IMPLEMENTASI PENYELESAIAN PERSAMAAN BURGERS DENGAN METODE BEDA HINGGA DALAM BAHASA PEMROGRAMAN JULIA

*F. Bukhari[1], S. Nurdiati[2], M.T. Julianto[3], M. K. Najib[4] dan R. H. Valentdio[5]

[1,2,3,4,5)] Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Institut Pertanian Bogor, Jl. Meranti, Kampus IPB Dramaga Bogor.
fahrenbu@apps.ipb.ac.id, ***corresponding author,** nurdiati@apps.ipb.ac.id
mkhoirun_najib@apps.ipb.ac.id, ruben_322@apps.ipb.ac.id

Abstract

Burgers equation is a partial differential equation used to modelling several events related to fluids. Burgers equation was firstly introduced by Harry Bateman in 1915 and later studied by Johannes Martinus Burgers in 1948. This study discusses solving Burgers equations with finite difference method. In this study, several parameters have been known for the Burgers equation and several cases of partitions are used in finite difference method. The result shows that the more partitions used, the numerical result obtained will be closer to the exact values. In this study, calculations are numerically carried out with the help of Julia programming language.

Keywords: *Burgers equation, finite difference method, Julia, partial differential equation*

1 Pendahuluan

Pemodelan sering digunakan orang dalam memahami kejadian di sekitarnya, khususnya dalam memahami kejadian alam. Tidak jarang persamaan matematika digunakan untuk dapat memodelkan permasalahan kejadian alam secara tepat. Dalam sejarah perkembangan persamaan matematika, para ilmuwan menemukan persamaan diferensial parsial sebagai salah satu persamaan yang dapat memodelkan perilaku kejadian alam secara lebih tepat. Salah satu persamaan diferensial parsial yang menarik perhatian orang adalah persamaan Burgers.

Persamaan Burgers merupakan salah satu persamaan diferensial parsial yang banyak digunakan dalam ilmu matematika terapan. Persamaan ini pertama kali diperkenalkan oleh Harry Bateman pada tahun 1915 dan kemudian dikembangkan oleh Johannes Martinus Burgers pada tahun 1948. Karena itu persamaan ini sering juga dinamakan sebagai Persamaan Bateman-Burgers.

Persamaan Burgers dapat diselesaikan secara analitik [6]. Tetapi yang menjadi persoalan dari penyelesaian secara analitik ialah tingkat kesulitan yang meningkat seiring dengan semakin rumitnya persamaan Burgers yang akan diselesaikan, serta tidak secara langsung menghasilkan solusi berupa nilai (angka) yang diinginkan. Karena itu, dalam penelitian ini akan dilakukan penyelesaian persamaan Burgers secara numerik dengan memanfaatkan metode beda hingga. Dalam penelitian ini, proses pencarian solusi numerik akan dilakukan menggunakan bahasa pemrograman Julia.

2020 Mathematics Subject Classification: 35F20, 35Q30, 65M06.
Diajukan: 11/11/2022, diterima: 26/06/2023.

2 Metode Penelitian

2.1 Persamaan Navier-Stokes

Persamaan Navier-Stokes merupakan bentuk diferensial dari hukum kedua Newton dalam masalah pergerakan dari suatu fluida. Persamaan ini fokus pada perubahan pada momentum partikel-partikel fluida yang bergantung pada gaya viskos internal dan gaya viskos tekanan eksternal yang bekerja pada fluida. Dapat disimpulkan bahwa persamaan Navier-Stokes menjelaskan kesetimbangan gaya-gaya yang bekerja pada fluid [10]. Dengan kata lain, persamaan Navier-Stokes merupakan persamaan diferensial parsial yang menggambarkan aliran fluida tidak dapat dimampatkan (*incompressible*). Persamaan Navier-Stokes dapat dituliskan dalam bentuk

$$\rho\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) = -\nabla P + \mu \nabla^2 u + F \tag{1}$$

dengan u adalah medan vektor kecepatan fluida, ∇P adalah tekanan fluida terhadap sumbu $x, y,$ dan z, μ adalah viskositas (kekentalan) suatu fluida, $\nabla u = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z}$ adalah laju kecepatan fluida terhadap sumbu $x, y,$ dan z, serta F adalah sumber yang menggambarkan gaya eksternal pada fluida [7]. Pada kasus khusus, bentuk persamaan Navier-Stokes dapat disederhanakan menjadi bentuk yang lebih sederhana.

