Prediksi Kemungkinan Hujan dengan menggunakan Algoritma *Machine Learning*

Hafizha Dini Giandra
Department of Informatics
Universitas Syiah Kuala
Banda Aceh, Aceh
giandra.23@mhs.usk.ac.id

Fadhilah Syafa
Department of Informatics
Universitas Syiah Kuala
Banda Aceh, Aceh
fadhilah.s@mhs.usk.ac.id

Mutaqqin
Department of Informatics
Universitas Syiah Kuala
Banda Aceh, Aceh
muttaqin4@mhs.usk.ac.id

***Abstract*—Prediksi hujan adalah topik penting yang terus menarik perhatian di seluruh dunia. Dalam era informasi yang berkembang pesat, pemahaman dan prediksi kondisi cuaca memiliki peran penting dalam berbagai aspek kehidupan manusia, termasuk sektor pertanian, transportasi, dan manajemen risiko bencana. Penelitian sebelumnya tentang pemodelan hujan dengan menggunakan teknik *machine learning* memiliki kesulitan memprediksi apakah akan terjadi hujan karena model terlalu lemah. Penelitian ini bertujuan untuk mengetahui pemilihan algortima *machine learning* yang tepat dalam memprediksi hujan. Terdapat empat algoritma yang digunakan dan dibandingkan dalam penelitian ini yaitu *Decision Tree* (DT), *Random Forest* (RF), *Logistic Regression* (LR), dan *Long Short-Term Memory* (LSTM). Hasil eksperimen menunjukkan bahwa algoritma LR mampu memberikan akurasi terbaik (hingga 0.797), namun tidak terdapat perbedaan nilai akurasi yang signifikan antar metode yang digunakan dan tidak ada yang menghasilkan nilai akurasi lebih dari 0.80. Selain itu, hasil eksperimen ini juga tidak lebih baik dari ekperimen sebelumnya dan hal tersebut dapat disebabkan oleh proses *pre-processing* pada *dataset* yang digunakan.**

***Keywords*— *machine learning*, prediksi hujan, *decision tree*, *random forest*, *logistic regression*, LSTM**

I. Introduction

Prediksi hujan masih menjadi tantangan besar bagi para ahli iklim. Hujan adalah komponen terpenting dari sistem iklim. Proses memprediksi terjadinya hujan di suatu wilayah memperhitungkan keakuratan prediksi, kesalahan dalam prediksi, dan estimasi volume curah hujan serta kemungkinan curah hujan di wilayah tertentu [1], sehingga untuk mengetahui hal-hal tersebut perlu mengumpulkan, menganalisis, dan melakukan penelitian terhadap berbagai data dan parameter meteorologi yang tersedia. Beberapa parameter dasar termasuk suhu (minimum dan maksimum), total curah hujan tahunan, dan kelembaban. Curah hujan dapat bervariasi dalam intensitas, variabilitas, dan frekuensi.

Kemungkinan terjadinya hujan sangat mempengaruhi alur kerja di berbagai bidang seperti perencanaan produksi pangan, pengelolaan sumber daya air, prediksi longsor dan/atau banjir, bahkan hal-hal trivial seperti perencanaan kegiatan sehari-hari terutama kegiatan *outdoor*. Prediksi hujan yang akurat dan tepat waktu diharapkan dapat memberikan tahap intervensi baru kepada sektor-sektor yang terkena dampak dari hujan. Namun, untuk perencanaan kegiatan sehari-hari, informasi apakah akan hujan atau tidak saja sudah cukup.

Permasalahan inilah yang menginspirasi kami untuk melakukan klasifikasi terkait kemungkinan hujan dengan menggunakan metode machine learning. Machine learning secara umum dapat didefinisikan sebagai metode komputasi yang menggunakan pengalaman untuk meningkatkan kinerja atau membuat prediksi yang akurat. Pengalaman pada machine learning mengacu pada informasi terdahulukan yang siap untuk dianalisis oleh sebuah metode. Machine learning memungkinkan komputer untuk menemukan solusi secara mandiri tanpa perlu diprogram secara eksplisit.

Penelitian ini dilakukan dengan menerapkan empat metode *machine learning*, yaitu *Decision Tree* (DT), *Random Forest* (RF), *Logistic Regression* (LR), dan *Long Short-Term Memory* (LSTM) pada histori data cuaca dari *dataset* yang kami dapat. Performa diantara metode tersebut sangat bervariasi, sehingga memberikan ruang untuk peningkatan dengan memvariasikan rasio pelatihan dan pengujian [2]. Oleh karena itu, pemilihan metode *machine learning* yang sesuai dalam mengklasifikasikan hujan di suatu wilayah sangat penting.

Hasil dari penelitian ini adalah untuk mengetahui apakah dengan membandingkan metode *Decision Tree*, *Random Forest*, *Logistic Regression*, dan LSTM dapat digunakan untuk membantu memprediksi hujan.

