

Lampiran 1

Perhitungan Manual Backpropagation

Normalisasi :

1. Menentukan variable data

Contoh Data 4 variabel dari bulan 1 trafik kendaraan :

No	Parameter			
	Minggu 1	Minggu 2	Minggu 3	Minggu 4
1	25343	24537	24408	23219
2	22341	23271	21965	23859
3	22081	23056	22965	23061
4	26882	23349	23375	23107
5	39691	34276	35333	38537

2. Selanjutnya menentukan nilai *min* dan *maks* dari data masukan yang berjumlah 5 data

Nilai *min* dan *maks* dari data trafik kendaraan :

No	Parameter			
	Minggu 1	Minggu 2	Minggu 3	Minggu 4
Min	22081	23056	21965	23061
Maks	39691	34276	35333	38537

3. Selanjutnya lakukan normalisasi data dengan menggunakan persamaan :

$$\text{Normalisasi} = \frac{\text{Data Asli} - \text{Min}}{\text{Maks} - \text{Min}}$$

Data yang telah di normalisasi :

No	Parameter			
	Minggu 1	Minggu 2	Minggu 3	Minggu 4
1	0.814764	0.868003	0.817250	0.98979
2	0.985235	0.980837	1	0.94843
3	1	1	0.925194	1
4	0.727370	0.973885	0.894524	0.99702
5	0	0	0	0

Backpropagation :

1. Inisialisasi bobot V_{ij} dan bobot W_{jk} dengan bilangan acak kecil

Inisialisasi bobot V_{ij} dari *input layer* ke *hidden layer* :

	V0	V1	V2	V3	V4
1	1	0.2	0.3	0.2	0.1
2	1	0.1	0.1	0.3	0.3
3	1	0.1	0.3	0.1	0.2
4	1	0.2	0.3	0.2	0.2

Inisialisasi bobot W_{jk} dari *hidden layer* ke *output layer* :

	Y
Z1	0.3
Z2	0.2
Z3	0.1
Z4	0.1
Z0	1

2. Menentukan Maksimum epoch, *minimum error*
3. Jika kondisi berhenti belum terpenuhi, ulangi langkah 4
4. Epoch = Epoch + 1

Fase *Feedforward* :

5. Tiap unit masukkan menerima sinyal dan diteruskan ke unit *hidden*
6. Hitung semua keluaran di unit *hidden* (Z_j) :

$$Z_{net_j} = X_0 \sum_{i=1}^n X_i V_{ij}$$

$$\begin{aligned} Z_{net_1} &= 1 \times (0.814 + 0,2) \times (0.868 + 0,3) \times (0.817 + 0,2) \times (0.989 + 0,1) \\ &= 1,311 \end{aligned}$$

$$\begin{aligned} Z_{net_2} &= 1 \times (0.814 + 0,1) \times (0.868 + 0,1) \times (0.817 + 0,3) \times (0.989 + 0,3) \\ &= 1,273 \end{aligned}$$

$$\begin{aligned} Z_{net_3} &= 1 \times (0.814 + 0,1) \times (0.868 + 0,3) \times (0.817 + 0,1) \times (0.989 + 0,2) \\ &= 1,163 \end{aligned}$$

$$\begin{aligned} Z_{net_4} &= 1 \times (0.814 + 0,2) \times (0.868 + 0,3) \times (0.817 + 0,2) \times (0.989 + 0,2) \\ &= 1,432 \end{aligned}$$

Hasil keluaran di *hidden layer* :

No	Z_{net_1}	Z_{net_2}	Z_{net_3}	Z_{net_4}
1	1.3	1.2	1.1	1.4

7. Hitung keluaran di *hidden layer* menggunakan fungsi aktivasi :

$$Z_j = \frac{1}{1 + \exp(-Z_{net_j})}$$

$$Z_1 = \frac{1}{1 + \exp(-1,3)} = 0,78583498$$

$$Z_2 = \frac{1}{1 + \exp(-1,2)} = 0,76852478$$

$$Z_3 = \frac{1}{1 + \exp(-1,1)} = 0,75026011$$

$$Z_4 = \frac{1}{1 + \exp(-1,4)} = 0,80218389$$

Hasil keluaran di *hidden layer* menggunakan fungsi aktivasi :

