

Praktikum Sistem Cerdas



NRP	: 3223600019
Nama	: Muhammad Bimo Fachrizky
Materi	: Membuat Program Aplikasi Perceptron
Tanggal	: Senin, 10 Maret 2025

Praktikum 3

Membuat Program Aplikasi Perceptron

I. Tujuan Pembelajaran

- Mahasiswa dapat memahami dan menjelaskan konsep Perceptron
- Mahasiswa dapat menjelaskan model Perceptron
- Mahasiswa dapat membuat aplikasi Perceptron

Software yang di perlukan

- Microsoft Visual C++
- PyCharm

II. Langkah percobaan

1. Logika AND

AND

X ₁	X ₂	Y
0	0	0
0	1	0
1	0	0
1	1	1

```
#include <iostream>
#include <cstdlib>
#include <ctime>

double d_rand() {
    return (((double)(rand() % 32767) / 32767.0 - 0.5) * 2.0);
}

int main() {
    int i, j, out, ERR, y;
    int x[3][4] = {
        {0, 0, 1, 1},
        {0, 1, 0, 1},
        {1, 1, 1, 1}
    };
};
```

```

int T[4] = {0, 0, 0, 1};
float w[3], O, LR = 0.1, init = 0.15;

for (i = 0; i < 3; i++) {
    w[i] = init * d_rand();
}

for (i = 0; i < 10; i++) {
    for (j = 0; j < 4; j++) {
        O = x[0][j] * w[0] + x[1][j] * w[1] + x[2][j] * w[2];
        out = (O > 0.0) ? 1 : 0;
        ERR = T[j] - out;
        if (ERR != 0) {
            w[0] = w[0] + LR * x[0][j] * ERR;
            w[1] = w[1] + LR * x[1][j] * ERR;
            w[2] = w[2] + LR * x[2][j] * ERR;
        }
    }
    printf("Err:%d\n", ERR);
}

for (i = 0; i < 4; i++) {
    printf("Input X1:");
    scanf("%d", &x[0][0]);
    printf("Input X2:");
    scanf("%d", &x[1][0]);
    O = x[0][0] * w[0] + x[1][0] * w[1] + x[2][0] * w[2];
    y = (O > 0.0) ? 1 : 0;
    printf("Output:%d\n", y);
}
}

```

Hasil Program Pada Modul Praktikum

"D:\PENS\Tekkom\Sistem Cerdas\2020 Sistem Cerdas\Praktikum Sistem Cer

```
Err:1
Err:1
Err:1
Err:1
Err:1
Err:1
Err:1
Err:0
Err:0
Err:0
Err:0
Input X1:0
Input X2:0
Output:0
Input X1:0
Input X2:1
Output:0
Input X1:1
Input X2:0
Output:0
Input X1:1
Input X2:1
Output:1

Process returned 0 (0x0)   execution time : 10.503 s
Press any key to continue.
```

2. Logika OR

OR

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

```
#include <iostream>
#include <cstdlib>
#include <ctime>

double d_rand() {
    return (((double)(rand() % 32767) / 32767.0 - 0.5) * 2.0);
}

int main() {
    srand((unsigned)time(0)); // Seed the random number generator
```

```

int i, j, out, ERR, y;
int x[3][4] = {
    {0, 0, 1, 1},
    {0, 1, 0, 1},
    {1, 1, 1, 1}
};
int T[4] = {0, 1, 1, 1};
float w[3], O, LR = 0.1, init = 0.15;

// Initialize weights
for (i = 0; i < 3; i++) {
    w[i] = init * d_rand();
}

// Training
for (i = 0; i < 10; i++) {
    for (j = 0; j < 4; j++) {
        O = x[0][j] * w[0] + x[1][j] * w[1] + x[2][j] * w[2];
        out = (O > 0.0) ? 1 : 0;
        ERR = T[j] - out;
        if (ERR != 0) {
            w[0] = w[0] + LR * x[0][j] * ERR;
            w[1] = w[1] + LR * x[1][j] * ERR;
            w[2] = w[2] + LR * x[2][j] * ERR;
        }
    }
    std::cout << "Err: " << ERR << std::endl;
}

// Running
for (i = 0; i < 4; i++) {
    std::cout << "Input X1: ";
    std::cin >> x[0][0];
    std::cout << "Input X2: ";
    std::cin >> x[1][0];
    x[2][0] = 1; // Bias input
    O = x[0][0] * w[0] + x[1][0] * w[1] + x[2][0] * w[2];
    y = (O > 0.0) ? 1 : 0;
    std::cout << "Output: " << y << std::endl;
}
}

