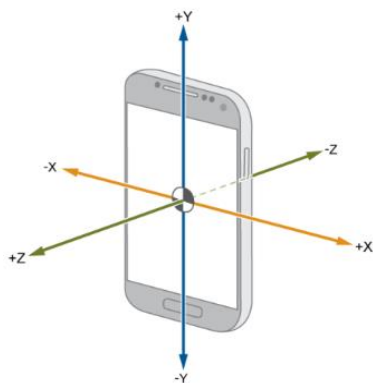
$$h_t = o_t * \tanh(C_t) \quad (6)$$

Pada kedua persamaan tersebut, O_t merupakan *output gate*. σ adalah fungsi *sigmoid*. W_o merupakan bobot pada *output gate*. h_{t-1} adalah *output* pada *time step* $t - 1$. x_t adalah *input* pada *time step* t . b_o merupakan bias pada *output gate*. h_t merupakan *output* pada *time step* t . \tanh adalah fungsi *tanh*. Apabila operasi *Cell State* telah menghasilkan nilai keluaran maka selanjutnya akan mengoperasikan *output gate*. Pada *output gate*, fungsi *sigmoid* akan dioperasikan terlebih dahulu untuk menentukan bagian *cell state* yang digunakan untuk hasil keluaran. Setelah dilakukan operasi *sigmoid*, operasi *tanh* akan dioperasikan kemudian hasilnya akan dikalikan dengan hasil keluaran dari fungsi *sigmoid* pada perhitungan sebelumnya.

2.2. Accelerometer

Sensor *accelerometer* adalah perangkat elektromagnetik yang berfungsi untuk mengukur nilai percepatan gerak suatu objek. Gaya statis disebabkan oleh gravitasi yang menarik suatu objek. sedangkan gaya dinamis disebabkan karena pergerakan sensor *accelerometer*. Prinsip kerja

sensor *accelerometer* menggunakan prinsip percepatan. Pada sensor *accelerometer* terdapat sebuah per dengan beban yang dilepaskan. Beban tersebut akan bergerak dengan suatu keadaan tertentu kemudian akan terhenti. Jika terdapat objek yang menyebabkan sensor *accelerometer* bergerak maka beban tersebut akan berayun kembali.

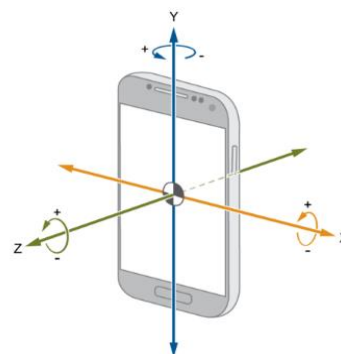


Gambar 2. Ilustrasi Sumbu Accelerometer

Terdapat tiga sumbu pada *accelerometer* diantaranya sumbu *x* untuk bergerak maju dan mundur, sumbu *y* untuk bergerak kanan dan kiri, dan sumbu *z* untuk bergerak atas dan bawah. Sensor *accelerometer* umumnya dapat ditemukan dengan mudah pada *smartphone* tipe terbaru yang biasanya sudah memiliki tingkat akurasi yang tinggi. Pada *smartphone* biasanya fitur ini digunakan untuk mengetahui posisi *smartphone* sehingga mampu untuk menyesuaikan orientasi layar. Sumbu pada sensor *accelerometer* digambarkan pada Gambar 3. Selain itu pengguna juga dapat dengan mudah mengakses nilai sensor dengan membuat kode program sederhana untuk mengambil rekam nilai sensor.

2.3. Gyroscope

Sensor *gyroscope* adalah perangkat sensor yang terdiri dari sebuah piringan atau roda yang bisa diputar dengan cepat pada satu sumbu tertentu secara bebas. Dengan adanya putaran tersebut, akan menyebabkan adanya perubahan momen gaya yang ditimbulkan secara tegak lurus dengan arah putaran. Sensor *gyroscope* berguna untuk mengukur orientasi objek berdasarkan momentum sudut, sehingga mampu untuk mengukur kecepatan sudut dari suatu rotasi dengan satuan radian per detik. Secara matematik, *gyroscope* dapat dimodelkan berdasarkan hukum energi kinetik dan perubahan momentum sudut. Gaya inersia yang ditimbulkan oleh sensor *gyroscope* dapat dimanfaatkan untuk memberi reaksi pada sistem keseimbangannya.



Gambar 3. Ilustrasi Sumbu Accelerometer

Prinsip kerja sensor *gyroscope* yaitu pada kondisi diam, tegangan output yang dihasilkan ketika sensor bergerak akan bernilai konstan. Sensor ini memiliki nilai berupa kecepatan sudut dari tiga sumbu. Nilai sumbu *x* adalah pergerakan kanan dan kiri, nilai sumbu *y* adalah pergerakan ke atas dan bawah, dan sumbu *z* adalah pergerakan ke depan dan belakang.