Z1	0.78
Z2	0.76
Z3	0.75
Z4	0,80

8. Hitung semua jaringan di *output layer* (Y) :

$$Y_{net_k} = W_0 \sum_{i=1}^n Z_j W_{jk}$$

$$= 1 \times (0,78 + 0,3) \times (0,76 + 0,2) \times (0,75 + 0,1) \times (0,80 + 0,1)$$

$$= 0,793152$$

Hasil keluaran di *output layer* :

No	Y_{net_1}
1	0,79

9. Hitung keluaran di *output layer* menggunakan fungsi aktivasi :

$$Y_k = \frac{1}{1 + \exp(-Y_{net_k})}$$

$$Y_1 = \frac{1}{1 + \exp(-0,79)} = 0,68783133$$

Hasil keluaran di *output layer* menggunakan fungsi aktivasi :

Y1	0,68
----	------

Fase *Backward* :

10. Menghitung *error output layer* pada eksternal *input* :

$$\delta_k = (T_k - Y_k) f'(Y_{net_k})$$

$$\delta_1 = (T_k - Y_k) y_k (1 - Y_k)$$

$$\delta_1 = (1 - 0,68) 0,68 (1 - 0,68)$$

$$\delta_1 = 0,069632$$

11. Menghitung perbaikan bobok W_{jk} :

$$\Delta W_{jk} = \alpha \cdot \delta_k \cdot z_j$$

$$\Delta w_{11} = 0,2 \cdot (0,069632) \cdot 0,78 = 0,01086$$

$$\Delta w_{12} = 0,2 \cdot (0,069632) \cdot 0,76 = 0,01058$$

$$\Delta w_{13} = 0,2 \cdot (0,069632) \cdot 0,75 = 0,01044$$

$$\Delta w_{14} = 0,2 \cdot (0,069632) \cdot 0,80 = 0,01114$$

Hasil perbaikan bobot W_{jk} :

W	Δw_{jk}
ΔW_{11}	0,0108
ΔW_{12}	0,0105
ΔW_{13}	0,0104
ΔW_{14}	0,0111

12. Menghitung δ_{net_j} pada hidden layer :

$$\delta_{net_j} = \delta_k w_{jk}$$

$$\delta_{net_1} = 0,069632 \cdot 0,3 = 0,0208896$$

$$\delta_{net_2} = 0,069632 \cdot 0,2 = 0,0139264$$

$$\delta_{net_3} = 0,069632 \cdot 0,1 = 0,0069632$$

$$\delta_{net_4} = 0,069632 \cdot 0,1 = 0,0069632$$

Hasil *error output layer* pada *hidden layer* :

Z_j	δ_{net_j}
1	0,0208896
2	0,0139264
3	0,0069632
4	0,0069632

13. Menghitung *error* di *hidden layer* :

$$\delta_j = \delta_{net_j} \cdot Z_j (1 - Z_j)$$

$$\delta_1 = 0,0208896 \cdot 0,78 (1 - 0,78) = 0,00358$$

$$\delta_2 = 0,0139264 \cdot 0,76 (1 - 0,76) = 0,00254$$

$$\delta_3 = 0,0069632 \cdot 0,75 (1 - 0,75) = 0,00130$$

$$\delta_4 = 0,0069632 \cdot 0,80 (1 - 0,80) = 0,00111$$

Hasil bobot δ_j pada *hidden layer* :

