

# LAPORAN TUGAS PROGRAM AI 1

## SIMULATED ANNEALING PADA FUNGSI

**CCH3F3**

**ARTIFICIAL INTELLIGENCE**

**SEMESTER GANJIL 2017/2018**

**S1 TEKNIK INFORMATIKA**



Aldika Wicaksono

1301150049

IF – 39 – 07

## Deskripsi Masalah

Pada studi kasus ini, program akan mencari nilai fungsi optimal minimal dari fungsi dengan menggunakan algoritma *Simulated Annealing*. *Simulated Annealing* adalah teknik pencarian acak dengan memanfaatkan analogi proses pembekuan kristal dalam suatu materi. Dan seiring penurunan suhu, kristal tersebut menguat (*Annealing*).

Dalam kasus ini, kita mencari nilai fungsi optimal minimal dengan algoritma *Simulated Annealing* dari fungsi :

$$f(x_1, x_2) = \left(4 - 2,1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$$

Dengan batasan  $-10 \leq x_1 \leq 10$  dan  $-10 \leq x_2 \leq 10$  dimana  $x^1$  dan  $x^2$  adalah bilangan *real*.

## Rancangan Metode

Logika *Simulated Annealing* dalam kasus ini :

- Evaluasi **Initial state** dengan mengeluarkan nilai variabel  $x^1$  dan  $x^2$  secara acak. Kemudian masukkan nilai variabel tersebut ke dalam fungsi. Apabila state tersebut **Goal state**. Maka keluar. Sebaliknya, Jadikan fungsi **Initial state** itu menjadi **Current state**.
- Inisialisai nilai **Temperature** dengan nilai besar. (Penjadwalan *Annealing*).
- Inisialisasi nilai fungsi **Best-so-Far state** dengan nilai fungsi **Current state**.
- Ulangi proses berikut hingga nilai **Temperature** mencapai 0 **atau** nilai fungsi sudah mencapai **Optimal minimal** :
  - o Keluarkan nilai variabel  $x^1$  dan  $x^2$  secara acak. Kemudian masukkan nilai variabel tersebut ke dalam fungsi.
  - o Jika **New state** memiliki nilai lebih kecil dari nilai **Current state**, maka substitusi nilai **Current state** dengan new state.
  - o Sebaliknya, Jika nilai **New state** lebih besar dari nilai **Current state**, maka keluarkan **bilangan acak** dari 0 sampai 1, maka substitusi nilai **Current state** dengan **New state** menggunakan rumus probabilitas  $p(\Delta E) = e^{-\Delta E/T}$  dimana  $(\Delta E) = f(\text{new state}) - f(\text{current state})$ .
  - o Jika hasil perbandingan  $p(\Delta E) > \text{bilangan acak}$  0..1 maka substitusi nilai **Current state** dengan **New state**
  - o Turunkan nilai **Temperature** dengan rumus  $\text{Temperature} = \text{Temperature} * \alpha$  dimana nilai  $\alpha$  berada di **rentang 0.8 s.d 0.99**

Keluarkan nilai **Best-so-Far state** sebagai **solusi**.

Dalam implementasi kasus ini, saya menggunakan bahasa pemrograman **Python** karena mudah untuk merancang rumus fungsi ke bahasa program.

