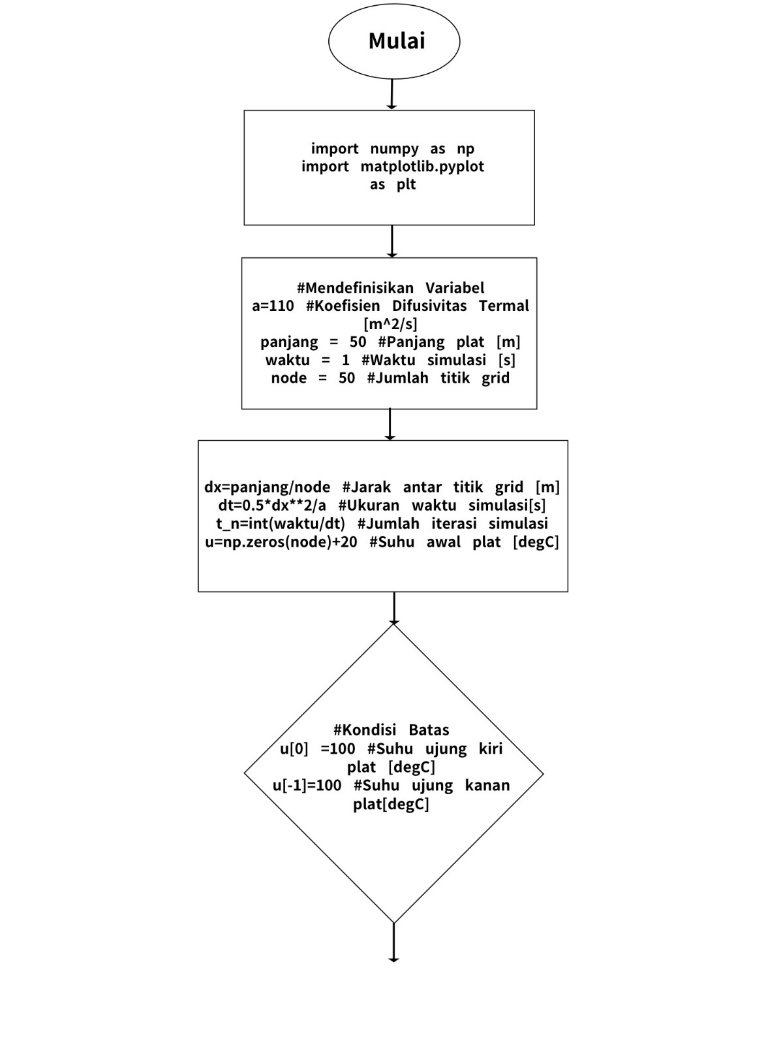
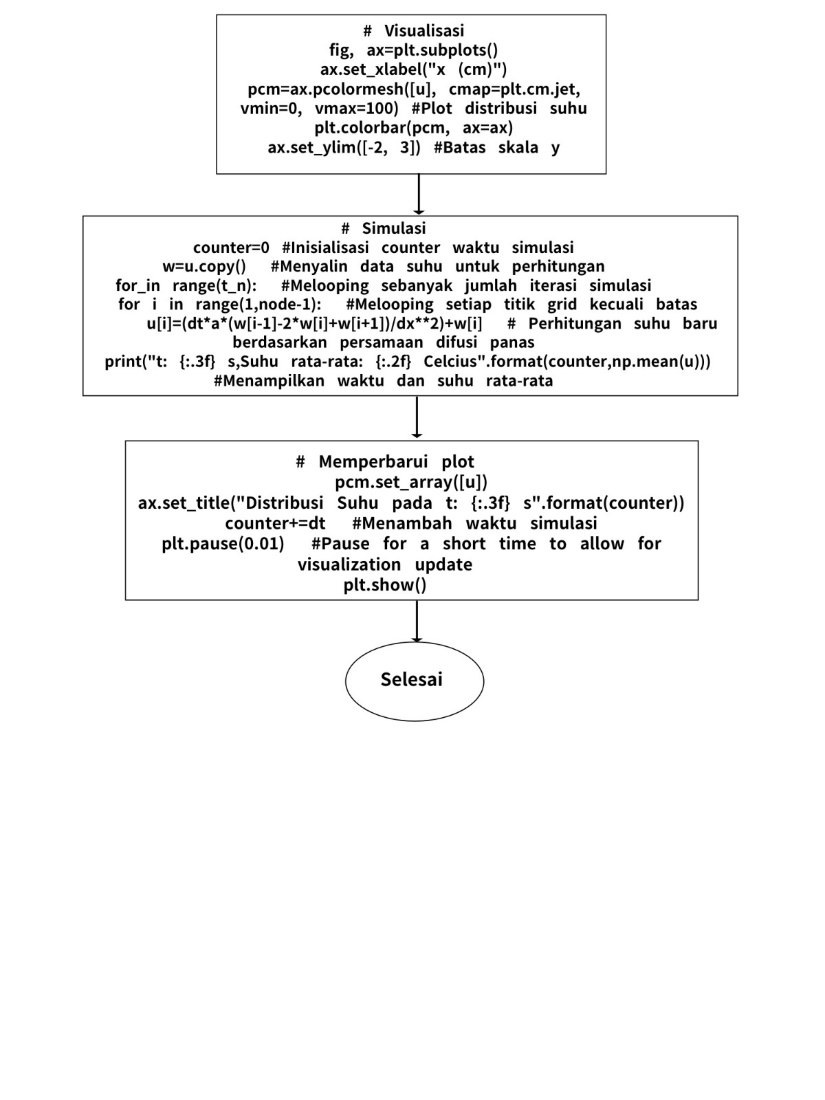
Tugas Akhir Praktikum Fisika Komputasi