Z_j	δ_j
1	0,00358
2	0,00254
3	0,00130
4	0,00111

14. Menghitung perbaikan bobot V_{ij} :

$$\Delta V_{ij} = \alpha \cdot \delta_j \cdot X_i$$

$$\Delta V_{10} = 0,2 \cdot 0,00358 \cdot 1 = 0,000716$$

$$\Delta V_{20} = 0,2 \cdot 0,00254 \cdot 1 = 0,000508$$

$$\Delta V_{30} = 0,2 \cdot 0,00130 \cdot 1 = 0,00026$$

$$\Delta V_{40} = 0,2 \cdot 0,00111 \cdot 1 = 0,000222$$

$$\Delta V_{11} = 0,2 \cdot 0,00358 \cdot 1 = 0,000716$$

$$\Delta V_{21} = 0,2 \cdot 0,00254 \cdot 1 = 0,000508$$

$$\Delta V_{31} = 0,2 \cdot 0,00130 \cdot 1 = 0,00026$$

$$\Delta V_{41} = 0,2 \cdot 0,00111 \cdot 1 = 0,000222$$

$$\Delta V_{12} = 0,2 \cdot 0,00358 \cdot 1 = 0,000716$$

$$\Delta V_{22} = 0,2 \cdot 0,00254 \cdot 1 = 0,000508$$

$$\Delta V_{32} = 0,2 \cdot 0,00130 \cdot 1 = 0,00026$$

$$\Delta V_{42} = 0,2 \cdot 0,00111 \cdot 1 = 0,000222$$

$$\Delta V_{13} = 0,2 \cdot 0,00358 \cdot 1 = 0,000716$$

$$\Delta V_{23} = 0,2 \cdot 0,00254 \cdot 1 = 0,000508$$

$$\Delta V_{33} = 0,2 \cdot 0,00130 \cdot 1 = 0,00026$$

$$\Delta V_{43} = 0,2 \cdot 0,00111 \cdot 1 = 0,000222$$

$$\Delta V_{14} = 0,2 \cdot 0,00358 \cdot 1 = 0,000716$$

$$\Delta V_{24} = 0,2 \cdot 0,00254 \cdot 1 = 0,000508$$

$$\Delta V_{34} = 0,2 \cdot 0,00130 \cdot 1 = 0,00026$$

$$\Delta V_{44} = 0,2 \cdot 0,00111 \cdot 1 = 0,000222$$

Hasil *update* bobot pada *input layer* di eksternal *output* :

	ΔV_0	ΔV_1	ΔV_2	ΔV_3	ΔV_4
1	0,000716	0,000716	0,000716	0,000716	0,000716
2	0,000508	0,000508	0,000508	0,000508	0,000508
3	0,00026	0,00026	0,00026	0,00026	0,00026
4	0,000222	0,000222	0,000222	0,000222	0,000222

14. *Update* semua bobot baru :

