

**LAPORAN FINAL PROJECT UAS  
KECERDASAN BUATAN**

**"Implementasi Jaringan Syaraf Tiruan Perceptron pada Aplikasi Gerbang Logika OR"**



**Dosen Pengampu :**

Yisti Vita Via, S.ST. M.Kom.  
NIP : 19860425 2021212 001

**Oleh :**

Mohammad Faris Al Fatih (22081010277)

**PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"  
JAWA TIMUR  
2024**



**MATA KULIAH KECERDASAN BUATAN  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**KATA PENGANTAR**

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan laporan dengan judul "Implementasi Jaringan Syaraf Tiruan Perceptron pada Aplikasi Gerbang Logika OR". Laporan ini disusun sebagai salah satu tugas individu dalam mata kuliah Kecerdasan Buatan di UPN Veteran Jawa Timur.

Laporan ini menjelaskan secara rinci tentang implementasi algoritma Perceptron, salah satu jenis jaringan syaraf tiruan sederhana, untuk memodelkan gerbang logika OR. Dalam penyusunan laporan ini, saya menggunakan teknologi web berupa HTML, CSS, dan JavaScript untuk mengembangkan aplikasi yang mampu melakukan pelatihan dan visualisasi hasil dari proses pembelajaran Perceptron.

Saya menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, saya sangat mengharapkan saran dan kritik yang membangun dari berbagai pihak untuk perbaikan dan penyempurnaan di masa yang akan datang.

Saya mengucapkan terima kasih kepada Ibu Yisti Vita Via, S.ST., M.Kom. selaku dosen pengampu mata kuliah Kecerdasan Buatan yang telah memberikan bimbingan dan dukungan selama proses pembelajaran. Tidak lupa, saya juga berterima kasih kepada teman-teman yang telah memberikan motivasi dan dukungan.

Semoga laporan ini dapat bermanfaat bagi para pembaca dan memberikan kontribusi positif dalam pengembangan ilmu pengetahuan, khususnya di bidang kecerdasan buatan.

Surabaya, Juni 2024

Mohammad Faris Al Fatih

NPM 22081010277



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**DAFTAR ISI**

KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
BAB I.....	1
PENDAHULUAN .....	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah.....	1
1.3    Tujuan.....	1
1.4    Manfaat.....	1
BAB II.....	2
METODOLOGI.....	2
2.1    Deskripsi Aplikasi.....	2
2.2    Teknologi yang Digunakan .....	2
2.3    Algoritma Perceptron .....	2
2.4    Input dan Output.....	3
2.5    Implementasi Kode .....	3
2.6    Validasi dan Pengujian.....	3
BAB III .....	4
HASIL DAN PEMBAHASAN.....	4
3.1    Pembahasan Kode .....	4
3.2    Output dari Kode.....	6
3.3    Analisis Hasil .....	7
BAB IV .....	9
PENUTUP.....	9
4.1    Kesimpulan.....	9
DAFTAR PUSTAKA .....	10
LAMPIRAN.....	11



**MATA KULIAH KECERDASAN BUATAN  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**BAB I  
PENDAHULUAN**

**1.1 Latar Belakang**

Jaringan syaraf tiruan merupakan salah satu metode dalam kecerdasan buatan yang meniru cara kerja otak manusia dalam memproses informasi. Salah satu jenis jaringan syaraf tiruan yang paling sederhana adalah Perceptron, yang digunakan untuk tugas klasifikasi biner. Implementasi Perceptron dapat membantu dalam memahami dasar-dasar jaringan syaraf tiruan dan aplikasinya dalam pemecahan masalah sederhana, seperti gerbang logika OR.

**1.2 Rumusan Masalah**

Bagaimana cara mengimplementasikan algoritma Perceptron untuk memodelkan gerbang logika OR dalam sebuah aplikasi web?

**1.3 Tujuan**

Tujuan dari laporan ini adalah untuk mengimplementasikan algoritma Perceptron pada aplikasi gerbang logika OR dan memvisualisasikan proses pelatihan serta hasilnya.

