

## BAB III

### METODOLOGI PENELITIAN DAN PERANCANGAN

#### 3.1 Metodologi Penelitian

Proses penelitian Klasifikasi Kelangsungan Hidup Penderita Gagal Jantung menggunakan *Extreme Learning Machine* (ELM) meliputi beberapa metode yang dilakukan oleh peneliti adalah penentuan topik, perumusan masalah, studi literatur, perancangan algoritma, implementasi algoritma, evaluasi, dan konsultasi penelitian. Metode yang digunakan sebagai berikut.

a. Perumusan Masalah

Kegiatan yang dilakukan pada tahap ini adalah mencari permasalahan yang sedang dihadapi dokter dan paramedis.

b. Studi Literatur

Tahap studi literatur dilakukan dengan mempelajari dan mendalami topik yang terkait dengan *Machine Learning*, algoritma *Extreme Learning Machine*, Normalisasi *Minmax*, penyakit gagal jantung, serta data pasien penderita gagal jantung kelas III dan IV pada situs Kaggle.com (<https://www.kaggle.com/andrewmvd/heart-failure-clinical-data>) dan penelitian yang terkait.

c. Perancangan Algoritma

Tahap perancangan algoritma yaitu melakukan perancangan proses klasifikasi berupa pembagian data, normalisasi, dan pembuatan model *Extreme Learning Machine*. Rancangan yang dibuat berupa *flowchart* dan perhitungan manual.

d. Implementasi Algoritma

Proses implementasi Algoritma dilakukan dengan mengaplikasikan hasil rancangan normalisasi dan algoritme *Extreme Learning Machine* pada tahap sebelumnya ke dalam *code*.

e. Evaluasi

Proses evaluasi dilakukan dengan menguji akurasi data menggunakan model *Extreme Learning Machine* yang telah dihasilkan. Evaluasi berupa menguji data keseluruhan menggunakan *Confusion Matrix* dengan Teknik *K-Fold Cross Validation* untuk mendapatkan akurasi rata-rata. Proses evaluasi dilakukan untuk mengetahui seberapa tepat model dapat melakukan klasifikasi data.

f. Konsultasi Penelitian

Tahap Konsultasi dilakukan dengan melakukan proses bimbingan dan diskusi dengan dosen pembimbing untuk menyempurnakan penelitian. Adapun penulisan laporan dilakukan dengan tujuan dokumentasi penelitian agar dapat digunakan dalam keperluan proses akademik. Selain itu diharapkan agar penelitian ini dapat dikembangkan lebih lanjut dengan mempertimbangkan hasil penelitian dan saran penelitian.

