



Penerapan *Machine Learning* untuk Prediksi Bencana Banjir

Elsa Pitaloka, Tegar Bagus Agung Hartanto, Sulis Sandiwarno*

Fakultas Ilmu Komputer, Sistem Informasi, Universitas Mercu Buana, Jakarta

Naskah Diterima: 27 Juli 2023; Diterima Publikasi: 5 Desember 2023
DOI: 10.21456/vol14iss1pp62-76

Abstract

Indonesia has a tropical climate because it is located at the equator, therefore Indonesia also only has two seasons, namely the dry and the rainy season. If the rainy season arrives and the rainfall is of high intensity, it has the potential to cause flooding. The losses caused by the flood disaster are quite large. To reduce the risk of disasters and losses due to floods, innovations in disaster mitigation are needed. Several previous studies have analyzed and predicted flood disasters using Machine Learning-based methods such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Naive Bayes. However, the method used in the study has problems such as a low level of accuracy and requires a long time to calculate the data. In this study we proposed a new model called algoritme Deep Neural Investigation Network (DNIN), which was combined from Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM). The process of proposed methods in the study consists of three parts, first is the Convolutional Neural Network (CNN) used to extract spatial features from flood data, and the Bidirectional Long Short-Term Memory (BiLSTM) is used to capture the temporal patterns of these features, and the final stage is to combine the results of the two methods to predict the level of flood hazard. As a result of the research conducted on rainfall data, information was obtained that the model we proposed was superior to the previous model in predicting and detecting flood disasters.

Keywords: Flood Prediction; Machine Learning; CNN; BiLSTM.

Abstrak

Indonesia beriklim tropis karena terletak pada garis katulistiwa, oleh karena itu Indonesia juga hanya memiliki dua musim, yaitu musim kemarau dan musim hujan. Apabila musim hujan tiba dan curah hujan intensitasnya tinggi, maka hal tersebut berpotensi menyebabkan bencana banjir. Kerugian yang ditimbulkan akibat bencana banjir cukup besar. Untuk mengurangi risiko bencana dan kerugian akibat banjir, diperlukan inovasi dalam mitigasi bencana. Beberapa penelitian sebelumnya telah melakukan analisis dan prediksi mengenai bencana banjir dengan menggunakan metode berbasis *Machine Learning* seperti *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Naive Bayes*. Akan tetapi metode yang digunakan dalam penelitian tersebut memiliki permasalahan seperti tingkat akurasi yang rendah dan membutuhkan waktu yang lama untuk melakukan perhitungan data. Dalam penelitian ini kami mengusulkan sebuah model baru yang dinamakan algoritma *Deep Neural Investigation Network* (DNIN), yang dikombinasikan dari *Convolutional Neural Network* (CNN) dan *Bidirectional Long Short Term Memory* (BiLSTM). Proses dari usulan metode dalam penelitian terdiri dari tiga bagian, yang pertama *Convolutional Neural Network* (CNN) digunakan untuk melakukan ekstraksi fitur spasial dari data banjir, selanjutnya *Bidirectional Long Short Term Memory* (BiLSTM) digunakan untuk menangkap pola temporal dari fitur-fitur tersebut. Kemudian tahap terakhir adalah menggabungkan hasil dari kedua metode tersebut. Hasil dari penelitian yang dilakukan terhadap data curah hujan, didapatkan informasi bahwa model yang kami usulkan lebih unggul dibandingkan dengan model sebelumnya dalam melakukan prediksi bencana banjir.

Kata kunci : Prediksi Banjir; *Machine Learning*; CNN; BiLSTM.

1. Pendahuluan

Air memiliki peran penting bagi kehidupan makhluk hidup. Banyak manfaat dari air untuk kehidupan makhluk hidup. Namun tidak hanya bermanfaat, air juga dapat menjadi salah satu faktor yang mempengaruhi terjadinya bencana alam, seperti bencana banjir maupun bencana kekeringan. Sungai menjadi salah satu tempat untuk menampung air dan sungai juga mengalirkan air dari hulu ke hilir. Namun, apabila air sungai datang secara berlebihan tidak ada daerah resapan dan penampungan air maka itu akan menyebabkan bencana banjir.

Banjir terjadi karena beberapa faktor yang dapat mempengaruhi, seperti kemampuan filtrasi tanah yang sangat lemah, sampah yang menumpuk pada daerah aliran sungai atau air yang mengakibatkan air tidak dapat mengalir, tingginya intensitas curah hujan, dan durasi hujan yang terlalu lama (Hasanah *et al.*, 2021).

Indonesia merupakan negara yang dilintasi garis katulistiwa hal tersebut menyebabkan Indonesia beriklim tropis, hanya memiliki dua musim, yaitu musim kemarau dan musim hujan, Indonesia juga memiliki intensitas curah hujan yang tinggi. Curah hujan dapat diprediksi dengan data historis masa lampau untuk memprediksi curah hujan yang akan datang selanjutnya. Untuk suatu daerah yang

*)Corresponding author: sulis.sandiwarno@mercubuana.ac.id

intensitas curah hujan tinggi dapat menyebabkan banjir, kemudian untuk daerah yang intensitasnya rendah dapat mengalami kekeringan.

Penelitian terdahulu sudah mengusulkan untuk menerapkan beberapa metode untuk melakukan prediksi bencana banjir dengan menggunakan metode KNN (Rangkuti *et al.*, 2021). Pada penelitian tersebut dilakukan penerapan Algoritma K-NN dalam memprediksi dan menghitung tingkat akurasi data cuaca di Indonesia. Data yang digunakan sebanyak 3623 data yang kemudian dibagi menjadi data pelatihan dan data uji dengan perbandingan 80:20. 80% untuk data pelatihan atau sebanyak 2898 dari 3623 data dan untuk data uji sebanyak 20% atau sebanyak 725 dari 3623 data. Menghasilkan suatu prediksi dengan tingkat akurasi data sebesar 0.8993 atau sekitar 89%. Hasil dari algoritma KNN merupakan algoritma yang paling bagus dalam prediksi banjir, nilai akurasi 88.94%, *error* 11.06% (Syamsurizal *et al.*, 2022).

Dwiasnati dan Devianto (2021) dalam penelitiannya, optimasi prediksi bencana banjir menggunakan algoritma SVM tingkat akurasi yang dihasilkan adalah 85.71%, kemudian penelitian Fitriyaningsih dan Basani (2019) prediksi menggunakan algoritma SVM menggunakan aplikasi *Ensemble Machine Learning* memiliki tingkat akurasi 94.44%. Fitriyah *et al.* (2022) melakukan studi komparasi algoritma klasifikasi C5.0, SVM, dan *Naïve Bayes* dalam prediksi banjir. Hasil penelitiannya menunjukkan bahwa nilai akurasi algoritma C5.0 dan SVM lebih akurat dan efisien namun waktu pemrosesannya tidak lebih cepat dari *Naïve Bayes*.

Penelitian sebelumnya sudah melakukan analisis untuk prediksi bencana banjir. Akan tetapi, metode-metode tersebut memiliki permasalahan seperti tingkat akurasi yang rendah dan membutuhkan waktu yang lama untuk melakukan perhitungan data. Untuk mengatasi masalah tersebut kami mengusulkan sebuah metode baru yang berbasis kepada *Deep Learning* untuk melakukan analisis terhadap prediksi banjir yang dinamakan *Deep Neural Investigation Network* (DNIN) *algorithm*, yang dikombinasikan dari *Convolutional Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (BiLSTM).

Adapun proses dari metode yang kami usulkan dalam penelitian adalah terdiri dari tiga bagian, yang pertama *Convolutional Neural Network* (CNN) digunakan untuk melakukan ekstraksi fitur spasial dari data banjir, kemudian yang ke dua yaitu *Bidirectional Long Short-Term Memory* (BiLSTM) digunakan untuk menangkap pola temporal dari fitur-fitur tersebut. Kemudian tahap terakhir adalah menggabungkan hasil dari kedua metode tersebut. Hasil dari penelitian yang dilakukan terhadap data curah hujan, didapatkan informasi bahwa model yang kami usulkan lebih unggul dibandingkan dengan

model sebelumnya dalam melakukan prediksi bencana banjir.

2. Kerangka Teori

Berikut beberapa teori yang digunakan terkait penelitian ini.

2.1. Pengantar Bencana Banjir

Bencana banjir berdampak terhadap segala macam aktifitas masyarakat seperti Kesehatan dan sosial-ekonomi terutama bagi masyarakat yang tinggal di dataran rendah, daerah bantaran sungai, dan daerah perkotaan (Azizulhaq *et al.*, 2021). Terjadinya banjir disebabkan oleh beberapa faktor seperti kondisi tanah, manusia, dan curah hujan. Intensitas curah hujan dan durasi hujan yang terlalu lama serta kemampuan filteris tanah yang sangat lemah, drainase saluran air yang tidak lancar sangat berpotensi menyebabkan bencana banjir (Hasanah *et al.*, 2021).

Penelitian sebelumnya sudah dilakukan analisis untuk prediksi bencana banjir. Akan tetapi, metode-metode tersebut memiliki permasalahan seperti tingkat akurasi yang rendah dan membutuhkan waktu yang lama untuk melakukan perhitungan data. Oleh karena itu, pada penelitian ini diusulkan sebuah metode baru yang berbasis kepada *deep learning* untuk melakukan analisis terhadap prediksi banjir yang dinamakan *Deep Neural Investigation Network* (DNIN) *algorithm*, yang dikombinasikan dari *Convolutional Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (BiLSTM).

2.2. Analisis dan Prediksi Banjir dengan Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan metode dalam *Supervised Learning Classification* dan *Regression* untuk mempelajari pola data dengan akurasi dan reproduktifitas yang optimal dan seimbang (Pisner and Schnyer, 2019). SVM bekerja dengan prinsip *Structural Risk Minimization* (SRM) untuk menemukan *hyperplane* terbaik yang memisahkan dua buah class pada *input space* (Nugroho *et al.*, 2003).

Beberapa penelitian terkait yang menggunakan metode *Support Vector Machine* yaitu Dwiasnati dan Devianto (2021), mengenai optimasi prediksi bencana banjir menggunakan Algoritma SVM untuk penentuan daerah rawan bencana banjir, menggunakan data intensitas curah hujan, debit air, luas wilayah, durasi hujan serta kepadatan penduduk. Diperoleh tingkat akurasi 85.71% dan AUC yang dihasilkan sebesar 0.84, serta diperoleh 97.62%. AUC yang dihasilkan sebesar 1.000 dari penggunaan metode SVM yang dioptimasi dengan PSO (*Particle Swarm Optimization*).

