BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tentang implementasi *extreme learning machine* dalam proses klasifikasi kualitas air di Danau Toba. Bab ini juga membahas tentang data yang digunakan serta proses normalisasi data. Selain itu, bab ini juga membahas tentang proses training, testing, dan validasi dari keluaran yang dihasilkan berdasarkan data yang dihasilkan dari proses normalisasi data.

# Data yang Digunakan

Data yang digunakan dalam penelitian ini memiliki format CSV (*Comma Separated Value*), di mana nilai dari setiap parameter dipisahkan oleh sebuah karakter *semicolon* (titik koma), seperti yang ditunjukkan pada gambar 3.1. Data ini didapat dari penelitian yang dilakukan oleh Rahmat *et al.* (2016) mengenai pengukuran kualitas air di Danau Toba. Pengukuran dilakukan di empat lokasi, yaitu:

* Haranggaol Horison, Kabupaten Simalungun;
* Ambarita, Kabupaten Samosir;
* Ajibata, Kabupaten Toba Samosir; dan
* Parapat, Kabupaten Simalungun.



**Gambar 3.1.** Bentuk dataset yang dihasilkan dari penelitian Rahmat *et al.* (2016)

Data yang digunakan dalam penelitian ini merekam hasil pengukuran kualitas air melalui parameter fisika dan kimia pada air. Parameter tersebut antara lain suhu air, tingkat keasaman air, tingkat oksigen terlarut (*dissolved oxygen*), tingkat potensi reduksi oksidasi (*oxidation reduction potential*), suhu udara, dan kelembaban udara.

# Analisis Sistem

Proses klasifikasi kualitas air Danau Toba berdasarkan data yang diperoleh dari Rahmat *et al.* (2016), dilakukan melalui beberapa langkah. Langkah-langkah tersebut ialah normalisasi data *training* dan data *testing*, penentuan jumlah neuron pada *hidden layer*, penentuan fungsi aktivasi, proses *training*, proses *testing*, dan proses validasi. Setiap langkah yang dilakukan akan dijelaskan dengan lebih terperinci pada bagian-bagian se-lanjutnya. Adapun arsitektur umum yang menggambarkan metode pada penelitian ini ditunjukkan pada Gambar 3.2.

  
**Gambar 3.2. Arsitektur umum**

## Preprocessing

Tahap awal dari proses klasifikasi kualitas air Danau Toba pada penelitian ini dimulai dari proses preprocessing. Tahap preprocessing, yang juga dapat disebut sebagai tahap normalisasi, adalah sebuah proses di mana data disesuaikan sehingga memenuhi batasan nilai tertentu. (Shalabi *et al.*, 2006) Hasil dari proses normalisasi adalah kumpulan data yang dapat digunakan pada tahap selanjutnya. Pada penelitian ini, data yang diterima oleh sistem akan diolah terlebih dahulu, sehingga dihasilkan set data yang dapat digunakan oleh *extreme learning machine* dengan perbandingan 60:40. Ini berarti 60 % dari data yang diproses akan digunakan sebagai data *training*, sedangkan sisa dari data yang diproses akan digunakan sebagai data *testing*.

Pada penelitian ini, *preprocessing* dilakukan melalui tiga tahap. Tahap pertama dari langkah *preprocessing* pada penelitian ini adalah melakukan penyaringan terhadap tiap baris data, sehingga hanya data dengan nilai numerik yang akan digunakan untuk proses selanjutnya. Tahap ini dilanjutkan dengan menyaring data sesuai dengan batasan normal setiap parameter, sehingga dihasilkan data yang memiliki nilai yang sesuai dengan batasan setiap parameter. Tahap terakhir dari langkah *preprocessing* adalah menghitung nilai data dengan batasan . Hasil dari proses preprocessing yang dilakukan adalah sebuah kumpulan data (dataset) yang sesuai dengan batasan yang ditentukan.

Metode normalisasi yang digunakan adalah *min-max normalization*, di mana normalisasi menghasilkan data yang memiliki batasan antara dua nilai tertentu (Patro & Sahu, 2015). Normalisasi akan dilakukan berdasarkan persamaan 3.1:

(3.1)

di mana merupakan hasil normalisasi dari data bernilai yang memiliki batasan antara .

Hasil akhir dari langkah preprocessing pada penelitian ini adalah dataset training dan dataset testing, yang telah disesuaikan pada rentang -1 hingga 1, sehingga dapat digunakan dalam proses *training*.

