



JURNAL RISET TEKNIK KOMPUTER (JURTIKOM)

Halaman Jurnal: <https://journal.smartpublisher.id/index.php/jurtikom>

Halaman UTAMA Jurnal : <https://journal.smartpublisher.id>



DOI: <https://doi.org/10.69714/gn6nym06>

IMPLEMENTASI SISTEM PREDIKSI CURAH HUJAN DENGAN PENERAPAN JARINGAN SYARAF TIRUAN BERBASIS *WEBSITE*

Syaifudin Fendi Prasetyo ^{a*}, Tino Feri Efendi ^b, Muqorobin ^c

^a Teknologi / Informatika, syaifudinfindip@gmail.com, ITB AAS Indonesia

^b Teknologi / Informatika, tinoferi8@gmail.com, ITB AAS Indonesia

^c Teknologi / Informatika, robbyvaullah@gmail.com, ITB AAS Indonesia

* coresspondence

ABSTRACT

This research focuses on developing a rainfall prediction system using the Backpropagation Artificial Neural Network (JST) method. The system is designed to predict future rainfall intensity by utilizing four main parameters: air temperature, air humidity, wind speed, and air pressure. Accurate rainfall prediction is essential in various fields, such as agriculture, transportation, and industry. Traditional systems for predicting rainfall are often unable to provide accurate and timely information. Backpropagation JST offers a promising solution to overcome this limitation. The rainfall prediction system developed using Backpropagation JST shows satisfactory performance. The system is able to predict rainfall intensity with a high degree of accuracy. These findings suggest that the Backpropagation JST-based rainfall prediction system can be a valuable tool for various sectors that require accurate and timely rainfall information. The system has the potential to improve efficiency and effectiveness in various activities, such as irrigation planning, disaster risk management, and industrial operations. This research can be extended by exploring other JST methods to improve prediction accuracy. In addition, this research can be implemented in other regions with different climatic conditions to test the generalizability of the system.

Keywords: Accuracy, Artificial Neural Network, Information System, Prediction, Rainfall.

Abstrak

Penelitian ini berfokus pada pengembangan sistem prediksi curah hujan menggunakan metode Jaringan Syaraf Tiruan (JST) *Backpropagation*. Sistem ini dirancang untuk memprediksi intensitas curah hujan di masa depan dengan memanfaatkan empat parameter utama: suhu udara, kelembaban udara, kecepatan angin, dan tekanan udara. Prediksi curah hujan yang akurat sangat penting dalam berbagai bidang, seperti pertanian, transportasi, dan industri. Sistem tradisional untuk memprediksi curah hujan seringkali tidak dapat memberikan informasi yang akurat dan tepat waktu. JST *Backpropagation* menawarkan solusi yang menjanjikan untuk mengatasi keterbatasan ini. Sistem prediksi curah hujan yang dikembangkan menggunakan JST *Backpropagation* menunjukkan kinerja yang memuaskan. Sistem ini mampu memprediksi intensitas curah hujan dengan tingkat akurasi yang tinggi. Temuan ini menunjukkan bahwa sistem prediksi curah hujan berbasis JST *Backpropagation* dapat menjadi alat yang berharga untuk berbagai sektor yang membutuhkan informasi curah hujan yang akurat dan tepat waktu. Sistem ini berpotensi meningkatkan efisiensi dan efektivitas dalam berbagai kegiatan, seperti perencanaan irigasi, manajemen risiko bencana, dan operasi industri. Penelitian ini dapat diperluas dengan mengeksplorasi metode JST lain untuk meningkatkan akurasi prediksi. Selain itu, penelitian ini dapat diimplementasikan di wilayah lain dengan kondisi iklim yang berbeda untuk menguji generalisasi sistem.

Kata Kunci: Akurasi, Curah hujan, Prediksi, Jaringan Syaraf Tiruan, Sistem Informasi

1. PENDAHULUAN

Sekarang ini, keadaan cuaca sering berubah tak menentu, hal itu menjadi salah satu penyebab prakiraan cuaca tidak dapat diprediksi dengan benar. Para ahli menjadikan hal ini sebagai tantangan untuk dapat menciptakan sesuatu yang dapat memperkirakan cuaca yang akan terjadi beberapa hari ke depan dan dengan perkiraan yang lebih akurat. Pentingnya perubahan cuaca ini membuat banyak pihak membutuhkan informasi perkiraan cuaca yang akurat, cepat dan lengkap untuk menunjang kehidupan sehari-hari [1].

Curah hujan adalah jumlah air hujan yang terakumulasi di suatu lokasi tertentu yang diukur dalam skala milimeter atau sentimeter dan dipengaruhi oleh beberapa faktor yang mempengaruhi cuaca. Intensitas curah hujan dapat menjadi dasar untuk memperkirakan dampak curah hujan seperti banjir, tanah longsor, dan kekeringan. Namun, untuk dapat memperkirakan dampak curah hujan tersebut, diperlukan sistem yang dapat menghitung intensitas curah hujan yang mana dapat digunakan juga untuk sistem prediksi curah hujan. Sehingga, penelitian ini akan digunakan untuk pembuatan sebuah sistem prediksi curah hujan berbasis *website* [2].

Untuk mengetahui curah hujan di masa yang akan mendatang, dibutuhkan sistem yang dapat memprediksi curah hujan tersebut. Sistem akan membaca 4 *parameter* yang mempengaruhi curah hujan sebagai input data yang nantinya akan dilakukan perhitungan prediksi curah hujan, parameter yang dimaksud di antaranya yaitu suhu udara, kelembaban udara, kecepatan angin, dan tekanan udara. Perhitungan tersebut bisa disebut sebagai teknik memprediksi curah hujan secara komputasi. Penelitian ini menghasilkan sistem yang mampu memprediksi curah hujan dengan cukup akurat, sehingga permasalahan ini dapat teratasi [4].

Menurut Dian Puspita Chandra dan Alethea Suryadibrata (2019), salah satu teknik yang tersedia untuk melakukan prediksi curah hujan secara komputasi adalah dengan penerapan Jaringan Syaraf Tiruan (JST). Jaringan Syaraf Tiruan adalah algoritma yang paling sesuai dengan permasalahan yang ada karena algoritma Jaringan Syaraf Tiruan dapat mengenali pola yang ada untuk kemudian mengklasifikasi curah hujan dengan akurasi yang baik [5]. Yang digunakan dalam penelitian ini adalah JST metode *Backpropagation* yang merupakan arsitektur jaringan multi-layer yang menggunakan JST untuk memproses data pelatihan dalam arah maju dan memperbaiki kesalahan secara terbalik. Metode ini memiliki keunggulan yaitu kemampuan belajar dari contoh yang diberikan [6]. Menurut Zulhamsyah (2019), *Backpropagation* dinilai dapat menyelesaikan permasalahan yang kompleks jika dibandingkan dengan metode/algoritma yang lainnya. Penelitian ini menghasilkan sebuah sistem berbasis *website* yang dapat melakukan perhitungan JST *Backpropagation* dengan menggunakan bahasa pemrograman Python dan beberapa framework pendukung. Sehingga, permasalahan yang telah disebutkan di atas diharapkan dapat teratasi dengan adanya sistem ini.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Dalam penelitian yang berjudul Prediksi Curah Hujan dengan Empat *Parameter* menggunakan *Backpropagation* (Studi Kasus: Stasiun Meteorologi Ahmad Yani) yang ditulis oleh Herdhyanti (2019). Penelitian ini sama-sama menggunakan empat *parameter* sebagai input layer dari perhitungan *Backpropagation*, akan tetapi penelitian ini tidak menghasilkan sistem sebagai output, melainkan hanya hasil analisis perhitungan prediksinya [7].

Dalam penelitian lain, dengan judul Sistem Prakiraan Cuaca Berbasis Android dengan Metode Jaringan Syaraf Tiruan *Backpropagation* yang ditulis oleh Arifnur (2023). Penelitian ini menghasilkan *output* berupa sebuah aplikasi berbasis android yang terintegrasi dengan mikrokontroler Arduino UNO dengan bantuan *bluetooth* [8].

Terkait sistem dengan bahasa pemrograman Python, ada penelitian berjudul Implementasi Jaringan Syaraf Tiruan untuk Prediksi Cuaca pada PLTA Sumatera Barat yang ditulis oleh Lubis (2022). Penelitian tersebut menggunakan *framework* Django dari Python untuk membuat API dan React dari JavaScript untuk membuat *website* dari segi tampilan penggunaannya, *website* tersebut juga hanya menampilkan hasil prediksi curah hujan, *user* tidak dapat melakukan hal lain selain melihat hasil prediksinya [1].

