

BAB 4 PERANCANGAN

Pada bab ini akan dibahas mengenai perancangan sistem sebagai langkah implementasi metode yang digunakan. Selain itu terdapat juga rancangan *flowchart* yang merupakan implementasi dari algoritme yang telah ditetapkan.

4.1 Identifikasi Permasalahan

Pada penelitian ini permasalahan yang harus diselesaikan ialah memprediksi beban pemanasan dan beban pendinginan. Data masukan dari pengguna nantinya akan diolah dengan menggunakan *Extreme Learning*. Sampel data yang digunakan berjumlah 130 data bangunan sebagaimana ditunjukkan pada Tabel 4.1 berikut (sampel data yang lebih lengkap akan ditunjukkan pada lampiran)

Tabel 4.1 Sampel Data

No.	RC	SA	WA	HL	CL
1	0.9	750	400	35	40
2	0.65	560	300	7	11
3	0.76	600	315	8	13
4	0.85	700	375	28	33
5	0.8	650	350	14	25
6	0.91	779	336	22	16
7	0.69	523	304	32	31
8	0.64	618	383	22	32
9	0.91	522	335	26	42
10	0.78	757	312	13	13
11	0.71	783	337	31	38
12	0.97	766	259	13	39
13	0.7	757	318	12	31
14	0.65	763	298	14	27
15	0.77	673	251	8	45
16	0.86	586	413	42	48
17	0.69	676	325	29	26
18	0.75	755	272	30	22
19	0.65	649	275	18	14
20	0.8	538	340	14	42
21	0.85	797	264	31	14
22	0.78	593	394	22	21
23	0.85	730	413	33	14
24	0.96	675	285	17	35
25	0.77	782	265	12	44

Prediksi beban pemanasan dan beban pendinginan ini menggunakan 3 fitur yaitu RC, SA dan WA, dengan RC adalah nilai *Relative Compactness*, SA adalah *Surface Area*, WA adalah *Wall Area*, HL adalah *Heating Load*, dan CL adalah *Cooling Load*. Tiga fitur tersebut menjadi jumlah *input layer* yang nantinya menjadi salah satu parameter dalam proses ELM. Sedangkan HL dan CL menjadi jumlah target yang juga merupakan parameter dalam proses ELM.

Parameter yang akan digunakan pada proses ELM meliputi: jumlah *hidden neuron*, matriks bobot W , matriks bias, jumlah *input layer*, jumlah *target*, jumlah data *training* dan data *testing*. Penentuan nilai matriks bobot W dilakukan secara *random* dengan *range* antara -1 sampai 1. Matriks bobot W mempunyai *ordo* jumlah *hidden neuron* x jumlah *input layer*. Untuk matriks bias mempunyai *ordo* 1 x jumlah *hidden neuron* dan mempunyai *range* nilai 0 sampai 1 secara *random*. Parameter-parameter tersebut digunakan untuk fase *training*, fase ini akan menghasilkan matriks bobot keluaran $\hat{\beta}$ yang digunakan pada fase *testing* untuk menghitung matriks prediksi $\hat{Y}$. Setelah matriks $\hat{Y}$ dihasilkan, maka tahap selanjutnya ialah menghitung nilai evaluasi dengan menggunakan MAPE.

4.2 Perancangan Antarmuka Pengguna

Antarmuka sistem ini terdiri dari empat halaman tab. Halaman tab pertama merupakan halaman Data untuk memasukkan dataset untuk proses ELM. Halaman tab kedua merupakan halaman untuk memasukan parameter ELM dan rasio data latih dan uji. Halaman tab ketiga untuk menampilkan hasil dari fase *training* dan fase *testing* serta hasil MAPE. Tab keempat untuk memprediksi HL dan CL berdasarkan masukan nilai RC, SA dan WA.

4.2.1 Rancangan Halaman Data

Rancangan halaman data merupakan halaman untuk memasukkan dataset yang berupa *file Microsoft Excel* (.xls / .xlsx). Selain itu terdapat *form* untuk mengisi nama *sheet* yang terdapat di *file Excel* tersebut. Pada halaman ini menampilkan tabel dataset yang telah dimuat untuk proses ELM. Rancangan halaman data ditampilkan pada Gambar 4.1.

Gambar 4.1 Halaman Data

4.2.2 Rancangan Halaman Parameter ELM

Rancangan halaman parameter merupakan halaman untuk memasukkan parameter ELM yaitu jumlah *hidden neuron* dan rasio data latih dan uji. Pada halaman ini akan menampilkan matriks bobot masukan W dan matriks bias setelah *button* Proses ELM ditekan. Rancangan halaman Parameter ELM dapat dilihat pada Gambar 4.2.

The screenshot shows a web interface with a navigation bar at the top containing four tabs: 'Data', 'Parameter ELM' (which is active), 'Hasil ELM', and 'Prediksi'. Below the tabs is a large container with the following elements:

- 'Jumlah hidden neuron' with a text input field.
- 'Rasio data' with a text input field.
- A grey button labeled 'Proses ELM'.
- 'Bobot Input' with a large rectangular text area.
- 'Bias' with a large rectangular text area.

Gambar 4.2 Halaman Parameter ELM

4.2.3 Rancangan Halaman Hasil ELM

Rancangan halaman hasil ELM adalah halaman yang menampilkan hasil fase *training*, hasil fase *testing* serta nilai MAPE. Pada hasil fase *training* akan ditampilkan matriks bobot keluaran $\hat{\beta}$. Sedangkan pada hasil fase *testing* akan ditampilkan hasil prediksi HL dan CL dari data *testing*. Untuk hasil MAPE akan ditampilkan rata-rata dari MAPE HL dan MAPE CL. Rancangan halaman Hasil ELM dapat dilihat pada Gambar 4.3.

The screenshot shows a web interface with the same navigation bar as Gambar 4.2, with 'Hasil ELM' now active. The main content area is structured as follows:

- A tab labeled 'Hasil Training' is active, showing a large text area labeled 'Bobot Keluaran'.
- Below it is a tab labeled 'Hasil Testing', which contains a large text area labeled 'Hasil Prediksi'.
- At the bottom right of the main container, there is a label 'MAPE' next to a text input field.

Gambar 4.3 Halaman Hasil ELM

4.2.4 Rancangan Halaman Prediksi

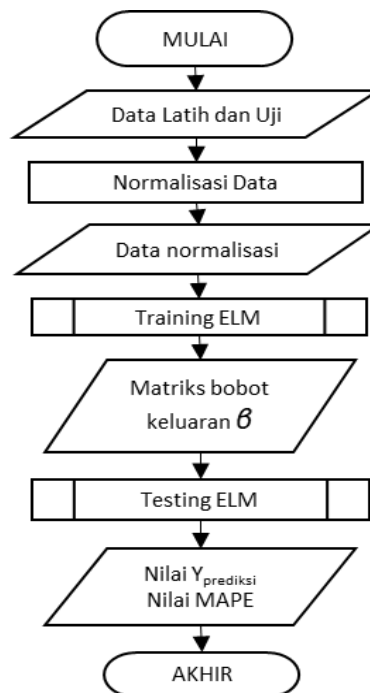
Rancangan halaman prediksi adalah halaman untuk memprediksi HL dan CL berdasarkan nilai RC, SA dan WA. Halaman ini menampilkan *form* untuk memasukan nilai RC, SA dan WA serta hasil prediksi nilai HL dan CL. Rancangan Halaman Prediksi dapat dilihat pada Gambar 4.4.

The image shows a web interface for the prediction page. It has a header with four tabs: 'Data', 'Parameter ELM', 'Hasil ELM', and 'Prediksi'. The 'Prediksi' tab is active. Below the tabs, there are three input fields labeled 'Relative Compactness', 'Surface Area', and 'Wall Area'. To the right of these fields is a button labeled 'Prediksi'. Below the input fields is a large rectangular box labeled 'Hasil Prediksi'.

Gambar 4.4 Halaman Prediksi

4.3 Perancangan Algoritme

Diagram alur algoritme *Extreme Learning Machine* merupakan tahapan dalam menyelesaikan masalah secara bertahap untuk mendapatkan hasil prediksi yang akan dilakukan pada penelitian. Diagram alir algoritme ELM dapat dilihat pada Gambar 4.5.



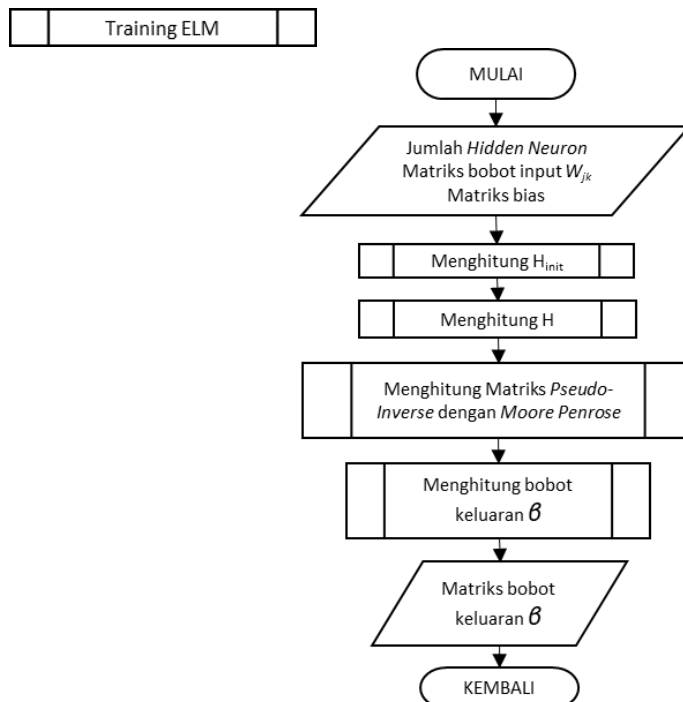
Gambar 4.5 Diagram Alir Algoritme ELM

Langkah-langkah proses prediksi beban pemanasan dan beban pendinginan bangunan menggunakan algoritme ELM berdasarkan Gambar 4.5 adalah sebagai berikut:

1. Sistem menerima masukan berupa dataset yang nantinya akan dibagi sebagai data latih dan data uji untuk proses ELM.
2. Dataset akan memasuki proses normalisasi pada range 0 sampai -1.
3. Proses normalisasi akan menghasilkan data yang telah dinormalisasi sebagai masukan untuk proses selanjutnya.
4. Melakukan proses *training* dari data yang telah dinormalisasi. Pada Gambar 4.6 akan ditunjukkan lebih jelas diagram alir proses *training*.
5. Melakukan proses *testing* menggunakan bobot keluaran $\hat{\beta}$ untuk menghasilkan prediksi. Diagram alir yang lebih jelas untuk proses *testing* akan ditunjukkan pada gambar 4.15.
6. Proses *testing* akan menghasilkan matriks prediksi $\hat{Y}$ dan nilai evaluasi MAPE. Dengan dihasilkannya matriks $\hat{Y}$ dan nilai MAPE menunjukkan bahwa berakhirnya proses ELM.

