

# **STUDI PEMECAHAN PERSAMAAN DIFUSI MENGUNAKAN METODE DEEP LEARNING**

**LAPORAN STUDI MANDIRI FI3092**

Oleh  
**MICHAEL ALFARINO BELLA PHUNGPUTRA**  
**NIM 10219111**



**PROGRAM STUDI SARJANA FISIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2022**

## ABSTRAK

# STUDI PEMECAHAN PERSAMAAN DIFUSI MENGGUNAKAN METODE DEEP LEARNING

Oleh

Michael Alfarino Bella Phungputra

NIM: 10219111

Peristiwa difusi dideskripsikan oleh Hukum Kedua Fick yang merupakan salah satu contoh persamaan diferensial parsial. Terdapat beberapa metode untuk memperoleh solusi dari persamaan diferensial parsial, yaitu metode analitik, metode beda hingga dengan skema *Forward-Time Centered-Space* (FTCS), serta *Deep Neural Network* (DNN). Dilakukan simulasi hasil persamaan difusi menggunakan metode numerik FTCS dan DNN. Kemudian, berdasarkan hasil simulasi FTCS dan DNN, dilakukan visualisasi solusi persamaan difusi serta penentuan nilai *mean-squared error* (MSE) untuk tiap metode. Diperoleh kesimpulan bahwa hasil dari DNN memiliki orde MSE yang sama dengan metode FTCS. Nilai MSE untuk DNN dan FTCS masing-masing adalah  $9.894 \times 10^{-5}$  dan  $8.830 \times 10^{-5}$ . Selain itu, DNN memberikan solusi yang kontinu serta tidak terikat oleh kisi (*grid*). Meski demikian, salah satu faktor yang membuat metode beda hingga unggul adalah waktu kalkulasinya yang jauh lebih cepat (5.133 detik) dibanding waktu kalkulasi dari DNN (11.43 menit).

Kata kunci: *deep neural network*, metode beda hingga, FTCS, persamaan difusi.

# DAFTAR ISI

<b>ABSTRAK</b> .....	<b>i</b>
<b>DAFTAR ISI</b> .....	<b>ii</b>
<b>DAFTAR GAMBAR</b> .....	<b>iv</b>
<b>DAFTAR TABEL</b> .....	<b>v</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
I.1 Latar Belakang .....	1
I.2 Rumusan dan Batasan Masalah .....	1
I.3 Tujuan .....	2
I.4 Metodologi .....	2
I.5 Sistematika Penulisan .....	2
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>4</b>
II.1 Persamaan Diferensial Parsial .....	4
II.2 Persamaan Difusi .....	4
II.3 Solusi dari Persamaan Difusi .....	4
II.3.1 <i>Separation of Variables</i> .....	4
II.3.2 Metode <i>Forward-Time Centered-Space</i> (FTCS) .....	5
II.3.3 <i>Deep Neural Network</i> (DNN) .....	6
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>9</b>
III.1 Penyelesaian Persamaan Difusi .....	9
III.1.1 Metode Separasi Variabel .....	9
III.1.2 Metode <i>Forward-Time Centered-Space</i> (FTCS) .....	10
III.1.3 Metode <i>Deep Neural Network</i> (DNN) .....	12
III.2 Ukuran Evaluasi Kinerja .....	13
<b>BAB IV HASIL DAN ANALISIS</b> .....	<b>14</b>
IV.1 Simulasi Difusi 1D Menggunakan Metode Separasi Variabel .....	14
IV.2 Simulasi Difusi 1D Menggunakan metode FTCS .....	14

IV.3 Simulasi Difusi 1D Menggunakan DNN.....	14
IV.4 Evaluasi Kinerja Metode FTCS dan DNN .....	14
<b>BAB V SIMPULAN DAN SARAN .....</b>	<b>17</b>
V.1 Simpulan .....	17
V.2 Saran .....	17
<b>DAFTAR PUSTAKA .....</b>	<b>18</b>

## DAFTAR GAMBAR

<b>Gambar II.1</b> Ilustrasi sebuah <i>Perceptron</i> [Nielsen, 2015] .....	6
<b>Gambar II.2</b> Ilustrasi sederhana dari <i>Deep Neural Network</i> [Nielsen, 2015] ..	7
<b>Gambar III.1</b> <i>Flowchart</i> proses perolehan data metode FTCS .....	11
<b>Gambar III.2</b> <i>Flowchart</i> proses <i>training</i> DNN .....	13
<b>Gambar IV.1</b> Perbandingan antara metode analitik, FTCS, dan DNN.....	15

## DAFTAR TABEL

**Tabel IV.1** Perbandingan puncak persebaran difusi dan MSE untuk tiap metode 15

# BAB I

## PENDAHULUAN

### I.1 Latar Belakang

Persamaan diferensial parsial (PDP) adalah persamaan yang menyatakan hubungan antara fungsi yang memiliki dua atau lebih variabel bebas dan turunan parsial dari fungsi terhadap variabel bebas tersebut. PDP muncul di sebagian besar bidang teknik dan sains. Terdapat banyak proses fisis diatur oleh persamaan diferensial parsial. Salah satunya adalah persamaan difusi yang mendeskripsikan perilaku makroskopis dari banyak mikro-partikel yang bergerak dan menumbuk secara acak pada daerah tertentu.

