

Tugas Besar 1 IF3070 Dasar Inteligensi Artifisial & IF3170

Inteligensi Artifisial

Pencarian Solusi Diagonal Magic Cube dengan Local Search



Kelompok 15

- | | | |
|----|------------------------|----------|
| 1. | Ariel Herfrison | 13522002 |
| 2. | Andhika Tantyo Anugrah | 13522094 |
| 3. | Kristo Anugrah | 13522024 |
| 4. | Chelvadinda | 13522154 |

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

2024/2025

DAFTAR ISI

DAFTAR ISI	2
DESKRIPSI PERSOALAN	3
PEMBAHASAN	4
2.1 Pemilihan objective function	4
2.2 Implementasi algoritma	5
2.2.1 Objective Function	5
2.2.2 Steepest Ascent Hill-Climbing	9
2.2.3 Simulated Annealing	12
2.2.4 Genetic Algorithm	15
2.2.5 Hill-Climbing with Sideways Move	25
2.3 Hasil Eksperimen dan Analisis	27
2.2.1 Steepest Ascent Hill-Climbing	27
• Eksperimen Jumlah Iterasi	27
2.2.2 Simulated Annealing	32
• Eksperimen Minimum Temperature	32
• Eksperimen Cooling Rate	40
2.2.3 Genetic Algorithm	48
• Eksperimen Jumlah Populasi	48
• Eksperimen Jumlah Iterasi	114
2.2.4 Hill-Climbing with Sideways Move	194
• Eksperimen Jumlah Iterasi	194
2.2.5 Analisis Hasil Eksperimen	198
KESIMPULAN DAN SARAN	204
3.1 Kesimpulan	204
3.2 Saran	204
PEMBAGIAN TUGAS	205
REFERENSI	206

DESKRIPSI PERSOALAN

Tugas besar ini bertujuan untuk menyelesaikan permasalahan Diagonal Magic Cube berukuran 5x5x5. Initial state dari suatu kubus adalah susunan angka 1 hingga 5^3 secara acak. Kemudian, tiap iterasi pada algoritma local search, langkah yang boleh dilakukan adalah menukar posisi dari 2 angka pada kubus tersebut (2 angka yang ditukar tidak harus bersebelahan). Khusus untuk genetic algorithm, boleh dilakukan penukaran posisi lebih dari 2 angka sekaligus dalam satu iterasi (tetapi hanya menukar posisi 2 angka saja juga diperbolehkan).

Diagonal magic cube merupakan kubus yang tersusun dari angka 1 hingga n^3 tanpa pengulangan dengan n adalah panjang sisi pada kubus tersebut. Angka-angka pada tersusun sedemikian rupa sehingga properti-properti berikut terpenuhi:

- Terdapat satu angka yang merupakan magic number dari kubus tersebut (Magic number tidak harus termasuk dalam rentang 1 hingga n^3 , magic number juga bukan termasuk ke dalam angka yang harus dimasukkan ke dalam kubus)
- Jumlah angka-angka untuk setiap baris sama dengan magic number
- Jumlah angka-angka untuk setiap kolom sama dengan magic number
- Jumlah angka-angka untuk setiap tiang sama dengan magic number
- Jumlah angka-angka untuk seluruh diagonal ruang pada kubus sama dengan magic number
- Jumlah angka-angka untuk seluruh diagonal pada suatu potongan bidang dari kubus sama dengan magic number

Dalam menyelesaikan permasalahan ini, kami akan mengimplementasikan empat algoritma local search:

1. Steepest Ascent Hill-Climbing
2. Simulated Annealing
3. Genetic Algorithm
4. Hill-Climbing with Sideways Move

Setiap algoritma akan dieksekusi sebanyak tiga kali untuk mencatat sejumlah aspek, termasuk state awal dan akhir kubus, nilai akhir dari objective function, plot nilai objective function terhadap jumlah iterasi yang dilewati, dan durasi proses pencarian.

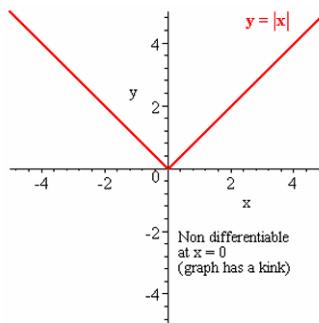
PEMBAHASAN

2.1 Pemilihan objective function

Perhatikan bahwa untuk diagonal magic cube $5 \times 5 \times 5$, jumlah angka di setiap baris, kolom, tiang, diagonal ruang serta diagonal bidang harus sama dengan 315. Hal ini dapat dihitung dengan kalkulasi sederhana. Ada 25 baris pada sebuah magic cube dengan jumlah yang sama, dengan jumlah total magic cube sama dengan 7875 (didapat dari $125 \times 126 / 2$). Jadi, setiap baris memiliki jumlah $7875 / 25 = 315$. Ini artinya, untuk kubus $5 \times 5 \times 5$, magic number kubus tersebut sama dengan 315. Dari sebuah state, kita dapat menghitung array 1 dimensi dengan panjang 109 yang merepresentasikan jumlah cube pada setiap baris, kolom, tiang, diagonal ruang, serta diagonal bidang. Sebuah solusi dari persoalan diagonal magic cube adalah sebuah state array sedemikian sehingga setiap elemen pada array jumlah sama dengan 315. Berbekal dari pengetahuan ini, dapat didefinisikan sebuah objective function sederhana sebagai berikut:

$$F(\text{sum}) = \sum_{i=1}^{109} (\text{sum}[i] - 315)$$

atau dengan kata lain nilai objective function dari sebuah array sum dengan panjang 109 sama dengan jumlah perbedaan absolut setiap elemennya dengan 315. Kelemahan dari objective function ini adalah penggunaan fungsi absolut yang non differentiable. Penggunaan fungsi non differentiable sebaiknya dihindari, karena ketika melakukan local search, fungsi yang non differentiable memiliki titik dimana gradien fungsi tidak terdefinisi.