Riza *et al.* (2020) melakukan penelitian berupa pemanfaatan kecerdasan buatan untuk meningkatkan mitigasi bencana banjir menggunakan metode SVM

menyatakan bahwa kualitas prediksi banjir dapat ditingkatkan dengan beberapa cara yaitu melakukan hibridasi metode yang digunakan untuk melakukan prediksi, penggunaan dekomposisi data, penggunaan metode *ansambel* untuk mengurangi ketidakpastian prediksi, serta penggunaan algoritma pengoptimal untuk meningkatkan kualitas dan penyesuaian yang lebih baik.

Penelitian lain oleh Delimayanti *et al.* (2021) terkait Pemanfaatan Metode Multiclass-SVM pada Model Klasifikasi Pesan Bencana Banjir di Twitter, ditemukan bahwa metode SVM, dapat dimanfaatkan untuk klasifikasi data dalam 3 kelas, dengan nilai akurasi tertinggi sebesar 77.90% dan diperoleh 87.03% dengan pendekatan metode *One Versus All* (OVA).

Fitrihanah *et al.* (2022), melakukan Studi Komparasi Algoritma Klasifikasi C5.0, SVM dan *Naive Bayes* dengan Studi Kasus Prediksi Banjir, didapatkan bahwa algoritma SVM membutuhkan waktu lebih lama dalam pemrosesan data dibandingkan algoritma *Naive Bayes*, dengan nilai akurasi yang diperoleh adalah 93.75%.

Fitriyaningsih dan Basani (2019) telah melakukan penelitian menggunakan algoritma *Propagation Neural Network* (BP-NN) dan algoritma *Support Vector Machine* (SVM) dalam melakukan prediksi kejadian banjir. Dalam penelitian ini, digunakan dataset dari BMKG dan Badan Pengawasan Sungai II (BWS II). Data yang digunakan adalah data curah hujan dan debit air selama dua tahun (2016-2017) dari tiga stasiun di Kabupaten Deli Serdang Indonesia, yaitu Sampali, Kualanamu, dan Tuntungan. Prediksi banjir dalam periode 6 hari memperoleh nilai akurasi 100%, namun untuk prediksi banjir dalam jangka Panjang seperti periode 30 hari, nilai RSME (*Root Mean Square Error*) yang diperoleh lebih besar dari periode 6 hari.

2.3. Analisis dan Prediksi Banjir Dengan K-Nearest Neighbor (KNN)

K-Nearest Neighbor (K-NN) merupakan sebuah metode klasifikasi pembelajaran terpandu yang menentukan hasil analisis prediksi dari sampel uji berdasarkan mayoritas dari kategori K-tetangga terdekat atau berdasarkan jarak data pembelajaran terdekat dengan obyek tersebut. K-NN (*K-Nearest Neighbor*) bertujuan untuk pengklasikasian objek baru atribut dan sampel latih. Pengklasifikasian berdasarkan pada memori. Diberikan titik uji, sehingga akan ditemukan sejumlah K objek (titik pelatihan) yang paling dekat dengan titik uji (Asmara *et al.*, 2021).

Rangkuti *et al.*, (2021) melakukan penelitian menggunakan Algoritma K-NN dalam memprediksi dan menghitung tingkat akurasi data cuaca di Indonesia. Data yang digunakan sebanyak 3623 data yang kemudian dibagi menjadi data pelatihan dan data uji dengan perbandingan 80:20, 80% untuk data

pelatihan atau sebanyak 2898 dari 3623 data dan untuk data uji sebanyak 20% atau sebanyak 725 dari 3623 data, yang menghasilkan suatu prediksi dengan tingkat akurasi data sebesar 0.8993 atau sekitar 89%.

Syamsurizal *et al.* (2022) melakukan penelitian berupa perbandingan metode *Data Mining* untuk prediksi banjir dengan algoritma *Naive Bayes* dan KNN. Menggunakan data set banjir dari situs *Kaggle.com* selama satu tahun (2020), diperoleh hasil akhir yaitu algoritma KNN (*K-Nearest Neighbor*) lebih akurat dibandingkan algoritma *Naive Bayes* dalam melakukan prediksi banjir dengan nilai akurasi 88.94% dan *error* 11.06%.

Utami *et al.* (2021) melakukan penelitian terkait prediksi cuaca di kota Palembang berbasis *Supervised Learning* menggunakan algoritma *K-Nearest Neighbour* memperoleh nilai MSE (*Mean Square Error*) sebesar 0.031 yang dikombinasi dengan bantuan LSH (*Locality Sensitive Hashing*).

Asmara *et al.* (2021) melakukan analisis Prediksi Banjir menggunakan algoritma K-NN (*K-Nearest Neighbor*) mengenai prediksi banjir lahar dingin pada lereng merapi menggunakan data curah hujan dari satelit yang mengkombinasikan Metode *10-Fold Cross Validation* dan uji akurasi untuk mendapatkan nilai optimal. Dalam penelitian tersebut menggunakan parameter data curah hujan dan akumulasi hujan pada kawasan lereng merapi. Nilai k optimal yang diperoleh dari hasil penelitian adalah 3-NN dengan akurasi rata-rata 72.307% untuk lokasi Gendol, 81.429% untuk lokasi Putih 1, dan diperoleh 1-NN dengan akurasi rata-rata 86.955% untuk lokasi Putih 2.

2.4. Analisis dan Prediksi Banjir Dengan Naive Bayes

Algoritma *Naive Bayes* metode yang juga dikenal dengan *Naive Bayes Classifier*. Algoritma *Naive Bayes* merupakan metode untuk klasifikasi baik *biner* dan *multiclass*. Algoritma *Naive Bayes* menerapkan teknik *supervised* klasifikasi objek di masa depan dengan menetapkan label kelas ke *instance*/catatan menggunakan probabilitas bersyarat. *Supervised* merujuk kepada klasifikasi pelatihan data yang sudah diberi label dengan kelas. Probabilitas bersyarat sendiri ialah ukuran peluang suatu peristiwa yang terjadi berdasarkan peristiwa lain yang telah (dengan asumsi, praduga, pernyataan, atau terbukti) terjadi.

Pada penelitian yang dilakukan oleh A'ziziyah *et al.* (2022), algoritma *Naive Bayes* merupakan prosedur klasifikasi probabilitas sederhana. Hasil dari penelitian tersebut mendapatkan bahwa menggunakan metode K-NN lebih baik yang di mana metode K-NN memperoleh akurasi 95% dibandingkan dengan metode *Naive Bayes* yang memperoleh akurasi 90%.

Kemudian pada penelitian Syamsurizal (2022), klasifikasi Bayes merupakan pengelompokan statistik yang memperkirakan probabilitas keanggotaan kelas, seperti probabilitas bahwa sebuah *tuple* milik kelas

tertentu. Pada *Naive Bayes*, peran nilai atribut di kelas tertentu tidak tergantung pada nilai atribut lainnya. Penelitian ini pun melakukan perbandingan antara algoritma *Naive Bayes* dan K-NN dengan hasil akhir adalah algoritma K-NN merupakan algoritma yang paling bagus dalam prediksi banjir dengan nilai akurasi 88.94% dan *error* 11.06% dibandingkan dengan *Naive Bayes* dengan nilai akurasi 74.36% dan *error* 25.64%.

Fitriana *et al.* (2022) melakukan penelitian studi komparasi algoritma klasifikasi C5.0, SVM dan *Naive Bayes*. Pada penelitian tersebut dapat digunakan untuk melakukan prediksi banjir untuk membantu pencegahan bencana agar mencegah terjadinya korban yang lebih banyak. Penelitian tersebut membandingkan algoritma SVM dan C5.0 dengan algoritma *Naive Bayes*. Hasil perbandingan algoritma tersebut mendapati algoritma C5.0 dan SVM memiliki nilai akurasi yang sama yaitu sebesar 93.75% sedangkan algoritma *Naive Bayes* memiliki nilai akurasi sebesar 81.25%. Sehingga kemudian disimpulkan bahwa algoritma C5.0 dan SVM lebih akurat digunakan untuk melakukan prediksi. Sedangkan algoritma *Naive Bayes* memiliki waktu pemrosesannya lebih cepat apabila dibandingkan algoritma C5.0 dan algoritma SVM.

Penelitian lainnya oleh Yoga *et al.* (2022), algoritma *Naive Bayes* merupakan salah satu metode klasifikasi yang populer dalam bidang *Machine Learning*. *Naive Bayes* tidak dapat melakukan inferensialitas dari banyak aturan yang terkait erat, namun dapat menghasilkan validitas teori probabilitas. Pada penelitian ini peneliti mengkombinasikan algoritma *Naive Bayes* untuk klasifikasi dan simple kringing untuk interpolasi peta curah hujan. Model harian memiliki akurasi sebesar 67% dan model bulanan memiliki akurasi sebesar 88%. Hal tersebut bisa menyimpulkan metode *Naive Bayes* kurang cocok untuk model harian namun cukup bagus untuk model bulanan, hal tersebut dapat terjadi karena model harian memiliki banyak data, sedangkan model *Naive Bayes* untuk data yang kecil.

Kemudian Triyanto *et al.* (2021) dalam penelitiannya melakukan analisis klasifikasi bencana banjir berdasarkan curah hujan. Pada penelitian tersebut peneliti menggunakan algoritma *Naive Bayes* untuk menganalisis klasifikasi bencana banjir berdasarkan curah hujan. Pengujian pada saat menggunakan *library* dari *Scikit-Learn* algoritma *Naive Bayes* mampu memberikan akurasi sebesar 79.16%, dibandingkan dengan penelitian sebelumnya yaitu "*Flood Prediction using Logistic Regression for Kerala State*" memiliki akurasi 75%, maka prediksi banjir belum signifikan. Sedangkan pengujian *Naive Bayes* diterapkan di *RapidMiner* sebagai pembanding, *output* dari *RapidMiner* cukup berbeda karena akurasi mencapai 98.31%.