2.2. Landasan Teori

2.2.1. Definisi Curah Hujan

Curah hujan merupakan ketinggian air hujan yang terkumpul dalam penakar hujan pada tempat yang datar, tidak menguap, tidak meresap dan tidak mengalir. Curah hujan juga dapat diartikan sebagai jumlah air hujan yang tertampung dalam alat pengukur hujan pada permukaan datar yang tidak menguap, atau mengalir [9].

2.2.2. Definisi Prediksi

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki. Prediksi melibatkan perkiraan peristiwa masa lalu secara sistematis dan informatif untuk memprediksi peristiwa masa depan [10].

2.2.3. Definisi Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan salah satu bagian dari AI yang memproses informasi yang didesain dengan menirukan cara kerja syaraf otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. JST dapat menimbulkan suatu pola pengetahuan melewati batas belajar atau *self-organizing*, sehingga menghasilkan nilai prediksi yang mungkin dapat berbeda dari nilai aslinya dan mampu menghasilkan pola baru yang merupakan hasil dari prediksi [3].

2.2.4. Definisi Backpropagation

Backpropagation merupakan salah satu metode dalam JST yang mengimplementasikan fungsi otak manusia ke dalam algoritma komputasi. *Backpropagation* adalah salah satu algoritma yang bekerja dengan cara meminimalkan *error* data sesuai dengan target *error* minimal yang diharapkan [3].

2.2.5. Definisi Python

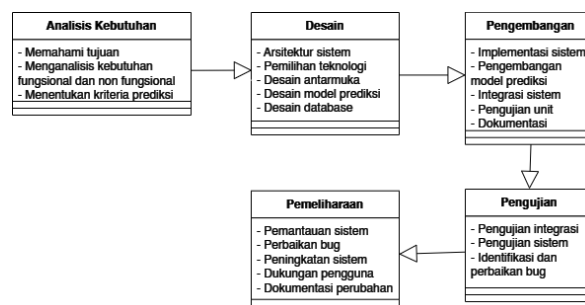
Python merupakan bahasa pemrograman yang diciptakan oleh Guido van Rossum pada tahun 1991. Tujuan utama dirancangnya bahasa ini adalah untuk memudahkan pembacaan kode oleh manusia. Ciri khas yang membuat Python berbeda adalah penggunaan indentasi yang signifikan dalam mengatur struktur kode. Indentasi yang konsisten dan akurat menggunakan spasi sangat penting dalam bahasa pemrograman ini [11].

2.2.6. Definisi Flask

Flask adalah sebuah web *framework* yang ditulis dengan bahasa pemrograman Python dan tergolong sebagai jenis *microframework*. Selain berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu website, Flask juga memiliki keunggulan yaitu dapat mengembangkan *website* yang terstruktur dan dapat mengatur *behaviour*-nya dengan lebih mudah. Meskipun termasuk dalam kategori *framework* yang ringan, Flask memiliki fungsi yang bisa dikembangkan sesuai kebutuhan [12].

3. METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian bertipe implementatif karena akan menghasilkan produk sebagai solusi terhadap permasalahan yang diangkat dalam penelitian, yaitu sebuah sistem yang di dalamnya terdapat penerapan JST *Backpropagation* guna memprediksi curah hujan. Pengembangan sistem dalam penelitian ini menggunakan model pengembangan SDLC (*Self Development Life Cycle*) metode *Waterfall*. Pengembangan model *waterfall* ini memiliki tahapan yang berurutan dari tahap analisis kebutuhan, hingga tahap pemeliharaan. Akan tetapi, Jika dalam salah satu tahap terdapat kesalahan, maka harus kembali ke tahap sebelumnya [13].



Gambar 1. Pengembangan model *waterfall*

3.1. Waktu dan Tempat Penelitian

Penelitian ini akan dilaksanakan dalam rentang waktu antara bulan Desember 2023 hingga bulan Mei 2024, dimulai dari tahap perencanaan hingga analisis hasil serta perancangan sistem. Oleh karena pengumpulan data yang dilakukan secara *online*, untuk tempat penelitian ini dilaksanakan secara daring dengan mengunduh data di beberapa situs penyedia data statistik cuaca untuk beberapa wilayah di Indonesia.

3.2. Bahan atau Materi Penelitian

Penelitian ini menggunakan data historis cuaca di beberapa wilayah di Indonesia sebagai bahan penelitian. Beberapa wilayah yang dimaksud adalah wilayah yang dilewati oleh sungai Bengawan Solo, mulai dari wilayah hulu di kabupaten Wonogiri, Jawa Tengah, hingga wilayah hilir di kabupaten Bojonegoro, Jawa Timur, yang data diambil dari situs BBWS Bengawan Solo. Guna memperluas skalabilitas sistem dan tingkat akurasi prediksi, akan digunakan juga beberapa data di wilayah lain di luar Bengawan Solo yaitu beberapa ibukota provinsi di Indonesia yang diambil dari situs data *online* BMKG. Seluruh data cuaca termasuk ke dalam populasi, sedangkan sampel yang digunakan untuk penelitian ini hanya data dalam rentang waktu dari Januari 2023 hingga Mei 2024.

3.3. Prosedur Penelitian atau Pengumpulan Data

Metode pengumpulan data dari penelitian ini adalah melalui observasi dengan mengunduh data langsung dari situs Balai Besar Wilayah Sungai (BBWS) Bengawan Solo Bengawan Solo (hidrologi.bbws-bsolo.net), situs data online BMKG (dataonline.bmkg.go.id), dan situs global penyedia data riwayat cuaca yaitu Freemeteo (freemeteo.co.id). Data yang diambil berupa data historis cuaca yang disajikan dalam satuan tertentu. Data yang sudah didapat akan diolah dan dimasukkan ke dalam *software* pengolahan data seperti Microsoft Excel, kemudian disimpan dalam ekstensi *xlsx*.

3.4. Pengolahan Data

Seluruh proses pengolahan data pada penelitian ini akan menggunakan sistem yang telah dibangun dengan beberapa spesifikasi yang telah disebutkan sebelumnya, mulai dari tahap pra-pemrosesan data hingga pemrosesan data.

3.4.1. Pra-pemrosesan Data

Sistem akan menyediakan fitur khusus untuk admin yang mana dalam fitur tersebut disediakan *form* untuk meng-upload *file* excel yang telah dibuat sebelumnya, lalu Python akan mengambil dan menyimpan *file* tersebut ke dalam sistem. Dengan menggunakan modul *pandas*, Python membaca *file* tersebut, mengolahnya menjadi tipe data *list* serta menghitung nilai minimal dan maksimal setiap variabel untuk keperluan pemrosesan data, kemudian data tersebut akan dikirim ke *database* MongoDB secara *online* dengan menggunakan modul *pymongo*. User dapat memasukkan data 4 parameter yang telah dijelaskan sebelumnya melalui *form* input yang telah disediakan atau terdapat opsi lain yaitu menggunakan data API cuaca berdasarkan lokasi pengguna sebagai input datanya. Setelah *user* mengirim data tersebut, Python akan mengambil dan mengolah data tersebut sehingga dapat digunakan untuk perhitungan *Backpropagation*.

Sebelum memasuki tahap perhitungan *Backpropagation*, Python akan mengambil data sampel yang sebelumnya di-upload ke *database*. Data tersebut dan data yang dikirim oleh *user* akan dinormalisasi dengan menggunakan fungsi tertentu di dalam Python. Normalisasi adalah proses mengubah sebuah nilai yang terlalu kecil maupun terlalu besar dalam bentuk normal, akan dilakukan penskalaan pada data sehingga data tersebut dapat diubah ke dalam skala tertentu (0-1). Normalisasi akan dilakukan menggunakan Python dengan menerapkan beberapa perhitungan yang sesuai dengan rumus normalisasi, setelah seluruh data dinormalisasi, tahap berikutnya adalah pemrosesan data dengan JST metode *Backpropagation*. Rumus perhitungan normalisasi dimuat pada persamaan berikut dengan *x* sebagai data input dari user, dihitung sesuai dengan parameter.

$$\text{Normalisasi} = \frac{x - \text{data minimal}}{\text{data maksimal} - \text{data minimal}} \quad (1)$$

3.4.2. Pemrosesan Data

Seluruh data yang telah dinormalisasi akan masuk ke perhitungan *Backpropagation* yang terbagi menjadi beberapa tahap sebagai berikut:

- 1) Menentukan bobot dan bias awal

Pada tahap ini akan dilakukan inisialisasi bobot dan bias yang akan digunakan pada perhitungan backpropagation, jumlah bobot dan bias bergantung pada jumlah neuron yang diperlukan pada input layer dan hidden layer.