Solusi dari persamaan diferensial parsial dapat diperoleh secara analitik ketika permasalahan sederhana dan tidak rumit. Jika permasalahannya rumit, dapat dilakukan diskritisasi domain masalah. Selanjutnya, metode *finite difference* (FDM) umum digunakan dalam menyelesaikan permasalahan tersebut.

Akan dianalisis performa *Deep Neural Network* (DNN) dalam menyelesaikan permasalahan difusi. Kemudian, hasil dari DNN akan dibandingkan bersama metode FDM dengan hasil analitik sebagai acuan.

### I.2 Rumusan dan Batasan Masalah

Rumusan masalah pada penelitian ini adalah pencarian alternatif solusi PDP melalui *Deep Neural Network* (DNN) dan membandingkan hasilnya dengan metode *finite difference* (metode beda hingga) dengan skema *Forward-Time Centered-Space* (FTCS). Masalah penelitian ini dibatasi pada persamaan yang dianalisis, yang merupakan persamaan difusi pada satu dimensi, dengan syarat batas Dirichlet dan syarat awal fungsi sinus.

### I.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Menentukan solusi persamaan difusi satu dimensi dengan metode analitik (separasi variabel).
2. Menentukan solusi persamaan difusi satu dimensi dengan metode *finite difference* / beda hingga dengan skema *Forward-Time Centered-Space* (FTCS).
3. Menentukan solusi persamaan difusi satu dimensi dengan *Deep Neural Network* (DNN).
4. Membandingkan hasil penyelesaian persamaan difusi metode analitik, FTCS, dan DNN.

### I.4 Metodologi

Secara umum, penelitian ini dibagi menjadi beberapa tahapan:

1. Studi literatur

Studi literatur dilakukan dalam mempelajari peristiwa difusi dan persamaan difusi beserta metode-metode penyelesaiannya

2. Simulasi solusi persamaan difusi

Dilakukan simulasi hasil persamaan difusi menggunakan metode numerik FTCS dan model DNN.

3. Pengolahan data

Berdasarkan hasil simulasi FTCS dan DNN, dilakukan visualisasi solusi persamaan difusi serta penentuan nilai *mean-squared error* MSE untuk tiap metode.

### I.5 Sistematika Penulisan

Setelah Pendahuluan pada Bab I ini, Bab II akan mengulas tentang teori dari persamaan difusi serta masing-masing metode solusi yang akan disimulasikan pada penelitian ini.



Bab III menjelaskan terkait metode yang digunakan pada penelitian ini yang meliputi penyelesaian persamaan difusi menggunakan metode separasi variabel, FTCS, dan DNN; serta ukuran evaluasi kinerja yang akan digunakan. Bab IV memaparkan tentang hasil yang diperoleh dari pengolahan data, visualisasi, hingga analisis kinerjanya. Terakhir, Bab V menyajikan simpulan yang diperoleh dari hasil penelitian serta perencanaan untuk penelitian Tugas Akhir II.

# BAB II

## TINJAUAN PUSTAKA

### II.1 Persamaan Diferensial Parsial

Persamaan diferensial parsial (PDP) adalah persamaan yang menyatakan hubungan antara fungsi yang memiliki dua atau lebih variabel bebas dan turunan parsial dari fungsi terhadap variabel bebas tersebut. Umumnya, dalam sebagian besar masalah di bidang teknik dan sains, variabel bebasnya adalah ruang  $(x, y, z)$  atau ruang dan waktu  $(x, y, z, t)$ . Variabel terikat tergantung pada masalah fisis yang dimodelkan. Solusi dari persamaan diferensial parsial adalah fungsi tertentu, misal  $f(x, y)$  atau  $f(x, t)$ , yang memenuhi PDE dalam domain tertentu, dan memenuhi kondisi awal dan/atau batas yang ditentukan pada batas domain yang diminati [Hoffman and Frankel, 2001].

### II.2 Persamaan Difusi

Persamaan diferensial parsial (II.1), atau yang sering dikenal dengan persamaan difusi, mendeskripsikan sebaran konsentrasi  $u$  dengan koefisien difusi  $D$ . Menurut Hukum Kedua Fick, persamaan difusi dituliskan sebagai berikut: [Nobel, 1991]

$$\frac{\partial u}{\partial t} = D \nabla^2 u \quad (\text{II.1})$$

dengan  $u = u(t, \mathbf{x})$ ,  $t$  bervariasi pada interval  $\mathbb{R}$ , dan  $\mathbf{x} \in \mathbb{R}^n$ .

Persamaan difusi memiliki aplikasi penting pada berbagai permasalahan sains seperti konduksi panas [Widder, 1976], transpor dan difusi polutan [Karima and Magdalena, 2021, Maitsa et al., 2021], serta deteksi polutan [Badia and Ha-Duong, 2002].