2.2 Persamaan Burgers

Persamaan Burgers merupakan bentuk khusus dari persamaan Navier-Stokes. Persamaan Burgers digunakan dalam bidang dinamika gas [1], gelombang kejut [11], dan arus lalu lintas [5]. Persamaan ini didapat dengan mengasumsikan bahwa gaya eksternal (F) bernilai 0 dan membagi persamaan dengan ρ sehingga didapatkan bentuk persamaan Navier-Stokes yang baru.

$$u_t + u \cdot \nabla u = -\frac{\nabla P}{\rho} + \upsilon \nabla^2 u, \tag{2}$$

dengan $\upsilon = \frac{\mu}{\rho} > 0$ adalah koefisien difusi pada suatu fluida, karena dalam fluida tidak termampatkan (*incompressible*) nilai ρ selalu konstan. Selanjutnya, jika diasumsikan bahwa tidak ada tekanan (nilai P bernilai 0) maka akan didapatkan persamaan baru sebagai berikut.

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = \upsilon \cdot \nabla u^2 \tag{3}$$

Persamaan inilah yang nantinya banyak diketahui dan digunakan sebagai persamaan Burgers. Dalam penerapannya, persamaan Burgers disesuaikan menjadi persamaan

Burgers satu dimensi dan persamaan Burgers dua dimensi. Dalam penelitian ini persmaan Burgers yang akan dipakai adalah persamaan Burgers satu dimensi.

2.3 Metode Beda Hingga

Penyelesaian Persamaan Diferensial Parsial (PDP) dalam metode numerik umumnya menggunakan metode beda hingga. Mengganti turunan yang ada pada persamaan differensial dengan diskritisasi beda hingga berdasarkan deret Taylor merupakan prinsip dari metode beda hingga [2]. Metode beda hingga bekerja dengan mengubah daerah variabel bebas menjadi grid berhingga yang disebut *mesh* dimana variabel bebas diaproksimasi [8].

Ada tiga jenis hampiran dalam metode beda hingga, yaitu hampiran beda maju (*forward difference*), hampiran beda mundur (*backward difference*), dan hampiran beda pusat (*central difference*) [9]. Dalam penelitian ini, hampiran yang akan digunakan ialah hampiran beda maju pada U_t dan beda mundur pada U_x sehingga turunan parsial bagi U terhadap x dan t dapat dituliskan dalam bentuk berikut.

$$U_x(t,x) = \frac{U(t,x) - U(t,x-\Delta x)}{\Delta x} \tag{4}$$

$$U_t(t,x) = \frac{U(t+\Delta t,x) - U(t,x)}{\Delta t} \tag{5}$$

Variabel x dipartisi menjadi diskret yang disebut *grid* $x_1, x_2, \dots, x_N$ dan t dipartisi pada $t_0, t_1, t_2, \dots, t_M$. Selanjutnya, banyaknya partisi pada variabel x akan disebut N_x dan banyaknya partisi pada variabel t akan disebut N_t. Selanjutnya, jika Δx adalah konstanta dan $x_{i-1} = x_i - \Delta x$ dan $t_{n+1} = t_n + \Delta t$ maka dapat diperoleh

$$U_x(t_n,x_i) = \frac{U(t_n,x_i) - U(t_n,x_{i-1})}{\Delta x} \tag{6}$$

$$U_t(t_n,x_i) = \frac{U(t_{n+1},x_i) - U(t_n,x_i)}{\Delta t} \tag{7}$$

Untuk turunan parsial dengan orde dua, metode beda hingga dapat dikonstruksi dengan melakukan pemangkasan deret Taylor sampai turunan parsial orde dua dan dengan menggunakan Δx serta $-\Delta x$.

$$U(t,x+\Delta x) = U(t,x) + \Delta x U_x(t,x) + \frac{\Delta x^2}{2!} U_{xx}(t,x) \tag{8}$$

$$U(t,x-\Delta x) = U(t,x) - \Delta x U_x(t,x) + \frac{\Delta x^2}{2!} U_{xx}(t,x) \tag{9}$$

Dengan menjumlahkan Persamaan 7 dan 8 serta menggunakan aproksimasi diperoleh

$$U_{xx}(t_n, x_i) = \frac{U(t_n, x_{i+1}) - 2U(t_n, x_i) + U(t_n, x_{i-1})}{\Delta x^2} \tag{10}$$

