

Jawaban UAS Fisika Komputasi (FI-3201)

Semester 2 2019-2020

M Nauval FR 10217023

Steffan RS 10217038

Humam 10217088

Awla FA 10217027

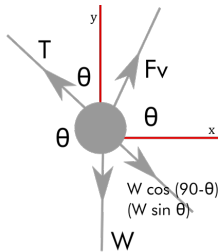
Hans MT 10217045

Wilbert I 10217088

May 1, 2020

Catatan: selain dibuat dalam format .tex dan pdf, jawaban ini juga tersedia dalam format .md

1 Soal No.1



- (a) Didapat persamaan dalam setiap arah gerak bandul sebagai berikut, untuk gerak sumbu x :

$$\Sigma F_x = m\ddot{x} \quad (1)$$

Pertama-tama, dengan melihat gambar dapat ditemukan hubungan

$$T \cos \theta - F_{visko_x} = m\ddot{x} \quad (2)$$

Jika dianalisa gaya searah \hat{r} ,

$$T - \omega \sin \theta = \frac{mv^2}{l} \quad (3)$$

sehingga

$$T = \omega \sin \theta + \frac{mv^2}{l} \quad (4)$$

dan

$$F_{visko_x} = 3\pi\eta D\dot{x}. \quad (5)$$

Jika disubstitusikan,

$$-mg \cos \theta \sin \theta + \frac{mv^2}{l} \cos \theta - 3\pi\eta D\dot{x} = m\ddot{x} \quad (6)$$

Substitusikan persamaan $\cos \theta = \frac{x}{l}$, $\sin \theta = \frac{y}{l}$, dan $w = -mg$, sehingga akan didapat

$$\ddot{x} + \frac{gxy}{l^2} + \frac{3\pi\eta D\dot{x}}{m} - \frac{v^2}{l^2}x = 0 \quad (7)$$

Selanjutnya, akan diturunkan persamaan gerak untuk sumbu y.

$$\Sigma F_y = m\ddot{y} \quad (8)$$

Dengan melihat gambar, dapat dilihat gaya yang bekerja di sumbu y adalah

$$T \sin \theta - mg - F_{visko_y} = m\ddot{y} \quad (9)$$

Substitusikan persamaan $F_{visko_y} = 3\pi\eta D\dot{y}$ dan $\sin \theta = \frac{y}{l}$, sehingga akan didapat

$$\ddot{y} + \frac{3\pi\eta D}{m}\dot{y} + \frac{(\dot{x}^2 + \dot{y}^2)y}{l^2} - \frac{gy^2}{l^2} = -g \quad (10)$$

(b) Untuk persamaan gaya di sumbu x,

$$\frac{g}{l^2}xy - \frac{3\pi\eta D}{m}\dot{x} - \frac{(\dot{x}^2 + \dot{y}^2)}{l^2}x = \ddot{x} \quad (11)$$

terdapat suku \ddot{x} sebagai komponen percepatan,, $\frac{3\pi\eta D}{m}\dot{x}$ sebagai komponen gaya gesek menggunakan hukum Stokes, $-\frac{(\dot{x}^2 + \dot{y}^2)}{l^2}x$ sebagai komponen gaya sentripetal untuk tegangan tali, dan $\frac{g}{l^2}xy$ adalah komponen gaya gravitasi di tegangan tali.

Untuk persamaan gaya di sumbu y,

$$\ddot{y} + \frac{3\pi\eta D}{m}\dot{y} + \frac{(\dot{x}^2 + \dot{y}^2)y}{l^2} - \frac{gy^2}{l^2} = -g \quad (12)$$

terdapat suku \ddot{y} sebagai komponen percepatan,, $\frac{3\pi\eta D}{m}\dot{y}$ sebagai komponen gaya gesek menggunakan hukum Stokes, $-\frac{(\dot{x}^2 + \dot{y}^2)y}{l^2}$ sebagai komponen gaya sentripetal untuk tegangan tali, dan $\frac{gy^2}{l^2}$ adalah komponen gaya gravitasi di tegangan tali. Selain itu, terdapat juga komponen $-g$ sebagai komponen gravitasi (vertikal sumbu y).

(c) Untuk benda jatuh bebas tanpa gesekan udara, maka persamaan (2) (gaya di sumbu x) berubah menjadi

$$\ddot{x} = 0 \quad (13)$$

Persamaan (3) (gaya di sumbu y) berubah menjadi

$$\ddot{y} = -g \quad (14)$$

Saat gaya jatuh bebas tanpa ada gesekan, komponen tegangan dan viskositas bisa diabaikan, sehingga $\eta = 0$ dan $l \approx \infty$.

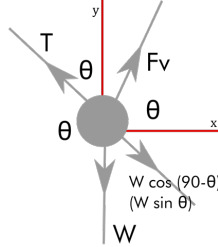
(d) Untuk simpangan kecil, maka $\frac{y}{l} \approx 1$ dan $\frac{x}{l} \approx \theta$. Tanpa gaya gesek, maka $\eta = 0$. Sehingga, didapat

$$\ddot{x} - \frac{(\dot{x}^2 + \dot{y}^2)}{l}\theta + g\theta = 0 \quad (15)$$

$$\ddot{y} - \frac{(\dot{x}^2 + \dot{y}^2)}{l} \theta + g = -g \quad (16)$$

2 Soal No.2

(a) Pertama diketahui posisi bandul sebagai berikut:



T adalah gaya tegang tali, F_v adalah gaya viskositas akibat udara, dan W adalah berat.