Gambar 2.1.1 Fungsi absolut tidak terdiferensiasi

Selain itu, fungsi tersebut juga memiliki kelemahan dalam hal pembagian reward. Tinjau dua buah array [315, 315, ..., 315, 331] dan [315, 315, ..., 307, 323]. Kedua array tersebut memiliki nilai objective function yang sama, yaitu 16. Meskipun begitu, menurut kami array kedua lebih baik dibandingkan dengan array pertama. Kami menilai array dengan distribusi angka yang merata lebih baik dibanding array dengan sebuah elemen yang berbeda jauh dengan magic number.

Dengan evaluasi tersebut, digunakan sebuah fungsi objektif yang kontinu dan lebih berfokus pada distribusi perbedaan yang merata. Fungsi tersebut adalah:

$$F(sum) = \sum_{i=1}^{109} (sum[i] - 315)^6$$

Fungsi tersebut merepresentasikan jumlah perbedaan setiap elemen array sum dengan 315 yang dipangkatkan 6. Dengan fungsi objektif ini, algoritma memiliki insentif untuk membuat perbedaan dalam array sum merata. Hal ini karena jarak antar nilai perbedaan sangatlah besar (perbedaan sebanyak 3 (jauh dengan perbedaan sebanyak 2 ($2^6 = 64$ $36 = 729$) memiliki perbedaan)). Selain itu, fungsi tersebut terdiferensiasi di seluruh titik, sehingga gradien fungsi terdefinisi di seluruh titik. Dengan fungsi objektif tersebut, ingin dicari sebuah minimum global yang bernilai 0. Hal ini karena $F(sum)$ yang bernilai 0 berarti seluruh elemen sum sama dengan konstanta magic number.

2.2 Implementasi algoritma

2.2.1 Objective Function

Fungsi/kelas	Deskripsi	Source Code
unitDist	Fungsi unitDist menghitung jarak antara jumlah angka pada satu baris, kolom, atau diagonal dengan magic number c. Semakin besar perbedaan antara jumlah angka dan magic number, semakin besar nilai jarak yang dihasilkan.	<pre>long long unitDist(long long x, long long c, int srs = 0){ return (x - c) * (x - c) * (x - c) * (x - c) * (x - c) * (x - c); }</pre>
distanceMat	Fungsi distanceMat menghitung penalti untuk setiap baris dan	<pre>long long distance(vector<int> &arr, int srs = 0){</pre>

	<p>kolom dalam satu bidang dari kubus, memanggil unitDist untuk menilai seberapa jauh hasil penjumlahan setiap baris dan kolom dari nilai c.</p>	<pre>vector<int> sumPillar(n * n, 0); long long ans = 0; for(int i = 0; i < arr.size(); i += n * n){ vector<int> temp; for(int j = i; j < i + n * n; j++){ sumPillar[j % (n * n)] += arr[j]; temp.push_back(arr[j]); } ans += distanceMat(temp, srs); } for(int i = 0; i < n * n; i++){ ans += unitDist(sumPillar[i], c, srs); } // Penambahan penalti untuk diagonal return ans; }</pre>
distanceVec	<p>Fungsi ini menghasilkan vektor penalti untuk berbagai komponen dalam kubus, termasuk baris, kolom, tiang, dan diagonal pada setiap potongan bidang serta ruang kubus.</p>	<pre>vector<int> distanceVec(vector<int> arr){ vector<int> sumPillar(n * n, 0); vector<int> ans; for(int i = 0; i < (int)arr.size(); i += n * n){ vector<int> temp; vector<int> sumHor(n, 0); for(int j = i; j < i + n * n; j++){ sumPillar[j % (n * n)] += arr[j]; sumHor[(j / n) % n] += arr[j]; } ans.insert(ans.end(), sumHor.begin(), sumHor.end()); } for(int i = 0; i < (int)arr.size(); i += n * n){ vector<int> temp; vector<int> sumVer(n, 0); for(int j = i; j < i + n * n; j++){ sumVer[j % n] += arr[j]; } ans.insert(ans.end(), sumVer.begin(), sumVer.end()); } ans.insert(ans.end(), sumPillar.begin(), sumPillar.end()); for(int k = 0; k < n; k++){ int i1 = 0, j1 = 0; int i2 = 0, j2 = n - 1; // D1 int st1 = 0, st2 = 0;</pre>

		<pre> for(int u = 0; u < n; u++){ st1 += arr[(k * n * n + i1 * n + j1)]; st2 += arr[(k * n * n + i2 * n + j2)]; i1++; i2++; j1++; j2--; } ans.push_back(st1); ans.push_back(st2); } for(int k = 0; k < n; k++){ int i1 = 0, j1 = 0; int i2 = 0, j2 = n - 1; // D2 int st3 = 0, st4 = 0; for(int u = 0; u < n; u++){ st3 += arr[(i1 * n * n + k * n + j1)]; st4 += arr[(i2 * n * n + k * n + j2)]; i1++; i2++; j1++; j2--; } ans.push_back(st3); ans.push_back(st4); } int r1 = 0, c1 = 0; int r2 = 0, c2 = n - 1; int r3 = n - 1, c3 = 0; int r4 = n - 1, c4 = n - 1; int s1 = 0, s2 = 0, s3 = 0, s4 = 0; for(int i = 0; i < n; i++){ s1 += arr[(r1 * n + c1) + i * n * n]; s2 += arr[(r2 * n + c2) + i * n * n]; s3 += arr[(r3 * n + c3) + i * n * n]; s4 += arr[(r4 * n + c4) + i * n * n]; r1++; c1++; r2++; c2--; r3--; c3++; r4--; c4--; } ans.push_back(s1); ans.push_back(s2); ans.push_back(s3); ans.push_back(s4); return ans; } </pre>
distance	Fungsi distance adalah objective	long long distance (vector<int>&arr){

