

## BAB 3

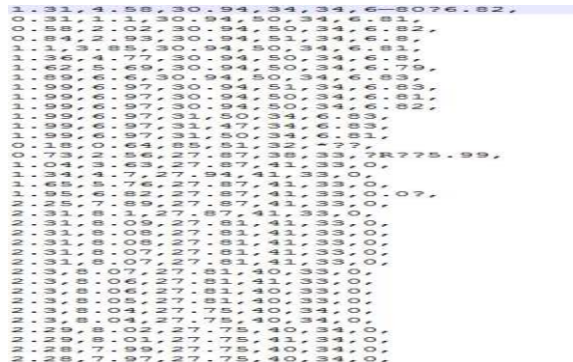
### ANALISIS DAN PERANCANGAN

Bab ini membahas tentang implementasi *extreme learning machine* dalam proses prediksi 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* dan *testing* dari keluaran yang dihasilkan berdasarkan data yang dihasilkan dari proses normalisasi data.

#### 3.1. 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.

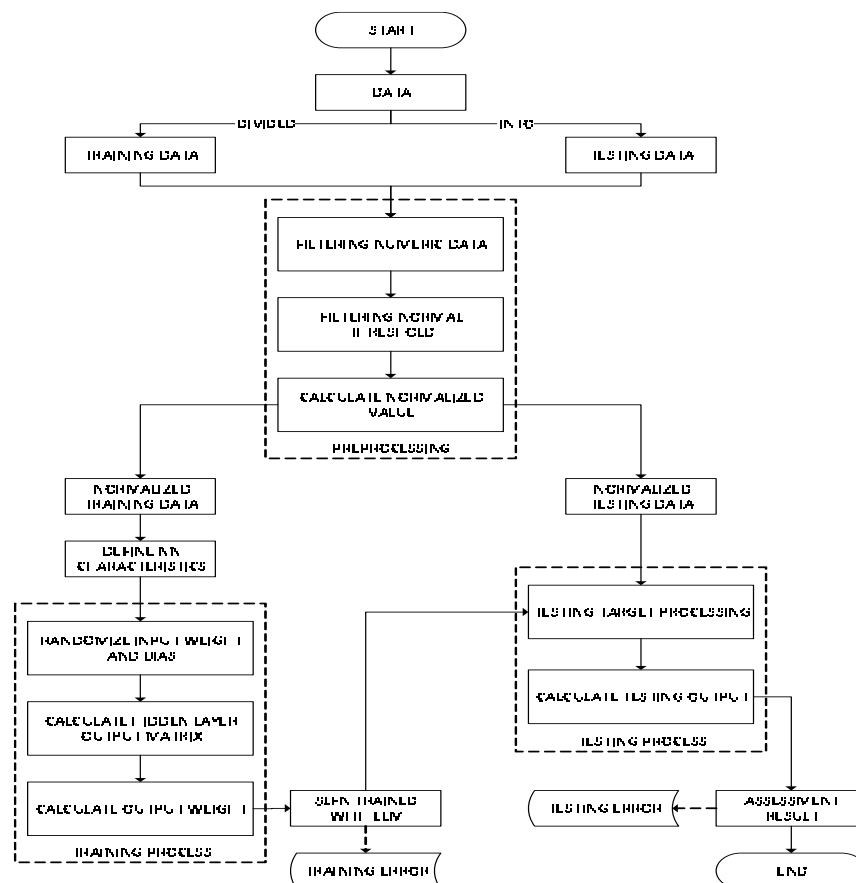


**Gambar 3.1.** Bentuk dataset yang dihasilkan dari penelitian Rahmat *et al.* (2016)

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.

### 3.2. 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 selanjutnya. Adapun arsitektur umum yang menggambarkan metode pada penelitian ini ditunjukkan pada Gambar 3.2.



Gambar 3.2. Arsitektur umum

### 3.2.1. 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  $[C, D]$ . 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:

$$A' = \left( \frac{A - A_{min}}{A_{max} - A_{min}} \right) * (D - C) + C \quad (3.1)$$

di mana  $A'$  merupakan hasil normalisasi dari data bernilai  $A$  yang memiliki batasan antara  $[C, D]$ .