3. METODOLOGI PENELITIAN

3.1 Data Penelitian

Dataset yang digunakan pada penelitian ini merupakan dataset primer yang diperoleh melalui proses praktik pengambilan data berdasarkan skenario pengambilan data yang telah dirancang. Dataset diambil dengan menggunakan program *android* yang dibuat khusus untuk perekaman data sensor *accelerometer* dan *gyroscope* beserta labelnya. Pengumpulan dataset dilakukan dengan 10 responden yang meletakkan *smartphone* pada saku celana sebelah kanan. Pada sekali gerakan statis direkam selama 5 detik, gerakan dinamis direkam selama 7 detik, dan gerakan transisi direkam selama 3 detik. Alur gerakan yang dipraktikkan pada proses pengambilan data digambarkan pada Tabel 1.

Tabel 1. Proporsi Waktu Tiap Aktivitas

Aktivitas	Proporsi Waktu
Berdiri	5 detik
Duduk	5 detik
Tidur	5 detik
Berdiri ke Duduk	3 detik
Berdiri ke Tidur	3 detik
Duduk ke Berdiri	3 detik
Duduk ke Tidur	3 detik
Tidur ke Berdiri	3 detik
Tidur ke Duduk	3 detik
Jalan	7 detik
Gerak ke Kanan	7 detik

Aktivitas	Proporsi Waktu
Gerak ke Kiri	7 detik
Lompat	7 detik
Naik Tangga	7 detik
Turun Tangga	7 detik

Masing – masing gerakan akan dilakukan dilakukan secara sekuensial dalam 10 kali percobaan pada 10 responden. Sehingga dari praktik dengan skenario pengambilan data tersebut akan menghasilkan dataset yang terdiri dari 1034250 data, 6 kolom fitur yaitu *Acc_x*, *Acc_y*, *Acc_z*, *Gyro_x*, *Gyro_y*, *Gyro_z* dan 1 kolom fitur label *state*.

3.2 Pemrosesan Data

Dataset primer yang telah berhasil dikumpulkan akan diproses terlebih dahulu sebelum data akan diproses dengan model. Tahap pemrosesan data ini dibutuhkan untuk mentransformasikan data sehingga dapat diproses untuk dilakukan pelatihan menggunakan model. Tahapan pemrosesan data yang dilakukan yaitu diantaranya, proses *encoding* label, *data normalization*, *data smoothing*, *data splitting*, SMOTE dan *sliding window*.

3.2.1 Encoding Label

Encoding merupakan salah satu tahap pada proses *preprocessing* data. Proses ini dilakukan dengan merubah nilai dengan tipe data kategori ke numerik sebelum diproses pada *machine learning*. Proses *encoding* pada penelitian ini merubah nilai label aktivitas pada kolom *state* menjadi angka dengan tipe data integer sehingga akan mudah dikenali oleh model.

3.2.2 Normalisasi Data

Proses normalisasi data pada penelitian ini dilakukan dengan menggunakan teknik *MinMax Scaler*. Tujuan dari data normalisasi ini untuk menyamakan nilai rentang data pada setiap kolomnya. Penggunaan normalisasi membuat nilai minimal dan maksimal pada setiap kolom tidak akan memiliki perbedaan nilai yang terlalu jauh namun juga tidak merubah skala rentang nilainya.

3.2.3 Moving Averages

Moving Averages atau juga disebut sebagai *rolling averages* merupakan suatu teknik yang digunakan untuk menganalisis data deret waktu dengan menghitung rata – rata subset berbeda dari sekumpulan data yang lengkap. Dalam perhitungannya *Moving Averages* melibatkan pengambilan rata – rata dataset dari waktu ke waktu, maka teknik ini sering disebut dengan *rolling mean*. Penggunaan *simple moving averages* ini digunakan sebagai teknik *smoothing* untuk mengambil data data yang penting saja tanpa menghilangkan informasi yang terdapat pada dataset tersebut.

Salah satu cara kerja *Moving Averages* ini dengan cara mengambil subset tetap dari rangkaian data yang lengkap. Rata – rata pergerakan pertama akan dilakukan dengan menghitung sekumpulan data tetap pertama, kemudian subset akan diubah dengan bergerak maju ke subset dari data tetap yang berikutnya dan dihitung juga nilai rata – ratanya. Peran utama dari *moving averages* pada data deret waktu yaitu untuk menangkap fluktuasi jangka pendek agar dapat fokus pada tren yang lebih panjang. Persamaan dari *Moving Averages* ditunjukkan pada Persamaan 7 (Aini dkk, 2018).

$$MA_n = \frac{\sum_{i=1}^n D_i}{n} \quad (7)$$

MA_n merupakan *moving averages* sebanyak n data. $\sum_{i=1}^n D_i$ adalah jumlah data sebanyak n . n adalah banyak data yang ditentukan untuk pergerakan maju.

3.2.4 Data Splitting

Proses *splitting* data atau pembagian data pada penelitian ini digunakan untuk membagi data menjadi data latih, data validasi dan data uji. Data latih digunakan pada proses pelatihan model dan memiliki jumlah sebesar 80% dari keseluruhan data. Data validasi digunakan untuk mengetahui akurasi validasi pada proses pelatihan, data validasi memiliki jumlah sebesar 10% dari keseluruhan data. Data uji digunakan untuk proses pengujian data ketika telah didapatkan model paling optimal dari hasil pelatihan. Besarnya jumlah data uji yaitu 10% dari keseluruhan data.