II. Related Works

Pada penelitian sebelumnya yang dilakukan oleh Anwar et al. (2020), membandingkan kinerja empat algoritma dari *machine learning*, yaitu J48, *Random Forest*, *Naïve Bayes*, dan *Multilayer Perceptron* (MLP) untuk prediksi hujan. Data histori cuaca harian diperoleh dari situs BMKG untuk stasiun meteorologi di Tanjung Mas, Semarang, Indonesia, mulai tahun 2013 hingga 2019. Hasil dari penelitian ini menunjukkan bahwa algoritma MLP dan J48 dapat memberikan akurasi terbaik (hingga 78.4%) dibandingkan dengan algoritma lain, walaupun perbedaannya kecil [3].

Sanie et al. (2020) menggunakan *ensemble learning* untuk meningkatkan efektivitas prediksi curah hujan. *Ensemble learning* adalah pendekatan yang menggabungkan beberapa pengklasifikasi *machine learning* untuk prediksi curah hujan, yang mencakup *Naïve Bayes*, *Decision Tree*, *Support Vector Machine, Random Forest*, dan *Neural Network* berdasarkan dataset yang diperoleh dari *Drainage and Irrigation Department* dan *Malaysian Meteorological Department*. Lebih khusus lagi, penelitian ini mengeksplorasi tiga penggabungan aljabar, yaitu probabilitas rata-rata (*average probability*), probabilitas maksimum (*maximum probability*), dan pemungutan suara mayoritas (*majority voting*). Hasil menunjukkan bahwa metode *ensemble* berdasarkan *majority voting* sangat efektif dalam meningkatkan performa prediksi curah hujan dibandingkan dengan klasifikasi individual [4]. Nilai dari *precision*, *recall*, dan *F-Measures* yang diperoleh dari *majority voting* adalah 71%, 77%, dan 76% dimana hasil tersebut lebih tinggi dibandingkan dengan penggabung aljabar lainnya.

Penelitian selanjutnya oleh Zhao et al. (2022) juga melakukan prediksi hujan berbasis *machine learning* menggunakan *dataset Rain in Australia*. Metode *machine learning* yang digunakan adalah *Decision Tree*, *Logistic Regression*, *Long Short Term Memory* (LSTM), *AdaBoost*, *Bagging Algorithm*, dan *K-Nearest Neighbor* (KNN) dengan K *value* dalam *range* 1 hingga 20. Penelitian tersebut memperoleh

akurasi tertinggi yaitu 85% dengan metode *Logistic Regression* diikuti oleh metode *AdaBoost* dengan akurasi 82%, akan tetapi hasil tersebut tidak sesuai dengan ekspektasi. Terdapat beberapa hipotesis tentang alasan mengapa akurasi tersebut tidak dapat ditingkatkan lebih lanjut. Penggabungan beberapa model untuk isu tertentu merupakan cara yang efektif untuk mengklasifikasikan. Jadi salah satu alasan mengapa akurasi tidak dapat ditingkatkan adalah karena model terlalu lemah dan orisinal, sehingga sampai batas tertentu masih belum dapat memprediksi apakah akan terjadi hujan.

Hudnurkar and Rayavarapu (2022) mengeksplorasi penggunaan algoritma *Support Vector Machine* (SVM) dan *Artificial Neural Network* (ANN) untuk klasifikasi biner curah hujan. Data klasifikasi curah hujan harian yang digunakan dalam penelitian ini diperoleh dari *National Data Center* (NDC) IMD. *Dataset* tersebut diperoleh dari tahun 2000 hingga 2018 untuk stasiun Shivajinagar di Pune, Maharashtra, India. Akurasi klasifikasi yang diperoleh 82.1% dan 82.8% dengan algoritma SVM dan ANN, untuk *dataset* yang tidak seimbang. Sementara parameter kinerja seperti *misclassification rate* dan *F1-score* menunjukkan bahwa hasil yang lebih baik dicapai dengan ANN, pemilihan parameter model untuk SVM kurang terlibat dibandingkan ANN [7]. Teknik adaptasi domain digunakan untuk klasifikasi curah hujan di dua stasiun lainnya di Nashik dan Chikalthana dengan jaringan yang dilatih untuk stasiun Shivajinagar. Hasil yang memuaskan kedua stasiun ini baru diperoleh setelah adanya perubahan metode pelatihan dari SVM dan ANN.

III. Methodology

Machine learning adalah salah satu teknologi terkini yang paling menarik. *Machine learning* telah diposisikan untuk mengatasi kekurangan kognisi manusia serta pemrosesan informasi, khususnya dalam menangani data yang besar, hubungannya, dan analisisnya. Secara umum, *machine learning* mempelajari penelitian dan konstruksi algoritma yang dapat dipelajari, dan memperoleh prediksi tentang data. Oleh karena itu, pendekatan *machine learning* dipilih untuk memprediksi hujan.

Metodologi penelitian yang digunakan dalam penelitian ini dibagi menjadi empat fase berbeda. Fase pertama adalah fase *dataset*, dimana mengidentifikasi data secara manual dengan menganalisis sumber, jumlah, dan detail lainnya. Selanjutnya, fase *pre-processing* yaitu mempersiapkan data untuk diproses lebih lanjut dengan membersihkan data (seperti, mengatasi *missing value*) dan menormalkan data. Data yang telah diproses sebelumnya kemudian digunakan pada fase ketiga untuk dianalisis menggunakan empat metode *machine learning* guna mengidentifikasi metode terbaik dari empat pengklasifikasi *machine learning* tersebut. Fase keempat dan terakhir berfokus pada penilaian terhadap kinerja metode. Masing-masing dari keempat fase ini dijelaskan lebih lanjut pada subbagian berikut.