### 3.2 Perancangan Algoritma

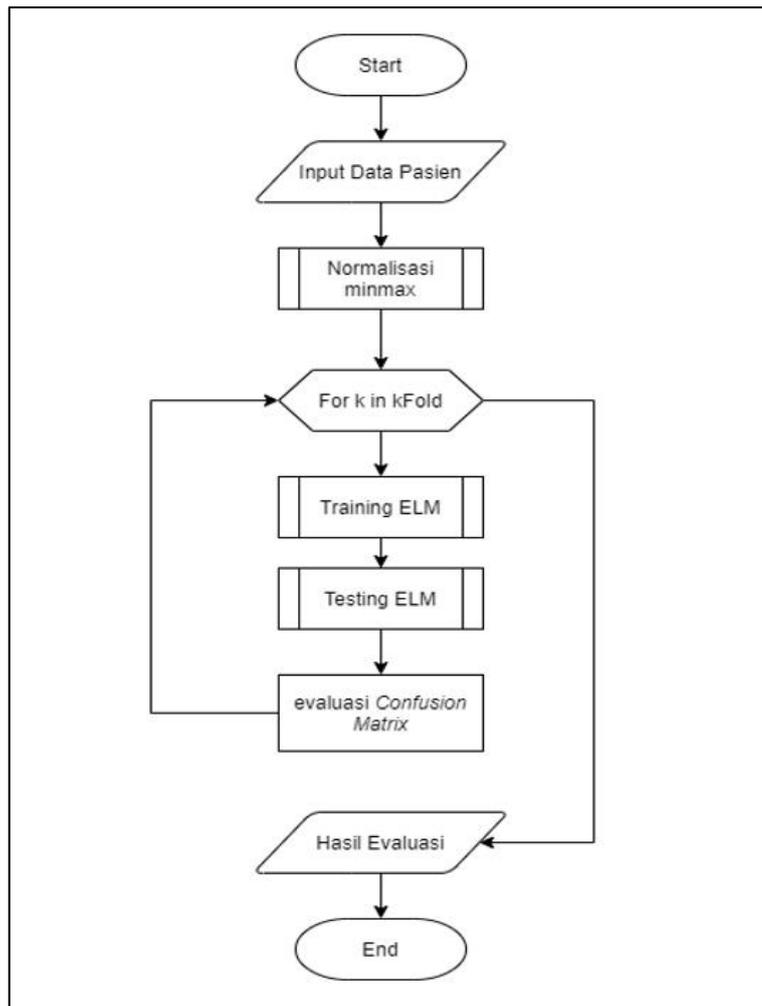
Pada tahap perancangan Algoritma yang dilakukan yaitu penggambaran *flowchart* dimulai dari alur algoritma hingga proses *testing* ELM dan manualisasi yang menjabarkan perhitungan algoritma menggunakan data sampel.

### 3.2.1 Flowchart

*Flowchart* menjelaskan mengenai alur kerja algoritma yang digunakan.

*Flowchart* penelitian dijabarkan sebagai berikut.

