1. Kelompok:

No	NIM	Nama
1	Mohamad Fachrur Ridwan	1103120122
2	Oscar Febri Ramadhan	1103120037
3	Faisal Bima	1103120176

2. Spesifikasi Program

Program yang dibuat bertujuan untuk mencari nilai minimum dari sebuah fungsi yang telah diberikan, berdasarkan interval yang diberikan. Yaitu, sebagai berikut:

Gunakan Algoritma Genetika (AG) untuk menemukan nilai minimum 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 $-3 \leq x_1 \leq 3$ dan $-3 \leq x_2 \leq 3$.

3. Analisis Masalah dan Desain GA

Pada kasus - kasus algoritma genetika pada umumnya, tujuan yang harus dicapai adalah mendapatkan individu terbaik yang diambil berdasarkan nilai fitness tertinggi. Pada kasus ini, tujuan yang harus dicapai adalah kebalikannya, yaitu mendapatkan individu terburuk yang didasarkan pada nilai paling minimum. Namun, kemungkinan untuk mendapatkan individu terburuk sangat kecil karena dalam *roulette wheel,* individu dengan nilai fitness tertinggi mendapatkan kesempatan terbesar untuk diambil menjadi orang tua dan seleksi survivor.
Oleh karena itu, fungsi fitness yang digunakan dibalik untuk membalikan keluarannya agar individu terburuk mendapatkan kesempatan terbesar. Secara spesifik, fungsi fitness diubah menjadi : 1 / [f(x) + 3^5].

Penyelesaian masalah :

Fungsi fitness :

- Fungsi fitness yang digunakan : 1 / [f(x) + 3^5]
- 1/f(x) : digunakan untuk membuat nilai minimum yang memiliki fitness yang tinggi, karena tujuan utama dari program yang dibuat adalah mencari nilai minimum dari sebuah fungsi. Karna dalam seleksi individu menjadi orang tua berdasarkan fitness dari individu tersebut (semakin besar nilai fitness nya, semakin besar pula kemungkinan individu tersebut terpilih menjadi orang tua).
- + 3^5 atau + 243 : digunakan untuk menghindari nilai minus, karena pada saat proses running program, nilai minimum yang didapat beberapa kali merupakan nilai negatif / minus. Nilai 3^5 atau 243 didapat berdasarkan perkiraan dari fungsi yang diberikan. Dilihat dari fungsi yang diberikan pada bagian (... - 2,1*x1^2 + ...) * x1^2, pada bagian ini di fungsi tersebut yang paling memungkinkan menghasilkan nilai minus/negatif. Berdasarkan interval yang diberikan yaitu, sebagai berikut : -3 < x1 < 3, -3 < x2 < 3. Sehingga jika dihitung, menjadi (... - 2,1 * 3^2 + ...) * 3^2, atau dapat dianggap -2,1 * 3^2 * 3^2. Sehingga untuk membuat nilai membuat menjadi tidak negatif / minus, pada fungsi fitness di tambahkan nilai 3^5 atau 243 (nilai ini dinaikan keatas agar tidak menjadi minus, berdasarkan perkiraan awal yang dijelaskan sebelumnya, karna kita tidak tahu berapa nilai minimum dari fungsi tersebut).
- Jika nilai fitness suatu individu negatif, makan individu tersebut tidak memiliki kemungkinan untuk menjadi orang tua (dalam kasus ini, individu terbaik merupakan individu dengan nilai sekecil mungkin / memiliki nilai minimum).

Desain GA :

- Rekombinasi & Mutasi : pada tahap rekombinasi dan mutasi, digunakan probabilitas 1 (pasti akan di rekombinasi dan mutasi, untuk mempercepat proses pencarian, karena jika pada proses rekombinasi dan mutasi terdapat nilai probabilitas yang bukan 1, ada kemungkinan tidak dilakukan proses rekombinasi dan mutasi, atau salah satu saja. Sehingga proses pencarian individu baru terbaik menjadi lebih lama).

4. Cara Kerja Program

Secara umum, alur kerja dari program yang dibangun menerapkan konsep algoritma genetika pada umumnya. Dimana pada tahap pertama akan dibentuk populasi individu secara acak, lalu masing - masing individu akan dievaluasi kecocokannya dengan hasil yang diinginkan dan diambil individu dengan kecocokan tertinggi, selanjutnya dilakukan seleksi orang tua dengan metode *roulette wheel* dan masing - masing pasangan orang tua melakukan reproduksi dengan melakukan proses crossover diselingi dengan mutasi, pada tahap akhir individu baru yang terbentuk

akan dibandingkan dengan individu orang tua untuk dilakukan seleksi survivor yang pada akhirnya akan terpilih individu - individu terbaik berdasarkan pada nilai fitness.

```
def initPopulasi(kl, interval, pops):
    return np.random.uniform(-interval, interval, (pops,kl))

def fitness(kr):
    return 1/( (((4 - (2.1 * (kr[0]*kr[0]) ) + (math.pow(kr[0], 4) / 3) )*(kr[0]*kr[0])) + (kr[0]*kr[1])
        + ((-4 + (4 * (kr[1]*kr[1]) )) *kr[1]*kr[1]))) + 243)
```