3.2.4. Synthetic Minority Oversampling Technique (SMOTE)

SMOTE merupakan metode untuk *oversampling* dimana data pada kelas minor akan diperbanyak dengan menggunakan data sintetis yang merupakan replikasi dari data minoritas itu sendiri. *Oversampling* pada teknik SMOTE mengambil *instance* dari kelas minoritas lalu akan mencari *k-Nearest Neighbor* pada setiap *instance*. Setelah itu akan menghasilkan *instance* sintetis dari kelas minoritasnya. Menggunakan teknik SMOTE untuk menangani kasus *imbalance class* mampu mengatasi masalah *overfitting* yang berlebihan (Chawla, 2002). Parameter yang digunakan untuk melakukan SMOTE pada penelitian ini yaitu data latih dan label dari data latih.

Algoritma dari teknik SMOTE sendiri pertama akan mengambil nilai selisih antara vektor dari fitur pada kelas yang minoritas dan nilai *nearest neighbor* dari kelas minoritas. Setelah kedua faktor tersebut ditemukan lalu nilai tersebut akan dikali dengan angka acak yang bernilai antara 0 sampai 1. Kemudian hasil dari perkalian tersebut akan ditambahkan dengan vektor fiturnya sehingga akan didapatkan nilai vektor yang baru (Jishan, 2015).

Perhitungan matematis dari teknik SMOTE ditunjukkan pada Persamaan 8 (Edi, 2002).

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \delta \quad (8)$$

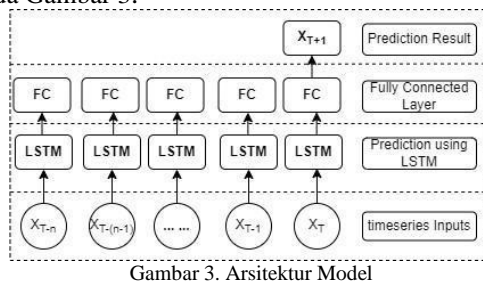
x_i merupakan vektor dari fitur pada kelas minoritas. \hat{x}_i adalah k - *Nearest Neighbor* untuk x_i . Sedangkan δ merupakan angka acak bernilai 0 sampai 1.

3.2.2. Sliding Window

Proses *sliding window* pada penelitian ini hanya dilakukan pada implementasi algoritma LSTM. Teknik *sliding window* digunakan untuk membentuk data menjadi bentuk sekuensial sehingga dapat diproses pada model LSTM yang mengolah data dalam bentuk iterasi. *Sliding window* pada penelitian ini digunakan *window* sebesar 64 dan *step* sebesar 7.

3.3 Modelling

Tahapan *modelling* merupakan tahapan implementasi model LSTM yang diusulkan pada penelitian ini. Arsitektur model tersebut ditunjukkan pada Gambar 3.



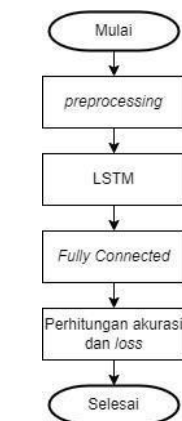
Gambar 3. Arsitektur Model

Arsitektur pada model LSTM yang digunakan adalah data yang telah di proses dengan tahap *pre-processing* seperti yang telah dijelaskan pada uraian sebelumnya. Kemudian data tersebut akan diproses pada layer LSTM dengan 32 unit dan *input shape* (64, 7). Setelah itu data keluarannya akan melewati lapisan *fully connected* pada lapisan ini digunakan *output* label yaitu berjumlah 16.

Proses yang terjadi pada model LSTM yang digunakan melewati tiga tahapan utama. Pertama data yang telah berhasil dikumpulkan dari skenario pengambilan data yang telah dibuat akan diproses dengan normalisasi, reduksi data, SMOTE, dan data split, sehingga data akan menjadi lebih siap untuk diproses lebih lanjut pada model. Pada model data yang sudah siap akan dimasukkan pada lapisan LSTM dengan 32 unit dan *input shape* (64, 7). Kemudian dari lapisan LSTM hasil dari lapisan LSTM akan dimasukkan pada lapisan *fully connected* layer dengan nilai output 16 sesuai dengan jumlah label kelas yang akan diprediksi melalui proses *encoding*. setelah memasuki rangkaian lapisan tersebut akan dicetak nilai akurasi dan *loss* pelatihannya.

3.4 Rancangan Eksperimen

Pada tahap ini eksperimen akan dilakukan untuk mengetahui metodologi usulan pada kasus klasifikasi aktivitas manusia. Pertama, eksperimen yang akan dilakukan yaitu mengoptimalkan model dengan menggunakan dataset primer. Kinerja model juga akan di optimalisasi dengan menggunakan *hyperparameter* yang telah ditetapkan sebelumnya pada model yang akan diusulkan. Kedua, model yang diusulkan akan dibandingkan dengan metode *supervised learning* lainnya yang mengacu pada penelitian yang dilakukan oleh Rabbi dkk. (2021) dan Ghate & Hemalatha (2023). Beberapa metode *supervised learning* tersebut adalah SVM, *Bagging Ensemble Decision Tree*, *Gradient Boosting*, *Random Forest*, *KNN*, *linear SVM*, Logistic Regression, dan ANN. Pada perbandingan ini model dibangun menggunakan repositori dataset dari UCI yang dapat diakses melalui <https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones>. Ilustrasi dari alur model dapat dilihat pada Gambar 4.