2) Fase *feedforward*

Seluruh nilai bobot dan bias akan dihitung dengan nilai *input* menggunakan persamaan tertentu guna mendapatkan nilai *neuron hidden layer* dan *neuron output layer*. Nilai tersebut akan dihitung kembali dengan perhitungan fungsi aktivasi untuk kemudian hasilnya digunakan pada perhitungan tahap *backward*.

Untuk menghitung *neuron hidden layer*, dapat menggunakan persamaan 2 dengan fungsi aktivasi *sigmoid* pada persamaan 3. Untuk menghitung *neuron output layer*, dapat menggunakan persamaan 4 dengan fungsi aktivasi *sigmoid* pada persamaan 5. Hasil akhir dari perhitungan *feedforward* merupakan hasil prediksi yang belum sempurna. Jika dibandingkan dengan nilai yang sebenarnya, masih terdapat *error* yang cukup besar. Nilai tersebut akan digunakan untuk tahap perhitungan berikutnya, sebagai bahan dasar ‘pembelajaran’ oleh sistem.

3) Fase *backward*

Nilai *feedforward* yang didapatkan pada perhitungan sebelumnya akan dijadikan sebagai nilai output sementara dari perhitungan backpropagation ini. Tahap berikutnya yaitu *backward* yang diawali dengan menghitung faktor error dari neuron output yang sudah diterima sebelumnya, kemudian menghitung nilai perubahan bobot dan bias *output layer* untuk nantinya nilai tersebut digunakan pada perhitungan *weight update*. Untuk menghitung nilai tersebut, dibutuhkan satu variabel tambahan, yaitu *learning rate*. Tahap berikutnya yaitu menghitung faktor *error* pada tiap bobot *hidden* untuk kemudian nilai tersebut digunakan untuk perhitungan *error* pada tiap *neuron hidden*. Setelah didapatkan nilai *error* pada tiap *neuron hidden*, selanjutnya yaitu menghitung nilai perubahan bobot dan bias *hidden layer* untuk nantinya nilai tersebut digunakan pada perhitungan *weight update*.

4) Fase *weight update*

Setelah didapatkan nilai perubahan bobot dan bias untuk *hidden layer* dan *output layer*, nilai-nilai tersebut kemudian digunakan untuk memperbaharui bobot dan bias yang sebelumnya telah diinisialisasikan, sesuai dengan peran dari bobot dan bias masing-masing.

5) Perulangan

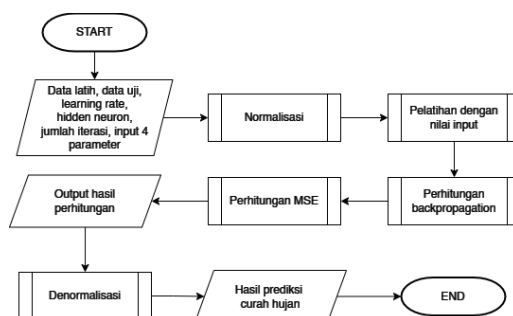
Sistem akan mengulangi perhitungan *Backpropagation* dari awal hingga menemukan nilai *error* yang paling kecil untuk hasil prediksi lebih akurat. Banyaknya pengulangan ditentukan oleh jumlah iterasi maksimum. Untuk mendapatkan nilai *error* dibutuhkan suatu perhitungan nilai *Mean Squared Error* (MSE). Agar lebih mudah dalam pemantauan perhitungan prediksi, terdapat suatu fungsi di dalam Python yang akan mencetak hasil perhitungan MSE sesuai dengan ketentuan tertentu. Dalam fungsi tersebut, terdapat penerapan logika untuk *epoch* atau iterasi maksimum yang digunakan dalam perhitungan. Jika perulangan perhitungan mencapai pada angka yang habis dibagi 10, maka akan dilakukan perhitungan MSE. Sehingga jika ada 250 *epoch*, hanya ada 25 kali perhitungan MSE dan itu tidak akan mempengaruhi hasil prediksi, karena perhitungan akan tetap dilakukan perulangan sesuai jumlah *epoch* yang ditentukan.

Setelah perulangan terhenti, akan muncul satu nilai *output* terakhir hasil dari 250 kali perhitungan MSE. Nilai tersebut akan dinormalisasi dan dijadikan sebagai hasil prediksi curah hujan. Data hasil prediksi terakhir yang semula antara angka 0 sampai 1 akan dinormalisasi menjadi nilai prediksi curah hujan yang sebenarnya. Hasil dari perhitungan denormalisasi kemudian akan ditampilkan ke sisi *front-end* dari sistemnya agar *user* dapat melihat prediksi curah hujan lengkap dengan klasifikasi yang dijelaskan dalam poin berikutnya.

3.5. Analisis dan Perancangan

Setelah dilakukan pengumpulan dan pengolahan data dengan bahasa pemrograman Python, perhitungan data tersebut akan diimplementasikan ke dalam sebuah sistem berbasis *website*. Hasil akhir dari sistem ini berupa nilai *Mean Squared Error* (MSE) dari penerapan *Backpropagation* yang kemudian diolah untuk mendapatkan *output* berupa nilai curah hujan dalam bentuk angka dengan beberapa klasifikasinya. *Parameter input* berupa suhu udara, kelembapan udara, kecepatan angin, dan tekanan udara sangat erat kaitannya dengan *parameter output* yang dalam hal ini curah hujan, *parameter input* yang disebutkan sebelumnya sangat mempengaruhi curah hujan dan juga dapat membantu meningkatkan akurasi prediksi.

Perancangan sistem pada penelitian ini dimulai dari membuat *flowchart* yang memuat alur-alur dalam memprediksi curah hujan dengan Jaringan Syaraf Tiruan metode *Backpropagation* seperti pada Gambar 2.



Gambar 2. *Flowchart* alur prediksi curah hujan

Berdasarkan *flowchart* di atas, sistem ini memungkinkan pengguna untuk mendapatkan informasi prediksi curah hujan dengan beberapa tahapan sesuai dengan penjelasan pada poin sebelumnya. Pertama, pengguna perlu memasukkan data parameter cuaca, seperti suhu udara, kelembapan udara, kecepatan angin, dan tekanan udara. Pengguna diberi opsi untuk dapat memasukkan *parameter* tersebut secara otomatis menggunakan data API cuaca berdasarkan lokasi yang nanti di-*input*-kan pengguna. Kedua, sistem akan memproses data *parameter* dan lokasi yang dimasukkan pengguna. Data *parameter* akan dinormalisasi terlebih dahulu untuk kemudian dimasukkan ke dalam model perhitungan JST *Backpropagation* untuk memprediksi curah hujan. Ketiga, hasil prediksi curah hujan dari model JST *Backpropagation* didenormalisasi untuk mendapatkan nilai curah hujan yang sebenarnya. Sistem kemudian menampilkan hasil prediksi curah hujan yang terbagi menjadi 6 klasifikasi. Klasifikasi curah hujan disajikan dalam tabel berikut [14]:

Tabel 1. Klasifikasi Curah Hujan	
Curah Hujan	Klasifikasi
0	Tidak Hujan
≤ 5	Hujan Sangat Ringan
$> 5 \ \& \ \leq 20$	Hujan Ringan
$> 20 \ \& \ \leq 50$	Hujan Sedang
$> 50 \ \& \ \leq 100$	Hujan Lebat
> 100	Hujan Sangat Lebat

Dengan menggunakan *framework* Flask yang ada pada Python, *front-end* dan *back-end* dari *website* yang dirancang akan menjadi satu kesatuan dalam satu direktori proyek. Dari sisi keamanan sistem, ada beberapa hal yang perlu ditambahkan ke dalam sistem guna mencegah terjadinya akses tidak sah terhadap sistem atau kejahatan-kejahatan melalui jalur sistem lainnya, seperti keamanan terhadap akun pengguna. Kata sandi yang digunakan oleh pengguna akan melalui proses enkripsi terlebih dahulu sebelum masuk ke dalam *database*, hal itu dilakukan guna mencegah orang lain dapat membobol kata sandi dari pengguna. Selain dari enkripsi kata sandi. Terdapat juga validasi *email* pengguna sebelum pengguna bisa masuk ke halaman *dashboard user*, akan dikirimkan sebuah pesan berisi kode OTP yang sudah terenkripsi ke email pengguna, kemudian pengguna harus memasukkan kode tersebut untuk validasi *email*.