#### 1. Flowchart Alur Algoritma

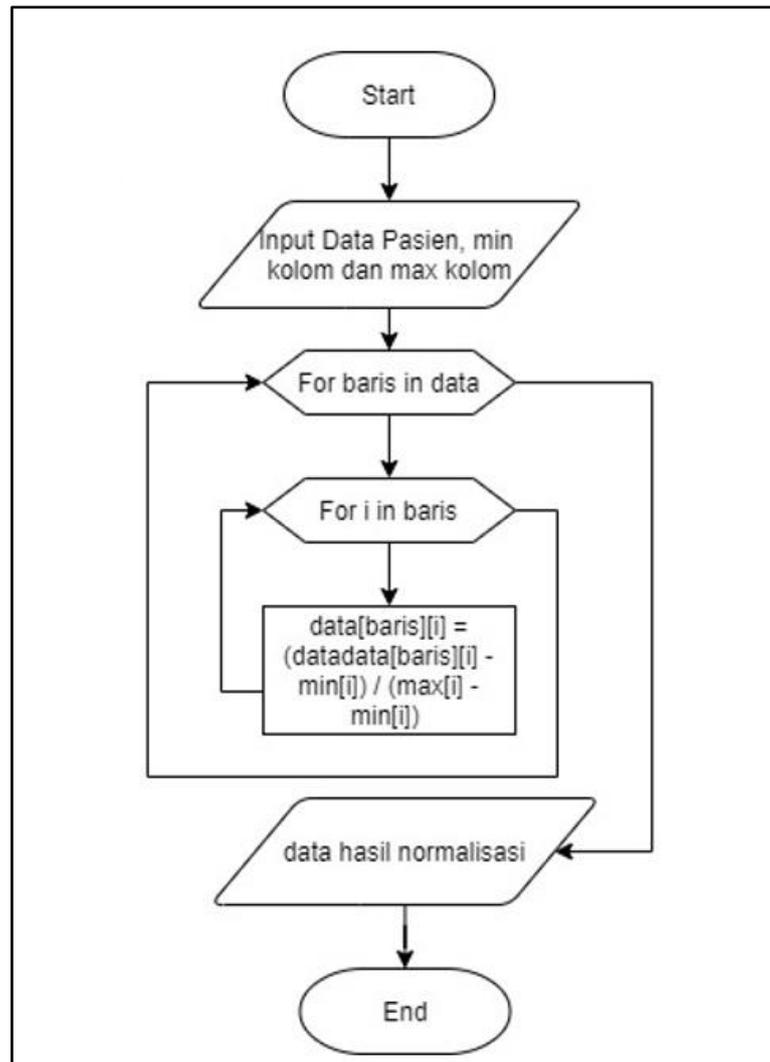


Gambar 3.1 *Flowchart* Alur Algoritma

*Flowchart* alur algoritma dimulai dengan menginputkan data latih yang kemudian akan dinormalisasi menggunakan minmax untuk menghindari kesenjangan nilai antarfitur. Hasil normalisasi tersebut akan digunakan sebagai input pada proses ELM. Kemudian data akan dibagi menjadi beberapa *fold*. Pada

setiap *fold* akan dilakukan *training* dan *testing* menggunakan ELM. Kemudian dilakukan perhitungan evaluasi menggunakan *confusion matrix*.

## 2. Flowchart Normalisasi Minmax

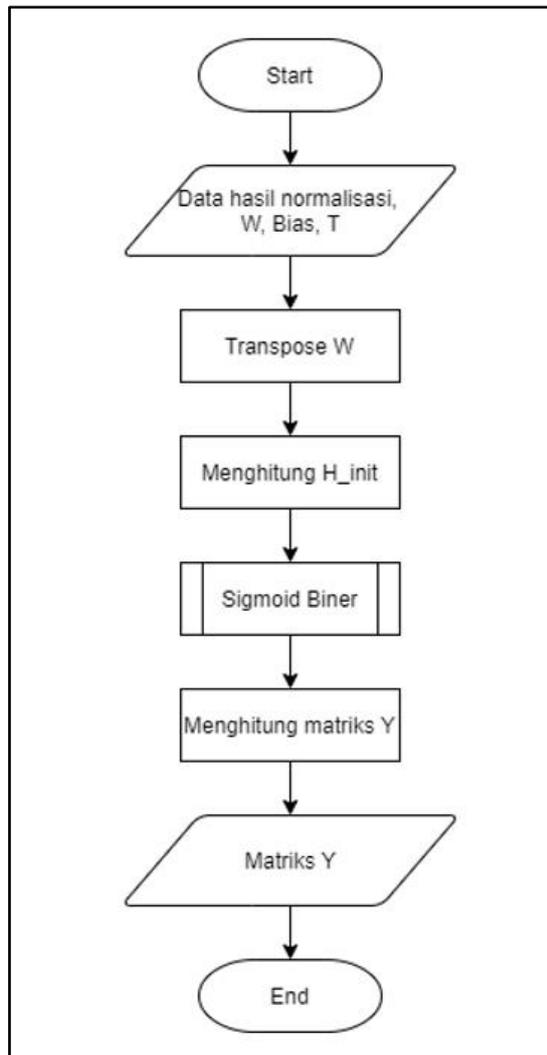


Gambar 3.2 *Flowchart* Normalisasi Minmax

*Flowchart* normalisasi menggunakan 3 parameter input, yaitu data penelitian, nilai minimal dari setiap fitur, dan nilai maksimal setiap fitur. Kemudian dilakukan proses perulangan pada setiap data pada fitur. Nilai pada fitur akan

dinormalisasi menggunakan Persamaan 6. Sehingga didapatkan data hasil normalisasi dengan rentang 0 sampai dengan 1.