Gambar 4. Alur Model

Pada penelitian yang dilakukan oleh Rabbi dkk, (2021) dan Ghate & Hemalatha (2023) algoritma yang menghasilkan akurasi tertinggi juga dihasilkan oleh algoritma SVM. Namun Kedua penelitian tersebut masih dilakukan dengan menggunakan dataset publik HAR dari UCI *Repository* yang memiliki dimensi dan ekstraksi fitur yang tinggi sehingga akan kurang efisien apabila model akan disiapkan untuk implementasi kasus nyata. Pada penelitian ini perbandingan akan dilakukan menggunakan metode machine learning dengan menggunakan repositori dataset UCI HAR dan juga menggunakan dataset primer yang dikumpulkan langsung oleh peneliti sesuai dengan skenario pengambilan data yang telah dirancang. Dataset primer yang digunakan memiliki jumlah yang besar dan bersifat *time series* sehingga penelitian akan menggunakan algoritma LSTM.

Untuk proses pelatihan dan pengujian akan menggunakan proporsi data 80% latih, 10% data validasi, dan 10% data uji. Untuk 80% data latih

selanjutnya akan digunakan untuk mencari hyperparameter terbaik dari setiap model.

3.5 Metode Evaluasi

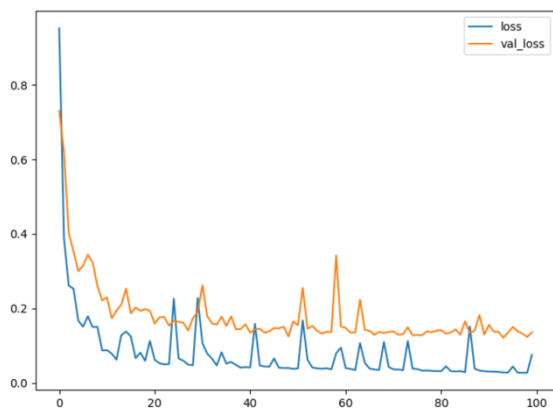
Model yang diusulkan perlu dievaluasi untuk mengetahui kehandalan model untuk mengklasifikasikan aktivitas manusia. Pada studi ini evaluasi model menggunakan *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Selain itu penelitian ini juga mencetak hasil *Confusion Matrix* untuk mendapatkan pemahaman kemampuan model lebih lanjut pada kasus klasifikasi aktivitas manusia.

4. HASIL

Pada bagian ini menjelaskan hasil eksperimen yang telah rancang pada bab sebelumnya. Hasil penelitian akan menjelaskan kinerja model yang didapatkan pada model yang diusulkan.

4.1 Hasil Kinerja Model Usulan

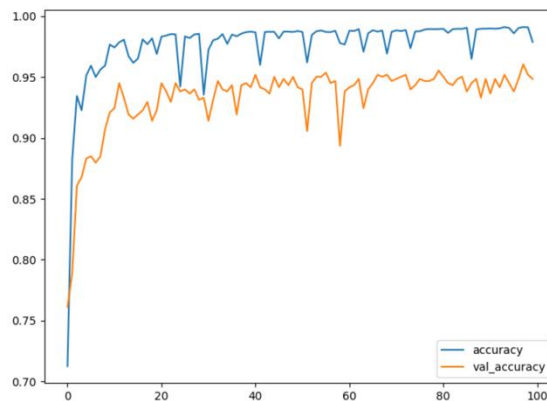
Penelitian ini menggunakan metode LSTM dengan 128 unit. *hyperparameter* yang digunakan diantaranya *learning rate* sebesar 0,001, *epoch* 100, dan *batch size* 64. Model yang diusulkan diharapkan mampu meningkatkan akurasi namun tetap efisien dalam penggunaan waktu komputasi. Pergerakan nilai *loss* pada proses pelatihan model pada penelitian ini digambarkan pada Gambar 5.



Gambar 5. Grafik Loss LSTM

Pada Gambar 5 merupakan grafik pergerakan nilai *loss* dengan iterasi sebanyak 100 epoch dengan *loss* yang dihasilkan yaitu 0,07. Grafik tersebut juga menunjukkan bahwa dengan model yang diusulkan berhasil melakukan pelatihan tanpa adanya masalah *overfitting*. Selain nilai *loss*, nilai akurasi juga dicetak pada pergerakan tiap *epoch* nya pada grafik yang digambarkan pada Gambar 6.

Pada Gambar 6 merupakan grafik pergerakan nilai akurasi dengan iterasi sebanyak 100 epoch dengan akurasi yang dihasilkan yaitu 0,95. Grafik pergerakan nilai akurasi pada tiap *epoch* ini juga menunjukkan bahwa model dapat melakukan klasifikasi dengan baik tanpa *overfitting*.



Gambar 6. Grafik Akurasi LSTM

4.2 Hasil Perbandingan Metode

Perbandingan metode dilakukan dengan membandingkan metode yang digunakan pada penelitian yang dilakukan oleh Ghate, V., dan Hemalatha C, S. (2023), metode yang digunakan pada penelitian yang dilakukan oleh Rabbi dkk, (2021), dan dengan menggunakan metode LSTM yang diusulkan. Perbandingan dilakukan secara dua tahap yaitu dengan menggunakan dataset sekunder dari HAR – UCI *Repository* dan menggunakan dataset primer. Perbandingan nilai akurasi pada algoritma beserta *hyperparameter* yang dilakukan oleh Ghate & Hemalatha (2023) dengan metode usulan menggunakan dataset sekunder ditunjukkan pada Tabel 3.