4. HASIL DAN PEMBAHASAN

4.1. Identifikasi Masalah Penelitian

Pengembangan sistem pada penelitian ini dimulai dengan memahami tujuan dibangunnya sistem, serta menganalisis kebutuhan dari segi fungsional dan non fungsional. Setelah melakukan analisis kebutuhan, selanjutnya adalah tahap desain, tahap pengembangan dan terakhir adalah dilakukan pengujian sistem terlebih dahulu untuk menganalisis beberapa bug yang ada dalam sistem lalu dilakukan beberapa perbaikan pada tahap pemeliharaan.

4.2. Analisis Kebutuhan

Bertujuan untuk dapat membangun sistem prediksi curah hujan yang mudah diakses dan digunakan oleh khalayak umum sesuai kebutuhan mereka dengan penerapan JST *Backpropagation* dengan akurasi prediksi

yang tepat. Kebutuhan pengguna dikumpulkan melalui survei yang dibagikan ke beberapa khalayak umum secara *online* melalui Google Form. Berdasarkan hasil survei yang diisi oleh responden dengan usia rata-rata 22 tahun, telah diperoleh hasil survei yang akan dijadikan acuan dalam perancangan sistem prediksi curah hujan.

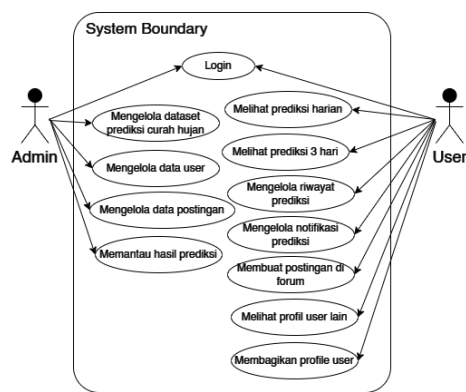
Selain analisis kebutuhan pengguna, terdapat juga analisis kebutuhan sistem. Kebutuhan sistem didefinisikan berdasarkan kebutuhan pengguna dan batasan proyek. Analisis dari segi *hardware*, *hardware* yang dibutuhkan dalam merancang sistem, yaitu laptop dengan RAM 4 GB dan processor Intel Core i3 Gen 6. Dari segi *software*, sistem ini dirancang menggunakan bahasa pemrograman Python. Sistem dirancang menggunakan laptop dengan sistem operasi Windows 10 dan menggunakan software Visual Studio Code.

4.3. Desain

4.3.1. Arsitektur Sistem

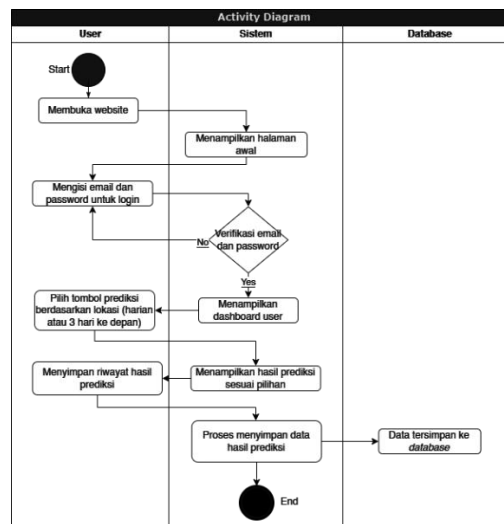
Arsitektur sistem dirancang untuk skalabilitas dan juga keamanan sistem dari berbagai segi. Perancangan arsitektur sistem ini menggunakan alat bantu UML (*Unified Modeling Language*) yang terdiri dari beberapa diagram berikut:

1) Use Case Diagram

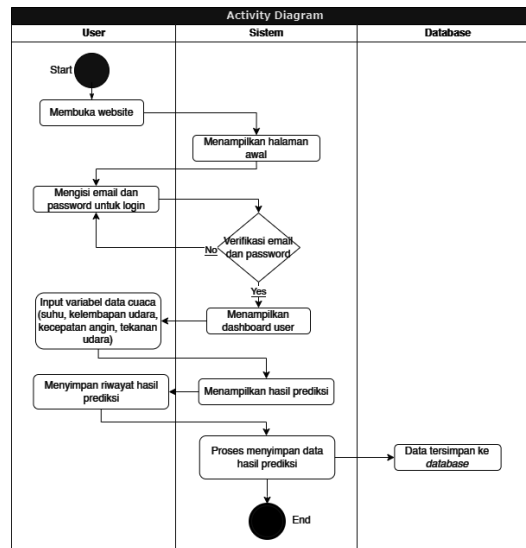


Gambar 3. Use Case Diagram user dan admin

2) Activity Diagram

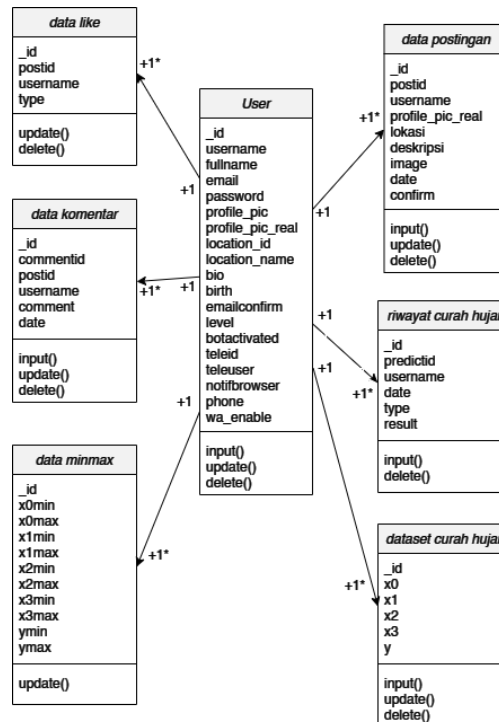


Gambar 4. Activity Diagram prediksi berdasarkan lokasi



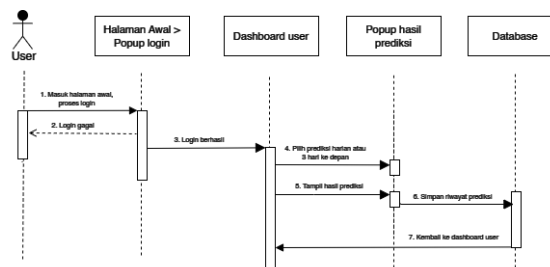
Gambar 5. Activity Diagram prediksi input manual

3) Class Diagram

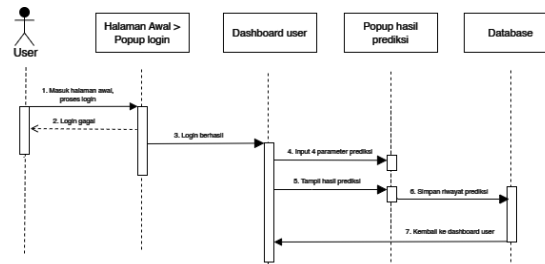


Gambar 6. Class Diagram

4) Sequence Diagram



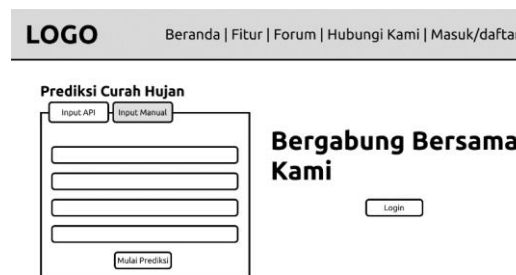
Gambar 7. Sequence Diagram prediksi berdasarkan lokasi



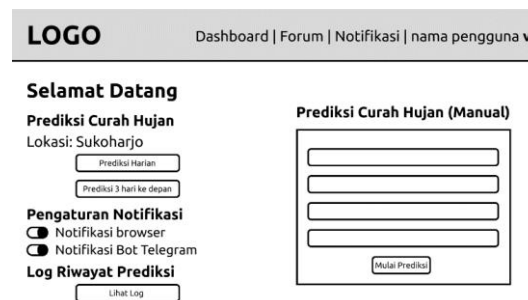
Gambar 8. Sequence Diagram prediksi input manual

4.3.2. Antarmuka Pengguna

Antarmuka pengguna dirancang untuk kemudahan dalam penggunaan dan juga estetika. Desain antarmuka pengguna akan mempermudah dalam tahap pengembangan sistem karena dapat dijadikan acuan dari segi tampilan sistemnya.