## Hasil dan *Screenshot Output Program*

### Hasil Output Program Pertama

C:\Python27\python.exe C:/Users/aldik/Desktop/Tupro#1\_Aldika\_Wicaksono.py

~~~Hasil Pencarian Nilai Optimal Minimum dengan Simulated Annealing~~~

Tentukan Nilai Optimal Minimum dari Fungsi :

$$f(x_1, x_2) = [4 - 2.1x_1^2 + x_1^4/3] x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

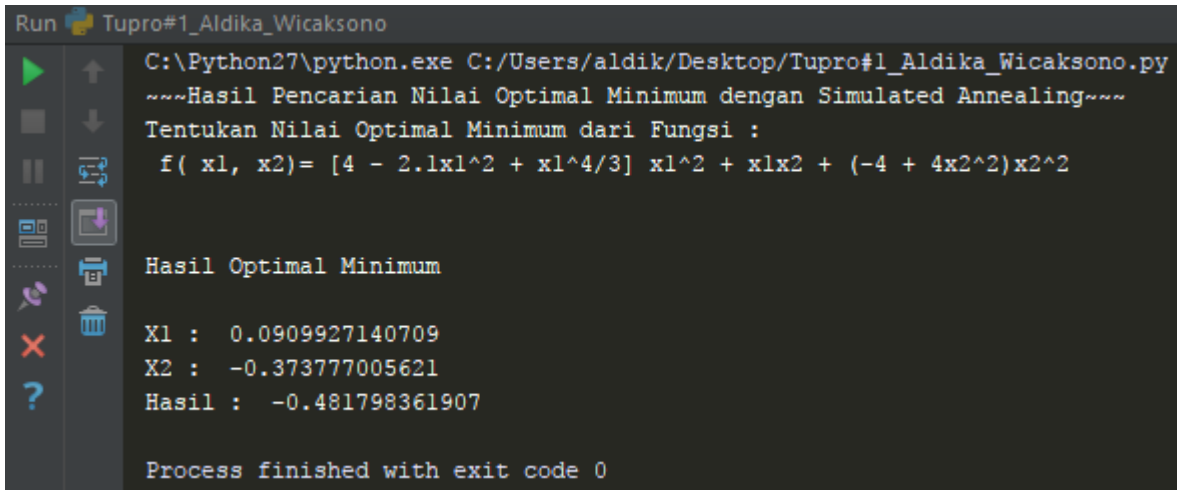
Hasil Optimal Minimum

X1 : 0.0909927140709

X2 : -0.373777005621

Hasil : -0.481798361907

Process finished with exit code 0



```
Run Tupro#1_Aldika_Wicaksono
C:\Python27\python.exe C:/Users/aldik/Desktop/Tupro#1_Aldika_Wicaksono.py
~~~Hasil Pencarian Nilai Optimal Minimum dengan Simulated Annealing~~~
Tentukan Nilai Optimal Minimum dari Fungsi :
f( x1, x2)= [4 - 2.1x1^2 + x1^4/3] x1^2 + x1x2 + (-4 + 4x2^2)x2^2

Hasil Optimal Minimum

X1 : 0.0909927140709
X2 : -0.373777005621
Hasil : -0.481798361907

Process finished with exit code 0
```

## Hasil Output Program Kedua

C:\Python27\python.exe C:/Users/aldik/Desktop/Tupro#1\_Aldika\_Wicaksono.py

~~~Hasil Pencarian Nilai Optimal Minimum dengan Simulated Annealing~~~

Tentukan Nilai Optimal Minimum dari Fungsi :

$$f(x_1, x_2) = [4 - 2.1x_1^2 + x_1^4/3] x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

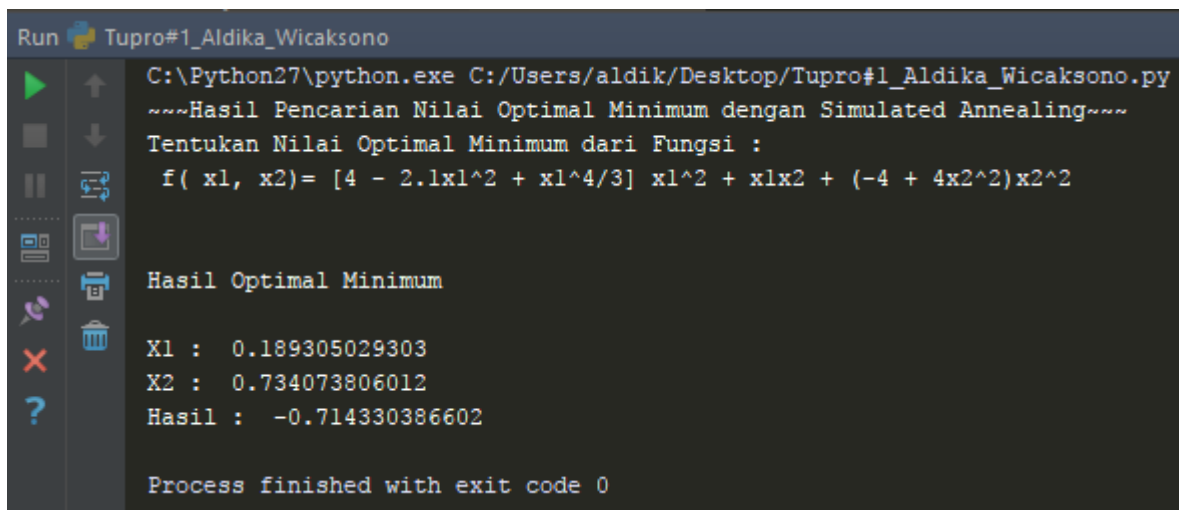
Hasil Optimal Minimum

X1 : 0.189305029303

X2 : 0.734073806012

Hasil : -0.714330386602

Process finished with exit code 0



