

Tugas Besar 1 IF3070 Dasar Inteligensi Artifisial Pencarian

Solusi Diagonal Magic Cube dengan Local Search



Disusun oleh:

Kelompok 42

Muhammad Rifa Ansyari	/ 18222004
Justin Lawrance	/ 18222006
Axelius Darwin	/ 18222016
Natanael Steven Simangunsong	/ 18222054

Program Studi Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

Daftar Isi

Daftar Isi	2
BAB I	3
Deskripsi Persoalan	3
BAB II	5
Pembahasan	5
2.1 Pemilihan Objective Function	5
2.2 Penjelasan Implementasi Algoritma Local Search	5
2.2.1 Steepest Ascent Hill-climbing	5
2.2.2 Hill-climbing with Sideways Move	7
2.2.3 Random Restart Hill-climbing	8
2.2.4 Stochastic Hill-climbing	10
2.2.5 Simulated Annealing	11
2.2.6 Genetic Algorithm	13
2.3 Hasil Eksperimen	18
2.3.1 Steepest Ascent Hill-climbing	18
2.3.2 Hill-climbing with Sideways Move	23
2.3.3 Random Restart Hill-climbing	27
2.3.4 Stochastic Hill-climbing	32
2.3.5 Simulated Annealing	36
2.3.6 Genetic Algorithm	41
2.4 Hasil Analisis	107
BAB III	110
Kesimpulan dan Saran	110
BAB IV	111
Pembagian Tugas	111
BAB V	112
Referensi	112

BAB I

Deskripsi Persoalan

Diagonal Magic Cube adalah sebuah variasi dari magic cube yang merupakan perpanjangan dari konsep magic square. Dalam persoalan ini, kubus yang digunakan berukuran $5 \times 5 \times 5$, di mana setiap sel dalam kubus tersebut diisi dengan angka-angka dari 1 hingga 125 (n^3) tanpa pengulangan.

Kubus ini memiliki beberapa properti utama yang harus dipenuhi:

1. Magic Number: Ada satu angka yang disebut *magic number* yang merupakan target jumlah yang harus dipenuhi pada tiap dimensi. Angka ini tidak termasuk dalam rentang angka 1 hingga 125 dan tidak merupakan angka yang dimasukkan ke dalam kubus. Pada kubus $5 \times 5 \times 5$ *magic number* bernilai 315, nilai tersebut menggunakan rumus.

$$M = \frac{n(n^3+1)}{2}$$

2. Baris: Jumlah dari angka-angka pada semua baris harus sama dengan magic number.
3. Kolom: Jumlah dari angka-angka pada semua kolom harus sama dengan magic number.
4. Tiang: Jumlah dari angka-angka pada semua kolom harus sama dengan magic number.
5. Diagonal Ruang: Jumlah dari angka-angka pada semua diagonal ruang harus sama dengan magic number.
6. Diagonal Potongan Bidang: Jumlah dari angka-angka pada semua diagonal dari suatu potongan bidang harus sama dengan magic number. Pada sebuah kubus terdapat 9 potongan bidang.

Pendekatan Solusi:

1. Initial State:
 - ◆ Kubus berukuran $5 \times 5 \times 5$ diinisialisasi dengan angka 1 hingga 125 yang diletakan secara acak pada posisi masing-masing sel
2. Proses Pencarian:
 - ◆ Proses pencarian dilakukan menggunakan metode local search yang mencakup:
 - Steepest Ascent Hill-climbing
 - Hill-climbing with Sideways Move
 - Random Restart Hill-climbing
 - Stochastic Hill-climbing
 - Simulated Annealing
 - Genetic Algorithm
3. Tujuan Akhir:

- ◆ Mencapai kondisi di mana hasil penjumlahan setiap angka pada setiap baris, kolom, tiang, diagonal ruang, dan diagonal pada potongan bidang dari kubus tersebut sama dengan magic number.

Penyelesaian Diagonal Magic Cube dapat dicapai dengan menggunakan metode *local search* yang tepat. Dengan memenuhi semua syarat yang diperlukan, seperti jumlah yang sama pada setiap baris, kolom, tiang, dan diagonal, kita dapat menghasilkan kubus yang sesuai dengan target magic number. Berikut ini adalah ilustrasi Diagonal Magic Cube.

67	18	119	106	5
116	17	14	73	95
40	50	81	65	79
56	120	55	49	35
36	110	46	22	101

Gambar 1 Ilustrasi Diagonal Magic Cube

BAB II

Pembahasan

2.1 Pemilihan Objective Function

Fungsi objektif merupakan fungsi yang akan dicari nilai atau solusi optimanya. Fungsi objektif akan mengembalikan nilai dari suatu state berdasarkan batasan yang ada. Untuk membuat *perfect magic cube* $5 \times 5 \times 5$, angka yang berada di pusat kubus harus bernilai 63, oleh karena itu, perlu diberikan pendekatan agar algoritma lebih memilih untuk menempatkan angka 63 di posisi tengah pada kubus. Jika kita mengambil kubus $3 \times 3 \times 3$ yang berada di tengah *perfect magic cube* $5 \times 5 \times 5$, maka seluruh potongan yang melalui pusat kubus (yaitu 63) jumlahnya akan bernilai 189, sehingga akan diberikan *state value* tambahan agar algoritma lebih memilih kondisi ini.

Berdasarkan informasi tersebut berikut batasan yang kami tentukan untuk menentukan objective function

- Setiap baris, kolom, atau tiang yang jumlahnya belum memenuhi magic number mendapatkan nilai 1 (total maksimal = $25 \times 3 = 75$)
- Sepasang diagonal bidang pada suatu bidang tertentu yang belum memenuhi magic number mendapatkan nilai 5 (total maksimal = $15 \times 5 = 75$)
- Setiap diagonal ruang yang belum memenuhi magic number mendapatkan nilai 18 (total maksimal = $4 \times 18 = 72$)
- Kubus yang pusatnya belum bernilai 63 mendapatkan nilai 1000
- Pada kubus $3 \times 3 \times 3$ yang berada di tengah magic cube, setiap potongan yang melalui pusat kubus yang jumlahnya tidak bernilai 189, mendapat nilai 100 (total maksimal = $13 \times 100 = 1300$)

Dari batasan tersebut, heuristik yang digunakan untuk permasalahan magic cube adalah sebagai berikut,

$h = -$ (total nilai yang didapatkan suatu state magic cube berdasarkan batasan yang diberikan)

Nilai minimum $h = - (75 + 75 + 72 + 1300 + 1000) = (- 2522)$

Nilai maksimum $h = 0$, yaitu ketika semua syarat magic cube sudah terpenuhi

2.2 Penjelasan Implementasi Algoritma Local Search

2.2.1 Steepest Ascent Hill-climbing

Algoritma *Steepest Ascent Hill-Climbing* melakukan iterasi terus menerus mencari *neighbor* dengan nilai fungsi objektif yang paling tinggi. Pada setiap iterasi, algoritma akan mengecek nilai fungsi objektif dari setiap *successor*, lalu memilih *successor* dengan nilai tertinggi sebagai *neighbor*. *Neighbor* kemudian dibandingkan nilai fungsi objektif pada *state* saat ini. Jika

nilai dari *neighbor* lebih besar dari *current value* maka *current* akan digantikan dengan *neighbor*. Jika nilai dari *neighbor* lebih kecil atau sama dengan *current value*, maka akan mengembalikan *state current*. Iterasi akan terus dilakukan hingga *neighbor value* tidak ada yang lebih dari *current value*.

Algoritma ini adalah versi dari hill-climbing di mana semua solusi yang mungkin digenerasi terlebih dahulu, lalu dipilih yang memiliki state value paling tinggi

- **Langkah-langkah:**

1. State awal random magic cube.
2. Generate semua neighbor yang mungkin. Jumlah neighbor pada setiap generasi adalah 7750 dihitung menggunakan rumus kombinasi,

$${}_nC_r = \frac{n!}{r!(n-r)!}$$

n = 125 (jumlah sel), r = 2 (jumlah sel yang ditukar)

3. Gunakan objective function untuk menghitung state value dari setiap solusi.
4. Pilih neighbor yang memiliki state value paling tinggi.
5. Ulangi langkah 2 sampai 4 hingga tidak ada lagi perbaikan yang bisa dilakukan.

Steepest Ascent Hill-climbing merupakan algoritma search yang kurang efektif karena Magic Cube mempunyai jumlah neighbor yang banyak sehingga perlu di generate semua terlebih dahulu sehingga memakan waktu yang lama. Serta algoritma ini mudah terjebak dalam local optima.

```
def steepest(arr) :  
    current = [i for i in arr]  
    print("state awal:" ,end="")  
    print(o.objective(current))  
    j=1  
    while True:  
        neighbor = c.neighbor(current)  
        neighborObjective = o.objective(neighbor)  
        print("iterasi " + str(j) + ": ",end="")  
        print(neighborObjective)  
        j+=1
```

```

if neighborObjective<=o.objective(current) :
    print("state akhir:" ,end="")
    print(o.objective(current))
    return current
current = neighbor

```

2.2.2 Hill-climbing with Sideways Move

Algoritma *Hill-climbing with Sideways Move* melakukan iterasi terus menerus mencari *neighbor* dengan nilai fungsi objektif yang paling tinggi. Pada setiap iterasi, algoritma akan mengecek nilai fungsi objektif dari setiap *successor*, lalu memilih *successor* dengan nilai tertinggi sebagai *neighbor*. *Neighbor* kemudian dibandingkan nilai fungsi objektif pada *state* saat ini. Jika nilai dari *neighbor* lebih besar atau sama dengan *current value* maka *current* akan digantikan dengan *neighbor*. Jika nilai dari *neighbor* lebih kecil dari *current value*, maka akan mengembalikan *state current*.

- **Langkah-langkah:**

1. State awal random magic cube.
2. Gunakan objective function untuk menghitung state value dari setiap solusi.
3. Memilih neighbor dengan state value yang lebih baik, lalu jika tidak ada, memilih state value yang sama.
4. Tetap menjalankan algoritma sampai puncak atau sampai terjebak.

```

def sideways(arr) :
    current = [i for i in arr]
    print("state awal:" ,end="")
    print(o.objective(current))
    j=1
    countsidestep = 0
    while True:
        neighbor = c.sidewaysneighbor(current)
        neighborObjective = o.objective(neighbor)
        print("iterasi " + str(j) + ": ",end="")
        print(neighborObjective)

```

```

        j+=1
        if neighborObjective<o.objective(current) or
countsidestep==50 :

            print("state akhir:" ,end="")
            print(o.objective(current))
            return current

        if neighborObjective == o.objective(current) :
            countsidestep+=1
        else :
            countsidestep = 0

        current = neighbor

```

2.2.3 Random Restart Hill-climbing

Algoritma *Random Restart Hill-Climbing* merupakan sekuensial dari algoritma *hill-climbing*, dimana jika iterasi berakhir dengan nilai maksimum lokal, maka algoritma akan diulang dengan *initial state* acak yang berbeda hingga menemukan nilai maksimum lokal. Pengulangan (*restart*) mungkin dilakukan lebih dari satu kali.

- **Langkah-langkah:**

1. Lakukan Steepest Ascent hill-climbing biasa.
2. Jika algoritma terjebak dalam local optima, restart dengan state awal magic cube yang random.
3. Ulangi hingga ditemukan solusi yang optima atau mencapai batas waktu.

Random Restart Hill-climbing merupakan algoritma yang serupa dengan Steepest Ascent dengan perbedaan, saat masih belum mencapai global optima, state magic cube akan diulang dari state random magic cube lalu kembali melakukan searching. Algoritma ini lebih baik daripada Steepest Ascent karena memiliki kemungkinan yang lebih tinggi untuk mendapatkan local optima atau mencapai global optima. Namun, algoritma ini membutuhkan waktu yang lebih lama karena jumlah neighbor yang banyak sehingga kurang efektif.


```

def steepest(arr) :
    current = [i for i in arr]
    print("state awal:" ,end="")
    print(o.objective(current))
    j=1
    while True:
        neighbor = c.neighbor(current)
        neighborObjective = o.objective(neighbor)
        print("iterasi " + str(j) + ": ",end="")
        print(neighborObjective)
        j+=1
        if neighborObjective<=o.objective(current) :
            print("state akhir:" ,end="")
            print(o.objective(current))
            return current
        current = neighbor

def main() :
    arr = c.getRandomCube()
    numRestart = 0
    maxRestart = 5
    start=t.time()
    maxarr = arr

    while (o.objective(arr) < 0 and numRestart <= maxRestart):
        print("Restart ke-" + str(numRestart+1))

        arr = c.getRandomCube()
        c.printArray(arr)
        arr = steepest(arr)
        c.printArray(arr)
        o.diagnostic(arr)
        if (o.objective(arr) >= o.objective(maxarr)): maxarr = arr
        numRestart += 1

```

```

end=t.time()
elapsed = end - start
print("duration: " + str(elapsed))
print("highest value:\n")
c.printArray(maxarr)
print("value: " + str(o.objective(maxarr)))
o.diagnostic(maxarr)
return 0

```

2.2.4 Stochastic Hill-climbing

Algoritma *Stochastic Hill-Climbing* akan melakukan iterasi menentukan suatu *random successor* dari *state* terkini dan kemudian membandingkan *value*-nya dengan *current value*. Pada setiap iterasi, akan dipilih sebuah *random successor* yang kemudian menjadi nilai *neighbor*. Jika *neighbor value* lebih besar dari *current value*, maka *current value* akan diperbaharui dengan *neighbor value*. Sebaliknya, jika *neighbor value* lebih kecil dari *current value*, akan dipilih kembali satu *random successor* yang akan menjadi nilai *neighbor* terkini.

- **Langkah-langkah:**

1. State awal random magic cube.
2. Secara acak memilih neighbor, jika state value lebih baik maka akan berpindah, jika tidak mendapatkan state value yang lebih baik maka akan memilih neighbor lagi secara acak dan mengevaluasinya kembali.
3. Lanjutkan sampai tidak ada tetangga yang lebih baik atau mencapai global optima atau mencapai iterasi yang sudah ditentukan.

Stochastic Hill-climbing dapat memberikan state value yang baik dalam waktu singkat karena tidak perlu mengevaluasi semua neighbor. Algoritma ini lebih efektif dalam menghindari terjebaknya di local optima dan lebih cepat dibandingkan Steepest, Random Restart, dan Sideways Move.

```

def stochastic(arr)->int :
    current = [i for i in arr]

```

```

neighbor = [i for i in arr]
print("state awal:" ,end="")
print(o.objective(current))

for i in range(78000) :
    neighbor = c.randomSuccessor(neighbor)

    if o.objective(neighbor) > o.objective(current):
        current = [i for i in neighbor]

    neighbor= [i for i in current]

    print("iterasi " + str(i+1) + ": ",end="")
    print(o.objective(current))

return current

```

2.2.5 Simulated Annealing

Algoritma *Simulated Annealing* merupakan salah satu versi dari *Stochastic Hill-Climbing*, namun memiliki sedikit perbedaan. Algoritma ini memperbolehkan *current state* berganti menjadi *neighbor* yang nilai objektifnya lebih kecil. Algoritma ini juga mempertimbangkan nilai T, yaitu parameter yang merupakan sebuah konstanta/nilai yang akan terus berkurang seiring dengan banyaknya iterasi. Jika nilai T sama dengan 0, algoritma melakukan terminasi. Pada setiap iterasi, akan dipilih sebuah *random successor* yang kemudian menjadi nilai *neighbor*. Lalu, akan dihitung nilai ΔE , dengan persamaan $\Delta E = \text{neighbor value} - \text{current value}$. Jika ΔE lebih besar dari 0, *current value* diperbaharui dengan *neighbor value*. Jika ΔE lebih kecil atau sama dengan 0, *current value* diperbaharui dengan *neighbor value* hanya jika $e^{\Delta E/T}$ lebih besar dari suatu nilai dalam *range* 0-1 yang kita tentukan di awal (bisa bersifat statis maupun *random*).

- **Langkah-langkah:**

1. Mulai dari state awal random magic cube.
2. Secara bertahap kurangi suhu, suhu ditentukan memakai suatu fungsi waktu yang semakin mengecil hingga suhu 0.

3. Pilih neighbor secara acak. Jika neighbor lebih baik, mengambil state value baru. Jika lebih buruk, mengambil dengan probabilitas tertentu yang bergantung pada suhu.
4. Lanjutkan hingga suhu nol dan tidak ada state value yang lebih baik.

Magic Cube merupakan persoalan yang besar dan kompleks dengan banyak local optima. Tetapi local optima ini bisa dihindari dengan Simulated Annealing dengan adanya probabilitas berpindah ke state value yang lebih buruk. Hal lain yang membuat algoritma Simulated Annealing lebih baik adalah fungsi waktu dapat dioptimasi untuk menghindari local optima. Walaupun Simulated Annealing sedikit lebih lambat dibanding Stochastic, algoritma ini masih lebih baik dalam mencari hasil dibanding Stochastic.

```
def simulatedAnnealing(arr) :  
  
    current = [i for i in arr]  
  
    next = [i for i in arr]  
    print("state awal:" ,end="")  
    print(o.objective(current))  
  
    j=0  
    stuck = 0  
    while True:  
        j+=1  
        t = schedule(j)  
        if t <= 0.000000000000000000001 or o.objective(current)==0:  
return current,stuck  
        next = c.randomSuccessor(next)  
        deltaE = o.objective(next) - o.objective(current)  
  
        a = r.random()  
  
        if deltaE > 0 :  
            current = [i for i in next]
```

```

        stuck=0
    elif deltaE <= 0 and a <= m.exp(deltaE/t) :
        current = [i for i in next]
        stuck=0

    stuck+=1

    next = [i for i in current]

    print("iterasi " + str(j) + ": ",end="")
    print(o.objective(current))

def schedule(a) :
    t=100000.0
    coolingRate = 0.9995
    T = t * m.pow(coolingRate,a)
    return T

```

2.2.6 Genetic Algorithm

Genetic Algorithm mencari solusi dengan memanfaatkan persilangan antara dua *state* yang disebut sebagai *parent*. Berbeda dengan algoritma *hill climbing*, *state value* dari algoritma ini menggunakan *fitness function* (F), yang pada kasus *diagonal magic cube* ini, merupakan jumlah dari baris, kolom, tiang, diagonal bidang, diagonal ruang yang sudah memenuhi *magic number* (kebalikan dari fungsi h pada *hill climbing*).

- **Langkah-langkah:**

1. Mulai dengan suatu populasi magic cube secara random.
2. Invert heuristic magic cube dari negatif (-2522 hingga 0) menjadi (0 hingga 2522) dengan cara $2522 + (\text{current heuristic})$. Hal ini supaya heuristic lebih besar memiliki persentase fitness lebih besar.
3. Evaluasi nilai fitness pada setiap solusi menggunakan fitness function
$$\text{fitness value} = \frac{\text{state value}}{\text{sum of population state value}} \times 100\%$$
4. Membuat suatu populasi baru (kosong).