Oleh : Juwita Puspita Dewi (1217030017)

1.





Penjelasan Program :

Program tersebut adalah simulasi numerik sederhana untuk menghitung dan memvisualisasikan distribusi suhu pada suatu plat. Simulasi ini menggunakan metode diferensiasi suhu (finite difference method) untuk memodelkan difusi panas pada plat dengan batas suhu yang diberikan.

Berikut adalah penjelasan langkah-langkah utama dalam program tersebut:

1. Variabel dan Parameter:

* a: Koefisien Difusivitas Termal [m^2/s].
* panjang: Panjang plat [m].
* waktu: Waktu simulasi [s].
* node: Jumlah titik grid.
* dx: Jarak antar titik grid pada sumbu x [m].
* dt: Ukuran waktu simulasi [s], dihitung berdasarkan stabilitas numerik (0.5 \* (dx^2) / a).
* t\_n: Jumlah iterasi simulasi.
* T: Array untuk menyimpan suhu pada setiap titik grid. Diinisialisasi dengan suhu awal 20°C.

2. Kondisi Batas:

* Ujung kiri plat (indeks 0) diatur dengan suhu 0°C.
* Ujung kanan plat (indeks -1) diatur dengan suhu 100°C.

3. Visualisasi:

* Membuat plot dengan menggunakan pcolormesh untuk memvisualisasikan distribusi suhu pada plat.
* Menambahkan color bar untuk skala warna pada plot.
* Menetapkan batas skala pada sumbu y.

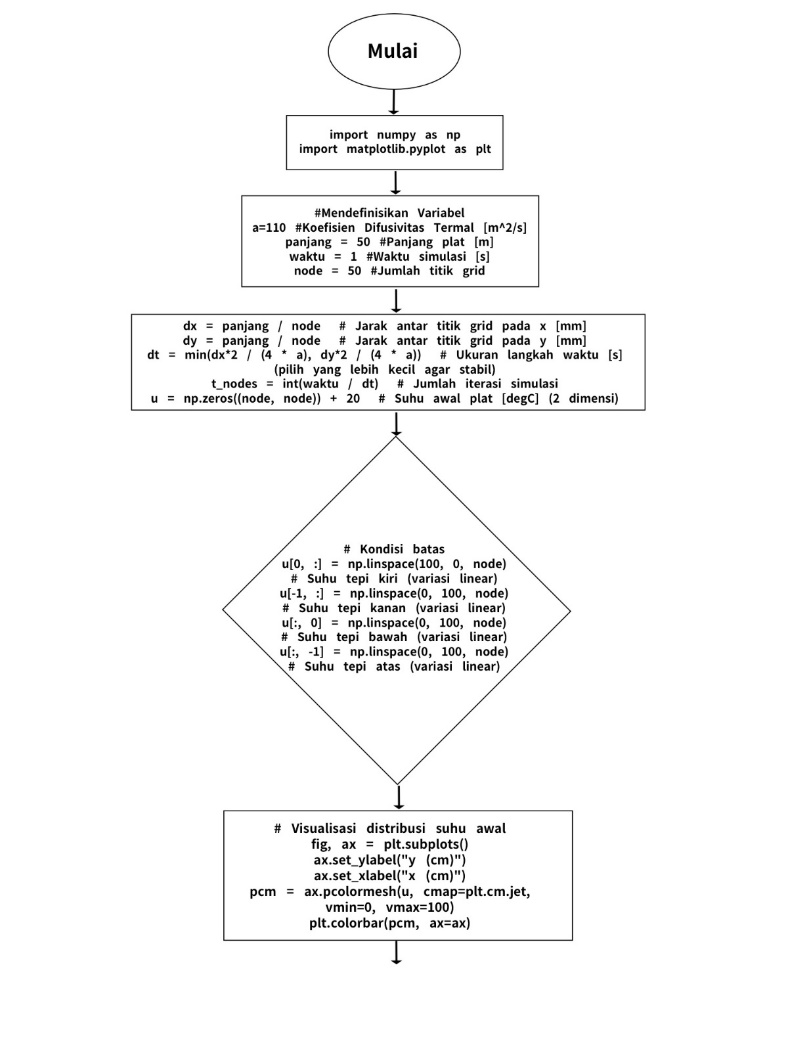
4. Simulasi:

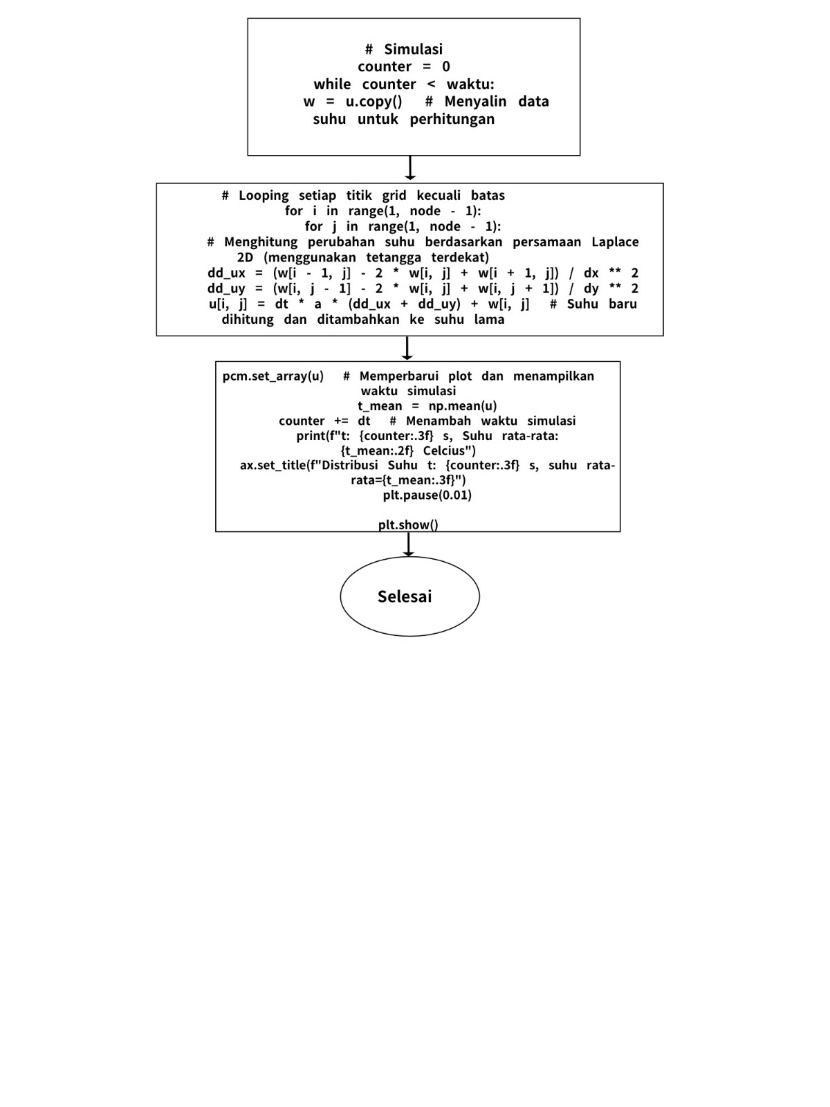
* Menggunakan loop while untuk melakukan simulasi hingga mencapai waktu simulasi yang ditentukan.
* Pada setiap iterasi, array w digunakan untuk menyimpan salinan data suhu sebelumnya.
* Melakukan loop for untuk menghitung suhu baru pada setiap titik grid kecuali batas.
* Perhitungan suhu baru dilakukan berdasarkan persamaan difusi panas menggunakan metode diferensiasi suhu.
* Memperbarui plot dan menampilkan distribusi suhu pada waktu tertentu.
* Menambahkan waktu simulasi dan suhu rata-rata pada setiap iterasi.
* Menambahkan jeda waktu (plt.pause(0.01)) untuk memperbarui visualisasi dengan interval waktu yang singkat.

5. Menampilkan Plot:

* Program akan menampilkan plot hasil simulasi setelah iterasi selesai.