```
Run Tupro#1_Aldika_Wicaksono
C:\Python27\python.exe C:/Users/aldik/Desktop/Tupro#1_Aldika_Wicaksono.py
~~~Hasil Pencarian Nilai Optimal Minimum dengan Simulated Annealing~~~
Tentukan Nilai Optimal Minimum dari Fungsi :
  f( x1, x2)= [4 - 2.1x1^2 + x1^4/3] x1^2 + x1x2 + (-4 + 4x2^2)x2^2

Hasil Optimal Minimum
X1 : 0.189305029303
X2 : 0.734073806012
Hasil : -0.714330386602

Process finished with exit code 0
```

### Hasil Output Program Ketiga

C:\Python27\python.exe C:/Users/aldik/Desktop/Tupro#1\_Aldika\_Wicaksono.py

~~~Hasil Pencarian Nilai Optimal Minimum dengan Simulated Annealing~~~

Tentukan Nilai Optimal Minimum dari Fungsi :

$$f(x_1, x_2) = [4 - 2.1x_1^2 + x_1^4/3] x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

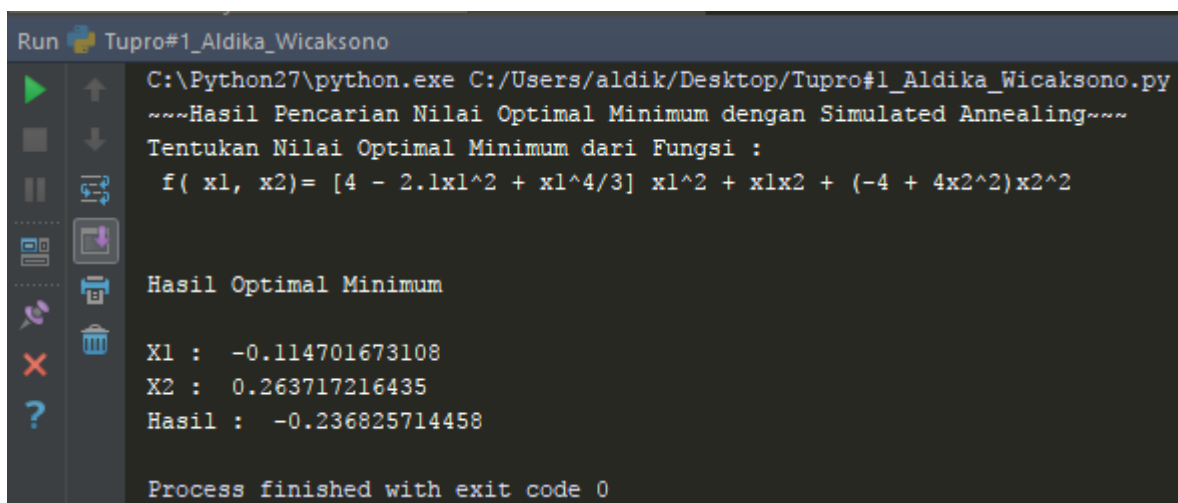
Hasil Optimal Minimum

X1 : -0.114701673108

X2 : 0.263717216435

Hasil : -0.236825714458

Process finished with exit code 0



```
Run Tupro#1_Aldika_Wicaksono
C:\Python27\python.exe C:/Users/aldik/Desktop/Tupro#1_Aldika_Wicaksono.py
~~~Hasil Pencarian Nilai Optimal Minimum dengan Simulated Annealing~~~
Tentukan Nilai Optimal Minimum dari Fungsi :
f( x1, x2)= [4 - 2.1x1^2 + x1^4/3] x1^2 + x1x2 + (-4 + 4x2^2)x2^2

Hasil Optimal Minimum
X1 : -0.114701673108
X2 : 0.263717216435
Hasil : -0.236825714458

Process finished with exit code 0
```