### II.3 Solusi dari Persamaan Difusi

#### II.3.1 *Separation of Variables*

Metode yang umum digunakan untuk menyelesaikan persamaan diferensial parsial seperti persamaan (II.1) secara analitis adalah separasi variabel. Pada metode ini, solusi

konsentrasi dipisah menjadi dua fungsi, fungsi yang bergantung hanya pada koordinat spasial dan fungsi yang bergantung hanya pada koordinat waktu. Sehingga dapat dituliskan sebagai berikut [Boas, 2005]:

$$u = F(\mathbf{x})T(t), \quad (\text{II.2})$$

Dalam beberapa kasus khusus, solusi dari persamaan diferensial parsial dapat dinyatakan dalam ekspresi bentuk tertutup dengan menggunakan metode ini. Namun, pada kenyataannya, sebagian besar masalah di bidang teknik dan sains yang melibatkan persamaan diferensial parsial tidak bisa diselesaikan dengan metode ini [Hoffman and Frankel, 2001].

### II.3.2 Metode *Forward-Time Centered-Space* (FTCS)

Metode *Forward-Time Centered-Space* adalah salah satu jenis pendekatan persamaan diferensial parsial menggunakan *Finite Difference Method* (FDM). FDM atau metode beda hingga adalah salah satu metode penyelesaian persamaan diferensial parsial (seperti persamaan II.1) secara numerik. Tujuan dari metode beda hingga untuk menyelesaikan persamaan diferensial parsial adalah mengubah persoalan kalkulus menjadi persoalan aljabar dengan cara:

1. Diskritisasi domain ruang dan waktu yang kontinu ke dalam kisi yang diskrit
2. Dekati masing-masing turunan parsial eksak yang ada dalam persamaan diferensial parsial dengan pendekatan beda hingga
3. Substitusikan pendekatan ini ke dalam persamaan diferensial parsial untuk memperoleh persamaan beda hingga
4. Selesaikan persamaan beda hingga yang dihasilkan

Pada metode FTCS, titik dasar untuk pendekatan beda hingga dari persamaan diferensial parsial adalah titik ke  $(i, n)$  pada kisi. Turunan parsial pertama terhadap waktu didekati dengan aproksimasi *forward-time* orde pertama, sedangkan turunan parsial kedua terhadap ruang didekati dengan aproksimasi *centered-space* orde kedua. Diberikan pendekatan FTCS untuk satu dimensi ruang dan satu dimensi waktu sebagai

berikut. [Hoffman and Frankel, 2001]

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t} \quad (\text{II.3a})$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \quad (\text{II.3b})$$

### Metode von Neumann dalam Analisis Stabilitas Persamaan Beda Hingga

Persamaan beda hingga dikatakan stabil jika menghasilkan solusi terbatas untuk persamaan diferensial parsial yang stabil, dan dikatakan tidak stabil jika menghasilkan solusi tak terbatas untuk persamaan diferensial parsial yang stabil.

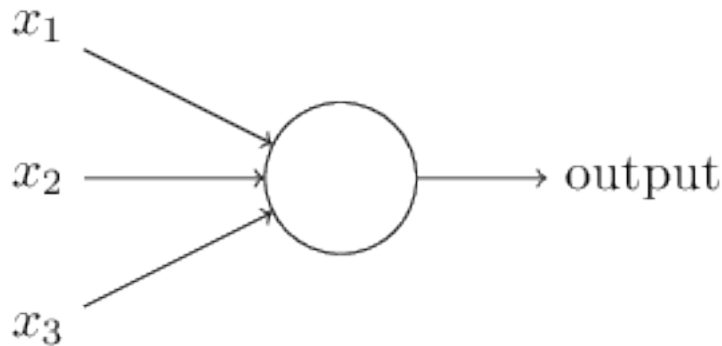
Untuk menganalisis kestabilan persamaan beda hingga, digunakan metode von Neumann yang menyatakan bahwa aproksimasi FTCS dari persamaan difusi (II.1) untuk kasus satu dimensi ruang dan satu dimensi waktu ( $u(t, \mathbf{x}) = u(t, x)$ ) harus memenuhi persyaratan berikut: [Hoffman and Frankel, 2001]

$$\frac{D\Delta t}{\Delta x^2} \leq \frac{1}{2} \quad (\text{II.4})$$

### II.3.3 Deep Neural Network (DNN)

#### Perceptron

*Perceptron* adalah sebuah fungsi yang mengambil sejumlah input ( $x_1, x_2, x_3, \dots$ ) dan menghasilkan satu buah keluaran. Di dalam sebuah *perceptron* terdapat sejumlah beb-