Persamaan gerak pada sumbu $\hat{\theta}$ dapat diturunkan dengan mengingat $\alpha = \frac{d^2\theta}{dt^2} = \frac{M}{I}$, dimana α adalah momentum sudut, M adalah momen, dan I adalah inersia.

Torsi ditentukan oleh proyeksi gaya ke arah tangensial:

$$M = -mgL \sin \theta \quad (17)$$

Momen inersia pendulum adalah momen inersia lingkaran:

$$I = mL^2 \quad (18)$$

Sehingga, persamaan gerak di sumbu $\hat{\theta}$ adalah, dengan mencoret suku-suku yang sama

$$\frac{d^2\theta}{dt^2} = \frac{M}{I} = \frac{g \sin \theta}{L} \quad (19)$$

$$\frac{d^2\theta}{dt^2} + \frac{g \sin \theta}{L} = 0 \quad (20)$$

Untuk persamaan gerak pada sumbu \hat{r} , berlaku

$$\omega \cos \theta + T = \frac{mv^2}{r} + m\dot{\theta}^2 R \quad (21)$$

$$-mg \cos \theta + T = m\dot{\theta}^2 l \quad (22)$$

$$\dot{\theta}^2 + \frac{g}{l} \cos \theta = \frac{T}{ml} \quad (23)$$

(b)

$$\frac{d^2\theta}{dt^2} - \frac{g\theta}{l} = 0 \quad (24)$$

Persamaan diferensial ini adalah persamaan diferensial karakteristik, sehingga dengan memisalkan $\lambda_{12} = \pm \sqrt{\frac{g}{l}}i$, sehingga:

$$\theta = C \exp \lambda_1 t + D \exp \lambda_2 t \quad (25)$$

$$\theta = C(\cos \sqrt{\frac{g}{l}}t + i \sin \sqrt{\frac{g}{l}}t) + D(\cos \sqrt{\frac{g}{l}}t - i \sin \sqrt{\frac{g}{l}}t) \quad (26)$$

$$\theta = (C + D) \cos \sqrt{\frac{g}{l}}t + i(C - D) \sin \sqrt{\frac{g}{l}}t \quad (27)$$

$$\theta = A \cos \sqrt{\frac{g}{l}}t + B \sin \sqrt{\frac{g}{l}}t \quad (28)$$

Saat $t = 0$, asumsikan bandul berada di amplitudo (θ_{max}) sehingga sudut maksimum.

$$\theta_{max} = A \cos 0 + 0 \Rightarrow \theta_{max} = A \quad (29)$$

Didapat $A =$ amplitudo, $B = 0$. Sehingga, pada akhirnya akan didapat

$$\theta = A \cos \sqrt{\frac{g}{l}}t, \dot{\theta} = -\sqrt{\frac{g}{l}}A \sin \sqrt{\frac{g}{l}}t \quad (30)$$

Untuk persamaan kedua, dengan mensubstitusikan persamaan sebelumnya,

$$\dot{\theta}^2 + \frac{g}{l} \cos \theta = \frac{T}{lm} \quad (31)$$

Karena menggunakan pendekatan sudut kecil, maka

$$T \approx \left(\left(\sqrt{\frac{g}{l}} A \sin \sqrt{\frac{g}{l}} t \right)^2 + \sqrt{\frac{g}{l}} \cos \theta \right) lm \quad (32)$$

$$T \approx A^2 g \sin^2 \left(\sqrt{\frac{g}{l}} t \right) + m \sqrt{gl} \cos \theta \quad (33)$$

Hasil dari perhitungan menggunakan metode analitik digrafikkan di

<https://plotly.com/~Avestory/3/#/>

(c) Dengan metode Euler,

$$\frac{df}{dx} = \frac{f(x+L) - f(x)}{h} \Rightarrow \frac{f^{i+1} - f^i}{\Delta x} \quad (34)$$

Persamaan $\ddot{\theta}$ dan $\dot{\theta}$ dapat dirubah menjadi

$$\dot{\theta} = \frac{d\theta}{dt} = \frac{\theta^{i+1} - \theta^i}{\Delta t} \Rightarrow \theta^{i+1} = \dot{\theta}^i \Delta t + \theta^i \quad (35)$$

dan

$$\ddot{\theta} = \frac{g}{l} \sin \theta \Rightarrow \frac{d\dot{\theta}^i}{dt} = \frac{g}{l} \sin \theta \quad (36)$$

$$\frac{\dot{\theta}^{i+1} - \dot{\theta}^i}{\Delta t} = \frac{g}{l} \sin \theta^i \quad (37)$$

$$\dot{\theta}^{i+1} = \frac{g}{l} \sin \theta^i \Delta t + \dot{\theta}^i \quad (38)$$

Hasil dari perhitungan menggunakan metode numerik digrafikkan di

<https://plotly.com/~Avestory/1/#/>

(d) Untuk menyelesaikan secara numerik, dibuat kode di C++ sebagai berikut:

```
#include <iostream>
#include <cmath>
#include <fstream>
```

```

using namespace std;

int main(){
/* variabel yang digunakan */
float g,l,h,m,tt;
int i,n;

g = 9.82;

/*besar panjang tali*/
l=1;
/*besar massa*/
m=0.5;
/*total waktu analisis*/
tt = 10;
/*step waktu yang digunakan*/
h = 0.1;

n = tt/h;
float theta[n],thetadot[n],f[n],T[n],t[n];

for (i=0;i<n;i++){
t[i] = 0;
f[i] = 0;
T[i] = 0;
theta[i] = 0;
thetadot[i] = 0;
}

/* pendefinisian awal */
theta[0] = 30*3.14/180;
thetadot[0] = 0;

/* logika euler dan input data yang didapat ke file theta.txt */
ofstream myfile;
myfile.open ("theta.txt");
for (i=0; i<=n; i++){
theta[i+1] = theta[i] + thetadot[i]*h;