Hasil akhir dari langkah *preprocessing* pada penelitian ini adalah dataset latih dan dataset uji, yang telah disesuaikan pada rentang -1 hingga 1, sehingga dapat digunakan dalam proses *training*. Rentang nilai yang digunakan dalam proses kalkulasi dijelaskan pada Tabel 3.1. Penghitungan indeks kualitas air akan dilakukan berdasarkan kriteria yang dijelaskan pada Tabel 2.1. Tidak seperti parameter dissolved oxygen dan

keasaman yang memiliki batas kewajaran tertentu, nilai normal pada ORP bersifat relatif, sesuai dengan

**Tabel 3.1.** Nilai normal, minimum, dan maksimum dari setiap parameter kualitas air (Lambrou *et al.*, 2012; Peraturan Pemerintah Nomor 82 Tahun 2001)

Parameter	Satuan	Nilai normal	Nilai minimum	Nilai maksimum
<i>Dissolved Oxygen</i>	mg/L	$\geq 6,0$	0,0	18,0
Keasaman (pH)	-	6 – 9	0,1	14,0
ORP	mV	+650 hingga +800	-2.000,00	+2.000,00
Suhu air	$^{\circ}\text{C}$	deviasi 3	6,0	100,0
Suhu udara	$^{\circ}\text{C}$	-	6,0	100,0
Kelembaban udara	%	-	0,0	100,0

Tahap *preprocessing* akan menghasilkan dua buah kumpulan data (*dataset*), yaitu kumpulan data latih (*training*) dan data uji (*testing*). Setiap baris dari *dataset* yang dihasilkan dari tahap ini merepresentasikan target keluaran berupa hasil perhitungan indeks kualitas air menurut metode Storet, dan hasil pengukuran setiap sampel yang telah mengalami proses normalisasi sesuai rentang  $[-1, 1]$ . Contoh dari beberapa baris data yang dihasilkan dari tahap *preprocessing* ditunjukkan oleh Gambar 3.3.

```

-21.000000 -1.000000 -1.000000 -0.682703 0.971775 0.058824 -0.120000
-15.000000 -1.000000 -1.000000 -0.678919 0.258814 0.058824 -0.140000
-15.000000 -1.000000 -1.000000 -0.678919 0.267604 -0.058824 -0.140000
-6.000000 -0.006667 -0.106000 -0.678919 0.274441 0.058824 -0.280000
-6.000000 0.003333 -0.097000 -0.675676 0.274441 0.058824 -0.300000
-6.000000 0.005556 -0.095000 -0.675676 0.275418 0.058824 -0.300000
-6.000000 0.010000 -0.091000 -0.678919 0.277371 0.058824 -0.280000
-6.000000 0.010000 -0.091000 -0.675676 0.277371 0.058824 -0.300000
-6.000000 0.011111 -0.090000 -0.678919 0.277371 0.058824 -0.300000
-6.000000 0.011111 -0.090000 -0.678919 0.276394 -0.176471 -0.140000
-6.000000 0.017778 -0.084000 -0.678919 0.276394 -0.058824 -0.140000
-6.000000 0.018889 -0.083000 -0.675676 0.277371 -0.058824 -0.140000
-6.000000 0.032222 -0.071000 -0.675676 0.277371 0.176471 -0.300000
-6.000000 0.025556 -0.077000 -0.678919 0.278347 -0.058824 -0.120000

```

**Gambar 3.3.** Contoh baris data pada *dataset* setelah proses normalisasi

### 3.2.2. Penentuan karakteristik artificial neural network

Pada tahap ini, ciri-ciri atau karakteristik dari *artificial neural network* yang akan digunakan pada *extreme learning machines*, 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.

### 3.2.2.1. 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:

- a. Jumlah neuron pada *hidden layer* harus melebihi jumlah neuron di *input layer* dan tidak boleh melebihi jumlah neuron di *output layer*;
- b. Jumlah neuron pada *hidden layer* harus setara dengan  $2/3$  (dua per tiga) dari jumlah neuron pada *input layer* dan *output layer*; dan
- c. 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  $n$ , di mana  $n = 1, 2, 3, \dots, 98, 99, 100$ . Hal ini dilakukan untuk mengetahui jumlah neuron pada *hidden layer* yang sesuai untuk proses prediksi.