5. Memilih dua parent secara random (semakin besar fitness value semakin mudah terpilih).
6. Lakukan crossover. Crossover mengambil dua parent, indeks [0-62] dari satu parent dan indeks [63-124] dari parent lainnya, parent dengan heuristic lebih besar mengisi indeks [0-62]. Lebih lengkapnya sebagai berikut:
 - Parent 1 mengisi indeks[0-62]
 - Parent 2 mengisi indeks[63-124]. Jika angka sudah ada akan diganti dengan -1
 - Iterasi child cube. Jika ditemukan -1 iterasi parent 2, jika tidak duplikat masukan nilai parent 2 ke child.
7. Crossover menghasilkan child. Child memiliki probabilitas kecil untuk mengalami mutasi. Mutasi dilakukan dengan menukar 2 angka pada 2 indeks random
8. Tahap 5 hingga 7 diulang hingga individu pada populasi baru berjumlah sama dengan populasi awal.
9. Populasi awal dan populasi akhir dibandingkan nilai max heuristicnya. Populasi dengan nilai heuristic lebih besar akan dipilih untuk melanjutkan iterasi. Hal ini dilakukan agar terjadi konvergensi pada GA sambil memberikan ruang untuk eksplorasi.
10. Lanjut iterasi berikutnya (mengulang langkah 2 - 9)

Genetic Algorithm membutuhkan waktu yang cukup lama karena melakukan lebih banyak komputasi dibandingkan Stochastic dan Simulated Annealing. Namun Genetic Algorithm lebih baik dalam melakukan eksplorasi, menghindari terjebak pada local optima, selain itu metode yang digunakan untuk melakukan crossover dapat dioptimasi, seperti memastikan child memiliki nilai 63 pada pusat kubus. Bila computing power dan waktu bukan merupakan constraint, Genetic Algorithm dinilai dapat menemukan global optima dengan lebih baik.

```
def isDuplicate(arr, num: int) -> bool:
    dupe = False
    for i in range(0, len(arr)):
        if arr[i] == num :
            dupe = True
            break
```

```

        return dupe

def visualizePopulation(population):
    for i in range(len(population)):
        print("cube: \n")
        c.printArray(population[i])
        print()
        print (f"objective function:
{o.objective(population[i])}\n\n")

def generatePopulation(population: int):
    populationInit = [0]*population

    for i in range (0, population):
        populationInit[i] = c.getRandomCube()

    return populationInit

def genetic(population, arr, iterationCount ):
    iterationCount[:] = [iterationCount[0] + 1]

    populationInit = arr
    stateVal = [0]*population
    fitnessVal = [0]*population

    for i in range (0, population):
        stateVal[i] = o.objective(populationInit[i])

    for i in range (0, population):
        stateVal[i] = 2522 + stateVal[i]

    print("state val: ")
    print(stateVal)

```

```

for i in range (0, population):
    fitnessVal[i] = (stateVal[i]/sum(stateVal))*100

populationNew = [0]*population
for l in range (0, population):
    parrIdx = [0]*2
    for i in range (0, population):
        for j in range (0,2):
            random_number = random.randint(1, 100)
            temp = 0
            for k in range (0,population):
                temp += fitnessVal[k]
                if random_number <= temp:
                    parrIdx[j] = k
                    break

    if stateVal[parrIdx[0]] >= stateVal [parrIdx[1]]:
        temp = populationInit[parrIdx[0]][0:63] + [-1]*(125-63)
        for i in range (63, 125):
            if not isDuplicate(temp,
                populationInit[parrIdx[1]][i]):
                temp[i] = populationInit[parrIdx[1]][i]

        for i in range (63, 125):
            if temp[i] == -1:
                for j in range (0, 63):
                    if not isDuplicate(temp,
                        populationInit[parrIdx[1]][j]):
                        temp[i] = populationInit[parrIdx[1]][j]
                        break
    else:
        temp = populationInit[parrIdx[1]][0:63] + [-1]*(125-63)
        for i in range (63, 125):

```



```

        if not isDuplicate(temp,
        populationInit[parrIdx[0]][i]):
            temp[i] = populationInit[parrIdx[0]][i]

    for i in range (63, 125):
        if temp[i] == -1:
            for j in range (0, 63):
                if not isDuplicate(temp,
                populationInit[parrIdx[0]][j]):
                    temp[i] = populationInit[parrIdx[0]][j]
                    break

    chance = 0.3
    if random.random() < chance:
        idx1 = random.randint(0,124)
        idx2 = random.randint(0,124)

        tempVal = temp[idx1]
        temp[idx1] = temp[idx2]
        temp[idx2] = tempVal

    populationNew[1] = temp

    print(f"iterasi ke-{iterationCount[0]}")
    print("Objective Function: ")
    print(stateVal)
    print(f"Max: {max(stateVal)}")
    print(f"Avg: {sum(stateVal)/len(stateVal)}\n")

    newStateVal = [0]*population
    for i in range (0, population):
        newStateVal[i] = o.objective(populationNew[i])
    for i in range (0, population):
        newStateVal[i] = 2522 + newStateVal[i]

```

```

        if (max(stateVal)) > (max(newStateVal)):
            return populationInit
        else:
            return populationNew

def main():
    population: int = 10
    iteration: int = 300000
    itercount = [0]
    start_time = t.time()

    print("state awal: ")
    result = generatePopulation(population)
    visualizePopulation(result)

    result = genetic(population, result, itercount)

    for i in range(0, iteration-1):
        result = genetic(population, result, itercount)

    end_time = t.time()
    print("\n\nstate akhir: ")
    visualizePopulation(result)
    print(f"Duration: {end_time - start_time}")

    return

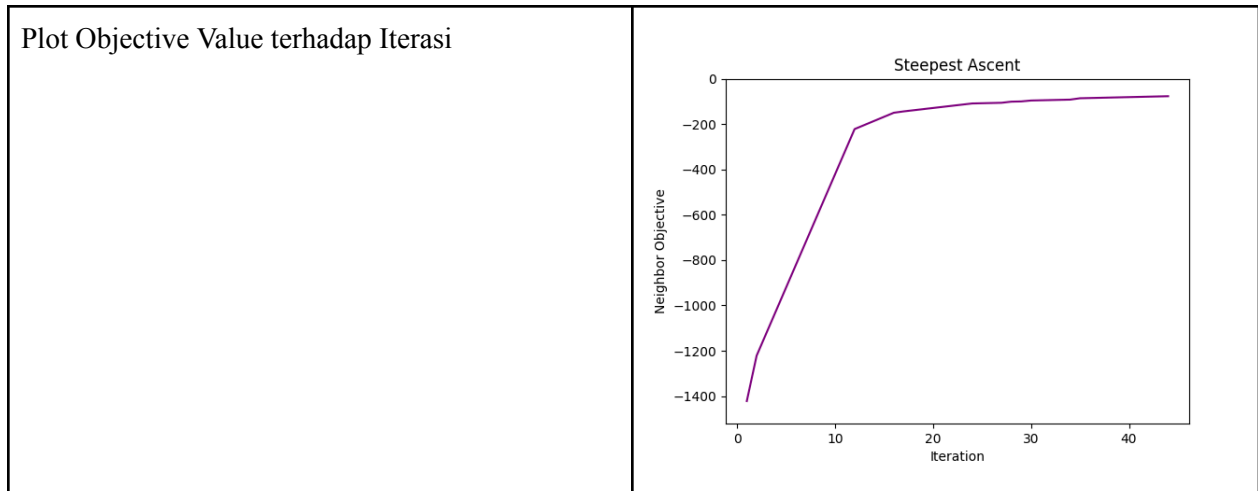
```

2.3 Hasil Eksperimen

2.3.1 Steepest Ascent Hill-climbing

Percobaan 1

Objective Value & Durasi	<div>state awal:-2422</div> <div>state akhir(iterasi 46): -77</div> <div>duration: 6.721860647201538</div>
State Awal	State Akhir
<div>107 26 42 34 100 120 64 48 103 88 92 93 87 78 24 99 30 41 89 9 28 75 46 97 29 76 94 82 101 72 12 73 11 119 16 122 53 6 33 23 62 31 15 60 102 3 105 37 81 65 117 14 69 123 121 114 115 96 45 7 47 77 35 80 25 74 67 13 20 50 85 55 98 52 57 49 43 61 2 112 5 84 111 39 95 22 71 63 56 8 40 90 58 70 10 38 68 106 51 83 109 1 118 59 125 110 18 66 21 124 91 36 104 113 116 17 27 4 79 44 54 32 19 108 86</div>	<div>107 26 42 34 72 120 38 48 103 88 92 93 87 104 24 99 30 41 61 4 28 122 6 97 17 76 108 82 101 21 12 73 11 119 114 75 70 91 55 23 62 31 15 60 102 3 105 44 81 65 117 14 110 123 121 10 106 113 59 116 47 77 63 49 25 74 67 13 20 52 9 36 125 50 5 80 58 89 39 112 124 66 111 95 2 94 71 35 56 32 40 7 115 53 16 64 68 37 8 83 109 100 118 45 98 69 18 84 1 57 46 33 78 22 90 27 96 85 79 43 54 51 86 29 19</div>



Percobaan 2

Objective Value & Durasi	<pre>70 75 82 93 state awal:-2522 iterasi 1: -173 state akhir(iterasi 45): -75 duration: 7.565250396728516</pre>
State Awal	State Akhir

```

83 116 84 106 111
10 49 112 101 65
26 85 34 42 117
107 105 91 71 6
93 20 36 40 25

29 30 125 14 37
92 54 18 33 81
41 115 109 28 64
120 19 44 48 24
88 66 23 102 97

118 74 11 9 46
63 61 108 123 12
51 95 4 68 50
69 58 114 119 2
89 43 57 70 87

45 82 35 16 21
78 31 17 55 86
53 104 60 22 94
67 90 39 27 110
56 121 124 75 122

96 3 52 79 103
5 1 7 62 100
38 15 72 113 98
59 80 77 8 13
76 73 32 99 47

```

```

83 38 84 106 111
10 49 112 101 65
26 85 34 42 117
89 76 91 19 6
93 20 103 88 30

29 25 125 120 37
92 54 18 36 81
105 115 109 28 74
14 71 44 48 24
97 121 1 102 40

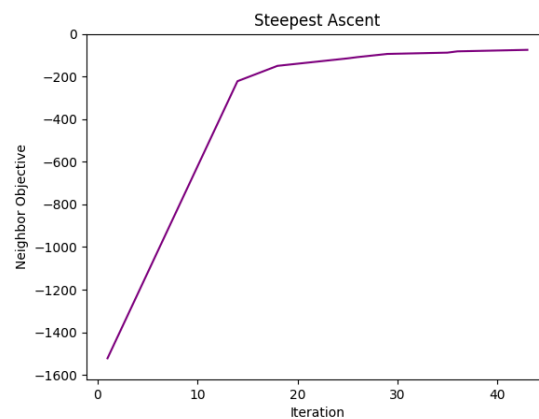
118 79 22 9 12
4 7 39 123 75
51 95 63 31 56
69 3 87 119 23
8 47 57 70 114

45 60 2 58 21
68 78 82 55 86
53 98 17 11 59
67 90 108 72 110
113 66 124 46 122

96 16 52 64 33
116 77 100 62 61
5 41 27 80 104
94 50 35 107 32
15 73 13 99 43

```

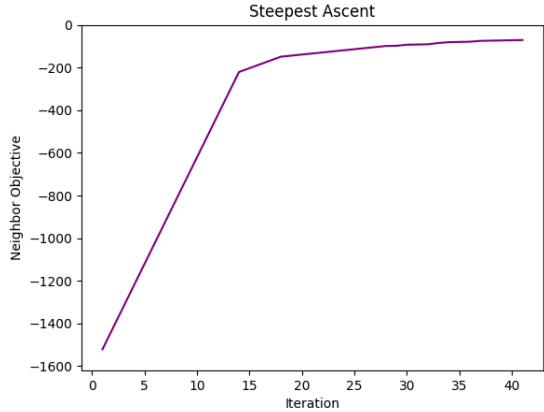
Plot Objective Value terhadap Iterasi



Percobaan 3

Objective Value & Durasi

state awal:-2521

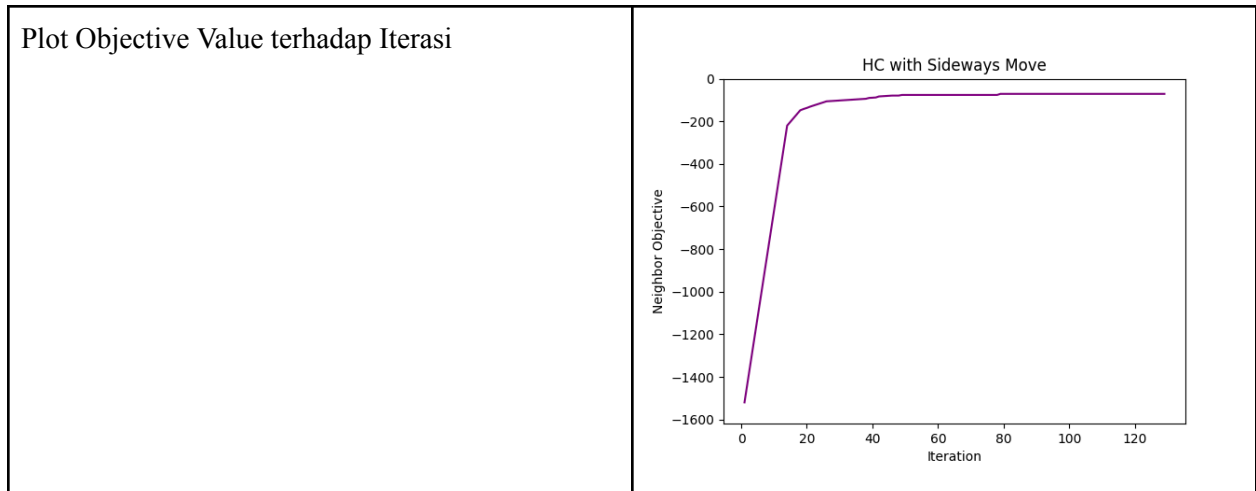
	<div>state akhir(iterasi 43): -71</div> <div>duration: 6.763972282409668</div>																				
State Awal	State Akhir																				
<div><div>34 99 9 14 36</div><div>62 66 31 63 116</div><div>114 125 84 103 89</div><div>76 5 50 29 87</div><div>67 92 55 35 23</div><div>109 25 58 51 38</div><div>86 24 79 17 73</div><div>80 47 61 102 71</div><div>90 88 20 13 44</div><div>40 27 97 41 100</div><div>39 12 120 119 85</div><div>124 16 75 22 49</div><div>93 43 96 53 11</div><div>19 46 33 74 18</div><div>30 95 113 59 37</div><div>108 45 68 70 83</div><div>52 81 115 1 112</div><div>32 69 111 122 77</div><div>121 15 54 4 21</div><div>91 3 106 118 105</div><div>2 65 8 101 104</div><div>28 107 57 123 56</div><div>26 7 72 60 6</div><div>78 94 110 82 48</div><div>98 42 10 117 64</div></div>	<div><div>34 99 9 14 36</div><div>62 66 31 96 12</div><div>114 125 84 103 89</div><div>76 10 55 29 100</div><div>67 116 117 35 124</div><div>109 25 58 119 1</div><div>86 24 79 111 73</div><div>80 4 61 57 46</div><div>98 88 20 13 44</div><div>75 27 97 60 87</div><div>39 64 120 51 85</div><div>6 16 93 22 49</div><div>40 43 63 83 11</div><div>19 104 33 110 107</div><div>91 54 81 74 37</div><div>108 45 115 70 53</div><div>52 113 106 38 112</div><div>32 69 65 122 94</div><div>121 15 47 102 68</div><div>30 3 105 5 41</div><div>2 17 8 101 59</div><div>28 21 95 123 56</div><div>26 7 42 18 23</div><div>78 77 71 82 48</div><div>90 72 118 50 92</div></div>																				
Plot Objective Value terhadap Iterasi	<div>Steepest Ascent</div>  <table border="1"><caption>Data points estimated from the Steepest Ascent plot</caption><thead><tr><th>Iteration</th><th>Neighbor Objective</th></tr></thead><tbody><tr><td>1</td><td>-1500</td></tr><tr><td>5</td><td>-1100</td></tr><tr><td>10</td><td>-700</td></tr><tr><td>15</td><td>-200</td></tr><tr><td>20</td><td>-100</td></tr><tr><td>25</td><td>-50</td></tr><tr><td>30</td><td>-30</td></tr><tr><td>35</td><td>-20</td></tr><tr><td>40</td><td>-50</td></tr></tbody></table>	Iteration	Neighbor Objective	1	-1500	5	-1100	10	-700	15	-200	20	-100	25	-50	30	-30	35	-20	40	-50
Iteration	Neighbor Objective																				
1	-1500																				
5	-1100																				
10	-700																				
15	-200																				
20	-100																				
25	-50																				
30	-30																				
35	-20																				
40	-50																				

2.3.2 Hill-climbing with Sideways Move

Keterangan : Parameter jumlah maximum sideways move adalah 50

Percobaan 1

Objective Value & Durasi	<div>state awal:-2520</div> <div>state akhir:-71</div> <div>duration: 37.021841287612915</div>
State Awal	State Akhir
<div> 1 19 92 9 109 11 100 72 23 88 80 64 103 18 62 67 55 77 117 110 93 105 34 30 98 116 57 26 32 95 33 74 107 84 124 90 48 73 121 56 106 3 58 39 54 51 24 101 2 20 44 13 78 5 45 49 102 43 75 108 53 66 85 120 111 118 8 36 91 71 46 61 16 83 60 99 27 22 12 47 6 52 31 96 125 21 17 114 112 29 63 35 10 119 65 94 104 86 37 79 25 42 59 70 69 76 87 14 123 89 28 81 40 97 113 15 115 41 38 7 82 50 68 4 122 </div>	<div> 4 89 92 1 44 2 23 78 100 76 62 71 103 29 28 39 55 64 40 68 57 77 34 125 101 41 32 46 115 50 99 7 107 91 27 117 14 114 109 9 106 30 95 74 20 15 84 80 11 94 26 65 72 116 121 18 24 90 75 108 53 6 63 120 111 118 51 36 102 8 45 61 16 54 60 93 73 43 124 33 85 52 31 96 3 81 17 12 112 42 66 35 19 119 22 79 37 98 104 47 25 105 59 70 69 88 87 48 10 38 49 5 13 97 113 83 123 58 86 110 82 21 67 56 122 </div>



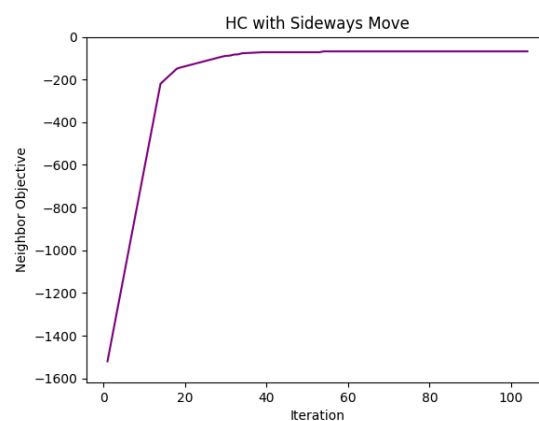
Percobaan 2

Objective Value & Durasi	<div>state awal:-2520</div> <div>state akhir (iterasi 106): -68</div> <div>duration: 29.19540047645569</div>
State Awal	State Akhir

47	11	1	99	97
59	57	56	112	75
42	123	115	108	91
93	109	23	90	8
35	92	62	77	76
107	82	14	41	78
37	113	124	121	74
114	125	31	25	118
65	24	33	27	19
21	18	85	66	17
120	51	46	94	106
10	3	44	89	53
119	12	87	26	84
28	111	4	117	5
7	100	110	36	54
34	20	58	30	38
81	2	32	16	104
98	50	69	61	60
9	80	64	55	45
101	71	49	68	96
73	70	95	72	29
52	86	39	13	116
105	79	102	122	40
63	43	103	67	6
88	15	22	83	48

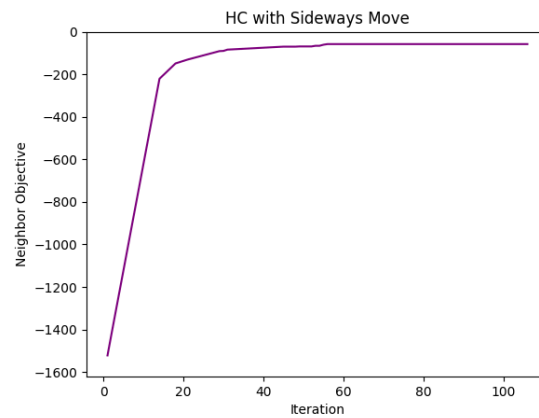
78	56	95	83	38
67	90	87	109	75
40	34	11	25	65
41	31	8	112	32
97	28	74	77	53
17	4	47	20	117
79	71	124	46	104
64	125	57	50	62
23	110	93	27	54
21	66	58	68	107
7	122	121	14	120
119	3	44	37	100
51	12	63	114	84
15	89	82	123	111
106	98	24	94	19
10	103	108	43	35
9	99	33	16	91
36	76	69	1	60
81	80	2	55	45
101	113	18	49	96
73	70	22	72	29
52	13	39	86	116
105	85	102	5	42
115	30	118	59	6
88	92	61	26	48

