

```
Untitled3.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks

import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage import convolve, generate_binary_structure

# Ukuran grid
N = 100

# Inisialisasi grid dengan suhu awal
grid = np.zeros((N, N)) + 0.5

# Kondisi awal - area 40x40 dipanaskan pada z=40 dan didinginkan pada z=90
grid[30:70, 30:70, 40] = 1 # Sisi panas
grid[30:70, 30:70, 90] = 0 # Sisi dingin

# Mask untuk posisi area tetap panas (1) dan dingin (0)
mask_pos = grid == 1
mask_neg = grid == 0

# Membuat kernel konvolusi
kern = generate_binary_structure(3, 1).astype(float) / 6
kern[1, 1, 1] = 0

# Fungsi untuk kondisi batas Neumann
def neumann(a):
    a[0, :, :] = a[1, :, :]; a[-1, :, :] = a[-2, :, :]
    a[:, 0, :] = a[:, 1, :]; a[:, -1, :] = a[:, -2, :]
    a[:, :, 0] = a[:, :, 1]; a[:, :, -1] = a[:, :, -2]
    return a

# Variabel untuk menyiapkan error
err = []
```

```
Untitled3.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan

+ Kode + Teks

# Variabel untuk menyiapkan error
err = []

# Jumlah iterasi
iters = 2000

# Iterasi untuk memperbarui grid menggunakan konvolusi
for i in range(iters):
    # Update grid menggunakan konvolusi
    grid_updated = convolve(grid, kern, mode='constant')

    # Menerapkan kondisi batas (Neumann)
    grid_updated = neumann(grid_updated)

    # Menerapkan kondisi batas (Dirichlet)
    grid_updated[mask_pos] = 1
    grid_updated[mask_neg] = 0

    # Menghitung error antara grid sebelumnya dan yang diperbarui
    err.append(np.mean((grid - grid_updated) ** 2))

    # Memperbarui grid dengan hasil konvolusi
    grid = grid_updated

# Menampilkan hasil dalam bentuk kontur suhu pada irisan z=40
slc = 40
plt.figure(figsize=(6, 5))
CS = plt.contour(np.arange(N) / N, np.arange(N) / N, grid[slc], levels=40)
plt.clabel(CS, CS.levels, inline=True, fontsize=6)
plt.xlabel('$z/z_0$')
plt.ylabel('$y/y_0$')
```



