

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR GAMBAR.....	2
DAFTAR TABEL.....	3
ABSTRAK.....	4
BAB 1. PENDAHULUAN	5
1.1. Latar Belakang	5
1.2. Rumusan Masalah.....	6
1.3. Tujuan	6
1.4. Manfaat	6
1.5. Batasan Masalah	7
BAB 2. PERANCANGAN DAN PENERAPAN.....	8
2.1. Desain Perangkat Keras	8
2.2. Perangkat lunak web aplikasi.....	9
2.3. <i>Machine learning</i> desain.....	10
BAB 3. HASIL DAN PEMBAHASAN	14
3.1. Pengujian pembacaan sensor ultrasonik HC-SR04.....	14
3.2. Pengujian pengambilan gambar pada <i>webcam</i>	15
3.3. Pengujian <i>machine learning</i>	17
BAB 4. KESIMPULAN	20
DAFTAR PUSTAKA	21

DAFTAR GAMBAR

Gambar 2.1. Blok diagram sistem.....	9
Gambar 2.2 Halaman <i>dashboard</i> data <i>display</i>	10
Gambar 2.3. Arsitektur <i>deep learning</i>	11
Gambar 2.4. <i>Flowchart</i> algoritma prediksi.....	12
Gambar 3.1. Hasil gambar tangkapan kamera dikirim ke <i>database</i> dan ditampilkan pada web	16
Gambar 3.2. Pengujian sistem secara keseluruhan	16
Gambar 3.3. Tampilan status kondisi	17
Gambar 3.4. Pengujian algoritma <i>machine learning</i>	18

DAFTAR TABEL

Tabel 3.1. Hasil pembacaan ketinggian permukaan air	14
---	----

ABSTRAK

Pada penelitian ini, kami mengusulkan pengembangan integrasi pendekatan sistem prediksi banjir menggunakan implementasi pendekatan jaringan syaraf tiruan pada sistem backend untuk meningkatkan akurasi hasil. Dalam penelitian ini, kami mengusulkan pengembangan integrasi pendekatan sistem prediksi banjir menggunakan pendekatan jaringan syaraf tiruan pada sistem backend. Pada perangkat Internet of Things(IoT) yang digunakan terdiri dari, raspberry pi yang dihubungkan dengan sensor ultrasonik dan modul kamera web. Sensor ultrasonik digunakan untuk mengukur jarak ke permukaan air, sedangkan modul kamera web digunakan untuk mengambil foto kondisi sungai saat ini. Eksperimen kami menunjukkan bahwa sistem kami berhasil memantau kondisi ketinggian air sungai, menampilkan data di platform web, dan menganalisis data untuk prediksi banjir menggunakan pembelajaran mesin dengan tingkat kesalahan MSE 0,038%.

Kata kunci: internet of things (IoT), deep learning, raspberry pi, prediksi banjir

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Kejadian bencana banjir pada dasarnya tidak dapat diprediksi oleh manusia. Namun sangat berbahaya dan sering menimbulkan korban jiwa dan harta benda. Dari perubahan iklim serta pola cuaca yang tidak menentu, sistem drainase yang cerdas penting dalam pengelolaan banjir untuk mencegah kemungkinan hilangnya nyawa dan harta benda di kota-kota dengan tingkat urbanisasi tinggi (Chen,2019). Selama ini wilayah perkotaan di Indonesia rawan banjir akibat curah hujan yang tinggi. Teknologi saat ini dapat menjadi alat dalam membantu mengatasi kurangnya kesadaran umat manusia akan terjadinya bencana banjir. Di tahun 2019 telah terjadi sebanyak lebih dari 200 tragedi banjir yang terjadi di Indonesia yang, tertinggi terjadi pad tahun 2019 yaitu sebanyak lebih dari 800 banjir yang terjadi (Badan Meteorologi Klimatologi dan Geofisika (BMKG), 2019).

Saat ini untuk mengetahui adanya kemungkinan banjir pada wilayah rawan banjir masih berdasarkan perkiraan hasil pemantauan manual yang dilakukan oleh manusia. Antisipasi dalam menghadapi bencana banjir yang akan terjadi belum maksimal. Untuk itu dirancang suatu sistem pendeteksi yang dapat mendeteksi dan melakukan prediksi secara otomatis kapan atau dimana akan terjadi bencana banjir sesuai dengan kriteria yang telah ditentukan dan dirancang. Implementasi kemajuan Information Communication and Technology (ICT) memberikan kontribusi signifikan untuk mengurangi dampak banjir terhadap masyarakat dan properti (Maspo, 2018). Sistem tersebut merupakan sistem berbasis IoT (Internet of Things) yang mampu memantau ketinggian air di permukaan sungai. Sistem pemantauan banjir berbasis IoT dan prediksi banjir yang ditanami dengan artificial neural network (ANN) yang dirancang diharapkan dapat meningkatkan skalabilitas dan keandalan sistem pengelolaan banjir (Bande, 2017). Teknologi Internet of Things (IoT) telah digunakan dalam berbagai sektor, termasuk untuk memprediksi, memantau, dan mendeteksi banjir (Samikwa, 2020).