A. Dataset

Dataset yang digunakan pada penelitian ini berasal dari situs Kaggle, yaitu *Rain in Australia dataset*. *Dataset* ini berisi hasil observasi cuaca harian dari berbagai lokasi di Australia selama 10 tahun (2007-2017). *Dataset* tersebut memiliki 145.640 baris data.

Terdapat 23 fitur pada *dataset* ini seperti lokasi, suhu, kecepatan angin, kelembaban, dan lain sebagainya. Tujuh diantara fitur-fitur tersebut termasuk label kelas merupakan data nominal dan sisanya adalah data numerik.

B. Pre-processing

Sebelum proses klasifikasi dilakukan, data atau *record* dan atribut harus melalui beberapa tahap pengolahan awal data (*pre-processing*).

Data yang tidak lengkap atau hilang menimbulkan ketidakadilan atau bahkan kesalahan, oleh karena itu untuk memperoleh data yang berkualitas dilakukan beberapa tahapan *pre-processing* sebagai berikut.

- Semua atribut nominal, kecuali label kelas, disingkirkan karena tidak relevan dengan proses klasifikasi. Atribut-atribut tersebut adalah "*Date*", tanggal observasi; "*Location*", lokasi data diambil; "*WindGustDir*", arah kecepatan angin tertinggi dalam periode 24 jam; "*WindDir9am*", arah angin pada jam 9 pagi; "*WindDir3pm*", arah angin pada jam 3 sore; dan "*RainToday*", apakah hari ini hujan atau tidak. *Dataset* akhir memiliki semua atribut numerik, kecuali label kelas yang nominal. Atribut akhir dalam *dataset* ditunjukkan pada Table I.
- Pengambilan baris data tanpa nilai *null* atau *NaN* pada semua kolom. Dari total 145.460 baris dalam *dataset* mentah didapat 58.090 baris (sekitar 40% dari jumlah awal) yang sama sekali tidak memiliki nilai *null* atau *NaN* pada semua kolom.
- Dari hasil *pre-prosesing* sebelumnya kelas "*Yes*" memiliki sampel sebanyak 12.729 baris dan kelas "*No*" memiliki sampel sebanyak 45.361 baris. Seluruh baris dengan kelas "*Yes*" digunakan dan diambil sebanyak 12.729 baris dari kelas "*No*" untuk digunakan. Jumlah total dataset yang siap digunakan adalah 25.458 baris dengan 16 atribut numerik dan 1 atribut label kelas.
- Normalisasi dataset dilakukan dengan menggunakan metode *MinMaxScaler* dari *library Sklearn*. Fungsi *MinMaxScaler* mengubah seluruh nilai data dalam *range* antara 0 hingga 1.
- Pemisahan data *training* dan data *testing* dengan rasio 70:30.

TABLE I
ATRIBUT PADA *Dataset*

Atribut	Tipe Data	Deskripsi
MinTemp	Numerik	Suhu minimum (°C)
MaxTemp	Numerik	Suhu maksimum (°C)
Rainfall	Numerik	Jumlah curah hujan pada hari itu (mm)
Evaporation	Numerik	Penguapan panci Kelas A (mm)
Sunshine	Numerik	Waktu pemaparan sinar matahari (jam)
WindGustSpeed	Numerik	Kecepatan hembusan angin terkuat (km/jam)
WindSpeed9am	Numerik	Kecepatan angin rata-rata selama 10 menit sebelum pukul 9 pagi (km/jam)
WindSpeed3pm	Numerik	Kecepatan angin rata-rata selama 10 menit sebelum pukul 3 sore (km/jam)
Humidity9am	Numerik	Kelembaban pada pukul 9 pagi (%)
Humidity3pm	Numerik	Kelembaban pada pukul 3 sore (%)
Pressure9am	Numerik	Tekanan atmosfer yang berkurang pada pukul 9 pagi (hpa)
Pressure3pm	Numerik	Tekanan atmosfer yang berkurang pada pukul 3 sore (hpa)
Cloud9am	Numerik	Sebagian langit tertutup awan pada pukul 9 pagi (oktas)
Cloud3pm	Numerik	Sebagian langit tertutup awan pada pukul 3 sore (oktas)
Temp9am	Numerik	Suhu pada pukul 9 pagi (°C)
Temp3pm	Numerik	Suhu pada pukul 3 sore (°C)
RainTomorrow	Nominal	Label kelas

C. Decision Tree

Decision tree (DT) atau pohon keputusan merupakan salah satu metode *machine learning* sederhana untuk klasifikasi dan regresi. Metode ini melakukan proses klasifikasi data berdasarkan fitur-fitur [5]. *Decision tree* belajar dari sekumpulan pertanyaan *if/else* atau *yes/no*, sehingga dapat membentuk sebuah pohon hirarki. Setiap *decision* menyebabkan *decision* lain atau prediksi [7]. Keuntungan metode ini adalah alur algoritmanya mudah dibaca dan dimengerti, serta kecepatan klasifikasi yang cepat.

