

# Mencari Nilai Minimum Global Dengan Metode *Simulated Annealing*

Irfan Dwi Prakoso – 1301160164

Program Studi Teknik Informatika, Fakultas Informatika  
Universitas Telkom Bandung  
Jl. Telekomunikasi Dayeuhkolot, Bandung 40257

## I. Masalah

Diberikan fungsi sebagai berikut :

$$f(x_1, x_2) = - \left| \sin(x_1) \cos(x_2) \exp \left( \left| 1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right|$$

Mencari nilai minimum global dari fungsi tersebut dengan batasan  $-10 \leq x_1 \leq 10$  dan  $-10 \leq x_2 \leq 10$ , menggunakan metode pencarian probabilistik *Simulated Annealing* (SA), yaitu metode pencarian menggunakan analogi pendinginan bahan logam panas secara perlahan.

## II. Analisis Penyelesaian Masalah

Metode *Simulated Annealing* memiliki keunggulan yaitu dapat memungkinkan nilai minimum tidak terjebak dalam minimum lokal fungsi. SA menggunakan rumus probabilitas yang dapat mengeluarkan pencarian dari jebakan minimum lokal. Step-step SA yang digunakan dalam *source code* diantaranya adalah :

1. Menginisialisasi T (temperature/suhu), c (factor penurunan suhu), n (jumlah iterasi).
2. Membangkitkan dua nilai  $(x_1, x_2)$  secara random, kemudian dimasukkan ke dalam rumus fungsi untuk menghasilkan  $f_1$ .
3. Jadikan  $f_1$  sebagai nilai minimum.
4. Membangkitkan dua nilai baru dengan batasan yang dikecilkan sebesar  $\pm 6$ , tetapi jika batasan bawah melebihi -10 maka batas bawah diset menjadi -10 dan batas atas diset menjadi 2, jika batasan atas melebihi 10 maka batas atas diset menjadi 10 dan batas bawah diset menjadi -2, agar range batas tetap  $\pm 6$ . Kemudian, dua nilai tersebut dimasukkan ke dalam rumus fungsi untuk menghasilkan  $f_2$ . Kemudian dicari selisih  $f_1$  dan  $f_2$  ( $\Delta f$ ).
5. Jika delta menghasilkan negative ( $f_2$  lebih kecil dari  $f_1$ ) maka nilai baru akan diterima. Jika delta bernilai positif maka ditinjau menggunakan *Metropolis Criterion*, yaitu dengan cara membangkitkan suatu nilai random dengan batas (0,1), kemudian mencari nilai P dengan rumus  $P[X_2] = e^{-\Delta f / kT}$ , dengan  $k = 1$ . Jika nilai random lebih kecil dari P maka nilai baru diterima.
6. Jika nilai yang diterima lebih kecil dari nilai minimum sekarang, maka nilai yang diterima terbaru menjadi nilai minimum
7. Kembali ke step 3, lakukan sebanyak n kali.
8. Kalikan T dengan c dan kembali ke step 3 hingga nilai T kecil.

Penentuan parameter yang mempengaruhi nilai minimum diantaranya :

1. T / Temperatur

Nilai awal temperatur dan batas terkecil temperatur akan mempengaruhi banyaknya iterasi yang akan dilakukan dan juga mempengaruhi probabilitas *Metropolis Criterion* diterima atau tidak, karena ketika temperatur masih besar, nilai P juga akan besar.