```

Hasil Percobaan Pada Modul Praktikum

```
"D:\PENS\Tekkom\Sistem Cerdas\2020 Sistem Cerdas\Praktikum Sistem Cer
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Err:0
Input X1:0
Input X2:0
Output:0
Input X1:0
Input X2:1
Output:1
Input X1:1
Input X2:0
Output:1
Input X1:1
Input X2:1
Output:1

Process returned 0 (0x0)   execution time : 10.299 s
Press any key to continue.
```

III. Hasil Percobaan

1. Logika AND

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 3>and
Err:1
Err:1
Err:1
Err:1
Err:1
Err:1
Err:1
Err:0
Err:0
Err:0
Err:0
Input X1:1
Input X2:0
Output:0
Input X1:0
Input X2:0
Output:0
Input X1:0
Input X2:1
Output:0
Input X1:1
Input X2:1
Output:1
```

2. Logika OR

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 3>or
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Input X1: 0
Input X2: 0
Output: 0
Input X1: 1
Input X2: 0
Output: 1
Input X1: 0
Input X2: 1
Output: 1
Input X1: 1
Input X2: 1
Output: 1
```

IV. Tugas

1. Buat program sederhana untuk kasus sederhana single perceptron AND dan OR
 - a. Single perceptron AND

```
#include <iostream>
#include <cstdlib>
#include <ctime>

double d_rand() {
    return (((double)(rand() % 32767) / 32767.0 - 0.5) * 2.0);
}

int main() {
    srand((unsigned)time(0));

    int i, j, out, ERR, y;
    int x[3][4] = {
        {0, 0, 1, 1},
        {0, 1, 0, 1},
        {1, 1, 1, 1}
    };
    int T[4] = {0, 0, 0, 1};
    float w[3], O, LR = 0.1, init = 0.15;
```

```

for (i = 0; i < 3; i++) {
    w[i] = init * d_rand();
}

for (i = 0; i < 10; i++) {
    for (j = 0; j < 4; j++) {
        O = x[0][j] * w[0] + x[1][j] * w[1] + x[2][j] * w[2];
        out = (O > 0.0) ? 1 : 0;
        ERR = T[j] - out;
        if (ERR != 0) {
            w[0] = w[0] + LR * x[0][j] * ERR;
            w[1] = w[1] + LR * x[1][j] * ERR;
            w[2] = w[2] + LR * x[2][j] * ERR;
        }
    }
    std::cout << "Err: " << ERR << std::endl;
}

for (i = 0; i < 4; i++) {
    std::cout << "Input X1: ";
    std::cin >> x[0][0];
    std::cout << "Input X2: ";
    std::cin >> x[1][0];
    x[2][0] = 1;
    O = x[0][0] * w[0] + x[1][0] * w[1] + x[2][0] * w[2];
    y = (O > 0.0) ? 1 : 0;
    std::cout << "Output: " << y << std::endl;
}

return 0;
}

```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 3>perAND.exe
Err: 1
Err: 1
Err: 1
Err: 1
Err: 1
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Input X1: 0
Input X2: 0
Output: 0
Input X1: 1
Input X2: 0
Output: 0
Input X1: 0
Input X2: 1
Output: 0
Input X1: 1
Input X2: 1
Output: 1
```

b. Single perceptron OR