### 3.2.2.2. 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*.

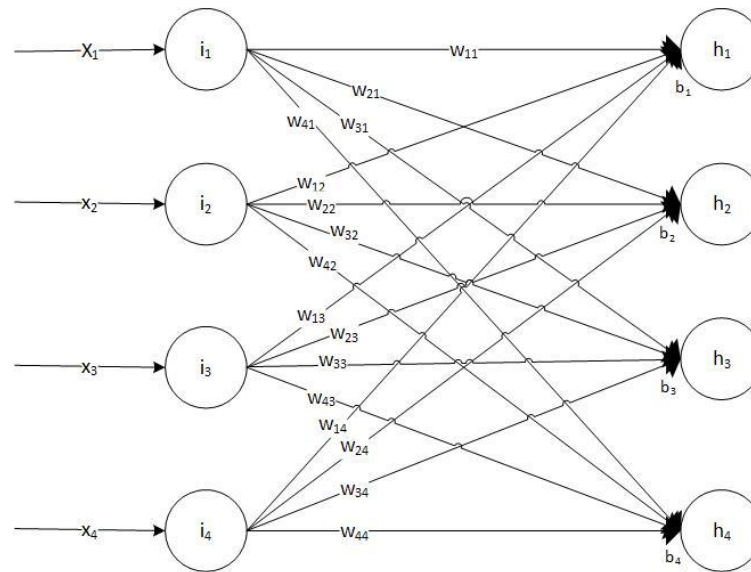
### 3.2.3. Proses training

Proses *training* merupakan tahap pertama yang dilakukan oleh *extreme learning machine* dalam menjalankan proses prediksi 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*. Proses *training* akan menghasilkan sebuah *artificial neural network* yang dapat digunakan untuk melakukan prediksi kualitas air.

#### 3.2.3.1. Pengacakan *input weight* dan *bias*

Tahap pertama yang dilakukan dari proses *training* pada penelitian ini adalah pemberian nilai *input weight* dan *bias*. Jumlah neuron pada *input layer* yang digunakan pada penelitian ini disesuaikan dengan jumlah parameter yang diterima dari *dataset* yang digunakan. *Input weight* dan *bias* dari *artificial neural network* pada penelitian ini diberikan secara acak. Ilustrasi dari proses pengacakan *input weight* dan *bias* dapat dideskripsikan seperti pada Gambar 3.4.



**Gambar 3.4.** Ilustrasi *input weight* dan *bias*

### 3.2.3.2. Penghitungan hidden layer output matrix

Setelah tahap pengacakan *input weight* dan *bias* selesai dilakukan, tahap yang akan dilakukan adalah penghitungan *hidden layer output matrix*. *Hidden layer output matrix* merupakan hasil pengolahan dari masukan yang telah diterima oleh neuron pada *hidden layer* dari neuron pada *input layer*. Pengolahan dilakukan menggunakan fungsi aktivasi yang telah ditentukan pada tahap sebelumnya. Hasil dari proses ini adalah sebuah matriks  $H$ , yang sesuai dengan persamaan 2.4.

### 3.2.3.3. Penghitungan output weight

Penghitungan *output weight* dilakukan setelah proses penghitungan *hidden layer output matrix* telah selesai dilakukan. Pada tahap ini, *output weight* akan dihitung berdasarkan persamaan 2.7. Hasil dari proses ini adalah sebuah matriks yang merepresentasikan *weight* dari setiap neuron pada *output layer*.

### 3.2.4. Proses testing

*Artificial neural network* yang telah dihasilkan pada proses *training*, akan diuji pada proses *testing*. Proses testing dilakukan untuk mengetahui efektivitas dari metode *extreme learning machine* dalam melakukan proses prediksi kualitas air. Proses *testing* dilakukan melalui dua tahap, yaitu pengolahan target keluaran dan penghitungan keluaran.

### 3.2.4.1. Pengolahan target keluaran

Dalam proses ini, sistem akan mengambil nilai dari target keluaran yang terdapat pada data uji (*testing*). Target keluaran merupakan nilai pertama yang terdapat pada tiap baris dalam *dataset*.