$$W_{jk} \text{ (Baru)} = W_{jk} \text{ (Lama)} + \Delta W_{jk}$$

$$W_{11} \text{ (Baru)} = 0,3 + 0,0108 = 0,30108$$

$$W_{12} \text{ (Baru)} = 0,2 + 0,0105 = 0,20105$$

$$W_{13} \text{ (Baru)} = 0,1 + 0,0104 = 0,10104$$

$$W_{14} \text{ (Baru)} = 0,1 + 0,0111 = 0,10111$$

$$V_{ij} \text{ (Baru)} = V_{ij} \text{ (Lama)} + \Delta V_{ij}$$

$$V_{10} \text{ (Baru)} = 1 + 0,000716 = 1,000716$$

$$V_{20} \text{ (Baru)} = 1 + 0,000508 = 1,000508$$

$$V_{30} \text{ (Baru)} = 1 + 0,00026 = 1,00026$$

$$V_{40} (\text{Baru}) = 1 + 0,000222 = 1,000222$$

$$V_{11} (\text{Baru}) = 0,2 + 0,000716 = 0,2000716$$

$$V_{21} (\text{Baru}) = 0,1 + 0,000508 = 0,1000508$$

$$V_{31} (\text{Baru}) = 0,1 + 0,00026 = 0,100026$$

$$V_{41} (\text{Baru}) = 0,2 + 0,000222 = 0,200222$$

$$V_{12} (\text{Baru}) = 0,3 + 0,000716 = 0,300716$$

$$V_{22} (\text{Baru}) = 0,1 + 0,000508 = 0,100508$$

$$V_{32} (\text{Baru}) = 0,3 + 0,00026 = 0,30026$$

$$V_{42} (\text{Baru}) = 0,3 + 0,000222 = 0,300222$$

$$V_{13} (\text{Baru}) = 0,2 + 0,000716 = 0,200716$$

$$V_{23} (\text{Baru}) = 0,3 + 0,000508 = 0,300508$$

$$V_{33} (\text{Baru}) = 0,1 + 0,00026 = 0,100026$$

$$V_{44} (\text{Baru}) = 0,2 + 0,000222 = 0,200222$$

$$V_{14} (\text{Baru}) = 0,1 + 0,000716 = 0,100716$$

$$V_{24} (\text{Baru}) = 0,3 + 0,000508 = 0,300508$$

$$V_{34} (\text{Baru}) = 0,2 + 0,00026 = 0,200026$$

$$V_{44} (\text{Baru}) = 0,2 + 0,000222 = 0,200222$$

Lampiran 2

Kode Program

1. Kelas Pelatihan :

```
public class Pelatihan extends javax.swing.JFrame {

    public Pelatihan() {

        initComponents();

    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent
    evt) {

        // TODO add your handling code here:

        JFileChooser chooser = new JFileChooser();

        chooser.setCurrentDirectory(new java.io.File("."));

        chooser.setSelectedFile(new File(""));

        chooser.setFileSelectionMode(JFileChooser.FILES_ONLY);

        chooser.setAcceptAllFileFilterUsed(false); //mematikan akses

        Allfile pada laman input file

        FileNameExtensionFilter filter = new

        FileNameExtensionFilter("excel", "xlsx");

        chooser.addChoosableFileFilter(filter);

        if (chooser.showOpenDialog(this) ==

        JFileChooser.OPEN_DIALOG) {

            //do when open
```

```

String path_file = chooser.getSelectedFile().getAbsolutePath();

jTextField1.setText(path_file);

DokumenManager dm = new DokumenManager();

ArrayList<ArrayList> data = dm.get_data(path_file);

DefaultTableModel hasil = new DefaultTableModel(new
String[]{
    "Jalan",
    "Minggu 1",
    "Minggu 2",
    "Minggu 3"}, 0);

jTable1.setModel(hasil);

for (int i = 0; i < data.size(); i++) {
    hasil.addRow(new Object[]{ data.get(i).get(0),
data.get(i).get(0), data.get(i).get(1), data.get(i).get(2)});
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
    ProsesManager pm=new ProsesManager();
    pm.do_proses_pelatihan(jTextField1.getText(),
Integer.valueOf(epoch.getText()), Integer.valueOf(eror.getText()));

```

```
JOptionPane.showMessageDialog(null, "Pelatihan Berhasil");  
}
```

2. Kelas Pengujian :

```
public class Pengujian extends javax.swing.JFrame {  
  
    public Pengujian() {  
  
        initComponents();  
  
    }  
  
    ArrayList<ArrayList> prediksi_kemacetan_asli;  
  
    private void jButton1ActionPerformed(java.awt.event.ActionEvent  
    evt) {  
  
        // TODO add your handling code here:  
  
        JFileChooser chooser = new JFileChooser();  
        chooser.setCurrentDirectory(new java.io.File("."));  
        chooser.setSelectedFile(new File(""));  
        chooser.setFileSelectionMode(JFileChooser.FILES_ONLY);  
        chooser.setAcceptAllFileFilterUsed(false); //mematikan akses  
        Allfile pada laman input file  
  
        FileNameExtensionFilter filter = new  
        FileNameExtensionFilter("excel", "xlsx");  
  
        chooser.addChoosableFileFilter(filter);
```

```

        if (chooser.showOpenDialog(this) ==
JFileChooser.OPEN_DIALOG) {

        //do when open

        String path_file = chooser.getSelectedFile().getAbsolutePath();

        jTextField1.setText(path_file);

        DokumenManager dm = new DokumenManager();

        ArrayList<ArrayList> prediksi_asli = dm.get_data(path_file);

        prediksi_kemacetan_asli = new ArrayList<>();

        for (int i = 0; i < prediksi_asli.size(); i++) {

                ArrayList<String> prediksi_kemacetan_asli_temp = new
ArrayList<>();

                for (int j = 0; j < prediksi_asli.get(i).size(); j++) {

                        if (Double.valueOf(prediksi_asli.get(i).get(j).toString()) >
20000) {

                                prediksi_kemacetan_asli_temp.add("Macet");

                        } else {

                                prediksi_kemacetan_asli_temp.add("Tidak Macet");

                        }

                }

                prediksi_kemacetan_asli.add(prediksi_kemacetan_asli_temp);

        }

        ArrayList<String> nama_jalan = new ArrayList<>();

        nama_jalan.add("Basuki Rahmat (Patal)");