### 3. Flowchart *Training ELM*

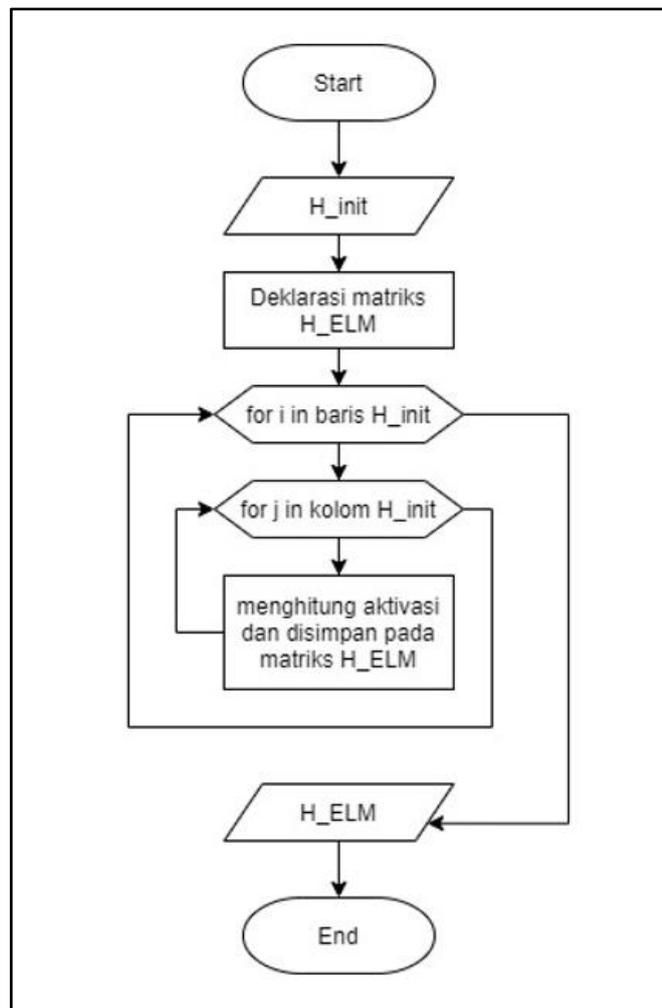


Gambar 3.3 *Flowchart Training ELM*

Proses *Flowchart Training ELM* membutuhkan beberapa input antara lain data hasil normalisasi, matriks bobot  $W$ , matriks bias, dan matriks target  $T$ . Kemudian dilakukan proses transpose matriks bobot  $W$ . Selanjutnya dilakukan perhitungan

matriks H init yang selanjutnya menjadi input pada aktivasi sigmoid biner. Setelah mendapatkan hasil aktivasi maka dilakukan perhitungan matriks *Moore Penrose* menggunakan Persamaan 2. Dan terakhir yaitu menghitung matriks B dengan persamaan 3 yang mana hasil matriks ini akan digunakan pada proses *testing*.

#### 4. Flowchart Fungsi Aktivasi Sigmoid

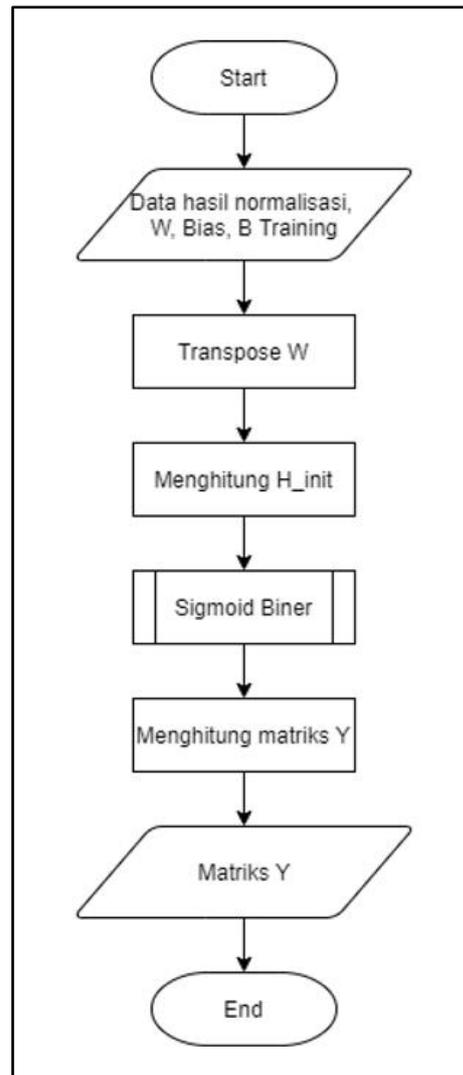


Gambar 3.4 *Flowchart* Fungsi Aktivasi Sigmoid

*Flowchart* fungsi aktivasi sigmoid memiliki satu input yaitu matriks  $H_{init}$ . Kemudian dilakukan deklarasi matriks H ELM untuk menyimpan hasil aktivasi sigmoid. Proses selanjutnya yaitu dilakukan perulangan pada setiap data kemudian