**Gambar II.1:** Ilustrasi sebuah *Perceptron* [Nielsen, 2015]

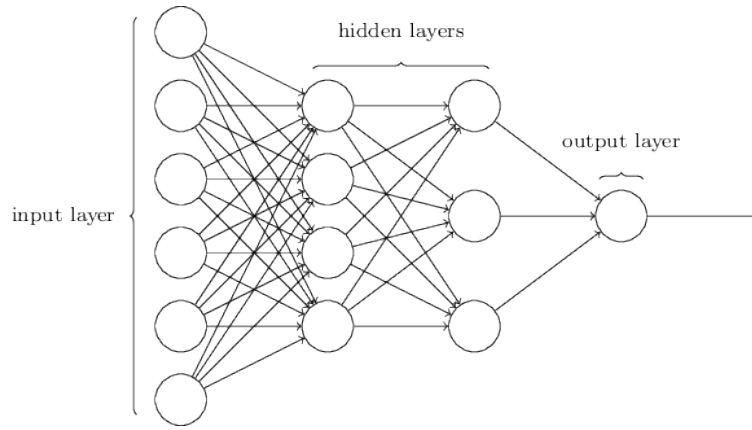
an (*weights*; ( $w_1, w_2, w_3, \dots$ )) dan *bias* ( $b_1, b_2, b_3, \dots$ ) yang menggambarkan tingkat

kepentingan dari suatu input terhadap output *perceptron*. Kemudian, hasil dari perhitungan input tersebut dimasukkan ke sebuah fungsi aktivasi ( $\sigma(x)$ ). Secara matematis output setiap *perceptron* dievaluasi seperti berikut: [Nielsen, 2015]

$$\text{output} = \sigma \left( \sum_i (w_i x_i + b_i) \right) \quad (\text{II.5})$$

### **Deep Neural Network**

*Deep Neural Network* adalah suatu struktur jaringan kompleks dari *perceptron* yang memiliki *hidden layer* lebih dari satu.



**Gambar II.2:** Ilustrasi sederhana dari *Deep Neural Network* [Nielsen, 2015]

Setiap input pada *input layer* (lapisan pertama) terhubung ke setiap perceptron pada *layer* berikutnya. Begitu seterusnya hingga diperoleh output pada *layer* terakhir.

$\mathcal{N}^L(\mathbf{x}) : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{out}}$  didefinisikan sebagai *neural network* dengan  $L$  buah *layer*, dengan *layer*  $\ell$  memiliki  $N^\ell$  buah *neuron* ( $N_0 = d_{in}$  dan  $N_L = d_{out}$ ). Matrix beban (*weight*) dan vektor *bias* pada *layer*  $\ell$  masing-masing didefinisikan sebagai  $\mathbf{W}^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$  dan  $\mathbf{b}^\ell \in \mathbb{R}^{N_\ell}$ . Menggunakan fungsi aktivasi yang didefinisikan sebagai  $\sigma$ , *neural network* didefinisikan secara rekursif seperti berikut: [Caterini and Chang, 2018]

$$\text{input layer:} \quad \mathcal{N}^0(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^{d_{in}} \quad (\text{II.6a})$$

$$\text{hidden layers:} \quad \mathcal{N}^\ell(\mathbf{x}) = \sigma \left( \mathbf{W}^\ell \mathcal{N}^{\ell-1}(\mathbf{x}) + \mathbf{b}^\ell \right) \in \mathbb{R}^{N^\ell} \quad (\text{II.6b})$$

$$\text{output layer:} \quad \mathcal{N}^L(\mathbf{x}) = \mathbf{W}^L \mathcal{N}^{L-1}(\mathbf{x}) + \mathbf{b}^L \in \mathbb{R}^{d_{out}} \quad (\text{II.6c})$$

Fungsi aktivasi ( $\sigma$ ) yang biasanya digunakan adalah fungsi logistik  $1/(1 + e^{-x})$ , fungsi tangen hiperbolik ( $\tanh x$ ), dan fungsi *rectified linear unit* (ReLU;  $\max\{x, 0\}$ ).

Perlu dilakukan proses *training* pada *neural network* dalam aproksimasi  $\mathcal{N}(\mathbf{x})$  untuk semua input  $\mathbf{x}$ . Proses ini melibatkan *data training* yang berupa input bersamaan dengan keluaran yang tepat.

### ***Loss Function***

Untuk mengukur seberapa baik model yang dibentuk, didefinisikan fungsi kerugian (*loss function*). Beberapa contoh dari *loss function* diantaranya adalah *mean-squared error* (MSE) dan *cross-entropy loss function* [Nielsen, 2015].

### ***Metode Optimasi Adam***

Dalam meminimalkan *loss function* yang digunakan, diperlukan suatu algoritma optimasi. *Adam* adalah sebuah metode optimasi stokastik yang efisien yang hanya membutuhkan gradien orde pertama dengan kebutuhan memori yang kecil. Nama Adam berasal dari *Adaptive moment estimation*. Metode ini menghitung *learning rate* secara individual dan adaptif untuk berbagai parameter, yang berasal dari estimasi momen pertama dan kedua dari gradien [Kingma and Ba, 2014].

# BAB III

## METODOLOGI PENELITIAN

### III.1 Penyelesaian Persamaan Difusi

Akan dicari solusi persamaan difusi (II.1) dalam satu dimensi ruang dan satu dimensi waktu dengan syarat batas Dirichlet dan nilai awal sebagai berikut.

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, 2], \quad t \in [0, 2] \quad (\text{III.1a})$$

$$u(0, t) = u(2, t) = 0 \quad (\text{III.1b})$$

$$u(x, 0) = \sin\left(\frac{\pi}{2}x\right) \quad (\text{III.1c})$$

Digunakan tiga metode untuk menyelesaikan persamaan difusi tersebut, yaitu metode separasi variabel, metode *Forward-Time Centered-Space* (FTCS), serta metode *Deep Neural Network* (DNN).