4.3.1 Tahap Training

Pada tahapan ini dilakukan proses *training* untuk memperoleh bobot keluaran yang didapatkan melalui beberapa langkah perhitungan. Pada Gambar 4.5 akan ditunjukkan diagram alir proses *training* untuk mendapatkan bobot keluaran.



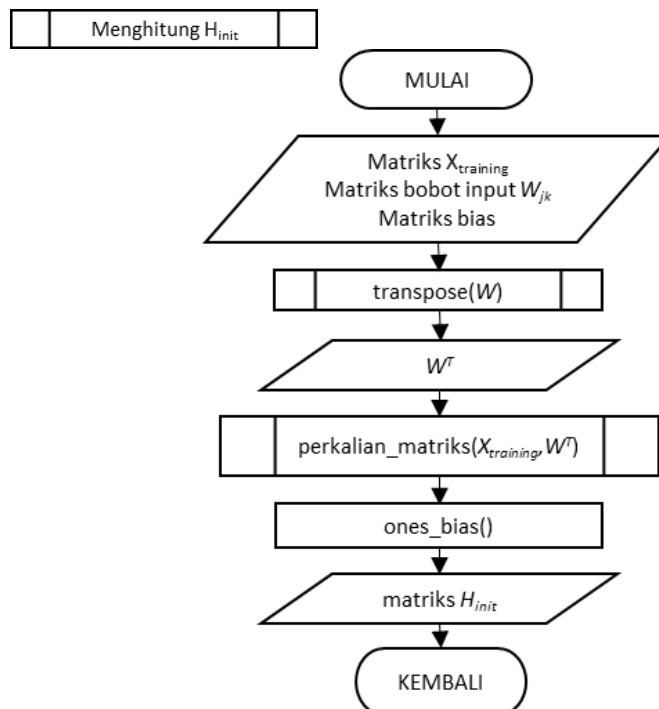
Gambar 4.6 Diagram Alir Proses *Training*

Berikut ini merupakan langkah-langkah proses *training* pada metode ELM berdasarkan gambar 4.5.

1. Sistem menerima parameter berupa jumlah hidden neuron, matriks bobot masukan W dan matriks bias.
2. Menghitung matriks H_{init} sebagai proses awal sebelum melakukan perhitungan matriks keluaran *hidden layer*. Proses perhitungan H_{init} yang lebih jelas ditunjukkan pada Gambar 4.7.
3. Menghitung matriks H menggunakan rumus aktivasi *sigmoid biner*.
4. Menghitung matriks *Moore-Penrose pseudo invers*. Pada Gambar 4.12 akan ditunjukkan lebih jelas diagram alir untuk menghitung matriks *Moore-Penrose Pseudo Inverse*.
5. Menghitung matriks bobot keluaran $\hat{\beta}$ yang nantinya akan digunakan pada proses testing untuk menghitung hasil prediksi $\hat{Y}$. Diagram alir untuk menghitung bobot keluaran akan ditunjukkan pada Gambar 4.14

4.3.1.1 Menghitung H_{init}

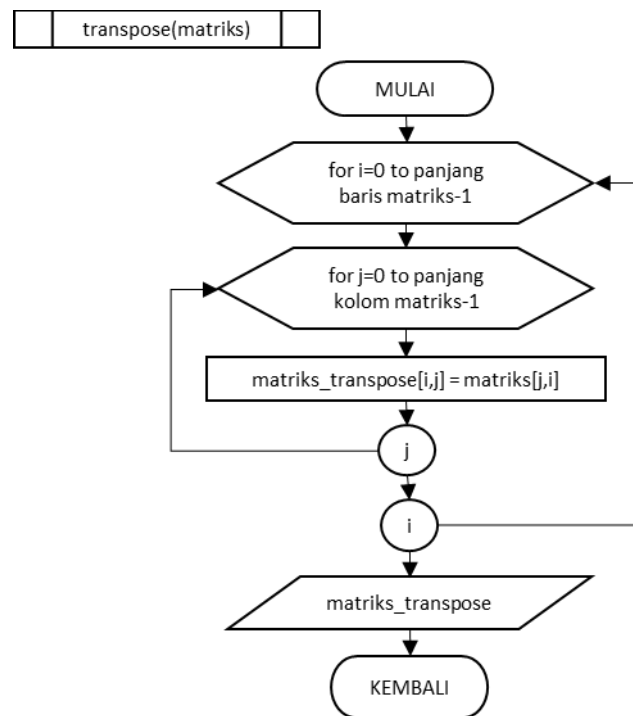
Pada tahap ini dilakukan perhitungan untuk menghasilkan matriks H_{init} dengan cara melakukan perkalian matriks $X_{training}$ dengan matriks bobot masukan W^T dan menambahkan dengan matriks bias. Diagram alir perhitungan H_{init} dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir Perhitungan H_{init}

Langkah-langkah proses perhitungan matriks H_{init} berdasarkan Gambar 4.7 adalah sebagai berikut:

1. Sistem membutuhkan matriks $X_{training}$, matriks bobot masukan W dan matriks bias sebagai parameter masukan untuk melakukan proses perhitungan H_{init} .
2. Melakukan proses *transpose* matriks untuk bobot masukan W sehingga menghasilkan matriks W^T dengan menggunakan fungsi *transpose* yang akan ditunjukkan pada Gambar 4.8
3. Melakukan perkalian matriks $X_{training}$ dengan W^T dengan menggunakan fungsi perkalian matriks yang akan ditunjukkan pada Gambar 4.9.
4. Hasil perkalian matriks $X_{training}$ dengan W^T akan ditambahkan dengan matriks bias sehingga menghasilkan matriks H_{init} .

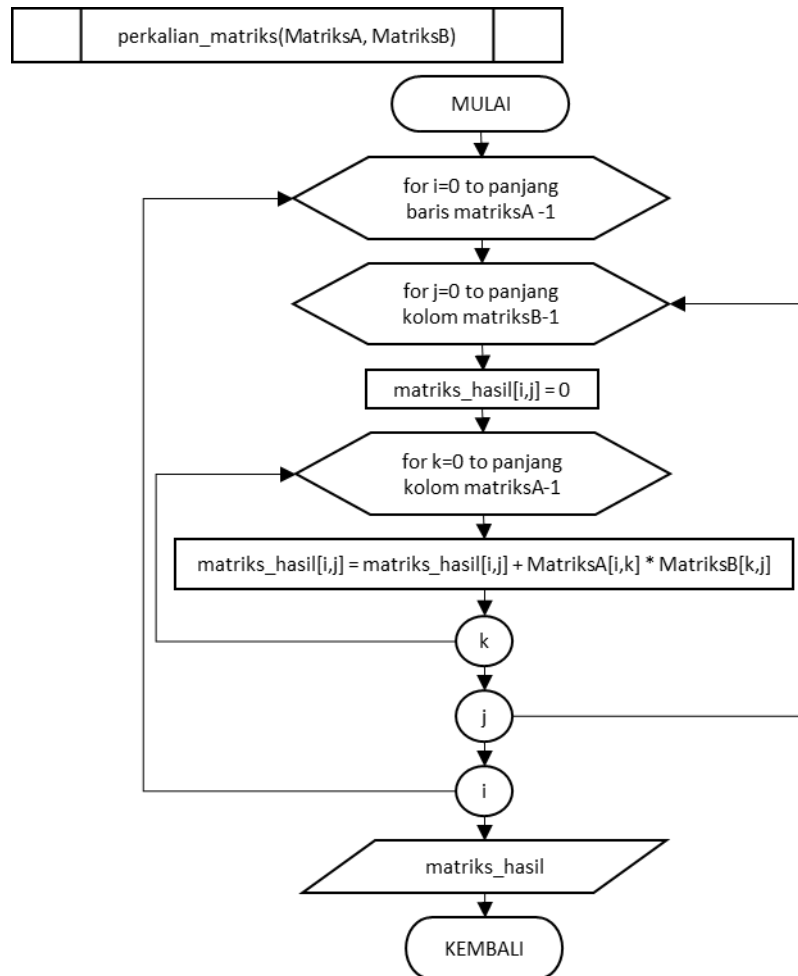


Gambar 4.8 Diagram Alir Transpose Matriks

Langkah-langkah untuk melakukan *transpose* matriks berdasarkan pada Gambar 4.8 adalah sebagai berikut:

1. Sistem menerima masukan berupa matriks yang akan *ditranspose*.
2. Membuat matriks transpose yang memiliki ordo kolom matriks masukan x baris matriks masukan.
3. Melakukan transpose matriks dengan cara mengubah baris dan kolom matriks masukan menjadi kolom dan baris baru yang disimpan pada matriks transpose.