Program ini memberikan pemahaman visual tentang bagaimana suhu pada suatu plat berkembang seiring berjalannya waktu, dengan memperhatikan kondisi batas yang telah ditentukan. Program ini menggunakan konsep dasar dari persamaan difusi panas pada satu dimensi.





Penjelasan Program :

Program ini merupakan simulasi numerik untuk menghitung dan memvisualisasikan distribusi suhu pada suatu plat dua dimensi. Simulasi ini menggunakan persamaan Laplace 2D dan metode diferensiasi suhu (finite difference method) untuk memodelkan perubahan suhu pada plat dengan batas suhu yang diberikan.

Berikut adalah penjelasan langkah-langkah utama dalam program tersebut:

1. Variabel dan Parameter:

* a: Koefisien Difusivitas Termal [m^2/s].
* panjang: Panjang plat [m].
* waktu: Waktu simulasi [s].
* node: Jumlah titik grid.
* dx: Jarak antar titik grid pada sumbu x [m].
* dy: Jarak antar titik grid pada sumbu y [m].
* dt: Ukuran langkah waktu [s], dihitung berdasarkan stabilitas numerik (pilih yang lebih kecil agar simulasi stabil).
* t\_nodes: Jumlah iterasi simulasi.
* u: Array dua dimensi untuk menyimpan suhu pada setiap titik grid. Diinisialisasi dengan suhu awal 20°C.

2. Kondisi Batas:

* Ujung kiri plat (kolom pertama) diatur dengan variasi linear dari 0 hingga 100.
* Ujung kanan plat (kolom terakhir) diatur dengan variasi linear dari 0 hingga 100.
* Ujung bawah plat (baris pertama) diatur dengan variasi linear dari 0 hingga 100.
* Ujung atas plat (baris terakhir) diatur dengan variasi linear dari 0 hingga 100.

3. Visualisasi Distribusi Suhu Awal:

* Membuat plot menggunakan pcolormesh untuk memvisualisasikan distribusi suhu awal pada plat.
* Menambahkan color bar untuk skala warna pada plot.

4. Simulasi:

* Menggunakan loop while untuk melakukan simulasi hingga mencapai waktu simulasi yang ditentukan.
* Pada setiap iterasi, array w digunakan untuk menyimpan salinan data suhu sebelumnya.
* Melakukan loop for untuk menghitung suhu baru pada setiap titik grid kecuali batas, berdasarkan persamaan Laplace 2D.
* Perhitungan suhu baru dilakukan menggunakan metode diferensiasi suhu.
* Memperbarui plot dan menampilkan distribusi suhu pada waktu tertentu.
* Menghitung dan mencetak suhu rata-rata pada setiap iterasi.
* Melakukan pause dan update plot setiap iterasi untuk visualisasi.

5. Menampilkan Suhu Rata-rata pada Waktu Tertentu:

* Menghitung dan mencetak suhu rata-rata pada waktu simulasi selesai.

Program ini memberikan pemahaman visual tentang bagaimana suhu pada suatu plat berkembang seiring berjalannya waktu, dengan memperhatikan kondisi batas yang telah ditentukan. Program ini menggunakan konsep dasar dari persamaan Laplace 2D untuk simulasi perubahan suhu pada suatu bidang dua dimensi.

2. Konduksi panas dalam satu dimensi dan dua dimensi adalah dua kasus yang berbeda dalam pemodelan fenomena transfer panas. Perbedaan utamanya terletak pada dimensi ruang dan kompleksitas masalah yang harus diatasi. Metode Finite Difference digunakan untuk menyelesaikan persamaan diferensial parsial yang menggambarkan konduksi panas dalam kasus-kasus ini. Berikut adalah beberapa perbedaan antara konduksi panas satu dimensi dan dua dimensi dalam penggunaan metode Finite Difference:

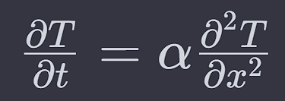
Konduksi Panas 1 Dimensi:

a. Dimensi Ruang:

* Sumbu Satu Dimensi: Konduksi panas dalam satu dimensi biasanya diwakili oleh suatu batang, plat, atau media yang panjangnya lebih signifikan daripada dimensinya yang lain.

b. Persamaan Diferensial:

* Persamaan Panas 1D: Persamaan diferensial parsial untuk konduksi panas satu dimensi biasanya berbentuk seperti ini:



c. Grid Finite Difference:

* + Grid Satu Dimensi: Pada metode finite difference, grid hanya diperlukan pada satu dimensi (biasanya sumbu x). Maka, perubahan suhu hanya bergantung pada posisi pada sumbu x.