```
#include <iostream>
#include <cstdlib>
#include <ctime>

double d_rand() {
    return (((double)(rand() % 32767) / 32767.0 - 0.5) * 2.0);
}

int main() {
    srand((unsigned)time(0));

    int i, j, out, ERR, y;
    int x[3][4] = {
        {0, 0, 1, 1},
        {0, 1, 0, 1},
        {1, 1, 1, 1}
    };

    int T[4] = {0, 1, 1, 1};
    float w[3], O, LR = 0.1, init = 0.15;
```

```

for (i = 0; i < 3; i++) {
    w[i] = init * d_rand();
}

for (i = 0; i < 10; i++) {
    for (j = 0; j < 4; j++) {
        O = x[0][j] * w[0] + x[1][j] * w[1] + x[2][j] * w[2];
        out = (O > 0.0) ? 1 : 0;
        ERR = T[j] - out;
        if (ERR != 0) {
            w[0] = w[0] + LR * x[0][j] * ERR;
            w[1] = w[1] + LR * x[1][j] * ERR;
            w[2] = w[2] + LR * x[2][j] * ERR;
        }
    }
    std::cout << "Err: " << ERR << std::endl;
}

for (i = 0; i < 4; i++) {
    std::cout << "Input X1: ";
    std::cin >> x[0][0];
    std::cout << "Input X2: ";
    std::cin >> x[1][0];
    x[2][0] = 1;
    O = x[0][0] * w[0] + x[1][0] * w[1] + x[2][0] * w[2];
    y = (O > 0.0) ? 1 : 0;
    std::cout << "Output: " << y << std::endl;
}

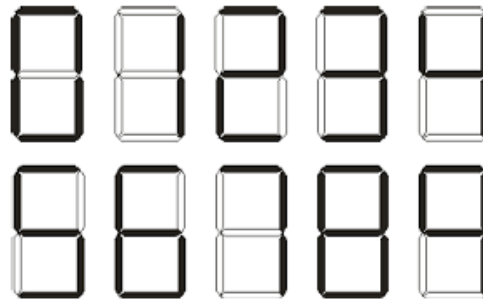
return 0;
}

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 3>perOR.exe
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Err: 0
Input X1: 0
Input X2: 0
Output: 0
Input X1: 0
Input X2: 1
Output: 1
Input X1: 1
Input X2: 0
Output: 1
Input X1: 1
Input X2: 1
Output: 1
```

2. Buat program untuk 7 segment input 7 output 10



```
#include<stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

double d_rand() {
    return(((double)(rand() % 32767) / 32767.0 - 0.5) * 2.0);
}

int main()
{
    char ch;
    int i, j, p, l, out[10], jum, ERR[10];
    int x[10][8] =
    { //a b c d e f g l
      { 1,1,1,1,1,1,0,1 },//0
      { 0,1,1,0,0,0,0,1 },//1
```

```

    { 1,1,0,1,1,0,1,1 },//2
    { 1,1,1,1,0,0,1,1 },//3
    { 0,1,1,0,0,1,1,1 },//4
    { 1,0,1,1,0,1,1,1 },//5
    { 0,0,1,1,1,1,1,1 },//6
    { 1,1,1,0,0,0,0,1 },//7
    { 1,1,1,1,1,1,1,1 },//8
    { 1,1,1,0,0,1,1,1 } //9
};
int T[10][10] =
{
    //0 1 2 3 4 5 6 7 8 9
    { 1,0,0,0,0,0,0,0,0,0 },
    { 0,1,0,0,0,0,0,0,0,0 },
    { 0,0,1,0,0,0,0,0,0,0 },
    { 0,0,0,1,0,0,0,0,0,0 },
    { 0,0,0,0,1,0,0,0,0,0 },
    { 0,0,0,0,0,1,0,0,0,0 },
    { 0,0,0,0,0,0,1,0,0,0 },
    { 0,0,0,0,0,0,0,1,0,0 },
    { 0,0,0,0,0,0,0,0,1,0 },
    { 0,0,0,0,0,0,0,0,0,1 }
};
float w[10][8], O[10], LR = 0.1f, init = 0.15f, error;
//inisialisasi bobot
for (j = 0; j < 10; j++)
{
    for (i = 0; i < 8; i++)
    {
        w[j][i] = init * d_rand();
        printf("%.2f ", w[j][i]);
    }
    printf("\n");
}
//training
for (l = 0; l < 100; l++)
{
    error = 0.0f; jum = 1;
    for (p = 0; p < 10; p++)
    {
        for (j = 0; j < 10; j++)
        {
            O[j] = 0.0;
            for (i = 0; i < 8; i++)

```