2.5. Pengantar Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan bagian dari *Deep Neural Network*, dengan jenis jaringan saraf tiruan yang digunakan dalam pengenalan dan pemrosesan gambar. Metode CNN dapat mengenali objek atau gambar baru berdasarkan fitur-fitur yang dideteksi. *Convolutional Neural Network* (CNN) dirancang khusus untuk memproses data piksel dan citra *visual*. CNN memiliki *neuron* yang diatur dalam 3 dimensi: *width*, *height*, dan *depth*. CNN memiliki *neuron* yang bekerja seperti *lobus frontal*, pada area *visual cortex* pada otak manusia dan hewan. *Visual cortex* merupakan area yang memproses informasi dalam bentuk rangsangan visual. Teknik utama pada metode CNN adalah *convolution*, di mana filter meluncur di atas *input* dan menggabungkan nilai *input* dan nilai *filter* pada peta fitur. CNN yang merupakan model *Deep Learning* yang telah digunakan untuk melakukan beberapa penelitian prediksi data deret waktu (Intan *et al.*, 2021). Salah satu keunggulan utama CNN adalah fitur *local perception* dan *weight sharing* yang dapat sangat mengurangi jumlah parameter, sehingga meningkatkan efisiensi pada proses pembelajaran. Dari sisi struktur, CNN utamanya terdiri dari dua bagian, yaitu *layer* konvolusi (*convolutional*) dan *layer* penyatuan (*pooling*) (Widiputra *et al.*, 2021).

Penelitian terdahulu yang menggunakan algoritma CNN (*Convolutional Neural Network*) ialah mendeteksi *Covid-19* dari data *image chest X-ray* yang dilakukan oleh Karno *et al.* (2021). Dengan menggunakan 4000 data *image chest x-ray*, 33% data sebagai data uji dan 67% sebagai data pelatihan. Hasil penelitian dapat memprediksi klasifikasi penderita COVID, *Lung Opacity*, Normal, *Viral Pneumonia* dengan presisi masing-masing 84%, 80%, 88%, dan 91%. Namun masih memiliki tingkat akurasi yang belum optimal yaitu sebesar 85,5%.

Penelitian lainnya dilakukan oleh Hermanto *et al.* (2021) terkait Sentimen Klasifikasi dengan *Word2vec* pada media *online* menggunakan Algoritma LSTM (*Long Short-Term Memory*)-CNN (*Convolutional Neural Network*). Menggunakan dataset judul artikel berbahasa Indonesia dari situs Detik *Finance*. Penelitian ini bertujuan pengklasifikasian judul berita berbahasa Indonesia berdasarkan sentimen positif negatif. Penelitian ini diperoleh kesimpulan bahwa kombinasi metode CNN-LSTM lebih baik dibandingkan metode LSTM dan kombinasi metode LSTM-CN. Metode CNN-LSTM memiliki nilai akurasi metode sebesar 62%, nilai *Recall* sebesar 76% dan nilai presisi 66%.

Penelitian lainnya ialah Pemodelan CNN untuk deteksi emosi berbasis *speech* Bahasa Indonesia oleh Aini *et al.* (2021) dengan dataset dari TV series berbahasa Indonesia. Penelitian ini bertujuan melakukan proses klasifikasi emosi, yaitu empat kelas label emosi marah, senang, netral, dan sedih.

Penelitian ini terdiri dari tujuh layer *convolutional 1D*, dua *max pooling*, dan empat *class output*. Menggunakan *epoch* sebanyak 80 kali dari 507 data, sebanyak 70% data latih dan 30% data uji. Hasil penelitian diperoleh nilai akurasi rata-rata sebesar 85% dengan kombainis fitur terbaik yaitu MFCC (*Mel-Frequency Cepstrum Coefficient*) dan *pitch*.

Penelitian yang dilakukan Widiputra *et al.* (2021) terkait Prediksi Indeks BEI dengan model *Ensemble Convolutional Neural Network* dan *Long Short-Term Memory*. Penelitian memiliki tujuan melakukan prediksi data deret waktu keuangan, memprediksi nilai indeks pasar saham untuk periode satu hari kedepan. Metode CNN digunakan untuk ekstraksi fitur penting dari data deret waktu dan menggunakan metode LSTM untuk memproses data deret waktu. Hasil penelitian diperoleh bahwa akurasi terbaik diberikan oleh model CNN-LSTM dengan peningkatan akurasi pada model CNN-LSTM sebesar 7.6% (CNN) dan 8.7% (LSTM) pada tahap pelatihan serta 2.7% (CNN) dan 1.6% (LSTM) pada tahap uji.

Penelitian lainnya dilakukan oleh Septiandi *et al.* (2021) terkait deteksi kedipan dengan metode CNN dan *Percentage of Eyelid Closure* (PERCLOS). Penelitian ini menghasilkan sistem dapat mengenali kondisi ketika mata tertutup atau mata terbuka dan menentukan lebar bukaan mata, dapat mengenali wajah dari objek bukan wajah dengan jarak pendeteksian optimal antara 40-70 cm. Model hasil pelatihan dapat mengklasifikasikan kondisi mata terbuka dan mata tertutup dengan menggunakan *Convolutional Neural Network* dengan arsitektur yang memiliki 3 *layer* mendapatkan hasil akurasi 98% dan *loss* 2.05%.

2.6. Pengantar *Bidirectional Long Short-Term Memory* (BiLSTM)

Bidirectional Long Short-Term Memory atau BiLSTM merupakan lanjutan atau dikembangkan dari *Long Short-Term Memory* atau LSTM yang dapat mengatasi kekurangan dari metode LSTM. *Long Short-Term Memory* atau LSTM sendiri merupakan algoritma *Deep Learning* yang dapat digunakan untuk memprediksi dan klasifikasi yang berhubungan dengan waktu. *Bidirectional Long Short-Term Memory* atau BiLSTM adalah salah satu varian LSTM yang umum digunakan. Input yang dimasukkan pada BiLSTM terdapat 2 jenis yaitu *input forward* dan *backward* (Sari *et al.*, 2021).

Penelitian sebelumnya OLEH Sari *et al.* (2021), BiSLTM merupakan salah satu jenis LSTM lanjutan dengan mempertimbangkan data *input forward* dan *backward* untuk meningkatkan akurasi pada hasil prediksi. LSTM sendiri adalah salah satu macam dari NN dan perkembangan system *Recurrent Neural Network* (RNN). LSTM digunakan untuk mengatasi kekurangan dari RNN terkait masalah *gadien vanishing*. Pada penelitian tersebut peneliti mengusulkan sistem prediksi arah dan kecepatan

angin menggunakan BiLSTM. Hal tersebut karena hasil prediksi menggunakan BiLSTM lebih baik dari beberapa sistem lain seperti FCNN dan SLSTM.

Kemudian pada penelitian Masri *et al.* (2020), peneliti menggunakan beberapa metode *Deep Learning* yaitu *Recurrent Neural Network* (RNN), *Long Short-Term Memory* (LSTM), dan *Bidirectional Long Short-Term Memory* (BiLSTM) untuk dibandingkan dan mendapatkan performa yang terbaik. Pada penelitian tersebut, peneliti membandingkan performa dari tiga metode *Deep Learning* tersebut untuk memprediksi *level* permukaan air laut selama 48 jam, 72 jam, dan 168 jam. Hasil percobaan dari penelitian menunjukkan bahwa BiLSTM memberikan hasil yang lebih baik dibandingkan dengan LSTM dan RNN, dikarenakan kinerja BiLSTM memungkinkan pelatihan tambahan dengan melintasi data input dua kali yaitu, dari kiri ke kanan dan dari kanan ke kiri.

Pada penelitian lain sebelumnya oleh Rahmadani *et al.* (2022) untuk mendeteksi emosi berdasarkan wicara menggunakan *Deep Learning*. Peneliti menggunakan beberapa model *Deep Learning* untuk dilakukan perbandingan, yaitu CNN, BiLSTM, BiGRU, CNN-BiLSTM, dan CNN-BiGRU untuk pengolahan data suara atau audio. Hasil dari penelitian tersebut didapati bahwa perbandingan hasil akurasi model yang muncul, model CNN-BiLSTM adalah model yang akurasi performanya paling baik yaitu 91.29% dalam mendeteksi emosi berdasarkan ucapan.

Kemudian pada penelitian yang dilakukan oleh Karyadi dan Santoso (2022) terkait prediksi kualitas udara dengan menggunakan beberapa metoda *Deep Learning* yang merupakan bagian *Recurrent Neural Network* (RNN), antara lain yaitu *Long Short-Term Memory* (LSTM), *Bidirectional Long Short-Term Memory* (BiLSTM) dan *Gated Recurrent Unit* (GRU). Pada penelitian tersebut unit yang diterapkan pada permasalahan memprediksi data *time series* kualitas udara dengan parameter suhu, kelembaban, Indeks Standar Pencemar Udara (ISPU), *particular matter* PM10, dan suhu. Hasil pengujian 3 model prediksi terhadap 4 variabel tersebut, berdasarkan kreteria penilain menggunakan RMSE dari data uji dan dibandingkan dengan standar deviasi, model LSTM dan BiLSTM, menunjukan hasil yang lebih bagus untuk permasalahan data yang bersifat *time series* kualitas udara dibanding model *Gated Recurrent Unit* (GRU) yang hasil yang kurang bagus.

Penelitian oleh Nugroho *et al.* (2021) sebelumnya telah melakukan penelitian untuk deteksi depresi dan kecemasan pengguna *Twitter* menggunakan *Bidirectional Long Short-Term Memory* (BiLSTM). Hal tersebut dapat dilakukan dengan teknik klasifikasi teks. Pendekatan yang dapat digunakan salah satunya ialah *Long Short-Term Memory* (LSTM) sebagai pengembangan arsitektur *Recurrent Neural Network* (RNN) dalam menangani masalah *vanishing gradient*. LSTM standar belum cukup untuk menerima informasi karena hanya mampu membaca kalimat dari

satu arah. *Bidirectional Long Short-Term Memory* (BiLSTM) merupakan LSTM lanjutan dua arah yang mampu menangkap informasi tanpa mengabaikan konteks dan arti dari suatu kalimat. BiLSTM yang diusulkan pada penelitian tersebut menunjukkan performa yang lebih tinggi dari semua model *Machine Learning* tradisional dan LSTM standar. Berdasarkan hasil pengujian pada penelitian tersebut, akurasi yang paling tinggi adalah akurasi yang diperoleh BiLSTM, yaitu mencapai 94.12%.