Dalam dunia industri, diagnosis prediksi pada suatu peralatan/mesin sangat dibutuhkan dan menjadi objek penelitian yang luas. Model kecerdasan buatan (AI) telah memberikan kontribusi yang signifikan selama beberapa dekade terakhir dengan memberikan peningkatan akurasi dan solusi ekonomis (Tabbussum, 2021). Salah satunya dengan menggunakan deep learning sebagai metode diagnosis dan prediksi. Deep learning dipilih dengan tujuan untuk mempercepat analisis dalam mesin dan meningkatkan efisiensi hasil diagnosis. Namun deep

learning cenderung bergantung pada jumlah sampel data pelatihan, yang berarti metode tersebut memerlukan sampel data dalam jumlah besar.

Dari penelitian yang telah dilakukan oleh (Gomathy, 2021), telah mampu membangun sebuah perangkat IoT yang berfungsi untuk monitoring kelembaban, suhu, dan ketinggian air, namun pada penelitian tersebut belum adanya algoritma yang mendalam untuk pengolahan data dari sensor. Lalu penelitian yang dilakukan oleh (Indrasari, 2018) juga telah mampu membuat sistem pemantau ketinggian air dengan kesalahan maksimum 4,3% dan menampilkan informasi ketinggian air dalam bentuk SMS, namun masih belum adanya juga algoritma mendalam untuk pengolahan data sensor. Namun penelitian yang dilakukan oleh (Khan, 2020) mampu membangun sistem berbasis IoT untuk memprediksi banjir dan sudah menerapkan metode machine learning berbasis random forest dengan akurasi hampir mendekati 93%, namun data input yang diolah masih belum menggunakan data gambar. Penelitian oleh (Manokij, 2019) telah mampu mengembangkan model klasifikasi dan prediksi menggunakan machine learning menggunakan convolutional neural network (CNN) untuk klasifikasi dan regresi linear untuk prediksi, namun RMSE yang didapat adalah sebesar 4,53%.

1.2. Rumusan Masalah

Dari penelitian yang dilakukan terdapat beberapa rumusan masalah, sebagai berikut:

- 1.2.1. Bagaimana merancang sebuah sistem yang dapat mendeteksi dini bencana banjir?
- 1.2.2. Bagaimana menerapkan metode deep learning untuk memprediksi banjir?
- 1.2.3. Bagaimana menerapkan sistem pemantau banjir yang dapat diakses secara online?

1.3. Tujuan

Dari penelitian yang dilakukan ini terdapat beberapa tujuan, sebagai berikut:

- 1.3.1. Merancang sebuah sistem yang dapat mendeteksi dini bencana banjir
- 1.3.2. Menerapkan metode deep learning untuk memprediksi banjir
- 1.3.3. Menerapkan sistem pemantau banjir yang dapat diakses secara online

1.4. Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah membuat sebuah sistem yang mampu memprediksi dini bencana banjir dan hasil prediksi dapat ditampilkan secara *online* berbasis halaman web.

1.5. Batasan Masalah

Berikut ini adalah batasan masalah yang ada penelitian ini, sebagai berikut:

- 1.5.1. Modul yang digunakan adalah *raspberry pi* dan kamera *webcam* sebagai mikroprosesor dan media kamera
- 1.5.2. Metode prediksi yang digunakan adalah *deep learning*
- 1.5.3. Hasil pemantauan kamera dalam bentuk gambar

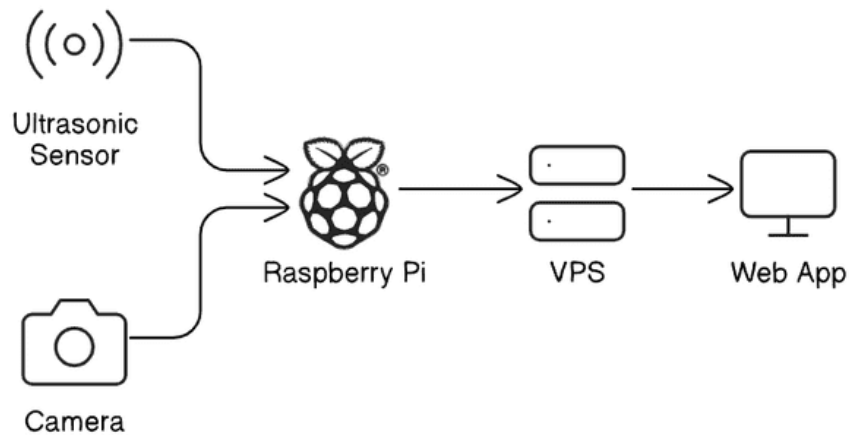
BAB 2. PERANCANGAN DAN PENERAPAN

2.1. Desain Perangkat Keras

Perancangan perangkat keras sistem, terdiri dari *raspberry pi 3*, sensor ultrasonik HC-SR04, dan kamera web. *raspberry pi 3* digunakan sebagai pengontrol utama untuk membaca data dari sensor ultrasonik dan kamera *webcam*. *Raspberry pi* juga terhubung dengan koneksi jaringan Internet Wifi untuk mengirimkan data ke *server*.