#### III.1.1 Metode Separasi Variabel

Substitusi persamaan (II.2) ke persamaan (III.1a), sehingga diperoleh

$$DT \frac{\partial^2 F}{\partial x^2} = F \frac{\partial T}{\partial t} \quad (\text{III.2a})$$

$$\frac{1}{F} \frac{\partial^2 F}{\partial x^2} = \frac{1}{DT} \frac{\partial T}{\partial t} \quad (\text{III.2b})$$

Ruas kiri dan kanan persamaan (III.2b) adalah fungsi dari  $x$  saja atau  $t$  saja, sehingga keduanya bernilai konstan dan dapat ditulis:

$$\frac{1}{F} \frac{\partial^2 F}{\partial x^2} = -k^2 \quad \implies \quad \frac{\partial^2 F}{\partial x^2} + k^2 F = 0 \quad (\text{III.3a})$$

$$\frac{1}{DT} \frac{\partial T}{\partial t} = -k^2 \quad \implies \quad \frac{\partial T}{\partial t} = -k^2 Dt \quad (\text{III.3b})$$

Dengan  $k$  adalah suatu konstanta. Solusi dari kedua persamaan tersebut adalah:

$$F(x) = A \sin(kx - \phi) \quad (\text{III.4a})$$

$$T(t) = e^{-k^2 D t} \quad (\text{III.4b})$$

Sehingga,

$$u(x, t) = F(x)T(t) = e^{-k^2 D t} A \sin(kx - \phi) \quad (\text{III.5})$$

Substitusikan syarat batas dan nilai awal (III.1b) dan (III.1c), sehingga diperoleh solusi:

$$u(x, t) = e^{-\left(\frac{\pi}{2}\right)^2 D t} \sin\left(\frac{\pi}{2} x\right) \quad (\text{III.6})$$

### III.1.2 Metode *Forward-Time Centered-Space* (FTCS)

Substitusi persamaan (II.3a) dan (II.3b) ke persamaan (III.1a). Sehingga diperoleh solusi:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = D \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \quad (\text{III.7a})$$

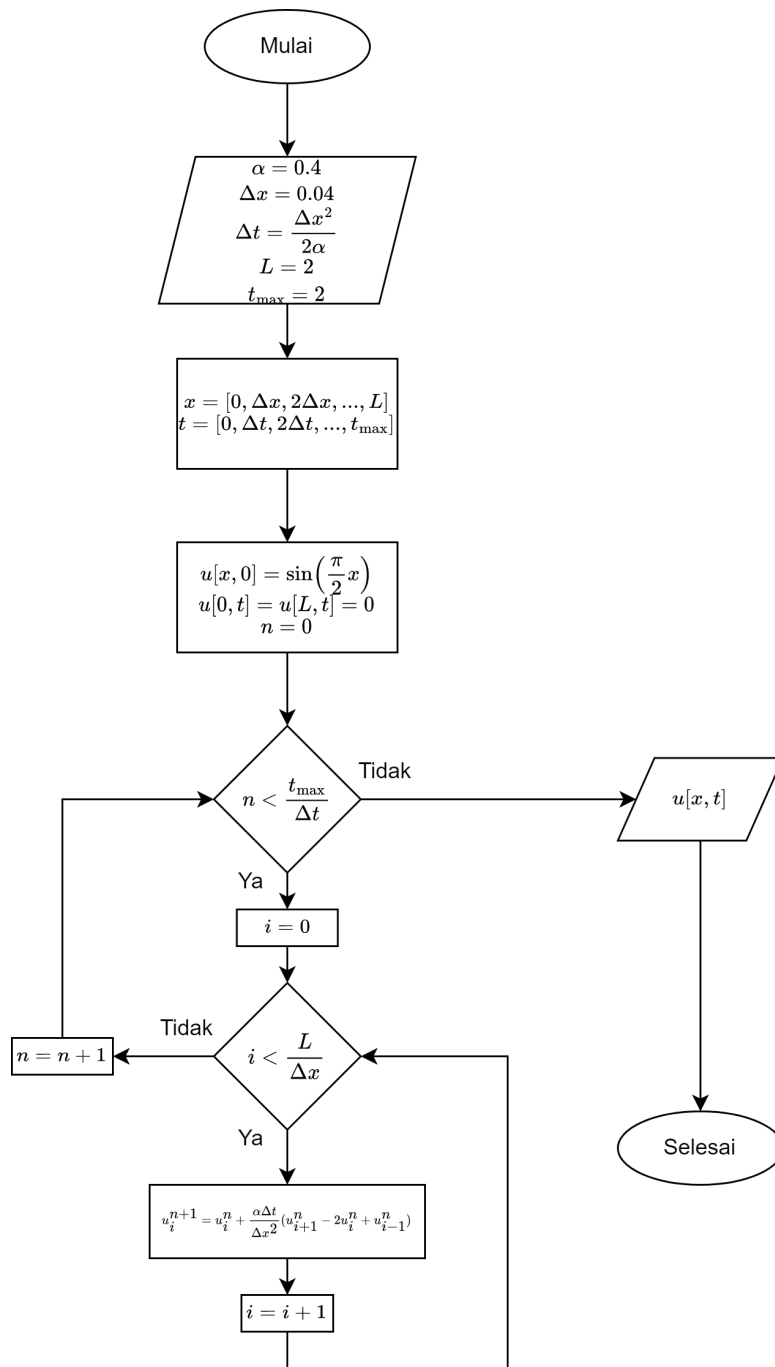
$$u_i^{n+1} = u_i^n + \frac{D \Delta t}{\Delta x^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (\text{III.7b})$$