3. Metode

3.1. Deskripsi Sumber Data

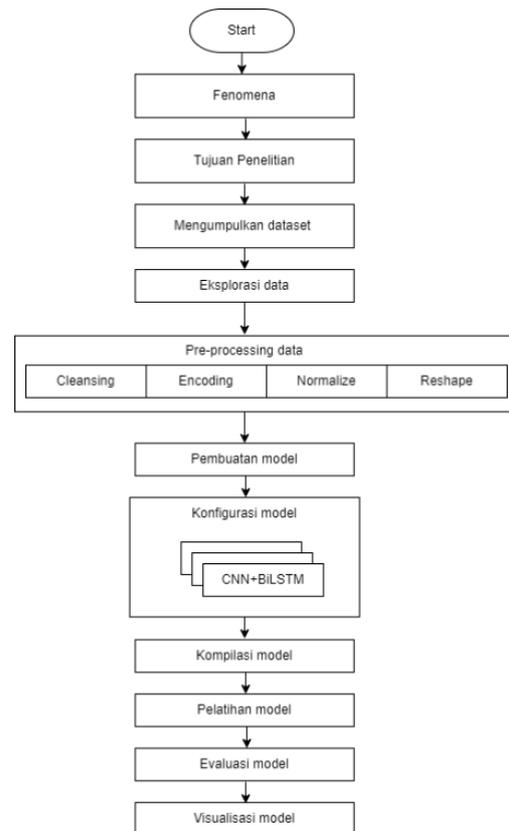
Pada penelitian ini, pengambilan *sample* dataset yaitu menggunakan publik dataset pada *website Kaggle*. Dataset yang digunakan dalam penelitian ini merupakan data *time series* di Kerala yang merupakan salah satu kota di India, dari bulan Januari tahun 1901 sampai bulan Desember tahun 2018. Data *time series* tersebut memiliki dua belas *feature* yang berisi data curah hujan.

3.2. Teknik Pengumpulan Data

Penelitian ini termasuk penelitian kuantitatif. Penelitian kuantitatif merupakan penelitian berdasarkan data numerik dengan menggunakan analisis secara matematis atau teknik statistik. Permasalahan penelitian kuantitatif memiliki cakupan wilayah yang luas dan tingkat variasi yang kompleks. Pendekatan kuantitatif menimbulkan kesulitan dalam mengontrol variabel lain yang dapat mempengaruhi proses penelitian baik secara langsung maupun tidak langsung. Untuk mencapai validitas yang tinggi, kehati-hatian juga diperlukan dalam definisi sampel, pengumpulan data, dan definisi alat analisis (Rangkuti *et al.*, 2021).

3.3. Diagram Alir Penelitian

Dalam metode penelitian ini terdiri dari beberapa tahapan, adapun proses dari metode yang diusulkan dalam penelitian adalah tahap pertama melakukan *research* data, kemudian data tersebut dieksplorasi selanjutnya dilakukan *preprocessing* seperti *cleansing*, *encoding*, normalisasi, *reshape*, dan pembagian dataset. Tahap kedua, melakukan pelatihan data dengan menggunakan klasifikasi CNN (*Convolutional Neural Network*) dan BiLSTM (*Bidirectional Long Short-Term Memory*). Tahap ketiga, melakukan evaluasi terhadap model dan prediksi menggunakan *k-fold cross validation* disertai visualisasi grafik akurasi, grafik *loss*, kurva ROC, AUC dan *confusion matrix*. Gambar 1. merupakan bagan dari alir penelitian “Penerapan *Machine Learning* untuk Prediksi Bencana Banjir”.



Gambar 1. Diagram Alir Penelitian

Berikut penjelasan metodologi penelitian “Penerapan *Machine Learning* untuk Prediksi Bencana Banjir”:

1. Tahap pertama merupakan aktivitas mengidentifikasi, memahami, dan mendefinisikan fenomena yang sedang terjadi dengan dilakukan analisis dan pengamatan lebih lanjut.
2. Tahap selanjutnya ialah mendefinisikan tujuan dari penelitian ini.
3. Tahap ketiga, melakukan pengumpulan data yang berhubungan atau relevan dengan tujuan penelitian.
4. Tahap selanjutnya melakukan analisis eksplorasi data dengan memahami karakteristik dataset meliputi.
5. Pada tahap *preprocessing* data dilakukan aktivitas *cleansing*, *encoding*, normalisasi, *reshape*, dan pembagian dataset.
6. Tahap pembuatan model meliputi penentuan dan perumusan arsitektur model yang tepat untuk nantinya model tersebut akan dilakukan prediksi.
7. Tahap konfigurasi model meliputi membuat model *Sequential*, menambahkan *hidden layer* seperti *convolution layer*, *pooling layer*, *Bidirectional LSTM layer*, dan *fully connected layer*.
8. Tahap kompilasi model meliputi menentukan fungsi *loss*, *optimizer*, dan metrik yang digunakan dalam pelatihan model.

9. Tahap evaluasi model adalah aktivitas untuk menguji model yang telah dibuat, menghitung matrik evaluasi dengan *k-fold cross validation*.
10. Tahap visualisasi meliputi menampilkkan grafik akurasi, grafik *loss*, kurva ROC dan *confusion matrix* hasil dari pelatihan, serta evaluasi model.

3.4. Preprocessing Data

Pre-processing data dilakukan pada setiap penggunaan algoritma *Convolutional Neural Network* (CNN), *Bidirectional Long Short-Term Memory* (BiLSTM), dan *Deep Neural Investigation Network* (DNIN). *Preprocessing* data merupakan proses untuk mempersiapkan data mentah menjadi data yang dapat digunakan dalam membangun model pembelajaran komputer (Prabuddhi and Seneviratne, 2020). Komputer hanya dapat melakukan klasifikasi dengan data yang bersifat numerik. Sehingga untuk memproses data berupa karakter huruf atau teks, data tersebut perlu diubah ke dalam bentuk numerik (Sulis Sandiwarno *et al.*, 2023). Pada penelitian ini *pre-processing* dilakukan dengan proses *cleansing*, *encoding*, *normalizes*, *reshape* dan *split*. Proses *cleansing*, *encoding*, *normalizes*, *reshape*, dan pembagian data yang digunakan seperti pada Tabel 1. berikut:

Tabel 1 Tabel Preprocessing

Proses	Implementasi
Cleansing	<code>dataset.isnull().sum()</code>
Encoding	<code>dataset['FLOODS'] = dataset['FLOODS'].replace(['YES', 'NO'], [1, 0])</code>
Normalizes	<code>scaler = MinMaxScaler()</code> <code>x = scaler.fit_transform(x)</code>
Reshape	<code>x=np.array(x).reshape((x.shape[0], x.shape[1], 1))</code>
Pembagian data latih dan uji	<i>Split</i> atau pembagian data latih (latih set) dan data uji (uji data) dengan porsi 80:20, 80% data latih dan 20% data uji (Widiputra <i>et al.</i> , 2021).

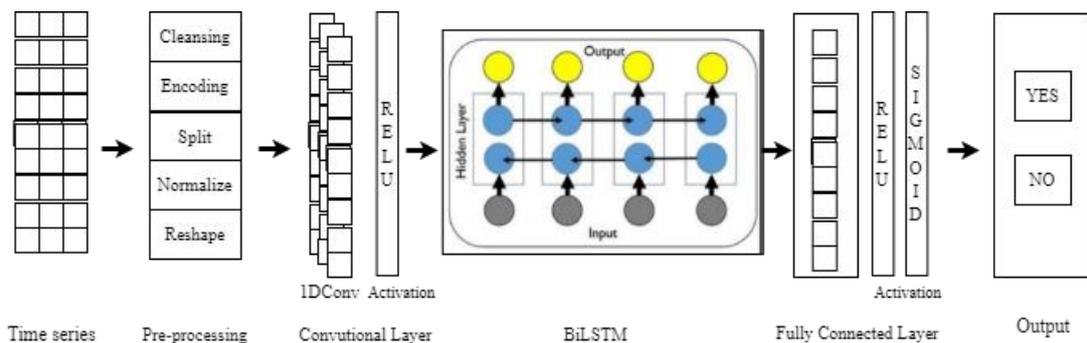
Dari proses di atas pada *cleansing* digunakan untuk memeriksa kolom yang tidak memiliki nilai atau kosong (*missing values*) dengan fungsi *isnull()* kemudian dihitung jumlah nilai yang kosong tersebut dengan fungsi *sum()* (Wulandari, A., 2022). Pada *encoding* digunakan untuk menggantikan nilai 'YES' dan 'NO' dalam kolom 'FLOODS' menjadi biner

sebagai *input* model, angka 1 mewakili 'YES' dan 0 mewakili 'NO'. Kemudian pada normalisasi fitur input menggunakan objek *MinMaxScaler* dari *library scikit-learn*. Objek *MinMaxScaler* digunakan untuk melakukan normalisasi fitur-fitur input yang disimpan dalam variabel *x*. Metode *fit_transform()* dari *MinMaxScaler* digunakan untuk menghitung statistik normalisasi (contoh, nilai minimum dan maksimum) dari data input *x* dan secara langsung mengubah *x* menjadi nilai-nilai yang telah dinormalisasi. Mengubah bentuk dari data dua dimensi menjadi data tiga dimensi (Wulandari, A., 2022).

3.5. Arsitektur Model

Arsitektur model dalam *Machine Learning* mengacu pada *structure* atau *configuration of the artificial neural network*, *neural network* tersebut digunakan untuk memproses data dan melakukan tugas seperti klasifikasi. Arsitektur model terdiri dari kumpulan lapisan yang saling berhubungan, dimana setiap lapisan memiliki peran dan fungsi tertentu dalam aliran informasi. Setiap lapisan arsitektur model memiliki satu set *neuron* atau unit yang menerima masukan dari lapisan sebelumnya, melakukan operasi matematika pada masukan tersebut, dan mengirimkan keluaran ke lapisan berikutnya. Selain kumpulan lapisan, arsitektur model juga menyertakan parameter penyetelan seperti fungsi aktivasi, metode pelatihan, dan metode pengoptimalan.