Untuk mempermudah penulisan, maka akan digunakan simbol U_i^n untuk menyatakan $U(t_n, x_i)$, atau

$$U_i^n = U(t_n, x_i) \tag{11}$$

sehingga Persamaan 6, 7, dan 10 dapat dituliskan menjadi

$$U_x(t_n, x_i) = \frac{U(t_n, x_i) - U(t_n, x_{i-1})}{\Delta x} = \frac{U_i^n - U_{i-1}^n}{\Delta x}$$

$$U_t(t_n, x_i) = \frac{U(t_{n+1}, x_i) - U(t_n, x_i)}{\Delta t} = \frac{U_i^{n+1} - U_i^n}{\Delta t}$$

$$U_{xx}(t_n, x_i) = \frac{U(t_n, x_{i+1}) - 2U(t_n, x_i) + U(t_n, x_{i-1})}{\Delta x^2} = \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}$$

2.4 Tahapan Penelitian

Penyelesaian persamaan Burgers diawali dengan proses diskretisasi persamaan Burgers menggunakan metode beda hingga (*finite difference method*) [3]. Dalam pengaplikasiannya, diterapkan metode hampiran beda maju (*forward difference*) untuk waktu, metode hampiran beda mundur (*backward difference*) untuk ruang, dan metode orde ke-2 (*second order method*) untuk persamaan diferensial orde-2. Dalam penelitian ini, bahasa pemrograman yang akan digunakan yaitu Julia.

Selanjutnya, ditetapkan kondisi awal dan juga nilai batas pada persamaan Burgers. Kondisi awal dan nilai batas bersumber dari [3]. Pada kondisi awal ini juga telah diberikan solusi analitik bagi persamaan tersebut sehingga dapat diketahui perbandingan antara hasil solusi numerik yang didapat dari metode beda hingga dengan hasil solusi eksak yang didapat dari solusi analitik yang telah diketahui.

Dalam pencarian solusi numerik bagi persamaan Burgers menggunakan metode beda hingga, digunakan nilai-nilai parameter yang bersumber dari Barba dan Forsyth (2018) dengan menambahkan beberapa kasus pada N_x sebagai berikut.

- N_x (banyaknya partisi pada x) : 101, 201, 301
- N_t (banyaknya partisi pada t) : 100
- v (viskositas bagi persamaan Burgers) : 0.07
- Δx (jarak antar partisi pada x) : $\frac{2\pi}{N_x - 1}$, dengan 2π merupakan selang maksimum dari nilai x yang ditetapkan yaitu $0 \leq x \leq 2\pi$
- Δt (jarak antar partisi pada t) : $\Delta x \cdot v$
- $t = 0$ sebagai kondisi t awal dalam memulai algoritma

Setelah didapatkan hasil solusi numerik dari perhitungan metode beda hingga, dibuat masing-masing plot bagi hasil solusi numerik serta plot dari solusi eksak yang didapat dari penyelesaian solusi analitik. Berdasarkan masing-masing plot yang telah dibuat, perbandingan antara solusi numerik dari metode beda hingga dengan solusi eksak dari persamaan solusi analitiknya dianalisis.

3 Hasil dan Pembahasan

3.1 Diskretisasi Menggunakan Metode Beda Hingga

Dalam penelitian ini, persamaan Burgers satu dimensi digunakan sebagai contoh kasus. Persamaan Burgers satu dimensi [4] dapat dituliskan sebagai berikut.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2} \tag{12}$$

dengan u menggambarkan kecepatan aliran sepanjang arah vektor x.

Persamaan Burgers akan diselesaikan dengan metode beda hingga. Pertama, diskretisasi dilakukan menggunakan beda hingga skema maju (*forward difference*) untuk persamaan waktu (t) , beda hingga skema mundur (*backward difference*) untuk persamaan ruang (x), dan beda hingga skema turunan spasial ke-2 untuk persamaan yang memiliki turunan parsial orde dua sehingga didapatkan persamaan baru hasil diskretisasi sebagai berikut.