```

```

f[i] = (g/l)*sin(theta[i+1])*-1;
thetadot[i+1] = thetadot[i] + f[i]*h;
T[i] = l*m*(thetadot[i]*thetadot[i] + g*cos(theta[i])/l);

t[i+1] = t[i] + h;

myfile << t[i] << " " << theta[i] << " " << thetadot[i] << " " << T[i]

cout << "t[" << i << "]: " << t[i] << "\n";
cout << "theta[" << i << "]: " << theta[i] << "\n";
cout << "thetadot[" << i << "]: " << thetadot[i] << "\n";
cout << "T[" << i << "]: " << T[i] << "\n" << "\n";
}
myfile.close();

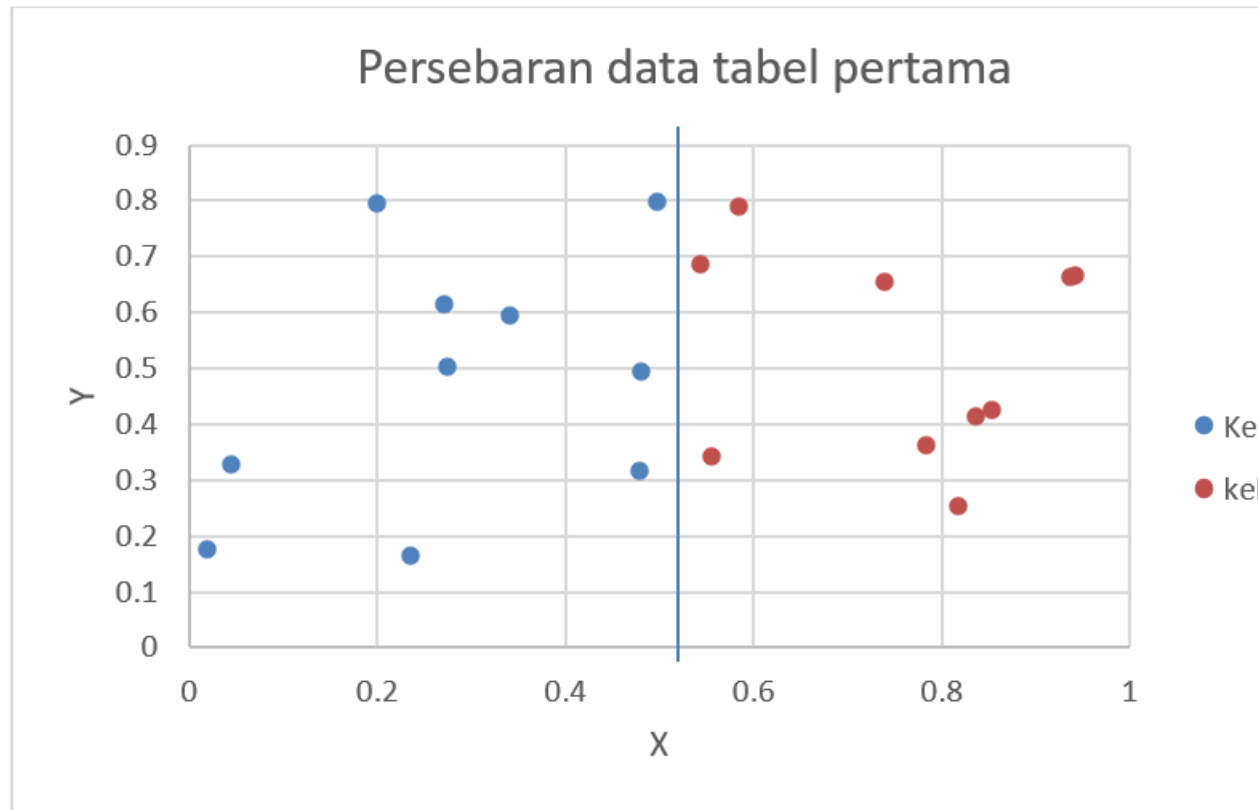
/* terminasi program */
return 0;
}

```

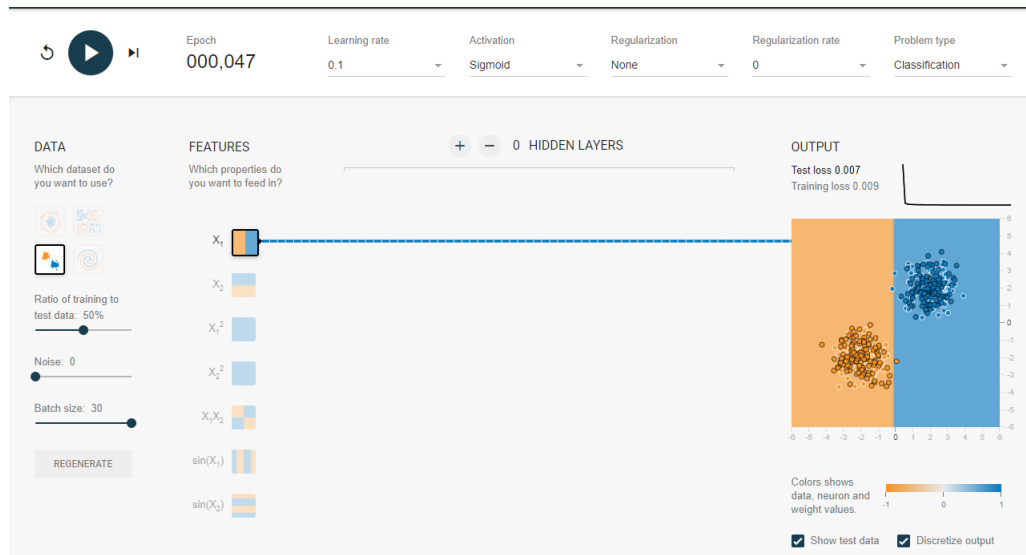
Kode juga akan disertakan di GitHub
(<https://github.com/mnauvalfr/uasfiskom/blob/master/kodeno2d.cpp>)
dan CPPSH (cpp.sh/2ybae)