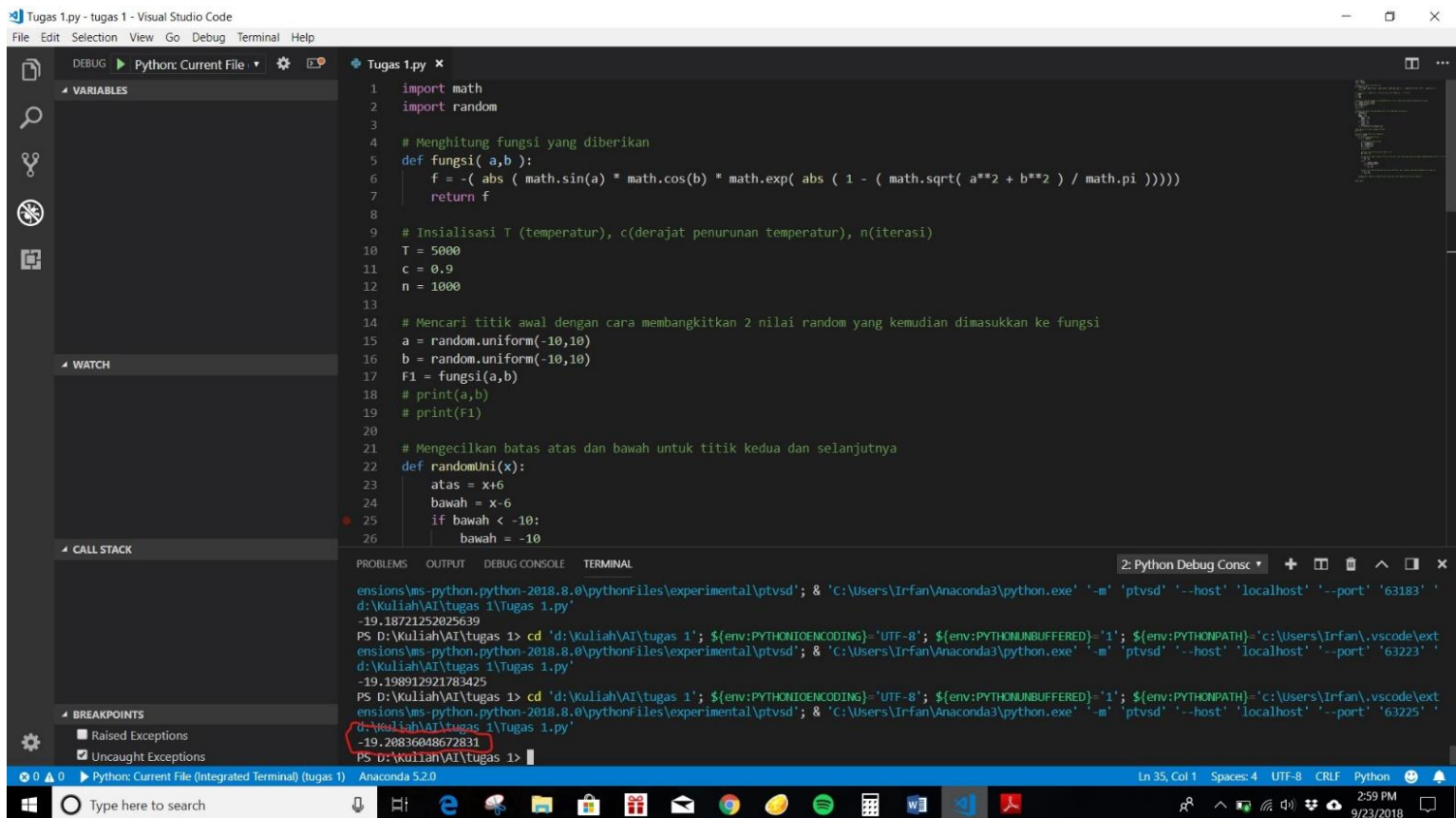
2. c / Faktor penurunan temperature

c dapat mempengaruhi banyaknya iterasi yang akan dilakukan. Jika nilai c semakin dekat dengan 1, maka semakin banyak iterasi yang bisa dilakukan.

3. n / Banyak iterasi yang dilakukan selama temperatur belum diturunkan

n dapat mempengaruhi banyaknya iterasi yang akan dilakukan. Semakin besar nilai n maka semakin besar iterasi yang akan dilakukan

### III. Hasil Terbaik dan Akurasi SA



The screenshot displays a Visual Studio Code editor with a Python file named 'Tugas 1.py'. The code implements a Simulated Annealing algorithm. It includes a function 'fungsi(a,b)' to calculate a fitness value, a 'randomUni(x)' function to generate random values within a specified range, and a main execution block that initializes parameters (T=5000, c=0.9, n=1000), generates initial random values for 'a' and 'b', and iterates the algorithm. The terminal output shows the execution of the script, with the final fitness value '-19.28836048672831' highlighted in red, indicating the best result achieved.

$$\text{Akurasi SA} = \left(1 - \left| \frac{f_A - f_R}{f_R} \right| \right) \times 100\% = \left(1 - \left| \frac{19.2083 - 19.2805}{19.2805} \right| \right) \times 100\% = 99.9999\%$$

### Referensi

- [1] <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/learn-43/lib/photoz/.g/web/glossary/anneal.html>
- [2] Suyanto. 2014. Artificial Intelligence. Revisi Kedua. Bandung : Penerbit Informatika.
- [3] Rao, Singiresu S. 2009. Engineering Optimization - Theory and Practice. Fourth Edition. New Jersey : John Wiley & Sons, Inc.