Gambar 9. Antarmuka halaman awal website



Gambar 10. Antarmuka dashboard user

4.3.3. Database

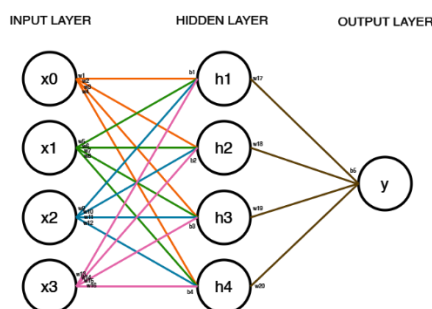
Database dirancang untuk efisiensi penyimpanan dan juga integritas data. Database dalam sistem ini akan menggunakan sebuah layanan database bernama MongoDB. Di dalam MongoDB sendiri terdapat sebuah cluster yang akan mawadahi database yang akan diperlukan dalam sebuah sistem. Di dalam database terdapat berbagai collections yang disusun dalam format JSON.

4.3.4. Model Prediksi

Model prediksi merupakan bagian penting dalam sistem prediksi curah hujan berbasis website. Dalam tahap ini, perlu merancang model *machine learning* yang dapat memprediksi curah hujan secara akurat. JST *Backpropagation* merupakan salah satu algoritma machine learning yang dapat digunakan untuk membangun model prediksi curah hujan. Data yang diperlukan untuk perhitungan prediksi berupa data historis cuaca yang terdiri dari 4 parameter input yang terdiri dari curah hujan, suhu udara, kelembapan udara, kecepatan angin, dan tekanan udara, serta parameter output berupa data curah hujan, seluruh data disajikan dalam satuan tertentu.

Proses pengolahan data diawali dengan *preprocessing* data dengan normalisasi, merubah keseluruhan data agar berada di *range* yang sama yaitu antara 0 dan 1 guna mempermudah proses perhitungan. Model JST *Backpropagation* pada penelitian ini menggunakan beberapa spesifikasi guna meningkatkan akurasi prediksi

dan mempermudah proses perhitungannya. Beberapa spesifikasi yang diperlukan mencakup, jumlah *layer* yang diperlukan, jumlah *neuron*, serta jenis fungsi aktivasi yang digunakan.



Gambar 11. Arsitektur model JST *Backpropagation*

JST *Backpropagation* menggunakan perhitungan *multilayer perceptron*. Dalam penelitian ini akan menggunakan 4 *neuron* pada *input layer*, 4 *neuron* pada *hidden layer* dengan 1 lapisan, dan 1 *neuron* pada *output layer*. Untuk melakukan perhitungan dengan jumlah *layer* dan *neuron* tersebut, dibutuhkan 16 *weights* atau bobot untuk perhitungan *hidden layer* dan 4 *weights* atau bobot untuk perhitungan *output layer*, serta 4 bias untuk perhitungan *hidden layer* dan 1 bias untuk perhitungan *output layer* agar perhitungan dapat berjalan dengan sempurna. Parameter perhitungan lain yang diperlukan berupa *learning rate*, *epochs*, *error*, dan MSE (*Mean Squared Error*) dengan masing-masing nilai ditentukan pada proses pelatihan model.

Proses pelatihan model dibagi menjadi beberapa tahap, dimulai dari inialisasi nilai *input* dan *output* untuk pelatihan model, inialisasi bobot dan bias, dan tahap perhitungan *backpropagation*. Sebagai data latih, akan digunakan angka biner 1 dan 0 sebagai nilai *input* dan *output*-nya. Sebagai bobot dan bias, akan digunakan nilai acak antara 0 dan 1. Melalui Microsoft Excel, angka acak didapatkan melalui fungsi RAND(). Sedangkan melalui Python, angka acak didapatkan melalui modul Numpy dengan fungsi np.random.normal().

Tabel 2. Inisialisasi *input* dan *output* awal dengan angka biner

X0	X1	X2	X3	Y
1	1	1	1	0
1	1	1	0	1
1	0	1	1	1
1	0	0	0	1
0	1	0	1	1
0	1	1	0	1
0	0	1	1	1
0	0	0	0	0

Tabel 3. Inisialisasi bobot dan bias

Bobot Hidden Layer			
0,8196	0,6142	0,5341	0,6309
0,1740	0,7146	0,1119	0,1273
0,1599	0,9469	0,7350	0,1016
0,6895	0,0472	0,7550	0,7698
Bobot Output Layer			
0,4015	0,0396	0,9988	0,2303
Bias Hidden Layer			
0,1386	0,8758	0,7368	0,2673
Bias Output Layer			
0,6238			

Backpropagation dihitung menggunakan beberapa persamaan dan dibagi menjadi beberapa tahapan yaitu *feedforward*, tahap *backward*, dan terakhir *weight update*. Hasil perhitungan dari salah satu sampel data biner di atas disajikan dalam data berikut:

1) Fase *feedforward*

$$h_i = (\sum w_i x_i) + b_i \quad (2)$$

$$h_1 = (w_1 x_1 + w_1 x_2 + w_1 x_3 + w_1 x_4) + b_1$$

$$h_1 = \left(\begin{matrix} 0,8196 \times 1 + 0,6142 \times 1 + \\ 0,5341 \times 1 + 0,6309 \times 1 \end{matrix} \right) + 0,1386$$

$$h_1 = 2,7372$$

$$f(h_1) = \frac{1}{1-e^{-h_1}} \quad (3)$$

$$f(h_1) = \frac{1}{1-2,71828^{-2,7372}}$$

$$f(h_1) = 0,9391$$

$$o = (\sum w_i x_i) + b_i \quad (4)$$

$$o = \left(\begin{matrix} w_{17} f(h_1) + w_{18} f(h_2) + w_{19} f(h_3) + \\ w_{20} f(h_4) \end{matrix} \right) + b_5$$

$$o = \left(\begin{matrix} 0,4015 \times 0,9391 + 0,0396 \times 0,8811 + \\ 0,9988 \times 0,8729 + 0,2303 \times 0,7650 \end{matrix} \right) + 0,6238$$

$$o = 2,0883$$

$$f(o) = \frac{1}{1-e^{-o}} \quad (5)$$

$$f(o) = \frac{1}{1-2,71828^{-2,0883}}$$

$$f(o) = 0,8893$$

2) Fase *backward*

$$e_{\text{output}} = (y - \text{output}) \times f'(o) \quad (7)$$

$$e_{\text{output}} = \left(\begin{matrix} (1 - 0,8893) \times \\ (0,8893 \times (1 - 0,8893)) \end{matrix} \right)$$

$$e_{\text{output}} = -0,0875$$

$$e_{\text{how}_i} = e_{\text{output}} \times w_{oi} \quad (10)$$

$$e_{\text{how}_{17}} = -0,0875 \times 0,4015$$

$$e_{\text{how}_{17}} = -0,0351$$

$$\Delta w_{oh_i} = \text{learnrate} \times e_{\text{output}} \times h_i \quad (8)$$

$$\Delta w_{oh_1} = 0,5 \times (-0,0875) \times 2,7372$$

$$\Delta w_{oh_1} = -0,1198$$

$$e_{h_i} = e_{\text{how}_i} \times f'(h_i) \quad (11)$$

$$e_{h_i} = -0,0351 \times (0,9391 \times (1 - 0,9391))$$

$$e_{h_i} = -0,0020$$

$$\Delta b_{ob_5} = \text{learnrate} \times e_{\text{output}} \quad (9)$$

$$\Delta b_{ob_5} = 0,5 \times (-0,0875)$$

$$\Delta b_{ob_5} = -0,0437$$

$$\Delta w_{hi} x_i = \text{learnrate} \times e_{h_i} \times x_i \quad (12)$$

$$\Delta w_{h_1 x_1} = 0,5 \times (-0,0020) \times 1$$

$$\Delta w_{h_1 x_1} = -0,0010$$

$$\Delta b_{hb_i} = \text{learnrate} \times e_{h_i} \quad (13)$$

$$\Delta b_{hb_1} = 0,5 \times (-0,020)$$

$$\Delta b_{hb_1} = -0,0010$$

3) Fase *weight update*

$$\text{new_}w_{hi} = w_{hi} + \Delta w_{hi} x_i \quad (14)$$

$$\text{new_}w_1 = 0,8196 + (-0,0010)$$

$$\text{new_}w_1 = 0,8185$$

$$\text{new_}w_{oi} = w_{oi} + \Delta w_{oh_i} \quad (16)$$

$$\text{new_}w_{17} = 0,4015 + (-0,1198)$$

$$\text{new_}w_{17} = 0,2817$$

$$\text{new_}b_{hi} = b_{hi} + \Delta b_{hb_i} \quad (15)$$

$$\text{new_}b_1 = 0,1386 + (-0,0010)$$

$$\text{new_}b_1 = 0,1375$$

$$\text{new_}b_5 = b_5 + \Delta b_{ob_5} \quad (17)$$

$$\text{new_}b_5 = 0,6238 + (-0,0437)$$

$$\text{new_}b_5 = 0,5799$$

4) Perulangan

Sistem akan mengulangi perhitungan *Backpropagation* dari awal hingga menemukan nilai *error* yang paling kecil untuk hasil prediksi lebih akurat. Untuk mendapatkan nilai *error* dibutuhkan suatu perhitungan nilai *Mean Squared Error* (MSE) menggunakan persamaan 18 dan 19 dengan n sebagai jumlah data latih.