3 Soal No.3

- (a) Dalam proses penyelesaian masalah, dataset yang telah diberikan dimasukkan terlebih dahulu ke dalam perangkat lunak Microsoft Excel untuk dianalisa bentuk persebaran datanya. Untuk tabel data pertama, terdapat persebaran data seperti berikut.

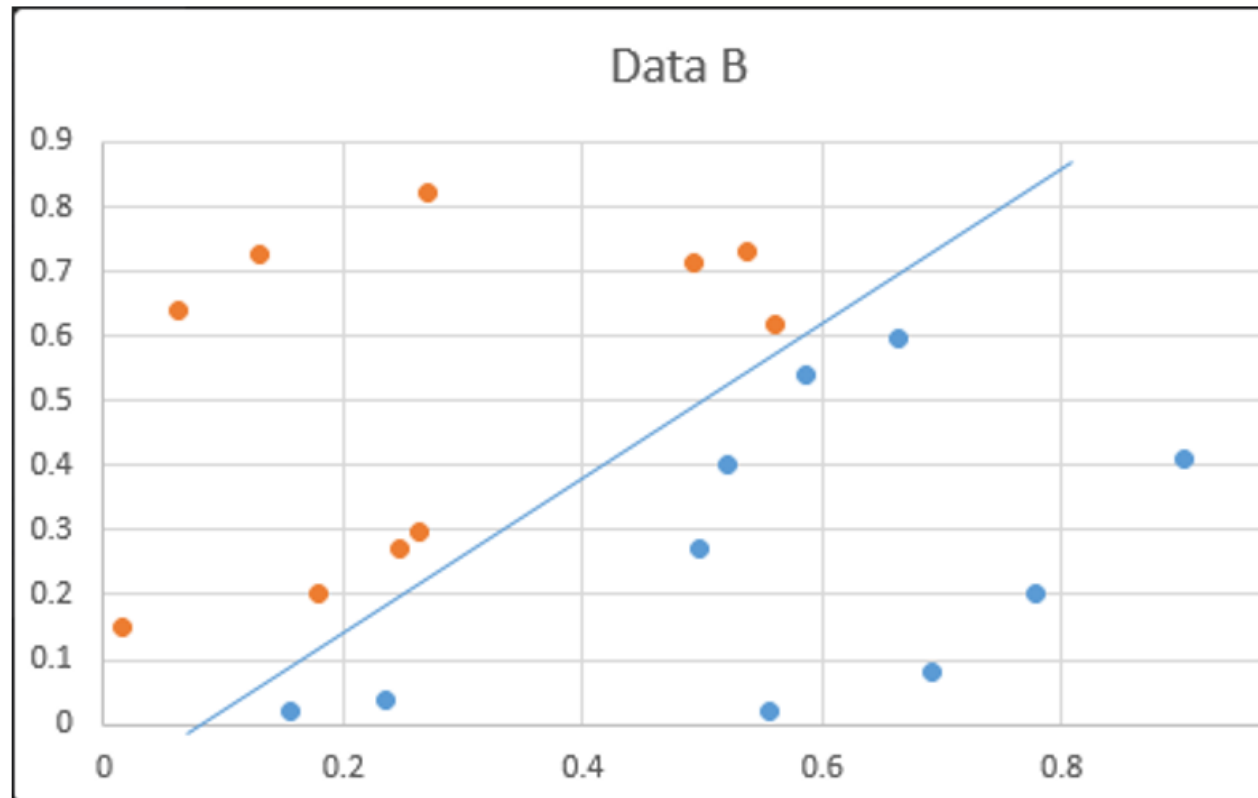


Berdasarkan persebaran data yang terdapat di tabel di atas, terlihat jelas bahwa data dapat dipisahkan dengan metode perceptron. Dengan data yang telah didapatkan, dilakukan uji coba di laman daring Artificial Neural Network Playground yang disediakan oleh Tensorflow. Ketentuan dan batasan uji coba yang terdapat adalah sebagai berikut.