4. Fungsi transpose ini akan mengembalikan (*return*) nilai matriks transpose.



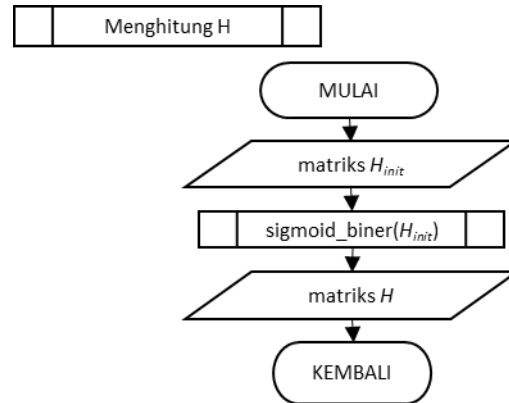
Gambar 4.9 Diagram alir perkalian matriks

Langkah-langkah untuk melakukan perkalian matriks berdasarkan Gambar 4.9 adalah sebagai berikut:

1. Sistem menerima masukan berupa 2 matriks yang akan dikalikan yaitu matriksA dan matriksB.
2. Membuat matriks hasil dengan panjang baris sama dengan panjang baris matriksA dan panjang kolom sama dengan panjang kolom matriksB.
3. Melakukan perkalian matriks dengan cara mengalikan baris matriksA dengan kolom matriksB sebanyak jumlah baris matriksA.
4. Fungsi perkalian matriks ini akan mengembalikan (*return*) nilai matriks hasil.

4.3.1.2 Menghitung H

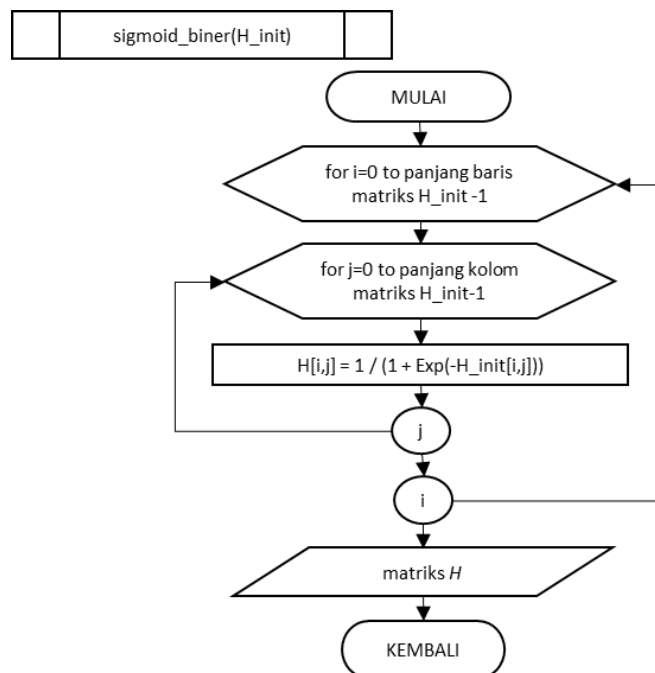
Pada tahap ini akan dilakukan perhitungan untuk menghitung matriks keluaran hidden layer yakni matriks H menggunakan fungsi aktivasi *sigmoid biner*. Diagram alir untuk perhitungan matriks H dapat dilihat pada Gambar 4.10.



Gambar 4.10 Diagram alir perhitungan matriks H

Langkah-langkah untuk melakukan perhitungan matriks keluaran *hidden layer* H berdasarkan Gambar 4.10 adalah sebagai berikut:

1. Sistem menerima masukan matriks H_{init} yang telah dihasilkan dari perhitungan matriks H_{init} .
2. Melakukan perhitungan dengan menggunakan fungsi sigmoid biner yang ditunjukkan lebih jelas pada gambar 4.11.
3. Fungsi ini akan mengembalikan (*return*) nilai matriks H



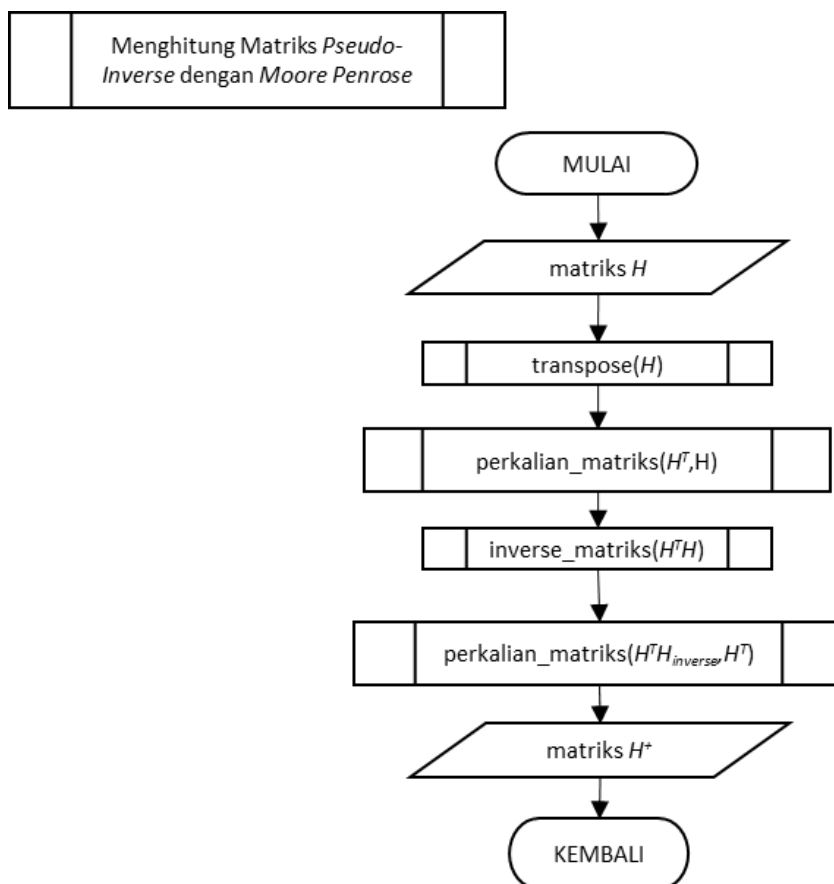
Gambar 4.11 Diagram alir fungsi sigmoid biner

Langkah-langkah untuk melakukan perhitungan *sigmoid biner* berdasarkan Gambar 4.11 adalah sebagai berikut:

1. Sistem menerima masukan matriks H_{init} yang telah dihasilkan dari perhitungan matriks H_{init} .
2. Melakukan perhitungan dengan menggunakan rumus *sigmoid biner* masing-masing nilai dari matriks H_{init} .
3. Fungsi ini akan mengembalikan (*return*) nilai matriks H .

4.3.1.3 Menghitung Matriks H^+ dengan Moore-Penrose

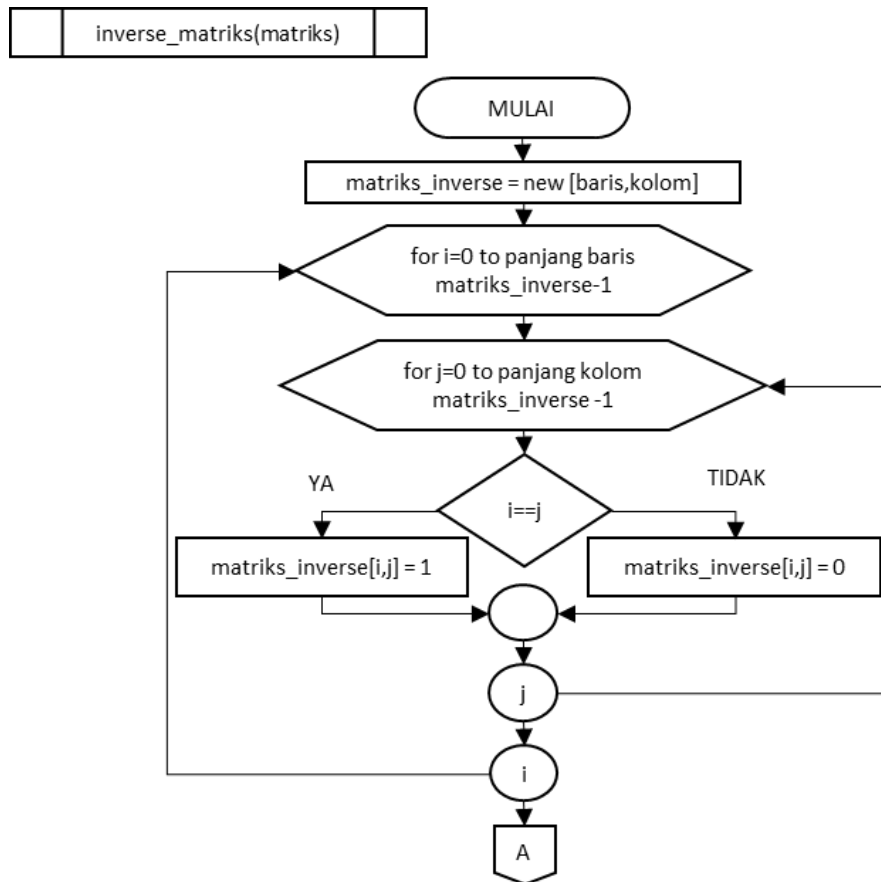
Pada tahap ini akan dilakukan perhitungan untuk menghitung matriks H^+ dengan *Moore-Penrose Pseudo Inverse* sebagai langkah setelah proses perhitungan matriks keluaran *hidden layer* H telah dilakukan. Proses ini akan menghasilkan matriks H^+ yang akan digunakan untuk proses selanjutnya yakni proses perhitungan bobot keluaran. Diagram alir proses perhitungan matriks H^+ dengan *Moore-Penrose Pseudo Inverse* dapat dilihat pada Gambar 4.12.