	<p>function utama yang mengembalikan total penalti untuk kubus dalam kondisi saat ini. Ini menggunakan unitDist pada semua baris, kolom, tiang, dan diagonal untuk menghitung seberapa jauh kubus dari kondisi ideal.</p>	<pre> int cnt = 0; vector<int> sumPillar(n * n, 0); long long ans = 0; for(int i = 0; i < (int)arr.size(); i += n * n){ vector<int> temp; for(int j = i; j < i + n * n; j++){ sumPillar[j % (n * n)] += arr[j]; temp.push_back(arr[j]); } ans += distanceMat(temp); } for(int i = 0; i < n * n; i++){ ans += unitDist(sumPillar[i], c); } int r1 = 0, c1 = 0; int r2 = 0, c2 = n - 1; int r3 = n - 1, c3 = 0; int r4 = n - 1, c4 = n - 1; int s1 = 0, s2 = 0, s3 = 0, s4 = 0; for(int i = 0; i < n; i++){ s1 += arr[(r1 * n + c1) + i * n]; s2 += arr[(r2 * n + c2) + i * n]; s3 += arr[(r3 * n + c3) + i * n]; s4 += arr[(r4 * n + c4) + i * n]; r1++; c1++; r2++; c2--; r3--; c3++; r4--; c4--; } ans += unitDist(s1, c); ans += unitDist(s2, c); ans += unitDist(s3, c); ans += unitDist(s4, c); for(int k = 0; k < n; k++){ int i1 = 0, j1 = 0; int i2 = 0, j2 = n - 1; // D1 int st1 = 0, st2 = 0; // D2 int st3 = 0, st4 = 0; // D3 int st5 = 0, st6 = 0; for(int u = 0; u < n; u++){ st1 += arr[(k * n * n + i1 * n + j1)]; st2 += arr[(k * n * n + i2 * n + j2)]; st3 += arr[(i1 * n * n + k * n } } </pre>
--	---	---

		<pre> + j1]); st4 += arr[(i2 * n * n + k * n + j2)]; st5 += arr[(j1 * n * n + i1 * n + k)]; st6 += arr[(j2 * n * n + i2 * n + k)]; i1++; i2++; j1++; j2--; } ans += unitDist(st1, c); ans += unitDist(st2, c); ans += unitDist(st3, c); ans += unitDist(st4, c); //ans += unitDist(st5, c); //ans += unitDist(st6, c); } return ans; } </pre>
--	--	---

2.2.2 Steepest Ascent Hill-Climbing

Fungsi/kelas	Deskripsi	Source Code
makeRandomCube (fungsi yang teradapat pada Genetic Algorithm)	<p>Input: ukuran kubus</p> <p>Output: kubus (dalam bentuk vector<int>)</p> <p>Fungsi ini menghasilkan konfigurasi kubus dengan angka unik yang telah diacak</p>	<pre> vector<int> makeRandomCube(int n) { random_device rd; mt19937 rng(rd()); vector<int> cube; for(int i = 1; i <= n * n * n; i++){ cube.push_back(i); } shuffle(begin(cube), std::end(cube), rng); for(int i = 0; i < cube.size(); i++){ if(cube[i] == 63){ swap(cube[i], cube[62]); break; } } return cube; } </pre>
main	Fungsi main dalam Steepest Ascent Climbing-Hill menerima input banyak populasi, lalu	<pre> int32_t main(){ //evalAnswer(); cout << "Ukuran Kubus: "; } </pre>

	<p>membangkitkan magic cube dengan elemen acak untuk melakukan pencarian solusinya dengan Steepest Ascent Climbing-Hill sampai algoritma ini tidak dapat menemukan neighbor yang nilainya lebih tinggi dari current neighbor.</p>	<pre> cin >> n; auto beg = high_resolution_clock::now(); c = n * (n * n * n + 1); c /= 2; vector<int> cube = makeRandomCube(n); vector<int> startingCube = cube; sumVec = distanceVec(cube); vector<int> ans; long long curDist = distance(cube); long long difference; int pos1, pos2; int steps = 0; int maxFail = 2e6; int minToRestart = 64; long long mini = LLONG_MAX; bool decreasingVal = true; ofstream logs("hill-climbing_log.txt"); while(decreasingVal) { decreasingVal = false; for(int pos1 = 0; pos1 < cube.size(); pos1++) { for(int pos2 = pos1 + 1; pos2 < cube.size(); pos2++) { swap(cube[pos1], cube[pos2]); long long newDist = distance(cube); if(newDist < curDist) { curDist = newDist; if(curDist < mini) { ans = cube; mini = curDist; } } } } } decreasingVal = true; </pre>
--	---	--

```

        } else {
            swap(cube[pos1], cube[pos2]);
        }
    }
    steps++;
    cout << curDist << " " <<
steps << endl;
    logs << curDist << " " <<
steps << endl;
}
logs.close();

ofstream
sc("start_cube.txt");
ofstream
fc("final_cube.txt");

for(auto &x:
startingCube){
    sc << x << " ";
}
sc << endl;
sc.close();

for(auto &x: ans){
    fc << x << " ";
}
fc << endl;
fc.close();

cout << endl;
cout << "Final objective
function value: " << mini << endl;
auto en = high_resolution_clock::now();
auto dur = duration_cast<microseconds>(en -
beg);
cout << "Duration (in
microsec): " << dur.count() << endl;

system("python3
plot_cube.py start_cube.txt
\"Starting Cube\" && python3
plot_cube.py final_cube.txt \"Final
Cube\"");
}

//** Sesuaikan dengan
sistem anda! */
system("python3
plot_annealing.py
hill-climbing_log.txt I \"Objective
function vs steps\"");