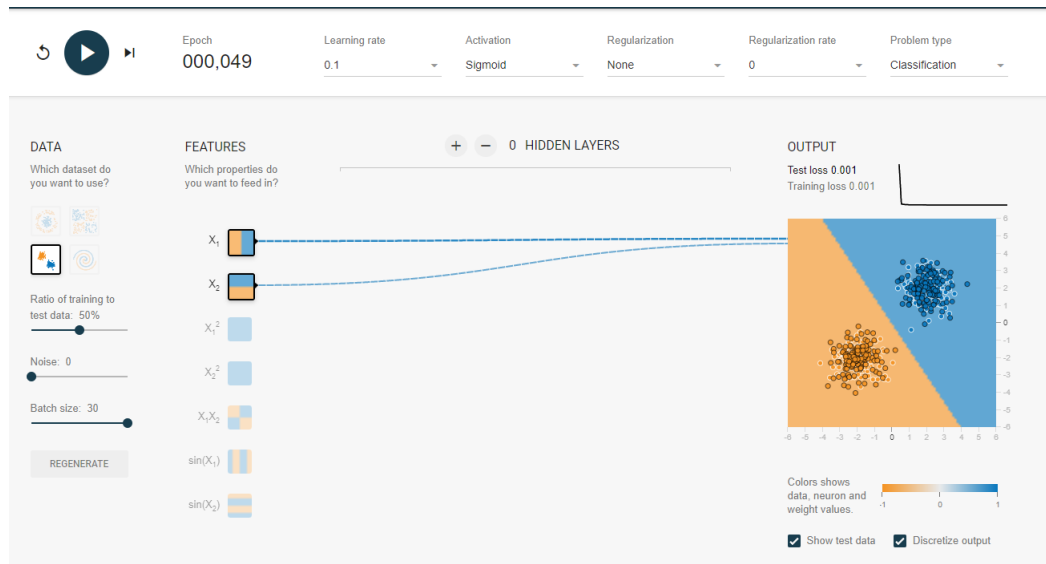


Berdasarkan data persebaran yang ada, dataset yang paling mirip dengan persebaran yang sudah dirumuskan adalah dataset yang dipilih. Ditentukan hanya satu neuron input karena dataset yang perlu dianalisa memiliki dua fitur di mana hanya satu fitur yang berkontribusi terhadap pembagian kelas ($N_1=1$). Hidden layer tidak digunakan pada uji coba ini karena berdasarkan persebaran data, data dapat dipisah menggunakan linear boundary. Tanpa hidden layer yang mengandung satu neuron, dapat dilihat bahwa hanya dalam 47 iterasi (epoch), data sudah secara rapi terpisah. Sehingga, arsitektur yang digunakan pada dataset ini adalah 1-0-1.

- (b) Dalam proses penyelesaian masalah, dataset yang telah diberikan dimasukkan terlebih dahulu ke dalam perangkat lunak Microsoft Excel untuk dianalisa bentuk persebaran datanya. Untuk tabel data pertama, terdapat persebaran data seperti berikut.

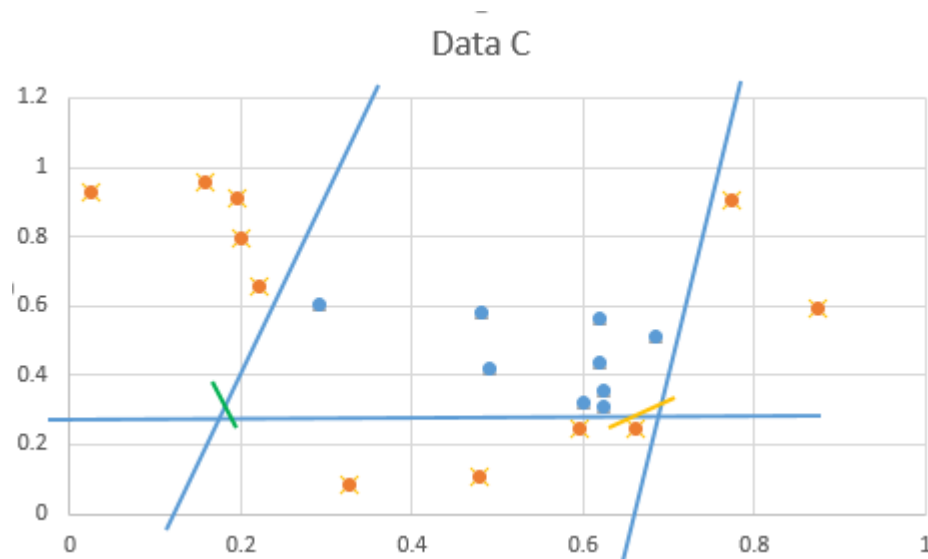


Berdasarkan persebaran data yang terdapat di tabel di atas, terlihat jelas bahwa data dapat dipisahkan dengan metode perceptron. Dengan data yang telah didapatkan, dilakukan uji coba di laman daring Artificial Neural Network Playground yang disediakan oleh Tensorflow. Ketentuan dan batasan uji coba yang terdapat adalah sebagai berikut.



Berdasarkan data persebaran yang ada, dataset yang paling mirip dengan persebaran yang sudah dirumuskan adalah dataset yang dipilih. Ditentukan dua neuron input karena dataset yang perlu dianalisa memiliki dua fitur ($N_1=2$). Hidden layer tidak digunakan pada uji coba ini karena berdasarkan persebaran data, data dapat dipisah menggunakan linear boundary. Tanpa hidden layer yang mengandung satu neuron, dapat dilihat bahwa hanya dalam 49 iterasi (epoch), data sudah secara rapi terpisah. Sehingga, arsitektur yang digunakan pada dataset ini adalah 2-0-1.

- (c) Untuk penentuan JST, akan lebih mudah jika terlebih dahulu divisualisasikan. Pertama, data yang diperoleh dimasukkan pada perangkat lunak Excel dan diurutkan outputnya agar terpisah data yang memberikan output kelas 0 dan 1. Kemudian, dibuat diagram scatter dengan warna output 1 dan 0 berbeda dengan Excel dan ditarik garis boundary layer agar bisa ditentukan berapa boundary layer yang dibutuhkan.