Dengan memilih nilai  $D = 0.4$ ,  $\Delta x = 0.04$ , maka sesuai metode von Neumann (II.4) diperoleh:

$$\Delta t \leq 0.002 \quad (\text{III.8})$$

Dipilih nilai  $\Delta t$  terbesar yaitu  $\Delta t = 0.002$ . Dari hasil tersebut, dilakukan komputasi secara numerik untuk memperoleh data solusi persamaan difusi dengan metode FTCS. Berikut merupakan alur flowchart yang menggambarkan proses komputasi dan perhitungan dalam melakukan pengambilan data.





**Gambar III.1:** Flowchart proses perolehan data metode FTCS

### III.1.3 Metode *Deep Neural Network* (DNN)

Untuk metode DNN, digunakan model *neural network* yang sama seperti persamaan (II.6). Kemudian, digunakan hasil dari metode beda hingga untuk menjadi *data training*. *Loss function* ( $\mathcal{L}$ ) yang digunakan adalah *mean-squared error* seperti pada persamaan berikut.

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n (y_n - \hat{y}_n)^2 \quad (\text{III.9})$$

Dengan  $n$  adalah jumlah data training,  $y_n$  adalah hasil masukan *neural network*, dan  $\hat{y}_n$  adalah nilai benar dari masukan (nilai dari metode beda hingga).

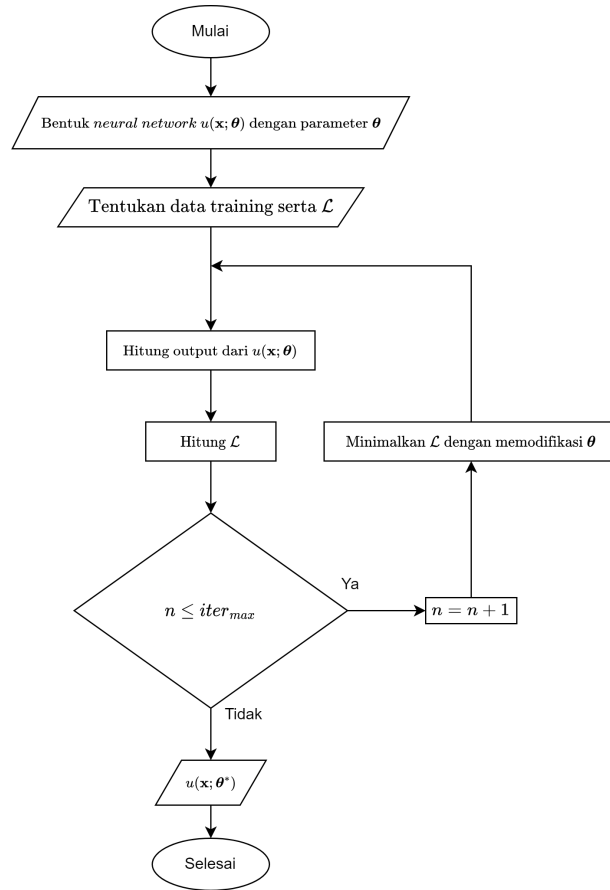
Fungsi aktivasi yang akan digunakan adalah fungsi sigmoid:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (\text{III.10})$$

serta digunakan metode optimasi berbasis gradien: Adam [Kingma and Ba, 2014]

Seperti yang telah dijelaskan sebelumnya, *loss function* (III.9) akan dioptimasi (diminimalkan) hingga ditemukan sekumpulan beban (*weights*) dan *bias* yang ideal untuk memodelkan persamaan difusi (III.1).

Berikut merupakan alur flowchart yang menggambarkan proses *training* dari DNN.



**Gambar III.2:** Flowchart proses *training* DNN

## III.2 Ukuran Evaluasi Kinerja

Untuk mengevaluasi kinerja metode FTCS dan metode DNN, keduanya akan dibandingkan dengan hasil analitik (separasi variabel) dengan mengevaluasi galat menggunakan *mean-squared error* (MSE).

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{u}(x, t) - u(x, t))^2 \quad (\text{III.11})$$

Dimana  $\hat{u}(x, t)$  adalah solusi dari FTCS dan DNN, dan  $n$  adalah jumlah titik pada solusi FTCS dan DNN.