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + `U_i^n \frac{U_i^n - U_{i-1}^n}{\Delta x} = v\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2} \tag{13}$$

3.2 Kondisi Awal dan Nilai Batas serta Persamaan Inisial

Dalam penelitian ini, telah ditentukan kondisi awal serta nilai batas untuk persamaan Burger sebagai berikut.

$$\begin{aligned} u &= -\frac{2v}{\phi} \times \frac{\partial \phi}{\partial x} + 4 \\ \phi &= \exp\left(\frac{-x^2}{4v}\right) + \exp\left(\frac{-(x-2\pi)^2}{4v}\right) \end{aligned} \tag{14}$$

Sementara itu, nilai batas yang ditetapkan ialah $u(0) = u(2\pi)$ sebagai nilai batas periodik. Berdasarkan [3], kondisi awal ini memiliki solusi analitik yang dapat dituliskan sebagai berikut.

$$u = -\frac{2v}{\phi} \times \frac{\partial \phi}{\partial x} + 4$$
$$\phi = exp\left(\frac{-(x-4t)^2}{4v(t+1)}\right) + \exp\left(\frac{-(x-4t-2\pi)^2}{4v(t+1)}\right) \quad (15)$$

Nilai batas yang ditetapkan bagi persamaan Burgers ini ialah $u(0) = u(2\pi)$ sebagai nilai batas periodik. Setelah itu, gunakan nilai ϕ pada persamaan (14) ke dalam kondisi awal pada persamaan u sehingga didapatkan persamaan u yang baru sebagai berikut

$$u = -\frac{2v}{\exp\left(\frac{-(x-4t)^2}{4v(t+1)}\right) + \exp\left(\frac{-(x-4t-2\pi)^2}{4v(t+1)}\right)} \times \frac{\partial\left(\exp\left(\frac{-(x-4t)^2}{4v(t+1)}\right) + \exp\left(\frac{-(x-4t-2\pi)^2}{4v(t+1)}\right)\right)}{\partial x} + 4 \quad (16)$$

3.3 Solusi Numerik Persamaan Burgers

Persamaan (16) digunakan sebagai kondisi awal bagi persamaan Burgers. Misalkan diambil nilai $N_x = 101$, sehingga dapat dicari nilai bagi Δx yaitu

$$\Delta x = \frac{2\pi}{101 - 1} = \frac{2\pi}{100} \quad (17)$$

Berdasarkan nilai Δx, dapat dicari nilai bagi Δt yaitu

$$\Delta t = \frac{2\pi}{100} \cdot 0.07 = \frac{14\pi}{10^4} \quad (18)$$

Dari Persamaan 13, dapat diketahui bahwa ketika kondisi awal serta nilai batas dimasukkan pada persamaan, maka variabel yang tidak diketahui nilainya ialah U_i^{n+1}. Variabel tersebut dapat dicari nilainya dengan mengubah Persamaan 13 menjadi bentuk sebagai berikut.

$$U_i^{n+1} = U_i^n - U_i^n \frac{\Delta t}{\Delta x}(U_i^n - U_{i-1}^n) + v\frac{\Delta t}{\Delta x^2}(U_{i+1}^n - 2U_i^n + U_{i-1}^n) \quad (18)$$

Setelah itu, dengan memasukkan nilai Δx dan Δt pada Persamaan 16 dan 17 maka akan didapatkan bentuk sederhana bagi U_i^{n+1} sebagai berikut

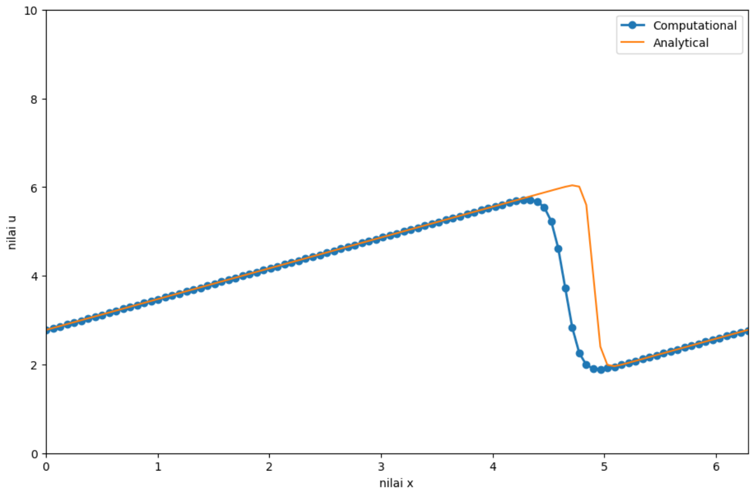
$$U_i^{n+1} = U_i^n - U_i^n \frac{7}{100}(U_i^n - U_{i-1}^n) + v\frac{14}{4\pi}(U_{i+1}^n - 2U_i^n + U_{i-1}^n) \tag{19}$$

Persamaan 19 digunakan untuk mencari solusi numerik bagi persamaan Burgers pada Persamaan 13. Pencarian solusi numerik bagi Persamaan 19 dilakukan dengan bantuan Julia. Hal yang sama juga berlaku bagi parameter N_x yang lain sehingga akan didapatkan beberapa persamaan bagi U_i^{n+1} yang berbeda.