$$\begin{aligned}
 \text{error} &= y - \text{output} \\
 \text{error} &= 0 - 0,8893 \\
 \text{error} &= -0,8893
 \end{aligned}
 \quad (18)
 \quad
 \begin{aligned}
 MSE &= \frac{\sum \text{error}^2}{n} \\
 MSE &= \frac{1,6193}{8} \\
 MSE &= 0,2024
 \end{aligned}
 \quad (19)$$

Telah didapatkan MSE untuk iterasi atau pembelajaran pertama, yaitu 0,2024. Jumlah iterasi atau *epoch* akan menentukan seberapa akurat sistem prediksi yang telah dibangun, semakin kecil nilai MSE-nya maka semakin akurat hasil prediksinya. Pada pelatihan model ini, dengan percobaan 10 *epoch*, didapatkan hasil MSE yaitu 0,1942. Selanjutnya, dilakukan perhitungan ulang dengan nilai input adalah nilai parameter cuaca yang sebenarnya yang telah didapatkan dari beberapa sumber penyedia data cuaca dan akan dihitung pada tahap evaluasi dan validasi model. Terdapat 500 data yang mana merupakan data historis cuaca yang telah dikumpulkan sebelumnya, terdiri dari data suhu udara (x0), kelembapan udara (x1), kecepatan angin (x2), tekanan udara (x3), dan curah hujan (y). Seluruh data akan dimasukkan ke dalam Microsoft Excel untuk pemodelan prediksi lebih lanjut dan ke dalam MongoDB untuk pemodelan prediksi di dalam sistem.

x0	x1	x2	x3	y
26,50	92,17	6,11	1008,00	50,20
26,50	81,74	6,11	1011,00	0,40
26,00	81,08	8,06	1012,00	0,20
27,00	88,03	5,83	1009,00	6,00
27,00	72,10	6,67	1014,00	9,60
28,00	87,87	8,89	1010,00	12,80
29,00	85,81	8,89	1010,00	4,80
28,00	77,87	7,22	1012,00	7,10
28,50	88,59	6,11	1008,00	0,00
28,50	79,19	6,67	1011,00	0,00

Gambar 12. Data historis cuaca

Setelah data tersebut dihitung dan diproses kembali menggunakan perhitungan yang sama seperti saat menggunakan angka biner, terdapat perbedaan pada hasil akhir prediksi. Pada perhitungan epoch ke 10, didapatkan hasil MSE yaitu 0,0138. Hasil tersebut lebih baik daripada perhitungan sebelumnya dikarenakan semakin kecil nilai MSE-nya maka semakin akurat perhitungan prediksinya.

4.4. Pengembangan

Tahap pengembangan sistem dimulai dari implementasi sistem yaitu pembuatan *database* untuk wadah *dataset*, dan pengembangan model prediksi. Tahap awal yang dilakukan adalah pembuatan *database* di dalam MongoDB, kemudian implementasi kode Python di dalam *software* Visual Studio Code dengan menerapkan *framework* Flask untuk dapat membangun sebuah *website*.

Pembuatan *database* melibatkan perancangan dan pembuatan struktur basis data untuk menyimpan data yang diperlukan untuk sistem. Pada tahap ini, dari *website* resmi MongoDB (cloud.mongodb.com) akan dibuat satu buah *cluster* untuk nantinya digunakan sebagai wadah dari *database* sistem ini. Di dalam *cluster* tersebut, akan dibuat *database* dengan nama *raincastdb*. *Raincastdb* akan menjadi wadah dari *collection-collection* yang dibutuhkan di dalam sistem seperti yang sudah dijelaskan sebelumnya. Terdapat 2 *database* yang akan dibuat di dalam *cluster* tersebut, yaitu *raincastdb* sebagai *database* utama dan *raincastujidb* sebagai *database* untuk pengujian perhitungan prediksi. Di dalam *database* *raincastdb* akan terdapat beberapa *collection* seperti *collection* untuk menyimpan data *user*, data postingan, atau data riwayat prediksi. Sedangkan di dalam *database* *raincastujidb* hanya terdapat *collection* untuk keperluan pengujian prediksi seperti data angka biner dan data asli untuk pengujian.

Pengembangan model prediksi diawali dengan membuat *Class* yang diperlukan, beberapa diantaranya *Class predictJST* yang digunakan untuk menempatkan model prediksi, kemudian *Class predictFunction* sebagai fungsi yang digunakan untuk memanggil model prediksi pada *Class predictJST*. Pada *Class predictJST* terdapat 2 fungsi, yaitu *JSTInitModel()* dan *JSTPredictResult()*. Pada kedua fungsi itulah model prediksi *JST Backpropagation* akan dikembangkan.

1) *JSTInitModel()*

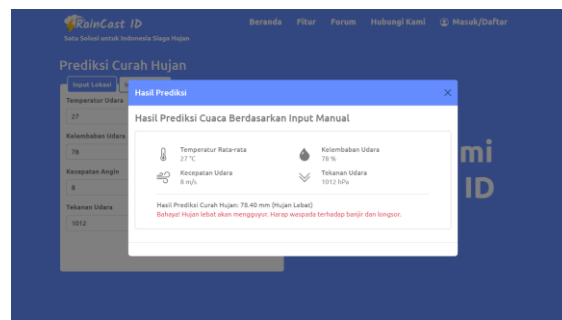
Fungsi *JSTInitModel()* digunakan untuk menginisialisasi bobot dan bias terakhir dari model JST setelah proses pembelajaran prediksi pada *epoch* yang ditentukan. Bobot dan bias akhir ini disimpan dalam database MongoDB untuk digunakan dalam perhitungan prediksi selanjutnya. Langkah awal yang dilakukan adalah menghapus koleksi data bobot dan bias yang ada di MongoDB, hal itu dilakukan guna memastikan tidak ada data lama yang tersisa sebelum menginisialisasi bobot dan bias baru. Selanjutnya yaitu inisialisasi variabel seperti *epoch* dan *learning rate* yang didapatkan dari *form input* di *client*. Setelah mendapatkan data tersebut, masuk ke langkah berikutnya yaitu mendefinisikan beberapa fungsi awal yang nanti digunakan pada perhitungan prediksi, seperti *sigmoid()* untuk fungsi aktivasi *sigmoid*, *deriv_sigmoid()* untuk turunan fungsi aktivasi, dan fungsi *mse_loss()* untuk menghitung nilai MSE. Di dalam fungsi *JSTInitModel()* dibuat *Class OurNeuralNetwork* yang di dalamnya terdapat fungsi-fungsi *Backpropagation* seperti Inisialisasi bobot dan bias yang menggunakan penerapan *np.random.normal()*, tahap *feedforward*, tahap *training* yang terdiri dari *backward*, *weight update*, dan perulangan berdasarkan *epoch* yang di-input, mencetak nilai MSE untuk memantau performa model, dan terakhir menyimpan bobot dan bias hasil *weight update* perhitungan *epoch* terakhir

Tahap selanjutnya yaitu memuat data curah hujan untuk proses pembelajaran dengan cara memanggil *Class OurNeuralNetwork* dan menyimpan hasil *output Class* tersebut di dalam variabel *network*. Dari variabel *network* tersebut akan dijalankan fungsi *train* untuk melatih model dengan *dataset* dan nilai curah hujan aktual, hasil fungsi tersebut akan menjadi nilai *output* yang ketika didenormalisasi akan mendapatkan nilai prediksi curah hujan.