Plot Objective Value terhadap Iterasi



Objective Value & Durasi	<div>state awal:-2522</div> <div>state akhir (iterasi 108): -58</div> <div>duration: 30.151792287826538</div>
State Awal	State Akhir
<div>44 99 13 31 119 32 120 7 71 38 79 124 54 48 39 45 110 93 104 101 84 1 9 103 8 63 12 21 4 64 94 56 78 60 81 118 43 82 125 10 27 26 29 53 61 22 62 112 97 76 24 117 107 65 40 90 50 92 98 109 113 33 35 80 14 47 85 59 2 30 34 114 15 23 83 106 17 46 20 18 123 55 28 57 121 66 91 52 86 74 96 75 122 68 58 51 19 108 6 11 102 77 36 37 25 100 89 67 70 5 73 72 41 105 42 16 88 69 49 3 115 95 87 111 116</div>	<div>10 99 107 31 11 38 104 34 42 32 87 122 54 20 39 3 110 27 120 117 101 36 19 18 24 76 88 21 41 35 94 58 78 51 56 74 40 82 1 52 65 69 98 71 61 22 62 23 85 84 8 114 90 108 118 17 124 67 29 109 9 33 63 93 121 30 97 59 2 96 43 12 15 113 83 5 50 80 123 119 112 55 28 57 14 66 125 44 86 70 13 75 48 68 81 60 91 72 6 103 102 95 47 111 25 105 89 92 26 100 73 4 46 106 53 45 16 7 49 64 115 77 79 37 116</div>

Plot Objective Value terhadap Iterasi



2.3.3 Random Restart Hill-climbing

Percobaan 1 : Jumlah restart 5

Objective Value & Durasi

state awal:-2522

best objective value: -70

duration: 28.926430702209473

State Awal

State Akhir Terbaik

25	112	15	32	8
31	28	109	47	24
86	124	6	44	85
94	125	26	43	72
68	123	57	21	63

90	38	29	71	104
19	5	2	13	111
69	67	103	96	116
12	114	22	92	73
84	39	117	95	52

107	110	70	65	88
14	119	18	45	102
75	101	35	106	120
115	4	41	121	56
3	7	36	98	77

51	93	83	48	17
33	27	78	76	79
1	100	105	61	46
59	58	62	118	87
91	108	99	50	9

30	80	113	11	37
66	97	89	16	64
49	53	122	81	10
74	55	42	23	34
20	82	60	40	54

123	100	45	85	5
109	117	84	115	42
96	93	18	8	7
73	25	50	68	30
61	97	81	19	80

43	55	17	15	9
53	21	92	91	116
78	120	40	69	48
52	113	62	122	124
44	41	36	89	107

90	54	82	51	72
95	98	112	39	88
32	24	63	102	83
29	87	14	28	71
67	20	27	75	23

47	38	66	119	111
11	4	64	13	59
2	57	86	6	70
99	35	34	105	12
37	125	60	56	106

46	104	77	74	65
110	26	79	22	16
118	33	94	1	31
114	101	76	58	10
121	108	49	103	3

Plot Objective Value terhadap Iterasi

Jumlah iterasi untuk setiap restart :

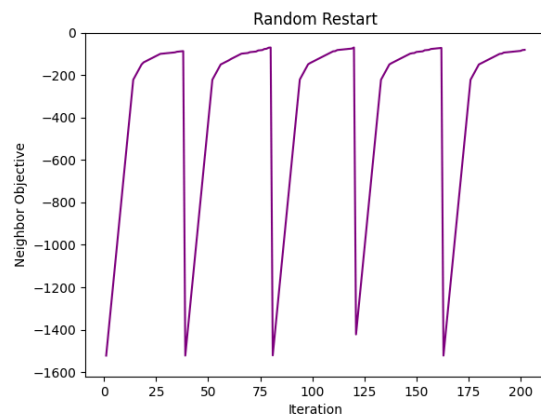
Iterasi untuk restart ke-1 : 40

Iterasi untuk restart ke-2 : 44

Iterasi untuk restart ke-3 : 42

Iterasi untuk restart ke-4 : 44

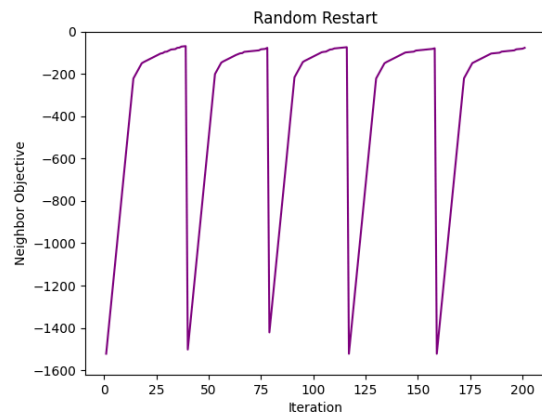
Iterasi untuk restart ke-5 : 42



Percobaan 2 : Jumlah restart 5

Objective Value & Durasi	<div>state awal: -2522</div> <div>best objective value: -69</div> <div>duration: 29.064345836639404</div>
State Awal	State Akhir Terbaik
<div>111 101 4 34 110</div> <div>40 10 8 61 100</div> <div>49 55 16 104 92</div> <div>27 37 122 7 96</div> <div>52 108 29 88 118</div> <div>95 93 71 124 77</div> <div>6 73 116 117 114</div> <div>1 19 45 38 3</div> <div>67 109 47 72 60</div> <div>70 36 85 90 26</div> <div>13 48 94 53 68</div> <div>102 98 31 30 106</div> <div>2 22 112 59 41</div> <div>115 66 82 9 97</div> <div>69 74 79 25 119</div> <div>15 63 84 46 103</div> <div>62 123 23 81 35</div> <div>17 39 51 76 65</div> <div>78 57 87 83 24</div> <div>5 121 28 44 18</div> <div>105 50 54 32 91</div> <div>21 42 89 125 86</div> <div>113 43 75 80 11</div> <div>58 99 107 64 120</div> <div>56 14 12 20 33</div>	<div>111 101 4 34 110</div> <div>40 119 8 61 100</div> <div>49 55 56 104 77</div> <div>27 37 122 7 18</div> <div>52 24 113 116 21</div> <div>95 93 30 36 92</div> <div>6 73 88 117 114</div> <div>1 19 45 87 125</div> <div>22 109 47 72 71</div> <div>70 112 96 90 26</div> <div>13 48 25 32 68</div> <div>10 98 44 60 33</div> <div>89 67 63 59 41</div> <div>115 66 82 28 11</div> <div>69 91 86 94 102</div> <div>106 5 84 46 103</div> <div>62 54 79 17 35</div> <div>51 39 81 107 3</div> <div>78 9 38 53 108</div> <div>124 121 57 31 2</div> <div>105 50 123 83 74</div> <div>118 42 85 65 23</div> <div>29 43 75 80 97</div> <div>58 99 14 64 120</div> <div>16 76 12 20 15</div>

Plot Objective Value terhadap Iterasi
 Jumlah iterasi untuk setiap restart :
 Iterasi untuk restart ke-1 : 41
 Iterasi untuk restart ke-2 : 41
 Iterasi untuk restart ke-3 : 40
 Iterasi untuk restart ke-4 : 44
 Iterasi untuk restart ke-5 : 45



Percobaan 3 : Jumlah restart 5

Objective Value & Durasi

state awal:-2521

best objective value: -53

duration: 33.56778597831726

State Awal

State Akhir Terbaik

53	14	121	109	51
112	71	79	2	124
27	38	111	41	92
10	35	93	26	6
46	5	29	81	16
100	4	36	115	47
110	72	34	96	80
113	13	30	107	31
123	77	98	45	20
114	104	106	82	12
28	97	85	91	39
49	21	1	60	44
88	76	66	59	67
120	15	94	37	63
70	55	48	87	7
32	90	78	9	50
42	64	22	116	57
8	84	19	24	83
108	75	118	74	65
23	33	43	119	58
68	11	73	62	102
122	95	89	17	69
3	105	54	125	101
56	18	86	117	52
40	25	61	99	103

66	106	30	1	12
75	111	72	19	32
23	70	122	84	65
120	45	116	3	6
35	62	43	73	95
15	55	101	4	117
79	16	89	118	47
98	113	77	97	28
86	82	17	112	51
7	71	25	107	9
21	80	102	40	48
33	18	59	88	93
58	76	63	50	68
99	38	67	108	27
105	46	24	64	78
22	52	53	115	87
69	14	109	44	57
5	29	49	13	121
34	8	37	110	81
123	74	103	26	124
31	90	96	61	91
119	92	54	36	94
125	85	41	42	39
2	104	10	20	56
114	100	83	11	60

Plot Objective Value terhadap Iterasi

Jumlah iterasi untuk setiap restart :

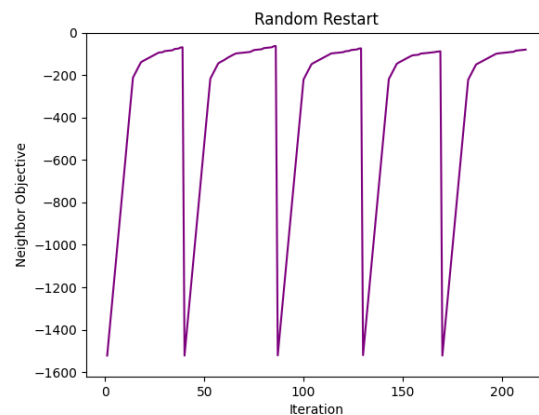
Iterasi untuk restart ke-1 : 43

Iterasi untuk restart ke-2 : 54

Iterasi untuk restart ke-3 : 49

Iterasi untuk restart ke-4 : 42

Iterasi untuk restart ke-5 : 48

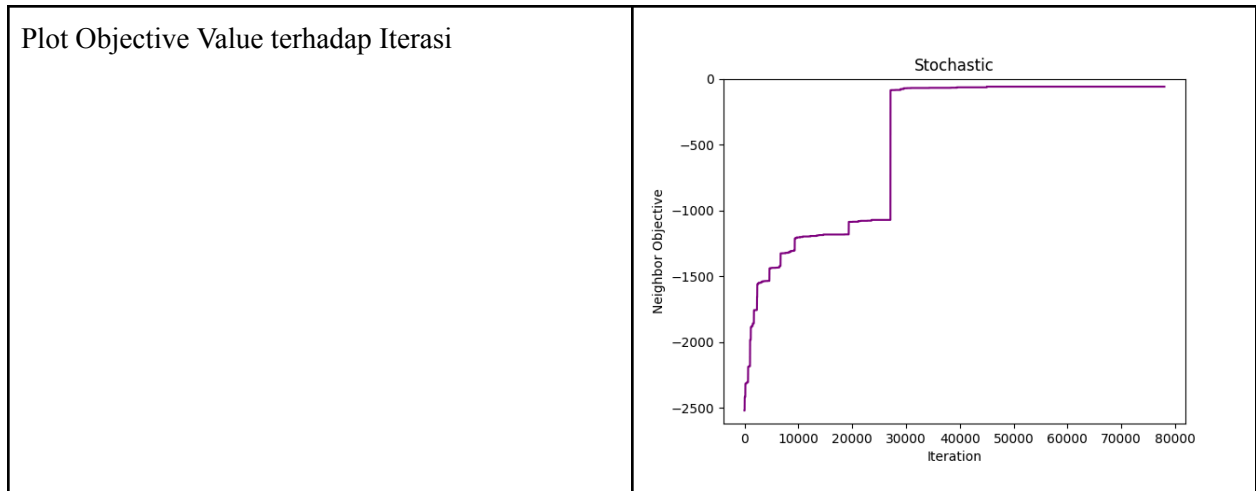


2.3.4 Stochastic Hill-climbing

Keterangan : Parameter jumlah iterasi adalah 78000

Percobaan 1

Objective Value & Durasi	<div>state awal:-2520</div> <div>iterasi 78000: -60</div> <div>duration: 51.96206259727478</div>
State Awal	State Akhir
<div>107 21 74 75 15 111 104 11 55 57 63 88 20 122 78 125 69 30 31 70 47 98 110 29 109 118 97 65 68 4 52 96 95 86 113 99 49 40 46 120 7 105 54 45 39 50 117 38 119 66 32 14 112 28 94 101 22 71 108 89 8 91 115 6 92 33 1 25 124 13 16 93 44 9 67 100 2 35 18 27 34 58 37 116 5 114 23 85 60 123 103 81 73 102 3 64 51 80 79 77 90 121 10 41 24 61 72 53 76 36 42 17 19 84 48 87 59 56 26 12 106 43 83 82 62</div>	<div>107 116 96 97 20 56 42 98 25 70 22 76 24 52 78 40 67 14 108 86 47 115 113 29 65 122 43 109 57 4 62 66 53 68 64 35 49 41 12 91 7 105 89 45 119 50 95 17 111 88 32 2 69 92 94 75 72 71 125 51 8 120 63 6 118 15 1 55 54 13 110 124 30 34 16 83 99 46 26 9 121 81 37 21 5 3 114 85 77 123 103 58 73 60 23 31 27 80 100 117 61 93 10 44 79 90 84 28 101 36 104 38 102 74 48 87 59 112 18 39 106 11 33 82 19</div>



Percobaan 2

Objective Value & Durasi	<div>state awal:-2521</div> <div>iterasi 78000: -62</div> <div>duration: 65.96958661079407</div>
State Awal	State Akhir

```

90 2 24 15 113
75 60 72 110 5
117 81 61 87 102
69 49 45 80 67
101 116 11 58 17

55 50 121 57 13
25 78 118 79 33
14 39 95 31 32
7 12 109 21 76
41 27 10 104 65

8 28 97 68 77
84 63 36 26 105
74 124 37 71 99
107 9 89 43 114
16 119 94 29 30

46 23 103 53 48
40 120 100 86 106
108 85 111 44 82
91 34 112 122 123
6 4 38 64 125

59 93 19 54 83
51 20 92 56 42
3 62 66 52 88
115 98 35 22 73
96 18 70 47 1

```

```

67 2 24 109 30
75 60 124 65 69
22 101 114 87 104
28 5 64 80 45
43 108 123 17 52

20 16 116 71 13
27 31 32 79 99
14 82 4 41 112
7 40 26 1 115
57 110 10 55 98

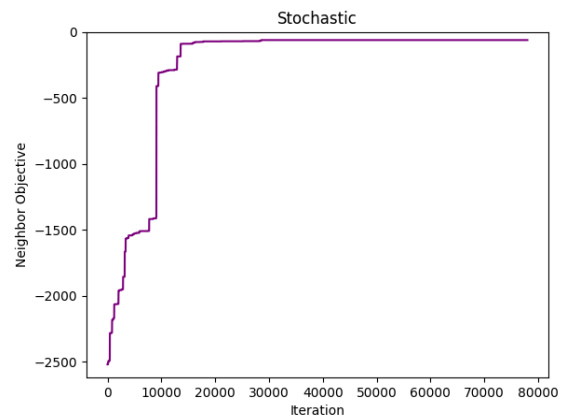
8 119 97 21 118
37 77 36 117 84
111 93 63 33 15
107 9 90 49 19
68 11 29 105 58

102 23 121 72 48
3 125 100 86 106
50 85 122 44 76
91 47 94 95 56
6 103 46 54 120

74 73 35 92 83
51 66 53 61 42
38 25 12 39 88
113 89 62 78 81
96 18 70 34 59

```

Plot Objective Value terhadap Iterasi



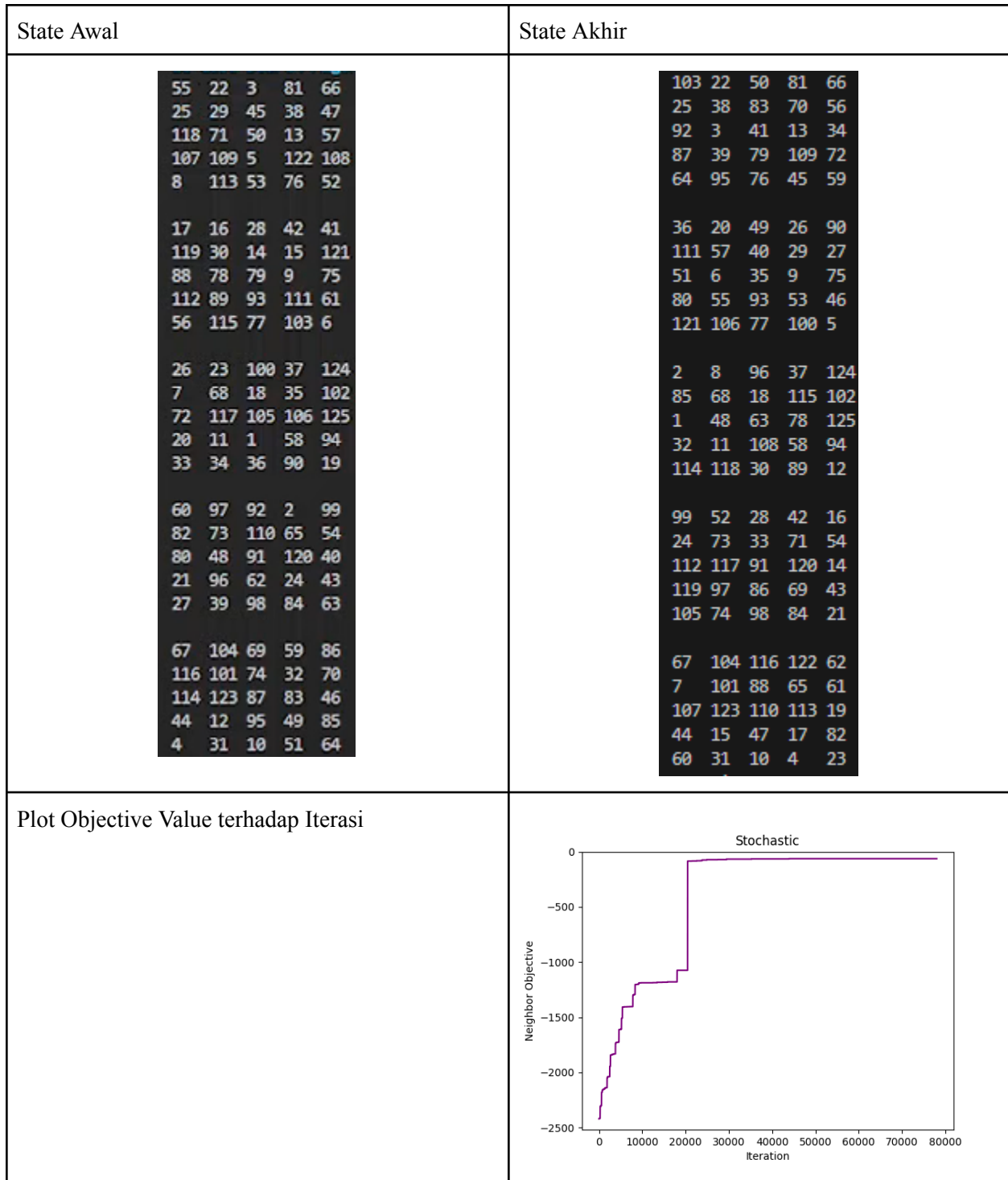
Percobaan 3

Objective Value & Durasi

```

state awal:-2421
iterasi 78000: -63
duration: 68.12341523170471

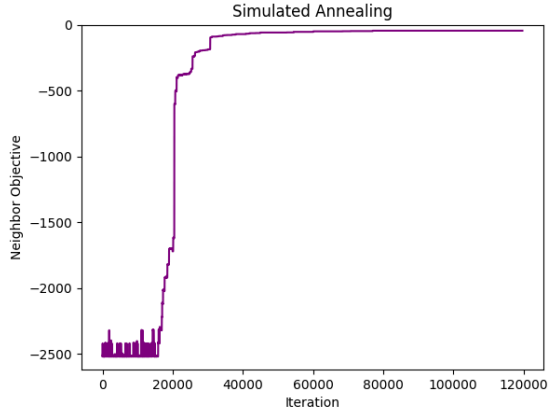
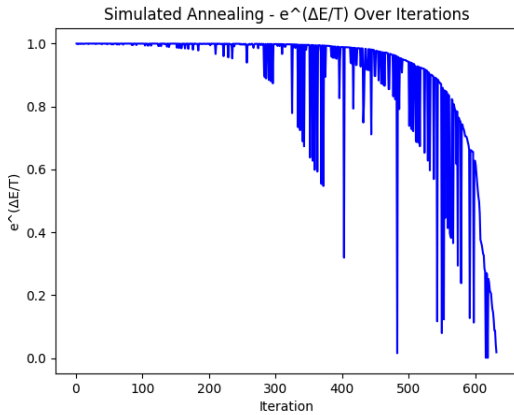
```