Sensor ultrasonik HC-SR04 digunakan untuk mengukur permukaan air sungai dari atas. Itu menggunakan gelombang ultrasonik untuk mengukur jangkauan objek. Pertama, sensor mengirimkan gelombang ultrasonik ke objek. Setelah itu sensor menerima gelombang ultrasonik yang dipantulkan benda, kemudian menghitung waktu yang dibutuhkan gelombang ultrasonik untuk sampai kembali. Hasil perhitungan kemudian dapat direpresentasikan dengan jarak benda. Kamera web digunakan untuk mengabadikan foto kondisi sungai secara periodik. Terhubung ke *raspberry pi 3* dengan kabel USB. Program untuk menangkap data sensor ultrasonik dan mengambil foto dari kamera *webcam* yang berjalan pada *raspberry pi 3* dikembangkan dengan bahasa pemrograman *python*.

Semua bagian tersebut disusun menjadi suatu sistem tertanam yang terintegrasi dan dapat bekerja secara bersamaan. *Raspberry pi 3* secara berkala mengirimkan semua data ke server database. Proses tersebut terus berjalan berulang kali selama *raspberry pi 3* dihidupkan dan dapat terhubung ke jaringan koneksi Internet Wifi. Jika koneksi internet gagal maka *raspberry pi 3* menyimpan datanya secara internal. Ketika koneksi internet kembali *online*, *raspberry pi 3* kemudian mengirimkan semua datanya ke *server*. Berikut ini adalah rancangan diagram blok perangkat keras dan perangkat lunak sistem yang terintegrasi.

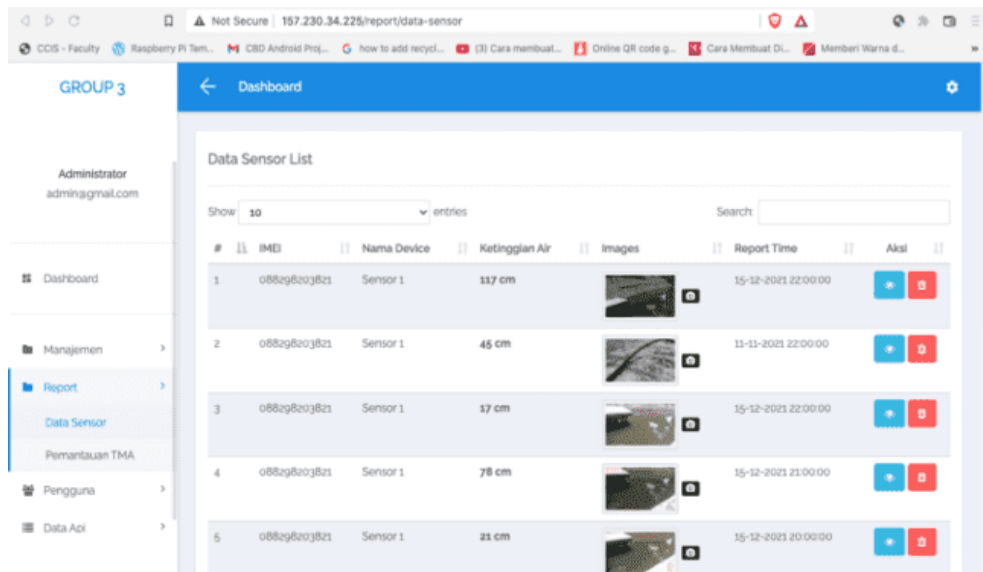


Gambar 2.1. Blok diagram sistem

Gambar 2.1 merupakan diagram blok sistem. Terdapat 5 blok utama yaitu blok kamera dan blok sensor ultrasonik yang terhubung dengan blok *raspberry pi*. Blok *raspberry pi* akan mengolah data dari sensor dan kamera, kemudian data tersebut diolah dan dikirim ke database yang ada di dalam VPS (*Virtual Private Server*). Data dari database ditampilkan di web untuk melihat ketinggian air dan hasil tangkapan kamera. *Raspberry pi 3* dan database dihubungkan menggunakan koneksi jaringan Internet Wifi untuk mengirimkan data sensor dan data gambar.

2.2. Perangkat lunak web aplikasi

Perancangan perangkat lunak sistem ini terdiri dari platform web dan platform database. Platform web diterapkan pada VPS (*Virtual Private Server*) berbasis *cloud*. Halaman web dikembangkan dengan bahasa pemrograman PHP dengan *framework Laravel*. Aplikasi web yang dibuat dengan *framework PHP* akan lebih stabil dan aman (Laaziri, 2019). Terdiri dari fitur manajemen pengguna, menampilkan data sensor dan data gambar kondisi sungai dari database.