# BAB IV

## HASIL DAN ANALISIS

### IV.1 Simulasi Difusi 1D Menggunakan Metode Separasi Variabel

Telah diselesaikan persamaan difusi (III.1) menggunakan metode separasi variabel dan diperoleh hasil seperti pada persamaan (III.6). Hasil ini digunakan sebagai tolok ukur untuk metode FTCS dan DNN.

### IV.2 Simulasi Difusi 1D Menggunakan metode FTCS

Dilakukan simulasi dengan langkah-langkah seperti pada Gambar III.1. Simulasi berjalan selama 5.133 detik.

Salah satu konsep dasar dari metode beda hingga adalah diskritisasi domain ruang dan waktu yang kontinu ke dalam kisi yang diskrit. Sehingga dapat dilihat bahwa pada metode FTCS, hasil yang diperoleh berupa titik-titik pada kurva, bukan kurva yang kontinu. Hasil dari metode ini akan selalu terikat pada kisi yang diskrit.

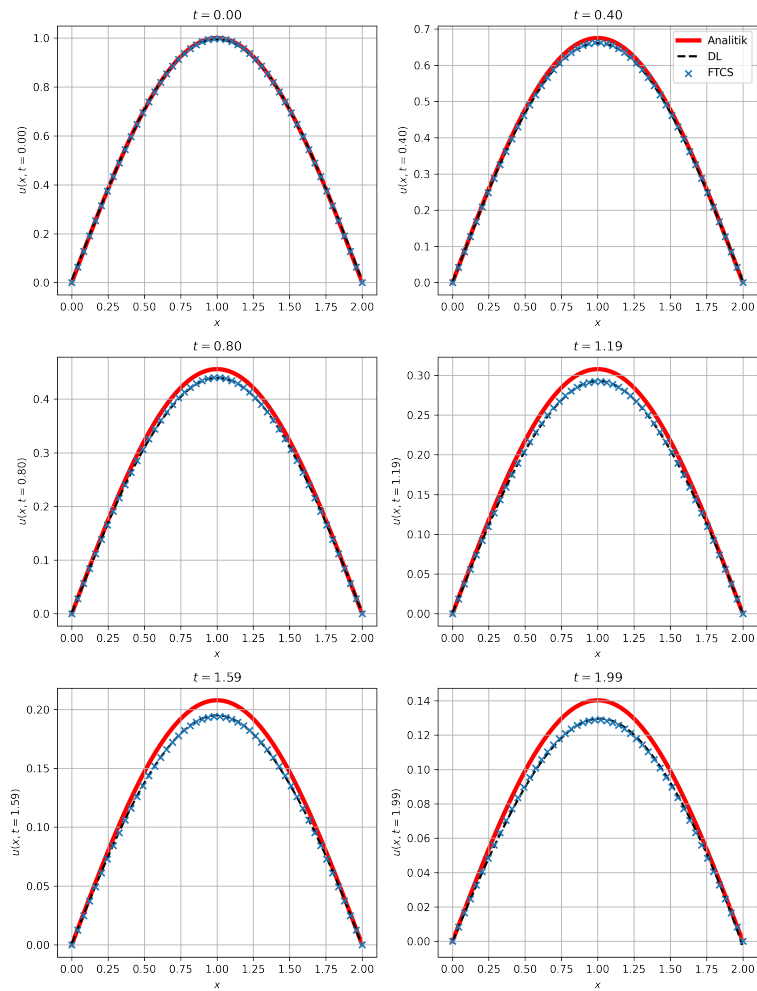
### IV.3 Simulasi Difusi 1D Menggunakan DNN

Dilakukan simulasi dengan langkah-langkah seperti pada Gambar III.2. Digunakan struktur DNN 3 layer, dengan jumlah *neuron / perceptron* per layer sebanyak 20. Simulasi berjalan selama 11.43 menit.

Hasil penyelesaian persamaan difusi melalui DNN menghasilkan solusi yang kontinu, sehingga dapat dievaluasi pada titik acak.

### IV.4 Evaluasi Kinerja Metode FTCS dan DNN

Hasil dari metode analitik, FTCS, dan DNN kemudian dibandingkan secara bersamaan. Berikut merupakan visualisasi hasil ketiga metode pada grafik yang merepresentasikan sebaran konsentrasi  $u(x, t)$  terhadap posisi  $x$  untuk tiap waktu  $t$



**Gambar IV.1:** Perbandingan antara metode analitik, FTCS, dan DNN

Kemudian, hasil dievaluasi dengan mencari puncak dari kurva serta selisih kuadrat rata-rata antara nilai estimasi (FTCS dan DNN) dan nilai sebenarnya (Analitik). Digunakan persamaan (III.11) untuk memperoleh nilai *mean-squared error*. Hasilnya disajikan pada Tabel IV.1

**Tabel IV.1:** Perbandingan puncak persebaran difusi dan MSE untuk tiap metode

Metode	Puncak	MSE
Analitik	0.9999	
FTCS	0.9974	$8.830 \times 10^{-5}$
DNN	0.9939	$9.894 \times 10^{-5}$

Tampak bahwa hasil dari DNN memiliki orde MSE yang sama dengan metode FTCS, dengan orde  $10^{-5}$ . Dari grafik juga terlihat bahwa metode DNN mendekati metode FTCS, karena memang menjadi acuan pada saat proses *training*.