2.3.5 Simulated Annealing

Percobaan 1 : (Jumlah iterasi : 119705)

Objective Value & Durasi	<div>State Awal: -2521</div> <div>State Akhir: -45 stuck: 30 duration: 19.834543228149414</div>
State Awal	State Akhir
<div>35 68 29 18 71</div> <div>42 94 72 39 70</div> <div>69 90 100 21 34</div> <div>93 43 54 114 5</div> <div>16 101 65 31 22</div> <div>106 84 86 89 25</div> <div>116 53 104 33 12</div> <div>111 6 13 15 7</div> <div>57 108 66 110 63</div> <div>76 47 91 55 27</div> <div>82 32 107 73 83</div> <div>117 4 38 50 3</div> <div>98 26 92 23 126</div> <div>17 119 11 56 88</div> <div>1 14 99 115 125</div> <div>52 20 28 121 77</div> <div>105 112 10 95 96</div> <div>46 80 87 122 9</div> <div>79 59 123 44 37</div> <div>58 48 30 78 67</div> <div>103 45 64 41 102</div> <div>24 49 124 74 75</div> <div>113 51 85 8 36</div> <div>19 61 109 97 62</div> <div>81 2 60 118 40</div>	<div>110 93 44 26 28</div> <div>27 108 88 103 125</div> <div>52 92 59 7 74</div> <div>105 77 55 23 21</div> <div>41 10 82 24 22</div> <div>3 75 20 35 87</div> <div>39 15 79 94 34</div> <div>37 48 69 45 89</div> <div>17 122 2 60 109</div> <div>40 51 106 91 71</div> <div>53 96 14 11 73</div> <div>13 58 118 95 113</div> <div>12 83 63 43 114</div> <div>46 31 8 68 80</div> <div>29 30 112 115 97</div> <div>61 100 6 65 121</div> <div>90 66 124 4 36</div> <div>50 81 57 78 76</div> <div>107 32 47 111 72</div> <div>116 54 120 33 5</div> <div>104 84 117 123 85</div> <div>62 18 38 56 1</div> <div>25 86 67 119 101</div> <div>99 49 19 70 64</div> <div>98 42 9 102 16</div>

Plot Objective Value terhadap Iterasi	
Plot $e^{(\Delta E/T)}$ terhadap iterasi	

Percobaan 2 : (Jumlah iterasi : 119705)

Objective Value & Durasi	<pre> State Awal: -2522 State Akhir: -55 stuck: 195 duration: 20.74031949043274 </pre>
State Awal	State Akhir

```

34 33 114 103 36
64 65 93 50 68
6 20 83 28 22
74 29 54 7 116
99 9 48 58 121

8 57 12 35 120
53 56 78 62 81
123 117 71 14 45
60 87 88 11 85
111 80 113 5 90

89 67 91 96 112
109 69 42 66 63
4 30 13 19 15
101 97 98 108 95
77 40 39 100 3

52 17 41 107 119
105 122 47 104 26
86 32 24 125 2
18 51 73 1 124
44 43 115 23 70

61 38 31 84 118
10 16 49 55 46
75 76 79 72 94
21 102 59 37 106
25 92 110 82 27

```

```

119 87 17 13 111
1 42 93 68 89
57 24 9 26 69
88 46 105 65 85
79 25 109 113 82

8 6 78 70 2
101 74 77 66 37
3 59 115 53 123
90 34 104 55 36
122 120 62 56 39

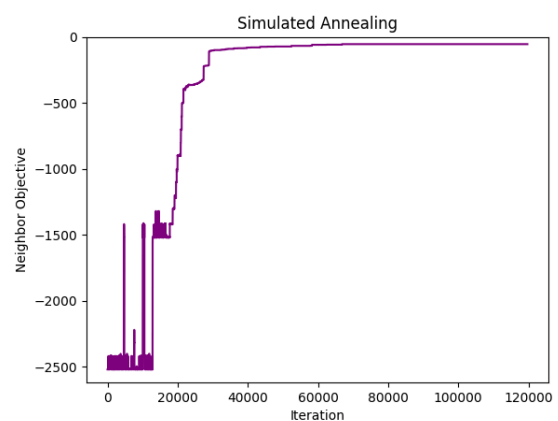
72 5 98 12 54
20 94 30 14 16
4 29 63 97 83
43 112 96 32 125
18 121 28 38 108

114 27 75 10 61
124 71 22 92 64
40 73 11 67 76
50 60 49 52 80
48 99 51 116 110

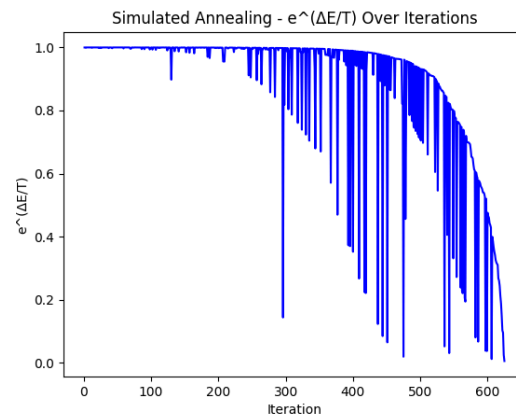
44 103 45 31 47
86 84 33 58 91
107 102 117 100 19
23 106 21 35 41
15 118 81 95 7

```

Plot Objective Value terhadap Iterasi



Plot $e^{(\Delta E/T)}$ terhadap iterasi



Percobaan 3: (Jumlah iterasi : 119705)

Objective Value & Durasi

State Awal: -2520

State Akhir: -53

stuck: 15

duration: 20.173282861709595

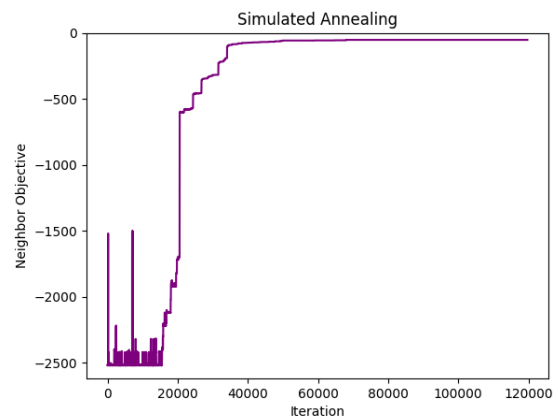
State Awal

State Akhir

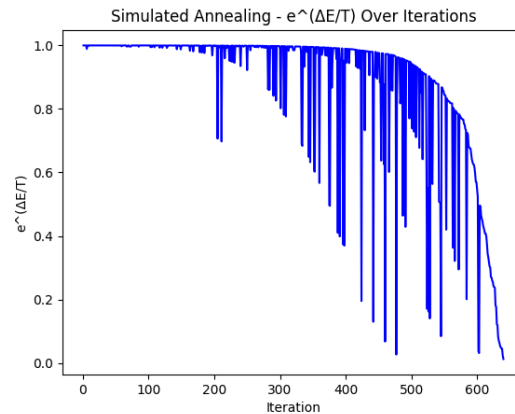
41	31	44	103	17
45	121	101	110	5
29	119	76	20	51
65	123	32	8	54
96	109	87	112	55
14	62	95	102	59
61	81	26	107	68
46	33	83	1	71
9	108	30	47	38
91	64	113	50	12
74	58	28	84	67
106	25	125	36	79
117	42	49	80	11
53	35	40	19	18
115	10	56	48	37
75	85	6	73	63
78	70	24	27	22
92	88	13	105	77
114	57	120	66	118
89	100	93	34	21
39	43	86	124	122
69	90	3	16	116
72	98	4	82	99
7	97	2	94	104
60	23	52	111	15

85	5	94	42	16
92	107	99	108	34
115	77	48	117	45
21	101	24	4	105
70	64	58	28	43
104	53	36	59	7
91	55	89	69	119
120	100	65	13	6
14	47	106	124	112
22	9	90	67	72
86	30	60	35	46
125	52	93	39	1
23	114	63	12	103
17	87	33	74	8
40	96	111	31	80
95	76	88	82	78
116	2	20	79	10
51	113	61	26	75
15	57	37	71	27
97	50	121	18	29
83	54	32	84	56
123	19	62	98	3
11	49	38	44	81
66	25	102	122	73
110	118	68	109	41

Plot Objective Value terhadap Iterasi



Plot $e^{(\Delta E/T)}$ terhadap iterasi



2.3.6 Genetic Algorithm

Max dan Avg yang berada di “Objective Value & Durasi” dikurangi 2522 untuk mendapat objective value yang negatif(Sudah tercantum dalam teks).

Percobaan 1 : Iterasi = 3000 & Jumlah Populasi = 5

Objective Value & Durasi	<div>Max: 545</div> <div>Avg: 530.2</div> <div>Duration: 20.056437492370605</div> <div>Max: -1977</div> <div>Avg: -1991.8</div> <div>Banyak Iterasi = 3000</div> <div>Jumlah Populasi = 5</div>
State Awal (Parent 1)	State Akhir (Anak 1)
objective function: -2521	objective function: -1995

<div>27 50 99 71 24 118 19 1 64 18 58 88 10 67 90 11 66 31 6 33 97 29 100 82 20 8 16 106 7 119 59 73 74 4 55 102 114 93 89 40 109 112 70 37 92 75 38 35 36 124 32 111 61 57 122 44 42 52 30 120 65 83 108 53 2 46 101 13 95 115 34 113 84 85 17 26 23 96 14 86 41 91 94 43 79 103 77 39 80 49 63 121 123 12 15 81 78 47 5 25 107 104 68 110 45 87 76 72 62 48 116 54 28 117 51 21 22 98 9 60 69 3 125 56 105</div>	<div>116 40 112 45 61 5 77 84 101 121 66 93 51 56 60 103 29 110 57 63 118 55 80 79 34 95 16 94 3 70 104 114 26 75 73 35 38 99 98 91 6 68 109 108 74 15 20 48 123 111 18 46 96 32 44 115 30 37 39 117 54 43 100 83 85 69 122 89 14 82 31 41 76 72 78 27 58 13 1 59 64 86 17 42 88 107 28 105 67 19 87 8 62 12 119 7 102 113 52 36 97 24 9 10 106 25 50 125 2 22 4 90 92 124 11 23 49 33 71 81 65 53 120 47 21</div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2522</div>	<div>objective function: -1995</div>

<div>25 85 8 53 62 92 81 49 5 115 106 97 3 34 29 88 41 79 20 109 42 111 36 13 23 9 70 65 124 17 48 61 35 45 38 50 68 63 71 16 116 22 26 87 6 46 83 98 64 76 102 24 93 94 96 103 15 55 114 59 43 44 125 18 107 84 57 73 101 31 95 69 47 7 112 100 104 12 21 10 60 118 30 28 82 110 123 121 40 117 33 67 122 58 37 108 19 74 80 52 113 1 91 90 99 51 105 2 66 27 78 75 56 120 54 32 77 119 86 11 4 39 72 89 14</div>	<div>116 40 112 45 61 5 77 84 101 121 66 93 51 56 60 103 29 110 57 63 118 55 80 79 34 95 16 94 3 70 104 114 26 75 73 35 38 99 98 91 6 68 109 108 74 15 20 48 123 111 18 46 96 32 44 115 30 37 39 117 54 43 100 83 85 69 122 89 14 82 31 41 76 72 78 27 58 13 1 59 64 86 17 42 88 107 28 105 67 19 87 8 62 12 119 7 102 113 52 36 97 24 9 10 106 25 50 125 2 22 4 90 92 124 11 23 49 33 71 81 65 53 120 47 21</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2521</div>	<div>objective function: -1995</div>

<div>103 125 56 122 90 84 64 124 80 33 70 25 62 95 48 92 117 24 22 47 75 40 85 106 105 82 35 114 3 45 38 54 110 71 74 4 50 89 30 87 44 26 19 109 96 6 5 37 29 73 102 8 68 11 1 112 28 34 72 31 77 116 39 118 13 108 94 93 46 17 119 20 16 101 58 83 91 57 18 36 7 111 65 12 69 86 81 115 123 76 15 52 51 27 99 79 120 14 53 107 63 55 121 97 104 61 10 23 32 43 9 59 66 21 78 2 67 88 42 113 41 60 100 49 98</div>	<div>116 40 112 45 61 5 77 84 101 121 66 93 51 56 60 103 29 110 57 63 118 55 80 79 34 95 16 94 3 70 104 114 26 75 73 35 38 99 98 91 6 68 109 108 74 15 20 48 123 111 18 46 96 32 44 115 30 37 39 117 54 43 100 83 85 69 122 89 14 82 31 41 76 72 78 27 58 13 1 59 64 86 17 42 88 107 28 105 67 19 87 8 62 12 119 7 102 113 52 36 97 24 9 10 106 25 50 125 2 22 4 90 92 124 11 23 49 33 71 81 65 53 120 47 21</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2521</div>	<div>objective function: -1977</div>

<div>38 102 110 94 95 105 21 115 37 68 60 109 98 82 44 66 34 123 6 1 3 51 49 32 76 85 64 100 70 52 16 84 114 40 36 73 5 30 119 107 80 61 2 96 103 31 99 101 75 93 89 90 69 77 39 41 20 72 67 106 15 71 53 48 65 113 54 62 26 79 17 121 87 125 9 18 104 78 58 27 11 124 120 23 97 92 74 8 112 22 81 4 59 83 19 43 46 47 118 12 108 57 56 14 63 10 91 13 45 88 35 55 24 117 7 33 42 25 122 28 29 50 116 86 111</div>	<div>116 40 112 45 61 5 77 84 101 121 66 93 51 56 60 103 29 110 57 63 118 55 80 79 34 95 16 94 3 70 104 114 26 75 73 35 38 99 98 91 6 68 109 108 74 15 20 48 123 111 18 46 96 32 44 115 30 37 39 117 54 43 100 83 85 69 122 89 14 82 31 41 76 72 78 27 58 13 1 59 64 86 17 42 88 107 28 52 67 19 87 8 62 12 119 7 102 113 81 36 92 24 9 10 106 25 50 97 2 22 4 90 125 124 11 23 49 33 71 105 65 53 120 47 21</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2522</div>	<div>objective function: -1997</div>

```

30 7 123 21 31
111 18 122 32 113
66 27 61 52 50
65 68 2 8 117
55 109 85 114 44

112 53 37 99 36
95 34 116 108 19
56 77 92 41 82
59 5 46 58 39
23 62 119 6 125

47 118 88 93 26
104 40 100 10 105
89 101 70 24 29
57 124 76 106 83
90 97 60 16 25

81 22 12 38 14
28 110 74 43 42
3 67 9 45 35
11 94 86 102 107
79 51 48 91 84

63 1 121 103 80
87 78 49 75 15
54 98 64 73 71
13 115 20 4 120
17 33 96 69 72

```

```

116 40 112 45 61
5 77 84 101 23
66 93 51 56 60
103 29 110 57 63
118 55 80 79 34

95 16 94 3 70
104 114 26 75 73
35 38 99 98 91
6 68 109 108 74
15 20 48 123 111

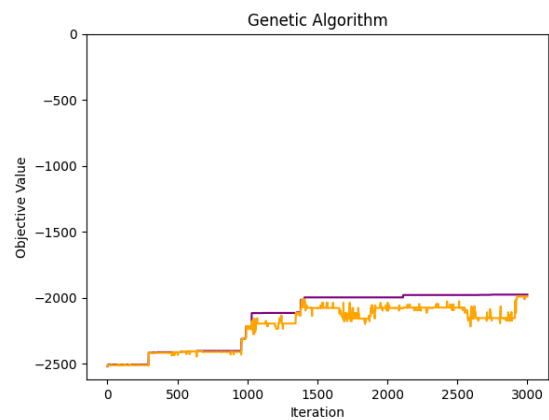
18 46 96 32 44
115 30 37 39 117
115 30 37 39 117
54 43 100 83 85
69 122 89 14 82
31 41 76 72 78

27 58 13 1 59
64 86 17 42 88
107 28 105 67 19
87 8 62 12 119
7 102 113 52 36

97 24 9 10 106
25 50 125 2 22
4 90 92 124 11
121 49 33 71 81
65 53 120 47 21

```

Plot Objective Value terhadap Iterasi



Percobaan 2 : Iterasi = 30000 & Populasi = 5

Objective Value & Durasi

```

Max: 2270
Avg: 1348.2

Duration: 200.84551358222961

```

	Max: -252 Avg: -1,173.8 Banyak Iterasi = 30000 Jumlah Populasi = 5
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2421</div> <div> 34 108 17 58 22 42 1 125 70 18 81 48 53 36 20 21 59 117 104 64 4 72 116 115 80 25 41 96 103 120 45 87 65 78 112 99 37 35 76 111 27 2 69 95 109 121 107 105 114 118 5 102 10 85 110 56 75 113 79 57 77 12 84 26 122 52 15 19 23 74 100 71 63 93 91 55 62 13 30 66 86 47 8 83 50 31 7 68 88 106 90 46 97 39 49 98 89 38 73 3 44 67 123 33 24 6 40 14 119 61 94 28 43 9 51 124 54 92 82 101 29 11 32 16 60 </div>	<div>objective function: -1403</div> <div> 116 120 104 37 123 66 70 118 4 60 108 114 81 13 42 101 53 35 1 54 45 72 22 88 56 83 93 115 38 102 34 87 47 46 86 5 41 58 59 121 36 99 69 31 90 43 94 11 117 23 17 32 112 91 122 9 28 107 21 55 29 75 63 51 103 39 12 30 52 48 6 3 10 97 106 68 124 96 14 27 111 80 89 61 92 64 7 2 65 105 67 95 33 78 18 125 110 82 57 49 71 119 85 16 15 26 109 100 79 19 77 76 44 113 62 25 73 24 84 50 20 40 74 8 98 </div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2503</div>	<div>objective function: -252</div>

<div>96 9 10 113 18 76 87 83 116 73 62 17 44 46 74 54 114 125 23 12 57 91 82 95 63 3 118 50 94 84 16 38 117 26 1 65 31 97 15 45 122 85 7 33 55 19 32 105 66 119 81 52 21 14 68 104 37 22 70 75 20 30 8 49 106 112 101 64 93 120 59 5 56 108 43 67 53 99 47 80 121 11 6 124 35 41 72 103 90 13 34 115 86 107 111 36 60 71 48 100 40 110 28 79 69 58 61 88 29 4 102 42 92 39 89 51 109 27 24 98 25 123 2 78 77</div>	<div>116 120 104 37 123 66 70 118 4 60 108 114 81 13 42 101 53 35 1 54 45 72 22 88 56 83 93 115 38 102 34 87 47 46 86 5 41 58 59 121 36 99 69 31 90 43 94 11 117 23 17 32 112 91 122 9 28 107 21 55 29 75 63 51 103 78 105 19 98 48 119 20 52 97 7 2 124 96 12 77 111 95 57 27 82 16 67 68 64 110 14 80 79 39 18 125 92 106 89 49 15 6 85 24 76 26 109 100 33 8 61 71 44 113 62 25 73 65 84 50 3 40 74 30 10</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2522</div>	<div>objective function: -252</div>

<div>118 67 73 16 116 2 6 28 114 87 71 108 99 83 53 94 24 52 70 29 103 86 1 62 37 32 9 95 7 75 25 59 38 93 115 35 50 81 13 89 119 124 60 121 42 110 12 109 23 11 69 64 22 58 47 31 90 112 61 10 30 49 96 36 105 3 26 102 72 5 63 111 77 122 56 33 84 55 51 54 107 65 14 98 68 18 41 76 117 120 106 91 78 44 57 21 97 88 125 4 34 74 101 17 27 82 80 85 113 46 15 92 8 39 45 19 100 104 123 79 40 48 66 20 43</div>	<div>116 120 104 37 123 66 70 118 4 60 108 114 81 13 42 101 53 35 1 54 45 72 22 88 56 83 93 115 38 102 34 87 47 46 86 5 41 58 59 121 36 99 69 31 90 43 94 11 117 23 17 32 112 91 122 9 28 107 21 55 29 75 63 51 103 78 105 19 98 48 119 20 52 97 7 2 124 96 12 77 111 95 57 27 82 16 67 68 64 110 14 80 79 39 18 125 92 106 89 49 15 6 85 24 76 26 109 100 33 8 61 71 44 113 62 25 73 65 84 50 3 40 74 30 10</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2516</div>	<div>objective function: -252</div>