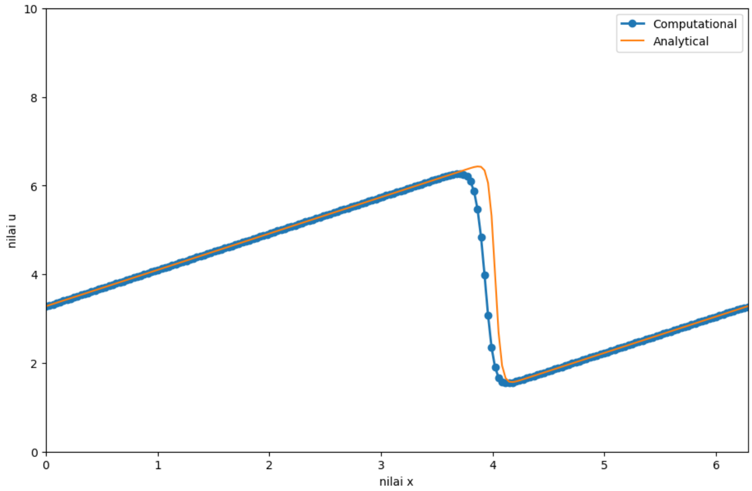
3.4 Plot Solusi Numerik pada Tiap Kasus Nilai Parameter

Plot dibuat untuk melihat gambaran dari hasil solusi numerik dari Persamaan 18 yang sudah dikerjakan. Pada penelitian ini, akan dibandingkan hasil plot solusi numerik pada Julia dengan beberapa kasus nilai N_x yang berbeda. Dibuat juga tabel hasil solusi numerik yang memuat beberapa nilai dari x untuk melihat galat dari masing-masing kasus terhadap solusi eksaknya. Hasil plot solusi numerik untuk setiap kasus nilai N_x dapat dilihat pada Gambar 1 berikut.

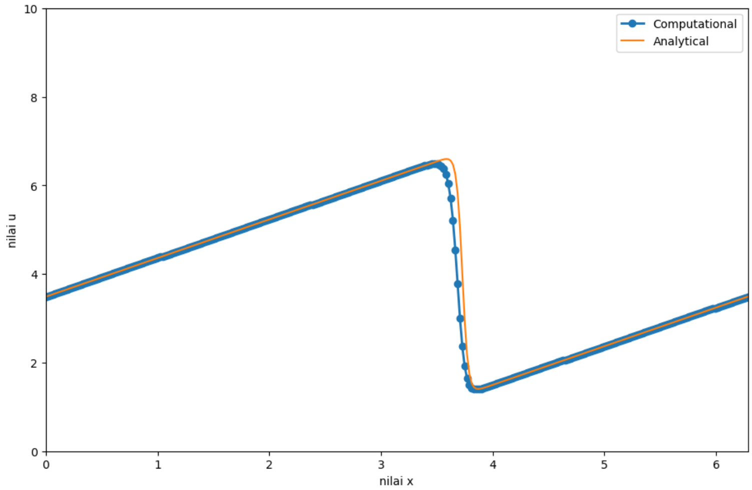
(a) $N_x = 101$



(b) $N_x = 201$



(c) $N_x = 301$



Gambar 1. Plot Hasil Solusi Numerik dengan $N_t = 100$ untuk setiap kasus nilai N_x

Berdasarkan Gambar 1 dapat terlihat perbedaan dari masing-masing plot solusi numerik dengan solusi eksaknya. Berdasarkan plot solusi numeriknya, terlihat bahwa gambar (6) memiliki plot solusi numerik yang paling mendekati plot solusi eksaknya sehingga dapat dikatakan bahwa semakin besar N_x yang digunakan maka solusi numerik yang dihasilkan akan semakin mendekati solusi eksaknya. Kemudian dengan menghitung galat (*error*) dengan metode *Mean Square Error* (MSE) untuk setiap kasus N_x, maka dapat dilihat perbandingan nilai MSE tiap kasus N_x sebagai berikut.