Dari gambar ini terlihat garis memiliki kemiringan artinya sumbu x dan sumbu y keduanya ikut berpengaruh terhadap hasil. Sehingga, bisa disimpulkan bahwa layer input yang digunakan adalah 2 yaitu fitur x dan y ($N1=2$). Dari gambar terlihat dibutuhkan minimal 3 boundary layer, jumlah boundary setara dengan jumlah neuron yang dibutuhkan hidden layer sehingga untuk hidden layer $N2$ dibutuhkan minimal 3 neuron untuk memisahkan. Agar training lebih cepat, untuk hidden layer $N2$, neuron dapat ditambah satu untuk menghaluskan boundary seperti diberikan oleh garis kuning sedangkan garis hijau diwakili oleh output layer. Tetapi, untuk kesederhanaan ANN maka sebenarnya cukup digunakan tiga neuron pada satu hidden layer ($N2=3$). Dari gambar, terlihat bahwa output yang diharapkan adalah pemisah yaitu 1 dan 0 sehingga bentuk output layer hanya memisahkan 2 data diskrit. Maka, output layer hanya memerlukan 1 yaitu pemisah 1 dan 0 ($N3 = 1$). Didapat $N1 - N2 - N3 = 2 - 3 - 1$.

- (d) Penjelasan sebenarnya sudah dijelaskan dari 3.a hingga 3.c., didapatkan dari visualisasi dataset pertama bahwa data terpisah secara jelas oleh satu garis sehingga diperlukan dua neuron input untuk dua fitur x dan y dan karena pemisah linear, maka tidak diperlukan hidden layer. Untuk dataset kedua, dipisahkan oleh garis miring sehingga kedua sumbu x dan y berpengaruh sehingga diperlukan dua neuron input. Tetapi, karena masih bisa dipisahkan boundary linear maka tidak diperlukan

hidden layer. Sedangkan, untuk data C, diperlukan dua input dan pemisah harus diwakili oleh 3 boundary sehingga diperlukan satu hidden layer dengan tiga neuron. Arsitektur JST yang sederhana penting karena semakin rumit arsitektur maka iterasi yang dilakukan semakin lama dan panjang serta resource yang diperlukan juga meningkat. Kita harus seefisien mungkin dalam mengalokasikan computational resource terutama jika yang akan diolah data yang besar.

4 Soal No.4

- (a) Kode yang dibuat adalah dengan input kromoson 00101110

```
main();
// Define main function
function main() {
  var p = "00101110";
  [xs, ys, cs] = getValues(p);
  console.log("p =",p);
  console.log("x =",xs);
  console.log("y =",ys);
  console.log("c =",cs);
}
function getValues() {
  var p = arguments[0];
  var xs = p.slice(0, 3);
  var ys = p.slice(3, 6);
  var cs = p.slice(6);

  return [xs, ys, cs];
}
```

Hasil yang diperoleh adalah sebagai berikut:

```
p = 0010110
```

```
x = 001
```

```
y = 011
```

```
c = 0
```

Kode yang sama dapat juga diakses di GitHub dan dapat dijalankan secara daring menggunakan jsconsole.com.

- (b) Fungsi *fitness* yang diminta dibuat dalam kode berikut, dengan $x_0=111$ dan $y_0=111$.

```
main();
// Define main function
function main() {
var p = "1010110";
[xs, ys, cs] = getValues(p);
var hasil = 1/(1+fitness(xs,ys)); //fungsi utuh fitting
console.log("p =",p);
console.log("x =",xs);
console.log("y =",ys);
console.log("c =",cs);
console.log("hasil = ",hasil);
}

function getValues() {
var p = arguments[0];
var xs = p.slice(0, 3);
var ys = p.slice(3, 6);
var cs = p.slice(6);
return [xs, ys, cs];
}
```

Kode yang sama dapat juga diakses di GitHub dan dapat dijalankan secara daring menggunakan jsconsole.com.

(c) Kode yang dibuat dengan nilai x_0 adalah 111 dan y_0 adalah 111.

```
main();

//
// Define main function
function main()
{ var p = "1010110";
  var q = "1011111";
  var r = "1111011";
  var s = "1110110";
  var t = "1111101";
  var threshold= 0.5;
  [x1, y1, c1] = getValues(p);
  [x2, y2, c2] = getValues(q);
  [x3, y3, c3] = getValues(r);
  [x4, y4, c4] = getValues(s);
  [x5, y5, c5] = getValues(t);
  var hasil1 = 1/(1+fitness(x1,y1));
  var seleksi1 = selection(threshold, hasil1, p)
  console.log("p =",p);
  console.log("x =",x1);
  console.log("y =",y1);
  console.log("c =",c1);
  console.log("hasil = ",hasil1);
  console.log("seleksi = ",p," ", seleksi1);
  var hasil2 = 1/(1+fitness(x2,y2));
  var seleksi2 = selection(threshold, hasil2, q)
  console.log("q =",q);
  console.log("x =",x2);
  console.log("y =",y2);
  console.log("c =",c2);
  console.log("hasil = ",hasil2);
  console.log("seleksi = ",q," ", seleksi2);
```



```

var hasil3 = 1/(1+fitness(x3,y3));
var seleksi3 = selection(threshold, hasil3, r)
console.log("r =",r);
console.log("x =",x3);
console.log("y =",y3);
console.log("c =",c3);
console.log("hasil = ",hasil3);
console.log("seleksi = ",r," ", seleksi3);
var hasil4 = 1/(1+fitness(x4,y4));
var seleksi4 = selection(threshold, hasil4, s)
console.log("s =",s);
console.log("x =",x4);
console.log("y =",y4);
console.log("c =",c4);
console.log("hasil = ",hasil4);
console.log("seleksi = ",s," ", seleksi4);
var hasil5 = 1/(1+fitness(x5,y5));
var seleksi5 = selection(threshold, hasil5, t)
console.log("t =",t);
console.log("x =",x5);
console.log("y =",y5);
console.log("c =",c5);
console.log("hasil = ",hasil5);
console.log("seleksi = ",t," ", seleksi5);
}

function getValues()
{ var p = arguments[0];
var xs = p.slice(0, 3);
var ys = p.slice(3, 6);
var cs = p.slice(6);
return [xs, ys, cs];
}

function fitness(a, b)
{ return(Math.sqrt(Math.pow((a - 111), 2) + Math.pow((b - 111),2)));
}

function selection(threshold, a, p)