```

		return 0; }
--	--	----------------

2.2.3 Simulated Annealing

Fungsi/kelas	Deskripsi	Source Code
distanceEff	Fungsi distanceEff digunakan untuk menghitung perubahan nilai objective function secara efisien saat swap dua elemen	<pre> long long distanceEff(int pos1, int pos2, long long curDist, vector<int> &cube, vector<int> &sumVec) { int i1, j1, k1; int i2, j2, k2; k1 = pos1 / (n * n); k2 = pos2 / (n * n); i1 = (pos1 / n) % n; i2 = (pos2 / n) % n; j1 = pos1 % n; j2 = pos2 % n; long long dE = curDist; long long bef1 = cube[k1 * n * n + i1 * n + j1]; long long bef2 = cube[k2 * n * n + i2 * n + j2]; dE -= unitDist(sumVec[k1 * 5 + i1], c); dE += unitDist(sumVec[k1 * 5 + i1] - bef1 + bef2, c); sumVec[k1 * 5 + i1] += bef2 - bef1; dE -= unitDist(sumVec[25 + k1 * 5 + j1], c); dE += unitDist(sumVec[25 + k1 * 5 + j1] - bef1 + bef2, c); sumVec[25 + k1 * 5 + j1] += bef2 - bef1; dE -= unitDist(sumVec[50 + i1 * 5 + j1], c); dE += unitDist(sumVec[50 + i1 * 5 + j1] - bef1 + bef2, c); sumVec[50 + i1 * 5 + j1] += bef2 - bef1; dE -= unitDist(sumVec[k2 * 5 + i2], c); dE += unitDist(sumVec[k2 * 5 + i2] - bef2 + bef1, c); sumVec[k2 * 5 + i2] += bef1 - </pre>

```

bef2;
    dE -= unitDist(sumVec[25 + k2 *
5 + j2], c);
    dE += unitDist(sumVec[25 + k2 *
5 + j2] - bef2 + bef1, c);
    sumVec[25 + k2 * 5 + j2] += bef1 -
bef2;
    dE -= unitDist(sumVec[50 + i2 *
5 + j2], c);
    dE += unitDist(sumVec[50 + i2 *
5 + j2] - bef2 + bef1, c);
    sumVec[50 + i2 * 5 + j2] += bef1 -
bef2;

    // D1
    if(i1 == j1){
        dE -= unitDist(sumVec[75 + 2 *
k1], c);
        dE += unitDist(sumVec[75 + 2 *
k1] - bef1 + bef2, c);
        sumVec[75 + 2 * k1] += bef2 -
bef1;
    }
    if(i1 + j1 == n - 1){
        dE -= unitDist(sumVec[75 + 2 *
k1 + 1], c);
        dE += unitDist(sumVec[75 + 2 *
k1 + 1] - bef1 + bef2, c);
        sumVec[75 + 2 * k1 + 1] += bef2 -
bef1;
    }
    // D2
    if(j1 == k1){
        dE -= unitDist(sumVec[85 + 2 *
i1], c);
        dE += unitDist(sumVec[85 + 2 *
i1] - bef1 + bef2, c);
        sumVec[85 + 2 * i1] += bef2 -
bef1;
    }
    if(j1 + k1 == n - 1){
        dE -= unitDist(sumVec[85 + 2 *
i1 + 1], c);
        dE += unitDist(sumVec[85 + 2 *
i1 + 1] - bef1 + bef2, c);
        sumVec[85 + 2 * i1 + 1] += bef2 -
bef1;
    }

    // D1
    if(i2 == j2){
        dE -= unitDist(sumVec[75 + 2 *
k2], c);
        dE += unitDist(sumVec[75 + 2 *
k2] - bef2 + bef1, c);
    }

```

```

        sumVec[75 + 2 * k2] += bef1 -
bef2;
    }
    if(i2 + j2 == n - 1){
        dE -= unitDist(sumVec[75 + 2
* k2 + 1], c);
        dE += unitDist(sumVec[75 + 2
* k2 + 1] - bef2 + bef1, c);
        sumVec[75 + 2 * k2 + 1] +=
bef1 - bef2;
    }
    // D2
    if(j2 == k2){
        dE -= unitDist(sumVec[85 + 2
* i2], c);
        dE += unitDist(sumVec[85 + 2
* i2] - bef2 + bef1, c);
        sumVec[85 + 2 * i2] += bef1 -
bef2;
    }
    if(j2 + k2 == n - 1){
        dE -= unitDist(sumVec[85 + 2
* i2 + 1], c);
        dE += unitDist(sumVec[85 + 2
* i2 + 1] - bef2 + bef1, c);
        sumVec[85 + 2 * i2 + 1] +=
bef1 - bef2;
    }

    // DR1
    if(i1 == j1 && j1 == k1){
        dE -= unitDist(sumVec[95], c);
        dE += unitDist(sumVec[95] -
bef1 + bef2, c);
        sumVec[95] += bef2 - bef1;
    }
    // DR2
    if(i1 == k1 && i1 + j1 == n - 1){
        dE -= unitDist(sumVec[96], c);
        dE += unitDist(sumVec[96] -
bef1 + bef2, c);
        sumVec[96] += bef2 - bef1;
    }
    // DR3
    if(j1 == k1 && i1 + j1 == n - 1){
        dE -= unitDist(sumVec[97], c);
        dE += unitDist(sumVec[97] -
bef1 + bef2, c);
        sumVec[97] += bef2 - bef1;
    }
    // DR4
    if(i1 == j1 && i1 + k1 == n - 1){
        dE -= unitDist(sumVec[98], c);
        dE += unitDist(sumVec[98] -
bef1 + bef2, c);
    }

```

		<pre> sumVec[98] += bef2 - bef1; } // DR1 if(i2 == j2 && j2 == k2){ dE -= unitDist(sumVec[95], c); dE += unitDist(sumVec[95] - bef2 + bef1, c); sumVec[95] += bef1 - bef2; } // DR2 if(i2 == k2 && i2 + j2 == n - 1){ dE -= unitDist(sumVec[96], c); dE += unitDist(sumVec[96] - bef2 + bef1, c); sumVec[96] += bef1 - bef2; } // DR3 if(j2 == k2 && i2 + j2 == n - 1){ dE -= unitDist(sumVec[97], c); dE += unitDist(sumVec[97] - bef2 + bef1, c); sumVec[97] += bef1 - bef2; } // DR4 if(i2 == j2 && i2 + k2 == n - 1){ dE -= unitDist(sumVec[98], c); dE += unitDist(sumVec[98] - bef2 + bef1, c); sumVec[98] += bef1 - bef2; } return dE; } </pre>
getProbability	Fungsi ini menghitung probabilitas untuk menerima solusi yang lebih buruk dalam algoritma Simulated Annealing. Ini memungkinkan solusi yang lebih buruk untuk diterima dengan probabilitas tertentu	<pre> double getProbability(long long difference, double temperature){ return exp(-1 * difference / temperature); } </pre>