dilakukan perhitungan aktivasi sigmoid menggunakan Persamaan 5. Hasil aktivasi kemudian disimpan di matriks H ELM.

### 5. Flowchart *Testing* ELM



Gambar 3.5 *Flowchart Testing* ELM

Proses *Flowchart Testing* ELM membutuhkan beberapa input antara lain data hasil normalisasi, matriks bobot  $W$ , matriks bias, dan matriks  $B$  hasil proses *training* ELM. Kemudian dilakukan proses transpose matriks bobot  $W$  dan

menghitung H init. Setelah itu dilakukan aktivasi menggunakan sigmoid biner. Kemudian dilakukan perhitungan matriks Y menggunakan Persamaan 4. Hasil dari matriks ini menjadi penentu kelas pada setiap data *testing*.

### 3.2.2 Manualisasi

Manualisasi menjabarkan mengenai proses perhitungan algoritma pada penelitian dengan cara manual. Hal ini bertujuan untuk memberikan gambaran umum terkait penggunaan algoritma *Extreme Learning Machine* (ELM). Proses perhitungan menggunakan data sampel dari setiap kelas.

#### 1. Data Sampel

Pada proses manualisasi, data sampel yang digunakan terdiri dari 7 data dengan enam data *training* (No 1 hingga 6) dan satu data *testing* (No 7). Data sampel terlihat pada Tabel 3.1. Adapun penjelasan fitur pada Tabel 3.1 antara lain usia (F1), anemia (F2), kreatinin fosfokinase (F3), diabetes (F4), *ejection fraction* (F5), tingkat tekanan darah (F6), *platelets* (F7), serum kreatinin (F8), level natrium serum sodium (F9), jenis kelamin (F10), perokok (F11), jenis kelamin (F12), dan kelas klasifikasi yaitu *death event* (F13).

Tabel 3.1 Data Sampel Manualisasi

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
1	90	1	47	0	40	1	204000	2,1	132	1	1	8	1
2	70	1	125	0	25	1	237000	1	140	0	0	15	1
3	60	1	315	1	60	0	454000	1,1	131	1	1	10	1
4	75	0	582	0	45	1	263358	1,18	137	1	0	87	0
5	80	0	898	0	25	0	149000	1,1	144	1	1	87	0
6	42	0	5209	0	30	0	226000	1	140	1	1	87	0
7	70	0	122	1	45	1	284000	1,3	136	1	1	26	1

## 2. Normalisasi

Pada proses normalisasi dilakukan penyamarataan skala data agar tidak terjadi kesenjangan antarfitur. Normalisasi yang digunakan yaitu Minmax menggunakan Persamaan 6. Proses perhitungan normalisasi minmax pada fitur umur dengan umur yaitu 90 sebagai berikut.

$$v'_{umur,1} = \frac{90 - 42}{90 - 42} = 1$$

Hasil perhitungan normalisasi minmax pada seluruh data sampel terlihat pada Tabel 3.2. Data No 1 sampai dengan data No 6 merupakan data *training*, sedangkan data No 7 merupakan data *testing*.