```

```

{ if (a>=threshold) { result="pass the selection";
} else { result="didnt pass the selection";
}
return (result);
}

```

Hasil yang diperoleh adalah sebagai berikut:

`p = 1010110`

`x = 101`

`y = 011`

`c = 0`

`hasil = 0.009852337480068215`

`seleksi = 1010110 didnt pass the selection`

`q = 1011111`

`x = 101`

`y = 111`

`c = 1`

`hasil = 0.09090909090909091`

`seleksi = 1011111 didnt pass the selection`

```
seleksi = 1011111 didnt pass the selection
```

```
r = 1111011
```

```
x = 111
```

```
y = 101
```

```
c = 1
```

```
hasil = 0.09090909090909091
```

```
seleksi = 1111011 didnt pass the selection
```

```
s = 1110110
```

```
x = 111
```

```
y = 011
```

```
c = 0
```

```
hasil = 0.009900990099009901
```

```
t = 1111101
```

```
x = 111
```

```
y = 110
```

```
c = 1
```

```
hasil = 0.5
```

```
seleksi = 1111101    pass the selection
```

Kode yang sama dapat juga diakses di GitHub dan dapat dijalankan secara daring menggunakan jsconsole.com.

- (d) Kode yang dipakai sama dengan kode untuk nomor 4b, dengan $x_0 = 101$, $y_0 = 100$.

```
main();

// Define main function
function main() {
var p = "1011001";    //input kromoson yang diuji
[xs, ys, cs] = getValues(p);
var hasil = 1/(1+fitness(xs,ys));
console.log("p =",p);
console.log("x =",xs);
console.log("y =",ys);
console.log("c =",cs);
console.log("hasil = ",hasil);

}

function getValues() {
var p = arguments[0];
```

```

var xs = p.slice(0, 3);
var ys = p.slice(3, 6);
var cs = p.slice(6);

return [xs, ys, cs];
}

function fitness(a, b) {
return(Math.sqrt(Math.pow((a - 101), 2) + Math.pow((b - 100),2)));
//ganti nilai dalam akar untuk kromoson referensi
}

```

Kromoson yang akan di-check adalah

- 1111011 , x = 111 dan y = 101

p = 1111011

x = 111

y = 101

c = 1

hasil = 0.09049875621120891

- 1001111 , x = 100 dan y = 111

`p = 1001111`

`x = 100`

`y = 111`

`c = 1`

`hasil = 0.08301951253873768`

- 1001001, $x = 100$ dan $y = 100$
-

`p = 1001001`

`x = 100`

`y = 100`

`c = 1`

`hasil = 0.5`

- 1111111, $x = 111$ dan $y = 111$

`p = 1111111`

`x = 111`

`y = 111`

`c = 1`

`hasil = 0.06302758521508411`

- 1011001, x = 101 dan y = 100

`p = 1011001`

`x = 101`

`y = 100`

`c = 1`

`hasil = 1`

Hasil maksimal yang diperoleh adalah kromoson 1011001 karena sama dengan kromoson threshold dengan fitness 1 dan kromoson yang paling mendekati 1011001 adalah kromoson 1001001 dengan nilai fitness 0,5 dari 5 iterasi yang dicoba.

5 Soal No.5

1. Tujuan

Salah satu pemanfaatan terbaik untuk fisika komputasi adalah dalam analisis desain dan data reaktor nuklir Tujuan dari analisis desain dan data nuklir adalah menentukan performa sebuah reaktor nuklir ,yang ditentukan oleh beberapa variabel. Dalam RBL ini, yang dibahas hanya dua, yaitu fluks sumber neutron dan daya reaktor.

2. Rumusan Masalah

-Bagaimana grafik distribusi flux neutron suatu sumber reaktor silinder?

-Berapa daya reaktor?

3. Usulan Metode

Metode:

Untuk menentukan flux sumber, digunakan metode *finite difference* untuk menyelesaikan persamaan transport berikut,

$$\frac{1}{v} \frac{\partial \phi}{\partial t} = \nabla \cdot D(\mathbf{r}) \nabla \phi - \Sigma_a(\mathbf{r}) \phi(\mathbf{r}, t) + \nu \sum_{if} \phi / k_{eff} \quad (1)$$

$$\frac{1}{v} \frac{\partial \phi}{\partial t} = \nabla \cdot D(\mathbf{r}) \nabla \phi - \Sigma_a(\mathbf{r}) \phi(\mathbf{r}, t) + \nu \sum_{ij} \phi / k_{eff}$$

Digunakan kondisi stasioner sehingga turunan flux terhadap waktu bernilai 0. Maka, operator turunan orde 2 dan sigma bisa didiskritisasi sebagai berikut,

$$\begin{aligned}
\int_{x_i-(1/2)\Delta}^{x_i+(1/2)\Delta} dx \Sigma_a(x) \phi(x) &\simeq \Sigma_{ai} \phi_i \Delta \\
\int_{x_i-(1/2)\Delta}^{x_i+(1/2)\Delta} dx \left(\frac{d}{dx} D(x) \frac{d\phi}{dx} \right) &\simeq D \frac{d\phi}{dx} \Big|_{x_i+1/2\Delta} - D \frac{d\phi}{dx} \Big|_{x_i-1/2\Delta} \\
&\simeq \frac{1}{2} (D_i + D_{i+1}) \frac{\phi_{i+1} - \phi_i}{\Delta} \\
&\quad - \frac{1}{2} (D_{i-1} + D_i) \frac{\phi_i - \phi_{i-1}}{\Delta}
\end{aligned} \tag{2}$$