Saat proses *training*, metode ini menganalisis data *training* untuk membuat model dengan menggunakan prinsip meminimalkan fungsi kerugian (*loss function*). Proses *training* metode ini secara umum dilakukan dalam tiga tahap, yaitu pemilihan fitur, pembuatan *decision tree*, dan *pruning* (pemangkasan). Berikut alur kerja proses pembelajaran metode *decision tree*.

1) Inisialisasi *root node* (akar).
2) Melakukan partisi *node* menggunakan atribut dengan nilai *information gain* tertinggi.
3) Lakukan proses (2) secara berulang hingga:
   - Semua sampel pada *sub-node* tersebut memiliki label kelas yang sama.
   - Mencapai kondisi yang telah ditetapkan sebelumnya, seperti maksimum kedalaman, atau batasan minimum sampel yang diperlukan untuk proses partisi.
   - Semua atribut telah digunakan untuk proses partisi.

Klasifikasi dengan metode DT dilakukan dengan menggunakan fungsi *DecisionTreeClassifier* dalam *library Sklearn*. Evaluasi hasil klasifikasi dilakukan dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

D. Random Forest

Random forest (RF) adalah metode pengembangan *decision tree* yang banyak digunakan untuk berbagai tujuan, termasuk regresi, klasifikasi, dan prediksi [4]. *Random forest* merupakan kumpulan *decision tree* yang dibangun dengan nilai parameter, seleksi fitur, dan jumlah sampel yang acak dan independen antar pohon. Pembangunan pohon dengan cara ini mengecilkan kemungkinan terjadinya *overfitting* terhadap model yang dihasilkan [3]. Seringnya terjadi *overfitting* merupakan salah satu kelemahan dari metode *decision tree*.

Pada awal pembentukan *tree*, *random forest* akan sama dengan *decision tree* dan selanjutnya akan mengelompokkan *weak learners* bersama-sama untuk membentuk sebuah *strong learner* [7]. Jadi *random forest* menggunakan berbagai teknik untuk menentukan keputusan akhir. Pada kasus klasifikasi, akan dibentuk kelas-kelas dan penentuan kelas diambil berdasarkan hasil *voting* dari *tree* yang terbentuk [7] dan pemenangnya akan ditentukan oleh *vote* terbanyak.

Klasifikasi dengan metode RF dilakukan dengan menggunakan fungsi *RandomForestClassifier* dalam *library Sklearn*. Evaluasi hasil klasifikasi dilakukan dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

E. Logistic Regression

Logistic regression (LR) adalah metode statistik yang biasanya digunakan untuk klasifikasi biner, yang berarti metode ini digunakan ketika variabel bersifat kategorikal atau diskrit dan hanya memiliki dua label kelas, seperti *Yes* atau *No*, 0 atau 1, dan *true* atau *false*. *Logistic regression* dalam statistika sering digunakan untuk melakukan prediksi probabilitas kejadian suatu peristiwa [7].

Metode ini menggunakan fungsi logistik (juga dikenal sebagai fungsi *sigmoid*) untuk mengubah hasil perhitungan linear dari fitur masukan menjadi nilai antara 0 atau nilai 1 [7]. Nilai ambang batas (*threshold*) yang digunakan biasanya adalah 0.5. Jika nilai prediksi lebih besar atau sama dengan 0.5 maka data tersebut akan diklasifikasikan menjadi kelas 1, jika tidak maka akan diklasifikasikan menjadi kelas 0.

Kelebihan dari *logistic regression* adalah metode implementasinya sederhana dibandingkan dengan metode lain dan waktu pelatihannya lebih sedikit, meluas ke beberapa prediksi yang disebut multinominal regression, jika datanya dapat dipisahkan secara linier sehingga memberikan akurasi yang baik.

Klasifikasi dengan metode LR dilakukan dengan menggunakan fungsi *LogisticRegression* dalam *library Sklearn*. Klasifikasi dengan metode ini juga akan dilakukan dengan menggunakan kode yang dibuat dari awal berdasarkan alur kerja metode *logistic regression* tanpa menggunakan *library Sklearn* yang didapat dari pembelajaran mata kuliah PDSAI. Evaluasi hasil klasifikasi keduanya dilakukan dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

F. Long Short Term Memory

Long Short Term Memory (LSTM) merupakan algoritma *deep learning* yang populer dan cocok digunakan untuk membuat prediksi dan klasifikasi yang berhubungan dengan waktu [5]. LSTM mula-mula diusulkan oleh Sepp Hochreiter dan Jürgen Schmidhuber pada tahun 1997[8].

Algoritma ini merupakan pengembangan dari algoritma RNN (*Recurrent Neural Network*) [7]. Dalam algoritma RNN, *output* dari langkah terakhir diumpankan kembali sebagai *input* pada langkah yang sedang aktif. Namun, algoritma RNN memiliki kesulitan mempelajari dan menyimpan informasi dalam rangkaian yang panjang karena masalah gradien hilang, yang mana gradien kesalahan terhadap bobot menjadi sangat kecil, menyebabkan model kesulitan mempelajari informasi jauh di masa lalu. Algoritma LSTM mampu menyimpan informasi untuk jangka waktu yang lama. Hal ini kemudian dapat digunakan untuk memproses, memprediksi, dan mengklasifikasikan informasi berdasarkan data deret waktu.

