

TUGAS PRAKTIKUM FISIKA KOMPUTASI
“PERSAMAAN LAPLACE POTENSIAL LISTRIK”

A. Kode program “Persamaan Laplace Potensial Listrik”

```
import numpy as np
import matplotlib.pyplot as plt
#plt.style.use(['science','notebook'])
from scipy.ndimage import convolve, generate_binary_structure

N = 100
grid = np.zeros((N,N,N))+0.5

grid[30:70,30:70,40] = 1
grid[30:70,30:70,90] = 0
mask_pos = grid==1
mask_neg = grid==0

yv, xv, zv = np.meshgrid(np.arange(N),np.arange(N),np.arange(N))
#grid = 1-zv/100

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

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

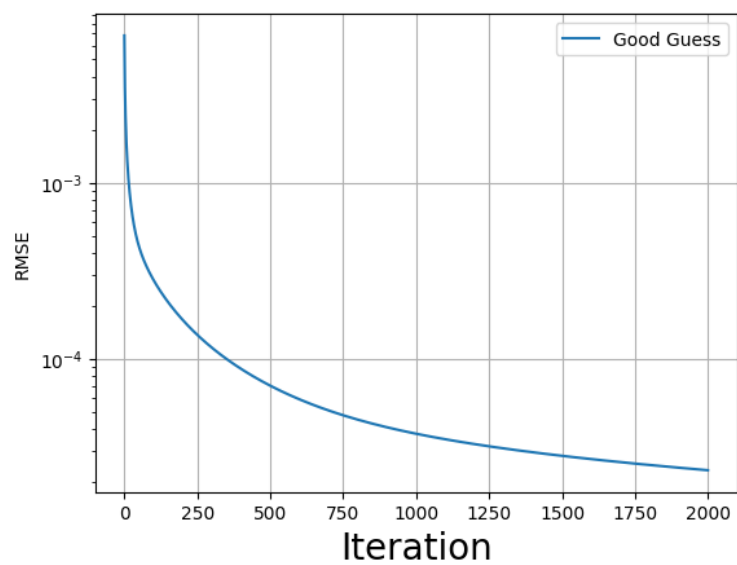
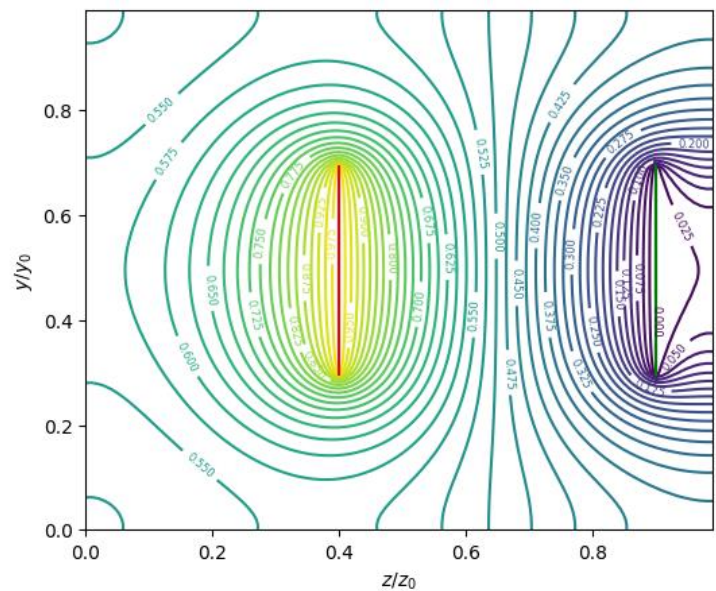
err = []
iters = 2000          #pengulangan
for i in range(iters):
    grid_updated = convolve(grid,kern, mode='constant')
    # Boundary conditions (neumann)
    grid_updated = neumann(grid_updated)
    # Boundary conditions (dirchlett)
    grid_updated[mask_pos] = 1
    grid_updated[mask_neg] = 0
    # See what error is between consecutive arrays
    err.append(np.mean((grid-grid_updated)**2))
    grid = grid_updated

slc = 40

plt.figure(figsize=(6,5)) #plotting grafik
CS = plt.contour(np.arange(100)/100, np.arange(100)/100,
grid[slc],
```

```
levels=40)
plt.clabel(CS, CS.levels, inline=True, fontsize=6)
plt.xlabel('$z/z_0$')
plt.ylabel('$y/y_0$')
plt.axvline(0.4, ymin=0.3, ymax=0.7, color='r')
plt.axvline(0.9, ymin=0.3, ymax=0.7, color='g')
plt.show()
plt.semilogy(np.sqrt(np.array(err)), label='Good Guess')
plt.legend()
plt.xlabel('Iteration', fontsize=20)
plt.ylabel(r'RMSE')
plt.grid()
```

B. Grafik yang dihasilkan



C. Algoritma pemrograman

1. Mengimport library yang diperlukan, yaitu numpy untuk kode program yang mengandung unsur matematis, matplotlib.pyplot untuk memunculkan grafik, serta scipy.ndimage untuk memanggil fungsi konvolusi yang akan digunakan.
2. Menentukan ukuran grid (N) dalam hal ini adalah kubus berukuran 100x100x100.
3. Membuat array numpy 3 dimensi dengan ukuran (N, N, N) dengan inisialisasi semua elemennya sebesar +0,5.
4. Menentukan nilai grid dimana 1 untuk suhu tinggi (panas) dengan nilai z sebesar 40 dan 0 untuk suhu rendah (dingin) dengan nilai z sebesar 90.
5. Memberikan masking dimana mask_pos akan bernilai True jika elemennya bernilai 1 (False jika 0), serta mask_neg akan bernilai True jika elemennya bernilai 0 (False jika 1).
6. Membuat kernel konvolusi dimana nilai 1/6 pada perintah akhir digunakan untuk menormalkan kernel sehingga jumlah semua elemennya adalah 1
7. Mendefinisikan fungsi Neumann untuk mengaplikasikan fungsi Laplace.
8. Melakukan iterasi/pengulangan sebanyak 2000 kali agar hasil yang didapat lebih akurat.
9. Melakukan beberapa tahapan konvolusi.
10. Menentukan nilai slc yang merupakan sebuah persegi berukuran 40x40 yang dipanaskan.
11. Melakukan plotting untuk grafik yang akan ditampilkan.
12. Berdasarkan grafik yang didapatkan, dapat diketahui bahwa distribusi suhu pada pelat logam tersebut tidak merata. Hal tersebut ditandai dengan kontur garis berwarna hijau kekuningan lebih mendominasi pada grafik tersebut. Hal ini menandakan bahwa suhu panas berpindah lebih cepat dibandingkan dengan suhu dingin. Bentuk kontur garis yang melengkung menunjukkan bahwa suhu panas menyebar secara bertahap dari daerah panas ke daerah dingin.
13. Berdasarkan grafik kedua yang didapatkan, dapat diketahui bahwa semakin tinggi nilai iterasi/pengulangannya maka akan semakin kecil pula nilai error yang didapatkannya. Hal ini menunjukkan bahwa nilai yang didapatkan pada iterasi yang tinggi akan semakin akurat.