Tabel 3. Perbandingan Akurasi Algoritma yang digunakan Ghate & Hemalatha (2023) dengan Dataset Sekunder

Metode	Akurasi
Ensemble DT	80,3%
SVM	96,33%
GB	93,14%
RF	90,05%
KNN	88,5%
Linear SVM	95,3%
LSTM (model usulan)	97%

Pada Tabel 3 menunjukkan bahwa pada penelitian yang dilakukan oleh Ghate & Hemalatha (2023) akurasi tertinggi yang didapatkan yaitu 96.33% dengan menggunakan metode SVM. Namun dengan menggunakan metode dan *hyperparameter* usulan didapat akurasi yang lebih tinggi yaitu sebesar 97%.

Selain perbandingan yang dilakukan dengan mengimplementasikan dataset sekunder, juga dilakukan perbandingan dengan dataset primer. Perbandingan nilai akurasi pada algoritma beserta *hyperparameter* yang dilakukan oleh Ghate & Hemalatha (2023) dengan metode usulan

menggunakan dataset primer ditunjukkan pada Tabel 4.

Tabel 4. Perbandingan Akurasi Algoritma yang digunakan Ghate & Hemalatha (2023) dengan Dataset Primer

Metode	Akurasi
SVM	46,99%
LR	38,14%
DT	68,06%
RF	72,61%
ANN	61,30%
LTSM (model usulan)	95%

Pada Tabel 3 menunjukkan bahwa pada penelitian yang dilakukan oleh Ghate & Hemalatha (2023) akurasi tertinggi yang didapatkan yaitu 76,89% dengan menggunakan metode SVM. Namun dengan menggunakan metode dan *hyperparameter* usulan didapat akurasi yang lebih tinggi yaitu sebesar 95%.

Dari perbandingan yang telah dilakukan, klasifikasi aktivitas manusia menggunakan nilai sensor *accelerometer* dan *gyroscope* memiliki hasil yang baik dengan menggunakan algoritma *tree base* dalam hal ini SVM dan *Random Forest*. Namun penggunaan *Deep Learning* dengan arsitektur yang ringan memberikan hasil akurasi lebih tinggi dan waktu komputasinya juga lebih efisien.

Perbandingan juga dilakukan pada penelitian selanjutnya dengan membandingkan hasil akurasi dari seluruh metode yang digunakan pada penelitian yang dilakukan Rabbi dkk, (2021) dan metode usulan, menggunakan dataset sekunder dan dataset primer. Perbandingan nilai akurasi pada algoritma beserta *hyperparameter* yang dilakukan oleh Rabbi dkk, (2021) menggunakan dataset sekunder ditunjukkan pada Tabel 5.

Tabel 5. Perbandingan Akurasi Algoritma yang digunakan Rabbi, J., dkk, (2021) dengan Dataset Sekunder

Metode	Akurasi
SVM	96,33%
LR	96,1%
DT	80,30%
RF	90,05%
ANN	93,80%
LTSM (model usulan)	97%

Tabel 5 menunjukkan bahwa pada penelitian yang dilakukan oleh Rabbi dkk, (2021) akurasi

tertinggi yang didapatkan yaitu 96,33% dengan menggunakan metode SVM. Namun dengan menggunakan metode dan *hyperparameter* usulan didapat akurasi yang lebih tinggi yaitu sebesar 95%.

Selain perbandingan yang dilakukan dengan mengimplementasikan dataset sekunder, juga dilakukan perbandingan dengan dataset primer. Perbandingan nilai akurasi pada algoritma beserta *hyperparameter* yang dilakukan oleh Rabbi dkk, (2023) dengan metode usulan menggunakan dataset primer ditunjukkan pada Tabel 6.

Tabel 6. Perbandingan Akurasi Algoritma yang digunakan Rabbi dkk, (2021) dengan Dataset Primer

Metode	Akurasi
SVM	46,99%
LR	38,14%
DT	68,06%
RF	72,61%
ANN	61,30%
LTSM (usulan model)	95%

Tabel 6 juga menunjukkan bahwa pada penelitian yang dilakukan oleh Rabbi dkk, (2021) akurasi tertinggi yang didapatkan yaitu 72,61% dengan menggunakan metode SVM. Namun dengan menggunakan metode dan *hyperparameter* usulan didapat akurasi yang lebih tinggi yaitu sebesar 95%.

Dari perbandingan metode usulan dengan penelitian yang dilakukan Rabbi dkk, (2021) yang telah dilakukan, klasifikasi aktivitas manusia menggunakan nilai sensor *accelerometer* dan *gyroscope* memiliki hasil yang baik dengan menggunakan algoritma *tree base* dalam hal ini SVM dan *Random Forest*. Namun penggunaan metode serta *hyperparameter* usulan didapat akurasi yang lebih tinggi.