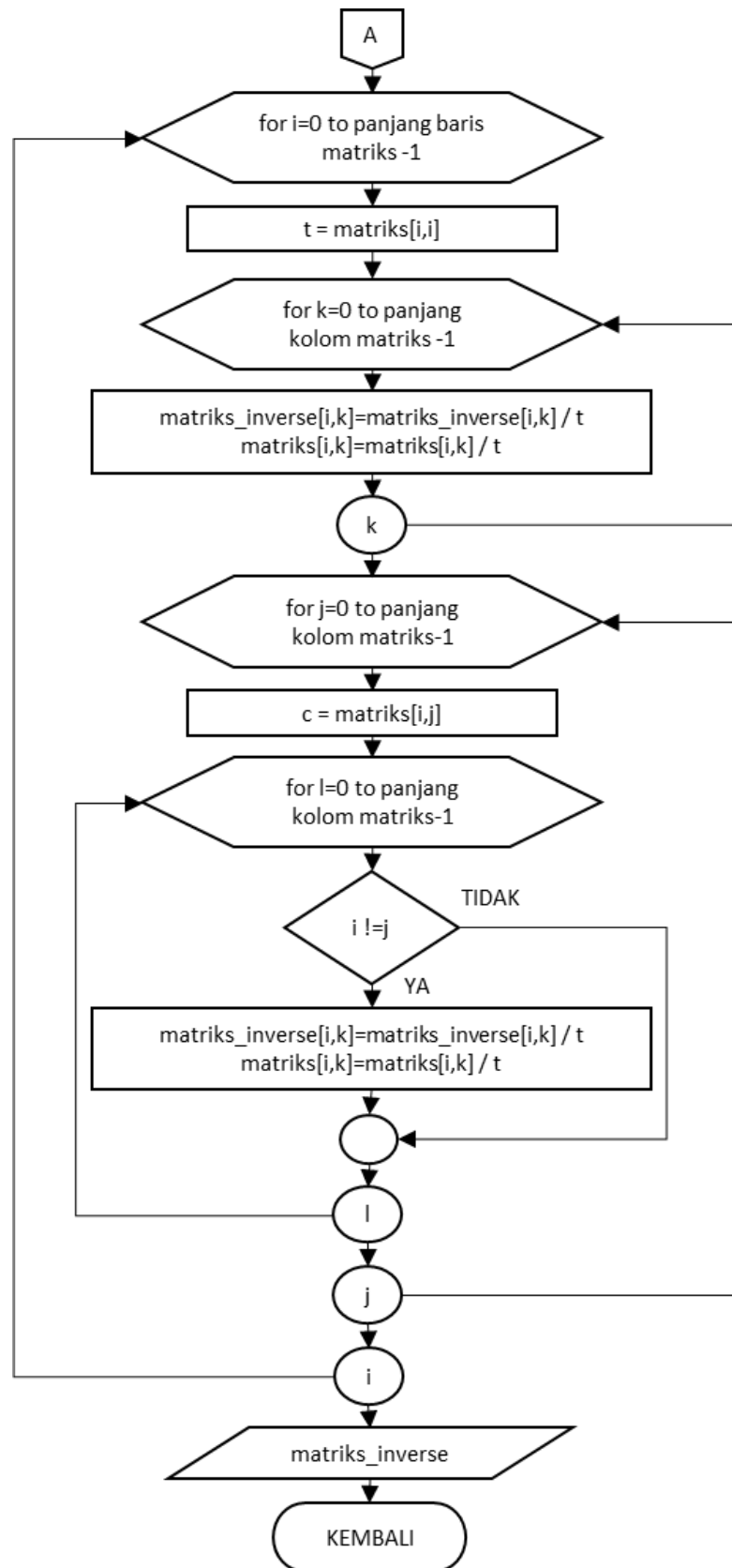


Gambar 4.12 Diagram Alir Perhitungan Matriks H^+ Moore-Penrose

Langkah-langkah untuk melakukan perhitungan matriks *Moore-Penrose Pseudo Invers* berdasarkan Gambar 4.12 adalah sebagai berikut:

1. Fungsi ini membutuhkan matriks keluaran *hidden layer* yaitu matriks H yang telah diperoleh dari tahapan sebelumnya.
2. Melakukan proses *transpose* matriks H dengan memanggil fungsi *transpose* sehingga menghasilkan matriks H^T . Proses transpose matriks dapat dilihat pada Gambar 4.8 yang telah dijelaskan sebelumnya.
3. Melakukan perkalian matriks antara matriks H^T dengan matriks H dengan memanggil fungsi perkalian matriks seperti pada Gambar 4.9. Proses ini menghasilkan matriks H^TH .
4. Melakukan invers matriks H^TH dari hasil perkalian matriks. Diagram alir untuk melakukan inverse matriks ditunjukkan pada Gambar 4.13. Proses ini menghasilkan matriks $H^TH_{inverse}$.
5. Melakukan perkalian matriks antara matriks $H^TH_{inverse}$ dengan matriks H^T dengan memanggil fungsi perkalian matriks seperti pada Gambar 4.9.
6. Fungsi perhitungan ini mengembalikan (*return*) nilai matriks H^+ .





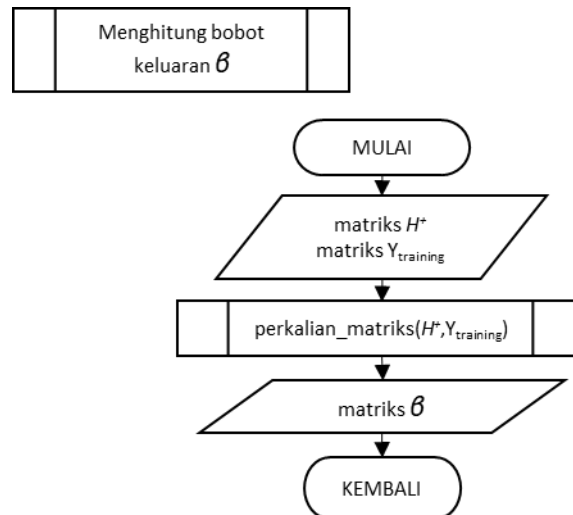
Gambar 4.13 Diagram Alir Invers Matriks

Langkah-langkah untuk melakukan invers matriks berdasarkan Gambar 4.13 adalah sebagai berikut:

1. Sistem menerima masukan berupa matriks yang akan diinvers.
2. Membuat matriks baru berupa matriks identitas dengan panjang baris dan kolom sesuai dengan matriks input.
3. Melakukan invers matriks menggunakan Operasi Baris Elementer (OBE) dengan mengubah matriks masukan menjadi matriks identitas sedangkan matriks identitas menjadi matriks baru sebagai hasil dari proses invers.
4. Fungsi invers matriks ini mengembalikan (*return*) nilai matriks invers.

4.3.1.4 Menghitung Bobot Keluaran $\hat{\beta}$

Pada tahap ini akan dilakukan perhitungan untuk menghitung bobot keluaran $\hat{\beta}$ dengan cara melakukan perkalian matriks H^+ dengan matriks $Y_{training}$. Hasil perkalian kedua matriks tersebut menghasilkan matriks bobot keluaran $\hat{\beta}$ yang akan digunakan pada tahap testing. Diagram alir perhitungan bobot keluaran $\hat{\beta}$ akan ditunjukkan pada Gambar 4.14.



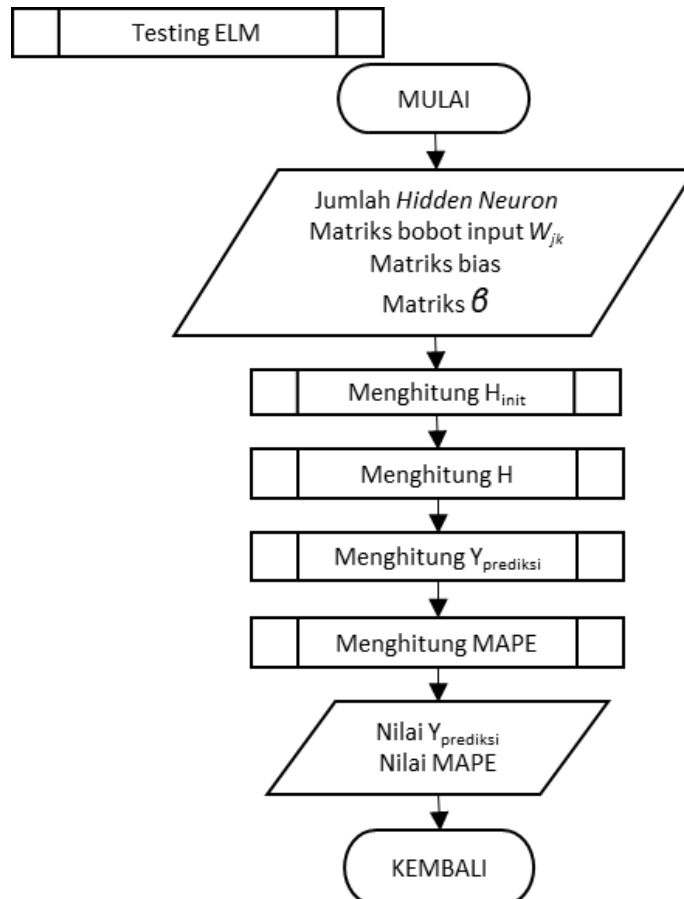
Gambar 4.14 Diagram Alir Perhitungan Bobot Keluaran $\hat{\beta}$

Langkah-langkah untuk melakukan perhitungan bobot keluaran $\hat{\beta}$ berdasarkan Gambar 4.14 adalah sebagai berikut:

1. Sistem menerima masukan berupa matriks H^+ Moore-Penrose Pseudo Invers dan target *training* $Y_{training}$.
2. Melakukan perkalian matriks untuk matriks H^+ Moore-Penrose Pseudo Invers dan $Y_{training}$ untuk menghasilkan bobot keluaran $\hat{\beta}$ dengan menggunakan fungsi perkalian seperti yang telah dijelaskan pada Gambar 4.9.
3. Fungsi ini akan mengembalikan (*return*) nilai matriks $\hat{\beta}$.

