

# LAPORAN TUGAS 1 – SIMULATED ANNEALING

Nama: Muhammad Rizky Pahlawan

NIM: 1301178549

Kelas: IF-41-GAB05

---

## Soal

Lakukan analisis, desain, dan implementasi algoritma *Simulated Annealing* (SA) ke dalam suatu program komputer untuk menemukan **nilai minimum** dari fungsi:

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

dengan batasan  $-10 \leq x_1 \leq 10$  dan  $-10 \leq x_2 \leq 10$ .

## Analisis

Diketahui sebuah fungsi sebagai berikut:

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

Untuk mencari nilai minimum dari fungsi tersebut dengan batasan  $-10 \leq x_1 \leq 10$  dan  $-10 \leq x_2 \leq 10$ , digunakan salah satu algoritma dari *informed search*, yaitu *Simulated Annealing*. Singkatnya, *Simulated Annealing* merupakan modifikasi dari algoritma *Hill Climbing*. Modifikasi tersebut merupakan penambahan suatu probabilitas ‘coba-coba’ dengan harapan terhindar dari kemungkinan terjebak di dalam *local minimal*.

Terdapat 10 langkah untuk menjalankan algoritma *Simulated Annealing*. Di antara langkah-langkah tersebut, terdapat beberapa langkah yang dinilai sangat vital atau sangat menentukan hasil akhir yang ingin dicapai, yaitu *Best So Far* (BSF). Langkah-langkah tersebut antara lain: inisiasi nilai awal dari  $x_1$  dan  $x_2$ , inisiasi nilai awal T, pemilihan teknik pengurangan T hingga T bernilai 0, serta pemilihan teknik untuk *generate*  $x_1$  baru dan  $x_2$  baru. Adapun rincian dari langkah-langkah tersebut akan dijabarkan pada bagian strategi.

Kemudian, untuk menyelesaikan permasalahan di atas, saya memilih menggunakan bahasa pemrograman *Python*.

## Strategi

Sebagaimana yang telah disinggung sebelumnya, terdapat 4 langkah yang sangat menentukan dalam pemecahan masalah di atas. Adapun rinciannya sebagai berikut.

1. Inisiasi nilai awal dari  $x_1$  dan  $x_2$   
Walaupun biasanya inisiasi awal suatu nilai adalah 0, namun pada kasus ini nilai awal dari  $x_1$  dan  $x_2$  dapat ditentukan secara bebas (*random*). Saya memilih untuk tidak

banyak berspekulasi untuk nilai awal dengan memberi nilai 1 untuk  $x_1$  dan nilai -1 untuk  $x_2$ .

2. Inisiasi nilai awal T

Nilai T merujuk pada nilai *Temperature*, dimana besaran nilai tersebut akan berkontribusi dalam proses pengulangan. Disini saya memilih untuk memberikan nilai  $T=1000$

3. Pemilihan teknik pengurangan T

Awalnya saya menggunakan teknik pengurangan  $T=T-1$  (T dikurangi 1 untuk setiap pengulangan). Hal ini berarti program akan menjalankan pengulangan sebanyak 1000 kali. Kemudian, saya mencoba untuk menggunakan teknik pengurangan  $T=T-0.1$  (T dikurangi 0.1 untuk setiap pengulangan). Hal tersebut mengakibatkan program akan menjalankan pengulangan 10 kali lebih banyak. Walaupun dalam pengaplikasiannya teknik ini memakan waktu lebih lama dari teknik sebelumnya, namun keluaran yang dihasilkan pun lebih baik.

4. Pemilihan teknik untuk men-*generate*  $x_1$  baru dan  $x_2$  baru

Pada langkah ini, diperlukan kejelian dan kreativitas untuk mendapatkan suatu teknik untuk men-*generate*  $x_1$  baru dan  $x_2$  baru. Hal ini agar nilai yang dieksplorasi oleh program lebih beragam sehingga dapat mendekati kepada nilai minimum global dari fungsi. Disini saya memilih teknik eksplorasi sebagai berikut.

- Nilai  $x_1$  baru =  $x_1$  awal + x, dimana x merupakan nilai random dari  $(0 < x < 1)$ . Apabila hasil penjumlahannya bernilai lebih dari 10, maka nilai  $x_1$  baru = -10.
- Nilai  $x_2$  baru =  $x_2$  awal - x, dimana x merupakan nilai random dari  $(0 < x < 1)$ . Apabila hasil penjumlahannya bernilai kurang dari -10, maka nilai  $x_2$  baru = 10.

### Hasil *running* terbaik

Setelah melakukan 10 langkah dalam algoritma *Simulated Annealing*, didapati hasil sebagai berikut.

#### 10. Return BSF sebagai solusi

```
In [34]: #print nilai x1 dan x2 yang tersimpan sebagai BSF setelah hasil pengulangan
print(BSF)
```

```
[8.059246422076294, 9.664668454396413]
```

```
In [35]: #print nilai Energy BSF
print(e_bsf)
```

```
-19.20832649235877
```

Nilai BSF menyimpan nilai  $x_1 = 8,059..$  dan  $x_2 = 9,664..$ , sehingga menghasilkan energi BSF sebesar -19,208...

Jika melihat kepada nilai  $E_0$  (Energi awal) yang sebesar -0,787..., maka dapat dihitung nilai Delta E akhir sebesar:  $\Delta E = E_{BSF} - E_{awal} = -19,208.. - (-0,787..) = -18,420...$  Maka, dapat disimpulkan bahwa nilai  $\Delta E$  bersifat bagus dan teknik yang digunakan cukup optimal.