**1.4 Manfaat**

- 1.4.1 Memahami konsep dasar algoritma Perceptron dan cara kerjanya.
- 1.4.2 Mengetahui cara mengembangkan aplikasi berbasis web untuk memodelkan jaringan syaraf tiruan sederhana.
- 1.4.3 Menyediakan alat bantu pembelajaran bagi mahasiswa dan peneliti dalam memahami proses pelatihan jaringan syaraf tiruan..



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**BAB II  
METODOLOGI**

**2.1 Deskripsi Aplikasi**

Aplikasi yang dikembangkan adalah sebuah alat bantu pembelajaran yang memvisualisasikan proses pelatihan jaringan syaraf tiruan Perceptron untuk memodelkan gerbang logika OR. Aplikasi ini memungkinkan pengguna untuk memasukkan nilai awal bobot dan bias, serta parameter pelatihan lainnya, dan kemudian melihat bagaimana nilai-nilai tersebut diperbarui selama pelatihan.

**2.2 Teknologi yang Digunakan**

Pengembangan aplikasi ini menggunakan teknologi web berikut:

- HTML: Untuk struktur halaman web.
- CSS: Untuk gaya dan tata letak halaman web.
- JavaScript: Untuk logika aplikasi, termasuk algoritma Perceptron dan manipulasi DOM.
- Bootstrap: Untuk desain responsif dan tata letak yang mudah.

**2.3 Algoritma Perceptron**

Algoritma Perceptron adalah algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi biner. Berikut adalah langkah-langkah utama dari algoritma ini:

- Inisialisasi: Tetapkan bobot awal  $w$  dan bias  $b$  dengan nilai acak atau nol.
- Forward Propagation: Hitung keluaran jaringan menggunakan fungsi aktivasi:

$$y_{in} = \sum_{i=1}^n w_i x_i + b$$
$$y = \begin{cases} 1 & \text{jika } y_{in} > 0 \\ 0 & \text{jika } y_{in} \leq 0 \end{cases}$$

- Pembaharuan Bobot: Perbarui bobot dan bias berdasarkan kesalahan  $e$  antara target  $t$  dan output  $y$ :

$$w_i \leftarrow w_i + \alpha e x_i$$



$$b \leftarrow b + \alpha e$$

- Ulangi: Ulangi langkah 2 dan 3 untuk setiap data pelatihan hingga error mencapai nol atau jumlah epoch maksimum tercapai.

## **2.4 Input dan Output**

- Input: Data masukan berupa pasangan nilai (x1, x2) untuk gerbang logika OR dan targetnya. Selain itu, nilai awal bobot (w1, w2), bias (wb), nilai ambang aktivasi, dan laju pembelajaran (learning rate) juga dimasukkan oleh pengguna.
- Output: Hasil pelatihan yang berupa tabel yang menunjukkan perubahan bobot dan bias pada setiap epoch hingga algoritma konvergen atau mencapai jumlah epoch maksimum.

## **2.5 Implementasi Kode**

Kode implementasi aplikasi ini ditulis dalam bahasa JavaScript dan HTML. Berikut adalah langkah-langkah utama implementasinya:

- Form Input: Pengguna mengisi form dengan nilai awal bobot, bias, nilai ambang, dan laju pembelajaran.
- Proses Pelatihan: Fungsi JavaScript mengambil nilai input, menjalankan algoritma Perceptron, dan memperbarui bobot dan bias hingga konvergen.
- Tabel Hasil: Hasil pelatihan ditampilkan dalam tabel yang menunjukkan perubahan bobot dan bias setiap epoch.

## **2.6 Validasi dan Pengujian**

Aplikasi ini diuji dengan data gerbang logika OR yang terdiri dari empat kombinasi input (0,0), (0,1), (1,0), dan (1,1) dengan target output masing-masing 0, 1, 1, dan 1. Pengujian dilakukan untuk memastikan bahwa Perceptron mampu belajar dan mencapai hasil yang diharapkan.