Tabel 1 Nilai MSE untuk setiap kasus pada N_x

Kasus N_x	Nilai MSE
101	0.491697
201	0.162538
301	0.060478

Berdasarkan hasil dari MSE yang didapat dari masing-masing kasus N_x dapat disimpulkan bahwa kasus $N_x = 301$ memberikan hasil yang paling mendekati solusi eksaknya jika dibandingkan dengan dua kasus lainnya. Berdasarkan hal tersebut dapat disimpulkan bahwa semakin besar nilai N_x, maka galat (*error*) dari solusi numerik yang didapatkan dari metode beda hingga akan semakin kecil. Lalu untuk waktu komputasi pada kasus $N_x = 101$ adalah sebesar 0.298904 detik, untuk kasus $N_x = 201$ sebesar 0.387956 detik, dan untuk kasus $N_x = 301$ sebesar 0.505140 detik. Berdasarkan hal tersebut dapat terlihat bahwa semakin besar nilai N_x yang digunakan maka semakin besar juga waktu komputasi yang dibutuhkan.

4 Simpulan dan Saran

Metode beda hingga dapat digunakan untuk mencari solusi numerik bagi persamaan Burgers. Dalam penerapannya, terlihat bahwa semakin besar nilai N_x yang digunakan, maka nilai solusi numerik yang didapat akan semakin mendekati nilai eksaknya. Berdasarkan hasil ini dapat disimpulkan juga bahwa Julia dapat digunakan untuk melakukan pencarian solusi numerik bagi persamaan Burgers.

Untuk penelitian selanjutnya, dapat dicari aplikasi lain selain Julia untuk membandingkan kecepatan dalam pencarian solusi numerik bagi persamaan Burgers. Dapat juga dicari metode numerik lain untuk mencari solusi bagi persamaan Burgers dan membandingkan galat pada masing-masing solusi numeriknya.

Daftar Pustaka

[1] Albeverio S, Korshunova A, Rozanova O. 2013. A probabilistic model associated with the pressureless gas dynamics. *Bulletin des Sciences Mathématiques*. 137(7):902–922. https://doi.org/10.1016/j.bulsci.2013.05.001.

[2] Awanda R, Oktafianto K, Arifin AZ, Fatihah N. 2019. Simulasi sebaran abu pabrik kapur menggunakan metode beda hingga. *Zeta Math Journal*. 4(2):34–39. https://doi.org/10.31102/zeta.2019.4.2.34-39.

[3] Barba L, Forsyth G. 2018. CFD Python: the 12 steps to Navier-Stokes equations. *Journal of Open Source Education*. [diakses 2022 Sept 19]. 1(9):21. http://doi.org/10.21105/jose.00021.

[4] Bonkile MP, Awasthi A, Lakshmi C, Mukundan V, Aswin VS. 2018. A systematic literature review of Burgers' equation with recent advances. *Pramana - Journal of Physics*, 90(69):1-21. https://doi.org/10.1007/s12043-018-1559-4.

[5] Clothier BE, Knight JH, White I. 1981. Burgers' equation: application to field constant-flux infiltration. *Soil Science*. 132(4):255–261. http://doi.org/10.1097/00010694-198110000-00001.

[6] Kuo CK, Lee SY. 2015. A new exact solution of Burgers' equation with linearized solution. *Mathematical Problems in Engineering*. 2015(414808):1–7. http://dx.doi.org/10.1155/2015/414808.

[7] Kasumo C. 2020. The adomian decomposition method solution of the inviscid burgers equation. *Journal of Mathematical and Computational Science*. 10(5):1834–1850. https://doi.org/10.28919/jmcs/4743.

[8] Maulidi I. 2018. Metode beda hingga untuk penyelesaian persamaan diferensial parsial masalah nilai batas dan masalah nilai awal. *OSF Preprints*. [diakses 2022 Sept 19]. 2(1):1–10. https://osf.io/q526f/download.

[9] Noviyani D, Yundari, Yudhi. 2019. Solusi persamaan difusi pada larutan gula dengan metode beda hingga. *Bimaster : Buletin Ilmiah Matematika, Statistika dan Terapannya*. 8(3):573-578. https://doi.org/10.26418/bbimst.v8i3.34026.

[10] Tiwow VA, Malago JD. 2015. Penerapan persamaan Navier-Stokes untuk kasus aliran fluida laminer pada pipa tidak horizontal. *Jurnal Sainsmat*. 4(1):51–56. https://doi.org/10.35580/sainsmat4112832015.

[11] Watanabe S, Ishiwata S, Kawamura K, Oh HG. 1997. Higher order solution of nonlinear waves. ii. shock wave described by Burgers equation. *Journal of the Physical Society of Japan*. 66(4):984–987. https://doi.org/10.1143/JPSJ.66.984