Dalam penelitian ini, kami mengusulkan sebuah model *Deep Neural Network* untuk melakukan prediksi bencana banjir. Arsitektur model ini terdiri dari kombinasi dua jaringan saraf dalam yang berbeda, yaitu algoritma CNN (*Convolutional Neural Network*) dan BiLSTM (*Bidirectional Long Short-Term Memory*) untuk memproses data *time series* dalam bentuk 1D (*Conv1D*) kemudian diteruskan ke *layer* BiLSTM (*Bidirectional Long Short-Term Memory*) untuk memodelkan hubungan temporal atau data *sequence*. Beberapa *layer Conv1D* digunakan untuk melakukan ekstraksi fitur dari data *time series*, dan *dropout layers* digunakan untuk mengurangi *overfitting*, serta penggunaan algoritma optimasi *Adam* untuk mengoptimalkan model seperti model pada gambar 2.



Gambar 2. Jaringan Arsitektur *Deep Neural Investigation Network* (DNIN)

Dari arsitektur model tersebut diusulkan sebuah *hybrid* model yang menggabungkan *Convolutional Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (BiLSTM) dalam satu arsitektur untuk memproses data *sequence* dengan harapan meningkatkan performa dalam tugas klasifikasi *biner* yaitu *Deep Neural Investigation Network* (DNIN).

Klasifikasi CNN pada penelitian ini terdiri dari *convolution layer* dan *fully connected layer*. Representasi dari pengklasifikasi CNN mengambil data sebagai *input* dan menunjuk hubungan antara lapisan model CNN untuk menentukan kelas data. Pada setiap lapisan *convolution* terdapat fungsi aktivasi relu (*Rectified Linear Unit*) yang bertugas mengubah nilai negatif menjadi nol dan mempertahankan nilai positif. Operasi konvolusi dapat dituliskan pada persamaan (1) (Widiasari *et al.*, 2018)

$$s(t) = (x * t)(t) = \sum_a^{\infty} -\infty x(a) * w(t - a) \quad (1)$$

Keterangan:

$S(t)$: Fungsi hasil operasi konvolusi

x : Input

w : Bobot (kernel)

Proses *Fully Connected Layer* adalah lapisan dimana semua *neuron* yang diaktifkan di lapisan atas terhubung ke *neuron* di lapisan berikutnya seperti *neural network*, setiap *neuron* menerima *input* dari semua *neuron* dalam lapisan sebelumnya dan menghasilkan *output* yang akan digunakan sebagai *input* untuk lapisan berikutnya. Lapisan ini terutama digunakan untuk MLP (*Multilayer Perceptron*), yang tujuannya adalah untuk mengubah ukuran data sehingga data dapat diklasifikasikan secara *linier*.

Perbedaan antara *fully connected layer* dan *convolutional layer* biasa adalah bahwa *neuron* pada *convolutional layer* hanya terhubung ke area *input* tertentu, sedangkan pada *fully connected layer* memiliki *neuron* yang terhubung penuh. (Adrian Rosebrock, 2009).

Fully connected layers berada pada bagian akhir model setelah lapisan-lapisan *convolutional* dan sebelum lapisan *output*. Setelah beberapa lapisan *Convolutional ID* dan *Dropout* yang digunakan untuk mengolah fitur-fitur dari data *input*, penambahan beberapa lapisan *fully connected (dense)* bertujuan untuk menggabungkan fitur-fitur yang telah diproses oleh lapisan-lapisan sebelumnya dan melakukan ekstraksi pola yang lebih kompleks sebelum akhirnya menghasilkan *output* (Sandiwarno, *et al.*, 2023). Pada lapisan *Fully connected layer* ditambahkan fungsi aktivasi *Relu* dan *sigmoid*. Fungsi aktivasi *sigmoid* mengambil *input* apa pun dan menghasilkan nilai antara 0 dan 1. Dalam konteks ini, fungsi aktivasi ini digunakan untuk menghasilkan probabilitas, sehingga nilai *output* dari lapisan ini akan berada dalam kisaran 0 hingga 1.

Klasifikasi Bidirectional LSTM (BiLSTM) adalah salah satu jenis arsitektur *Recurrent Neural Network* (RNN) yang digunakan dalam berbagai tugas pemodelan *data sequence*, dan *time series*. BiLSTM merupakan pengembangan dari LSTM (*Long Short-Term Memory*) tradisional. BiLSTM lebih baik daripada LSTM dalam menyelesaikan tugas *sequence modeling* (Liu *et al.*, 2017). BiLSTM membantu mempelajari dan menangkap konteks dengan lebih baik dengan cara meningkatkan kinerja algoritma dengan melatih dua proses algoritma LSTM, yaitu *input forward* dan *backward* secara paralel *output* (Sandiwarno, *et al.*, 2023).

Pengklasifikasi terdiri dari tiga gerbang sebagai *input gate*, *output gate* dan *forget gate*. *Input gate* mengatur aliran informasi baru yang dimasukkan ke dalam sel LSTM, *forget gate* menghapus informasi yang tidak relevan, dan *output gate* mengontrol informasi yang digunakan untuk menghasilkan *output*. Dengan memanfaatkan kedua arah pemrosesan data, BiLSTM mampu menangkap konteks dari kedua arah (masa lalu dan masa depan) dalam data *sequence*, yang meningkatkan kemampuan model untuk memahami dan menggeneralisasi pola yang kompleks dalam data *sequence*, operasi LSTM dua arah (BiLSTM) dapat dijabarkan pada persamaan (2). Dengan demikian, BiLSTM menjadi pilihan yang kuat dalam berbagai aplikasi yang memerlukan pemahaman konteks dan hubungan temporal dalam data *sequence* (Rizky, M.G, 2021)

$$y_t = W_{hy}^{\rightarrow} \vec{h}_t + W_{hy}^{\leftarrow} \vec{h}_t \quad (2)$$

Keterangan:

y_t : *output gate* LSTM dua arah

W_{hy}^{\rightarrow} : nilai boot untuk *output gate* LSTM maju

\vec{h}_t : nilai keluaran LSTM maju

W_{hy}^{\leftarrow} : nilai boot untuk *output gate* LSTM mundur

\vec{h}_t : nilai keluaran LSTM mundur

4. Hasil dan Pembahasan

Dalam bagian ini, akan menjelaskan mengenai beberapa informasi seperti: *experimental setup*, dataset, *process learning*, dan evaluasi.

4.1. Experimental Setup

Penelitian ini menggunakan perangkat keras yang dilengkapi *Processor* AMD Ryzen 7 dengan *Radeon Graphics* dan RAM 16 GB, menggunakan sistem operasi windows 11 dengan bahasa pemrograman *Python* 3.10.6. *Library* yang digunakan *Scikit-learn* dan *TensorFlow* untuk implementasi *classifiers Machine Learning*. Berikut pada Tabel 2. merupakan daftar *library python* yang digunakan.

Tabel 2 Daftar Library Python Digunakan

Library	Fungsi
<i>Pandas</i>	Library untuk membuat <i>table</i> , mengecek data, dan mengubah dimensi data.
<i>Numpy</i>	Library yang membantu operasi perhitungan numerik yang disimpan dalam bentuk array.
<i>Matplotlib</i>	Library untuk visualisasi data.
<i>TensorFlow</i>	Library yang menyediakan kumpulan alur kerja untuk mengembangkan dan melatih model.
<i>Scikit Learn</i>	Library yang dibangun berdasarkan <i>library Numpy, Scipy, Matplotlib, Pandas, IPython</i> dan <i>Sympy</i> .

4.2. Dataset

Dalam penelitian ini menggunakan publik dataset, pengambilan *sample* dataset pada *website Kaggle* berisi data curah hujan dari tahun 1901 sampai tahun 2018. Dataset yang diperoleh terdapat 16 atribut yaitu: *subdivision, year, jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec, annual rainfall, floods*. Dalam mengaplikasikan dataset menggunakan *python*, dataset yang akan digunakan dilakukan proses *import* terlebih dahulu. Gambar 3. merupakan contoh dataset yang digunakan.

Gambar 3 Contoh Dataset

4.3. Langkah Evaluasi Untuk Machine Learning Classifiers

Proses evaluasi dilakukan setelah *process learning* data pada setiap algoritma yang digunakan yaitu *Deep Neural Investigation Network (DNIN)*. Setelah model dilatih menggunakan data pelatihan, evaluasi dilakukan memahami dapat melakukan prediksi yang akurat dan mengukur kinerjanya dalam menggeneralisasi pada data baru. Proses evaluasi membantu dalam mengevaluasi kinerja model, mengidentifikasi kelemahan dan memperbaikinya, serta pemahaman terkait model dapat digeneralisasikan pada data baru yang tidak digunakan dalam pelatihan. Pada penelitian ini, dilakukan evaluasi *performance* pada setiap proses klasifikasi dengan menggunakan *K-Fold Cross Validation* dengan $K=5$ dan mengukur kinerja model dalam *average precision (Pre)*, *Recall (Rec)*, dan *F1-score (F1)*, operasi dapat dituliskan pada persamaan (3), (4), dan (5) (Batbaatar *et al.*, 2019) berikut:

$$R = \frac{TP}{TP+FN} \quad (3)$$

$$P = \frac{TP}{TP+FP} \quad (4)$$

$$F1 = \frac{2 \times P \times R}{P+R} \quad (5)$$

Menurut Rohani *et al.* (2018), *K-Fold Cross Validation* berfungsi untuk menilai kinerja proses suatu model dengan membagi sampel data secara acak dan mengelompokkan data tersebut sebanyak nilai K . *K-fold* dalam penelitian $K = 5$. Kemudian salah satu kelompok *K-Fold* tersebut akan dijadikan sebagai data uji sedangkan sisa kelompok yang lain akan dijadikan sebagai data latih. Pada penelitian ini deep learning parameters setting ditampilkan di Tabel 3.

Tabel 3 Parameters setting deep learning classifiers.

Parameter Setting	DNIN
Learning rate	0.0001
Batch size	32
Dropout	0.5
Epochs	200

4.4. Implementasi Model

Pada penelitian ini, konfigurasi model *Deep Neural Investigation Network (DNIN)* terdiri dari *layer convolutional* 1 dimensi, *layer Bidirectional Long Short-Term Memory (BiLSTM)*, dan *layer Fully connected* (Widiputra *et al.*, 2021). Konfigurasi parameter model ditunjukkan pada tabel 4.