4.3.2 Tahap Testing

Proses *testing* dilakukan setelah proses *training* telah dilakukan dan menghasilkan bobot keluaran $\hat{\beta}$. Tahap ini masih membutuhkan parameter bobot masukan W dan matriks bias yang digunakan pada proses *training*. Diagram alir proses *testing* ditunjukkan pada Gambar 4.15.



Gambar 4.15 Diagram Alir Proses Testing

Langkah - langkah untuk melakukan proses *testing* berdasarkan Gambar 4.15 adalah sebagai berikut:

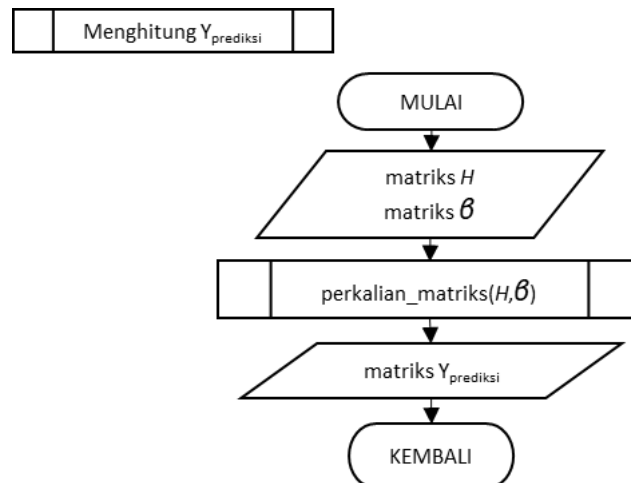
1. Sistem menerima masukan berupa data *testing* $X_{testing}$, matriks bobot masukan W , matriks bias, dan bobot keluaran $\hat{\beta}$.
2. Melakukan perhitungan matriks H_{init} sama seperti perhitungan H_{init} yang ada di proses *training* pada Gambar 4.7 hanya saja menggunakan data *testing* $X_{testing}$.
3. Melakukan perhitungan matriks H sama seperti perhitungan matriks H yang ada di proses *training* pada Gambar 4.10 hanya saja menggunakan matriks H_{init} hasil dari proses perhitungan matriks H_{init} proses *testing*.
4. Melakukan perhitungan matriks prediksi $\hat{Y}$ dengan cara melakukan perkalian matriks antara matriks H dengan matriks $\hat{\beta}$. Diagram alir

perhitungan matriks prediksi akan ditunjukkan lebih jelas pada Gambar 4.16.

5. Melakukan perhitungan nilai evaluasi MAPE dengan membandingkan matriks prediksi $\hat{Y}$ dengan matriks target testing $Y_{testing}$. Diagram alir yang menjelaskan perhitungan nilai MAPE akan ditunjukkan pada Gambar 4.17.
6. Proses ini akan menghasilkan nilai matriks Y prediksi atau $\hat{Y}$ dan nilai MAPE.

4.3.2.1 Perhitungan Y prediksi $\hat{Y}$

Tahap ini dilakukan setelah proses perhitungan matriks keluaran *hidden layer* H telah dilakukan sehingga menghasilkan matriks H . Pada tahap ini membutuhkan matriks H dan matriks $\hat{\beta}$ yang telah diperoleh dari proses training. Kedua matriks tersebut dikalikan untuk menghasilkan matriks $\hat{Y}$. Diagram alir perhitungan $\hat{Y}$ akan ditunjukkan pada Gambar 4.16.



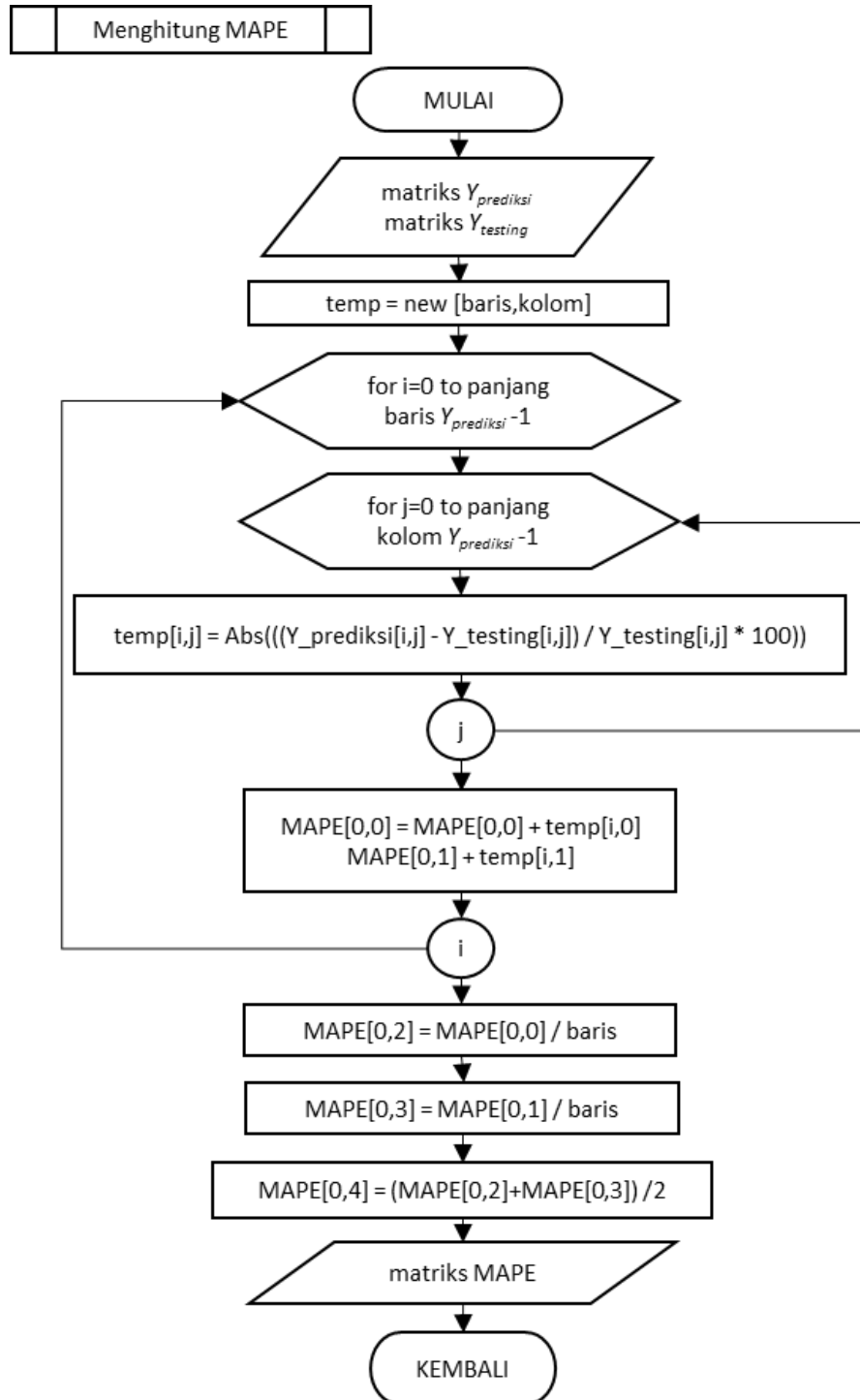
Gambar 4.16 Diagram Alir Perhitungan Y prediksi $\hat{Y}$

Langkah - langkah untuk melakukan perhitungan $Y_{prediksi}$ atau $\hat{Y}$ berdasarkan Gambar 4.16 adalah sebagai berikut:

1. Sistem menerima masukan berupa matriks H hasil perhitungan matriks keluaran *hidden layer* H dan bobot keluaran $\hat{\beta}$ hasil dari proses *training*.
2. Melakukan perkalian matriks antara matriks H dengan matriks $\hat{\beta}$ dengan menggunakan fungsi perkalian matriks seperti yang telah dijelaskan pada Gambar 4.9 sebelumnya.
3. Fungsi ini akan mengembalikan (*return*) nilai matriks $Y_{prediksi}$ atau $\hat{Y}$.

4.3.2.2 Menghitung MAPE

Pada tahap ini akan dilakukan perhitungan nilai evaluasi MAPE. Pada proses ini akan dihasilkan matriks MAPE yang terdiri dari MAPE untuk HL, MAPE untuk CL dan rata-rata MAPE. Diagram alir perhitungan MAPE akan ditunjukkan pada Gambar 4.17.