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**BAB III  
HASIL DAN PEMBAHASAN**

### **3.1 Pembahasan Kode**

#### **3.1.1 Struktur HTML**

Kode HTML yang digunakan untuk membangun antarmuka aplikasi terdiri dari beberapa elemen utama:

- **Form Input:** Formulir untuk memasukkan nilai awal bobot ( $W_1$ ,  $W_2$ ), bias ( $W_b$ ), nilai ambang aktivasi, dan learning rate.
- **Tabel Hasil:** Tabel yang akan menampilkan hasil pelatihan, termasuk nilai input, target, bobot lama, bobot baru, dan keluaran setiap epoch.

#### **3.1.2 Fungsi JavaScript**

Beberapa fungsi JavaScript utama yang digunakan dalam aplikasi ini adalah:

- **thresholdBinary:** Fungsi aktivasi yang mengembalikan output biner (0 atau 1) berdasarkan nilai ambang.

```
function thresholdBinary(x) {  
    return x > 0 ? 1 : 0;  
}
```

- **forwardPropagation:** Fungsi yang menghitung keluaran jaringan berdasarkan input, bobot, dan bias.

```
function forwardPropagation(input, weight, bias) {  
    let net = bias;  
    for (let i = 0; i < input.length; ++i) {  
        net += input[i] * weight[i];  
    }  
    return net;  
}
```

- **trainPerceptron:** Fungsi utama yang melatih jaringan Perceptron menggunakan data input, target, bobot awal, bias, dan parameter pelatihan lainnya.

```
function trainPerceptron(  
    input,  
    weight,  
    target,  
    maxEpochs,  
    learning_rate,  
    bias,  
) {  
    let tableBody = document.getElementById('tableBody');  
    let html = '';  
  
    // Inisialisasi bobot dan bias  
    let initialWeight = [...weight];  
    let initialBias = bias;
```



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

```
// Inisialisasi bobot dan bias
let initialWeight = [...weight];
let initialBias = bias;

// Iterasi untuk pelatihan
let epoch = 0;
while (epoch < maxEpochs) {
  let total_error = 0.0;
  html += `<tr><td colspan="17"><strong>Epoch ${
    epoch + 1
  }</strong></td></tr>`;
  let allMatched = true; // Untuk mengecek apakah semua Y sama dengan target
  for (let i = 0; i < input.length; ++i) {
    let output, net;
    net = forwardPropagation(input[i], weight, bias);
    let activated_output = thresholdBinary(net);
    let error = target[i] - activated_output;

    // Pembaruan bobot dan bias
    for (let j = 0; j < input[i].length; ++j) {
      weight[j] += learning_rate * error * input[i][j];
    }
    bias += learning_rate * error;

    // Tampilkan hasil dalam tabel
    html += `<tr>`;
    html += `<td>${i + 1}</td>`;
    html += `<td>${input[i][0]}</td>`; // Kolom x1
    html += `<td>${input[i][1]}</td>`; // Kolom x2
    html += `<td>${bias.toFixed(2)}</td>`; // Kolom b
    html += `<td>${target[i]}</td>`;
    if (epoch === 0) {
      html += `<td>0.00</td>`; // Kolom w1 lama
      html += `<td>0.00</td>`; // Kolom w2 lama
      html += `<td>0.00</td>`; // Kolom wb lama
    } else {
      html += `<td>${initialWeight[0].toFixed(2)}</td>`; // Kolom w1 lama
      html += `<td>${initialWeight[1].toFixed(2)}</td>`; // Kolom w2 lama
      html += `<td>${initialBias.toFixed(2)}</td>`; // Kolom wb lama
    }
    html += `<td>${net.toFixed(2)}</td>`; // Kolom Y_in
    html += `<td>${activated_output}</td>`; // Kolom Y
    html += `<td>${
      activated_output === target[i] ? 'Ya' : 'Tidak'
    }</td>`; // Kolom Cek kondisi (Y=T?)
    html += `<td>${(learning_rate * error * input[i][0]).toFixed(
      2,
    )}</td>`; // Kolom dw1
    html += `<td>${(learning_rate * error * input[i][1]).toFixed(
      2,
    )}</td>`; // Kolom dw2
    html += `<td>${(learning_rate * error).toFixed(2)}</td>`; // Kolom dwb
    html += `<td>${weight[0].toFixed(2)}</td>`; // Kolom w1 baru
    html += `<td>${weight[1].toFixed(2)}</td>`; // Kolom w2 baru
    html += `<td>${bias.toFixed(2)}</td>`; // Kolom wb baru
    html += `</tr>`;

    total_error += error * error;

    // Periksa apakah semua Y sudah sama dengan target
    if (activated_output !== target[i]) {
      allMatched = false;
    }
  }

  // Jika total error adalah nol, berhenti pelatihan
  if (allMatched) {
    html += `<tr><td colspan="17">Karena semua nilai Y (data 1-${input.length}) sama dengan semua T (data 1-${input.length}), maka berhenti pelatihan.</td></tr>`;
    break;
  }
  epoch++;
}

// Jika mencapai maksimum epochs
if (epoch === maxEpochs) {
  html += `<tr><td colspan="17">Mencapai maksimum iterasi (${maxEpochs}) tanpa konvergensi.</td></tr>`;
}

// Masukkan HTML ke dalam tabel
tableBody.innerHTML = html;
}
```