2.2.4 Genetic Algorithm

Fungsi/kelas	Deskripsi	Source Code
findOptimum	Input: populasi Output: long long	long long findOptimum (vector<vector<int>>

	Fungsi ini mencari dan mengembalikan nilai objective function paling optimal dalam populasi	<pre>population) { long long optimum = distance(population[0]); for (int i = 1; i < population.size(); i++) { long long fitness = distance(population[i]); if (fitness < optimum) { optimum = fitness; } } return optimum; }</pre>
findAverage	<p>Input: populasi Output: long long</p> <p>Fungsi ini mencari dan mengembalikan rata-rata nilai objective function dalam populasi</p>	<pre>long long findAverage(vector<vector<int>> population) { long long average = 0; for (int i = 0; i < population.size(); i++) { long long fitness = distance(population[i]); average += fitness; } return average/population.size(); }</pre>
makeRandomCube	<p>Input: ukuran kubus Output: kubus (dalam bentuk vector<int>)</p> <p>Fungsi ini menghasilkan konfigurasi kubus dengan angka unik yang telah diacak</p>	<pre>vector<int> makeRandomCube(int n) { random_device rd; mt19937 rng(rd()); vector<int> cube; for(int i = 1; i <= n * n * n; i++){ cube.push_back(i); } shuffle(begin(cube), std::end(cube), rng); for(int i = 0; i < cube.size(); i++){ if(cube[i] == 63){ swap(cube[i], cube[62]); break; } } return cube; }</pre>
makePopulation	Input: ukuran populasi, ukuran	<pre>vector<vector<int>> makePopulation(int nPop, int n) {</pre>

	<p>kubus</p> <p>Output: populasi berisi kubus-kubus</p> <p>Fungsi ini menghasilkan populasi berisi kubus dalam bentuk <code>vector<vector<int>></code>.</p>	<pre>vector<vector<int>> population; for (int i = 0; i < nPop; i++) { vector<int> cube = makeRandomCube(n); population.push_back(cube); } return population; }</pre>
selection	<p>Input: populasi</p> <p>Output: populasi baru hasil seleksi</p> <p>Fungsi ini melakukan <i>selection</i> dengan metode <i>roulette wheel</i>. Karena kubus/individu memiliki <i>fitness</i> yang lebih baik jika nilai <i>objective-function</i>-nya lebih kecil, <i>fitness</i> merupakan selisih nilai <i>objective function</i> dengan total dari semua <i>objective function</i>. Setiap kubus/individu diberi porsi dalam <i>roulette wheel</i> dengan porsi lebih besar diberikan kepada kubus dengan <i>fitness</i> yang lebih besar (<i>objective function</i> lebih kecil). Porsi dalam <i>roulette wheel</i> ditandai dengan <i>boundary</i> yang menandai batas porsi setiap kubus. Fungsi akan memutar <i>roulette wheel</i> dengan cara memilih angka bulat random di antara 0 dan 100 sebanyak <i>n</i> kali dengan <i>n</i> merupakan jumlah populasi. Apabila angka tersebut jatuh dibawah porsi/<i>boundary</i> suatu kubus/individu, maka kubus/individu tersebut akan dipilih.</p>	<pre>vector<vector<int>> selection(vector<vector<int>> population) { // sum objective function double totalObjFunc = 0; vector<double> objFuncArr; for (int i = 0; i < population.size(); i++) { vector<int> cube = population[i]; double objFunc = distance(cube); totalObjFunc += objFunc; objFuncArr.push_back(objFunc); } // find fitness double totalFitness = 0; vector<double> fitnessArr; for (int i = 0; i < population.size(); i++) { double fitness = totalFitness - objFuncArr[i]; totalFitness += fitness; fitnessArr.push_back(fitness); } // create boundaries for roulette // wheel vector<double> boundaries; for (int i = 0; i < population.size(); i++) { double fitness = fitnessArr[i]; double portion = fitness / totalFitness; if (i == 0) { boundaries.push_back(portion); } else { }</pre>

```

        boundaries.push_back(boundaries[i-1]+portion);
    }
}

// spin roulette wheel
vector<vector<int>> selection;
for (int i = 0; i < population.size(); i++) {
    int min = 0;
    int max = 100;
    double x = getRandomInt(min, max)/100.0;

    bool found = false;
    int j = 0;
    while (!found && j < boundaries.size())
    {
        if (x <= boundaries[j]) {

selection.push_back(population[j]);
        found = true;
    }
    j++;
}

if (!found && x == 1) {

selection.push_back(population[population.size()-1]);
}
}

return selection;
}