Gambar 4.17 Diagram Alir Perhitungan MAPE

Langkah - langkah untuk melakukan perhitungan nilai MAPE berdasarkan Gambar 4.17 adalah sebagai berikut:

1. Sistem ini membutuhkan matriks Y prediksi atau $\hat{Y}$ dan matriks target testing $Y_{testing}$.
2. Melakukan perhitungan nilai evaluasi MAPE dengan membandingkan matriks Y prediksi atau $\hat{Y}$ dengan matriks target testing $Y_{testing}$. Rumus MAPE dapat dilihat pada persamaan 2.15.
3. Proses ini mengembalikan (*return*) nilai matriks MAPE yang terdiri dari nilai MAPE untuk HL, MAPE untuk CL dan rata-rata MAPE.

4.4 Perhitungan Manual

Pada tahap perhitungan manualisasi terbagi menjadi tiga bagian yaitu Normalisasi Data, Perhitungan Tahap *Training* dan Perhitungan Tahap *Testing*. Manualisasi yang akan dilakukan ini bertujuan untuk memprediksi beban pemanasan (HL) dan beban pendinginan (CL) menggunakan algoritma *Extreme Machine Learning*. Perhitungan manualisasi akan menghasilkan nilai evaluasi *Mean Absolute Percentage Error* (MAPE).

4.4.1 Normalisasi Data

Pada tahap ini akan dilakukan proses normalisasi data menggunakan rumus normalisasi Min-max seperti yang telah dijabarkan pada Persamaan 2.1 dengan range normalisasi mulai 0.1 sampai 0.99. Pada manualisasi ini akan dinormalisasi data *training* dan data *testing* berjumlah 20 data, data training 15 data sedangkan data testing 5 data. Berikut ini contoh perhitungan normalisasi data:

$$y = \frac{x - Min}{Max - Min} * (B_{atas} - B_{bawah}) + B_{bawah}$$

$$y = \frac{0.8 - 0}{1 - 0} * (0.99 - 0.1) + 0.1$$

$$y = 0.8 * 0.89 + 0.1$$

$$y = 0.712 + 0.1$$

$$y = 0.812$$

Diketahui x adalah nilai yang akan dinormalisasi, *Max* dan *Min* ialah nilai maksimum dan minimum yang ada pada fitur *Relative Compactness* (RC) yaitu 1 dan 0. Sedangkan batas atas dan bawah adalah range nilai normalisasi yaitu 0.99 dan 0.1. Tabel data training yang telah dinormalisasi dapat dilihat pada Tabel 4.2.

Tabel 4.2 Data *Training* Normalisasi

RC	SA	WA	HL	CL
0.81	0.68	0.41	0.22	0.32
0.79	0.77	0.38	0.22	0.22
0.79	0.70	0.32	0.17	0.50
0.81	0.58	0.40	0.22	0.47
0.79	0.80	0.34	0.21	0.49
0.76	0.70	0.42	0.17	0.45
0.85	0.73	0.35	0.38	0.48
0.72	0.75	0.38	0.25	0.38
0.72	0.78	0.35	0.34	0.32
0.86	0.60	0.39	0.41	0.42
0.84	0.66	0.45	0.40	0.41
0.83	0.69	0.35	0.33	0.49
0.66	0.80	0.33	0.48	0.23
0.75	0.63	0.44	0.48	0.38
0.71	0.75	0.34	0.33	0.31

Sedangkan tabel data *testing* yang telah dinormalisasi dapat dilihat pada Tabel 4.3.

Tabel 4.3 Data *Testing* Normalisasi

RC	SA	WA	HL	CL
0.69	0.76	0.38	0.25	0.42
0.90	0.67	0.36	0.47	0.42
0.86	0.65	0.40	0.33	0.52
0.91	0.67	0.42	0.45	0.21
0.80	0.58	0.45	0.25	0.23

4.4.2 Tahap Training

Pada tahap *training* ini, data yang digunakan sebanyak 15 data dengan jumlah *hidden neuron* sebanyak 3 serta menggunakan fungsi aktivasi *sigmoid biner*. Parameter yang dibutuhkan untuk melakukan proses *training* antara lain, matriks bobot masukan W , dan matriks bias. Matriks bobot masukan W memiliki *ordo* jumlah *hidden neuron* x *input layer*, dengan jumlah *input layer* sebanyak 3 yakni RC, SA dan WA sedangkan matriks bias memiliki *ordo* 1 x *hidden neuron*. Sebelum melakukan proses *training*, data *training* yang telah dinormalisasi dibagi menjadi $X_{training}$ dan $Y_{training}$. $X_{training}$ digunakan sebagai *input layer* sedangkan $Y_{training}$ sebagai target. Data $X_{training}$ dapat dilihat pada Tabel 4.4

Tabel 4.4 Data $X_{training}$

$X_{training}$	RC	SA	WA
1	0.81	0.68	0.41
2	0.79	0.77	0.38
3	0.79	0.70	0.32
4	0.81	0.58	0.40
5	0.79	0.80	0.34
6	0.76	0.70	0.42
7	0.85	0.73	0.35
8	0.72	0.75	0.38
9	0.72	0.78	0.35
10	0.86	0.60	0.39
11	0.84	0.66	0.45
12	0.83	0.69	0.35
13	0.66	0.80	0.33
14	0.75	0.63	0.44
15	0.71	0.75	0.34

Data $Y_{training}$ digunakan sebagai target dapat dilihat pada tabel 4.5 sebagai berikut:

Tabel 4.5 Data $Y_{training}$

$Y_{training}$	HL	CL
1	0.22	0.32
2	0.22	0.22
3	0.17	0.50
4	0.22	0.47
5	0.21	0.49
6	0.17	0.45
7	0.38	0.48
8	0.25	0.38
9	0.34	0.32
10	0.41	0.42
11	0.40	0.41
12	0.33	0.49
13	0.48	0.23
14	0.48	0.38
15	0.33	0.31

4.4.2.1 Proses inisialisasi matriks bobot masukan W dan matriks bias

Pada proses inisialisasi nilai bobot masukan dan bias, nilai dipilih secara acak dengan rentang nilai tertentu. Untuk bobot masukan, nilai diacak pada rentang -1 sampai 1. Sedangkan bias, nilai diacak pada rentang 0 sampai 1. Jumlah *hidden neuron* yang digunakan sebanyak 3 dan input layer sebanyak 3 maka terbentuk matriks bobot masukan dengan ukuran 3x3 dan matriks bias dengan ukuran 1x3. Matriks bobot masukan dapat dilihat pada Tabel 4.6.

Tabel 4.6 Matriks Bobot Masukan

W	RC	SA	WA
1	-0.25	0.31	-0.21
2	0.89	-0.45	0.11
3	0.48	0.64	0.36

Berikut adalah matriks bias yang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Matriks bias

Bias	1	2	3
1	0.44	0.3	0.2

4.4.2.2 Menghitung matriks H_{init}

Pada proses menghitung H_{init} , matriks $X_{training}$ dikalikan dengan matriks *transpose* bobot masukan lalu ditambah *ones* dengan matriks bias. Untuk itu diperlukan matriks *transpose* dari matriks bobot masukan agar proses perhitungan H_{init} dapat berjalan. Matriks *transpose* ialah matriks yang mengubah nilai kolom menjadi nilai baris dan nilai baris menjadi nilai kolom. Matriks *transpose* bobot masukan dapat dilihat pada Tabel 4.8 dan Matriks H_{init} dapat dilihat pada Tabel 4.9.

Tabel 4.8 Matriks Transpose bobot masukan (W^T)

W	1	2	3
RC	-0.25	0.89	0.48
SA	0.31	-0.45	0.64
WA	-0.21	0.11	0.36

$$\begin{aligned} H_{init(1,1)} &= X_{training} \cdot W^T + b \\ &= ((0.812 \cdot -0.25) + (0.6785 \cdot 0.31) + (0.4115 \cdot -0.21)) + 0.44 \\ &= 0.3609 \end{aligned}$$

Tabel 4.9 Matriks H_{init}

H_{init}	1	2	3
1	0.36092	0.76262	1.17214
2	0.401994	0.700204	1.212368
3	0.392444	0.719953	1.140705
4	0.331888	0.806497	1.105141
5	0.4199	0.67767	1.207277
6	0.378008	0.70776	1.161994
7	0.380002	0.765824	1.197558
8	0.410671	0.649038	1.163596
9	0.426967	0.631799	1.171108
10	0.331114	0.834025	1.135864
11	0.340219	0.798594	1.189192
12	0.374341	0.765005	1.166729
13	0.45364	0.563883	1.149285
14	0.356532	0.730429	1.124329
15	0.422134	0.63551	1.146615

4.4.2.3 Menghitung matriks keluaran *hidden neuron H*

Setelah didapatkan matriks H_{init} maka selanjutnya dilakukan perhitungan matriks keluaran *hidden neuron H* dengan menggunakan fungsi aktivasi *sigmoid biner*. Berikut ini adalah contoh perhitungan untuk mendapatkan matriks keluaran *hidden neuron* dengan nilai *exponen* = 2.71828183. Hasil matriks keluaran *hidden neuron H* dengan fungsi aktivasi sigmoid biner secara keseluruhan dapat dilihat pada Tabel 4.10 sebagai berikut.

$$H = \frac{1}{1 + \exp(-(X.W^T) + b)}$$

$$H = \frac{1}{1 + \exp(-H_{init})}$$

$$H_{(1,1)} = \frac{1}{1 + \exp(-0.3609)}$$

$$= 0.5893$$

Di atas merupakan contoh perhitungan untuk menghitung nilai elemen ke-(1,1) matriks keluaran *hidden neuron H* menggunakan fungsi aktivasi *sigmoid biner* menghasilkan nilai 0.5893.