Gambar 2.2 Halaman *dashboard* data *display*

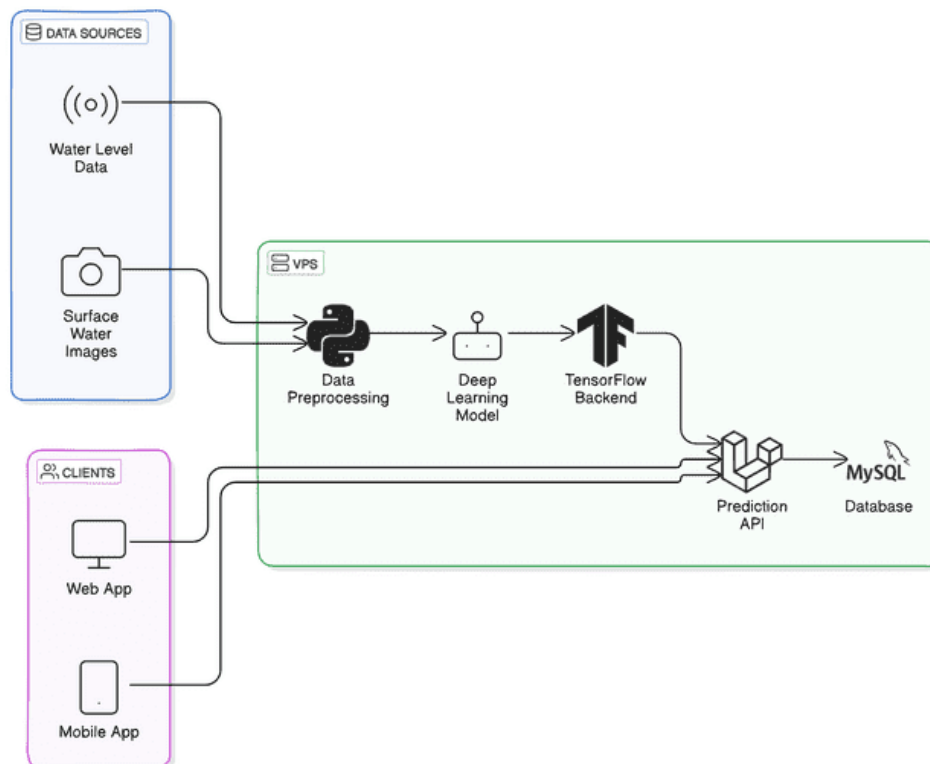
Gambar 2.2 menampilkan menu laporan data dari sistem. Terlihat pada terdapat informasi ketinggian air saat data diterima dan hasil gambar yang diambil. Setelah itu, aplikasi web terhubung dengan platform database. Platform database menggunakan database MySQL. Database ini dapat dihubungkan langsung ke aplikasi Web. Untuk implementasinya, aplikasi Web dan platform database MySQL diinstal pada satu VPS cloud kemudian dihubungkan satu sama lain. Keamanan platform dikelola oleh token dan otentikasi sesi. Oleh karena itu, autentikasi tersebut harus melindungi data aplikasi web dari akses yang tidak sah. Pada gambar 6 adalah halaman user management, untuk melakukan manajemen terhadap user dan hak akses user, untuk keamanan dari web aplikasi.

2.3. *Machine learning* desain

Teknologi yang canggih seperti *machine learning* menawarkan manfaat besar bagi sektor teknologi, karena sangat efektif dalam memantau karakteristik perilaku normal dan abnormal dari berbagai mesin (Rani, 2020). Desain *machine learning* pada sistem ini terdiri dari algoritma *deep learning* yang berjalan pada platform Jupyter Notebook. Metode *deep learning* memiliki efek tertentu dan secara efektif dapat meningkatkan efek pemrosesan pada citra digital (Zhang, 2022). Platform ini bekerja dengan *source code* berbasis bahasa pemrograman Python. Notebook Jupyter juga diterapkan pada VPS berbasis cloud yang sama. Algoritma *deep learning* yang dikembangkan dengan bahasa pemrograman Python dengan *library machine learning*. Penggunaan *library* dapat membantu menyederhanakan pengembangan algoritma *deep learning*. *Library Machine learning* yang digunakan adalah *Keras*, *NumPy*,

Pandas, dan *Tensorflow*. *Keras* adalah pustaka Python *open source*, gratis yang kuat untuk mengembangkan dan mengevaluasi model *deep learning*.