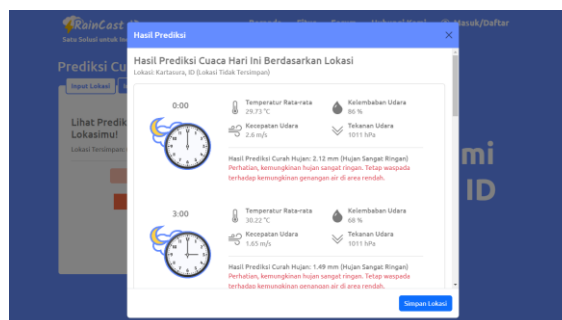
2) *JSTPredictResult()*

Fungsi ini akan memanggil bobot dan bias yang sebelumnya disimpan ke dalam MongoDB. Pemanggilan bobot dan bias masih menggunakan modul *pymongo* dengan penerapan fungsi *find()*. Sebelum dilakukan perhitungan *Backpropagation* menggunakan bobot baru, terlebih dahulu dibuat fungsi aktivasi *sigmoid* seperti pada fungsi *JSTInitModel()*.

Terdapat *Class PredictJST()* yang berisi inisialisasi bobot dan bias menggunakan data yang diambil dari *database*, dan fungsi *feedforward* untuk perhitungan prediksi secara cepat dan mengembalikan *output*. Hasil dari perhitungan *Class PredictJST* dimasukkan ke dalam variabel *result* dan dikembalikan untuk ditampilkan pada *client*. Berikut merupakan tampilan dari *client* ketika melakukan prediksi curah hujan, baik secara manual maupun berdasarkan lokasi pengguna.



Gambar 13. Tampilan website saat prediksi manual



Gambar 14. Tampilan website saat prediksi berdasarkan lokasi

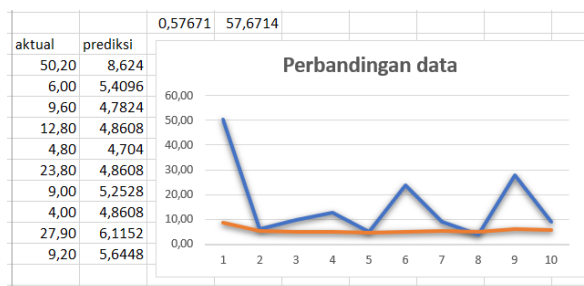
4.5. Pengujian

Pada tahapan ini dilakukan pengujian sistem untuk memastikan fungsionalitas yang sesuai dengan kebutuhan, performa dan keandalan sistem saat melakukan prediksi curah hujan dengan JST *Backpropagation*. Pengujian ini dibagi menjadi dua, yaitu pengujian fungsional dan pengujian validasi prediksi. Pengujian fungsional adalah pengujian secara fungsionalitas sistem yang telah dibangun, pengujian ini akan menggunakan metode *Black Box*.

Tabel 4. Tabel pengujian *Black Box* proses prediksi

Jenis Uji	Komponen yang diuji	Skenario Uji	Hasil yang Diharapkan	Hasil yang Dihasilkan	Status Uji	Hasil Uji
Uji Normal	Form Input Manual	Masukkan 4 parameter secara lengkap, tombol prediksi ditekan	Tampil <i>popup</i> hasil prediksi manual	Tampil <i>popup</i> hasil prediksi manual	Normal	Diterima
Uji Salah	Form Input Manual	Masukkan 4 parameter secara tidak lengkap, tombol prediksi ditekan	Tombol tidak dapat ditekan karena ada <i>form</i> yang belum diisi	Tombol tidak bisa ditekan	Normal	Diterima
Uji Normal	Prediksi berdasarkan lokasi	Menekan tombol prediksi berdasarkan lokasi	Lokasi terdeteksi dan muncul <i>popup</i> hasil prediksi otomatis	Lokasi terdeteksi dan muncul <i>popup</i> hasil prediksi otomatis	Normal	Diterima
Uji Salah	Prediksi berdasarkan lokasi	Menekan tombol prediksi berdasarkan lokasi	Lokasi tidak terdeteksi dan muncul peringatan	Lokasi tidak terdeteksi dan muncul peringatan	Normal	Diterima

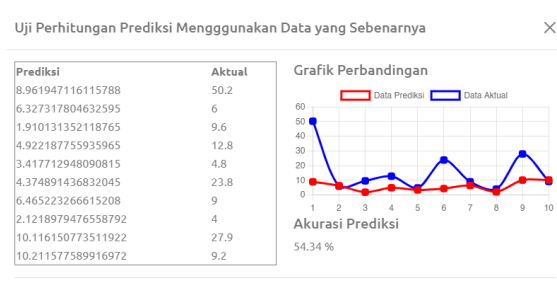
Pada pengujian validasi prediksi, untuk membandingkan hasil perhitungan manual dengan sistem, akan diberlakukan perhitungan dengan jumlah *epoch* yang sama, yaitu 20. Hal ini dimaksudkan agar dapat meminimalkan proses perhitungan manualnya. Perhitungan prediksi akan menggunakan sampel 10 data curah hujan beserta 4 *paramter input*-nya yang sebenarnya, kemudian akan dibandingkan antara hasil prediksi dengan hasil yang sebenarnya. Pada perhitungan manual dengan Microsoft Excel, didapat hasil yang tertera pada gambar 15. Rata- rata akurasi pada perhitungan manual adalah 56,67% dengan perbandingan data yang masih jauh selisihnya.



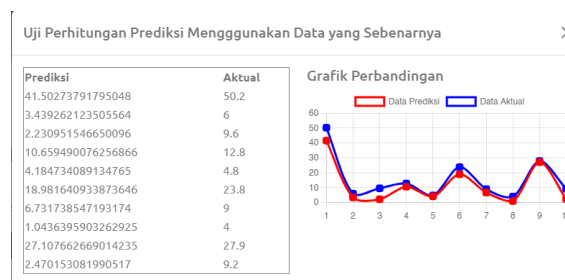
Gambar 15. Hasil perhitungan manual dengan Microsoft Excel

Perhitungan tersebut dimasukkan ke dalam sistem dengan penerapan yang sudah dibuat sebelumnya, hasil ditujukan seperti pada gambar 16. Pada hasil tersebut, ditujukan bahwa rata-rata akurasi yang dihasilkan adalah 54,34%, hanya selisih 2,33 % dari perhitungan manual dengan Microsoft Excel. Perbedaan dalam perhitungan dapat disebabkan karena hasil pada perhitungan error berbeda pada angka desimalnya. Oleh karena hasil akurasi yang ditujukan sistem ada di angka 54,34%, maka perlu dilakukan perhitungan ulang dengan menggunakan jumlah *epoch* yang berbeda. Telah dilakukan perhitungan prediksi dengan jumlah *epoch* sebanyak 100 dan 500. Hasil perhitungan ditujukan pada gambar 17 dan 18, terdapat perbedaan yang cukup signifikan antara perhitungan dengan 20 *epoch* dan perhitungan dengan *epoch* lebih dari 100. Perhitungan dengan menggunakan *epoch* 100 dapat menaikkan nilai akurasi menjadi 85,53 %, sedangkan dengan menggunakan *epoch* 500 menaikkan nilai akurasi menjadi 87,43 %. Dilihat dari grafik

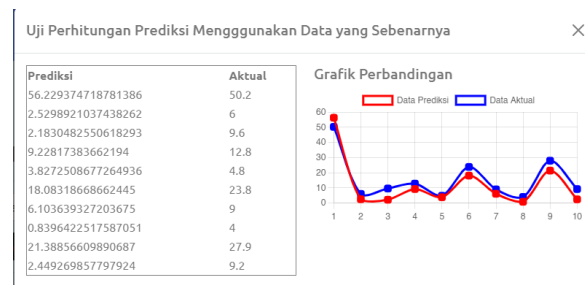
perbandingannya, perhitungan dengan menggunakan *epoch* 100 dapat menaikkan nilai akurasi menjadi 85,53 %, sedangkan dengan menggunakan *epoch* 500 menaikkan nilai akurasi menjadi 87,43 %.500 epoch menghasilkan nilai curah hujan yang lebih akurat dari 20 epoch. Dapat ditarik kesimpulan bahwa jumlah *epoch* sangat berpengaruh pada hasil prediksi curah hujan. Semakin banyak nilai *epoch* maka semakin akurat perhitungan prediksinya.