Tabel 4 Konfigurasi Parameter Model

Parameter	Nilai
Convolutional layer 1	
Filter	64
Kernel_size	3
Aktivasi	Relu
Input_shape	$x_{latih}.shape[1], 1$
Kernel_reg	$tf.keras.regularizers.l2(0.01)$
Convolutional layer 2	
Filter	128
Kernel_size	3
Aktivasi	Relu
Convolutional layer 3	
Filter	128
Kernel_size	3
Aktivasi	Relu
Dropout	0.5
Bidirectional LSTM layer 1	
Neuron	256
Return_seq	True
Dropout	0.5
Bidirectional LSTM layer 2	
Neuron	128
Dropout	0.5
Fully Connected layer 1	
Neuron	128
Aktivasi	Relu
Dropout	0.5
Fully Connected layer 2	
Neuron	128
Aktivasi	Relu
Dropout	0.5
Fully Connected layer 3	
Neuron	64

Aktivasi	Neuron
Dropout	0.5
Fully Connected layer 4	
Neuron	1
Aktivasi	sigmoid
Optimizer	adam
Learning rate	0.0001
Loss function	binary_crossentropy
Epochs	200
Batch size	32

4.5. Hasil Eksperimen

Setelah implementasi model, pada bagian ini dilakukan proses pelatihan pada model *Deep Neural Investigation Network* (DNIN), skenario yang dilakukan yaitu menggunakan parameter dasar dan parameter tuning dengan mengubah beberapa parameter seperti *epochs*, *batch size*, *learning rate*. *Epoch* adalah satu siklus pengulangan dari proses learning terhadap model pada seluruh data latih. Satu *epoch* dapat diartikan bahwa *Deep Neural Investigation Network* (DNIN) telah melakukan proses lerning dari data latih secara keseluruhan dan berulang untuk mencapai nilai error dan akurasi yang sebaik mungkin. Sedangkan Batch Size merupakan jumlah sampel data yang disebar dalam proses pelatihan (Mustakim and Hayati, 2021).

Hasil dari eksperimen dengan pengaturan *epochs* dan *batch size* ditunjukkan pada Tabel 5. Dari seluruh eksperimen pada pengaturan *epochs* dan *batch size* didapatkan hasil terbaik yaitu pada *epoch 150 batch 64* dan *epoch 200 batch 32*. Pada *epoch 150 batch 64*, diperoleh nilai akurasi sebesar 0.9496, *precision* 0.9482, *Recall* 0.95, *F1-score* 0.9485, AUC 0.9776 dan ROC 1.00. Sedangkan hasil yang diperoleh pada *epoch 200 batch 32* yaitu akurasi 0.9496, *precision* 0.9346, *Recall* 0.9833, *F1-score* 0.9547, AUC 0.9804 dan ROC 0.99.

Tabel 5 Eksperimen *Epoch and Batch Size* DNIN

Batch size	Precision	Recall	F1-score	AUC	ROC
Epoch 100					
32	0.8779	0.9166	0.8845	0.9858	1.00
64	0.8539	0.9166	0.8729	0.9546	1.00
128	0.5850	1.0	0.7338	0.9723	0.97
Epoch 150					
32	0.9120	1.0	0.9532	0.9943	1.00
64	0.9482	0.95	0.9485	0.9776	1.00
128	0.8739	0.9833	0.9233	0.9885	1.00
Epoch 200					
32	0.9346	0.9833	0.9547	0.9804	0.99
64	0.9160	0.9166	0.9076	0.9858	0.98
128	0.9433	0.9333	0.9342	0.9845	0.99

Eksperimen selanjutnya yaitu, mengatur *learning rate* yang ditunjukkan pada Tabel 6, hasil terbaik pelatihan dengan *learning rate* sebesar 0.001 dengan *epoch 150 batch 64* memperoleh nilai akurasi sebesar 0.9496, *precision* 0.9482, *Recall*

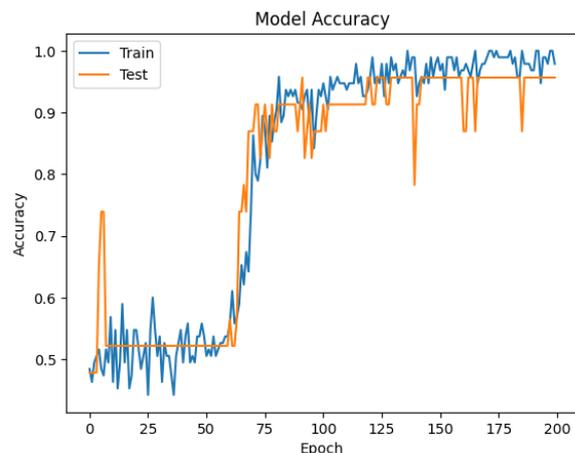
0.95, *F1-score* 0.9485, AUC 0.9776 dan ROC 1.00 dan *epoch 200 batch 32* akurasi 0.9496, *precision* 0.9346, *Recall* 0.9833, *F1-score* 0.9547, AUC 0.9804 dan ROC 0.99.

Tabel 6 Eksperimen Learning Rate

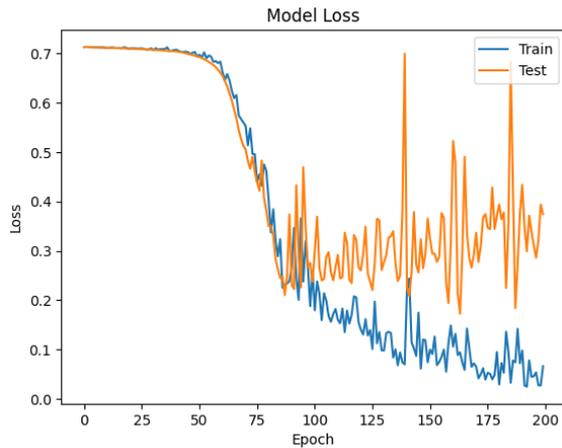
Batch size	Epoch	Precision	Recall	F1-score	AUC	ROC
Learning Rate 0.01						
64	150	0.5086	1.0	0.6742	0.5	0.50
32	200	0.5086	1.0	0.6742	0.5	0.50
Learning Rate 0.001						
64	150	0.9060	0.9833	0.9393	0.9791	1.00
32	200	0.9080	0.95	0.9264	0.9833	0.98
Learning Rate 0.0001						
64	150	0.9482	0.95	0.9485	0.9776	1.00
32	200	0.9346	0.9833	0.9547	0.9804	0.99

Dari hasil pelatihan dan pengujian yang dilakukan dengan menggunakan parameter dasar dan parameter *tuning* pada model yang diusulkan yaitu DNIN, diperoleh komposisi terbaik pada *epoch 150, batch 64, learning rate 0.0001* dan *epoch 200, batch 32, learning rate 0.0001*. Hal ini membuktikan bahwa perubahan nilai parameter tersebut mempengaruhi performa dan waktu komputasi.

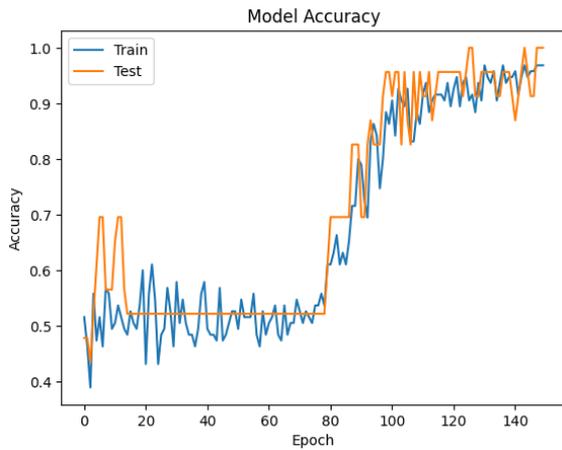
Hasil dari pengujian eksperimen terbaik tertuang dalam grafik akurasi dan *loss*. Pada Gambar 4. adalah garfik akurasi dari *epoch 200 batch 32*, Gambar 5. adalah garfik *loss* dari *epoch 200 batch 32*, Gambar 6. adalah garfik akurasi dari *epoch 150 batch 64*, Gambar 7. adalah garfik *loss* dari *epoch 150 batch 64*, Pada grafik tersebut terdapat dua variabel yaitu variabel latih dan variabel uji, variabel latih digambarkan dengan warna biru dan variabel uji digambarkan dengan warna oranye. Dari grafik tersebut dapat disimpulkan bahwa semakin tinggi nilai *epoch* maka akurasi semakin tinggi, begitu juga sebaliknya, semakin tinggi nilai *epoch* maka nilai *loss* semakin rendah.



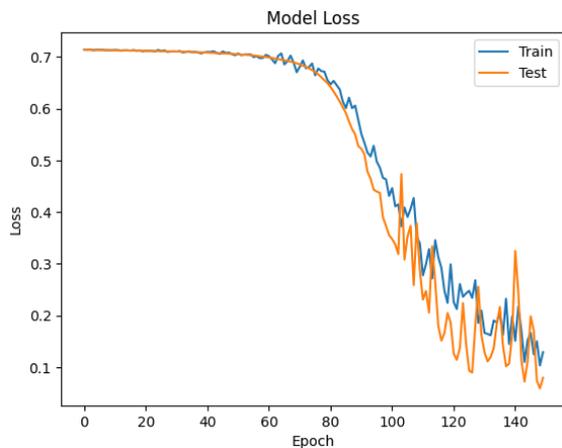
Gambar 4. Grafik Accuracy Epoch 200 Batch 32



Gambar 5. Grafik Loss Epoch 200 Batch 32



Gambar 6. Grafik Accuracy Epoch 150 Batch 64



Gambar 7. Grafik Loss Epoch 150 Batch 64

4.6. Hasil Evaluasi

Confusion matrix model DNIN (CNN + BiLSTM). Evaluasi model pada penelitian ini dilakukan menggunakan *confusion matrix*, dengan cara menghitung rasio prediksi benar dan salah. Evaluasi hasil digunakan dalam menghitung kinerja model yang telah dibuat, seperti pada grafik

dilihat pada Gambar 8. *Confusion matrix* pada nantinya akan menghasilkan beberapa model evaluasi diantaranya (Alghifari *et al.*, 2022):

Precision didapat dari nilai dari rasio prediksi *true positive* terhadap data bernilai *positive* pada keseluruhan data dijabarkan dalam persamaan (6).