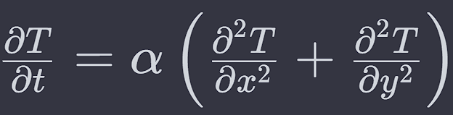
Konduksi Panas 2 Dimensi:

a. Dimensi Ruang:

* + Bidang Dua Dimensi: Konduksi panas dalam dua dimensi terjadi pada bidang atau permukaan yang memiliki dimensi panjang dan lebar.

b. Persamaan Diferensial:

- Persamaan Panas 2D:Persamaan diferensial parsial untuk konduksi panas dua dimensi umumnya berbentuk seperti ini:



c. Grid Finite Difference:

* + Grid Dua Dimensi: Pada metode finite difference untuk konduksi panas dua dimensi, grid diperlukan pada kedua sumbu x dan y. Perubahan suhu dihitung berdasarkan turunan kedua suhu terhadap kedua sumbu.

Kompleksitas Masalah:

a. Konduksi Panas 1D:

* + Sederhana:Masalah konduksi panas satu dimensi relatif lebih sederhana karena hanya ada satu dimensi ruang yang perlu diperhitungkan.

b. Konduksi Panas 2D:

* + Lebih Kompleks:Masalah konduksi panas dua dimensi lebih kompleks karena melibatkan perhitungan suhu dalam dua dimensi ruang.

Pemilihan Metode Finite Difference:

a. Konduksi Panas 1D:

* + 1D Finite Difference:

Metode finite difference pada satu dimensi melibatkan grid hanya pada sumbu x.

b. Konduksi Panas 2D:

* + 2D Finite Difference: Metode finite difference pada dua dimensi melibatkan grid pada kedua sumbu x dan y.

Dalam kedua kasus tersebut, metode Finite Difference digunakan untuk mendiskritisasi persamaan diferensial parsial menjadi bentuk yang dapat dihitung secara numerik pada suatu grid. Pemilihan dimensi dan kompleksitas masalah akan tergantung pada kondisi fisik sistem yang sedang dipelajari.

3. Program t=0 (1 Dimensi)

import numpy as np

import matplotlib.pyplot as plt

# Mendefinisikan Variabel

a = 500  # Koefisien Difusivitas Termal [m^2/s]

panjang = 2.5  # Panjang plat [m]

waktu = 1.5  # Waktu simulasi [s]

node = 50  # Jumlah titik grid

dx = panjang / node  # Jarak antar titik grid [m]

dt = 0.5 \* dx\*\*2 / a  # Ukuran waktu simulasi [s]

t\_n = int(waktu / dt)  # Jumlah iterasi simulasi

T = np.zeros(node) + 20  # Suhu awal plat [degC]

# Kondisi Batas

T[0] = 0  # Suhu ujung kiri plat [degC]

T[-1] = 100  # Suhu ujung kanan plat [degC]

# Visualisasi

fig, ax = plt.subplots()

ax.set\_xlabel("x (cm)")

pcm = ax.pcolormesh([T], cmap=plt.cm.jet, vmin=0, vmax=100)  # Plot distribusi suhu

plt.colorbar(pcm, ax=ax)

ax.set\_ylim([-2, 3])  # Batas skala y

counter = 0  # Inisialisasi counter waktu simulasi

w = T.copy()  # Menyalin data suhu untuk perhitungan

for i in range(1, node - 1):  # Melooping setiap titik grid kecuali batas

        T[i] = (dt \* a \* (w[i - 1] - 2 \* w[i] + w[i + 1]) / dx\*\*2) + w[i]  # Perhitungan suhu baru berdasarkan persamaan difusi panas

counter += dt  # Menambah waktu simulasi

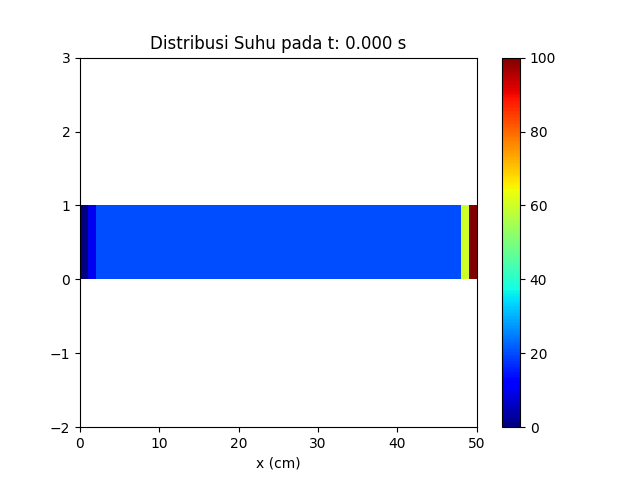
print("t: {:.3f} s, Suhu rata-rata: {:.2f} Celcius".format(counter, np.mean(T)))  # Menampilkan waktu dan suhu rata-rata

# Memperbarui plot

pcm.set\_array([T])

ax.set\_title("Distribusi Suhu pada t: {:.3f} s".format(counter))

plt.show()



Gambar Grafik saat waktu 0 sekon.