```

```

nama_jalan.add("Basuki Rahmat (Polda)");
nama_jalan.add("A. Rivai (Charitas)");
nama_jalan.add("A. Rivai (Bukit)");
nama_jalan.add("Kol. H. Burlian (Polda)");
nama_jalan.add("Kol. H. Burlian (AAL)");
nama_jalan.add("Demang Lebar Daun (Polda)");
nama_jalan.add("Demang Lebar Daun (Bukit Besar)");
nama_jalan.add("Jendral Sudirman (RS Charitas)");
nama_jalan.add("Jendral Sudirman (Polda)");
nama_jalan.add("Veteran (RS. Charitas)");
nama_jalan.add("Veteran (Rajawali)");

DefaultTableModel hasil = new DefaultTableModel(new
String[]{
    "Jalan",
    "Minggu 1",
    "Minggu 2",
    "Minggu 3"}, 0);

jTable1.setModel(hasil);

for (int i = 0; i < prediksi_kemacetan_asli.size(); i++) {
    hasil.addRow(new Object[]{nama_jalan.get(i),
prediksi_kemacetan_asli.get(i).get(0),
prediksi_kemacetan_asli.get(i).get(1),
prediksi_kemacetan_asli.get(i).get(2)});

```

```
    }  
  }  
}
```

3. Kelas ProsesManager :

```
public class ProsesManager {  
    public static double round(float d, int decimalPlace) {  
        BigDecimal bd = new BigDecimal(Float.toString(d));  
        bd = bd.setScale(decimalPlace,  
BigDecimal.ROUND_HALF_UP);  
        return bd.doubleValue();  
    }  
    public void do_proses_pelatihan(String filepath, int epoch, double  
maximum_error) {  
        DokumenManager dm = new DokumenManager();  
        ArrayList<ArrayList> data = dm.get_data(filepath);  
        //min  
        ArrayList<Double> minimum = new ArrayList<>();  
        for (int i = 1; i < data.get(0).size(); i++) {  
            double minimum_temp =  
Double.valueOf(data.get(0).get(i).toString());  
            for (int j = 0; j < data.size(); j++) {
```

```

        if (minimun_temp >=
Double.valueOf(data.get(j).get(i).toString())) {
            minimun_temp =
Double.valueOf(data.get(j).get(i).toString());
        }
    }
    minimum.add(minimun_temp);
}

//maksimum
ArrayList<Double> maksimum = new ArrayList<>();
for (int i = 1; i < data.get(0).size(); i++) {
    double maksimum_temp =
Double.valueOf(data.get(0).get(i).toString());
    for (int j = 0; j < data.size(); j++) {
        if (maksimum_temp <=
Double.valueOf(data.get(j).get(i).toString())) {
            maksimum_temp =
Double.valueOf(data.get(j).get(i).toString());
        }
    }
    maksimum.add(maksimum_temp);
}

```



```

//normalisasi

ArrayList<ArrayList> normalisasi_data = new ArrayList<>();

for (int i = 0; i < data.size(); i++) {

    ArrayList<Double> data_temp = new ArrayList<>();

    for (int j = 1; j < data.get(i).size(); j++) {

        data_temp.add(

            (Double.parseDouble(data.get(i).get(j).toString()) -

minimum.get(j - 1))

            / (maksimum.get(j - 1) - minimum.get(j - 1))

        );

    }

    normalisasi_data.add(data_temp);

}

//backproagation

ArrayList<ArrayList> v = new ArrayList<>();

for (int i = 0; i < 3; i++) {

    ArrayList<Double> v_temp = new ArrayList<>();

    for (int j = 0; j < 3; j++) {

        Random rand = new Random();

        v_temp.add(round(rand.nextFloat(), 3));

    }

    v.add(v_temp);

}