<div>20 44 50 98 115 96 90 19 49 61 30 108 77 45 33 47 5 52 36 22 46 65 21 102 27 70 7 43 91 15 110 67 85 105 119 116 38 124 81 103 112 37 40 2 3 11 89 29 73 35 56 32 87 55 64 69 118 48 122 39 34 104 97 59 9 68 12 66 82 125 83 111 101 42 78 16 72 18 17 31 123 121 84 41 99 54 1 13 53 94 51 93 120 62 76 26 95 80 86 71 88 117 23 28 106 63 8 113 79 14 60 92 25 57 58 6 75 100 10 109 4 24 107 74 114</div>	<div>116 120 104 37 123 66 70 118 4 60 108 114 81 13 42 101 53 35 1 54 45 72 22 88 56 83 93 115 38 102 34 87 47 46 86 5 41 58 59 121 36 99 69 31 90 43 94 11 117 23 17 32 112 91 122 9 28 107 21 55 29 75 63 51 103 78 105 19 98 48 119 20 52 97 7 2 124 96 12 77 111 95 57 27 82 16 67 68 64 110 14 80 79 39 18 125 92 106 89 49 15 6 85 24 76 26 109 100 33 8 61 71 44 113 62 25 73 65 84 50 3 40 74 30 10</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2516</div>	<div>objective function: -252</div>

```

100 31 115 60 33
78 30 106 112 76
36 69 51 19 26
101 108 124 123 103
64 74 83 45 12

84 65 99 28 73
4 97 118 121 111
61 81 22 38 90
32 59 18 68 1
63 23 3 114 104

53 80 21 119 95
54 94 8 98 44
82 110 15 62 39
46 17 34 11 7
25 35 85 9 102

6 72 14 75 120
67 29 24 42 37
125 2 113 92 41
48 50 27 91 109
77 43 52 57 5

86 16 70 105 10
79 93 89 87 13
55 71 96 116 117
49 58 66 47 56
107 122 20 88 40

```

```

116 120 104 37 123
66 70 118 4 60
108 114 81 13 42
101 53 35 1 54
45 72 22 88 56

83 93 115 38 102
34 87 47 46 86
5 41 58 59 121
36 99 69 31 90
43 94 11 117 23

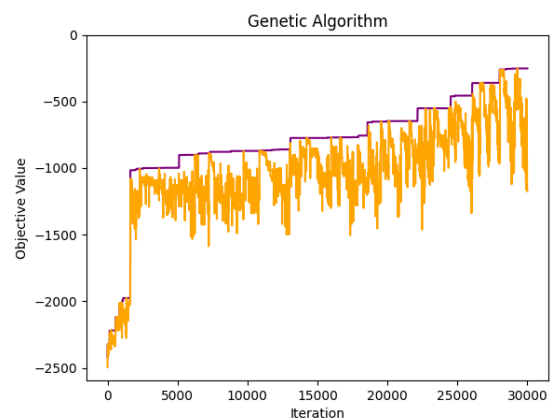
17 32 112 91 122
9 28 107 21 55
29 75 63 51 103
78 105 19 98 48
119 20 52 97 7

2 124 96 12 77
111 95 57 27 82
16 67 68 64 110
14 80 79 39 18
125 92 106 89 49

15 6 85 24 76
26 109 100 33 8
61 71 44 113 62
25 73 65 84 50
3 40 74 30 10

```

Plot Objective Value terhadap Iterasi



Percobaan 3 Iterasi = 300000 & Populasi = 5

Objective Value & Durasi

```

Max: 2466
Avg: 1854.8

```

	<div>Duration: 1887.2502200603485</div> <div>Max: -56 Avg: -667.2 Banyak Iterasi = 300000 Jumlah Populasi = 5</div>
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2521</div> <div><div>649990103 2229559580 508393135 532365169 3541729486 851081098264 4466605268 110156111745 4357106363 124371197473 12142592010 3170871292 253076104101 981257177113 11179896954 858288878 1001419123118 84241072120 4976333122 62114819734 11211111547 5140482638 10227213267 116105461791 9618395675</div></div>	<div>objective function: -745</div> <div><div>81101113932 8192411926 106951123116 255706190 936843530 124461087665 91333810431 6959754757 4111012742 12801850114 6864863758 231117107103 53102632440 101910967105 97621214434 21252820120 229911872125 6794811574 11354889345 5251297114 96436087117 7398122749 585161383 100895612377 9482661578</div></div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2519</div>	<div>objective function: -845</div>

<div>15 73 100 88 67 43 16 26 42 2 110 48 79 31 24 22 91 121 106 14 68 34 23 122 5 90 86 69 115 41 33 47 50 30 35 125 94 70 4 111 82 83 66 76 3 19 78 120 27 58 107 80 9 96 85 11 21 65 97 57 52 17 46 74 87 18 102 93 109 56 112 89 99 12 81 45 13 113 53 37 101 54 92 7 25 108 36 55 77 49 72 10 38 29 84 114 105 39 63 124 8 98 6 118 44 60 123 62 59 40 103 75 51 119 116 104 61 32 95 1 64 28 20 71 117</div>	<div>8 110 111 39 32 81 92 4 119 26 106 95 112 3 116 2 55 70 61 90 9 36 84 35 30 124 46 108 76 65 91 33 38 104 31 69 59 75 47 57 41 1 101 27 42 12 80 18 50 114 68 64 86 37 58 23 11 17 107 103 53 85 63 24 40 10 19 109 67 105 97 62 121 44 34 21 25 28 20 120 22 99 118 72 125 6 79 48 115 74 113 54 88 93 45 52 51 29 71 14 96 43 60 87 117 89 98 122 7 49 5 102 16 13 83 100 73 56 123 77 94 82 66 15 78</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2522</div>	<div>objective function: -745</div>

<div>95 73 110 64 89 2 65 102 69 46 18 124 13 28 90 100 88 14 6 54 118 72 121 70 112 67 30 116 86 103 68 97 10 119 41 96 7 8 58 9 20 75 31 53 50 122 63 123 62 11 45 25 114 52 24 99 81 92 42 49 105 44 4 91 57 66 26 87 79 48 107 120 38 23 113 32 40 17 34 84 101 74 33 109 83 37 117 3 115 111 78 43 94 27 22 21 93 125 80 71 39 19 5 1 104 106 47 51 82 36 59 85 29 16 76 55 61 77 35 15 12 60 56 108 98</div>	<div>8 110 111 39 32 81 92 4 119 26 106 95 112 3 116 2 55 70 61 90 9 36 84 35 30 124 46 108 76 65 91 33 38 104 31 69 59 75 47 57 41 1 101 27 42 12 80 18 50 114 68 64 86 37 58 23 11 17 107 103 53 102 63 24 40 10 19 109 67 105 97 62 121 44 34 21 25 28 20 120 22 99 118 72 125 6 79 48 115 74 113 54 88 93 45 52 51 29 71 14 96 43 60 87 117 73 98 122 7 49 5 85 16 13 83 100 89 56 123 77 94 82 66 15 78</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -1521</div>	<div>objective function: -945</div>

<div>114 100 89 113 33 58 81 83 59 84 52 50 42 16 77 110 92 60 102 22 86 48 78 45 18 103 26 2 39 1 53 34 14 51 36 82 38 101 79 61 9 123 87 49 122 76 35 95 21 28 32 12 72 55 46 88 70 115 120 117 15 94 63 54 30 108 74 109 7 62 67 69 75 104 8 112 98 73 17 85 31 111 90 27 118 65 99 3 10 44 29 97 23 24 68 96 64 5 4 25 116 57 6 13 124 40 66 71 93 37 80 107 119 91 11 105 106 47 19 43 20 41 56 125 121</div>	<div>8 110 111 39 32 81 92 4 119 26 106 95 112 3 116 2 55 70 61 90 9 36 84 35 30 124 46 108 76 65 91 33 38 104 31 69 59 75 47 57 41 1 101 27 42 12 80 18 50 114 68 64 86 37 58 23 11 17 107 103 53 85 63 24 40 10 44 109 67 105 97 62 121 19 34 21 25 28 20 120 22 99 118 72 125 6 79 48 115 74 113 54 88 93 45 52 51 29 71 14 96 43 60 87 117 89 98 122 7 49 5 102 16 13 83 100 73 56 123 77 94 82 66 15 78</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2516</div>	<div>objective function: -56</div>

```

44 24 107 51 55
113 120 1 58 86
78 20 12 29 7
9 125 43 88 114
23 64 2 30 105

122 13 110 74 79
68 101 98 38 31
69 77 118 91 16
5 112 8 123 84
25 96 36 52 99

27 95 73 15 124
119 28 26 71 94
97 108 83 60 34
45 70 93 102 65
104 6 117 67 80

32 57 56 3 54
62 33 42 10 18
75 19 85 82 22
41 106 4 90 46
121 53 61 48 14

49 100 89 87 59
21 111 40 92 109
17 35 37 47 39
103 76 63 72 81
116 50 11 66 115

```

```

8 110 111 39 32
81 92 4 119 26
106 95 112 3 116
2 55 70 61 90
9 36 84 35 30

124 46 108 76 65
91 33 38 104 31
69 59 75 47 57
41 1 101 27 42
12 80 18 50 114

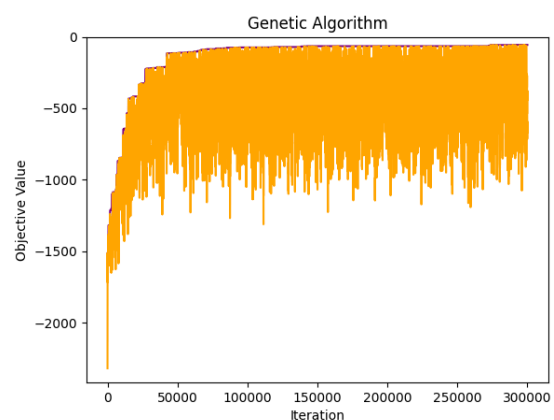
68 64 86 37 58
23 11 17 107 103
53 102 63 24 73
21 19 109 115 105
97 62 40 89 29

49 78 28 5 14
72 99 25 125 54
6 79 51 67 120
113 22 88 93 60
52 48 98 121 74

96 43 44 87 117
45 34 122 7 83
20 85 16 123 10
100 71 56 66 77
94 82 13 15 118

```

Plot Objective Value terhadap Iterasi



Percobaan 4 Iterasi = 3000 & Populasi = 7

Objective Value & Durasi

```

[338, 337, 338, 338, 369]
Max: 569
Avg: 559.4285714285714

```


	<div>Duration: 16.28183603286743</div> <div>Max: -1953</div> <div>Avg: -1,962.552</div> <div>Banyak Iterasi = 3000</div> <div>Jumlah Populasi = 7</div>
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2522</div> <div><div>13 5 107 70 33</div><div>6 17 22 113 103</div><div>123 36 28 84 67</div><div>52 30 98 34 47</div><div>72 44 10 89 35</div><div>15 114 125 57 42</div><div>56 46 65 83 124</div><div>26 75 118 18 31</div><div>21 92 50 41 51</div><div>85 76 105 29 81</div><div>9 19 104 53 78</div><div>86 43 60 116 25</div><div>120 20 95 14 69</div><div>110 59 94 11 66</div><div>24 77 79 63 91</div><div>27 58 112 2 4</div><div>82 64 108 71 111</div><div>106 61 117 12 3</div><div>40 97 7 45 115</div><div>55 39 87 1 38</div><div>93 80 54 96 16</div><div>68 100 122 119 90</div><div>62 37 32 121 48</div><div>88 109 74 101 49</div><div>8 73 23 99 102</div></div>	<div>objective function: -1964</div> <div><div>21 61 91 30 113</div><div>115 57 12 72 5</div><div>95 68 87 27 35</div><div>7 100 14 31 8</div><div>25 65 70 81 104</div><div>49 108 50 17 77</div><div>48 98 51 103 92</div><div>46 58 4 53 26</div><div>102 107 39 88 63</div><div>79 94 83 118 47</div><div>22 56 24 86 43</div><div>1 84 82 89 40</div><div>55 59 52 117 29</div><div>109 37 44 6 122</div><div>78 34 74 111 3</div><div>32 121 96 123 99</div><div>41 38 120 19 36</div><div>2 62 76 80 116</div><div>106 15 75 110 11</div><div>93 42 28 66 20</div><div>23 97 33 18 101</div><div>114 67 64 71 90</div><div>69 54 73 112 125</div><div>119 9 85 16 10</div><div>13 60 45 124 105</div></div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2321</div>	<div>objective function: -1965</div>

<div>87 61 39 28 40 31 108 56 29 114 17 73 78 54 43 53 91 2 118 102 55 124 21 122 8 49 6 41 1 103 9 89 101 125 51 70 59 35 76 99 12 107 46 38 64 79 110 22 13 47 77 72 44 69 48 10 45 82 3 67 117 11 50 83 120 109 15 63 121 112 111 66 42 4 32 115 105 25 62 86 100 90 80 19 81 94 71 85 5 95 18 33 75 92 52 113 98 65 16 20 84 26 30 14 96 123 34 93 88 116 60 27 119 24 23 106 7 58 37 74 36 68 97 57 104</div>	<div>21 61 91 30 113 115 57 12 72 77 95 68 87 27 35 7 100 14 31 8 25 65 70 81 104 49 108 50 17 5 48 98 51 103 92 46 58 4 53 26 102 107 39 88 63 79 94 83 118 47 22 56 24 86 43 1 84 82 89 40 55 59 52 117 29 109 37 44 6 122 78 34 74 111 3 32 121 96 123 99 41 38 120 19 36 2 62 76 80 116 106 15 75 110 11 93 42 28 66 20 23 97 33 18 101 114 67 64 71 90 69 54 73 112 125 119 9 85 16 10 13 60 45 124 105</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2521</div>	<div>objective function: -1964</div>

<div>58 24 2 81 43 118 123 11 47 9 39 111 117 53 61 56 119 52 34 54 40 121 51 7 62 116 94 31 112 84 35 82 107 49 25 80 18 106 98 108 65 78 66 67 79 70 101 87 120 60 72 75 57 125 10 1 103 96 124 122 14 22 76 64 4 97 104 55 63 92 15 77 73 17 105 99 59 29 3 27 32 109 28 90 68 88 69 37 12 113 6 95 100 21 86 115 50 5 85 13 71 91 42 16 20 33 41 8 30 38 19 23 45 44 48 74 89 93 36 110 114 26 46 102 83</div>	<div>21 61 91 30 113 115 60 12 72 5 95 68 87 27 35 7 100 14 31 8 25 65 70 81 104 49 108 50 17 77 48 98 51 103 29 46 58 4 53 26 102 107 39 88 63 79 94 83 118 47 22 56 24 86 43 1 84 82 89 40 55 59 52 117 92 109 37 44 6 122 78 34 74 111 3 32 121 96 123 99 41 38 120 19 36 2 62 76 80 116 106 15 75 110 11 93 42 28 66 20 23 97 33 18 101 114 67 64 71 90 69 54 73 112 125 119 9 85 16 10 13 57 45 124 105</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2522</div>	<div>objective function: -1964</div>

<div>53 108 51 110 26 121 50 78 73 19 107 12 43 105 119 83 34 106 94 67 63 38 85 70 21 82 62 45 77 2 114 102 41 93 8 18 61 13 30 91 99 65 14 35 103 125 3 122 59 22 113 97 17 6 84 28 23 31 120 68 57 109 116 40 72 100 9 55 58 16 10 66 33 101 80 47 15 48 4 60 46 29 36 117 71 123 1 86 20 124 64 104 37 98 49 39 5 89 32 25 118 115 24 27 75 7 87 88 56 96 74 69 76 95 54 79 42 111 92 90 112 52 11 81 44</div>	<div>21 61 91 30 113 115 94 117 72 5 95 68 87 27 35 7 100 14 31 8 25 65 70 81 104 49 108 50 17 77 48 98 51 103 29 46 58 4 53 26 102 107 39 88 63 79 57 83 118 47 22 56 24 86 43 1 84 82 89 40 55 59 52 12 92 109 37 44 6 122 78 34 74 111 3 32 121 96 123 99 41 38 120 19 36 2 62 76 80 116 106 15 75 110 11 93 42 28 66 20 23 97 33 18 101 114 67 64 71 90 69 54 73 112 125 119 9 85 16 10 13 60 45 124 105</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2422</div>	<div>objective function: -1953</div>

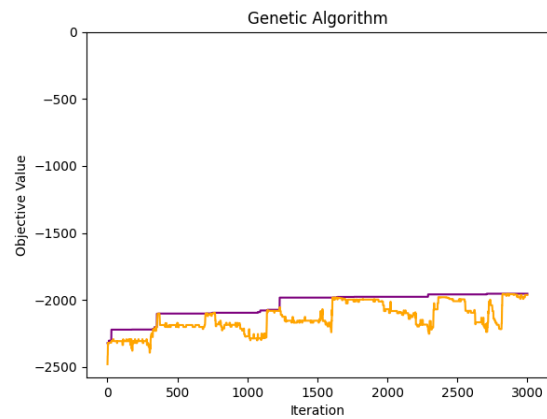
<div>94 43 59 101 17 99 83 87 116 41 48 2 39 54 103 78 119 108 110 28 96 106 71 105 88 52 82 112 46 4 21 32 95 8 63 123 3 49 5 124 14 51 57 58 72 35 98 68 36 73 67 76 15 85 13 70 120 62 84 91 16 29 34 75 125 40 42 102 7 20 115 24 10 90 86 6 81 37 44 97 22 55 77 53 9 64 107 25 45 113 92 121 79 80 23 122 27 69 74 26 50 114 47 111 38 11 118 18 65 60 19 61 93 33 30 109 31 89 56 104 1 100 117 66 12</div>	<div>21 61 91 30 113 115 57 12 72 5 95 68 87 27 35 7 100 14 31 8 25 65 70 81 104 49 108 50 17 77 48 98 51 103 29 46 58 4 53 26 102 107 39 88 63 79 94 83 118 47 22 56 24 86 43 1 84 82 89 40 55 59 52 117 64 109 37 44 34 120 66 6 125 111 3 32 121 96 123 99 41 38 122 19 36 2 62 76 80 116 106 15 75 110 11 93 42 28 78 20 23 97 33 18 101 92 67 114 71 90 69 54 73 112 74 119 9 85 16 10 13 60 45 124 105</div>
State Awal (Parent 6)	State Akhir (Anak 6)
<div>objective function: -2522</div>	<div>objective function: -1963</div>

<div>118 71 120 29 60 15 101 93 62 64 91 119 121 52 12 104 42 94 105 28 55 5 43 88 59 32 58 24 46 110 123 21 22 6 56 50 61 76 38 82 68 31 11 87 86 54 83 80 13 45 102 14 51 40 41 9 3 74 122 112 53 85 33 67 124 72 95 79 117 69 17 36 125 89 103 34 26 20 114 57 18 66 84 16 115 78 98 39 75 2 99 97 111 113 37 23 10 96 65 30 4 109 116 49 8 108 44 90 1 7 48 107 77 63 35 73 47 92 27 70 106 81 25 100 19</div>	<div>21 61 91 30 113 115 57 12 72 5 95 68 87 27 35 7 100 14 31 8 25 65 70 81 104 49 108 50 17 77 48 98 51 103 29 46 58 4 53 26 102 107 39 88 63 79 94 83 118 47 22 56 24 86 43 1 84 82 89 40 55 59 52 117 92 109 37 44 6 122 78 34 74 111 3 32 121 96 123 99 41 38 120 19 36 2 62 76 80 116 106 15 75 110 114 93 42 28 66 20 23 97 33 18 101 11 67 64 71 90 69 54 73 112 125 119 9 85 16 10 13 60 45 124 105</div>
State Awal (Parent 7)	State Akhir (Anak 7)
<div>objective function: -2521</div>	<div>objective function: -1965</div>

74	116	3	55	105
36	79	28	92	15
4	96	119	66	81
16	39	60	11	112
33	8	38	48	24
43	109	68	44	118
77	71	61	21	108
59	120	88	65	97
111	95	104	72	57
5	17	83	9	110
18	25	94	122	12
2	98	103	115	76
86	27	26	53	46
84	30	62	6	124
82	19	10	31	89
20	34	101	64	47
45	42	29	78	91
63	70	22	121	125
52	35	123	41	90
7	67	99	54	113
87	13	14	56	73
50	58	107	49	23
32	69	114	1	100
117	93	75	40	102
85	106	37	80	51