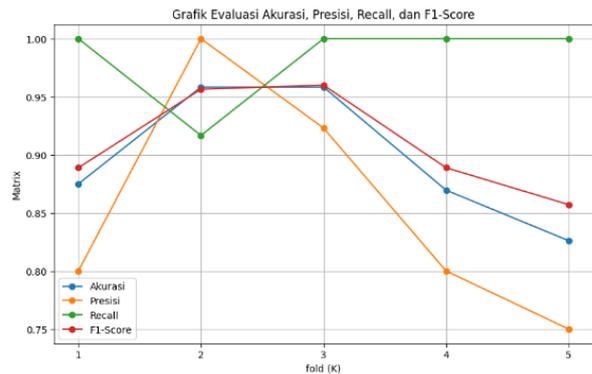
$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Recall didapat dari hasil rasio *true positive* terhadap data yang bernilai *true positive* pada keseluruhan data dijabarkan dalam persamaan (7).

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

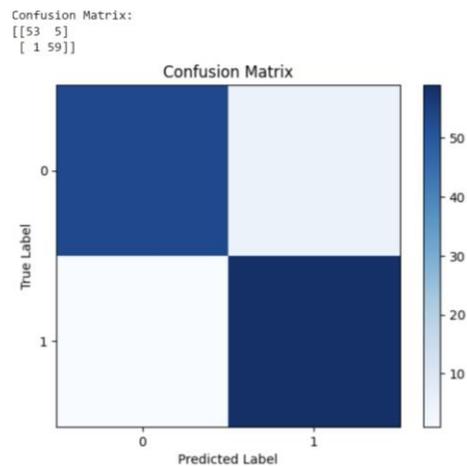
F1 score yaitu nilai *precision* dan *Recall* yang telah digabungkan dengan menggunakan persamaan (8).

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (8)$$

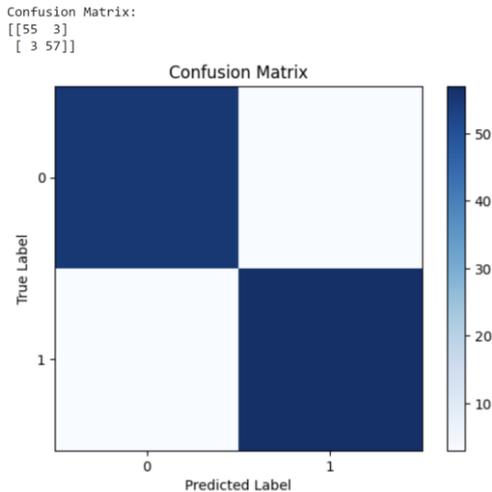


Gambar 8. Garfik Evaluasi Accuracy, Precision, Recall, F1-score

Evaluasi dari hasil pengujian eksperimen dengan parameter *epoch 200 batch 32* pada gambar 9, dan *epoch 150 batch 64* pada gambar 10, diperoleh hasil yang berberda. Pada *epoch 200 batch 32* pada gambar 9, terdapat nilai *true positive 59*, *true negative 53*, *false positive 5*, dan *false negative 1*. Kemudian, pada *epoch 150 batch 64* pada gambar 10, terdapat nilai *true positive 57*, *true negative 55*, *false positive 3*, dan *false negative 3*.



Gambar 9. Confusion matrix Epoch 200 Batch 32

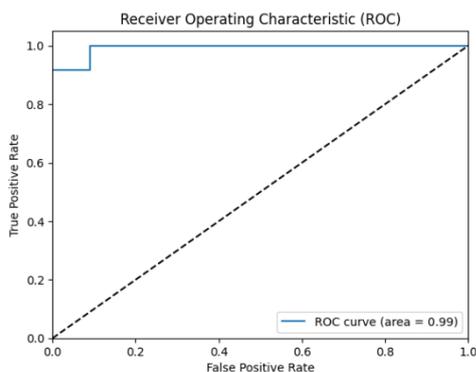


Gambar 10. Confusion matrix Epoch 150 Batch 64

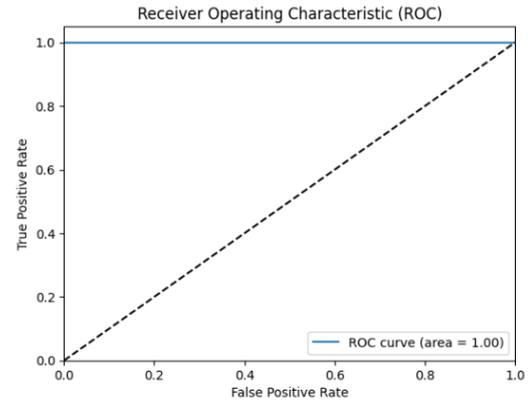
Receiver Operating Characteristics (ROC) model DNIN (CNN + BiLSTM). ROC merupakan kurva dua dimensi yang dibentuk dari plot True Positive Rate (TPR) dan plot False Positive Rate (FPR) yang sangat berguna untuk memvisualisasi serta mengevaluasi pengklasifikasi. TPR atau yang disebut juga dengan Sensitivity atau Recall merupakan kemampuan model dalam mengukur presentase positif yang terdeteksi oleh model dari semua kasus positif sebenarnya. FPR merupakan kemampuan model dalam mengukur presentase negatif yang salah diprediksi sebagai positif oleh model dari semua kasus yang negatif sebenarnya. Semakin tinggi nilai dari TPR, dan semakin lebih rendah nilai FPR, maka dikatakan model semakin baik. Nilai TPR dan FPR didasarkan pada confusion matrix. Perhitungan nilai TPR dan FPR dijabarkan dalam persamaan (9) dan (10) (Nurrohmah, 2022).

$$TPR = \frac{\text{Positives correctly classified}}{\text{Total positives}} = \frac{TP}{TP+FP} \quad (9)$$

$$FPR (1 - \text{specificity}) = \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}} = \frac{FP}{FP+TN} \quad (10)$$



Gambar 11 Grafik ROC Epoch 200 Batch 32



Gambar 12. Grafik ROC Epoch 150 Batch 64

Kemudian hasil evaluasi dari hasil pengujian eksperimen dengan parameter epoch 200 batch 32 dan epoch 150 batch 64 juga terdapat hasil berberda. Pada epoch 200 batch 32 terdapat nilai ROC curve 0.99. Pada epoch 150 batch 64 terdapat nilai ROC curve 1.00.

4.7 Komparasi Algoritma

Pada tahap ini dilakukan komparasi antara model Deep Neural Investigation Network (DNIN) yang diusulkan dengan model algoritma lainnya yaitu Convolutional Neural Network (CNN) dan Bidirectional Long Short-Term Memory (BiLSTM). Hasil komparasi model dapat dilihat pada tabel 7. Berdasarkan tabel 7 telah terbukti bahwa model Deep Neural Investigation Network (DNIN) dengan nilai akurasi tertinggi yaitu 0.9496 lebih unggul dibandingkan model CNN dan BiLSTM yang tanpa dikombinasikan.

Tabel 7 Komparasi Model

	Model	Precision	Recall	F1-score	AUC	ROC
CNN	Epoch 150	0.9036	0.8989	0.8985	0.9469	0.95
	Batch 64					
	Epoch 200	0.9247	0.9228	0.9227	0.9469	0.95
	Batch 32					
BiLSTM	Epoch 150	0.9433	0.9333	0.9342	0.9315	0.95
	Batch 64					
	Epoch 200	0.9283	0.8667	0.8948	0.9765	0.96
	Batch 32					
DNIN	Epoch 150	0.9482	0.95	0.9485	0.9776	1.00
	Batch 64					
	Epoch 200	0.9346	0.9833	0.9547	0.9804	0.99
	Batch 32					

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan Hasil analisis, eksperimen, dan pengujian terhadap data curah hujan dan bencana banjir menunjukkan bahwa model yang diajukan yaitu *Deep Neural Investigation Network* (DNIN), lebih unggul dibandingkan dengan metode *Convolutional Neural Network* (CNN) dan *Bidirectional Long Short-Term Memory* (BiLSTM) yang tidak digabungkan, pada eksperimen *epoch* 200, *batch* 64 dan *learning rate* 0.0001, diperoleh akurasi 0.9496, *precision* 0.9346, *Recall* 0.9833, *F1-score* 0.9547, *AUC* 0.9804 dan *ROC* 0.99. menunjukkan bahwa dengan karakteristik data *time series univariate*, model DNIN yang diusulkan dapat dijadikan alternatif untuk memprediksi banjir.

5.2 Saran

Untuk penelitian selanjutnya, disarankan melakukan lebih banyak percobaan atau eksperimen terhadap parameter dasar dan parameter *tuning* dari model DNIN seperti eksperimen jumlah *hidden layer*, nilai *dropout*, jumlah *filter*, dan jumlah *node* atau *neuron*, yang selanjutnya, dapat juga mengimplementasikan metode lain selain *hyperparameter tuning* dalam meningkatkan kinerja algoritma misalnya dengan menerapkan metode *feature selection* dan *ensemble learning*.