```

        {
            O[j] = O[j] + x[p][i] * w[j][i];
        }
        if (O[j] > 0.0)
            out[j] = 1;
        else
            out[j] = 0;
        ERR[j] = T[p][j] - out[j];
        error = fabs(error + ERR[j]);
        jum++;
        if (ERR[j] != 0)
        {
            for (i = 0; i < 8; i++)
            {
                w[j][i] = w[j][i] + LR * x[p][i] * ERR[j];
            }
        }
    }
}
error = error / jum;
printf("error=%.5f\n", error);
}
for (;;) {
    //running
    for (i = 0; i < 7; i++)
    {
        printf("Input X[%d]:", i);
        scanf_s("%d", &x[0][i]);
    }
    for (j = 0; j < 10; j++)
    {
        O[j] = 0.0;
        for (i = 0; i < 8; i++)
        {
            O[j] = O[j] + x[0][i] * w[j][i];
        }
        if (O[j] > 0.0)
            out[j] = 1;
        else
            out[j] = 0;
        printf("Output[%d]:%d\n", j, out[j]);
    }
}

```

[illegible]

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
E:\Program Files\Documents\Kuliah Semester 4\Praktikum Sistem Cerdas\praktikum 3>7segment.exe
-0.15 0.02 -0.09 0.09 0.03 -0.01 -0.04 0.12
0.10 0.07 -0.10 0.11 0.06 0.00 -0.06 -0.15
-0.12 -0.04 -0.11 -0.10 0.15 -0.02 -0.11 -0.15
-0.15 -0.04 0.01 0.02 0.03 0.03 -0.10 0.05
-0.01 -0.04 -0.13 0.03 0.08 0.09 0.01 -0.06
0.11 0.07 0.14 0.13 0.01 -0.11 -0.01 -0.08
0.11 -0.09 0.08 0.10 0.15 0.15 0.03 -0.03
-0.07 -0.06 0.10 -0.14 -0.04 -0.12 0.05 -0.13
-0.15 0.13 -0.07 -0.07 0.03 0.06 0.10 0.07
-0.00 -0.09 0.07 -0.01 -0.01 0.13 0.07 -0.12
error=0.01980
error=0.00990
error=0.00990
error=0.03960
error=0.01980
error=0.00990
error=0.00990
error=0.01980
error=0.01980
error=0.01980
error=0.00990
error=0.01980
error=0.01980
error=0.00990
error=0.01980
error=0.00990
error=0.00000
error=0.01980
error=0.01980
error=0.00990
error=0.01980
error=0.01980
error=0.00990
error=0.00000
error=0.00000
error=0.00000
Input X[0]:1
Input X[1]:1
Input X[2]:1
Input X[3]:1
Input X[4]:1
Input X[5]:1
Input X[6]:0
Output[0]:1
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Output: 0
```

```
Input X[0]:0
Input X[1]:1
Input X[2]:1
Input X[3]:0
Input X[4]:0
Input X[5]:0
Input X[6]:0
Output[0]:0
Output[1]:1
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Output: 1
```

```
Input X[0]:1
Input X[1]:1
Input X[2]:0
Input X[3]:1
Input X[4]:1
Input X[5]:0
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:1
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Output: 2
Input X[0]:
```

```
Input X[0]:1
Input X[1]:1
Input X[2]:1
Input X[3]:1
Input X[4]:0
Input X[5]:0
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:1
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Output: 3
Input X[0]:
```



```
Input X[0]:0
Input X[1]:1
Input X[2]:1
Input X[3]:0
Input X[4]:0
Input X[5]:1
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:1
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Output: 4
Input X[0]:
```

```
Input X[0]:1
Input X[1]:0
Input X[2]:1
Input X[3]:1
Input X[4]:0
Input X[5]:1
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:1
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Output: 5
Input X[0]:
```

```
Input X[0]:1
Input X[1]:0
Input X[2]:1
Input X[3]:1
Input X[4]:1
Input X[5]:1
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Input X[0]:
```

```
Input X[0]:1
Input X[1]:1
Input X[2]:1
Input X[3]:0
Input X[4]:0
Input X[5]:0
Input X[6]:0
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:1
Output[8]:0
Output[9]:0
Output: 7
Input X[0]:
```

```
Input X[0]:1
Input X[1]:1
Input X[2]:1
Input X[3]:1
Input X[4]:1
Input X[5]:1
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:1
Output[9]:0
Output: 8
Input X[0]:
```

```
Input X[0]:1
Input X[1]:1
Input X[2]:1
Input X[3]:1
Input X[4]:0
Input X[5]:1
Input X[6]:1
Output[0]:0
Output[1]:0
Output[2]:0
Output[3]:0
Output[4]:0
Output[5]:0
Output[6]:0
Output[7]:0
Output[8]:0
Output[9]:0
Input X[0]:
```