21	61	91	30	113
115	79	12	72	5
95	68	87	27	35
7	100	14	31	8
25	65	70	81	104
49	108	50	17	77
48	98	51	103	29
46	58	4	53	26
102	107	39	88	63
57	94	83	118	47
22	56	24	86	43
1	84	82	89	40
55	59	52	117	92
109	37	44	6	122
78	34	74	111	3
32	121	96	123	99
41	38	120	19	36
2	62	76	80	116
106	15	75	110	11
93	42	28	66	20
106	15	75	110	11
106	15	75	110	11
93	42	28	66	20
23	97	33	18	101
114	67	64	71	90
69	54	73	112	125
119	9	85	16	10
13	60	45	124	105

Plot Objective Value terhadap Iterasi



Percobaan 5 Iterasi = 30000 & Populasi = 7

Objective Value & Durasi	<div>Max: 1771</div> <div>Avg: 1389.857142857143</div> <div>Duration: 230.06505060195923</div> <div>Max: -751</div> <div>Avg: -1,132.143</div> <div>Banyak Iterasi = 30000</div> <div>Jumlah Populasi = 7</div>
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2522</div> <div><div>56 52 97 109 40</div><div>12 43 22 34 61</div><div>9 83 96 50 54</div><div>30 98 36 18 73</div><div>112 92 20 75 87</div><div>51 108 84 45 64</div><div>101 11 49 42 38</div><div>59 82 116 27 119</div><div>68 41 86 1 24</div><div>88 37 120 74 89</div><div>26 113 19 66 69</div><div>76 70 62 55 77</div><div>29 60 2 32 58</div><div>94 90 71 33 121</div><div>23 125 99 25 111</div><div>65 123 72 107 7</div><div>53 13 85 79 57</div><div>5 95 102 39 17</div><div>93 46 122 21 48</div><div>105 104 44 6 103</div><div>4 115 28 15 114</div><div>63 118 35 10 91</div><div>14 80 78 106 3</div><div>67 31 124 117 16</div><div>110 100 81 47 8</div></div>	<div>objective function: -1419</div> <div><div>71 73 7 85 27</div><div>114 10 68 37 12</div><div>49 4 43 16 87</div><div>8 40 103 23 112</div><div>34 19 118 70 109</div><div>121 32 44 115 91</div><div>123 80 75 97 47</div><div>48 113 88 105 35</div><div>82 56 108 6 99</div><div>90 3 33 11 79</div><div>72 28 57 98 54</div><div>119 95 29 125 59</div><div>53 66 63 20 89</div><div>55 1 15 78 25</div><div>18 36 64 38 122</div><div>124 14 13 77 45</div><div>41 61 100 5 62</div><div>86 76 101 81 96</div><div>110 120 69 51 104</div><div>65 107 106 117 50</div><div>60 83 84 46 31</div><div>92 111 9 24 2</div><div>102 116 94 17 93</div><div>26 74 42 52 58</div><div>30 67 21 39 22</div></div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2521</div>	<div>objective function: -751</div>

<div>89 121 109 10 30 38 95 18 41 96 69 23 79 71 74 19 36 107 118 110 39 83 24 124 111 80 98 49 31 51 93 62 67 84 91 4 16 87 50 40 53 44 7 75 2 92 97 9 6 54 72 55 15 112 3 102 77 32 26 115 45 22 25 63 99 12 81 120 20 94 88 105 5 14 113 73 11 104 85 52 13 61 8 103 114 68 123 1 101 86 59 82 78 66 122 64 33 119 17 58 60 35 27 28 29 37 34 100 56 90 70 46 117 108 48 47 57 42 125 43 116 21 65 106 76</div>	<div>71 73 8 85 27 114 10 68 37 12 49 4 43 16 87 7 40 103 23 112 34 19 118 70 109 121 32 44 115 91 123 80 75 97 47 48 113 88 105 35 82 56 108 6 99 90 3 33 11 79 72 28 57 98 54 119 95 29 125 59 53 66 63 60 51 101 1 15 31 25 74 67 106 64 26 96 30 89 77 124 18 120 5 45 62 102 21 93 13 69 78 61 22 46 94 65 107 38 36 2 17 104 76 58 92 84 86 9 24 20 100 122 83 110 39 117 116 42 52 14 81 41 50 111 55</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2404</div>	<div>objective function: -1418</div>

<div>11 91 86 96 3 89 78 68 23 12 58 15 56 110 82 120 51 37 94 112 16 75 26 121 97 113 61 92 6 99 57 80 21 84 39 33 4 35 32 124 76 81 102 27 67 40 122 22 117 1 87 54 2 10 20 109 95 28 73 72 123 59 36 111 71 107 85 104 31 119 60 49 69 106 53 38 24 47 79 52 25 50 116 93 83 118 44 19 65 70 13 30 90 63 48 45 14 55 5 125 29 43 41 105 62 8 42 98 101 34 64 114 108 115 103 77 74 100 46 17 66 18 7 88 9</div>	<div>71 73 8 85 27 114 10 68 37 12 49 4 43 16 87 7 40 103 23 112 34 19 118 70 109 121 32 44 115 91 123 80 75 97 47 48 113 88 105 35 82 56 108 6 99 90 3 33 11 79 72 28 57 98 54 119 95 29 125 59 53 66 63 20 89 55 1 15 78 25 18 36 64 38 122 124 14 13 77 45 41 61 100 5 62 86 76 101 81 96 110 120 69 51 104 46 107 106 117 50 60 83 84 65 31 92 111 9 24 2 102 116 94 17 93 26 74 42 52 58 30 67 21 39 22</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2522</div>	<div>objective function: -752</div>

<div>116 61 50 41 120 94 32 40 27 46 16 66 101 119 89 70 111 22 54 2 58 74 56 98 121 67 12 14 117 118 42 78 33 104 65 35 24 62 103 30 92 91 108 122 85 86 93 57 69 53 13 29 28 49 11 84 36 63 51 10 124 110 34 55 4 26 23 95 97 83 100 17 105 112 113 44 6 123 64 38 45 9 107 7 48 72 20 76 88 31 3 87 75 102 5 18 43 8 39 106 47 99 15 96 73 125 80 37 81 79 68 109 52 21 25 77 114 82 59 19 1 71 90 115 60</div>	<div>71 73 8 85 27 114 10 68 37 12 49 4 43 61 87 7 40 103 23 112 34 19 118 70 109 121 32 44 115 91 123 80 75 97 47 48 113 88 105 35 82 56 108 6 99 90 3 33 11 79 72 28 57 98 54 119 95 29 125 59 53 66 63 60 51 101 1 15 31 25 74 67 106 64 26 96 30 89 77 124 18 120 5 45 62 102 21 93 13 69 78 16 22 46 94 65 107 38 36 2 17 104 76 58 92 84 86 9 24 20 100 122 83 110 39 117 116 42 52 14 81 41 50 111 55</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2422</div>	<div>objective function: -751</div>

<div>55 109 68 74 111 25 47 85 116 36 102 92 83 115 17 63 42 87 122 15 53 95 14 20 39 56 8 10 75 40 4 5 37 86 18 112 124 72 82 106 6 60 9 105 26 65 30 73 101 2 44 22 110 113 52 66 11 77 1 80 70 97 41 16 107 33 31 19 28 120 125 99 64 46 71 43 119 117 49 114 103 98 34 94 48 35 69 54 96 118 50 93 27 104 90 45 21 121 91 67 3 57 76 58 38 32 61 79 12 29 81 123 100 23 24 108 78 89 13 88 62 59 84 51 7</div>	<div>71 73 8 85 27 114 10 68 37 12 49 4 43 16 87 7 40 103 23 112 34 19 118 70 109 121 32 44 115 91 123 80 75 97 47 48 113 88 105 35 82 56 108 6 99 90 3 33 11 79 72 28 57 98 54 119 95 29 125 59 53 66 63 60 51 101 1 15 31 25 74 67 106 64 26 96 30 89 77 124 18 120 5 45 62 102 21 93 13 69 78 61 22 46 94 65 107 38 36 2 17 104 76 58 92 84 86 9 24 20 100 122 83 110 39 117 116 42 52 14 81 41 50 111 55</div>
State Awal (Parent 6)	State Akhir (Anak 6)
<div>objective function: -2522</div>	<div>objective function: -1418</div>

<div>32 117 68 57 123 43 6 125 35 92 81 60 55 122 67 112 98 79 115 37 107 14 116 28 7 9 62 69 110 99 84 91 5 36 113 93 44 111 77 75 10 46 86 29 74 33 51 2 53 47 102 100 1 39 18 22 50 66 58 103 119 71 48 42 27 101 8 95 31 85 121 89 24 109 34 4 78 76 108 124 40 38 82 88 13 3 120 63 54 73 45 70 30 97 23 21 11 87 15 52 41 105 16 106 94 49 83 19 61 65 90 104 56 80 12 20 64 114 25 59 72 26 118 17 96</div>	<div>71 73 8 85 27 114 10 68 37 12 49 4 43 16 87 7 40 103 23 112 34 19 118 70 109 121 32 44 115 91 123 80 75 97 47 48 113 88 105 35 82 56 108 6 99 90 3 33 11 79 72 28 57 98 54 119 95 29 125 59 53 66 63 20 89 55 1 15 78 25 18 36 64 38 122 124 14 13 77 45 41 61 100 5 62 86 76 101 81 96 30 120 69 51 104 65 107 106 117 50 60 83 84 46 31 92 111 9 24 2 102 116 110 17 93 26 74 42 52 58 94 67 21 39 22</div>
State Awal (Parent 7)	State Akhir (Anak 7)
<div>objective function: -2521</div>	<div>objective function: -751</div>

```

96 27 31 55 32
125 22 44 35 107
122 23 36 75 98
71 24 70 83 105
10 7 48 76 77

62 79 51 8 113
69 28 91 100 26
1 85 4 17 88
114 16 19 74 120
102 97 110 61 63

99 29 86 46 104
53 12 41 39 57
112 80 3 64 13
115 49 54 89 60
33 6 52 40 42

43 101 90 116 5
108 18 95 37 67
2 119 20 15 78
11 92 93 87 65
30 45 68 94 121

58 118 109 111 25
73 9 84 123 56
34 47 72 38 21
103 82 66 106 117
59 50 81 124 14

```

```

71 73 8 85 27
114 10 68 37 12
49 4 43 16 87
7 40 103 23 112
34 19 118 70 109

121 32 44 115 91
123 80 75 97 47
48 113 88 105 35
82 56 108 6 99
90 3 33 11 79

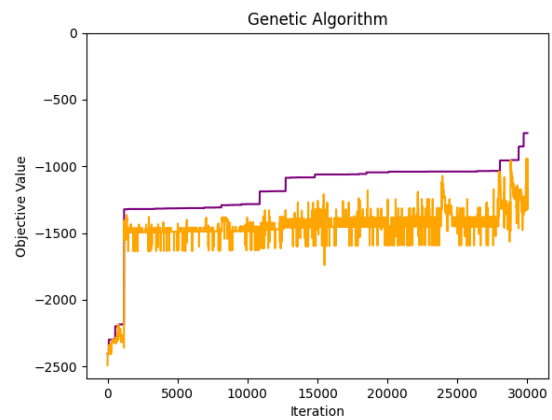
72 28 57 98 54
119 95 29 125 59
53 66 63 60 51
101 1 15 31 25
74 67 106 64 26

96 30 89 77 124
18 120 5 45 62
102 21 93 13 69
78 61 22 46 94
65 107 38 36 2

17 104 76 58 92
84 86 9 24 20
100 122 83 110 39
117 116 42 52 14
81 41 50 111 55

```

Plot Objective Value terhadap Iterasi



Percobaan 6 Iterasi = 300000 & Populasi = 7

Objective Value & Durasi

```

Max: 2461
Avg: 2068.285714285714

Duration: 2498.562263250351

```

	Max: -61 Avg: -453.72 Banyak Iterasi = 300000 Jumlah Populasi = 7
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2522</div> <div> 112 82 43 122 38 57 24 71 115 124 92 74 35 45 117 17 37 104 41 61 59 36 94 39 89 20 99 25 54 87 90 18 77 52 49 9 69 63 62 28 34 64 42 110 121 105 80 109 53 58 56 1 23 102 32 19 67 2 86 47 46 16 107 79 6 70 11 116 106 10 68 97 101 111 91 88 114 26 120 14 83 21 30 125 60 96 15 119 95 103 12 123 85 75 76 40 8 93 108 84 27 55 29 51 72 66 4 48 78 73 65 3 113 5 13 7 44 98 81 50 22 118 33 31 100 </div>	<div>objective function: -597</div> <div> 26 43 12 45 123 59 124 24 11 67 125 99 33 20 15 66 49 98 88 18 89 1 114 81 54 21 120 122 53 105 70 40 7 91 56 104 116 46 107 22 101 113 47 34 25 60 6 38 93 74 65 31 14 121 110 84 78 44 76 42 30 90 63 36 73 71 50 82 117 55 118 119 79 5 23 61 109 85 48 58 94 92 35 13 32 51 19 80 10 28 115 112 95 111 57 8 52 41 64 68 72 102 62 29 37 97 2 16 69 83 108 27 87 106 4 39 77 17 96 75 3 103 9 86 100 </div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2522</div>	<div>objective function: -61</div>

<div>40 103 114 97 91 31 108 44 74 69 57 71 116 98 49 28 45 27 68 29 37 121 111 58 12 18 77 52 17 47 51 92 61 125 88 123 1 79 60 10 78 16 30 26 96 34 75 39 3 81 4 94 56 41 95 85 5 54 104 46 13 20 89 53 48 2 9 120 115 67 19 36 122 72 100 107 80 59 109 24 102 35 25 14 82 65 62 113 87 21 99 84 32 64 93 33 42 50 70 23 11 106 73 6 76 119 63 7 55 101 112 43 86 90 83 118 110 8 124 38 117 66 105 22 15</div>	<div>26 43 12 45 123 59 124 24 11 67 125 99 33 20 15 66 49 98 88 18 89 1 114 81 54 21 120 122 53 105 70 40 7 91 56 104 116 46 107 22 101 113 47 34 25 60 6 38 93 74 65 31 14 121 110 84 78 44 76 42 30 90 63 36 73 71 50 82 48 55 61 95 112 5 16 118 51 85 117 58 94 92 79 13 32 109 19 80 10 17 115 35 119 86 57 8 52 41 9 68 72 23 62 29 37 97 2 102 69 83 108 27 87 106 4 39 77 28 96 75 3 103 64 111 100</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2522</div>	<div>objective function: -543</div>

<pre>57 25 22 10 75 29 31 1 19 112 72 43 82 106 67 44 33 102 21 14 46 91 125 63 77 89 30 116 107 24 11 62 73 15 40 87 45 26 80 94 115 20 39 104 95 59 122 70 58 79 93 51 49 100 50 118 16 18 111 101 60 88 35 123 113 97 71 41 78 76 124 68 6 7 48 61 34 4 3 81 52 85 23 55 84 96 90 27 117 119 83 47 9 109 38 121 64 54 32 114 56 42 2 17 74 103 69 8 37 99 66 92 65 13 105 12 5 36 98 86 28 120 108 53 110</pre>	<pre>26 43 12 45 123 59 124 24 11 67 125 99 33 20 15 66 49 98 88 18 89 1 114 81 54 21 64 122 53 105 70 40 7 91 56 104 116 46 107 22 101 113 47 34 25 60 6 38 93 74 65 31 14 121 110 84 78 44 76 42 30 90 63 36 73 71 50 82 117 55 118 95 79 5 23 61 109 85 48 58 94 92 35 13 32 51 19 80 10 28 115 112 119 111 57 8 52 41 120 68 72 102 62 29 37 97 2 16 69 83 108 27 87 106 4 39 77 17 96 75 100 103 9 86 3</pre>
State Awal (Parent 4)	State Akhir (Anak 4)
<pre>objective function: -2522</pre>	<pre>objective function: -543</pre>

<pre>8 77 64 62 99 76 74 75 92 5 16 125 107 123 103 82 94 66 27 86 57 39 45 29 12 115 22 7 112 14 91 21 37 26 2 87 85 28 95 56 70 84 73 104 100 54 106 120 122 30 36 1 108 35 53 110 40 15 11 10 34 109 25 18 13 78 24 71 48 90 88 9 58 19 113 6 42 23 60 114 51 83 96 49 3 65 20 52 101 59 80 17 118 69 31 98 121 4 124 55 93 119 38 102 105 79 89 44 72 32 61 47 81 117 41 97 116 63 111 33 46 67 43 50 68</pre>	<pre>26 43 12 45 123 59 124 24 11 67 125 99 33 20 15 66 49 98 88 18 89 1 114 81 54 21 64 122 53 105 70 40 7 91 56 104 116 46 107 22 101 113 47 34 25 60 6 38 93 74 65 31 14 121 110 84 78 44 76 42 30 90 63 36 73 71 50 82 117 55 118 95 79 5 23 61 109 85 48 58 94 92 35 13 32 51 19 80 10 28 115 112 119 111 57 8 52 41 120 68 72 102 62 29 37 97 2 16 69 83 108 27 87 106 4 39 77 17 96 75 100 103 9 86 3</pre>
State Awal (Parent 5)	State Akhir (Anak 5)
<pre>objective function: -2522</pre>	<pre>objective function: -601</pre>

<div>112 87 47 113 69 67 120 66 118 2 18 17 28 77 27 80 36 116 38 125 100 24 89 92 107 82 43 72 51 54 39 83 95 16 56 14 7 90 12 93 42 1 41 58 91 115 34 106 37 6 97 32 94 9 103 25 15 22 124 81 64 96 88 123 20 3 49 8 33 108 70 10 55 122 21 68 121 29 86 40 59 53 63 79 78 26 4 60 46 110 45 102 57 61 104 11 84 62 73 30 75 98 19 101 44 65 76 50 117 114 48 119 5 31 23 35 74 99 111 85 71 52 13 105 109</div>	<div>26 43 12 45 123 59 124 24 11 67 125 99 33 20 15 66 49 98 88 18 89 1 114 81 54 21 120 122 53 105 70 40 7 91 56 104 116 46 107 22 101 113 47 34 25 60 6 38 93 74 65 31 14 121 110 84 78 44 76 42 30 90 63 36 73 71 50 82 117 55 118 95 79 5 23 61 109 85 48 58 94 16 35 13 32 51 19 80 10 28 115 112 119 111 57 8 52 41 64 68 72 102 62 29 37 97 2 92 69 83 108 27 87 106 4 39 77 17 96 75 3 103 9 86 100</div>
State Awal (Parent 6)	State Akhir (Anak 6)
<div>objective function: -2521</div>	<div>objective function: -496</div>

<div>77 42 36 101 87 83 76 53 32 66 109 44 113 1 14 104 10 63 26 22 4 80 105 90 55 94 73 57 119 108 92 115 7 86 96 120 123 39 50 117 21 48 62 60 114 85 18 52 5 78 111 102 72 116 41 17 69 82 20 43 122 34 58 124 24 16 64 125 38 88 51 33 84 45 79 28 98 6 103 65 11 59 71 112 121 30 99 100 93 12 61 8 49 54 15 3 56 89 67 46 2 25 68 95 40 106 74 75 107 19 35 118 9 81 97 37 13 110 27 70 31 47 29 23 91</div>	<div>26 43 12 45 123 59 124 24 11 67 125 99 33 20 15 66 49 98 88 18 89 1 114 81 54 21 120 122 53 105 70 40 7 91 56 104 116 46 107 22 101 113 47 34 25 60 6 38 93 74 65 31 14 121 110 84 78 44 76 42 30 90 63 36 73 71 50 82 117 55 118 95 79 5 23 61 109 85 48 58 94 92 35 13 32 51 19 80 10 28 115 112 119 111 57 8 52 41 64 68 72 102 62 29 37 97 2 16 69 83 108 27 87 106 4 39 77 17 96 75 3 103 9 86 100</div>
State Awal (Parent 7)	State Akhir (Anak 7)
<div>objective function: -2522</div>	<div>objective function: -543</div>

```

7  56 46 55 79
96 81 94 31 117
87 25 34 92 100
74 88 16 11 95
38 51 62 43 102

99 5  123 57 89
3  64 80 98 1
75 120 73 105 90
40 76 69 82 124
66 42 86 58 22

67 108 91 54 77
6  50 28 10 113
72 14 101 109 13
121 20 29 52 48
97 45 84 24 116

122 104 111 78 60
106 23 18 114 107
49 4  61 35 119
110 32 118 112 47
44 59 21 71 26

53 68 37 27 85
8  125 12 83 30
36 63 93 19 103
9  15 39 17 2
33 65 41 115 70