Gambar 16. Hasil perhitungan sistem dengan 20 *epoch*



Gambar 17. Hasil perhitungan sistem dengan 100 *epoch*



Gambar 18. Hasil perhitungan sistem dengan 500 *epoch*

4.6. Pemeliharaan

Tahap ini menitikberatkan pada pemantauan rutin sistem guna memastikan kinerjanya optimal dan dapat memenuhi kebutuhan pengguna. Sistem yang selesai dibangun bisa diakses secara *online* oleh khalayak umum setelah melalui proses *deploy* ke sebuah situs *deploy* bernama Vercel (vercel.com). Sistem bisa diakses melalui URL raincast-id.vercel.app. Dalam pemeliharaan sistem, ada beberapa hal yang akan dilakukan guna menjaga sistem tetap berjalan dengan normal yang disajikan dalam beberapa poin berikut:

- 1) Penyempurnaan Fungsi dan Fitur: Dilakukan berdasarkan umpan balik dari pengguna atau kebutuhan baru yang muncul.
- 2) Pemeliharaan *Database*: Termasuk pemulihan data saat terjadi kegagalan, optimasi kinerja *database*, dan pembaruan skema *database* jika diperlukan.
- 3) Pemantauan Kinerja: Dilakukan secara terus-menerus untuk mengidentifikasi masalah kinerja dan memastikan responsivitas dan efisiensi sistem tetap terjaga.
- 4) Pemantauan Sistem secara *Online*: Sistem yang di-*deploy* akan dipantau secara berkala di dalam situs Vercel itu sendiri kemudian pengembang akan mencatat apabila menemukan sebuah *bug*.

5. KESIMPULAN DAN SARAN

Berdasarkan perhitungan, implementasi, pengujian dan percobaan sistem pada penelitian ini dapat disimpulkan menjadi beberapa poin, sebagai berikut:

- 1) Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki. Curah hujan dapat

diprediksi dengan Jaringan Syaraf Tiruan metode *Backpropagation* dengan menggunakan *input layer* dengan 4 *input neuron*, 1 *hidden layer* dengan 4 *hidden neuron* dan satu *output layer* dengan 1 *output neuron*.

- 2) Perhitungan prediksi curah hujan pada penelitian ini menggunakan *dataset* yang digunakan adalah data historis cuaca di beberapa daerah di Indonesia dari awal tahun 2023 hingga Mei 2024.
- 3) Jumlah *epoch* dan *learning rate* berpengaruh pada hasil akurasi prediksi. Semakin besar nilai *learning rate* dan nilai *epoch*-nya, maka semakin akurat perhitungan prediksinya. Pada penelitian ini telah dilakukan percobaan perhitungan menggunakan *learning rate* 0,5 dengan menggunakan jumlah *epoch* yang berbeda-beda, yaitu 20, 100, dan 500. Dari nilai akurasi, perhitungan prediksi curah hujan dengan nilai *epoch* 20 menunjukkan nilai akurasi 54,34 %. Dengan menggunakan *epoch* 100 dapat menaikkan nilai akurasi menjadi 85,53 %, sedangkan dengan menggunakan *epoch* 500 menaikkan nilai akurasi menjadi 87,43 %.
- 4) Sistem yang dibangun diberi nama RainCast ID. Sistem tersebut berhasil dibangun menggunakan bahasa pemrograman Python dengan *framework Flask* sebagai *backend*-nya dan menggunakan JavaScript sebagai dinamisasi *website*-nya.

Terdapat beberapa saran untuk pengembangan sistem pada penelitian ini kedepannya agar lebih baik, yaitu:

- 1) Metode *Backpropagation* dapat digabungkan dengan metode Jaringan Syaraf Tiruan lainnya.
- 2) Sistem prediksi curah hujan berbasis *website* yang diberi nama RainCast ID sudah memberikan kemudahan bagi pengguna. Namun, perlu terus dikembangkan dengan menambah *fitur-fitur* yang dapat meningkatkan pengalaman pengguna, seperti *notifikasi* pada *email*.
- 3) Disarankan untuk sistem ini dikembangkan untuk menjadi sebuah aplikasi baik itu *desktop PC* maupun *Android*.

DAFTAR PUSTAKA

- [1] B. P. Lubis, Deddy Prayama, dan Hidra Amnur, "Implementasi Jaringan Syaraf Tiruan Untuk Prediksi Cuaca Pada Plta Sumatera Barat," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 2, hal. 36–41, 2022, doi: 10.30630/jitsi.3.2.61.
- [2] G. Z. Muflih, S. Sunardi, dan A. Yudhana, "Jaringan Saraf Tiruan Backpropagation untuk Prediksi Curah Hujan di Wilayah Kabupaten Wonosobo," *MUST J. Math. Educ. Sci. Technol.*, vol. 4, no. 1, hal. 45, 2019, doi: 10.30651/must.v4i1.2670.
- [3] F. Fitriyanti, "Aplikasi Jaringan Syaraf Tiruan Backpropagation dalam Prediksi Curah Hujan Bulanan di Kabupaten Wajo Sulawesi Selatan," *JPF (Jurnal Pendidik. Fis. Univ. Islam Negeri Alauddin Makassar)*, vol. 11, no. 1, hal. 44–55, 2023, doi: 10.24252/jpf.v11i1.33142.
- [4] J. Badriyah, A. Fariza, dan T. Harsono, "Prediksi Curah Hujan Menggunakan Long Short Term Memory," *J. Media Inform. Budidarma*, vol. 6, no. 3, hal. 1297, 2022, doi: 10.30865/mib.v6i3.4008.
- [5] A. P. Iskandar, "Efektifitas Jaringan Syaraf Tiruan Metode Backpropagation Dalam Memprediksi Potensi Banjir," *J. Technopreneursh. Inf. Syst.*, vol. 3, no. 2, hal. 50–56, 2020.
- [6] A. Zulhamsyah, S. Saifullah, dan M. R. Lubis, "Penerapan Backpropagation Dalam Memprediksi Produksi Kelapa Sawit Unit Kebun Marjandi," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, hal. 779–787, 2019, doi: 10.30865/komik.v3i1.1693.
- [7] A. Herdhyanti, L. Muflikhah, dan I. Cholissodin, "Prediksi Curah Hujan dengan Empat Parameter menggunakan Backpropagation (Studi Kasus: Stasiun Meteorologi Ahmad Yani)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 12, hal. 2548–964, 2022, [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [8] A. A. Arifnur, J. Rahmadoni, dan U. M. Wahyuni, "Sistem Prakiraan Cuaca Berbasis Android dengan Metode Jaringan Syaraf Tiruan Backpropagation," *Media Inform.*, vol. 22, no. 2, hal. 106–117, 2023, doi: 10.37595/mediainfo.v22i2.180.
- [9] Dian Puspita Chandra dan Alethea Suryadibrata, *Analisis Prediksi Curah Hujan Dan Kelembaban Udara Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation di Kabutapen Muaro Jambi*. 2019.
- [10] S. Suparji, S. Sriyanto, dan S. Lestari, "Metode Pendugaan Curah Hujan Dasarian Menggunakan Jaringan Syaraf Tiruan Dengan Algoritma Back-Propagation (Studi Kasus : Stasiun Klimatologi Pesawaran Lampung)," hal. 139–144, 2022.
- [11] S. O. Pramudya, F. W. Nabili, R. Himawan, dan A. P. Sari, "Sistem Pendeteksi Diabetes Menggunakan Algoritma Tsukamoto Pada Bahasa Pemrograman Python," vol. 3, hal. 53–57, 2023.

Implementasi Sistem Prediksi Curah Hujan dengan Penerapan Jaringan Syaraf Tiruan Berbasis Website (Syaifudin Fendi Prasetyo)

- [12] M. Prima, T. Menggunakan, dan P. Dengan, “Perancangan Sistem Informasi Penyewaan Thermoking Pada PT . Moderen Prima Transportasi Menggunakan Python Dengan Framework Flask,” vol. 1, no. 1, hal. 24–34, 2023.
- [13] I. Nurjanah, “Sistem Ujian Komprehensif Berbantu Komputer Dengan Framework Laravel,” 2022.
- [14] I. Gede *et al.*, “Klasifikasi Curah Hujan Harian Menggunakan Learning Vector Quantization,” *J. Ilmu Komput. Indones.*, vol. 7, no. 2, hal. 1–7, 2022.