Struktur algoritma LSTM terdiri atas *neural network* dan beberapa blok memori yang berbeda. Blok memori ini disebut sebagai *cell*. *State* dari *cell* dan *hidden state* akan diteruskan ke *cell* berikutnya. Informasi yang dikumpulkan oleh algoritma LSTM kemudian akan disimpan oleh *cell* dan manipulasi memori dilakukan oleh komponen yang disebut dengan *gate*. Ada tiga jenis *gate* pada algoritma LSTM, di antaranya *forget gate*, *input gate*, dan *output gate*.

Klasifikasi dengan metode LSTM dilakukan dengan menggunakan *ensemble* fungsi LSTM dan fungsi *Dense* dalam *library TensorFlow*. Fungsi *Dense* digunakan untuk memutuskan kelas sampel dengan menggunakan fungsi *sigmoid*. Evaluasi hasil klasifikasi dilakukan dengan menghitung nilai *accuracy*, *precision*, *recall*, dan *F1-score*.

G. Evaluation Metrics

Untuk tujuan mengevaluasi metode yang diusulkan, *information retrieval metrics* digunakan. Evaluasi dilakukan dengan menggunakan beberapa pengukuran yaitu *accuracy*, *precision*, *recall*, dan *F1-score*. Semua pengukuran ini didasarkan pada *false positive* (FP), *false negative* (FN), *true positive* (TP), dan *true negative* (TN). *Precision* dan *recall* adalah pengukuran yang diperlukan untuk menunjukkan performa model untuk kelas tertentu (yang sangat berguna dalam *dataset* dengan kelas tidak seimbang, yang tidak dapat diketahui dengan *accuracy*.

Model yang didapat akan memprediksi 2 kelas, yaitu "*Yes*" yang berarti besok akan hujan dan "*No*" yang berarti besok tidak hujan. R2, SSE, dan MSE lebih baik untuk nilai kontinu, sedangkan model dalam projek ini tidak memprediksi keluaran seperti itu.

Accuracy didefinisikan sebagai total dari hasil yang diklasifikasikan dengan benar (TP dan TN) dibagi dengan semua hasil pengujian. *Accuracy* dapat dihitung menggunakan (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision mengevaluasi hasil *true positive* (TP) terhadap total hasil yang diprediksi positif (TP dan FP). *False positive* (FP) merupakan entitas yang diklasifikasikan secara salah, yang dapat dihitung menggunakan (2).

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall digunakan dalam menilai hasil positif yang diklasifikasikan dengan benar (TP) terhadap total hasil yang positif pada kenyataannya (TP dan FN). Evaluasi ini dihitung seperti yang ditunjukkan pada (3).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Terakhir, *F1-score* dapat dilihat secara sederhana sebagai rata-rata dari *recall* dan *precision* yang menunjukkan performa model secara keseluruhan, dapat dihitung seperti yang ditunjukkan pada (4).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

IV. Result and Discussion

Empat metode klasifikasi yang digunakan yaitu *Decision Tree* (DT), *Random Forest* (RF), *Logistic Regression* (LR), dan *Long Short-Term Memory* (LSTM) untuk *dataset Rain in Australia* dengan jumlah sampel sebanyak 25.458 baris dengan 16 atribut numerik dan 1 atribut label kelas. Tahapan klasifikasi dataset ini dilakukan dengan bahasa pemrograman *Python3* menggunakan *Integrated Development Environments* (IDE) *Jupyter Lab*.

Pada penelitian ini dilakukan *tuning* parameter dengan mengidentifikasi kumpulan parameter terbaik dari setiap model klasifikasi akan digunakan. *Tuning* parameter yang berbeda digunakan untuk menyetel setiap pengklasifikasi agar menghasilkan hasil yang sangat akurat. Serangkaian percobaan dilakukan untuk mendapatkan nilai optimal dari masing-masing pengklasifikasi. Hasil kinerja dari empat pengklasifikasi kemudian dievaluasi dengan menggunakan *accuracy*, *precision*, *recall*, dan *F1-score*.

Klasifikasi dengan metode DT dilakukan dengan menggunakan fungsi *DecisionTreeClassifier* dalam *library Sklearn* dengan menggunakan beberapa nilai parameter untuk mendapatkan hasil klasifikasi yang paling baik. Parameter-parameter yang disesuaikan adalah *criterion*, *max_depth*, *min_samples_leaf*, dan *max_features*. Nilai-nilai yang diujicobakan untuk parameter *criterion* adalah *entropy*, *gini*, dan *log_loss*; *max_depth* adalah 4, 5, 6, dan 7; *min_samples_leaf* adalah 50, 100, 150, 200, dan 250; dan *max_features* adalah 9, 11, 13, dan 15. Hasil klasifikasi terbaik didapat dengan nilai *criterion* = *gini*, *max_depth* = 6, *min_samples_leaf* = 150, dan *max_features* = 13, dengan nilai akurasi sebesar 0.786 dan *confusion matrix* yang dihasilkan dapat dilihat pada Fig. 1.