5. PEMBAHASAN

Pembahasan terkait hasil yang telah diuraikan dijelaskan dengan lebih detail dalam bentuk nilai *Precision*, *Recall* dan *F1-score* pada metode terbaik pada penelitian Ghate dan Hemalatha (2023) dan Rabbi dkk,(2021) yang dibandingkan dengan nilai *Precision*, *Recall*, dan *F1-score* pada metode usulan. Hasil perbandingan nilai *Precision*, *Recall*, dan *F1-score* dengan menggunakan dataset sekunder ditunjukkan pada Tabel 7.

Tabel 7. Perbandingan *precision*, *recall*, dan *f1-score* Algoritma SVM dan LSTM dengan Dataset Sekunder

Metode	Precision	Recall	F1-score
SVM (Ghate dan Hemalatha, 2023)	95%	92%	93%
SVM (Rabbi dkk, 2021)	95%	92%	93%
LSTM (model usulan)	97%	97%	97%

Penelitian yang dilakukan oleh Ghate & Hemalatha (2023) dan Rabbi dkk, (2021) dengan dataset sekunder menunjukkan bahwa SVM merupakan algoritma yang terbaik dalam memproses klasifikasi pada dataset yang memiliki dimensi yang besar. Namun penerapan algoritma SVM pada dataset primer tidak menghasilkan akurasi yang baik dikarenakan dataset primer memiliki bentuk *time series* dan jumlah data yang besar. Pada dataset tersebut algoritma LSTM dengan *hyperparameter* usulan memiliki hasil paling baik karena pada permasalahan klasifikasi menggunakan dataset dengan nilai sensor yang besar membutuhkan algoritma yang mampu menyimpan informasi dengan lebih panjang. Dengan begitu menggunakan algoritma LSTM, model dapat mempelajari data dengan alur maju dan mundur dan menggunakan *hyperparameter* usulan menjadikan model lebih efisien waktu dalam proses komputasi.

Perbandingan nilai *Precision*, *Recall*, dan *F1-score* juga dilakukan dengan menggunakan dataset primer. Hasil perbandingan ini ditunjukkan pada Tabel 8.

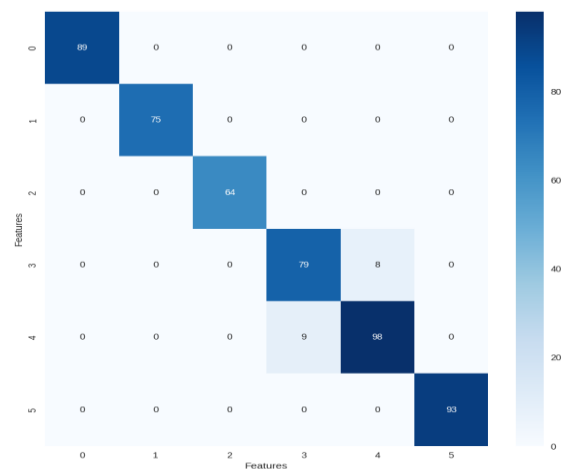
Tabel 8. Perbandingan *precision*, *recall*, dan *F1-score* Algoritma Random Forset dan LSTM dengan Dataset Primer

Metode	Precision	Recall	F1-score
Random Forest (Ghate & Hemalatha, 2023)	71%	73%	73%
Random Forest (Rabbi dkk, 2021)	76%	77%	77%
LSTM (model usulan)	96%	95%	95%

Penelitian yang dilakukan oleh Ghate & Hemalatha (2023) dan Rabbi dkk, (2021) dengan dataset primer menunjukkan bahwa *Random Forest* merupakan algoritma yang terbaik dalam memproses klasifikasi pada dataset yang memiliki dimensi yang besar. Namun penerapan algoritma *Random Forest* pada dataset primer tidak menghasilkan akurasi yang baik dikarenakan dataset primer memiliki bentuk *time series* dan jumlah data yang besar. Pada dataset tersebut algoritma LSTM dengan *hyperparameter* usulan memiliki hasil paling baik karena pada permasalahan klasifikasi menggunakan dataset

dengan nilai sensor yang besar membutuhkan algoritma yang mampu menyimpan informasi dengan lebih panjang. Penggunaan algoritma LSTM pada data sensor yang menyerupai data sinyal memberi hasil yang lebih baik pada dataset sekunder maupun primer, hasil tersebut dikarenakan LSTM mampu mengingat informasi data dengan lebih panjang sehingga dapat mempelajari data tersebut dengan lebih baik.

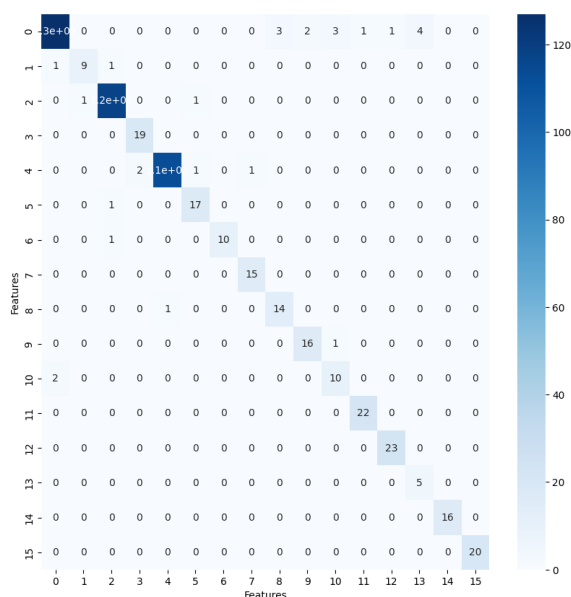
Selain hasil perbandingan pada nilai *Precision*, *Recall*, dan *F1-score* juga dicetak nilai *Confusion Matrix* dari hasil klasifikasi menggunakan LSTM pada dataset primer maupun sekunder. Hasil *Confusion Matrix* pada dataset sekunder ditunjukkan pada Gambar 7.