Tabel 3.2 Hasil normalisasi data sampel

No	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
1	1	1	0	0	0,43	1	0,18	1	0,08	1	1	0
2	0,58	1	0,02	0	0	1	0,29	0	0,69	0	0	0,09
3	0,38	1	0,05	1	1	0	1	0,09	0	1	1	0,02
4	0,69	0	0,1	0	0,57	1	0,37	0,16	0,46	1	0	1
5	0,79	0	0,16	0	0	0	0	0,09	1	1	1	1
6	0	0	1	0	0,14	0	0,25	0	0,69	1	1	1
7	0,58	0	0,02	1	0,57	1	0,44	0,27	0,38	1	1	0,23

## 3. Proses Training ELM

Training ELM diawali dengan melakukan inisialisasi beberapa komponen yang dibutuhkan, seperti jumlah *hidden neuron*, matriks bobot *hidden layer*, matriks bias dan matriks target. Pada proses manualisasi, jumlah *hidden neuron* yang digunakan sebanyak 5. Matriks bobot *hidden layer* diinisialisasi dengan menentukan bilangan secara acak dari 0 hingga 1 dan terlihat pada Tabel 3.3.

Matriks bias diinisialisasi menggunakan bilangan acak antara 0 hingga 1 dan terlihat pada Tabel 3.4. Dan matriks target dapat dilihat pada Tabel 3.5.

Tabel 3.3 Matriks bobot *hidden layer*

HN	1	2	3	4	5	6	7	8	9	10	11	12
1	0,51	0,01	0,67	0,73	0,83	0,3	0,37	0,11	0,76	0,84	0,77	0,54
2	0,41	0,71	0,88	0,21	0,45	0,3	0,65	0,18	0,33	0,92	0,81	0,09
3	0,31	0,35	0,49	0,32	0,8	0,73	0,63	0,15	0,92	0,08	0,57	0,98
4	0,26	0,27	0,17	0,97	0,22	0,51	0,36	0,55	0,27	0,52	0,05	0,6
5	0,78	0,24	0,39	0,56	0,09	0,98	0,55	0,32	0,27	0,66	0,01	0,76

Tabel 3.4 Matriks Bias

	1	2	3	4	5
BIAS	0,1	0,14	0,05	0,17	0,59

Tabel 3.5 Matriks Target

Data ke-	Kelas 0	Kelas 1
1	-1	1
2	-1	1
3	-1	1
4	1	-1
5	1	-1
6	1	-1

Setelah mendapatkan komponen yang dibutuhkan maka selanjutnya dilakukan perhitungan matriks H dengan aktivasi *hidden layer*. Pada penelitian, fungsi aktivasi yang digunakan adalah sigmoid biner sesuai dengan Persamaan 5. Proses perhitungan dilakukan menggunakan Persamaan 1. Berikut merupakan contoh perhitungan data pertama pada fitur umur.

$$H_{1,1} = \frac{1}{1 + \exp^{-(X_{training} \cdot W_{hidden}^T + b)}} = \frac{1}{1 + \exp^{-(1 \cdot 0,51 + \dots + (0 \cdot 0,54) + 0,1)}} = 3,13$$

Tabel 3.6 Matriks H ELM *training*

	1	2	3	4	5	6
1	3,13	3,81	2,76	2,5	3,74	3,13
2	1,4	1,83	2,22	1,44	2,69	1,4
3	3,9	4,11	2,97	2,72	3,05	3,9
4	3,19	2,51	3,24	2,47	3,99	3,19
5	3,54	2,78	2,93	1,89	2,99	3,54
6	3,67	3,3	3,07	1,82	2,74	3,67

Kemudian dilakukan proses perhitungan matriks *Moore Penrose* menggunakan Persamaan 2. Moore Penrose mengalikan matriks inversi dari perkalian H *transpose* dengan matriks H, kemudian dikalikan lagi dengan H *transpose*. Hasil generalisasi *Moore Penrose* terlihat pada Tabel 3.7.