```

<p><code>crossOver</code></p> <p>Input: populasi Output: populasi baru hasil cross-over</p> <p>Fungsi ini menerapkan cross-over berdasarkan ordered cross over. Fungsi menerapkan <i>cross-over</i> untuk setiap dua <i>parent</i>. Misal kubus (<i>parent</i> dan <i>child</i>) direpresentasikan sebagai array <i>integer</i>. Pertama-tama, fungsi akan mengambil 2 titik random (<i>start</i> dan <i>end</i>) sebagai <i>range index</i> – penanda bagian yang akan di <i>cross-over</i>. Fungsi kemudian akan mengambil subarray dari kedua <i>parent</i> di antara <i>range index</i> tersebut dan menyimpannya (subarray dari <i>parent2</i> yang disimpan sebagai <i>slice1</i> dan sebaliknya). Fungsi kemudian akan mengiterasi <i>parent</i> untuk mengisi elemen <i>child</i>. Apabila index yang sedang di-iterasi di dalam <i>range index</i> bagian <i>cross-over</i>, maka fungsi meng-<i>append</i> subarray yang telah disimpan sebelumnya (subarray dari <i>parent2</i> yang disimpan sebagai <i>slice1</i> di-<i>append</i> ke <i>child1</i> dan sebaliknya). Apabila tidak, jika elemen pada index yang sedang di-iterasi muncul di dalam subarray yang telah/akan di-<i>append</i>, maka elemen diabaikan (<i>skip</i>). Jika elemen pada index yang sedang di-iterasi tidak muncul di dalam subarray yang telah/akan di-<i>append</i>, maka elemen di-<i>append</i> ke dalam <i>child</i>.</p> <p>Apabila jumlah populasi ganjil, maka individu/kubus terakhir akan di-<i>cross-over</i> dengan individu/kubus kedua terakhir</p>	<pre> vector<vector<int>> crossOver(vector<vector<int>> population) { // cross over using ordered // cross over int nPop = population.size(); if (nPop % 2 == 1) { // if population is odd // one-sided crossover int point1 = getRandomInt(0, n*n*n-2); int point2 = getRandomInt(0, n*n*n-2); int start = min(point1, point2); int end = max(point1, point2); vector<int> parent1 = population[nPop-2]; vector<int> parent2 = population[nPop-1]; vector<int> slice(parent1.begin()+start, parent1.begin()+end+1); vector<int> child; // cross over to child int j = 0; // iterator for parent int p = 0; // iterator for next element to add int q = 0; // iterator for slice while (j < parent2.size() && p < parent2.size()) { // if between start and end, append slice if (p >= start && p <= end) { child.push_back(slice[q]); p++; q++; continue; } int num = parent2[j]; auto it = find(slice.begin(), slice.end(), num); if (it != slice.end()) { // if slice contains num (duplicate), skip j++; continue; } // else add num to child child.push_back(num); p++; } } } </pre>
	<p>IF3170 Inteligensi Artifisial - Tugas Besar 1 19</p>

	<p>secara <i>one-sided</i> (individu kedua terakhir tidak menerima <i>cross-over</i> dari individu terakhir).</p>	<pre> j++; } population[nPop-1] = child; } for (int i = 0; i < nPop-(nPop%2); i += 2) { int point1 = getRandomInt(0, n*n*n-2); int point2 = getRandomInt(0, n*n*n-2); int start = min(point1, point2); int end = max(point1, point2); vector<int> parent1 = population[i]; vector<int> parent2 = population[i+1]; vector<int> slice1(parent2.begin()+start, parent2.begin()+end+1); vector<int> slice2(parent1.begin()+start, parent1.begin()+end+1); vector<int> child1; vector<int> child2; // cross over to child 1 int j = 0; // iterator for parent int p = 0; // iterator for next element to add int q = 0; // iterator for slice while (j < parent1.size() && p < parent1.size()) { // if between start and end, append slice if (p >= start && p <= end) { child1.push_back(slice1[q]); p++; q++; continue; } int num = parent1[j]; auto it = find(slice1.begin(), slice1.end(), num); if (it != slice1.end()) { // if slice contains num (duplicate), skip j++; continue; } } } </pre>
--	---	--

		<pre> // else add num to child child1.push_back(num); p++; j++; } // cross over to child 2 j = 0; // iterator for parent p = 0; // iterator for next element to add q = 0; // iterator for slice while (j < parent2.size() && p < parent2.size()) { // if between start and end, append slice if (p >= start && p <= end) { child2.push_back(slice2[q]); p++; q++; continue; } int num = parent2[j]; auto it = find(slice2.begin(), slice2.end(), num); if (it != slice2.end()) { // if slice contains num (duplicate), skip j++; continue; } // else add num to child child2.push_back(num); p++; j++; } population[i] = child1; population[i+1] = child2; } return population; } </pre>
mutation	<p>Input: populasi Output: populasi baru hasil mutasi Fungsi ini melakukan mutasi terhadap setiap individu dalam populasi. Untuk setiap individu,</p>	<pre> vector<vector<int>> mutation(vector<vector<int>> population) { for (int i = 0; i < population.size(); i++) { int y = getRandomInt(1, n*n*n-1); // num of mutations </pre>

	<p>fungsi akan menentukan jumlah acak, misalnya n, sebagai jumlah <i>swap</i> yang akan dilakukan (jumlah elemen yang akan dimutasi). Kemudian, untuk sebanyak n kali, melakukan swap antara 2 elemen. 2 elemen yang akan di-<i>swap</i> ditentukan secara acak.</p>	<pre> for (int j = 0; j < y; j++) { int x = getRandomInt(1, 100); if (x < 5) { // 5 chance int pos1 = getRandomInt(0, n * n * n - 1); int pos2 = getRandomInt(0, n * n * n - 1); while(pos1 == pos2 pos1 == 62 pos2 == 62){ pos1 = getRandomInt(0, n * n * n - 1); pos2 = getRandomInt(0, n * n * n - 1); } for (int j = 0; j < population.size(); j++) { swap(population[j][pos1], population[j][pos2]); } } } return population; } </pre>
main	<p>Fungsi main dalam Genetic Algorithm menerima input banyak populasi dan banyak iterasi, lalu menggabungkan fungsi selection, crossOver, dan mutation untuk melakukan GeneticAlgorithm sejumlah banyak iterasi dengan jumlah populasi berdasarkan input.</p>	<pre> int32_t main(){ cout << "Ukuran Kubus: "; cin >> n; int nPop; cout << "Banyak Populasi: "; cin >> nPop; int maxIter; cout << "Banyak Iterasi: "; cin >> maxIter; auto beg = high_resolution_clock::now(); c = n * (n * n * n + 1); c /= 2; // make the cube vector<vector<int>> population = makePopulation(nPop, n); cout << "Start Population: \n"; for (int i = 0; i < population.size(); i++) { cout << "Cube " << i+1 << ":" \n"; printCube(population[i]); } } </pre>

```

        for (int i = 0; i < population.size(); i++) {
            long long fitness = distance(population[i]);

            cout << "Fitness Cube " <<
i+1 << ":" << fitness << "\n";
        }
        long long averageStart = findAverage(population);
        cout << "Average Fitness: " <<
averageStart << endl;
        long long optimumStart = findOptimum(population);
        cout << "Optimum: " <<
optimumStart << endl;

        ofstream logs("genetic_log.txt");
        logs << optimumStart << " " <<
averageStart << " " << 0 << endl;

        for (int itr = 0; itr < maxIter;
itr++){
            // selection
            vector<vector<int>>
selectionRes = selection(population);

            // Cross-Over
            vector<vector<int>>
crossOverRes = crossOver(selectionRes);

            // mutation
            vector<vector<int>>
mutationRes = mutation(crossOverRes);

            // next iteration
            population = mutationRes;

            if (itr > 0 && itr != maxIter-1
&& (itr+1) % 20 == 0) {
                cout << "Iteration-" << itr+1
<< ":" \n";
                long long averageItr =
findAverage(population);
                cout << "Average Fitness: " <<
averageItr << endl;

                long long optimumItr =
findOptimum(population);
                cout << "Optimum: " <<
optimumItr << endl;
}
}