Gambar 7. *Confusion matrix* LSTM dengan dataset sekunder

Confusion Matrix tersebut menunjukkan bahwa hasil klasifikasi dengan menggunakan metode LSTM dengan *hyperparameter* usulan dengan menggunakan dataset sekunder. *Confusion Matrix* ini dilakukan dengan cara melakukan prediksi terhadap data uji terlebih dahulu. Kemudian dari hasil prediksi data uji dicetak nilai *Confusion Matrix*. Hasil tersebut menunjukkan bahwa model telah mampu melakukan klasifikasi dengan cukup baik. Hal ini berarti nilai prediksi dengan nilai kenyataan memiliki tingkat kesesuaian yang tinggi.

Selain hasil perbandingan pada nilai *confusion matrix* pada dataset sekunder juga dicetak nilai *confusion matrix* dari hasil klasifikasi menggunakan LSTM pada dataset primer. Hasil *confusion matrix* pada dataset sekunder ditunjukkan pada Gambar 8. Gambar 8 tersebut menunjukkan bahwa hasil klasifikasi dengan menggunakan metode LSTM dengan *hyperparameter* usulan dengan menggunakan dataset primer. Hasil *confusion matrix* tersebut juga menunjukkan bahwa metode LSTM telah mampu melakukan klasifikasi dengan baik. Namun nilai nya tak sebaik pada dataset sekunder dikarenakan dataset primer memiliki jumlah data yang lebih besar daripada dataset sekunder. Besarnya dataset pada nilai sensor *smartphone* ini apabila diterapkan pada kasus riil akan menjadi kelemahan dalam proses

klasifikasinya. Sehingga diperlukan model yang mampu mempelajari data dengan baik dan memiliki waktu komputasi yang lebih cepat.



Gambar 8. Confusion Matrix LSTM dengan dataset primer

6. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa pada beberapa paper sebelumnya algoritma SVM dan *Random Forest* memiliki hasil akurasi yang baik pada penerapan dataset publik HAR - UCI. Pernyataan tersebut sesuai dengan kegunaan SVM yang cocok digunakan pada dataset *supervised* yang memiliki dimensi besar. Namun pada penerapan dataset primer dengan jumlah data yang sangat besar dan bersifat *time series deep learning* dengan arsitektur yang disesuaikan dapat memberikan hasil yang lebih baik dan memiliki efisiensi pada waktu komputasi. Pada Penelitian ini akurasi yang didapatkan dengan menggunakan dataset sekunder dengan algoritma LSTM yaitu sebesar 97% sedangkan dengan algoritma SVM hanya mencapai 96,33%. Hasil klasifikasi dataset primer dengan algoritma LSTM memiliki akurasi sebesar 95% sedangkan algoritma *Random Forest* hanya mencapai 76,89%. Dari percobaan yang telah dilakukan dapat disimpulkan bahwa penggunaan LSTM mampu meningkatkan performa klasifikasi pada dataset sensor *accelerometer* dan *gyroscope* baik dataset dari UCI repository maupun dataset primer. Karena penggunaan LSTM mampu mengingat data dengan lebih baik pada penerapan data sensor yang bersifat *timeseries*.

Pada pengujian perbandingan algoritma, algoritma terbaik yang didapat yaitu algoritma LSTM. Namun pada penggunaan algoritma *Deep Learning* mungkin akan membutuhkan waktu komputasi lebih lama, untuk itu pengujian terhadap waktu komputasi selama pelatihan juga perlu

dilakukan untuk menghasilkan performa model optimal dengan waktu komputasi paling efisien.

Penelitian selanjutnya dapat melakukan beberapa hal untuk meningkatkan model. Pertama, dataset pada penelitian ini masih terbatas. Penelitian selanjutnya dapat mengumpulkan dataset dengan skenario yang lebih natural supaya tingkat prediksi pada kasus nyata juga memiliki hasil yang baik. Selain itu, meskipun model yang diimplementasikan saat ini sudah cukup tinggi, penelitian selanjutnya dapat mengevaluasi model LSTM yang disesuaikan dengan karakteristik aktivitas sehingga model tidak harus menproses keseluruhan data.