```

```

ArrayList<Double> w = new ArrayList<>();

for (int i = 0; i < 3; i++) {

    Random rand = new Random();

    w.add(round(rand.nextFloat(), 3));

}

int temp_epoch = 0;

double error = 0.0;

while (temp_epoch < epoch && error < maximum_error) {

    ArrayList<ArrayList> data_digunakan = new ArrayList<>();

    for (int i = 0; i < normalisasi_data.size(); i++) {

        ArrayList<Double> data_digunakan_temp = new

ArrayList<>();

        for (int j = 0; j < 3; j++) {

data_digunakan_temp.add(round(Float.valueOf(normalisasi_data.get(i

).get(j).toString()), 5));

        }

        data_digunakan.add(data_digunakan_temp);

    }

    ArrayList<Double> b = new ArrayList<>();

    for (int i = 0; i < data_digunakan.size(); i++) {

        b.add(1.0);

    }

```

```

ArrayList<Double> t = new ArrayList<>();

for (int i = 0; i < data_digunakan.size(); i++) {
    t.add(1.0);
}

ArrayList<ArrayList> z_net = new ArrayList<>();

for (int i = 0; i < data_digunakan.size(); i++) {
    ArrayList<Double> z_net_temp = new ArrayList<>();
    for (int j = 0; j < v.size(); j++) {
        double z_temp = 0;
        for (int k = 0; k < data_digunakan.get(i).size(); k++) {
            z_temp += round((float)
Float.valueOf(v.get(j).get(k).toString()), 5)
                * round((float)
Float.valueOf(data_digunakan.get(i).get(k).toString()), 5);
        }
        z_net_temp.add(round((float) z_temp, 5));
    }
    z_net.add(z_net_temp);
}

System.out.println("znet");

System.out.println(z_net);

ArrayList<ArrayList> f_z_net = new ArrayList<>();

for (int i = 0; i < z_net.size(); i++) {

```

```

ArrayList<Double> f_z_net_temp = new ArrayList<>();
for (int j = 0; j < z_net.get(i).size(); j++) {
    double temp = 1 / (1
        + (Math.pow(2.71828183, -
(Double.valueOf(z_net.get(i).get(j).toString()))));
    f_z_net_temp.add(
        round((float) temp, 5)
    );
}
f_z_net.add(f_z_net_temp);
}
System.out.println("f z net");
System.out.println(f_z_net);
ArrayList<Double> y_net = new ArrayList<>();
for (int i = 0; i < f_z_net.size(); i++) {
    double y_temp = 0.0;
    for (int j = 0; j < f_z_net.get(i).size(); j++) {
        y_temp += round(Float.valueOf(w.get(j).toString()), 5)
            *
round(Float.valueOf(f_z_net.get(i).get(j).toString()), 5);
    }
    y_net.add(round((float) y_temp, 5));
}

```

```

System.out.println("y_net");

System.out.println(y_net);

ArrayList<Double> f_y_net = new ArrayList<>();

for (int i = 0; i < y_net.size(); i++) {

    double temp = 1 / round((float) (1

        + (Math.pow(2.71828183, -y_net.get(i)))), 5);

    f_y_net.add(

        round((float) temp, 5)

    );

}

System.out.println("f_y_net");

System.out.println(f_y_net);

ArrayList<Double> kesalahan_output = new ArrayList<>();

for (int i = 0; i < f_y_net.size(); i++) {

    double temp = (t.get(i) - f_y_net.get(i)) * f_y_net.get(i) * (1

- f_y_net.get(i));

    kesalahan_output.add(

        round((float) temp, 5)

    );

}

System.out.println("kesalahan_output");

System.out.println(kesalahan_output);

```

```

ArrayList<ArrayList> w_baru = new ArrayList<>();

for (int i = 0; i < f_z_net.size(); i++) {

    ArrayList<Double> w_baru_temp = new ArrayList<>();

    for (int j = 0; j < f_z_net.get(i).size(); j++) {

        double temp = 0.5 * kesalahan_output.get(i)

            * Double.parseDouble(f_z_net.get(i).get(j).toString()

                );

        w_baru_temp.add(

            round((float) temp, 5)

        );

    }

    w_baru.add(w_baru_temp);

}

System.out.println("w_baru");

System.out.println(w_baru);

ArrayList<ArrayList> eror_net = new ArrayList<>();

for (int i = 0; i < kesalahan_output.size(); i++) {

    ArrayList<Double> eror_net_temp = new ArrayList<>();

    for (int j = 0; j < w.size(); j++) {

        double temp = kesalahan_output.get(i)

            * w.get(j);

        eror_net_temp.add(

            round((float) temp, 5)