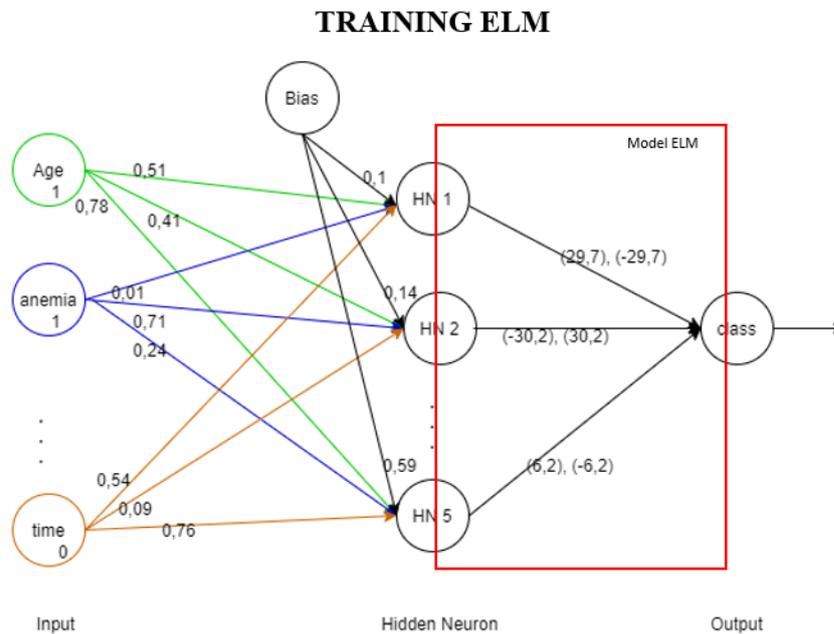
Tabel 3.7 Matriks *Moore Penrose* ELM *training*

	1	2	3	4	5	6
1	6,83	-10	-12	4,53	11,4	-1,3
2	7,92	3,12	4,26	-16	-3,8	5,13
3	-34,9	9,47	21,1	5,18	-12	12,5
4	-6,86	-0,4	15,4	7,78	-11	-5,3
5	26,4	-2	-27	-0,6	14,9	-11
6	6,83	-10	-12	4,53	11,4	-1,3

Tahap terakhir yaitu menghitung matriks *output* ELM  $\beta$  dengan Persamaan 3. Matriks *output* dihitung dengan mengalikan matriks *Moore Penrose* dengan matriks target kelas yang telah diinisialisasi sebelumnya. Hasil perhitungan matriks *output* ELM  $\beta$  terlihat pada Tabel 3.8.

Tabel 3.8 Matriks *output* ELM  $\beta$  *training*

	1	2
1	29,7	-29,7
2	-30,2	30,2
3	9,57	-9,6
4	-16,4	16,4
5	6,2	-6,2



Gambar 3.6 Ilustrasi *Neural Network* proses *Training* ELM

Ilustrasi proses *training* ELM pada Gambar 3.6 terdiri dari 3 bagian yaitu *input*, *hidden neuron*, dan *output*. Pada bagian input terdapat 12 neuron yang merupakan data *training* yang terdiri dari 12 fitur, yaitu *age*, *anemia* hingga *time* (Tabel 3.2 No 1-6). Kemudian antara *input* dan *hidden neuron* terdapat garis yang terhubung yaitu bobot *hidden neuron* (Tabel 3.3). *Hidden neuron* juga terhubung dengan neuron bias (Tabel 3.4). Kemudian antara *hidden neuron* dan *output* terdapat model hasil

perhitungan ELM (Tabel 3.8) yang mana model tersebut akan digunakan dalam perhitungan pada proses *testing*.

#### 4. Proses *Testing* ELM

Pada proses *testing* ELM, hal pertama yang dilakukan adalah menghitung matriks H dengan aktivasi sigmoid. Adapun matriks *hidden layer* dan matriks bias yang digunakan yaitu sama dengan matriks pada proses *training*, hanya saja data yang digunakan yaitu data *testing*. Hasil perhitungan H *testing* ELM terlihat pada Tabel 3.9. Berikut merupakan contoh perhitungan data pertama pada fitur umur.

$$H_{1,1} = \frac{1}{1 + \exp^{-(X_{testing} \cdot W_{hidden}^T + b)}} = \frac{1}{1 + \exp^{-((0,58 * 0,51) + \dots + (0,23 * 0,54) + 0,1)}} = 4,04$$

Tabel 3.9 Matriks H ELM *testing*

	1	2	3	4	5
H	4,04	3,24	3,24	2,87	3,32

Kemudian dilakukan perhitungan matriks *Moore Penrose* menggunakan Persamaan 2 sama seperti pada proses *training*. Hasil generalisasi *Moore Penrose* pada proses *testing* terlihat pada Tabel 3.10.