DAFTAR PUSTAKA

- ANGUITA D., GHIO. A., & ONETO L, 2013. A Public Domain Dataset for Human Activity Recognition Using Smartphones. *European Symposium on Artificial Neural Network, Computational Intelligent and Machine Learning*.
- M. ULLAH, H. ULLAH, S. D. KHAN and F. A. CHEIKH 2019. Stacked Lstm Network for Human Activity Recognition Using Smartphone Data. 8th European Workshop on Visual Information Processing (EUVIP), Roma, Italy, 2019, pp. 175-180, doi: 10.1109/EUVIP47703.2019.8946180.
- R. MUTAGEKI and D. S. HAN. 2020. A CNN-LSTM Approach to Human Activity Recognition. International Conference on Artificial Intelligence in Information and Communication (ICAIC), Fukuoka, Japan, 2020, pp. 362-366, doi: 10.1109/ICAIC48513.2020.9065078.
- S. YU and L. QIN. 2018. Human Activity Recognition with Smartphone Inertial Sensors Using Bidir-LSTM Networks. 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Huhhot, China, 2018, pp. 219-224. doi: 10.1109/ICMCCE.2018.00052.
- SAGHEER, A., & KOTB, M. 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, 323, 203-213. doi: 10.1016/j.neucom.2018.09.082
- CHOLISSODIN, I. K. PINASTHIKA., J. H. PRASTIWI., & R. DEWANTARA., 2021. Social Computing to Create Government Public Policy Document Blueprint Draft Based on Social Media Data About COVID-19 Using LSTM and MMR Hybrid Algorithm. *International Conference on Green Technology*, p. 2.
- YU, S., & QIN, L. (2018). Human activity recognition with smartphone inertial sensors using bidir-LSTM networks. *Proceedings - 2018 3rd International Conference on*

- Mechanical, Control and Computer Engineering, ICMCCE 2018*, 219–224.
- KHAN, A. M., LEE, Y. K., LEE, S. Y., & KIM, T. S. 2010. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE Transactions on Information Technology in Biomedicine*, 14(5), 1166–1172. <https://doi.org/10.1109/TITB.2010.2051955>
- STITSON, M. O., WESTON, J. A. E., GAMMERMAN, A., VOVK, V., & VAPNIK, V. 1996. *Theory of Support Vector Machines: Vol. +*.
- RABBI, J., FUAD, MD. T. H., & AWAL, MD. A. (2021). *Human Activity Analysis and Recognition from Smartphones using Machine Learning Techniques*. <http://arxiv.org/abs/2103.16490>
- BEVILACQUA, A., MACDONALD, K., RANGAREJ, A., WIDJAYA, V., CAULFIELD, B., & KECHADI, T. 2019. *Human Activity Recognition with Convolutional Neural Netowrks*. https://doi.org/10.1007/978-3-030-10997-4_33
- SUTOYO, E., ASRI FADLURRAHMAN, M., Telekomunikasi Jl Terusan Buah Batu, J., Dayeuhkolot, K., Bandung, K., & Barat, J. (n.d.). *JEPIN (Jurnal Edukasi dan Penelitian Informatika) Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network*.
- CHAWLA, N. V., BOWYER, K. W., HALL, L. O., & KEGELMEYER, W. P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).
- KHALILZADEH, J., & TASCI, A. D. A. 2017. Large sample size, significance level, and the effect size: Solutions to perils of using big data for academic research. *Tourism Management*, 62, 89–96. <https://doi.org/10.1016/j.tourman.2017.03.026>
- REISS, A., HENDEBY, G., & STRICKER, D. 2013. *A Competitive Approach for Human Activity Recognition on Smartphones* (Vol. 2013). <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-97302>
- SAGHEER, A., & KOTB, M. 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, 323, 203–213. <https://doi.org/10.1016/j.neucom.2018.09.082>
- AINI, N., SINURAT, S., & ADELINA HUTABARAT, S. 2018. Penerapan Metode Simple Moving Average Untuk Memprediksi Hasil Laba Laundry Karpet Pada CV. Homecare. In *Jurnal Riset Komputer (JURIKOM)* (Vol. 5, Issue 2). <http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom>[Page]167
- ZHANG, M., & SAWCHUK, A. A. 2012. A feature selection-based framework for human activity recognition using wearable multimodal sensors. *BODYNETS 2011 - 6th International ICST Conference on Body Area Networks*, 92–98. <https://doi.org/10.4108/icst.bodynets.2011.247018>
- YULIANTI, R., PASEK, G., WIJAYA, S., & BIMANTORO, D. F. (N.D.). *Pengenalan Pola Tulisan Tangan Suku Kata Aksara Sasak Menggunakan Metode Moment Invariant dan Support Vector Machine (Handwritten Sasak Ancient Script Recognition using Moment Invariant and Support Vector Machine)*. <http://jcosine.if.unram.ac.id/>
- JISHAN, S. T., RASHU, R. I., HAQUE, N., & RAHMAN, R. M. 2015. Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique. *Decision Analytics*, 2(1). <https://doi.org/10.1186/s40165-014-0010-2>
- GHATE, V., & HEMALATHA C, S. 2023. A comprehensive comparison of machine learning approaches with hyper-parameter tuning for smartphone sensor-based human activity recognition. *Measurement: Sensors*, 30. <https://doi.org/10.1016/j.measen.2023.100925>
- BACHTIAR, F. A., ARIFIEN, Z., PUTRA, A. R., & AKBAR, P. I. 2021. Feature Selection with Particle Swarm Optimization for Human Activity Recognition Using Learning Vector Quantization. *ACM International Conference Proceeding Series*, 132–138. <https://doi.org/10.1145/3479645.3479693>

Halaman ini sengaja dikosongkan