```

```

        );
    }
    eror_net.add(eror_net_temp);
}
System.out.println("eror_net");
System.out.println(eror_net);
ArrayList<ArrayList> kesalahan_lapisan = new ArrayList<>();
for (int i = 0; i < eror_net.size(); i++) {
    ArrayList<Double> kesalahan_lapisan_temp = new
ArrayList<>();
    for (int j = 0; j < eror_net.get(i).size(); j++) {
        double temp =
Double.parseDouble(eror_net.get(i).get(j).toString())
            *
Double.parseDouble(f_z_net.get(i).get(j).toString())
            * (1 -
Double.parseDouble(f_z_net.get(i).get(j).toString()));
        kesalahan_lapisan_temp.add(
            round((float) temp, 5)
        );
    }
    kesalahan_lapisan.add(kesalahan_lapisan_temp);
}

```

```

System.out.println("kesalahan_lapisan");

System.out.println(kesalahan_lapisan);

ArrayList<ArrayList> perubahan_bobot = new ArrayList<>();

for (int i = 0; i < kesalahan_lapisan.size(); i++) {

    ArrayList<Double> perubahan_bobot_temp = new

ArrayList<>();

    for (int j = 0; j < kesalahan_lapisan.get(i).size(); j++) {

        perubahan_bobot_temp.add(round((float) (0.5

            *

Double.parseDouble(kesalahan_lapisan.get(i).get(j).toString())

                * 1), 5)

        );

    }

    for (int j = 0; j < kesalahan_lapisan.get(i).size(); j++) {

        perubahan_bobot_temp.add(round((float) (0.5

            *

Double.parseDouble(kesalahan_lapisan.get(i).get(j).toString())

                * 1), 5)

        );

    }

    for (int j = 0; j < kesalahan_lapisan.get(i).size(); j++) {

        perubahan_bobot_temp.add(round((float) (0.5

```



```

        *
        Double.parseDouble(kesalahan_lapisan.get(i).get(j).toString())
            * 1), 5)
    );
}
for (int j = 0; j < kesalahan_lapisan.get(i).size(); j++) {
    perubahan_bobot_temp.add(round((float) (0.5
        *
        Double.parseDouble(kesalahan_lapisan.get(i).get(j).toString())
            * 1), 5)
        );
}
perubahan_bobot.add(perubahan_bobot_temp);
}
System.out.println("perubahan_bobot");
System.out.println(perubahan_bobot);
ArrayList<ArrayList> update_bobot_w = new ArrayList<>();
for (int i = 0; i < w_baru.size(); i++) {
    ArrayList<Double> update_bobot_w_temp = new
ArrayList<>();
    for (int j = 0; j < w_baru.get(i).size(); j++) {
        double temp = w.get(j)

```

```

        *
        Double.parseDouble(w_baru.get(i).get(j).toString());

        update_bobot_w_temp.add(
            round((float) temp, 5)
        );
    }
    update_bobot_w.add(update_bobot_w_temp);
}

System.out.println("update_bobot_w");
System.out.println(update_bobot_w);

ArrayList<Double> sum_bobot_w = new ArrayList<>();
for (int i = 0; i < update_bobot_w.get(0).size(); i++) {
    double sum_temp = 0;
    for (int j = 0; j < update_bobot_w.size(); j++) {
        sum_temp +=
Double.valueOf(update_bobot_w.get(j).get(i).toString());
    }

    sum_temp = sum_temp / update_bobot_w.size();
    sum_bobot_w.add(round((float) sum_temp, 5));
}

ArrayList<Double> v_ij = new ArrayList<>();
v_ij.add(1.0);
v_ij.add(1.0);

```

```

v_ij.add(1.0);

for (int i = 0; i < 3; i++) {

    ArrayList<Double> v_temp = new ArrayList<>();

    for (int j = 0; j < 3; j++) {

        v_ij.add(Double.valueOf(v.get(j).get(i).toString()));

    }

}

System.out.println("v_ij");

System.out.println(v_ij);

ArrayList<ArrayList> update_bobot_v = new ArrayList<>();

for (int i = 0; i < perubahan_bobot.size(); i++) {

    ArrayList<Double> update_bobot_v_temp = new

ArrayList<>();

    for (int j = 0; j < perubahan_bobot.get(i).size(); j++) {

        double temp =

Double.parseDouble(perubahan_bobot.get(i).get(j).toString())

        * v_ij.get(j);

        temp=round((float)temp,5);

        update_bobot_v_temp.add(

            round((float) temp, 5)

        );

    }

    update_bobot_v.add(update_bobot_v_temp);