```

```

        logs << optimumItr << " "
<< averageItr << " " << itr+1 <<
endl;
    }
}

cout << "Final Population: \n";
for (int i = 0; i < population.size(); i++) {
    cout << "Cube " << i+1 << ":" 
\n";
    printCube(population[i]);
}

for (int i = 0; i < population.size(); i++) {
    cout << "Fitness Cube " <<
i+1 << ":" <<
distance(population[i]) << "\n";
}

long long averageEnd =
findAverage(population);
cout << "Average Fitness: " <<
averageEnd << endl;

long long optimumEnd =
findOptimum(population);
cout << "Optimum: " <<
optimumEnd << endl;

logs << optimumEnd << " " <<
averageEnd << " " << maxIter <<
endl;

auto en =
high_resolution_clock::now();
auto dur =
duration_cast<microseconds>(en -
beg);
cout << "Duration (in microsec): "
<< dur.count() << endl;

/** Sesuaikan dengan sistem
anda! **/ 

system("C:/Python/Python312/pyth
on.exe plot_genetic.py
genetic_log.txt I log");

return 0;
}

```

2.2.5 Hill-Climbing with Sideways Move

Fungsi/kelas	Deskripsi	Source Code
makeRandomCube (fungsi yang teradapat pada Genetic Algorithm)	<p>Input: ukuran kubus</p> <p>Output: kubus (dalam bentuk vector<int>)</p> <p>Fungsi ini menghasilkan konfigurasi kubus dengan angka unik yang telah diacak</p>	<pre>vector<int> makeRandomCube(int n) { random_device rd; mt19937 rng(rd()); vector<int> cube; for(int i = 1; i <= n * n * n; i++){ cube.push_back(i); } shuffle(begin(cube), std::end(cube), rng); for(int i = 0; i < cube.size(); i++){ if(cube[i] == 63){ swap(cube[i], cube[62]); break; } } return cube; }</pre>
main	<p>Fungsi main dalam Climbing-Hill with Sideways Move menerima input banyak populasi, lalu membangkitkan magic cube dengan elemen acak untuk melakukan pencarian solusinya dengan Steepest Ascent Climbing-Hill sampai algoritma ini tidak dapat menemukan neighbor yang nilainya lebih tinggi atau sama dengan current neighbor.</p>	<pre>int32_t main(){ //evalAnswer(); cout << "Ukuran Kubus: "; cin >> n; auto beg = high_resolution_clock::now(); c = n * (n * n * n + 1); c /= 2; vector<int> cube = makeRandomCube(n); vector<int> startingCube = cube; sumVec = distanceVec(cube); vector<int> ans; long long curDist = distance(cube); long long difference; int pos1, pos2; int steps = 0; int maxFail = 2e6; int minToRestart = 64; long long mini =</pre>

```

LLONG_MAX;

    bool decreasingVal = true;

    ofstream
logs("hill-climbing_log.txt");

    while(decreasingVal) {
        decreasingVal = false;
        for(int pos1 = 0; pos1 <
cube.size(); pos1++) {
            for(int pos2 =
pos1 + 1; pos2 < cube.size(); pos2++) {
                swap(cube[pos1],
cube[pos2]);
                long      long
newDist = distance(cube);

                if(newDist     <=
curDist) {
                    curDist
= newDist;

                if(curDist <= mini) {
                    ans = cube;
                    mini = curDist;
                }
            }
            decreasingVal = true;
        } else {
            swap(cube[pos1], cube[pos2]);
        }
        steps++;
        cout << curDist << " " <<
steps << endl;
        logs << curDist << " " <<
steps << endl;
    }
    logs.close();

    ofstream
sc("start_cube.txt");
    ofstream
fc("final_cube.txt");

    for(auto
startingCube){&x:
        sc << x << " ";
}

```

```

    }

    sc << endl;
    sc.close();

    for(auto &x: ans){
        fc << x << " ";
    }
    fc << endl;
    fc.close();

    cout << endl;
    cout << "Final objective
function value: " << mini << endl;
    auto en = high_resolution_clock::now();
    auto dur = duration_cast<microseconds>(en - beg);
    cout << "Duration (in
microsec): " << dur.count() << endl;

    system("python3
plot_cube.py start_cube.txt
\"Starting Cube\" && python3
plot_cube.py final_cube.txt \"Final
Cube\"");
}

/** Sesuaikan dengan
sistem anda! */
system("python3
plot_annealing.py
hill-climbing_log.txt I \"Objective
function vs steps\"");

return 0;
}

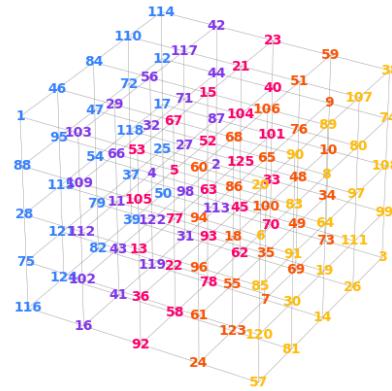
```