Tabel 3.10 Matriks *Moore Penrose* ELM *testing*

	1	2	3	4	5
<i>Moore Penrose</i>	4,04	3,24	3,24	2,87	3,32

Setelah itu dilakukan perhitungan matriks Y menggunakan persamaan 4. Matriks Y didapatkan dengan mengalikan matriks *Moore Penrose* yang didapatkan

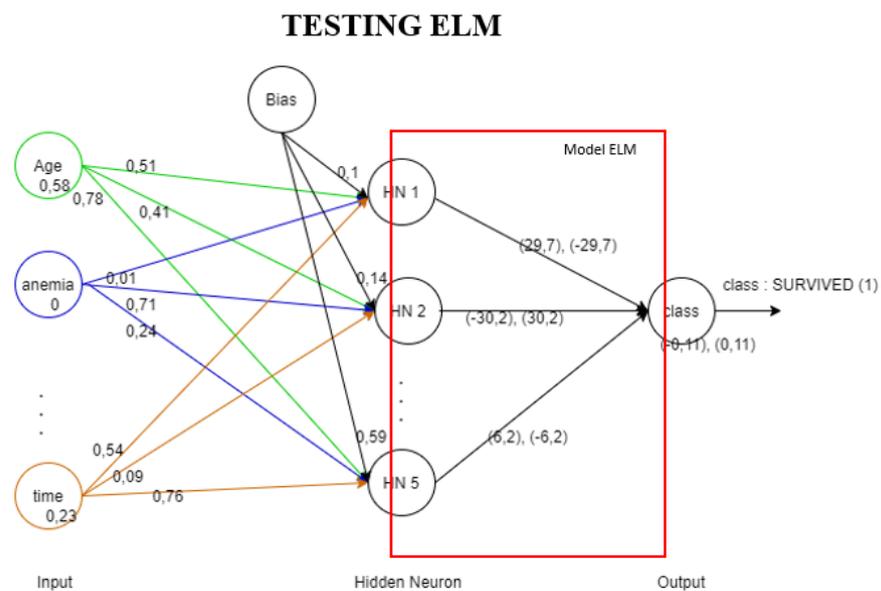
pada proses sebelumnya dengan matriks output ELM  $\beta$  hasil dari proses *training*.

Tabel 3.11. Ilustrasi proses *training* ELM dapat dilihat pada Gambar 3.7.

Tabel 3.11 Matriks Y ELM *testing*

	Kelas 0	Kelas 1
1	-0,11	0,11

Dari hasil perhitungan matriks Y, didapatkan bahwa data testing terklasifikasikan pada kelas 1, karena nilai yang lebih besar diantara matriks ada pada kelas 1. Hasil perhitungan pun sesuai dengan kelas sebenarnya.



Gambar 3.7 Ilustrasi *Neural Network* proses *Testing* ELM

Ilustrasi proses *testing* ELM pada Gambar 3.7 terdiri dari 3 bagian yaitu *input*, *hidden neuron*, dan *output*. Pada proses *testing*, input yang digunakan merupakan data *testing* (Tabel 3.2 No 7). Adapun bobot bias dan bobot *hidden neuron* yang

digunakan yaitu sama dengan yang digunakan pada proses *training*. Untuk mendapatkan hasil klasifikasi maka dibutuhkan matriks *output* ELM  $\beta$  yang merupakan model yang dihasilkan di proses *training* (Tabel 3.8). Sehingga hasil perhitungan yang didapatkan yaitu data *testing* diklasifikasikan ke dalam kelas 1 atau *survived*.