Program t=1,5 (1 dimensi)

import numpy as np

import matplotlib.pyplot as plt

# Mendefinisikan Variabel

a = 500  # Koefisien Difusivitas Termal [m^2/s]

panjang = 2.5  # Panjang plat [m]

waktu = 1.5  # Waktu simulasi [s]

node = 50  # Jumlah titik grid

dx = panjang / node  # Jarak antar titik grid [m]

dt = 0.5 \* dx\*\*2 / a  # Ukuran waktu simulasi [s]

t\_n = int(waktu / dt)  # Jumlah iterasi simulasi

T = np.zeros(node) + 20  # Suhu awal plat [degC]

# Kondisi Batas

T[0] = 0  # Suhu ujung kiri plat [degC]

T[-1] = 100  # Suhu ujung kanan plat [degC]

# Visualisasi

fig, ax = plt.subplots()

ax.set\_xlabel("x (cm)")

pcm = ax.pcolormesh([T], cmap=plt.cm.jet, vmin=0, vmax=100)  # Plot distribusi suhu

plt.colorbar(pcm, ax=ax)

ax.set\_ylim([-2, 3])  # Batas skala y

counter = 0  # Inisialisasi counter waktu simulasi

while counter < waktu :

    w = T.copy()  # Menyalin data suhu untuk perhitungan

    for i in range(1, node - 1):  # Melooping setiap titik grid kecuali batas

        T[i] = (dt \* a \* (w[i - 1] - 2 \* w[i] + w[i + 1]) / dx\*\*2) + w[i]  # Perhitungan suhu baru berdasarkan persamaan difusi panas

    print("t: {:.3f} s, Suhu rata-rata: {:.2f} Celcius".format(counter, np.mean(T)))  # Menampilkan waktu dan suhu rata-rata

    # Memperbarui plot

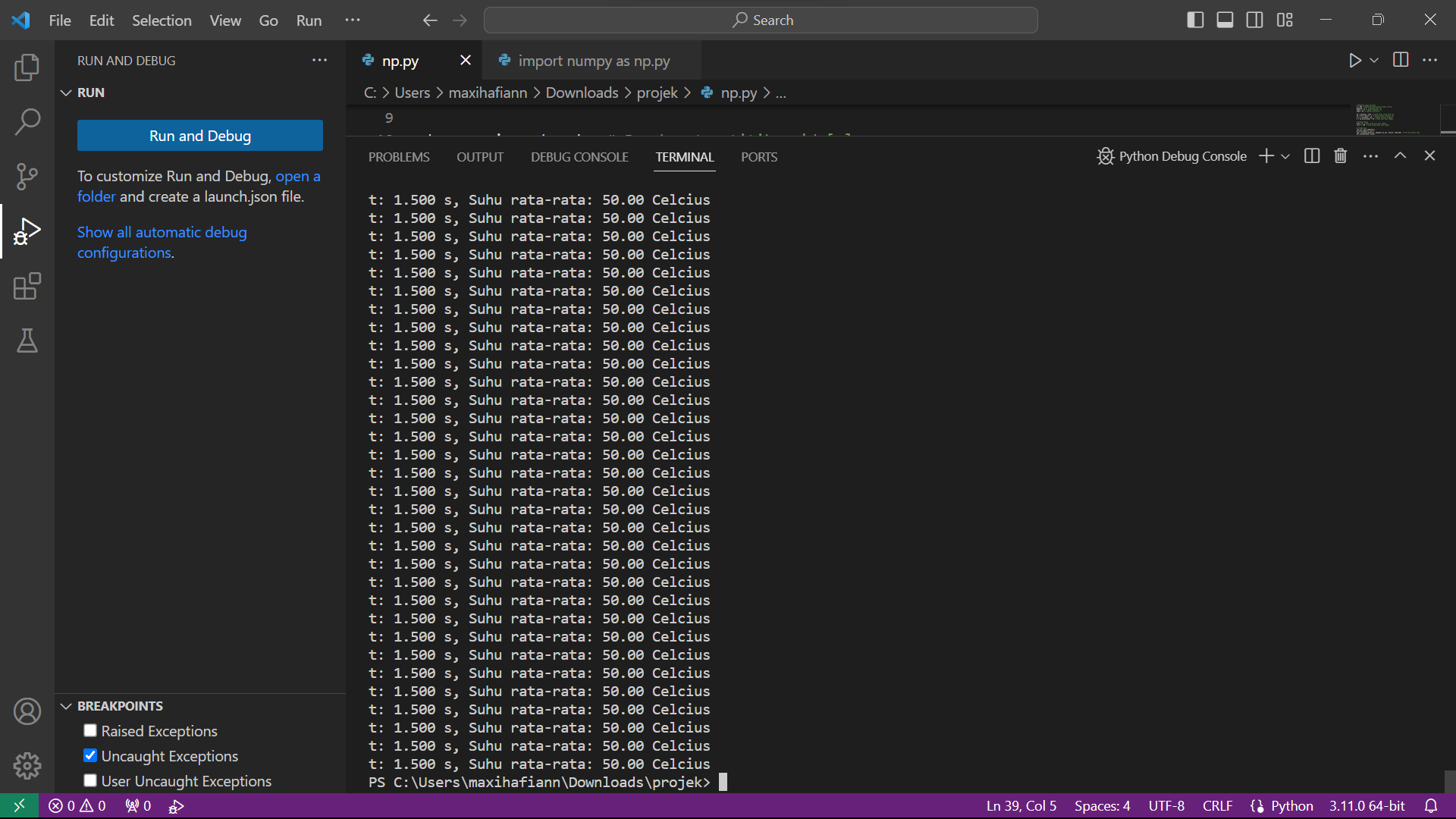
    pcm.set\_array([T])