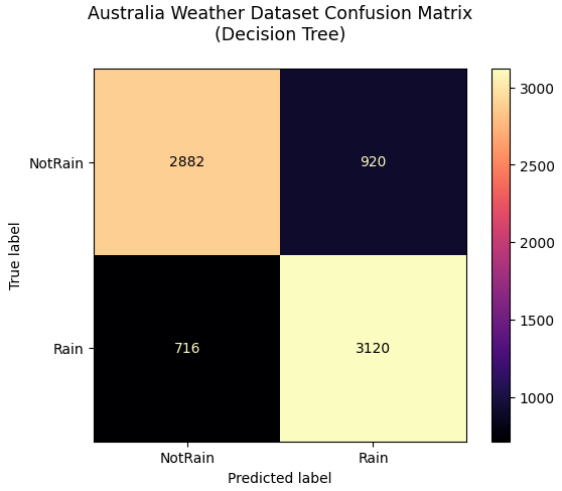

Fig. 1. Hasil *confusion matrix* menggunakan *Decision Tree*

Klasifikasi dengan metode RF dilakukan dengan menggunakan fungsi *RandomForestClassifier* dalam *library Sklearn* dengan menggunakan beberapa nilai parameter untuk mendapatkan hasil klasifikasi yang paling baik. Parameter-parameter yang disesuaikan adalah *n_estimators*, *criterion*, *max_depth*, *min_samples_leaf*, dan *max_features*. Nilai-nilai yang diujicobakan untuk parameter *n_estimator* adalah 100, 150, 200, dan 250; *criterion* adalah *entropy*, *gini*, dan *log_loss*; *max_depth* adalah 4, 5, 6, 7, dan 8; *min_samples_leaf* adalah 50, 100, 150, 200, dan 250; dan *max_features* adalah 9, 11, 13, dan 15. Hasil klasifikasi terbaik didapat dengan nilai *n_estimator* = 250, *criterion* = *entropy*, *max_depth* = 7, *min_samples_leaf* = 100, dan *max_features* = 11, dengan nilai akurasi sebesar 0.796 dan *confusion matrix* yang dihasilkan dapat dilihat pada Fig. 2.

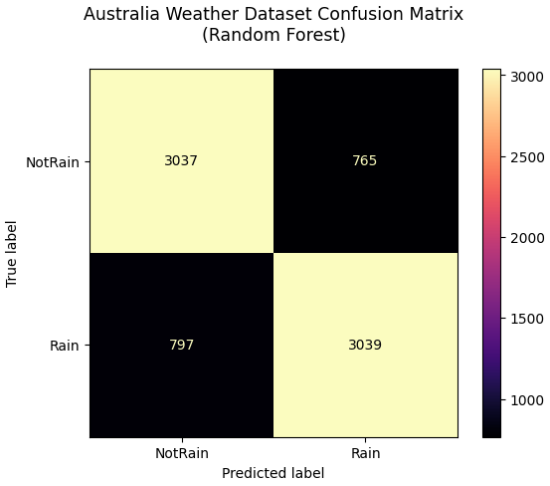

Fig. 2. Hasil *confusion matrix* menggunakan *Random Forest*

Klasifikasi dengan metode LR dilakukan dengan menggunakan dua cara, yaitu menggunakan fungsi *LogisticRegression* dalam *library Sklearn* dan menggunakan kode tanpa bantuan *library Sklearn*. Pada cara yang pertama, proses klasifikasi dengan fungsi *LogisticRegression* menggunakan beberapa nilai parameter untuk mendapatkan hasil klasifikasi yang paling baik. Parameter-parameter yang disesuaikan adalah *max_iter*, *solver*, dan *tol*. Nilai-nilai yang diujicobakan untuk parameter *max_iter* yaitu 250, 500, 750, dan 1000; *solver* yaitu *lbfgs*, *liblinear*, *newton-cg*, *newton-cholesky*, *sag*, dan *saga*; dan *tol* yaitu 0,005, 0,001, 0,0005, dan 0,0001. Hasil klasifikasi terbaik yang didapat adalah dengan *max_iter* = 250, *solver* = *sag*, dan *tol* = 0,005, dengan nilai akurasi 0.797 dan *confusion matrix* pada Fig. 3.

Selanjutnya cara yang kedua yaitu tanpa *library Sklearn*. Kode tersebut dihasilkan dari pembelajaran materi *logistic regression* pada mata kuliah PDSAI. Proses klasifikasi menggunakan kode tanpa bantuan *Sklearn* menggunakan beberapa parameter, yaitu *alpha*, *epochs*, dan *batch_size*. Nilai-nilai yang diujicobakan untuk parameter *alpha* yaitu 0,005, 0,001, 0,0005, dan 0,0001; *epochs* yaitu 2500, 5000, 7500, dan 10000; dan *batch_size* yaitu 5%, 10%, 15% dan 20%. Hasil klasifikasi terbaik yaitu dengan *alpha* = 0,0005, *epochs* = 10000, dan *batch_size* = 10%, dengan nilai akurasi 0.796 dan *confusion matrix* pada Fig. 4.