Tabel 4.10 Matriks keluaran *hidden neuron H*

<i>H</i>	1	2	3
1	0.589263	0.681922	0.763532
2	0.599167	0.668233	0.770718
3	0.596871	0.672597	0.757809
4	0.582219	0.691363	0.751222
5	0.603459	0.663218	0.769817
6	0.593393	0.669906	0.761695
7	0.593873	0.682617	0.76809
8	0.601249	0.656794	0.761986
9	0.605149	0.652897	0.763345
10	0.58203	0.697205	0.756919
11	0.584244	0.689674	0.766597
12	0.592508	0.682439	0.762553
13	0.611504	0.637351	0.75938
14	0.588201	0.674899	0.754791
15	0.603994	0.653738	0.758892

4.4.2.4 Menghitung matriks H^+ *Pseudo-Inverse* dengan *Moore-Penrose*

Setelah mendapatkan matriks keluaran *hidden neuron* yang telah dihitung menggunakan fungsi aktivasi *sigmoid biner* maka selanjutnya menghitung matriks H^+ *Pseudo-Inverse* dengan *Moore-Penrose*. Diawali dengan membuat matriks *transpose* dari matriks keluaran *hidden neuron H* sehingga berdimensi 3x15. Matriks *transpose H* (H^T) dapat dilihat pada table 4.11.

$$H^+ = (H^T . H)^{-1} . H^T \quad (3.4)$$

Tabel 4.11 Matriks *transpose* keluaran *hidden neuron H*

H^T	1	2	3	...	13	14	15
1	0.589	0.599	0.597	...	0.612	0.588	0.604
2	0.682	0.668	0.673	...	0.637	0.675	0.654
3	0.764	0.771	0.758	...	0.759	0.755	0.759

Setelah matriks H^T telah dibentuk maka selanjutnya perkalian H^T dengan matriks H sehingga terbentuk matriks 3x3. Matriks 3x3 tersebut dapat dilihat lebih jelas pada Tabel 4.12. Berikut adalah contoh perhitungannya.

$$= H^T . H \quad (3.5)$$

$$\begin{aligned}
&= (0.589*0.589) + (0.599*0.599) + (0.597*0.597) + \dots + (0.612*0.612) + \\
&\quad (0.588*0.588) + (0.604*0.604) \\
&= 5.314026
\end{aligned}$$

Tabel 4.12 Matriks hasil perkalian H^T dengan H

$H^T.H$	1	2	3
1	5.314026	5.993975	6.801112
2	5.993975	6.77072	7.675082
3	6.801112	7.675082	8.706042

Selanjutnya membuat matriks *inverse* dari matriks 3x3 hasil perkalian H^T dengan H . Matriks *inverse* merupakan matriks yang jika dikalikan dengan dirinya sendiri maka akan menghasilkan matriks identitas. Metode untuk menyelesaikan invers yang digunakan pada penelitian ini adalah Operasi Baris Elementer (OBE) dengan cara menyendingkan matriks yang akan diinverskan dengan matriks identitas kemudian mengubah matriks yang akan diinverskan menjadi matriks identitas sedangkan matriks identitas menjadi hasil invers matriks. Jadi matriks baru yang dihasilkan dari matriks identitas menjadi hasil dari invers matriks. Perhitungan invers matriks menggunakan OBE adalah sebagai berikut:

1. Bentuk matriks identitas (I)

$$[H | I] = \left[\begin{array}{ccc|ccc} 5.314 & 5.994 & 6.801 & 1 & 0 & 0 \\ 5.994 & 6.770 & 7.675 & 0 & 1 & 0 \\ 6.801 & 7.675 & 8.706 & 0 & 0 & 1 \end{array} \right]$$

2. $R1 : 5.315 \rightarrow R1$

$$[H | I] = \left[\begin{array}{ccc|ccc} 1 & 1.128 & 1.280 & 0.188 & 0 & 0 \\ 5.994 & 6.770 & 7.675 & 0 & 1 & 0 \\ 6.801 & 7.675 & 8.706 & 0 & 0 & 1 \end{array} \right]$$

3. $R2 - (5.994 \times R1) \rightarrow R2$

$$[H | I] = \left[\begin{array}{ccc|ccc} 1 & 1.128 & 1.280 & 0.188 & 0 & 0 \\ 0 & 0.010 & 0.004 & -1.128 & 1 & 0 \\ 6.801 & 7.675 & 8.706 & 0 & 0 & 1 \end{array} \right]$$

4. $R3 - (6.801 \times R1) \rightarrow R3$

$$[H | I] = \left[\begin{array}{ccc|ccc} 1 & 1.128 & 1.280 & 0.188 & 0 & 0 \\ 0 & 0.010 & 0.004 & -1.127 & 1 & 0 \\ 0 & 0.004 & 0.001 & -1.278 & 0 & 1 \end{array} \right]$$

5. $R2 : 0.009 \rightarrow R2$

$$[H | I] = \begin{bmatrix} 1 & 1.128 & 1.280 \\ 0 & 1 & 0.400 \\ 0 & 0.004 & 0.001 \end{bmatrix} \begin{bmatrix} 0.188 & 0 & 0 \\ -109.900 & 97.437 & 0 \\ -1.278 & 0 & 1 \end{bmatrix}$$

6. $R1 - (1.128 \times R2) \rightarrow R1$

$$[H | I] = \begin{bmatrix} 1 & 0 & 0.828 \\ 0 & 1 & 0.400 \\ 0 & 0.004 & 0.001 \end{bmatrix} \begin{bmatrix} 124.145 & -109.900 & 0 \\ -109.900 & 97.437 & 0 \\ -1.278 & 0 & 1 \end{bmatrix}$$

7. $R3 - (0.003 \times R2) \rightarrow R3$

$$[H | I] = \begin{bmatrix} 1 & 0 & 0.828 \\ 0 & 1 & 0.400 \\ 0 & 0 & -3.685 \end{bmatrix} \begin{bmatrix} 124.145 & -109.900 & 0 \\ -109.900 & 97.437 & 0 \\ -0.828 & -0.400 & 1 \end{bmatrix}$$

8. $R3 : -3.685 \rightarrow R3$

$$[H | I] = \begin{bmatrix} 1 & 0 & 0.828 \\ 0 & 1 & 0.400 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 124.145 & -109.900 & 0 \\ -109.900 & 97.437 & 0 \\ 22482.506 & 10852.743 & -27131.414 \end{bmatrix}$$

9. $R1 - (0.828 \times R3) \rightarrow R1$

$$[H | I] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.400 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -18506.032 & -9103.050 & 22482.506 \\ -109.900 & 97.437 & 0 \\ 22482.506 & 10852.743 & -27131.414 \end{bmatrix}$$

10. $R2 - (0.400 \times R3) \rightarrow R2$

$$[H | I] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -18506.032 & -9103.050 & 22482.506 \\ -9103.050 & -4243.731 & 10852.743 \\ 22482.506 & 10852.743 & -27131.414 \end{bmatrix}$$

Tabel yang menunjukkan hasil inverse dapat dilihat pada Tabel 4.13.

Tabel 4.13 **Matriks *inverse* hasil perkalian H^T dengan H**

$(H^T.H)^{-1}$	1	2	3
1	-18506.032	-9103.050	22482.506
2	-9103.050	-4243.731	10852.743
3	22482.506	10852.743	-27131.414

Setelah mendapatkan matriks *inverse* dari perkalian H^T dengan H maka matriks *inverse* tersebut dikalikan dengan matriks *transpose* H (H^T). Untuk matriks hasil perkalian dapat dilihat pada Tabel 4.14 Berikut adalah contoh perhitungannya.

$$H^+_{(1,1)} = (-18506.032*0.5893) + (-9103.050*0.6819) + (22482.50*0.7635) \\ = 53.608$$

Tabel 4.14 Matriks *pseudo-inverse* H^+

H+	1	2	3	...	13	14	15
1	53.608	156.510	-130.946	...	-45.583	-59.315	-66.743
2	28.426	74.357	-63.357	...	-29.940	-26.961	-36.413
3	-66.853	-187.733	158.241	...	62.092	70.193	84.330

4.4.2.5 Menghitung bobot keluaran $\hat{\beta}$

Setelah mendapat matriks *pseudo-inverse* H^+ dengan *Moore-Penrose* selanjutnya melakukan perhitungan bobot keluaran $\hat{\beta}$ dengan cara melakukan perkalian matriks H^+ dengan matriks target $Y_{training}$. Hasil matriks bobot keluaran $\hat{\beta}$ dapat dilihat pada Tabel 4.15. Berikut contoh perhitungannya

$$\beta = H^+ . Y_{training}$$

$$\beta_{(1,1)} = (53.608*0.2246) + (156.51*0.2157) + (-130.946*0.1712) + \dots + (-45.583*0.4827) + (-59.315*0.4827) + (-66.743*0.3314)$$

$$\beta_{(1,1)} = -16.55$$

Tabel 4.15 Matriks bobot keluaran $\hat{\beta}$

$\hat{\beta}$	1	2
1	-16.55	-33.91
2	-8.11	-13.63
3	20.49	39.02

4.4.3 Tahap Testing

Pada tahap *testing* ini, data yang digunakan sebanyak 5 data dengan jumlah *hidden neuron* seperti pada proses *training* yaitu 3 serta menggunakan fungsi aktivasi *sigmoid biner*. Parameter yang dibutuhkan untuk melakukan proses *testing* antara lain, matriks bobot masukan W , dan matriks bias. Matriks bobot masukan dan bias yang digunakan pada tahap *testing* ini sama dengan matriks bobot masukan dan bias yang digunakan pada tahap *training*. Sebelum melakukan proses *testing*, data *testing* yang telah dinormalisasi dibagi menjadi $X_{testing}$ dan $Y_{testing}$. $X_{testing}$ digunakan sebagai *input layer* sedangkan $Y_{testing}$ sebagai target. Data $X_{testing}$ dapat dilihat pada Tabel 4.16