Selain itu, DNN memberikan solusi yang kontinu karena solusinya yang berbentuk fungsi. Hal ini menjadikan hasil dari DNN berguna untuk permasalahan lebih lanjut yang memerlukan fungsi kontinu. DNN juga tidak terikat oleh kisi (*grid*), sehingga hasil  $u(\mathbf{x})$  dapat diperoleh pada titik acak pada domain input.

Namun, keunggulan yang dimiliki oleh metode FTCS adalah waktu kalkulasinya yang jauh lebih cepat (5.133 detik) dibanding waktu kalkulasi dari DNN (11.43 menit).

Metode yang lebih cocok untuk menyelesaikan permasalahan diferensial parsial adalah metode *Physics-Informed Neural Network* (PINN), dimana DNN dimodifikasi dan digabungkan dengan hukum-hukum fisika. *Loss function* dari metode tersebut merupakan PDP yang menjadi permasalahan. Hasil yang diperoleh dari metode tersebut tidak bergantung pada metode lain. Tidak diperlukan sintesis *data training* menggunakan data dari hasil analitik maupun hasil dari metode beda hingga. [Lu et al., 2021, Raissi et al., 2017].

# BAB V

## SIMPULAN DAN SARAN

### V.1 Simpulan

Berdasarkan penelitian yang telah dilakukan, telah diselesaikan persamaan difusi satu dimensi dengan syarat batas Dirichlet dan nilai awal  $\sin(\frac{\pi}{2}x)$  menggunakan tiga metode: analitik, FTCS, dan DNN. Hasil dari DNN memiliki orde MSE yang sama dengan metode FTCS. Nilai MSE untuk DNN dan FTCS masing-masing adalah  $9.894 \times 10^{-5}$  dan  $8.830 \times 10^{-5}$ .

Selain itu, solusi yang dihasilkan metode FTCS berupa titik-titik pada kurva, bukan kurva yang kontinu. Sedangkan DNN memberikan solusi yang kontinu serta tidak terikat oleh kisi (*grid*), sehingga menjadikan hasil dari DNN berguna untuk permasalahan lebih lanjut yang memerlukan fungsi kontinu. Meski demikian, salah satu faktor yang membuat metode beda hingga unggul adalah waktu kalkulasinya yang jauh lebih cepat (5.133 detik) dibanding waktu kalkulasi dari DNN (11.43 menit).

### V.2 Saran

Perlu digunakan metode lain yang lebih cocok untuk menyelesaikan permasalahan diferensial parsial. Contohnya adalah metode *Physics-Informed Neural Network* (PINN), dimana DNN dimodifikasi dan digabungkan dengan hukum-hukum fisika. *Loss function* dari metode tersebut merupakan PDP yang menjadi permasalahan. Hasil yang diperoleh dari metode tersebut tidak bergantung pada metode lain. Tidak diperlukan sintesis *data training* menggunakan data dari hasil analitik maupun hasil dari metode beda hingga. [Lu et al., 2021, Raissi et al., 2017].

## DAFTAR PUSTAKA

- [Badia and Ha-Duong, 2002] Badia, A. E. and Ha-Duong, T. (2002). On an inverse source problem for the heat equation. application to a pollution detection problem. *Journal of Inverse and Ill-posed Problems*, 10(6):585–599.
- [Baydin et al., 2015] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2015). Automatic differentiation in machine learning: a survey.
- [Boas, 2005] Boas, M. (2005). *Mathematical Methods in the Physical Sciences*. Wiley.
- [Caterini and Chang, 2018] Caterini, A. and Chang, D. (2018). *Deep Neural Networks in a Mathematical Framework*. SpringerBriefs in Computer Science. Springer International Publishing.
- [Hoffman and Frankel, 2001] Hoffman, J. and Frankel, S. (2001). *Numerical Methods for Engineers and Scientists, Second Edition*,. Taylor & Francis.
- [Karima and Magdalena, 2021] Karima, N. and Magdalena, I. (2021). The 2nd, 4th, and 6th-order finite difference schemes for pollutant transport equation. In *Proceedings of the 1st International Conference on Mathematics and Mathematics Education (ICMMEd 2020)*, pages 29–33. Atlantis Press.
- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.
- [Lu et al., 2021] Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021). Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.
- [Maitsa et al., 2021] Maitsa, T., Hafiyyan, Q., Adityawan, M., Magdalena, I., Adi Kuntoro, A., and Kardhana, H. (2021). Development of a 2d numerical model for pollutant transport using ftcs scheme and numerical filter. *Makara Journal of Technology*, 25:119.
- [Nielsen, 2015] Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.



- [Nobel, 1991] Nobel, P. (1991). *Physicochemical and Environmental Plant Physiology*. Academic Press.
- [Raissi et al., 2017] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations.
- [Verma, 2000] Verma, A. (2000). An introduction to automatic differentiation. *Current Science*, 78(7):804–807.
- [Widder, 1976] Widder, D. (1976). *The Heat Equation*. Pure and Applied Mathematics. Elsevier Science.