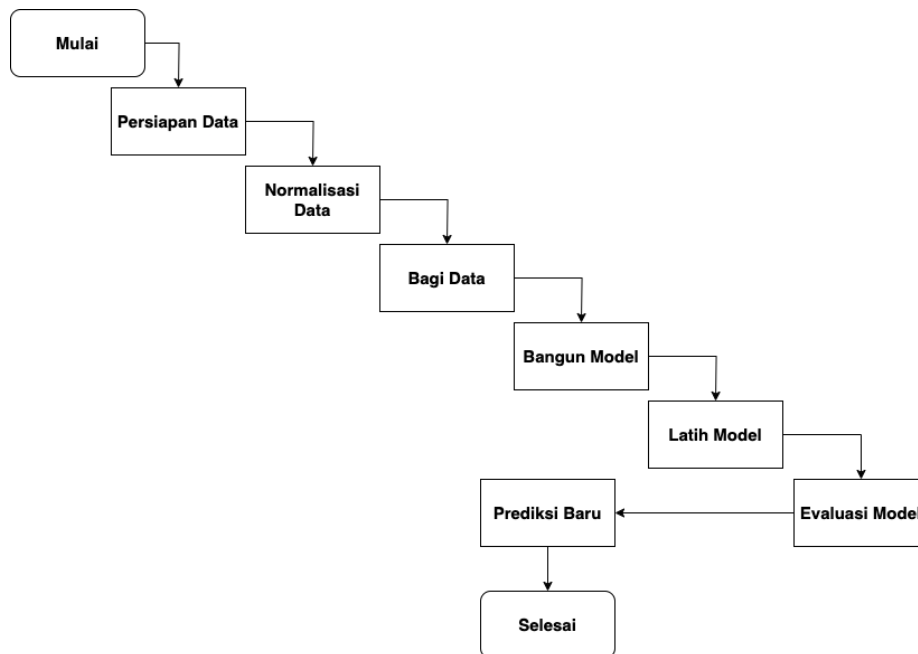
TensorFlow adalah perpustakaan Python untuk komputasi numerik cepat yang dibuat dan dirilis oleh Google. Ini adalah *library* dasar yang dapat digunakan untuk membuat model *deep learning* secara langsung atau dengan menggunakan *library wrapper* yang menyederhanakan proses yang dibangun di atas *TensorFlow*.



Gambar 2.3. Arsitektur *deep learning*

Gambar 2.3 menunjukkan arsitektur sistem yang digunakan untuk mendeteksi dan memprediksi ketinggian air menggunakan model *deep learning*. Arsitektur ini mencakup berbagai komponen mulai dari sumber data hingga antarmuka pengguna akhir. Data ketinggian air dan gambar permukaan air dikumpulkan dari sensor ultrasonik HC-SR04 dan kamera *webcam*. Selanjutnya, data ini dikirim ke tahap pra-pemrosesan dimana data dibersihkan, dinormalisasi, dan dipersiapkan untuk digunakan oleh model *deep learning*. Model *deep learning* menggunakan *TensorFlow* sebagai *backend* untuk melakukan komputasi prediksi. Hasil prediksi dari model ini disediakan melalui *prediction API* menggunakan *laravel*, yang memungkinkan berbagai aplikasi untuk mengakses hasil prediksi dengan mudah. Semua hasil prediksi dan data terkait disimpan dalam *database MySQL*, untuk keperluan analisis lebih lanjut dan pelacakan historis. Informasi mengenai ketinggian air dapat diakses oleh pengguna melalui

aplikasi web dan mobile, memberikan kemudahan akses untuk memonitor ketinggian air secara *real-time*. Dengan integrasi data ketinggian air dan analisis gambar permukaan air, sistem ini mampu memberikan prediksi yang lebih akurat dan andal, yang kemudian disajikan melalui antarmuka yang mudah digunakan.



Gambar 2.4. *Flowchart* algoritma prediksi

Pada gambar 2.4 adalah flowchart dari algoritma *deep learning* yang digunakan untuk melakukan predikis dini bencana banjir. Pada tahap awal adalah persiapan data ketinggian air dan kategori bahaya, berupa normal, awas, siaga, dan berbahaya. Kategori bahaya tersebut akan diencode menjadi nilai numerik untuk nantinya akan diolah sebagai label pada *deep learning*. Selanjutnya adalah tahap normalisasi dimana data ketinggian air dinormalisasi menggunakan *StandardScaler* untuk memastikan skala yang seragam. Selanjutnya ada tahap bagi data dimana data dibagi menjadi data latih dan data uji menggunakan *train_test_split*. Selanjutnya tahap untuk membangun model yang dimana model ini yang akan digunakan untuk melakukan prediksi dini bencana banjir, menggunakan model jaringan saraf tiruan (*neural network*) dibangun menggunakan *TensorFlow* dan *Keras*. Pada tahapan pembangunan mode ini terdapat beberapa bagian yang akan digunakan, pertama terdapat *input layer* yang berfungsi merujuk pada cara input data dimasukkan ke dalam model untuk diproses. Lalu ada penambahan *hidden layer* yang berfungsi untuk menentukan cara bagaimana *neuron-neuron* dalam layer tersebut akan menghitung dan mentransfer informasi dari *input* ke *output* agar dapat diolah. Pada *hidden layer* ini menggunakan fungsi aktivasi berupa *ReLU (Rectified*

Linear Unit). Tahapan terakhir pada membangun model ini adalah *ouput layer* yang bertujuan untuk menghasilkan *output* yang sesuai dengan pengolahan dan mendapatkan hasil prediksi yang. *Output layer* bertanggung jawab untuk mengubah representasi internal yang dihasilkan dari layer-layer sebelumnya menjadi bentuk yang dapat digunakan untuk pemecahan masalah yang menghasilkan prediksi, dimana fungsi aktivasi yang digunakan adalah *sigmoid*. Lalu setelah model dibuat, tahapan berikutnya adalah melakukan pengujian atau evaluasi model tersebut untuk mendapatkan seberapa besar akurasi yang dihasilkan oleh mode tersebut menggunakan data latih. Tahap terakhir adalah melakukan prediksi dengan model yang sudah dibangun.