### 3.2.4.2. Penghitungan keluaran

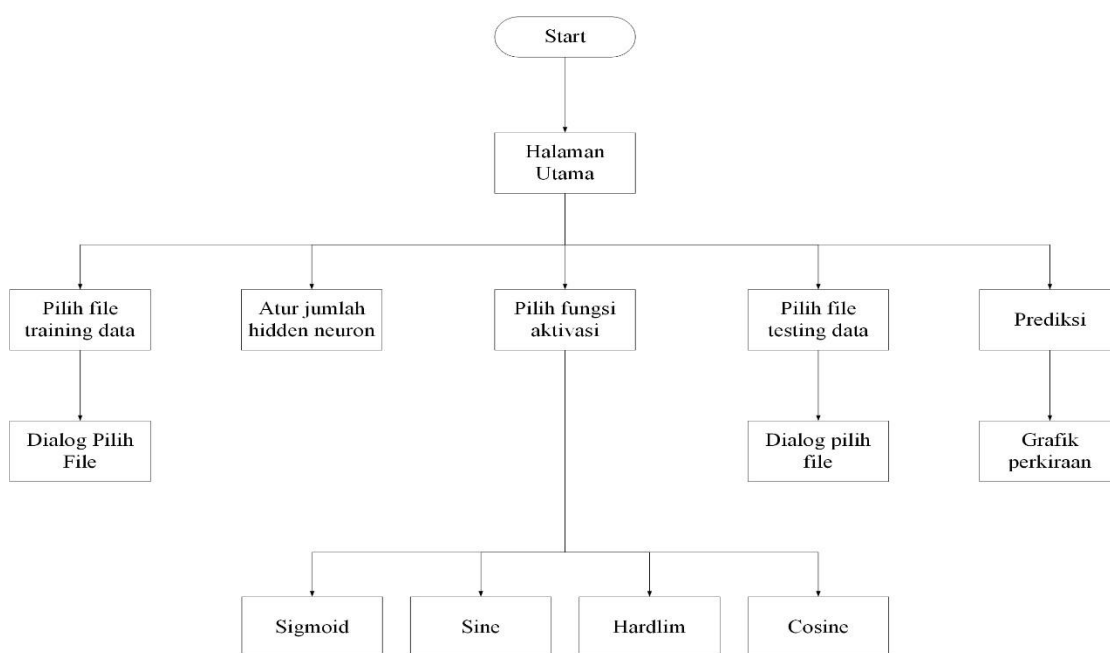
Setelah target keluaran berhasil didapatkan, tahap yang selanjutnya dilakukan adalah menghitung keluaran dari parameter yang diberikan. Proses penghitungan dilakukan menggunakan artificial neural network yang telah dilatih pada proses training sebelumnya. Hasil dari proses penghitungan akan diteruskan oleh sistem untuk ditampilkan dalam grafik yang menunjukkan pergerakan kualitas air pada setiap percobaan.

## 3.3. Perancangan Antarmuka Sistem

Tampilan antarmuka sistem yang dibuat pada penelitian ini akan digunakan untuk proses pelatihan dan pengujian prediksi kualitas air.

### 3.3.1. Perancangan menu sistem

Struktur antarmuka yang akan dibuat pada sistem ditunjukkan oleh gambar 3.5.



**Gambar 3.5.** Struktur antarmuka sistem



### 3.3.2. Rancangan tampilan halaman utama

Halaman utama, yang merupakan tampilan awal dari aplikasi yang dirancang dalam penelitian ini, berisikan judul penelitian, nama peneliti, logo institusi, dan susunan menu. Menu yang ditampilkan dalam halaman utama aplikasi terdiri atas menu untuk memilih *file* data latih (*training*), mengatur jumlah *hidden neuron*, memilih fungsi aktivasi, memilih *file* data uji (*testing*), dan memulai proses prediksi, seperti yang ditunjukkan pada gambar 3.6. Rincian dari menu yang digunakan dalam aplikasi adalah sebagai berikut:

1. Menu pilih *file* data latih (*training*), adalah menu berupa kotak teks dan sebuah tombol “Pilih” untuk memilih *file* data latih, yang digunakan untuk memilih *file* data latih (*training*) yang akan digunakan dalam proses prediksi kualitas air. Tombol “Pilih” di dalam boks pilih *file* data latih akan membuka sebuah dialog untuk memilih file. *File* yang dapat dipilih dalam dialog adalah *file* dengan ekstensi “.txt”.
2. Menu atur jumlah hidden neuron, adalah menu berupa sebuah *text box* yang tidak dapat diubah nilainya dan dua buah tombol *plus* dan *minus*, yang berfungsi untuk mengatur jumlah neuron pada *hidden layer*. Nilai *default* dari jumlah neuron pada *hidden layer* dalam aplikasi ini adalah 1, dan dapat diubah dalam rentang 1 hingga 100.
3. Menu pilih fungsi aktivasi, adalah menu berupa sebuah *combo box*, yang digunakan untuk memilih fungsi aktivasi yang digunakan dalam proses prediksi kualitas air. Fungsi aktivasi yang dapat dipilih adalah fungsi *sigmoid*, fungsi *sine*, fungsi *hardlim*, dan fungsi *cosine*.
4. Menu pilih file data uji (*testing*), adalah menu berupa kotak teks dan tombol “Pilih”, yang akan digunakan untuk memilih *file* data uji (*testing*). Tombol “Pilih” pada menu ini akan membuka sebuah jendela dialog untuk memilih *file* data uji yang akan digunakan dalam proses prediksi. *File* data uji yang dapat digunakan dalam aplikasi adalah *file* dengan ekstensi “.txt”.
5. Menu mulai proses prediksi, adalah sebuah menu berupa tombol dengan label “Prediksi” pada aplikasi, yang akan memulai proses *training* dan proses *testing* menggunakan *extreme learning machine*.

App

Prediksi Kualitas Air di Danau Toba Menggunakan Extreme Learning Machine

oleh: Eric Suwarno (121402071)

File training

Pilih file terlebih dahulu

Pilih file

File testing

Pilih file terlebih dahulu

Pilih file

Karakteristik Neural Network

Jumlah hidden neuron

1

+

+

Fungsi aktivasi

sigmoid

sine

hardlim

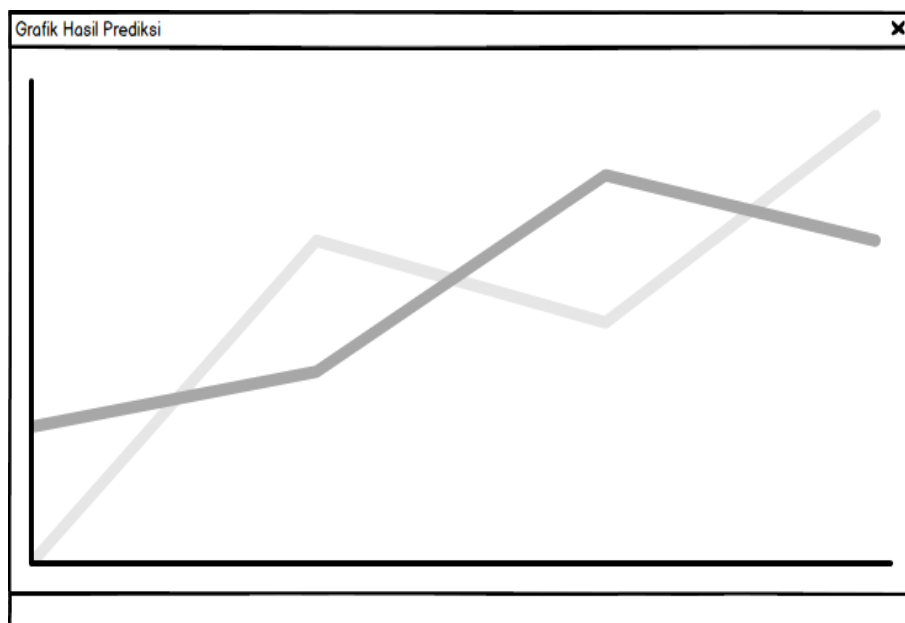
cosine

Prediksi

**Gambar 3.6.** Rancangan halaman utama aplikasi

### 3.4. Hasil Akhir

Hasil akhir yang didapatkan dari proses prediksi kualitas air pada penelitian ini adalah grafik yang menjelaskan perkiraan kualitas air untuk setiap baris data pada kumpulan data uji. Prediksi indeks kualitas air yang dihasilkan akan dibandingkan dengan indeks kualitas air yang didapat dari hasil pengukuran, untuk mengetahui tingkat akurasi dari hasil prediksi. Ilustrasi dari grafik hasil perkiraan prediksi yang akan ditampilkan oleh aplikasi ditunjukkan oleh Gambar 3.7.



**Gambar 3.7.** Ilustrasi tampilan hasil prediksi