### 3.1.3 Proses Pelatihan

Fungsi startTraining menginisialisasi nilai-nilai dari form, memanggil fungsi trainPerceptron, dan menampilkan hasil pelatihan dalam tabel.





**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

```
function startTraining() {
  // Ambil nilai input dari form
  let initialW1 = parseFloat(document.getElementById('initialW1').value);
  let initialW2 = parseFloat(document.getElementById('initialW2').value);
  let initialWb = parseFloat(document.getElementById('initialWb').value);
  let activationThreshold = parseFloat(
    document.getElementById('activationThreshold').value,
  );
  let learningRate = parseFloat(
    document.getElementById('learningRate').value,
  );

  // Validasi input
  if (
    isNaN(initialW1) ||
    isNaN(initialW2) ||
    isNaN(initialWb) ||
    isNaN(activationThreshold) ||
    isNaN(learningRate)
  ) {
    alert('Harap masukkan nilai yang valid untuk semua input.');
```

```
    return;
  }

  // Input dan target untuk gerbang OR
  let input = [
    [0, 0],
    [0, 1],
    [1, 0],
    [1, 1],
  ];
  let target = [0, 1, 1, 1];

  // Bobot awal
  let weight = [initialW1, initialW2];
  let bias = initialWb;

  // Panggil fungsi untuk melatih perceptron
  trainPerceptron(input, weight, target, 1000, learningRate, bias);
}
```

### 3.2 Output dari Kode

Data ke	Input		Target	Bobot dan bias lama			Y <sub>in</sub>	Y	Cek kondisi (apakah Y=T?)	Perbaikan bobot			Bobot dan Bias baru			
	x1	x2		w1	w2	wb				dw1	dw2	dwb	w1	w2	wb	
1	0	0	0													
2	0	1	1													
3	1	0	1													
4	1	1	1													



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

Setelah pengguna memasukkan nilai awal bobot, bias, nilai ambang, dan learning rate, aplikasi akan melakukan pelatihan Perceptron dengan data input gerbang logika OR. Berikut adalah contoh output yang dihasilkan:

Masukkan nilai awal W1:

0

Masukkan nilai awal W2:

0

Masukkan nilai awal Wb:

0

Masukkan aktivasi/nilai ambang:

0.5

Masukkan learning rate (a):