BAB 3. HASIL DAN PEMBAHASAN

Untuk memverifikasi kinerja sistem pengembangan kami, kami melakukan percobaan di sebuah sungai kecil di Kota Depok, Jawa Barat, Indonesia. Eksperimen tersebut terdiri dari eksperimen perangkat keras, eksperimen perangkat lunak, dan eksperimen *machine learning*. Telah dilakukan pengujian terhadap kemampuan membaca data sensor ultrasonik dan pengambilan gambar dari *raspberry pi 3*, yang meliputi pengujian pengiriman data tersebut ke server database yang di-*deploy* pada platform *cloud*.

3.1. Pengujian pembacaan sensor ultrasonik HC-SR04

Pengujian dilakukan pada permukaan sungai dengan membandingkan hasil pengukuran sensor HC-SR04, untuk mengukur ketinggian permukaan air dari daratan dan dibandingkan dengan pengukuran dari meteran digital.

Tabel 3.1. Hasil pembacaan ketinggian permukaan air

Timestamp	Measured Height (cm)	Actual Height (cm)	Error (cm)	Akurasi
10:00:00	100	98	2	97.96
10:05:00	105	104	1	99.04
10:10:00	110	108	2	98.15
10:15:00	115	114	1	99.12
10:20:00	120	118	2	98.31
10:25:00	125	123	2	98.37
10:30:00	130	129	1	99.23
10:35:00	135	134	1	99.25
10:40:00	140	139	1	99.28







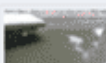














10:45:00	145	144	1	99.31
----------	-----	-----	---	-------

Tabel 3.1 di atas menyajikan data kalibrasi sensor ultrasonik yang digunakan untuk mendeteksi ketinggian air. Pengukuran dilakukan pada interval waktu lima menit, mulai dari pukul 10:00:00 hingga 10:45:00. Kolom-kolom dalam tabel ini mencakup waktu pengukuran (Timestamp), ketinggian air yang diukur oleh sensor (Measured Height), ketinggian air sebenarnya yang diketahui (Actual Height), kesalahan pengukuran (Error), dan persentase akurasi pengukuran (Accuracy).

Data menunjukkan bahwa terdapat perbedaan kecil antara ketinggian yang diukur oleh sensor. Kesalahan pengukuran berkisar antara 1 hingga 2 cm. Persentase akurasi sensor bervariasi dari 97.96% hingga 99.31%, yang menunjukkan performa yang sangat baik dari sensor dalam mendeteksi ketinggian air. Secara keseluruhan, tabel ini menunjukkan bahwa sensor ultrasonik memiliki akurasi yang tinggi dan dapat diandalkan untuk mendeteksi ketinggian air dengan kesalahan pengukuran yang relatif kecil. Hal ini penting dalam aplikasi-aplikasi yang memerlukan pengukuran ketinggian air yang presisi, seperti pengelolaan banjir seperti pada percobaan yang dilakukan ini.

3.2. Pengujian pengambilan gambar pada *webcam*

Pada pengujian pengambilan gambar dengan *webcam* ini akan mencoba melakukan pengambilan gambar pada permukaan air sungai dan mencobakan mengirim pada database dan ditampilkan di halaman web.

#	IMEI	Nama Device	Ketinggian Air	Images	Report Time	Aksi
1	088298203821	Sensor 1	117 cm		15-12-2021 22:00:00	 
2	088298203821	Sensor 1	45 cm		11-11-2021 22:00:00	 
3	088298203821	Sensor 1	17 cm		15-12-2021 22:00:00	 
4	088298203821	Sensor 1	78 cm		15-12-2021 21:00:00	 
5	088298203821	Sensor 1	21 cm		15-12-2021 20:00:00	 
6	088298203821	Sensor 1	23 cm		15-12-2021 19:00:00	 
7	088298203821	Sensor 1	77 cm		15-12-2021 15:00:00	 

Gambar 3.1. Hasil gambar tangkapan kamera dikirim ke *database* dan ditampilkan pada web



Gambar 3.2. Pengujian sistem secara keseluruhan



Lokasi : Device 1 - Kali Beji
Hari : Senin | 27/12/2021 | 12:41:23

TMA : 42 cm
TMA Normal : 0 - 119
TMA Siaga 3 : 120 - 249
TMA Siaga 2 : 250 - 299
TMA Siaga 1 : > 300

Status : **SIAGA 4 / NORMAL**

Prediksi TMA :

Pukul 12:45 : 76
Pukul 13:00 : 67
Pukul 14:00 : 80
Pukul 14:23 : 81
Pukul 15:00 : 77
Pukul 15:30 : 81
(C) 2021