Sehingga, jika dimasukkan secara diskrit,

$$a_{i,i-1} \phi_{i-1} + a_{i,i} \phi_i + a_{i,i+1} \phi_{i+1} = \frac{1}{\lambda} f_i \phi_i \equiv S_i, \quad i = 1, \dots, I-1 \tag{3}$$

Bisa didapatkan parameter diskritisasi,

$$\begin{aligned}
a_{i,i-1} &= -\frac{1}{2} \left(\frac{D_i + D_{i-1}}{\Delta^2} \right) \left(1 - \frac{c}{2i-1} \right) \\
a_{i,i} &= \Sigma_{ai} + \frac{1}{2} \left(\frac{D_{i-1} + 2D_i + D_{i+1}}{\Delta^2} \right) \\
a_{i,i+1} &= -\frac{1}{2} \left(\frac{D_{i+1} + D_i}{\Delta^2} \right) \left(1 + \frac{c}{2i-1} \right) \\
f_i &= v \Sigma f_i
\end{aligned} \tag{4}$$

Dengan bentuk matriks A yang harus diselesaikan dan bisa dituliskan,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & \dots \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & \dots \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & \dots \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & \dots \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{N-1} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ \vdots \\ S_{N-1} \end{pmatrix} \tag{5}$$

Sehingga kemudian, bisa dilakukan update untuk estimasi eigenvalue

$$\begin{aligned}\lambda^{(1)} &= \frac{\sum_{i=1}^{I-1} v \Sigma_{fi} \phi_i^{(1)} \Delta}{\sum_{i=1}^{I-1} \Delta [a_{i,i-1} \phi_{i-1}^{(0)} + a_{i,i} \phi_i^{(0)} + a_{i,i+1} \phi_{i+1}^{(0)}]} \\ &\approx \frac{\sum_{i=1}^{I-1} v \Sigma_{fi} \phi_i^{(1)} \Delta}{\sum_{i=1}^{I-1} v \Sigma_{fi} \phi_i^{(0)} \Delta / \lambda^{(0)}}\end{aligned}\quad (6)$$

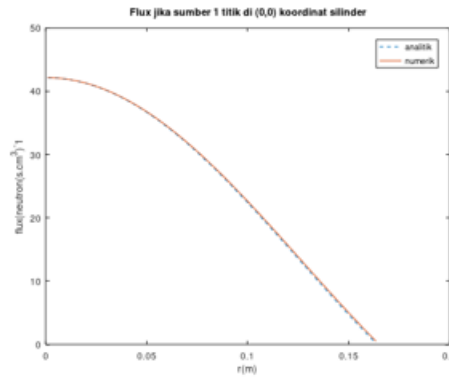
Yang kemudian, sumber fisi baru bisa dituliskan,

$$S_i^{(1)} = \frac{1}{\lambda^{(1)}} v \Sigma_{fi} \phi_i^{(1)} \quad (7)$$

Dan dilakukan looping hingga didapatkan vektor eigen fluks.

Perhitungan distribusi fluks dilakukan dengan menggunakan metode *finite difference* dengan *central-limit theorem*. Metode ini mengubah persamaan (1) menjadi (5) dengan penurunan (2) hingga (4). Selanjutnya, dilakukan penebakan keff awal dan fluks awal. Persamaan (5) dengan sumber pada persamaan (7) sejatinya adalah persamaan nilai eigen. Maka dilakukan metode pencarian nilai eigen. Metodenya adalah, pertama dicari source dengan persamaan (7). Source dan matriks A kemudian diselesaikan dengan eliminasi *gauss* untuk mendapat flux yang baru. Flux yang baru digunakan untuk mencari nilai eigen baru yaitu keff dengan persamaan (6).

Hasil yang diperoleh untuk koordinat silinder 1 dimensi diramalkan mengikuti fungsi *bessel*, sebagai berikut



Untuk mencari daya, metodenya adalah pertama, dilakukan pencarian rapat daya sumber dengan menggunakan persamaan (7). Lalu, dengan menggunakan rapat daya yang diperoleh, dilakukan integrasi benda putar dengan metode integrasi *riemaan* atau simpson 1/3. Selanjutnya, hasil integrasi akan berupa daya untuk 1 lempengan r . Hasil ini diintegrasikan kembali dengan rapat daya dari sumbu- z untuk didapatkan daya total.

Hasil yang diperoleh berupa nilai daya misalnya 1097 MW

4. Prakiraan hasil dan analisis

Hasil dari pencarian fluks adalah distribusi fluks, K_{eff} (koefisien pengali neutron efektif) awal, dan fluks awal. Perhitungan fluks sangat penting untuk menentukan performa reaktor nuklir, salah satunya adalah daya. Flux yang dihasilkan akan turun mengikuti fungsi *bessel*. Sementara itu, hasil perhitungan daya adalah nilai numerik.

5. Referensi

1. Ding, Zechuan.(2018). Solving Bateman Equation for Xenon Transient Analysis Using Numerical Methods.
2. Duderstadt, J. and Hamilton, L. (1976). Nuclear Reactor Analysis. New York: Wiley and Sons.
3. Stacey, Weston M. (2018). Nuclear Reactor Physics. 3rd ed. Wiley.