```

```

    }

    System.out.println("update_bobot_v");

    System.out.println(update_bobot_v);

    ArrayList<Double> sum_bobot_v = new ArrayList<>();

    for (int i = 0; i < update_bobot_v.get(0).size(); i++) {

        double sum_temp = 0;

        for (int j = 0; j < update_bobot_v.size(); j++) {

            sum_temp +=

Double.valueOf(update_bobot_v.get(j).get(i).toString());

        }

        sum_temp = sum_temp / update_bobot_v.size();

        sum_bobot_v.add(round((float) sum_temp, 5));

    }

    System.out.println(sum_bobot_v.size());

    int langkah = 0;

    ArrayList<ArrayList> bobot_v_new = new ArrayList<>();

    for (int i = 0; i < 3; i++) {

        ArrayList<Double> bobot_v_new_temp = new

ArrayList<>();

        for (int j = 0; j < 4; j++) {

            bobot_v_new_temp.add(sum_bobot_v.get(langkah));

            langkah++;

        }

```

```
        bobot_v_new.add(bobot_v_new_temp);
    }

    v = new ArrayList<>(bobot_v_new);
    w = new ArrayList<>(sum_bobot_w);

    System.out.println("v");

    System.out.println(v);

    System.out.println("w");

    System.out.println(w);

    temp_epoch++;

    eror = 0.0;

    for (int i = 0; i < t.size(); i++) {
        eror += f_y_net.get(i) - t.get(i);
    }

    eror = Math.pow(eror, 2);

    eror = 0.5 * eror;

    System.out.println("eror");

    System.out.println(eror);
}

String simpan_V = "";

for (int i = 0; i < v.size(); i++) {
    for (int j = 0; j < v.get(i).size(); j++) {
        simpan_V += String.format("%.5f", v.get(i).get(j)) + "\t";
    }
}
```

```

        simpan_V += "\r\n";
    }

    String simpan_w = "";
    for (int j = 0; j < w.size(); j++) {
        simpan_w += String.format("%.5f", w.get(j)) + "\r\n";
    }

    //end latihan

    String file_w = "wt.txt";
    dm.save_arraylist(file_w, simpan_w);

    String file_v = "v.txt";
    dm.save_arraylist(file_v, simpan_V);

    //uji bp
}

```

4. Kelas DokumenManager :

```

public class DokumenManager {

    public ArrayList<ArrayList> get_data(String Path) {

        Dokumen DokumenManager = new Dokumen();

        ArrayList<ArrayList> data =

        DokumenManager.load_excel(Path);

        return data;

    }
}

```

```

public ArrayList<ArrayList> read_data_to_arraylist(String filepath)
throws IOException {

    ArrayList<ArrayList> data = new ArrayList<>();

    BufferedReader br = null;

    try {

        br = new BufferedReader(new FileReader(filepath));

        try {

            StringBuilder sb = new StringBuilder();

            String line = br.readLine();

            ArrayList<String> data_angka = new ArrayList<>();

            while (line != null) {

                if (!line.equals("")) {

                    String split_line[] = line.split("\\t");

                    data_angka = new ArrayList<>();

                    for (int i = 0; i < split_line.length; i++) {

                        data_angka.add(split_line[i]);

                    }

                    sb.append(line);

                    sb.append(System.lineSeparator());

                    line = br.readLine();

                    data.add(data_angka);

                }else{

                    line = br.readLine();

```

```
        }  
    }  
    } finally {  
        br.close();  
    }  
    } catch (FileNotFoundException ex) {  
Logger.getLogger(DokumenManager.class.getName()).log(Level.SEVERE,  
ERE, null, ex);  
    } finally {  
        try {  
            br.close();  
        } catch (IOException ex) {  
Logger.getLogger(DokumenManager.class.getName()).log(Level.SEVERE,  
ERE, null, ex);  
        }  
    }  
    return data;  
}
```