0.8

Mulai Pelatihan

Data ke	Input			Target	Bobot dan bias lama			Y <sub>in</sub>	Y	Cek kondisi (apakah Y=T?)	Perbaikan bobot			Bobot dan Bias baru		
	x1	x2	b		w1	w2	wb				dw1	dw2	dwb	w1	w2	wb
Epoch 1																
1	0	0	0.00	0	0.00	0.00	0.00	0.00	0	Ya	0.00	0.00	0.00	0.00	0.00	0.00
2	0	1	0.80	1	0.00	0.00	0.00	0.00	0	Tidak	0.00	0.80	0.80	0.00	0.80	0.80
3	1	0	0.80	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.00	0.80	0.80
4	1	1	0.80	1	0.00	0.00	0.00	1.60	1	Ya	0.00	0.00	0.00	0.00	0.80	0.80
Epoch 2																
1	0	0	0.00	0	0.00	0.00	0.00	0.80	1	Tidak	0.00	0.00	-0.80	0.00	0.80	0.00
2	0	1	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.00	0.80	0.00
3	1	0	0.80	1	0.00	0.00	0.00	0.00	0	Tidak	0.80	0.00	0.80	0.80	0.80	0.80
4	1	1	0.80	1	0.00	0.00	0.00	2.40	1	Ya	0.00	0.00	0.00	0.80	0.80	0.80
Epoch 3																
1	0	0	0.00	0	0.00	0.00	0.00	0.80	1	Tidak	0.00	0.00	-0.80	0.80	0.80	0.00
2	0	1	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.00	0.80	0.00
3	1	0	0.80	1	0.00	0.00	0.00	0.00	0	Tidak	0.80	0.00	0.80	0.80	0.80	0.80
4	1	1	0.80	1	0.00	0.00	0.00	2.40	1	Ya	0.00	0.00	0.00	0.80	0.80	0.80
Epoch 4																
1	0	0	0.00	0	0.00	0.00	0.00	0.00	0	Ya	0.00	0.00	0.00	0.80	0.80	0.00
2	0	1	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
3	1	0	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
4	1	1	0.00	1	0.00	0.00	0.00	1.60	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00

Karena semua nilai Y (data 1-4) sama dengan semua T (data 1-4), maka berhenti pelatihan.

Masukkan nilai awal W1:

0

Masukkan nilai awal W2:

0

Masukkan nilai awal Wb:

0

Masukkan aktivasi/nilai ambang:

0.5

Masukkan learning rate (a):

0.8

Mulai Pelatihan

3	1	0	0.80	1	0.00	0.00	0.00	0.00	0	Tidak	0.80	0.00	0.80	0.80	0.80	0.80
4	1	1	0.80	1	0.00	0.00	0.00	2.40	1	Ya	0.00	0.00	0.00	0.80	0.80	0.80
Epoch 3																
1	0	0	0.00	0	0.00	0.00	0.00	0.80	1	Tidak	0.00	0.00	-0.80	0.80	0.80	0.00
2	0	1	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
3	1	0	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
4	1	1	0.00	1	0.00	0.00	0.00	1.60	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
Epoch 4																
1	0	0	0.00	0	0.00	0.00	0.00	0.00	0	Ya	0.00	0.00	0.00	0.80	0.80	0.00
2	0	1	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
3	1	0	0.00	1	0.00	0.00	0.00	0.80	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00
4	1	1	0.00	1	0.00	0.00	0.00	1.60	1	Ya	0.00	0.00	0.00	0.80	0.80	0.00

Pada epoch 4, semua nilai keluaran (Y) sesuai dengan target (T), sehingga pelatihan berhenti.

### 3.3 Analisis Hasil

Proses pelatihan jaringan Perceptron untuk memodelkan gerbang logika OR melalui beberapa epoch dapat dianalisis dari tabel hasil yang telah disajikan. Pada epoch pertama, terjadi perubahan bobot dan bias berdasarkan kesalahan antara



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

keluaran (Y) dengan target (T). Data pertama menunjukkan keluaran yang sesuai dengan target tanpa perbaikan bobot, sementara data kedua mengalami perubahan bobot dan bias setelah keluaran tidak sesuai. Pada data ketiga dan keempat, setelah pembaruan dari data kedua, keluaran sudah sesuai dengan target tanpa perbaikan tambahan.