Tabel 4.16 Data $X_{testing}$

$X_{testing}$	RC	SA	WA
1	0.69	0.76	0.38
2	0.90	0.67	0.36
3	0.86	0.65	0.40
4	0.91	0.67	0.42
5	0.80	0.58	0.45

Data $Y_{testing}$ digunakan sebagai target dapat dilihat pada tabel 4.17 sebagai berikut:

Tabel 4.17 Data $Y_{testing}$

$Y_{testing}$	HL	CL
1	0.25	0.42
2	0.47	0.42
3	0.33	0.52
4	0.45	0.21
5	0.25	0.23

4.4.3.1 Menghitung matriks H_{init}

Pada proses menghitung H_{init} , matriks $X_{testing}$ dikalikan dengan matriks *transpose* bobot masukan lalu ditambah *ones* dengan matriks bias. Untuk itu diperlukan matriks *transpose* dari matriks bobot masukan agar proses perhitungan H_{init} dapat berjalan. Matriks *transpose* ialah matriks yang mengubah nilai kolom menjadi nilai baris dan nilai baris menjadi nilai kolom. Matriks *transpose* bobot masukan dapat dilihat pada Tabel 4.8 dan Matriks H_{init} dapat dilihat pada Tabel 4.18. Berikut ini perhitungan matriks H_{init} :

$$\begin{aligned}
 H_{init(1,1)} &= X_{testing} \cdot W^T + b \\
 &= ((0.69 \cdot 0.25) + (0.76 \cdot 0.31) + (0.38 \cdot 0.21)) + 0.44 \\
 &= 0.423
 \end{aligned}$$

Tabel 4.18 Data matriks H_{init}

H_{init}	1	2	3
1	0.423	0.613	1.150
2	0.346	0.841	1.189
3	0.343	0.815	1.167
4	0.333	0.853	1.220
5	0.325	0.803	1.118

4.4.3.2 Menghitung matriks keluaran *hidden neuron H*

Setelah didapatkan matriks H_{init} maka selanjutnya dilakukan perhitungan matriks keluaran *hidden neuron H* dengan menggunakan fungsi aktivasi *sigmoid biner*. Berikut ini adalah contoh perhitungan untuk mendapatkan matriks keluaran *hidden neuron* dengan nilai *exponen* = 2.71828183. Hasil matriks keluaran *hidden neuron H* dengan fungsi aktivasi sigmoid biner secara keseluruhan dapat dilihat pada Tabel 4.19 sebagai berikut.

$$H = \frac{1}{1 + \exp(-(X.W^T) + b)}$$
$$H = \frac{1}{1 + \exp(-H_{init})}$$
$$H_{(1,1)} = \frac{1}{1 + \exp(-0.423)}$$
$$= 0.604$$

Perhitungan manual di atas merupakan contoh perhitungan manual untuk mencari nilai elemen matriks ke [1,1] dari matriks H. Perhitungan tersebut menggunakan fungsi aktivasi *sigmoid biner* dan membutuhkan matriks H_{init} yang telah didapatkan pada proses perhitungan sebelumnya.

Tabel 4.19 Matriks keluaran *hidden neuron H*

H	1	2	3
1	0.604	0.649	0.760
2	0.586	0.699	0.766
3	0.585	0.693	0.763
4	0.582	0.701	0.772
5	0.581	0.691	0.754

4.4.3.3 Menghitung prediksi $\hat{Y}$

Setelah didapatkan matriks H maka selanjutnya dilakukan perhitungan matriks prediksi $\hat{Y}$ dengan cara melakukan perkalian matriks keluaran *hidden neuron H* dengan matriks bobot keluaran $\hat{\beta}$ yang telah didapatkan pada tahap training. Matriks prediksi $\hat{Y}$ dapat dilihat pada Tabel 4.20. Berikut adalah perhitungan matriks prediksi $\hat{Y}$:

$$\hat{Y} = H.\hat{\beta}$$
$$\hat{Y}(1,1) = (0.604*-16.55) + (0.649*-8.11) + (0.760*20.49)$$
$$\hat{Y}(1,1) = 0.300$$

Tabel 4.20 Matriks prediksi $\hat{Y}$

$\hat{Y}$	HL	CL
1	0.300	0.312
2	0.341	0.526
3	0.319	0.476
4	0.491	0.824
5	0.228	0.307

4.4.3.4 Menghitung MAPE

Setelah mendapatkan matriks prediksi $\hat{Y}$ maka selanjutnya mencari nilai evaluasi MAPE dengan cara membandingkan matriks prediksi $\hat{Y}$ dengan matriks target $Y_{testing}$. Berikut persamaan dan contoh perhitungan MAPE.

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_{prediksi} - y_{testing}}{y_{testing}} \times 100 \right|$$

$$MAPE_{HL} = \frac{1}{5} (19.208 + 28.012 + 3.648 + 9.8585 + 9.3971)$$

$$MAPE_{HL} = 14.024$$

Di atas merupakan perhitungan untuk mencari nilai MAPE dengan target *Heating Load* (HL).

$$MAPE_{CL} = \frac{1}{15} (25.856 + 25.148 + 8.1333 + 298.62 + 31.591)$$

$$MAPE_{CL} = 77.869$$

Di atas merupakan perhitungan untuk mencari nilai MAPE dengan target *Cooling Load* (HL).

$$MAPE_{testing} = \frac{MAPE_{HL} + MAPE_{CL}}{2}$$

$$MAPE_{testing} = \frac{14.024 + 77.869}{2}$$

$$MAPE_{testing} = 45.947$$

Di atas merupakan perhitungan untuk mencari rata-rata nilai MAPE dari MAPE *Heating Load* dan MAPE *Cooling Load*. Jadi rata-rata nilai MAPE yang diperoleh dari perhitungan manual adalah 45.947.

4.5 Perancangan Skenario Pengujian

Pengujian yang akan dilakukan pada penelitian ini merupakan pengujian terkait algoritme *Extreme Machine Learning* yang digunakan untuk memprediksi beban pemanasan dan beban pendinginan bangunan. Tujuan dari pengujian ini adalah untuk mengetahui kinerja dari metode yang digunakan. Setiap pengujian akan dilakukan dengan 10 kali percobaan. Skenario pengujian yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Pengujian Perbandingan Data Latih dan Data Uji
2. Pengujian Jumlah *Hidden Neuron*
3. Pengujian Fungsi Aktivasi

4.5.1 Pengujian Perbandingan Data Latih dan Data Uji

Pengujian perbandingan data latih dan data uji dilakukan untuk mengetahui pengaruh dari rasio jumlah data latih dan jumlah data uji terhadap kemampuan algoritme *Extreme Learning Machine* dalam mengenali pola data. Pengujian ini terdiri 9 jenis perbandingan data latih dan uji yaitu 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, 40%:60%, 30%:70%, 20%:80%, 10%:90%. Rancangan pengujian untuk perbandingan data latih dan data uji akan ditunjukkan pada Tabel 4.21.

Tabel 4.21 Tabel Skenario Pengujian Perbandingan Data Latih dan Uji

Percobaan ke- <i>i</i>	Nilai MAPE Pada Perbandingan Data Latih dan Data Uji								
	90:10	80:20	70:30	60:40	50:50	40:60	30:70	20:80	10:90
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
Rata-rata MAPE									

4.5.2 Pengujian Jumlah *Hidden Neuron*

Pengujian jumlah *hidden neuron* dilakukan untuk mengetahui pengaruh jumlah *hidden neuron* terhadap nilai evaluasi MAPE dalam implementasi algoritme ELM. Jumlah *hidden neuron* yang akan digunakan pada pengujian ini antara lain 1, 2, 3, 5, 10, 20, 30, 50, 100 dengan *range* bobot masukan -1 sampai 1 dan *range* bias 0 sampai 1. Perbandingan data latih dan data uji menggunakan

rasio 70%:30% untuk data latih dan data uji. Rancangan pengujian untuk pengujian jumlah *hidden neuron* akan ditunjukkan pada Tabel 4.22.

Tabel 4.22 Tabel Skenario Pengujian Jumlah *Hidden Neuron*

Percobaan ke- <i>i</i>	Nilai MAPE Pada Jumlah <i>Hidden Neuron</i>								
	1	2	3	5	10	20	30	50	100
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
MAPE									

4.5.3 Pengujian Fungsi Aktivasi

Pengujian fungsi aktivasi dilakukan untuk mengetahui pengaruh penggunaan fungsi aktivasi terhadap nilai evaluasi MAPE dalam implementasi algoritme ELM. Fungsi aktivasi digunakan pada proses menghitung nilai keluaran *hidden neuron* yaitu matriks *H*. Fungsi aktivasi yang akan digunakan pada pengujian ini antara lain fungsi *sin*, fungsi *sigmoid biner*, fungsi *radial basis*, fungsi *linear*, dan fungsi *sigmoid bipolar*. Perbandingan data latih dan data uji menggunakan rasio 70%:30% untuk data latih dan data uji. Rancangan pengujian untuk penggunaan fungsi aktivasi akan ditunjukkan pada Tabel 4.23.

Tabel 4.23 Tabel Skenario Pengujian Jumlah *Hidden Neuron*

Percobaan ke- <i>i</i>	Nilai MAPE Pada Fungsi Aktivasi				
	<i>Sigmoid Biner</i>	<i>Linear</i>	<i>Sigmoid Bipolar</i>	<i>Sin</i>	<i>Radial Basis</i>
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
MAPE					