12:41

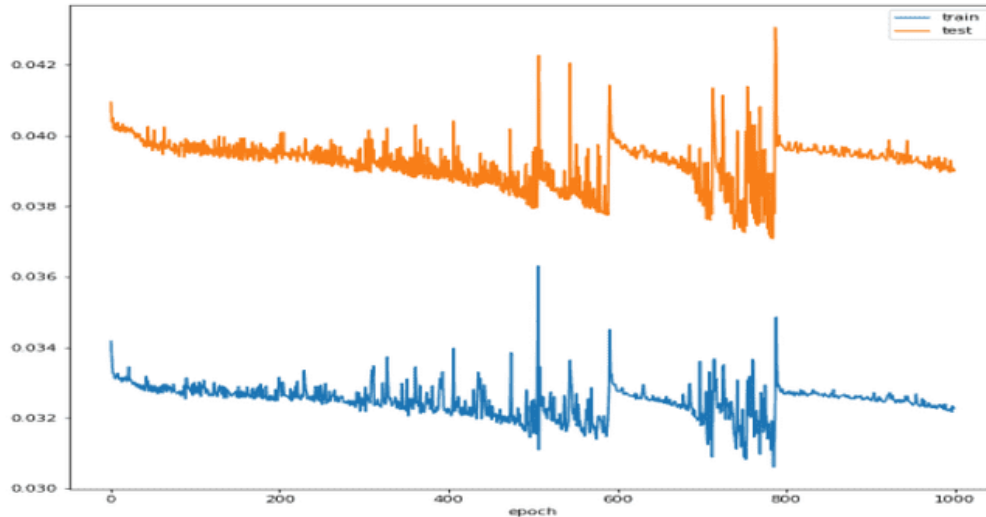
Gambar 3.3. Tampilan status kondisi

Pada gambar 3.1 adalah hasil pengambilan gambar dari webcam yang ada pada *raspberrypi* yang berhasil dijalankan, dan melakukan pengujian pengiriman data ke server *database* di *cloud Virtual Private Server (VPS)*, serta menampilkan gambar tangkapan pada halaman web dan terkirim dengan baik ketika koneksi Internet *Wifi* tersedia. Gambar 3.2 adalah pengujian keseluruhan sistem yang dimana terdapat kamera webcam dan juga sensor ultrasonik. Kami melakukan pengujian halaman web dan database di *VPS cloud*. Serta pada gambar 3.3 adalah tampilan status dan kondisi keadaan pada saat melakukan pengujian. Pengujian yang dilakukan meliputi pengujian halaman login, halaman pengelolaan pengguna, halaman tampilan data, dan halaman prediksi banjir. Semua halaman di halaman web kami uji secara *black box* dan berjalan dengan sukses seperti yang diharapkan. Disimpulkan bahwa semua fitur pada *platform* Web berhasil berfungsi sesuai harapan.

3.3. Pengujian *machine learning*

Eksperimen pengujian *machine learning* dengan algoritma *deep learning* telah diuji pada *Jupyter Notebook*. Aplikasi *Jupyter Notebook* yang di-*deploy* di VPS telah terhubung dengan database dan aplikasi web. Pengujian *machine learning* mencakup perbandingan data set

training dan data set *testing*. Metode *machine learning* sangat bergantung pada penggunaan teknik komputasi yang tepat dalam mendesain algoritma dalam data *training* (Mosavi,2018). Data yang diperoleh berupa data permukaan air yang diukur dengan sensor ultrasonik HC-SR04 dan data prediksi cuaca.



Gambar 3.4. Pengujian algoritma *machine learning*

Gambar 3.4 menunjukkan bentuk grafik data training dan data testing, dimana data training ditampilkan dengan garis biru dan data testing ditampilkan dengan garis oranye. Dari gambar 8 terlihat bahwa data testing mampu mengikuti jalur data training. Jarak antara kedua data hasil tersebut tidak terlalu besar, kurang lebih $\pm 0,01$. Dari pengujian machine learning, kita dapat menyimpulkan bahwa target memiliki domain yang hampir sama dengan pengujian data. Artinya machine learning dengan algoritma *deep learning* telah berhasil memprediksi hasil target. Dari pengujian ini, kami dapat menyimpulkan bahwa machine learning cukup akurat dalam memprediksi target data sesuai dengan kumpulan data training yang diberikan.

Kami juga menghitung nilai error dari machine learning. Untuk menghitungnya kami menggunakan perhitungan fungsi *Mean Squared Error (MSE)*. Dengan rumus sebagai berikut.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 \quad (1)$$

Pada persamaan 1 di atas, n adalah jumlah total data pengamatan, y_i adalah nilai sebenarnya pada pengamatan ke- i , y'_i adalah nilai prediksi pada pengamatan ke- i , $(y_i - y'_i)^2$ adalah kesalahan kuadrat antara nilai sebenarnya dan nilai prediksi untuk pengamatan ke- i . Dari persamaan MSE di atas maka dapatlah hasil MSE atau *error* dari *machine learning* sebesar 0,0038%.

Kesimpulannya, keakuratan *machine learning* menggunakan algoritma *deep learning* ini masih dapat diandalkan untuk digunakan sebagai alat untuk prediksi dini banjir pada sistem.

BAB 4. KESIMPULAN

Penelitian ini menyajikan pengembangan sistem prediksi bencana banjir berbasis perangkat IoT dengan implementasi *deep learning* pada sistem *backend*. Perangkat IoT terdiri dari *raspberry pi*, sensor ultrasonik HC-SR04 dan kamera *webcam*.