Pada tahap ini, dilakukan inisialisasi fungsi untuk men-*generate* populasi awal secara acak dan fungsi fitness sesuai dengan masalah yang diberikan.

```
def selectIndParent(pop):
    total_fitness = 0
    par = 0
    for i in xrange(len(pop)):
        total_fitness += fitness(pop[i])

    dart = random.uniform(0,1)*total_fitness
    temp = 0
    for i in xrange(len(pop)):
        temp += fitness(pop[i])
        if dart < temp:
            par = i
            break
    return par
```

Baris fungsi berikut menggambarkan proses seleksi orang tua dengan evaluasi kecocokan keadaan individu dari hasil yang diinginkan dan diambil individu dengan kecocokan tertinggi.

```
def rekombinasi(pop, indPar):
    child = np.zeros((2,2))
    for i in xrange(len(indPar)):
        parInd = i + 1
        if parInd > (2-1):
            parInd = 0
        for j in range(0,1):
            child[i][j] = pop[int(indPar[i])][j]
        for j in range(1,2):
            child[i][j] = pop[int(indPar[parInd])][j]
    return child

def mutasi(kr, kl, interval):
    mut = random.randint(0,kl-1)
    kr[mut] = (random.uniform(-interval,interval))
    return kr
```

Pada tahap ini, masing - masing individu orang tua yang telah diseleksi akan melakukan reproduksi dengan proses *crossover* dan diselingi dengan mutasi.

```
def fitnessRank(pop):
    return np.array((  np.insert(pop, len(pop[0]), [(fitness(pop[i])) for i in xrange(len(pop))], axis=1)  ))

def selectSurvivor(pop, child):
    sort = fitnessRank(np.append(child,pop,axis=0))[np.argsort(fitnessRank(np.append(child,pop,axis=0))[:,-1])][::-1][:len(pop),:]
    return sort[:,:-1], sort[:,-1]
```

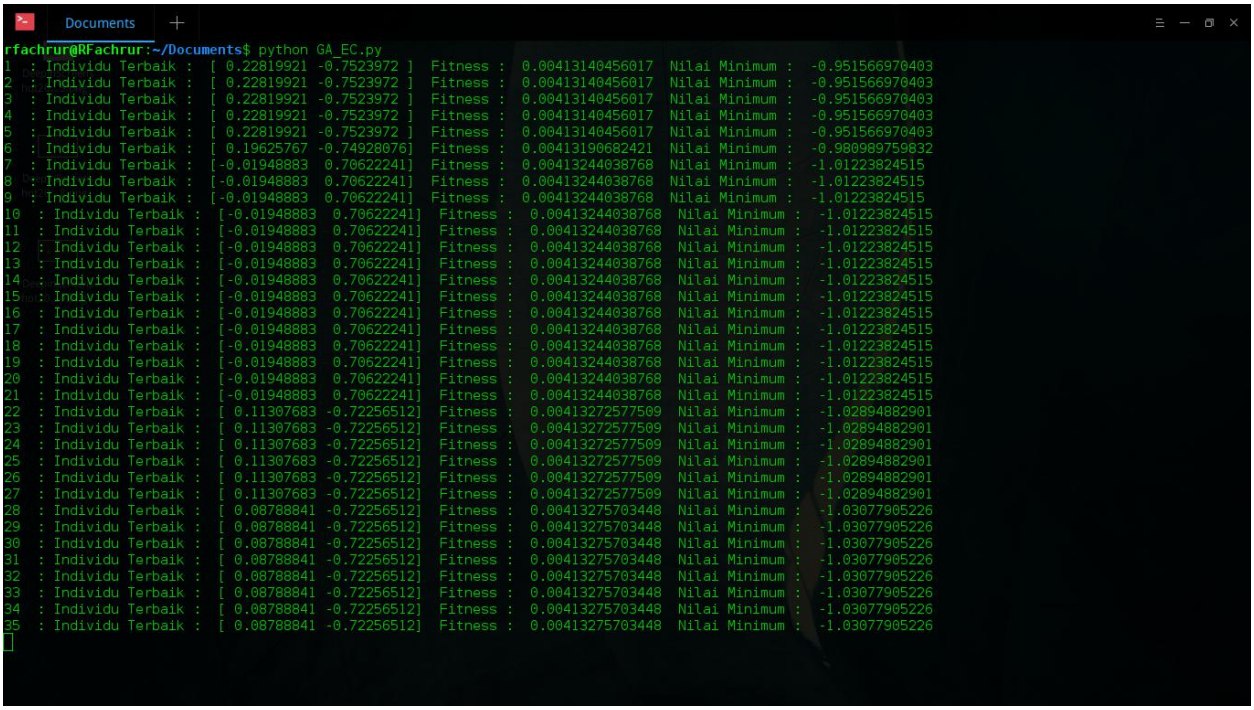
Setelah individu - individu baru terbentuk, tahap selanjutnya adalah seleksi *surivor* dimana individu baru akan dibandingkan nilai fitnessnya dengan individu orang tua. Lalu nilai fitness dari masing - masing individu akan dilakukan pe-*ranking*-an untuk diambil individu - individu terbaik.