    ax.set\_title("Distribusi Suhu pada t: {:.3f} s".format(counter))

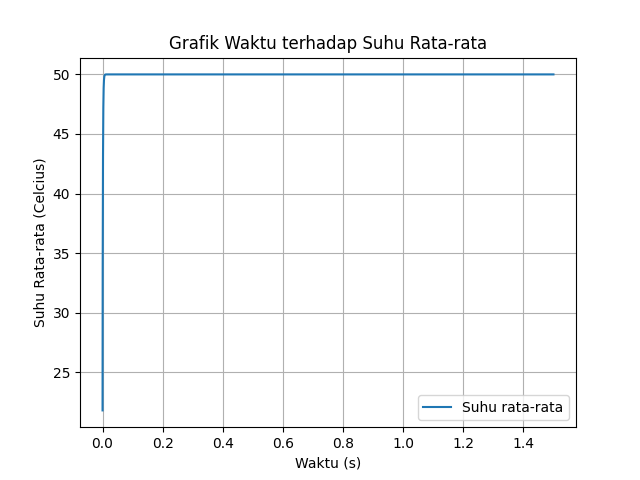
    counter += dt  # Menambah waktu simulasi

    plt.pause(0.01)  # Pause for a short time to allow for visualization update

plt.show()



Gambar suhu dan waktu.



Grafik waktu terhadap suhu.

Program t=0 (2 Dimensi) :

import numpy as np

import matplotlib.pyplot as plt

# Mendefinisikan Variabel

a = 50  # Koefisien Difusivitas Termal [m^2/s]

panjang = 0.5  # Panjang plat [m]

waktu = 1.5  # Waktu simulasi [s]

node = 50  # Jumlah titik grid

dx = panjang / node  # Jarak antar titik grid pada x [mm]

dy = panjang / node  # Jarak antar titik grid pada y [mm]

dt = min(dx \*\* 2 / (4 \* a), dy \*\* 2 / (4 \* a))  # Ukuran langkah waktu [s] (pilih yang lebih kecil agar stabil)

t\_nodes = int(waktu / dt)  # Jumlah iterasi simulasi

u = np.zeros((node, node)) + 20  # Suhu awal plat [degC] (2 dimensi)

# Kondisi batas

u[0, :] = 0   # Suhu tepi kiri (variasi linear)

u[-1, :] = 100   # Suhu tepi kanan (variasi linear)

u[:, 0] = np.linspace(0, 100, node)  # Suhu tepi bawah (variasi linear)

u[:, -1] = np.linspace(0, 100, node)  # Suhu tepi atas (variasi linear)

# Visualisasi distribusi suhu awal

fig, ax = plt.subplots()

ax.set\_ylabel("y (cm)")

ax.set\_xlabel("x (cm)")

pcm = ax.pcolormesh(u, cmap=plt.cm.jet, vmin=0, vmax=100)

plt.colorbar(pcm, ax=ax)

counter = 0

w = u.copy()  # Menyalin data suhu untuk perhitungan

# Looping setiap titik grid kecuali batas

for i in range(1, node - 1):

    for j in range(1, node - 1):