## Penentuan karakteristik artificial neural network

Pada tahap ini, ciri-ciri atau karakteristik dari artificial neural network yang akan digunakan pada extreme learning machine, akan ditentukan sebelum proses training dilakukan. Ciri-ciri atau karakteristik yang akan ditentukan pada penelitian ini adalah jumlah node pada *hidden layer* dan fungsi aktivasi yang digunakan pada setiap node.

### Penentuan jumlah node pada hidden layer

Menurut Heaton (2008), penentuan jumlah node pada *artificial neural network*, khususnya pada *hidden layer*, merupakan hal yang penting sebelum menjalankan *training*, karena *hidden layer* berperan penting dalam penghitungan hasil akhir dari *artificial neural network*.

Jumlah node pada *hidden layer* yang tidak optimal dapat menyebabkan permasalahan tertentu pada proses *training*. *Hidden layer* yang memiliki jumlah node terlalu sedikit dapat menyebabkan kondisi *underfitting*, di mana node yang tersedia tidak dapat bekerja secara maksimal untuk mendeteksi sinyal yang diterima dari *input layer*. Sebaliknya, jumlah node yang terlalu banyak dapat berakibat pada bertambahnya waktu yang dibutuhkan *artificial neural network* untuk memproses data. Selain itu, jumlah node yang terlalu banyak juga dapat berakibat pada kondisi overfitting, di mana jumlah informasi yang diterima tidak cukup untuk diproses dalam training karena banyaknya kapasitas pemrosesan informasi yang dimiliki jaringan. Penentuan jumlah neuron pada *hidden layer* dapat memenuhi aturan-aturan berikut:

1. Jumlah neuron pada *hidden layer* harus melebihi jumlah neuron di *input layer* dan tidak boleh melebihi jumlah neuron di *output layer*;
2. Jumlah neuron pada *hidden layer* harus setara dengan 2/3 (dua per tiga) dari jumlah neuron pada *input layer* dan *output layer*; dan
3. Jumlah neuron pada *hidden layer* tidak boleh melebihi dua kali jumlah neuron pada *input layer*.

Aturan mengenai jumlah neuron pada *hidden layer* ini dapat digunakan sebagai pertimbangan. Walaupun begitu, proses menentukan jumlah neuron pada *hidden layer* merupakan proses *trial and error*. Hal ini dilakukan agar jaringan dapat menyesuaikan diri dengan masalah yang akan diselesaikan.

Pada penelitian ini, proses training akan dilakukan dengan jumlah neuron pada hidden layer bernilai , di mana . Hal ini dilakukan untuk mengetahui jumlah neuron pada hidden layer yang sesuai untuk proses klasifikasi.

### Penentuan fungsi aktivasi

Tahap yang dilakukan setelah jumlah neuron pada *hidden layer* adalah menentukan fungsi aktivasi yang akan digunakan neuron dalam proses *training* dan proses *testing*. Fungsi aktivasi yang digunakan dalam penelitian ini adalah fungsi sigmoid, *sine*, *cosine*, dan *hardlim*.

## Proses training

Proses *training* merupakan tahap pertama yang dilakukan oleh *extreme learning machine* dalam menjalankan proses klasifikasi kualitas air. Proses *training* adalah sebuah proses di mana sebuah *artificial neural network* akan dilakukan konfigurasi untuk dapat menghasilkan keluaran yang dibutuhkan dengan memberikan sebuah kumpulan data tertentu. (Krose & van der Smagt, 1996) Hasil akhir dari proses ini adalah sebuah *artificial neural network* yang telah dilatih untuk memberikan hasil yang sesuai dengan data yang telah diberikan selama proses *training*.

Pada penelitian ini, sebuah artificial neural network dengan jumlah neuron pada hidden layer dan fungsi aktivasi neuron yang telah ditentukan pada langkah sebelumnya, akan dilatih menggunakan *extreme learning machine*. Proses *training* dilakukan melalui tiga tahap, yaitu pengacakan *input weight* dan *bias*, penghitungan *hidden layer output matrix*, dan penghitungan *output weight*.

### Pengacakan input weight dan bias

Tahap pertama yang dilakukan dari proses training pada penelitian ini adalah pemberian nilai *input weight* dan *bias*. *Input weight* dan *bias* dari *artificial neural network* pada penelitian ini diberikan secara acak. Ilustrasi dari hasil pengacakan *input weight* dan *bias* dapat dideskripsikan seperti pada Tabel 3.1.

**Tabel 3.1.** Ilustrasi hasil pengacakan *input weight* dan *bias*

### Penghitungan hidden layer output matrix

### Penghitungan output weight

## Proses testing