Algoritma machine learning yang kami gunakan adalah deep learning. Deep learning memiliki akurasi lebih tinggi dibandingkan algoritma machine learning lainnya. Dari pengujian yang kami lakukan terbukti bahwa machine learning yang diterapkan, dapat memprediksi hasil secara akurat dengan nilai MSE sebesar 0,038%.

Untuk menampilkan data kami menggunakan aplikasi web dengan framework PHP dan Laravel. Platform database yang digunakan adalah MySQL sebagai tempat penyimpanan data. Seluruh sistem bekerja dengan baik sesuai dengan yang diharapkan pada saat dilakukan pengujian sistem integrasi. Namun pada tulisan ini kami hanya menggunakan satu perangkat IoT. Untuk penelitian selanjutnya perlu ditingkatkan dengan penerapan lebih dari satu perangkat IoT. Perangkat dapat ditempatkan pada beberapa titik lokasi sungai untuk mendapatkan data yang lebih banyak sehingga machine learning dapat memperoleh data yang lebih banyak dan pengolahan data dapat lebih akurat.

DAFTAR PUSTAKA

- Badan Meteorologi Klimatologi dan Geofisika (BMKG). 2019. *Mengenal Bencana Hidrometeorologi*.
- Bande, S., & Shete, V. v. (2017). Smart flood disaster prediction system using IoT & neural networks. *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 189–194. <https://doi.org/10.1109/SmartTechCon.2017.8358367>.
- Chen, C., & Pang, Y. (2019). Exploring Machine Learning Techniques for Smart Drainage System. *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, 63–70. <https://doi.org/10.1109/BigDataService.2019.00015>.
- Gomathy, C. K., G G Lasya, P., & Kumar, H. (2021). A Study on Iot Based Flood Detection Management System. *International Journal of Engineering and Advanced Technology*, 10(4), 130–133. <https://doi.org/10.35940/ijeat.d2407.0410421>.
- Indrasari, W., Iswanto, B. H., & Andayani, M. (2018). Early Warning System of Flood Disaster Based on Ultrasonic Sensors and Wireless Technology. *IOP Conference Series: Materials Science and Engineering*, 335(1), 12005. <https://doi.org/10.1088/1757-899X/335/1/012005>.
- Khan, W., Hussain, A. J., Alaskar, H., Baker, T., Ghali, F., A-Jumeily, D., & Al-Shamma'a, A. (2020). Prediction of Flood Severity Level via Processing IoT Sensor Data Using a Data Science Approach. *IEEE Internet of Things Magazine*, 3(4), 10–15. <https://doi.org/10.1109/IOTM.0001.1900110>.
- Laaziri, M., Benmoussa, K., Khouilji, S., & Kerkeb, M. L. (2019). A Comparative study of PHP frameworks performance. *Procedia Manufacturing*, 32, 864–871. <https://doi.org/https://doi.org/10.1016/j.promfg.2019.02.295>.
- Manokij, F., Sarinnapakorn, K., & Vateekul, P. (2019). Forecasting Thailand's Precipitation with Cascading Model of CNN and GRU. *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1–6. <https://doi.org/10.1109/ICITEED.2019.8929975>.
- Maspo, N., Harun, A. N., Goto, M., Naw, M. N. M., & Haron, N. A. (2018). Development of Internet of Thing (IoT) technology for flood prediction and Early Warning System (EWS). <https://api.semanticscholar.org/CorpusID:115143488>.
- Mosavi, A., Ozturk, P., & Chau, K. (2018). Flood Prediction Using Machine Learning Models: Literature Review. *Water*, 10(11). <https://doi.org/10.3390/w10111536>.
- Rani, D. S., Jayalakshmi, G. N., & Baligar, V. P. (2020). Low Cost IoT based Flood Monitoring System Using Machine Learning and Neural Networks: Flood Alerting and Rainfall Prediction. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 261–267. <https://doi.org/10.1109/ICIMIA48430.2020.9074928>.
- Samikwa, E., Voigt, T., & Eriksson, J. (2020). Flood Prediction Using IoT and Artificial Neural Networks with Edge Computing. *2020 International Conferences on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 234–240. <https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00053>.
- Singh, S., Kakade, V., Mehta, V., & Gunjan, R. (2021). FLOOD CONTROL WITH WATER LEVEL PREDICTION SYSTEM USING MACHINE LEARNING. *International Research Journal of Engineering and Technology*. www.irjet.net.

- Tabbussum, R., & Dar, A. Q. (2021). Performance evaluation of artificial intelligence paradigms—artificial neural networks, fuzzy logic, and adaptive neuro-fuzzy inference system for flood prediction. *Environmental Science and Pollution Research*, *28*(20), 25265–25282. <https://doi.org/10.1007/s11356-021-12410-1>.
- Zhang, J. (2022). Processing and compression of underwater image based on deep learning. *Optik*, *271*, 170168. <https://doi.org/https://doi.org/10.1016/j.ijleo.2022.170168>.