        # Menghitung perubahan suhu berdasarkan persamaan Laplace 2D (menggunakan tetangga terdekat)

        dd\_ux = (w[i - 1, j] - 2 \* w[i, j] + w[i + 1, j]) / dx \*\* 2

        dd\_uy = (w[i, j - 1] - 2 \* w[i, j] + w[i, j + 1]) / dy \*\* 2

        u[i, j] = dt \* a \* (dd\_ux + dd\_uy) + w[i, j]  # Suhu baru dihitung dan ditambahkan ke suhu lama

t\_mean = np.mean(u)

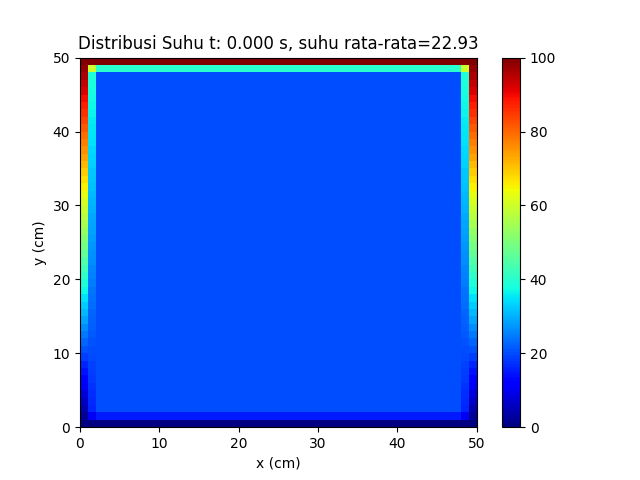
print(f"t: {counter:.3f} s, Suhu rata-rata: {t\_mean:.2f} Celcius")

# Memperbarui plot dan menampilkan waktu simulasi

pcm.set\_array(u)

ax.set\_title(f"Distribusi Suhu t: {counter:.3f} s, suhu rata-rata={t\_mean:.2f}")

plt.show()



Grafik saat waktu 0 sekon.

Program t=1,5 (2 Dimensi) :

import numpy as np

import matplotlib.pyplot as plt

# Mendefinisikan Variabel

a = 50  # Koefisien Difusivitas Termal [m^2/s]

panjang = 0.5  # Panjang plat [m]

waktu = 1.5  # Waktu simulasi [s]

node = 50  # Jumlah titik grid

dx = panjang / node  # Jarak antar titik grid pada x [mm]

dy = panjang / node  # Jarak antar titik grid pada y [mm]

dt = min(dx \*\* 2 / (4 \* a), dy \*\* 2 / (4 \* a))  # Ukuran langkah waktu [s] (pilih yang lebih kecil agar stabil)

t\_nodes = int(waktu / dt)  # Jumlah iterasi simulasi

u = np.zeros((node, node)) + 20  # Suhu awal plat [degC] (2 dimensi)

# Kondisi batas

u[0, :] = 0   # Suhu tepi kiri (variasi linear)

u[-1, :] = 100   # Suhu tepi kanan (variasi linear)

u[:, 0] = np.linspace(0, 100, node)  # Suhu tepi bawah (variasi linear)

u[:, -1] = np.linspace(0, 100, node)  # Suhu tepi atas (variasi linear)

# Visualisasi distribusi suhu awal

fig, ax = plt.subplots()

ax.set\_ylabel("y (cm)")

ax.set\_xlabel("x (cm)")

pcm = ax.pcolormesh(u, cmap=plt.cm.jet, vmin=0, vmax=100)

plt.colorbar(pcm, ax=ax)

counter = 0

w = u.copy()  # Menyalin data suhu untuk perhitungan

# Looping setiap titik grid kecuali batas

for i in range(1, node - 1):

    for j in range(1, node - 1):

        # Menghitung perubahan suhu berdasarkan persamaan Laplace 2D (menggunakan tetangga terdekat)

        dd\_ux = (w[i - 1, j] - 2 \* w[i, j] + w[i + 1, j]) / dx \*\* 2

        dd\_uy = (w[i, j - 1] - 2 \* w[i, j] + w[i, j + 1]) / dy \*\* 2

        u[i, j] = dt \* a \* (dd\_ux + dd\_uy) + w[i, j]  # Suhu baru dihitung dan ditambahkan ke suhu lama

t\_mean = np.mean(u)

print(f"t: {counter:.3f} s, Suhu rata-rata: {t\_mean:.2f} Celcius")

# Memperbarui plot dan menampilkan waktu simulasi

pcm.set\_array(u)

ax.set\_title(f"Distribusi Suhu t: {counter:.3f} s, suhu rata-rata={t\_mean:.2f}")

plt.show()