Klasifikasi dengan metode LSTM dilakukan dengan menggunakan beberapa fungsi dalam *library TensorFlow*. Pertama fungsi LSTM digabungkan dengan fungsi *Dense* menggunakan fungsi *Sequential* dengan parameter *units* dan *activation*. Hasil sekuen tersebut kemudian di*compile* dengan parameter yang digunakan adalah *optimizer*. Pada fungsi LSTM yang digunakan nilai-nilai yang diujicobakan untuk parameter *units* adalah 16, 32, dan 64, untuk *activation* digunakan *relu* dan *tanh*. Pada saat *compile* nilai-nilai yang diujicobakan untuk parameter *optimizer* adalah *adam* dan *nadam*. Kemudian pada proses *fitting* nilai-nilai yang diujicobakan untuk parameter *epochs* adalah 10, 15, dan 20; dan *batch_size* adalah 16, 32, dan 64. Hasil klasifikasi terbaik didapat dengan menggunakan nilai *units* = 64, *activation* = *tanh*, *optimizer* = *adam*, *epochs* = 20, dan *batch_size* = 16, dengan nilai akurasi sebesar 0.793 dan *confusion matrix* yang dihasilkan dapat dilihat pada Fig. 5.

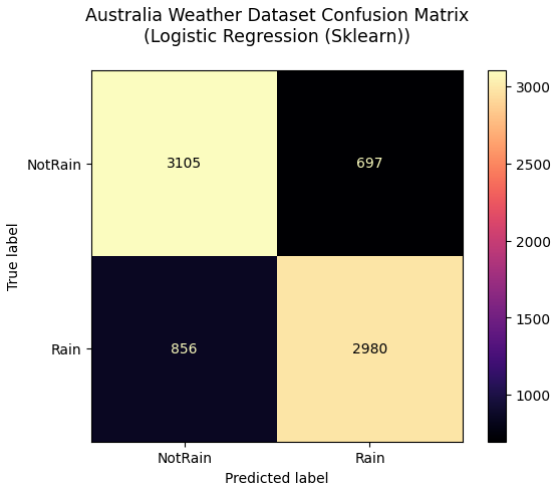

Fig. 3. Hasil *confusion matrix* menggunakan *Logistic Regression* (*Sklearn*)

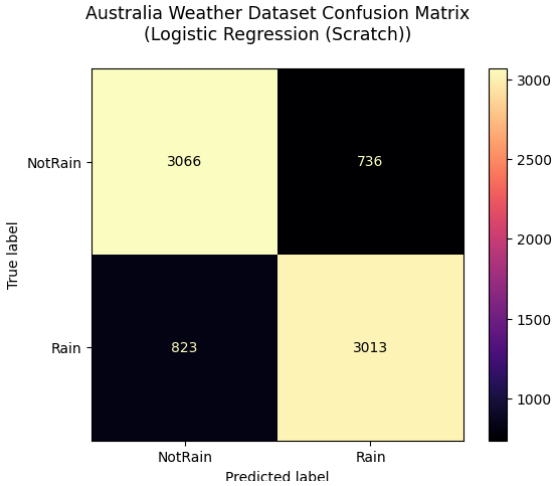

Fig. 4. Hasil *confusion matrix* menggunakan *Logistic Regression* (*Code*)

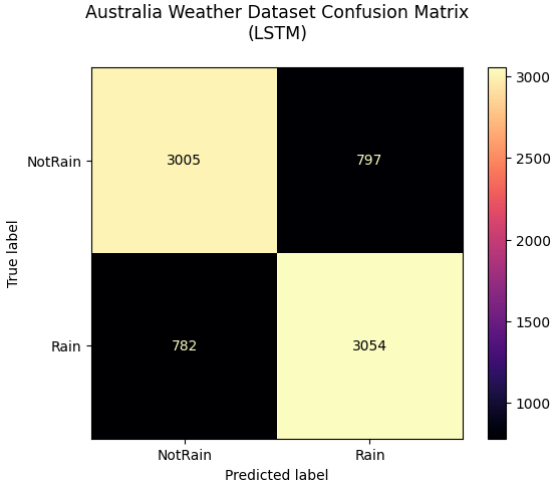

Fig. 5. Hasil *confusion matrix* menggunakan LSTM

Nilai *accuracy*, *precision*, *recall*, dan *F1-score* dari hasil terbaik pada keempat metode klasifikasi yang telah dilakukan dapat dilihat pada Table II.

TABLE II
HASIL TERBAIK MODEL KLASIFIKASI

Metode	***Recall***	***Precision***	***F1-score***	***Accuracy***
DT	0.813	0.772	0.792	0.786
RF	0.792	0.799	0.796	0.796
LR (*Sklearn*)	0.777	0.810	0.793	0.797
LR (*Code*)	0.785	0.804	0.794	0.796
LSTM	0.796	0.793	0.795	0.793

Eksperimen ini menghasilkan nilai akurasi dari masing-masing metode yang tidak jauh berbeda, namun tidak ada yang menghasilkan nilai lebih dari 0.80. Kemiripan nilai-nilai tersebut dapat dijadikan dasar perkiraan bahwa nilai akurasi yang dihasilkan bukan dipengaruhi oleh metode klasifikasi dan nilai parameter, tapi oleh *dataset* yang digunakan.