```

```

26 43 12 45 123
59 124 24 11 67
125 99 33 20 15
66 49 98 88 18
89 1  114 81 54

21 64 122 53 105
70 40 7  91 56
104 116 46 107 22
101 113 47 34 25
60 6  38 93 74

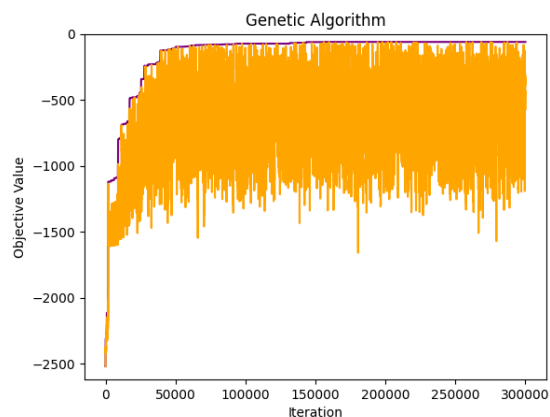
65 31 14 121 110
84 78 44 76 42
30 90 63 36 73
71 50 82 117 55
118 95 79 5  23

61 109 85 48 58
94 92 35 13 32
51 19 80 10 28
115 112 119 111 57
8  52 41 120 68

72 102 62 29 37
97 2  16 69 83
108 27 87 106 4
39 77 17 96 75
100 103 9  86 3

```

Plot Objective Value terhadap Iterasi



Percobaan 7 Iterasi = 3000 & Populasi = 10

Objective Value & Durasi

```

Max: 1216
Avg: 1030.5

```

```

Duration: 29.504251956939697

```

	Max: -1,306 Avg: -1,491.5 Banyak Iterasi = 3000 Jumlah Populasi = 10
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2522</div> <div> 51 70 96 115 62 54 69 7 107 124 59 31 71 63 106 95 111 4 76 65 82 24 89 44 75 108 47 103 35 90 120 81 6 30 40 61 101 92 19 28 37 46 43 91 97 16 3 33 114 50 87 38 118 93 15 12 29 74 78 84 109 88 20 57 60 117 8 66 110 22 80 113 64 36 73 18 98 72 116 53 10 27 26 56 85 94 42 1 23 58 104 86 49 102 77 11 52 17 125 121 67 5 2 39 123 45 55 14 105 32 122 21 119 100 99 68 34 112 41 83 25 13 79 48 9 </div>	<div>objective function: -1306</div> <div> 37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 124 86 97 109 26 83 23 2 114 73 55 34 102 101 66 82 100 117 63 62 7 105 80 88 24 38 65 99 46 11 28 47 111 64 22 87 121 50 94 40 79 103 81 27 61 30 1 59 42 12 13 52 18 49 95 5 75 31 107 43 8 93 9 104 72 36 48 120 115 125 39 78 113 118 112 15 29 74 77 4 25 </div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2521</div>	<div>objective function: -1511</div>

<div>98 37 81 61 70 62 9 19 11 63 119 5 33 15 65 77 51 52 109 74 110 100 115 35 93 54 57 76 113 72 116 26 47 68 123 22 71 29 107 89 125 78 111 18 44 8 2 102 87 60 120 7 27 88 104 4 1 38 122 105 59 34 99 58 36 43 67 39 48 64 92 85 79 84 82 3 55 80 42 50 30 32 90 23 106 24 94 91 14 69 118 17 124 28 117 112 46 56 95 25 12 86 53 20 41 10 121 83 73 103 66 97 96 49 16 101 13 31 6 21 45 108 114 40 75</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 113 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 94 9 103 121 52 29 104 87 15 88 39 64 38 112 118 93 50 7 28 46 99 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 12 31 61 47 22 11 65 49 124 72 27 24 120 40 36 48</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2522</div>	<div>objective function: -1511</div>

<div>39 122 51 22 5 18 26 13 106 77 35 90 37 59 56 73 102 7 99 31 48 40 41 82 9 68 116 111 19 23 104 79 120 49 11 115 12 125 94 92 124 105 62 6 50 76 61 8 78 28 96 112 64 29 74 85 58 86 57 98 91 87 1 108 52 75 32 83 2 54 20 16 47 117 89 93 14 44 67 100 113 42 66 3 65 69 84 27 123 10 46 109 95 110 43 71 4 70 103 25 80 36 21 81 34 24 97 53 15 38 72 114 118 30 63 101 119 17 45 121 60 33 55 88 107</div>	<div>37 85 60 84 69 119 89 35 32 113 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 124 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 22 112 9 103 121 52 29 104 87 15 88 39 64 38 46 118 93 50 7 28 99 12 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 11 31 61 47 43 49 65 94 71 72 27 24 120 40 36 48</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2521</div>	<div>objective function: -1511</div>

<div>62 89 42 5 92 73 93 121 50 45 15 3 44 120 72 115 125 74 119 8 106 26 81 18 24 70 51 111 117 2 83 13 23 103 30 80 11 20 82 57 35 79 94 14 98 58 33 4 97 12 84 77 110 68 59 31 41 123 69 22 104 67 56 112 71 53 9 60 43 1 27 109 16 90 76 36 108 34 87 55 107 100 54 10 65 86 124 29 113 48 75 38 116 88 102 17 25 66 96 7 32 46 99 85 122 91 101 95 61 49 40 28 63 47 52 19 78 114 118 105 6 64 21 37 39</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 124 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 94 9 103 121 52 29 104 87 15 88 39 64 38 112 118 93 50 7 28 46 99 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 12 31 61 47 22 11 65 49 113 72 27 24 120 40 36 48</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2522</div>	<div>objective function: -1511</div>

<div>8 51 41 6 99 12 15 7 108 103 76 45 9 91 30 43 14 73 11 100 110 25 21 2 75 74 72 67 69 4 48 116 40 42 88 32 47 19 63 38 39 96 125 106 17 55 89 93 83 104 94 102 33 122 66 107 13 86 97 29 54 95 114 79 27 58 65 44 3 101 119 115 50 112 62 26 23 53 59 121 70 82 28 56 5 18 113 37 92 71 31 105 52 80 117 81 77 85 24 36 61 124 109 68 46 98 78 49 57 22 35 120 64 10 118 20 34 1 16 84 123 90 111 87 60</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 113 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 112 9 103 121 52 29 104 87 15 88 39 64 38 46 118 93 50 7 28 99 12 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 11 31 61 47 22 49 65 94 124 72 27 24 120 40 36 48</div>
State Awal (Parent 6)	State Akhir (Anak 6)
<div>objective function: -2422</div>	<div>objective function: -1516</div>

<div>24 9 31 54 59 48 125 7 36 13 123 53 38 34 71 4 73 35 79 69 99 51 1 21 86 27 37 72 77 5 65 41 94 28 83 110 22 39 63 45 50 61 122 98 62 97 30 119 78 74 10 96 88 85 64 67 80 120 11 16 40 49 47 102 66 108 70 112 118 3 82 32 2 104 58 124 107 19 43 91 111 15 117 68 95 52 57 87 56 29 75 90 20 14 8 42 84 92 12 76 116 46 101 33 17 100 55 60 81 6 93 26 106 23 89 103 109 121 105 18 25 114 113 115 44</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 23 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 124 86 97 109 74 83 123 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 94 9 103 121 52 29 104 87 15 88 39 64 38 112 46 93 50 7 28 99 118 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 12 31 61 47 22 11 65 49 113 72 27 24 120 40 36 48</div>
State Awal (Parent 7)	State Akhir (Anak 7)
<div>objective function: -2416</div>	<div>objective function: -1511</div>

<div>97 49 64 67 41 11 92 90 77 48 36 114 26 51 99 98 121 87 42 96 69 61 84 85 13 122 68 29 108 40 3 56 79 75 125 93 44 95 45 22 94 43 24 119 5 25 35 120 82 117 12 39 116 21 103 110 89 115 17 23 123 30 124 106 37 107 28 55 57 7 86 53 118 74 71 91 2 83 8 111 70 47 10 52 33 76 54 31 59 66 81 14 80 15 113 32 9 1 18 19 38 88 62 104 102 46 50 72 105 65 27 78 4 112 63 34 6 20 73 60 16 100 58 101 109</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 113 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 94 9 103 121 52 29 104 87 15 88 39 64 38 112 118 93 50 7 28 46 99 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 12 31 61 47 22 11 65 49 124 72 27 24 120 40 36 48</div>
State Awal (Parent 8)	State Akhir (Anak 8)
<div>objective function: -2421</div>	<div>objective function: -1511</div>

<div>37 28 101 84 69 104 89 35 30 116 9 46 51 71 39 110 38 92 106 64 67 76 21 88 58 123 55 96 82 105 26 86 122 91 1 107 62 78 45 53 121 117 19 83 56 6 57 2 95 54 65 13 72 73 10 109 102 18 70 112 100 41 48 20 93 49 80 60 24 77 118 43 52 79 33 47 99 59 66 74 25 114 94 90 108 81 7 103 119 97 42 17 32 12 120 23 111 87 15 31 75 3 61 44 16 85 14 5 125 11 63 8 115 124 4 68 113 22 98 34 29 40 50 36 27</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 124 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 94 9 103 121 52 29 104 87 15 88 39 64 38 47 118 93 50 7 28 46 99 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 12 31 61 112 22 11 65 49 113 72 27 24 120 40 36 48</div>
State Awal (Parent 9)	State Akhir (Anak 9)
<div>objective function: -2522</div>	<div>objective function: -1511</div>

<div>89 27 96 59 66 40 124 80 5 85 98 28 93 10 35 50 6 119 118 30 120 60 42 65 83 34 82 32 39 73 87 33 108 2 111 117 122 57 46 4 84 91 51 72 52 38 21 29 37 54 114 16 123 18 97 101 56 58 94 112 45 95 49 110 79 12 107 13 81 67 55 24 88 44 115 14 53 22 63 90 61 36 9 105 69 3 17 1 71 106 99 68 74 25 15 70 116 48 77 8 103 47 86 7 41 11 104 78 113 26 76 109 92 125 20 64 19 100 102 43 62 23 121 31 75</div>	<div>37 85 60 84 69 119 89 35 32 71 53 116 51 20 17 110 70 92 106 16 67 76 21 14 58 123 90 96 19 54 56 10 122 91 3 57 108 68 45 33 44 41 98 6 124 86 97 109 74 83 23 2 114 73 55 34 102 101 66 82 100 117 63 107 4 8 81 1 59 18 105 43 94 9 103 121 52 29 104 87 15 88 39 64 38 112 118 93 50 7 28 46 99 26 62 115 95 77 25 5 75 78 111 30 80 125 13 79 42 12 31 61 47 22 11 65 49 113 72 27 24 120 40 36 48</div>
State Awal (Parent 10)	State Akhir (Anak 10)
<div>objective function: -2522</div>	<div>objective function: -1516</div>

```

24 19 64 68 90
123 12 17 43 78
59 63 88 120 13
94 25 38 62 6
54 48 32 22 67

112 31 92 15 105
124 118 73 89 69
20 71 76 61 103
29 51 93 84 98
99 52 83 125 95

2 74 47 27 44
97 36 7 81 72
37 85 18 55 26
86 30 113 122 1
66 91 114 4 34

60 119 5 10 50
106 100 111 57 116
33 14 23 65 109
77 39 101 46 21
96 35 9 28 49

8 16 110 3 42
40 75 70 121 58
41 80 82 45 107
102 87 115 108 11
53 117 56 79 104

```

```

37 85 60 84 69
119 89 35 32 71
53 116 51 20 17
110 70 92 106 16
67 76 21 14 58

123 90 96 19 54
56 10 122 12 3
57 108 68 45 33
44 41 98 6 124
86 97 109 74 83

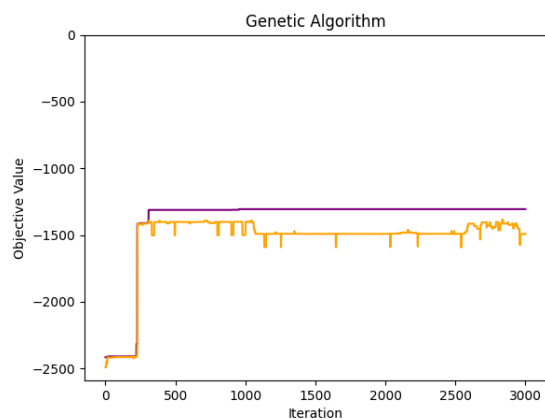
23 2 114 73 55
34 102 101 66 82
100 117 63 107 4
8 81 1 59 18
105 43 94 9 103

121 52 29 104 87
15 88 39 64 38
47 118 93 50 7
28 46 99 26 62
115 95 77 25 5

75 78 111 30 80
125 13 79 42 91
31 61 112 22 11
65 49 113 72 27
24 120 40 36 48

```

Plot Objective Value terhadap Iterasi



Percobaan 8 Iterasi = 30000 & Populasi = 10

Objective Value & Durasi

```

Max: 2170
Avg: 1233.2

```

```

Duration: 302.85709023475647

```

	Max: -352 Avg: -1,288.8 Banyak Iterasi = 30000 Jumlah Populasi = 10
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2521</div> <div> 13 83 98 42 124 52 7 118 111 95 33 79 109 65 37 63 16 71 39 113 22 92 96 58 30 74 14 9 87 99 35 112 41 117 125 8 55 60 31 2 57 54 12 101 90 110 59 94 86 82 23 43 115 106 107 93 3 104 102 121 48 108 75 25 69 123 49 38 76 116 80 29 88 4 100 119 34 26 56 64 20 84 62 15 91 67 103 61 105 19 50 27 28 5 66 21 32 17 97 53 72 73 1 77 120 89 10 78 11 51 24 47 36 45 18 68 40 46 114 122 85 44 70 6 81 </div>	<div>objective function: -352</div> <div> 86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 59 102 124 101 95 87 2 103 19 75 61 114 15 43 45 79 1 106 98 105 85 20 49 17 57 50 77 18 48 80 3 38 32 29 97 73 21 64 23 83 70 100 13 52 71 24 94 10 111 108 104 34 68 53 12 125 72 112 42 88 55 14 </div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2522</div>	<div>objective function: -1415</div>

<div>95 94 54 39 37 99 55 114 89 40 11 96 100 45 17 69 109 67 90 107 64 2 12 70 84 41 86 56 60 9 49 43 22 108 106 76 65 33 79 53 82 115 47 63 5 35 80 50 93 111 125 121 38 66 52 118 102 42 81 88 91 18 4 87 13 32 59 110 117 97 123 51 15 34 6 122 116 112 23 119 75 44 92 58 31 104 48 46 62 113 1 20 27 29 16 77 25 10 120 19 71 68 98 24 28 101 21 73 7 124 74 36 72 85 14 78 83 61 26 30 105 57 103 3 8</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 106 21 124 88 34 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 38 104 1 12 98 72 23 42 20 55 59</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2422</div>	<div>objective function: -1415</div>

<div>80 35 39 20 125 36 102 9 122 1 77 62 40 48 8 69 118 55 34 121 73 96 89 24 105 23 66 88 4 78 15 81 103 72 63 64 76 99 109 123 108 12 43 41 11 32 74 57 58 85 82 54 86 10 87 46 124 5 7 79 50 30 91 21 27 120 97 117 110 106 101 75 6 65 13 116 25 84 49 56 3 33 95 98 67 14 17 22 29 51 38 60 42 90 37 19 52 16 93 61 59 45 68 71 113 18 47 28 107 70 83 2 115 53 112 111 44 92 114 26 31 94 104 119 100</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 31 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 5 25 7 37 67 63 106 21 124 88 38 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 34 104 1 12 98 72 23 42 20 55 59</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2522</div>	<div>objective function: -1415</div>

<div>71 10 78 13 63 45 114 70 23 75 106 36 15 72 26 108 66 97 22 116 109 43 29 104 51 91 93 20 57 2 33 34 11 88 84 14 21 38 103 81 5 122 37 1 64 54 30 105 41 76 50 79 110 119 18 125 53 95 52 86 59 35 44 65 87 16 9 8 17 98 48 68 69 99 80 96 111 90 62 24 6 27 77 73 123 49 82 25 94 58 39 100 85 107 101 74 113 46 124 56 61 7 121 117 67 55 112 102 28 4 120 92 19 83 3 89 31 60 115 47 42 32 118 40 12</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 106 21 124 88 38 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 34 104 1 12 98 72 23 42 20 55 59</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2522</div>	<div>objective function: -1415</div>

<div>21 59 30 110 112 100 98 73 114 78 85 101 99 55 107 69 2 123 7 18 19 88 75 49 46 35 87 3 68 54 16 86 62 96 70 72 74 39 37 23 64 56 28 36 5 11 125 22 8 82 31 32 53 34 116 89 33 122 41 48 13 93 111 26 38 66 121 25 24 12 50 118 63 15 20 95 115 113 104 27 117 1 108 14 6 109 81 119 91 10 42 40 77 9 92 43 58 79 65 80 17 71 61 57 83 29 105 60 67 47 103 120 4 94 97 51 76 90 124 106 44 84 102 45 52</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 106 21 124 88 38 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 34 104 1 12 98 72 23 42 20 55 59</div>
State Awal (Parent 6)	State Akhir (Anak 6)
<div>objective function: -2522</div>	<div>objective function: -1415</div>

<div>121 36 33 41 2 85 38 49 59 89 101 83 46 75 110 10 53 120 112 99 71 102 122 29 78 103 58 18 57 100 60 19 40 51 42 124 39 61 22 86 80 32 81 15 45 43 96 62 55 8 119 31 5 1 88 77 90 34 73 84 13 9 95 79 94 74 28 104 76 125 17 52 113 68 123 98 56 30 16 11 93 82 12 67 109 63 21 27 47 7 72 14 37 20 66 4 97 24 87 64 114 116 106 115 70 107 44 111 92 50 35 6 118 91 105 54 25 48 3 108 23 26 65 69 117</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 106 21 124 88 34 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 38 104 1 12 98 72 23 42 20 55 59</div>
State Awal (Parent 7)	State Akhir (Anak 7)
<div>objective function: -2521</div>	<div>objective function: -1315</div>

<div>101 82 76 37 81 11 13 38 73 19 91 33 86 14 77 29 40 22 24 69 70 75 112 35 5 16 74 89 43 45 114 109 118 78 99 47 51 53 87 10 88 98 65 26 28 6 30 27 59 110 125 44 95 61 66 90 80 60 39 21 108 113 119 4 41 92 58 15 63 105 36 124 64 121 103 116 122 8 54 115 17 111 48 71 123 32 79 83 120 7 31 94 93 52 49 67 97 46 2 20 100 117 57 107 50 104 1 106 55 68 102 34 18 84 62 56 25 12 96 42 23 9 72 3 85</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 64 21 124 101 38 75 19 103 2 87 61 17 106 73 88 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 34 12 1 104 98 72 23 42 20 55 59</div>
State Awal (Parent 8)	State Akhir (Anak 8)
<div>objective function: -2522</div>	<div>objective function: -1415</div>

<div>97 24 12 64 117 15 120 125 8 76 57 123 20 7 37 102 45 29 52 30 114 84 48 116 41 74 46 55 68 113 11 99 39 92 40 5 28 61 31 66 108 34 44 100 49 4 47 22 60 112 75 19 78 56 98 71 104 51 79 25 67 91 111 32 35 10 62 95 21 90 103 50 26 27 36 70 17 13 122 1 96 65 81 3 72 69 118 115 106 73 89 101 14 59 42 109 18 43 58 54 16 9 107 119 23 82 53 80 63 94 86 124 88 105 83 110 93 33 85 121 87 2 77 38 6</div>	<div>86 44 27 121 84 58 74 41 42 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 46 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 106 21 124 88 34 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 52 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 38 104 1 12 98 72 23 56 20 55 59</div>
State Awal (Parent 9)	State Akhir (Anak 9)
<div>objective function: -2522</div>	<div>objective function: -1415</div>