Daftar Pustaka

- A'ziziyah A.A., Nugroho. I.I., Sabillillah, R., Aji, B.A.S., Amiroh, K., 2022. Perbandingan Sistem Deteksi Banjir Menggunakan Algoritma Naive Bayes Dan K-NN Berbasis IOT. Indonesian Journal on Computer and Information Technology: Vol. 7(1). <https://doi.org/10.31294/ijcit.v7i1.12394>
- Aezwani, W., Josdi, N.L.N.B., Man, M., Triana, Y.S., 2022. An evaluation of artificial neural networks and random forests for heart disease prediction. Journal of Hunan University Natural Sciences, 49(2), 41–49. <https://doi.org/10.55463/issn.1674-2974.49.2.4>
- Aini, Y.K., Santoso, T.B., Dutono, T., 2021. Pemodelan CNN Untuk Deteksi Emosi Berbasis Speech Bahasa Indonesia. In Jurnal Komputer Terapan: Vol. 7(1). <https://doi.org/10.35143/jkt.v7i1.4623>
- Alghifari, D R., Edi, M., Firmansyah, L., 2022. Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia. Jurnal Manajemen Informatika (JAMIKA), 12(2), 89-99. <https://doi.org/10.34010/jamika.v12i2.7764>
- Asmara, R. A., Prasetyo, A., Stevani, S., Hapsari, R. I., 2021. Prediksi Banjir Lahar Dingin Pada Lereng Merapi Menggunakan Data Curah Hujan Dari Satelit. Studi, Jurnal Informatika Polinema: Vol 7(2). <https://doi.org/10.33795/jip.v7i2.494>
- Azizulhaq, M. A., Suhendi, A., Setianingsih, C., 2021. Dashboard Sistem Peringatan Dini Prediksi Banjir Menggunakan Metode Radial Basis Function Berbasis Web Web Based Dashboard for Flood Early Warning Prediction System Using Radial Basis Function Method. e-Proceeding of Engineering: 8(1), 334.
- Batbaatar, E., Li, M., Ryu, K.H., 2019. Semantic-Emotion Neural Network for Emotion Recognition From Text. IEEE Access: vol. 7, pp. 111866-111878, <https://doi.org/10.1109/ACCESS.2019.2934529>
- Delimayanti, M.K., Sari, R., Laya, M., Faisal, M.R., Pahrul, 2021. Edu Komputika Journal Pemanfaatan Metode Multiclass-SVM pada Model Klasifikasi Pesan Bencana Banjir di Twitter. In Edu Komputika (Vol. 8, Issue 1). <http://journal.unnes.ac.id/sju/index.php/edukom>
- Dwiasnati, S., Devianto, Y., 2021. Optimasi Prediksi Bencana Banjir menggunakan Algoritma SVM untuk penentuan Daerah Rawan Bencana Banjir. Prosiding SISFOTEK, 5(1),202–207.
- Fitriana, D., Gunawan, W., Sari, A.P., 2022. Studi Komparasi Algoritma Klasifikasi C5.0, SVM dan Naive Bayes dengan Studi Kasus Prediksi Banjir Comparative Study of Classification Algorithm between C5.0, SVM and Naive Bayes with Case Study of Flood Prediction. In Februari (Vol. 21, Issue 1). <https://doi.org/10.33633/tc.v21i1.5348>
- Fitriyaningsih, I., Basani, Y., 2019. Prediksi Kejadian Banjir dengan Ensemble Machine Learning Menggunakan BP-NN dan SVM. Jurnal Teknologi dan Sistem Komputer, vol. 7, no. 3, pp. 93-97. <https://doi.org/10.14710/jtsiskom.7.3.2019.93-97>
- Graves, A., 2013. Generating Sequence with Recurrent Neural Networks. Computation Science. <https://doi.org/10.1007/978-3-642-24797-2>
- Hasanah, M. A., Soim, S., Handayani, A.S., 2021. Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir. In Journal of Applied Informatics and Computing (JAIC) (Vol. 5, Issue 2). <https://doi.org/10.30871/jaic.v5i2.3200>
- Hermanto, D.T., Setyanto, A., Luthfi, E.T., 2021. Algoritma LSTM-CNN untuk Sentimen Klasifikasi dengan Word2vec pada Media Online LSTM-CNN Algorithm for Sentiment Classification with Word2vec on Online Media. Creative Information Technology Journal (CITEC

- JOURNAL), 8(1), 64–77.
<https://doi.org/10.24076/citec.2021v8i1.264>
- Intan, I., Ghani, A.D., Nurdin, Koswara, A.T.C., 2021. Analisis Performansi Prakiraan Cuaca Menggunakan Algoritma Machine Learning Performance Analysis of Weather Forecasting using Machine Learning Algorithms. *Jurnal Pekommas*: Vol. 6(2), 1–8.
<https://doi.org/10.30818/jpkm.2021.2060221>
- Karno, A.S.B., Hastomo, W., Efendi, Y., Irawati, D.R., 2021. Konferensi Nasional Ilmu Komputer (KONIK) 2021 Arsitektur Alexnet Convolution Neural Network (CNN) Untuk Mendeteksi Covid-19 Image Chest-Xray. *Proceeding KONIK (Konferensi Nasional Ilmu Komputer: Vol. 5(1), 482–485.*
- Karyadi, Y., Santoso, H., 2022. Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU. *Jurnal Teknik Informatika dan Sistem Informasi*: Vol. 9(1), 671-684.
<https://doi.org/10.35957/jatisi.v9i1.1588>
- Liu, W., Liu, P., Yang, Y., Gao, Y., and Yi, J., 2017. An attention-based syntax-tree and tree-LSTM model for sentence summarization. *International Journal of Performability Engineering*, 13(5), 775–782.
<https://doi.org/10.23940/ijpe.17.05.p20.775782>
- Masri, F., Saepudin, D., Aditya, D., 2020. Forecasting of Sea Level Time Series using Deep Learning RNN, LSTM, and BiLSTM, Case Study in Jakarta Bay, Indonesia. *e-Proceeding of Engineering: Vol.7(2), 8544.*
- Mustakim, F., Hayati, N., 2021. Algoritma Artificial Neural Network pada Text-based Chatbot Frequently Asked Question (FAQ) Web Kuliah Universitas Nasional. *Jurnal Teknologi Informasi Dan Komunikasi*, 5(4), 2021.
<https://doi.org/10.35870/jtik.v5i4.261>
- Nugroho, A.S., Witarto, A.B., Handoko, D., 2003. Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika 1. <http://asnugroho.net>
- Nugroho, K.S., Akbar, I., Suksmawati, A.N., Istiadi, 2023. Deteksi Depresi dan Kecemasan Pengguna Twitter Menggunakan Bidirectional LSTM. *The 4th Conference on Innovation and Application of Science and Technology (CIASTECH 2021)*
- Nurrohmah, H., 2022. Klasifikasi Berita Hoax Berbahasa Indonesia Menggunakan Bidirectional Long Short-Term Memory (Bi-LSTM) (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).
- Pisner, D.A., Schnyer, D.M., 2019. Support vector machine. In *Machine Learning: Methods and Applications to Brain Disorders* (pp. 101–121). Elsevier. <https://doi.org/10.1016/B978-0-12-815739-8.00006-7>
- Prabuddhi, W.A.M., Seneviratne, B.L.D., 2020. Long Short Term Memory Modelling Approach for Flood Prediction: An Application in Deduru Oya Basin of Sri Lanka. *20th International Conference on Advances in ICT for Emerging Regions (ICTer)*: pp. 226-231,
<https://doi.org/10.1109/ICTer51097.2020.9325438>
- Rahmadani, S., Rahayu, C.S., Salim, A., Cahyo, K.N., 2022. Deteksi Emosi Berdasarkan Wicara Menggunakan Deep Learning Model. *Jurnal Informatika Teknologi dan SAINS (JINTEKS)*: vol. 4(3), pp. 220-224.
<https://doi.org/10.51401/jinteks.v4i3.1952>
- Rangkuti, M.Y.R., Alfansyuri, M.V., Gunawan, W., 2021. Penerapan Algoritma K-Nearest Neighbor (KNN) dalam Memprediksi dan Menghitung Tingkat Akurasi Data Cuaca di Indonesia. 2(2).
- Riza, H., Santoso, E.W., Tejakusuma, I.G., Prawiradisastra, F., Prihartanto, P., 2020. Utilization Of Artificial Intelligence To Improve Flood Disaster Mitigation. *Jurnal Sains Dan Teknologi Mitigasi Bencana*, 15(1), 1–11.
<https://doi.org/10.29122/jstmb.v15i1.4145>
- Rizky, M.G., 2021. TA: Analisis Perbandingan Metode LSTM dan BiLSTM untuk Klasifikasi Sinyal Jantung Phonocardiogram (Doctoral dissertation, Universitas Dinamika).
- Rohani, A., Taki, M., AbdollahPour, M., 2018. A novel soft computing model (Gaussian process regression with K-fold cross validation) for daily and monthly solar radiation forecasting (Part: I). *Renewable Energy*: Vol. 115, 411-422.
<https://doi.org/10.1016/j.renene.2017.08.061>
- Sandiwarno, S., Niu, Z., Nyamawe, A.S., 2023. A Novel Hybrid Machine Learning Model for Analyzing E-Learning Users' Satisfaction, *International Journal of Human-Computer Interaction*, [10.1080/10447318.2023.2209986](https://doi.org/10.1080/10447318.2023.2209986)
- Sari, A.P, Hakim, E.A., Prasetya, D.A., Arifuddin, R., Adi, P.D.P., 2021. Sistem Prediksi Kecepatan dan Arah Angin Menggunakan Bidirectional Long Short-Term Memory. In *Seminar Keinsinyuran. Seminar Keinsinyuran 2021.*
- Septiandi, L.A., Yuniarno, E.M., Zaini, A., 2021. Deteksi Kedipan dengan Metode CNN dan Percentage of Eyelid Closure (PERCLOS). *JURNAL TEKNIK ITS*, 10(1).
<https://doi.org/10.12962/j23373539.v10i1.61174>
- Syamsurizal, Cumel, Zamri, D., Rahmadden, 2022. Perbandingan Metode Data Mining untuk Prediksi Banjir Dengan Algoritma Naïve Bayes dan KNN. *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat*, 1(1), 40-48
- Triyanto, S., Sunyoto, A., Arief, M.R., 2021. Analisis Klasifikasi Bencana Banjir Berdasarkan Curah Hujan Menggunakan Algoritma Naïve Bayes. *Joisie Journal of Information System And Informatics Engineering*, 5(Desember), 109–117.
<https://doi.org/10.35145/joisiej.v5i2.1785>

- Utami, A. S., Rini, D. P., and Lestari, E., 2021. Prediksi Cuaca di Kota Palembang Berbasis Supervised Learning Menggunakan Algoritma K-Nearest Neighbour. *Jurnal Penelitian Ilmu dan Teknologi Komputer*: Vol. 13(1).
- Widiasari, I.R., Nugoho, L.E., Widyawan, Efendi, R., 2018. Context-based Hydrology Time Series Data for A Flood Prediction Model Using LSTM. 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE): pp. 385-390. <https://doi.org/10.1109/ICITACEE.2018.8576900>
- Widiputra, H., Mailangkay, A., Gautama, E., 2021. Prediksi Indeks BEI dengan Ensemble Convolutional Neural Network dan Long Short-Term Memory. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*: Vol. 5(3), 456–465. <https://doi.org/10.29207/resti.v5i3.3111>
- Wulandari, A., 2022. Analisis Kinerja Algoritma Cnn dan LSTM untuk Memprediksi Tinggi Muka Air di DKI Jakarta, Indonesia (Doctoral dissertation, Universitas Mercu Buana Jakarta).
- Yoga, I.K., Prasetyowati, S.S., Sibaroni, Y., 2022. Prediction And Mapping Rainfall Classification Using Naive Bayes And Simple Kriging. *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*: Vol 7(4). <https://doi.org/10.29100/jipi.v7i4.3264>