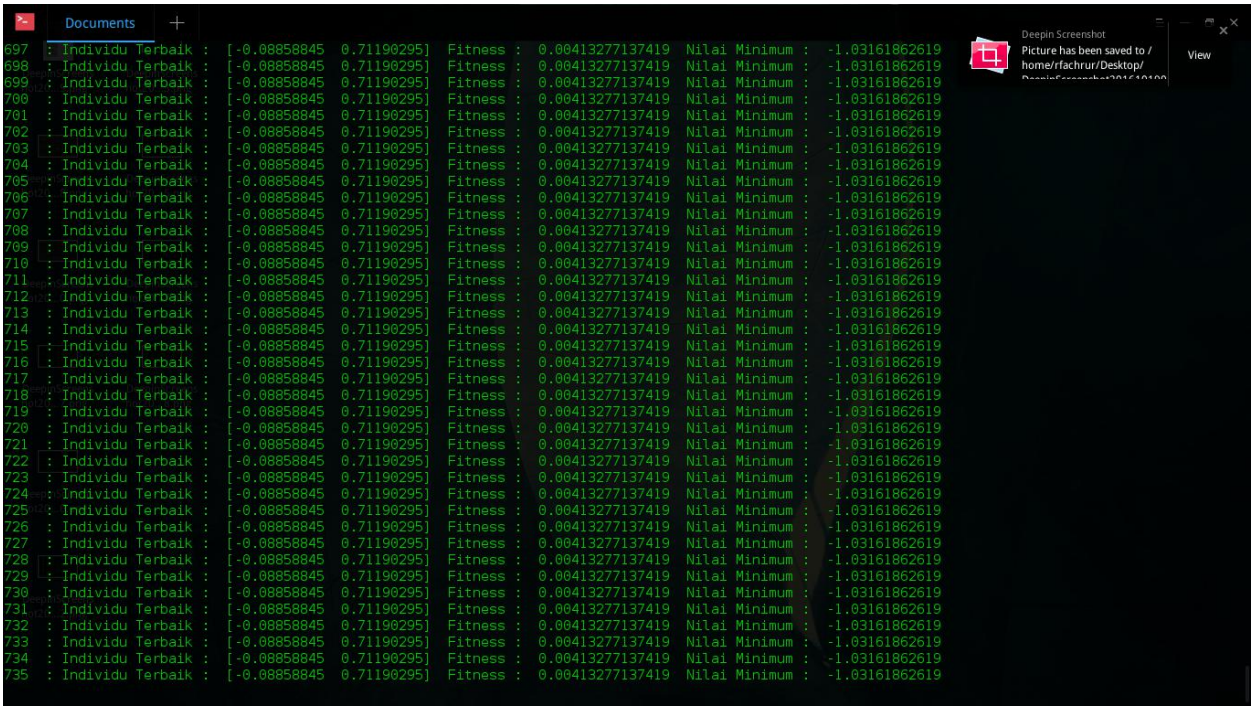
```
def ga(kl, interval, pops, gens):
    pop = initPopulasi(kl, interval, pops)
    gen = 1
    while gen <= gens:
        ch = evolusi(pop, kl, interval)
        pop, sortFitness = selectSurvivor(pop,ch)
        print gen,' : Individu Terbaik : ' ,pop[0], ' Fitness : ' ,sortFitness[0], ' Nilai Minimum : ', ((1/sortFitness[0]) - 243)
        gen += 1


ga(2, 3, 50, 2000)
```

Parameter yang digunakan untuk menyelesaikan kasus ini adalah sebagai berikut : ga(2, 3, 50, 2000); dimana 2 merepresentasikan panjang kromosom, 3 merepresentasikan panjang interval sesuai dari fungsi yang diberikan, 50 merepresentasikan jumlah satu populasi, dan 2000 merepresentasikan jumlah satu generasi

5. Pengujian Program dan Screenshot

[HASIL UJI PROGRAM]






Documents
Deepin Screenshot
Picture has been saved to /
home/rfachrur/Desktop/
View
1962 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1963 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1964 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1965 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1966 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1967 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1968 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1969 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1970 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1971 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1972 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1973 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1974 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1975 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1976 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1977 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1978 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1979 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1980 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1981 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1982 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1983 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1984 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1985 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1986 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1987 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1988 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1989 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1990 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1991 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1992 : Individu Terbaik : [-0.09100241 0.71237149] Fitness : 0.0041327714354 Nilai Minimum : -1.03162220997
1993 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
1994 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
1995 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
1996 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
1997 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
1998 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
1999 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
2000 : Individu Terbaik : [-0.08976653 0.71257309] Fitness : 0.00413277154079 Nilai Minimum : -1.03162838073
rfachrur@RFachrur:~/Documents$