Epoch kedua mengalami perubahan kembali karena kesalahan yang tersisa pada data pertama, sementara data lainnya tidak memerlukan perbaikan setelah pembaruan dari data sebelumnya. Pada epoch ketiga, proses pelatihan masih berlanjut dengan perubahan bobot dan bias karena kesalahan pada data pertama, sementara data lainnya menunjukkan keluaran yang sesuai dengan target tanpa perubahan lebih lanjut. Pada epoch keempat, semua data menunjukkan keluaran yang sesuai dengan target, mengindikasikan bahwa pelatihan telah konvergen.

Hasil ini menunjukkan bahwa algoritma Perceptron efektif dalam mempelajari dan memodelkan fungsi gerbang logika OR setelah beberapa epoch. Proses visualisasi pada tabel memberikan gambaran yang jelas tentang bagaimana bobot dan bias berubah selama pelatihan, memfasilitasi pemahaman tentang mekanisme kerja Perceptron dalam konteks aplikasi ini.



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**BAB IV  
PENUTUP**

**4.1 Kesimpulan**

Laporan ini mengimplementasikan algoritma Perceptron untuk memodelkan gerbang logika OR dalam konteks aplikasi web. Algoritma Perceptron dimulai dengan inisialisasi bobot dan bias awal, yang kemudian digunakan untuk menghitung output jaringan melalui proses forward propagation menggunakan fungsi aktivasi yang sesuai. Selanjutnya, bobot dan bias diperbarui berdasarkan kesalahan antara output yang dihasilkan dengan target yang diharapkan untuk setiap data pelatihan. Proses ini diulang secara iteratif hingga algoritma konvergen atau mencapai jumlah iterasi maksimum yang telah ditentukan. Aplikasi ini dikembangkan menggunakan teknologi web seperti JavaScript, HTML, CSS, dan Bootstrap, yang memungkinkan pengguna untuk mengatur nilai awal bobot, bias, serta parameter lainnya, dan memantau perubahan bobot serta bias melalui visualisasi dalam bentuk tabel selama proses pelatihan. Dengan demikian, implementasi Perceptron dalam aplikasi ini tidak hanya mengilustrasikan mekanisme dasar dari algoritma pembelajaran mesin tersebut, tetapi juga memberikan pemahaman praktis tentang penggunaannya dalam memodelkan fungsi klasifikasi sederhana seperti gerbang logika OR.



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**DAFTAR PUSTAKA**

- Sari, W. P. (2020). Kecerdasan Buatan: Pendekatan Komputasi Neural. Penerbit Andi.
- Sarwani, M. (2017). Pembelajaran Mesin: Konsep dan Implementasi. Penerbit Informatika.
- Budiono, M. A. (2016). Jaringan Syaraf Tiruan: Teori dan Aplikasi. Penerbit Graha Ilmu.
- Tika, G., & Adiwijaya. (2014). Klasifikasi Topik Berita Berbahasa Indonesia menggunakan Multilayer Perceptron. Jurnal Ilmiah, 7.
- Indrawati, R. (2018a). Penerapan Algoritma Perceptron pada Pola Penentuan Nilai Status Kelulusan Sidang Skripsi. Jurnal Aplikasi Ilmu Komputer (JAIK).
- Indrawati, R. (2018b). Pengenalan Pola Penyakit Hati Menggunakan Algoritma Kohonen dengan Metode Perceptron. Jurnal Aplikasi Ilmu Komputer (JAIK), 4.



**MATA KULIAH ANALISIS DAN DESIN SISTEM  
PROGRAM STUDI INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
JAWA TIMUR**

---

**LAMPIRAN**

Link Code : <https://github.com/farisalfatih/kcb/blob/main/index.html>

Link Video Penjelasan :

[https://www.youtube.com/watch?v=AAhndImMEPA&ab\\_channel=MohammadFarisAlFatih](https://www.youtube.com/watch?v=AAhndImMEPA&ab_channel=MohammadFarisAlFatih)