V. Analisa

Program pertama Logika AND menggunakan perceptron sederhana untuk mengenali operasi logika AND. Data input terdiri dari dua nilai biner (X_1 dan X_2), dengan bias tambahan ($X_3 = 1$). Target output T mengikuti tabel kebenaran AND. Bobot awal diinisialisasi secara acak menggunakan fungsi `d_rand()`, lalu diperbarui selama 10 iterasi dengan metode perceptron learning rule. Setiap iterasi menghitung output O , membandingkannya dengan target, dan memperbarui bobot jika terjadi kesalahan ($ERR \neq 0$).

Program kedua Logika OR hampir sama dengan perceptron logika AND, tetapi menggunakan target output T sesuai tabel kebenaran OR. Perbedaan utama adalah nilai target, di mana operasi OR menghasilkan 1 jika salah satu input bernilai 1. Proses pelatihan juga dilakukan dalam 10 iterasi dengan pembaruan bobot berdasarkan kesalahan output. Setelah pelatihan, program meminta input dari pengguna dan memprediksi outputnya.

Program ketiga / program tugas 1 AND adalah versi sederhana dari perceptron untuk operasi logika AND. Struktur dasarnya mirip dengan program pertama, tetapi menggunakan `std::cout` dan `std::cin` untuk menangani input dan output. Program juga menambahkan bias ($X_3 = 1$) untuk meningkatkan kinerja pembelajaran. Setelah 10 iterasi pelatihan, bobot yang diperoleh digunakan untuk memprediksi hasil dari input yang dimasukkan oleh pengguna.

Program keempat / program tugas 2 OR adalah versi sederhana dari perceptron logika OR, yang memiliki struktur serupa dengan versi AND single perceptron. Target output T mencerminkan operasi OR, dan bobot diperbarui selama 10 iterasi berdasarkan kesalahan output. Bias ($X_3 = 1$) juga digunakan untuk membantu pembelajaran. Setelah pelatihan, pengguna dapat memasukkan dua nilai input, dan program akan memprediksi hasilnya berdasarkan bobot yang telah dipelajari.

Program kelima adalah program untuk mengenali angka dari 0 hingga 9 berdasarkan pola 7-Segment display. Setiap angka direpresentasikan oleh 7 input biner yang menandakan nyala atau matinya segmen. Target output T adalah vektor satu-hot (one-hot encoding) yang merepresentasikan angka yang diinginkan. Bobot awal diinisialisasi secara acak, lalu diperbarui menggunakan perceptron learning rule. Proses pelatihan dilakukan hingga 100 iterasi untuk meminimalkan kesalahan (error). Setelah pelatihan, bobot yang diperoleh digunakan untuk mengklasifikasikan angka yang diberikan sebagai input.

VI. Kesimpulan

Kelima program ini menggunakan perceptron learning rule untuk mengenali pola logika dan angka. Program pertama dan kedua melatih perceptron untuk operasi logika AND dan OR dengan dua input biner dan satu bias. Bobot awal diinisialisasi

secara acak dan diperbarui selama 10 iterasi agar model dapat memprediksi hasil sesuai tabel kebenaran. Versi sederhana dari program AND dan OR memiliki cara kerja yang sama, tetapi menggunakan console input-output dengan `std::cout` dan `std::cin`. Bias tetap digunakan, dan bobot diperbarui dalam 10 iterasi sebelum digunakan untuk memprediksi input dari pengguna. Program kelima lebih kompleks karena mengenali angka 0–9 berdasarkan pola 7-Segment display. Angka direpresentasikan dalam format biner dengan one-hot encoding sebagai target output. Pelatihan dilakukan selama 100 iterasi agar model lebih akurat dalam mengklasifikasikan angka.