<div>49 85 50 80 79 107 118 121 31 20 86 59 4 17 114 109 55 11 5 120 123 52 9 111 112 78 43 90 41 35 56 25 29 91 2 101 21 14 15 36 88 104 26 105 95 62 18 92 82 106 117 46 68 48 97 32 54 1 19 81 122 61 100 44 37 58 93 70 33 77 87 16 69 51 65 124 39 10 71 98 89 103 63 113 12 72 53 30 119 38 75 57 40 47 6 34 27 42 23 84 8 108 7 13 22 73 94 45 110 60 83 116 24 3 125 102 115 64 66 74 99 67 28 76 96</div>	<div>86 44 27 121 84 58 74 41 56 110 16 119 5 22 33 96 93 81 92 89 26 82 40 113 62 115 122 120 9 66 54 123 52 78 116 118 76 69 109 8 90 36 35 28 30 11 4 6 117 47 99 107 91 65 60 51 39 31 25 7 37 67 63 106 21 124 88 38 75 19 103 2 87 61 17 64 73 101 95 85 15 70 57 53 3 46 80 105 50 77 18 13 114 112 100 94 29 97 43 102 14 45 83 125 79 48 49 71 24 32 10 111 108 68 34 104 1 12 98 72 23 42 20 55 59</div>
State Awal (Parent 10)	State Akhir (Anak 10)
<div>objective function: -2503</div>	<div>objective function: -1316</div>


```

71 92 101 41 24
76 60 77 85 125
49 16 56 50 3
80 87 79 2 112
48 67 95 43 97

55 74 58 103 57
120 114 89 18 6
13 96 23 69 68
44 37 121 118 15
34 1 123 100 111

14 73 115 9 36
25 84 21 39 93
52 53 104 98 62
94 19 72 86 27
42 40 59 66 32

113 5 124 20 88
61 116 38 108 81
90 7 75 65 64
10 26 31 117 105
106 28 35 4 78

30 91 47 99 107
110 33 17 119 45
83 63 22 29 12
82 8 54 46 70
102 122 109 11 51

```

```

86 44 27 121 84
58 74 41 56 110
16 119 5 22 33
96 93 81 92 89
26 82 40 113 62

115 122 120 9 66
54 123 46 78 116
118 76 69 109 8
90 36 35 28 30
11 4 6 117 47

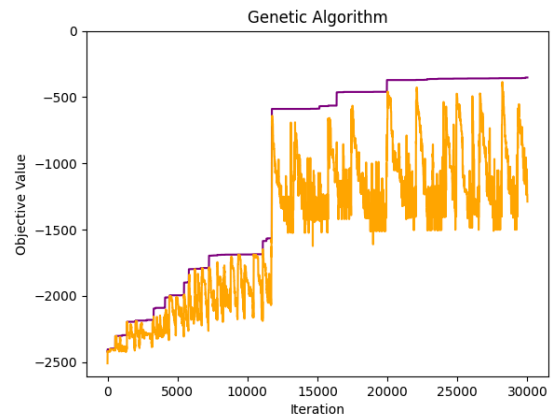
99 107 91 65 60
51 39 31 25 7
37 67 63 106 21
124 101 38 75 19
103 2 87 98 17

64 73 45 95 83
15 70 57 53 3
52 80 105 50 77
18 13 114 112 100
94 29 97 43 102

14 88 85 125 79
48 49 71 24 32
10 111 108 68 34
104 1 12 61 72
23 42 20 55 59

```

Plot Objective Value terhadap Iterasi



Percobaan 9 Iterasi = 300000 & Populasi = 10

Objective Value & Durasi

```

Max: 2461
Avg: 1546.9

```

```

Duration: 2312.688306570053

```

Max: -61

	Avg: -975.1 Banyak Iterasi = 300000 Jumlah Populasi = 10
State Awal (Parent 1)	State Akhir (Anak 1)
<div>objective function: -2522</div> <div> 81 114 78 93 28 80 83 61 122 91 12 31 47 10 54 102 89 38 73 94 119 85 6 115 21 19 68 75 39 8 4 72 109 42 22 86 70 36 2 123 62 13 55 77 69 108 44 96 101 106 111 5 26 79 14 60 9 110 20 52 56 88 74 120 105 71 49 32 112 125 65 51 11 27 29 23 82 43 117 35 15 7 17 116 103 34 41 48 121 18 40 90 58 53 63 95 87 113 57 76 67 97 118 25 16 59 107 124 104 45 37 1 92 66 3 33 46 98 64 30 100 99 84 50 24 </div>	<div>objective function: -1203</div> <div> 49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 110 46 125 55 50 66 112 39 37 119 105 92 75 11 58 95 97 </div>
State Awal (Parent 2)	State Akhir (Anak 2)
<div>objective function: -2521</div>	<div>objective function: -1203</div>

<div>64 41 23 57 124 5 111 106 44 6 30 27 83 86 125 100 108 35 42 29 119 12 98 84 16 91 3 1 13 104 99 85 87 112 19 67 58 46 50 15 120 36 94 109 96 74 103 39 90 22 92 66 21 110 48 52 121 107 76 79 56 59 25 88 122 18 118 60 114 123 75 31 101 33 116 45 47 69 9 4 17 71 115 89 24 97 20 78 43 51 40 26 8 93 61 14 80 34 49 11 28 65 73 117 53 95 55 38 77 81 62 68 82 2 70 37 32 7 72 10 63 113 105 54 102</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 110 46 125 55 50 66 112 39 37 119 105 92 75 11 58 95 97</div>
State Awal (Parent 3)	State Akhir (Anak 3)
<div>objective function: -2521</div>	<div>objective function: -1203</div>

<div>99 18 43 119 121 38 55 7 24 113 67 53 109 85 51 80 8 32 3 69 102 22 70 21 98 56 59 88 9 122 45 16 17 72 20 90 101 40 92 37 107 94 11 30 73 10 33 75 39 50 36 79 95 58 47 114 66 68 42 89 87 91 31 106 61 62 112 54 104 100 108 5 110 52 49 19 125 34 115 25 4 111 103 117 105 27 84 2 81 26 64 57 74 35 44 46 28 82 120 14 86 13 83 60 71 12 29 41 124 63 76 48 123 15 97 77 23 1 65 6 96 78 93 116 118</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 110 105 125 55 50 66 112 39 37 119 46 92 75 11 58 95 97</div>
State Awal (Parent 4)	State Akhir (Anak 4)
<div>objective function: -2521</div>	<div>objective function: -61</div>

<div>105 118 117 119 23 4 25 5 94 85 8 95 101 80 72 28 36 91 98 44 102 124 82 115 107 18 16 46 22 92 87 66 30 104 20 14 121 53 81 10 39 64 32 49 99 96 88 6 114 75 41 9 2 35 31 73 77 112 63 108 45 84 26 1 116 50 65 54 29 38 83 78 71 61 74 110 55 93 57 33 113 68 109 37 76 15 111 42 12 86 47 51 97 79 103 48 52 3 17 43 60 106 100 56 27 13 11 58 90 122 69 70 19 120 21 59 34 40 7 125 24 67 123 89 62</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 95 119 33 48 62 97 24 69 35 55 67 3 76 13 82 108 28 70 112 32 8 56 19 15 23 10 107 120 105 124 74 6 83 85 44 89 98 101 115 84 121 17 57 37 58 110 46 125 61 50 66 20 75 102 114 47 92 39 11 86 123 77</div>
State Awal (Parent 5)	State Akhir (Anak 5)
<div>objective function: -2521</div>	<div>objective function: -1203</div>

<div>62 29 70 21 124 119 31 22 102 14 74 46 89 72 56 50 83 30 5 11 19 111 97 106 67 101 78 79 51 38 118 41 81 52 90 28 32 123 108 3 114 75 95 43 103 57 104 109 6 85 76 64 117 105 82 20 112 40 2 86 87 47 9 49 88 18 10 98 42 100 65 92 94 93 58 25 71 99 17 125 110 44 113 8 35 39 77 24 26 66 91 60 115 121 23 84 80 55 122 1 37 116 68 73 15 59 45 61 69 7 53 54 48 120 12 36 13 34 33 63 16 27 4 96 107</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 110 46 125 55 50 66 112 39 37 119 105 92 75 11 58 95 97</div>
State Awal (Parent 6)	State Akhir (Anak 6)
<div>objective function: -2519</div>	<div>objective function: -1203</div>

<div>85 5 11 67 60 44 4 2 89 20 94 101 34 115 45 53 22 106 68 13 37 64 15 80 39 57 77 102 71 28 112 87 73 105 95 96 30 36 111 99 118 74 110 114 79 35 119 12 123 93 86 84 65 66 98 41 1 56 32 3 97 75 92 27 108 7 55 10 14 50 90 42 72 63 109 26 104 51 49 54 107 83 100 46 59 120 24 58 16 81 9 25 33 116 48 76 124 113 8 122 31 18 23 61 69 82 29 40 91 70 78 117 47 62 52 19 43 17 121 88 125 21 103 6 38</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 110 46 125 55 50 66 112 39 37 119 105 92 75 11 58 95 97</div>
State Awal (Parent 7)	State Akhir (Anak 7)
<div>objective function: -2522</div>	<div>objective function: -1204</div>

<div>92 15 94 86 75 55 62 87 107 29 38 2 108 106 24 9 7 44 46 19 23 41 115 122 67 80 111 16 65 28 3 74 12 79 37 58 17 6 31 64 100 95 33 96 81 82 72 14 30 103 117 91 53 121 39 10 68 77 70 114 59 22 69 85 104 88 83 110 99 60 47 43 116 27 25 71 66 36 63 40 20 84 124 18 45 4 78 102 118 5 1 61 123 101 32 56 90 8 11 50 73 21 42 57 120 105 119 51 113 48 26 54 13 109 49 89 97 76 52 93 35 112 34 125 98</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 11 121 69 62 110 46 125 55 50 66 112 39 37 119 105 92 75 101 58 95 97</div>
State Awal (Parent 8)	State Akhir (Anak 8)
<div>objective function: -2522</div>	<div>objective function: -1203</div>

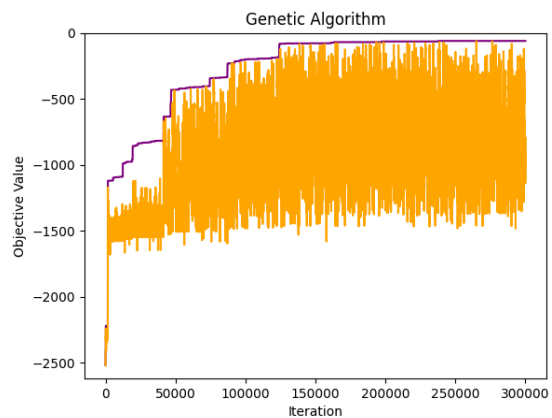
<div>122 79 93 17 113 9 39 55 11 59 66 30 6 124 107 41 92 12 81 67 76 73 69 60 43 91 121 26 56 74 77 96 27 89 72 19 63 57 51 44 3 36 13 103 62 111 70 123 112 102 80 15 32 68 104 114 53 65 117 108 42 71 82 95 1 61 24 29 34 49 20 110 98 120 118 94 83 100 101 7 10 16 90 58 115 119 23 4 14 33 48 52 45 37 99 88 54 125 87 105 40 97 8 46 21 5 38 28 2 106 85 75 64 84 31 78 116 35 25 22 47 50 86 109 18</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 110 46 125 55 50 66 112 39 37 119 105 92 75 11 58 95 97</div>
State Awal (Parent 9)	State Akhir (Anak 9)
<div>objective function: -2521</div>	<div>objective function: -1203</div>

<div>85 58 14 107 9 40 81 39 12 95 115 70 19 116 41 47 22 50 53 36 112 13 88 80 79 61 59 62 120 24 3 109 26 118 125 94 27 64 77 65 71 55 92 100 38 11 45 123 43 114 37 117 60 34 72 121 25 122 102 49 30 101 67 113 111 6 106 52 76 16 91 87 23 90 5 84 4 18 21 110 93 42 17 1 124 51 73 33 2 83 32 119 15 29 28 35 63 78 105 8 98 48 57 31 86 68 44 104 82 99 75 108 69 66 89 74 96 46 20 56 103 10 7 97 54</div>	<div>49 4 22 72 87 113 18 68 59 93 1 65 41 42 106 81 122 12 79 104 109 51 60 7 25 9 73 38 40 117 36 52 2 21 90 96 116 103 111 30 16 118 94 14 34 88 53 54 43 5 80 91 71 26 100 27 29 64 78 99 45 31 63 123 114 17 47 33 57 15 23 35 6 67 3 76 13 20 108 28 70 82 32 86 56 98 24 85 10 107 120 124 77 74 19 83 8 44 89 61 102 115 84 48 101 121 69 62 66 46 125 55 50 110 112 39 37 119 105 92 75 11 58 95 97</div>
State Awal (Parent 10)	State Akhir (Anak 10)
<div>objective function: -2521</div>	<div>objective function: -1203</div>

10	123	98	49	15
53	100	6	1	111
31	76	57	19	8
5	48	50	33	77
51	104	41	102	21
29	118	20	114	16
54	37	46	69	30
63	110	58	86	105
18	13	62	90	113
68	117	55	2	119
88	42	28	99	12
120	85	7	106	27
45	122	71	89	81
101	47	4	108	109
67	107	73	52	91
78	116	83	9	97
124	22	38	84	80
93	56	103	96	26
60	35	92	32	14
87	65	34	94	115
79	11	40	25	125
23	43	61	70	75
121	36	74	39	59
17	72	3	82	24
64	112	44	66	95

49	4	22	72	87
113	18	68	59	93
1	65	41	42	106
81	122	12	79	104
109	51	60	7	25
9	73	38	40	117
36	52	2	21	90
96	116	103	111	30
16	118	94	14	34
88	53	54	43	5
80	91	71	26	100
27	29	64	78	99
45	31	63	123	114
17	47	33	57	15
23	35	6	67	3
76	13	20	108	28
70	82	32	86	56
98	24	85	10	107
120	124	77	74	19
83	8	44	89	61
102	115	84	48	101
121	69	62	110	46
125	55	50	66	112
39	37	119	105	92
75	11	58	95	97

Plot Objective Value terhadap Iterasi



2.4 Hasil Analisis

Berdasarkan eksperimen yang dilakukan terhadap setiap algoritma pencarian lokal pada bagian sebelumnya, berikut ini merupakan tabel yang menyimpulkan hasil eksperimen tiap algoritma dengan Objective Value terbaik yang diambil,

Algoritma	Objective Value Terbaik	Durasi Rata-rata(Durasi Eksperimen) (Detik)
Steepest Ascent Hill Climbing	-71	7.01 (6.72, 7.56, 6.76)
Hill Climbing with Sideways Move	-58	32.12 (37.02, 29.19, 30.15)
Hill Climbing with Random Restart	-53	30.51 (28.92, 29.06, 33.56) (Tergantung banyaknya restart)
Stochastic Hill Climbing	-60	62.01 (51.96, 65.96, 68,12)
Simulated Annealing	-45	20.24 (19.83, 20.74, 20,17)
Genetic Algorithm	-56	16 sampai 2500 (20.05, 200.84, 1887.25, 16.28, 230.06, 2498.56, 29.5, 302.85, 2312.68) (Tergantung banyaknya iterasi)

Berdasarkan hasil eksperimen, terlihat bahwa algoritma Simulated Annealing memiliki objective value yang paling mendekati 0. Hal ini dikarenakan algoritma ini memungkinkan perpindahan ke neighbor yang objective valuenya lebih kecil. Algoritma Simulated Annealing dapat keluar dari jebakan lokal optima sehingga memperbesar ruang eksplorasi untuk mendapatkan solusi yang lebih baik. Pemilihan neighbor pada algoritma ini juga dilakukan secara acak sehingga bisa lebih cepat.

Algoritma Hill-Climbing with Random Restart mendapat nilai objective value terbaik kedua. Algoritma ini menggunakan algoritma Steepest Ascent Hill Climbing, dengan memberikan parameter random restart, yaitu memulai pencarian dari beberapa titik awal yang berbeda secara acak. Semakin banyak jumlah restart, kemungkinan mendapatkan solusi yang lebih optimal juga semakin baik. Namun di sisi lain, algoritma ini sangat bergantung pada random initial state-nya.

Genetic Algorithm mendapat nilai objective value terbaik ketiga. Ini bisa didapat karena hasil objective value Genetic Algorithm sangat dipengaruhi oleh iterasinya. Berdasarkan eksperimen, 3000 iterasi mendapatkan nilai objective terbaik yaitu -1306 dan 30000 iterasi mendapatkan nilai objective terbaik yaitu -252. Genetic Algorithm ini juga sangat lama untuk mendapatkan hasil objective value yang baik karena proses crossover dan mutasi dilakukan secara acak.

Algoritma Hill-Climbing with Sideways Move mendapat nilai objective terbaik ke-empat. Algoritma ini mirip dengan Steepest Ascent Hill Climbing, namun dengan sedikit modifikasi, yaitu memperbolehkan memilih neighbor yang mempunyai objective value yang sama. Langkah tersebut juga akan dibatasi sehingga saat sudah mencapai batas, pencarian tetap akan berhenti.

Stochastic Hill-Climbing mendapatkan nilai objective terbaik ke lima. Algoritma ini tidak konsisten karena menggunakan perpindahan neighbor ke nilai objective yang lebih baik secara acak. Hal ini membuat algoritma Stochastic sangat mungkin untuk terjebak di local optima. Lalu berdasarkan eksperimen, Algoritma ini memiliki durasi eksekusi kedua paling lama dikarenakan jumlah iterasi yang dilakukan banyak, sehingga algoritma ini sangat bergantung kepada jumlah iterasi untuk nilai objective dan durasi yang dihasilkan,

Steepest Ascent Hill-Climbing memiliki nilai objective terburuk tetapi memiliki nilai eksekusi paling cepat dibanding algoritma lainnya. Algoritma ini memilih nilai objective paling baik setiap iterasinya yang membuat Steepest Ascent lebih efisien dalam waktu eksekusi. Tetapi hal ini yang membuat Steepest Ascent mudah terjebak di local optima karena tidak ada cara untuk keluar dari local optima.

BAB III

Kesimpulan dan Saran

Berdasarkan eksperimen dan analisis yang dilakukan, algoritma Simulated Annealing memiliki hasil nilai objective dan waktu eksekusi yang terbaik dibanding algoritma lainnya. Algoritma ini dapat menemukan solusi yang mendekati 0 dalam waktu yang singkat, hanya lebih lama dibanding Steepest Ascent. Algoritma Genetic Algorithm memiliki hasil nilai objective dan waktu terburuk dibanding algoritma lainnya. Hal ini dikarenakan crossover dan mutasi yang dilakukan secara acak. Genetic Algorithm perlu iterasi yang banyak untuk memberikan nilai objective yang baik sehingga membutuhkan waktu yang banyak juga tetapi nilai objective yang diberikan tidak sepadan dengan waktu yang digunakan untuk eksekusi algoritma.

BAB IV
Pembagian Tugas

Nama	NIM	Tugas
Muhammad Rifa Ansyari	18222004	Membuat <i>source code</i> , Membantu membuat plot
Justin Lawrance	18222006	Membuat <i>source code</i> , Mengerjakan dokumen
Axelius Davin	18222016	Membantu <i>source code</i> Sideways Move, Membuat plot, Mengerjakan dokumen
Natanael Steven S	18222054	Membantu <i>source code</i> Simulated Annealing, Membuat plot, Mengerjakan dokumen

BAB V

Referensi

<https://www.trump.de/magic-squares/magic-cubes/cubes-1.html>

<https://www.obitko.com/tutorials/genetic-algorithms/encoding.php>

<https://mathworld.wolfram.com/news/2003-11-18/magiccube/>

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.

<https://stackoverflow.com/questions/41057618/genetic-algorithm-with-binary-feasibility>

<https://stackoverflow.com/questions/37285603/sorting-by-genetic-algorithm-duplicate-data-in-cross-over>

<https://www.trump.de/magic-squares/magic-cubes/cubes-1.html>

<https://www.magischvierkant.com/three-dimensional-eng/magic-features/>

<https://towardsdatascience.com/rubiks-cube-solver-96fa6c56fbe4>

<https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/>

<https://mathworld.wolfram.com/news/2003-11-18/magiccube/>