

Laporan Tugas Program 1

Kecerdasan Buatan

A. Analisis

Masalah yang akan diselesaikan dalam Tugas Program 1 Kecerdasan Buatan ini adalah mencari nilai minimum dari sebuah fungsi dua variabel, yaitu :

$$f(x_1, x_2) = -\left(\sin(x_1)\cos(x_2) + \frac{4}{5}\exp\left(1 - \sqrt{x_1^2 + x_2^2}\right)\right)$$

dimana fungsi tersebut memiliki batasan $-10 \leq x_1 \leq 10$ dan $-10 \leq x_2 \leq 10$.

B. Strategi Penyelesaian Masalah

Pada kasus ini, algoritma **Simulated Annealing** menjadi solusi dalam masalah mencari nilai minimum dari fungsi di atas. Algoritma **Simulated Annealing** adalah sebuah algoritma optimisasi dan minimisasi yang bersifat generik. Algoritma ini dapat digunakan untuk mencari nilai minimum global yang optimum dari sebuah fungsi.

Program minimasi ini dibuat dengan menggunakan bahasa pemrograman **Python** dengan library math dan menggunakan Spyder Text Editor.

C. Metode Penyelesaian

1. Pertama, inisiasi fungsi dan parameter yang digunakan, dan tetapkan nilai dari temperature(100), final temperature(0.1), dan alpha(0.9999).

```
def func(x1,x2):  
    return -1 * (math.sin(x1)*math.cos(x2) + (4/5)*math.exp(1 - math.sqrt(x1*x1 + x2*x2)))  
  
def value(koordinat):  
    return func(koordinat[0],koordinat[1])  
  
def prob(n_minimum, c_minimum, temp): # n = new; c = current  
    dt = n_minimum - c_minimum      # dt = delta_e  
    return math.exp(-(dt)/temp)      # temp = temperature  
  
step = 0.9999  
temp = 100  
f_temp = 0.1
```

2. Kedua, mulai dilakukan pengambilan bilangan random untuk x1 dan x2 dalam batasan -10 dan 10. Nilai x1 dan x2 yang didapatkan tadi menjadi current koordinat, dan current minimum didapatkan dari perhitungan fungsi dengan x1 dan x2 pada current koordinat. Hasil dari current minimum dan current koordinat sementara akan menjadi best minimum dan best koordinat karena belum ada pembandingan lain.

```
x1 = uniform(-10,10)  
x2 = uniform(-10,10)  
  
c_koordinat = [x1,x2]  
c_minimum = value(c_koordinat)  
b_minimum = c_minimum  
b_koordinat = c_koordinat
```

3. Lalu, masuk dalam kondisi perulangan ketika nilai temperature masih lebih tinggi dari final temperature maka akan dilakukan pengambilan angka secara random lagi untuk x1 dan x2 dan akan didapatkan nilai new koordinat dan new minimum.

```
while (temp > f_temp):  
    x1 = uniform(-10,10)  
    x2 = uniform(-10,10)  
    n_koordinat = [x1,x2]  
    n_minimum = value(n_koordinat)
```

4. Akan dibandingkan mana nilai yang lebih optimum antara new minimum atau current minimum. Jika nilai new minimum ternyata lebih kecil dari current minimum, maka new minimum tersebut akan menjadi current minimum. Kemudian current minimum tersebut dibandingkan dengan best minimum, jika ternyata nilai current minimum lebih kecil, maka akan menjadi nilai best minimum begitupun current koordinat nya akan menjadi best koordinat.

```
if (n_minimum < c_minimum):  
    c_minimum = n_minimum  
    c_koordinat = n_koordinat  
  
if (c_minimum < b_minimum):  
    b_minimum = c_minimum  
    b_koordinat = c_koordinat
```

5. Namun, jika kondisinya nilai new minimum lebih besar dari nilai current minimum, maka akan dihitung probabilitas apakah nilai tersebut optimum dan akan diambil bilangan random antara 0 dan 1. Jika hasil probabilitas tersebut lebih besar dari bilangan random nya , maka nilai new minimum tersebut optimum dan akan menjadi current minimum begitupun dengan new koordinat akan menjadi current koordinat

```
p = prob(n_minimum, c_minimum, temp)  
if (p > random.uniform(0,1)):  
    c_minimum = n_minimum  
    c_koordinat = n_koordinat
```

6. Kemudian hitung penentu iterasi temperature.

```
temp = temp * step
```

7. Keluarkan hasil akhir yaitu nilai minimum, beserta koordinat nya.

```
print("-----")  
print("HASIL AKHIR")  
print("-----")  
print("Nilai minimum fungsi : ",b_minimum)  
print("Pada koordinat [x1, x2] : ",b_koordinat)
```

D. Screenshot Hasil Akhir

```
-----  
TUGAS PROGRAM 1 KECERDASAN BUATAN  
BRENDA IRENA - IFIK 40 03 - 1301164641  
-----  
-0.6770490001718374 [-7.707980847969433, -2.3235194781080644]  
-0.8799485832467737 [7.399282918333842, 6.079369042769915]  
-0.9634640877899365 [-4.644496791300757, 6.0163058408103325]  
-1.4623813792572715 [1.0126658199327903, -0.4786657144580424]  
-1.5661656478054717 [1.2519174329265752, 0.08541415282888565]  
-1.6744813346235383 [0.530100609628974, -0.28780606530970587]  
-1.9425758515555624 [0.22773244141534654, 0.0624286098031277]  
-1.9617797152165162 [0.07800479411905314, -0.12013562574477987]  
-1.9685112711887474 [0.1472356731472111, 0.09759570415701546]  
-1.9865876978700192 [0.17160982733656915, -0.054902129607221184]  
-2.0540365452924494 [0.10730002324057608, 0.026719225933879542]  
-2.1545125626040855 [0.0005285057899246937, -0.009522605715934773]  
-----  
HASIL AKHIR  
-----  
Nilai minimum fungsi : -2.1545125626040855  
Pada koordinat [x1, x2] : [0.0005285057899246937, -0.009522605715934773]
```