Nilai akurasi yang dihasilkan dari ekperimen ini tidak lebih baik dari hasil eksperimen yang dilakukan oleh Zhao et al. (2022). Pada eksperimen ini dan pada eksperimen yang dilakukan oleh Zhao et al. menggunakan tiga metode yang sama dari beberapa metode yang digunakan, yaitu DT, LR, dan LSTM. Eksperimen Zhao et al. menghasilkan nilai akurasi sebesar 78% dengan metode DT dan 85% untuk metode LR dan LSTM. Hal ini dapat disebabkan oleh *pre-processing* yang dilakukan sebelum dilakukan proses klasifikasi oleh masing-masing eksperimen.

Zhao et al. menggunakan seluruh atribut yang terdapat pada *dataset*, termasuk atribut nominal yang diubah ke bentuk numerik dengan proses *encoding* (20 atribut) [5], sedangkan eksperimen ini hanya menggunakan atribut-atribut numerik (16 atribut). Selain itu Zhao et al. juga melakukan *Exploratory Data Analysis* (EDA) sehingga dapat mengetahui tingkat korelasi antar atribut yang digunakan sebagai dasar dalam menentukan bobot untuk masing-masing atribut [5], sedangkan eksperimen ini tidak menggunakan bobot dalam proses pengklasifikasiannya. Kemungkinan lain adalah sedikitnya jumlah sampel yang digunakan dalam proses klasifikasi. Sebelum dilakukan *pre-processing* data, *dataset* mentah memiliki jumlah sampel sebanyak 145.460 baris. Setelah dilakukan *pre-processing*, hasil akhir sampel yang digunakan berjumlah 25.458 baris atau hanya 17.5% dari jumlah baris data mentah.

V. CONCLUSION

Eksperimen ini melakukan pengklasifikasian pada dataset Rain in Australia dengan menggunakan empat metode klasifikasi yaitu *Decision Tree* (DT), *Random Forest* (RF), *Logistic Regression* (LR), dan *Long Short-Term Memory* (LSTM). Nilai akurasi tertinggi diperoleh dari metode *Logistic Regression*, yaitu sebesar 0.797, namun tidak ada perbedaan nilai akurasi yang signifikan antara metode-metode yang digunakan, dengan selisih nilai dengan akurasi terendah hanya 0.011. Penulis memperkirakan bahwa hal ini bukan disebabkan oleh metode yang digunakan, namun oleh *dataset*nya.

Selain itu, hasil eksperimen ini juga tidak lebih baik dari ekperimen sebelumnya, yaitu eksperimen oleh Zhao et al. (2022). Penulis memperkirakan bahwa hal ini disebabkan oleh perbedaan *pre-processing* yang dilakukan dan sedikitnya jumlah sampel yang digunakan relatif dengan jumlah sampel *dataset* mentah. Diharapkan dalam penelitian lanjutan dapat dilakukan *pre-processing* yang lebih *advance* agar data yang digunakan menghasilkan nilai akurasi yang lebih baik.

ACKNOWLEDGMENT

Penulis mengucapkan terima kasih kepada Prof. Dr. Taufik Fuadi Abidin, S.Si., M.Tech (Universitas Syiah Kuala) yang telah mendukung penelitian ini.

REFERENCES

[1] X. Zhang, S. N. Mohanty, A. K. Parida, S. K. Pani, B. Dong, and X. Cheng, "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha," IEEE Access, vol. 8, pp. 30223-30233, February 2020.

[2] N. K. A. Appiah-Badu, Y. M. Missah, L. K. Amekudzi, N. Ussiph, T. Frimpong, and E. Ahene, "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana," IEEE Access, vol. 10, pp. 5069-5082, December 2021.

[3] M. T. Anwar, W. Hadikurniawati, E. Winarno, and W. Widiyatmoko, "Performance Comparison of Data Mining Techniques for Rain Prediction Models in Indonesia," 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 83-88, December 2020.

[4] N. S. Sani, A. H. A. Rahman, A. Adam, I. Shlash, and M. Aliff, "Ensemble Learning for Rainfall Prediction," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 11, November 2020.

[5] Y. Zhao, H. Shi Y. Ma, M. He, H. Deng, and Z. Tong, "Rain Prediction Based on Machine Learning," Proceedings of the 2022 8th International Conference on Humanities and Social Science Research (ICHSSR 2022), vol. 664, pp. 2957-2970, Juni 2022.

[6] S. Hudnurkar and N. Rayavarapu, "Binary classification of rainfall time-series using machine learning algorithms," International Journal of Electrical and Computer Engineering (IJECE), vol. 12, no. 2, pp. 1945-1954, April 2022.

[7] B. Purnama, Pengantar Machine Learning. Bandung, BDG: Penerbit Infomatika, 2019.

[8] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-80, November 1997.