2.3 Hasil Eksperimen dan Analisis

2.2.1 Steepest Ascent Hill-Climbing

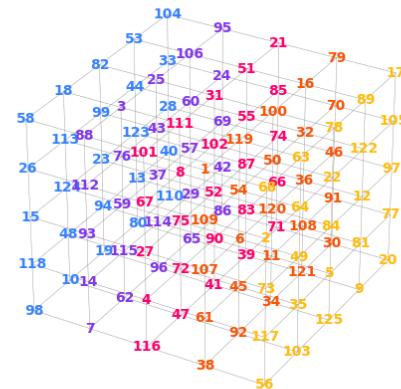
- Eksperimen Jumlah Iterasi
 - Variansi 1:
 - Sampel 1:
 - State Awal:

Starting Cube

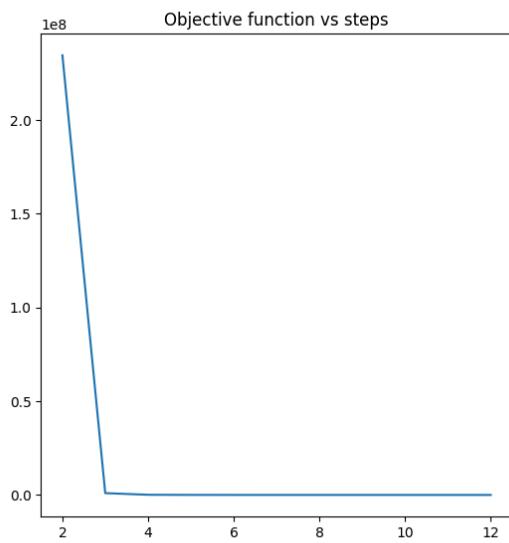


- State Akhir:

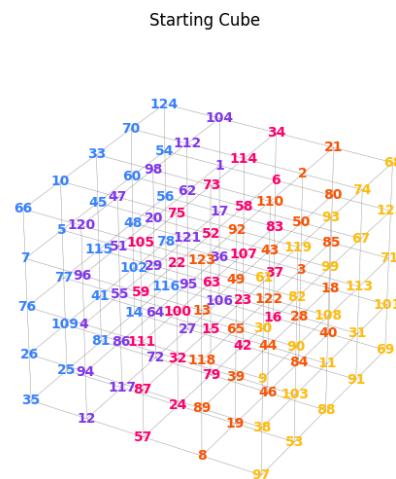
Final Cube



- Nilai objektif akhir yang dicapai: 4596
- Banyaknya iterasi: 12
- Plot nilai objektif terhadap banyaknya iterasi:

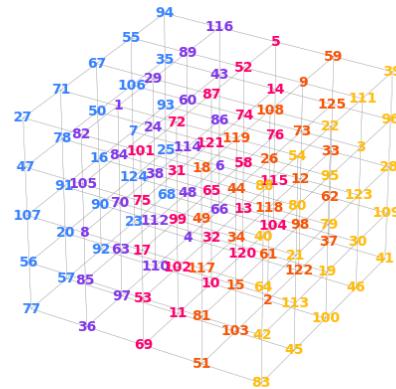


- Durasi pencarian: 1.129321 sekon
- Sampel 2:
 - State Awal:

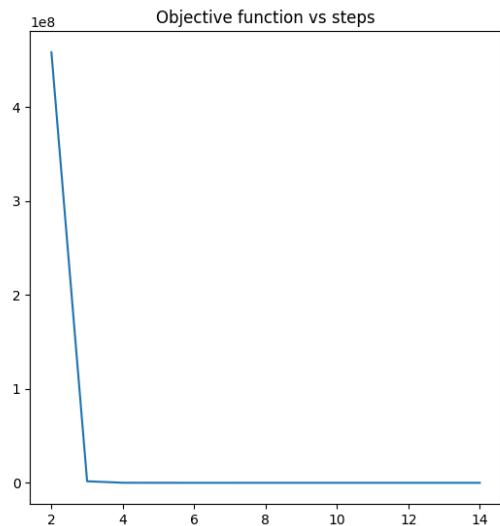


- State Akhir:

Final Cube

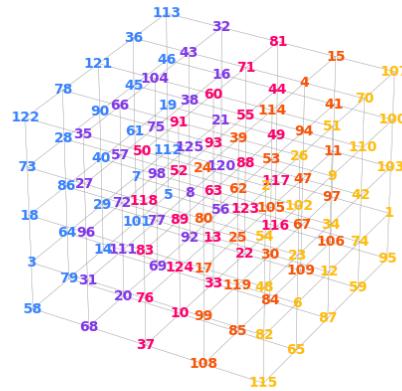


- Nilai objektif akhir yang dicapai: 4052
- Banyaknya iterasi: 14
- Plot nilai objektif terhadap banyaknya iterasi:



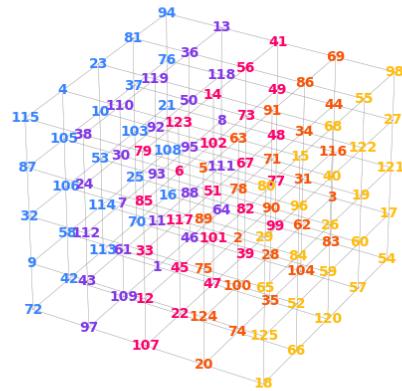
- Durasi pencarian: 1.068420 sekon
- Sampel 3:
 - State Awal:

Starting Cube

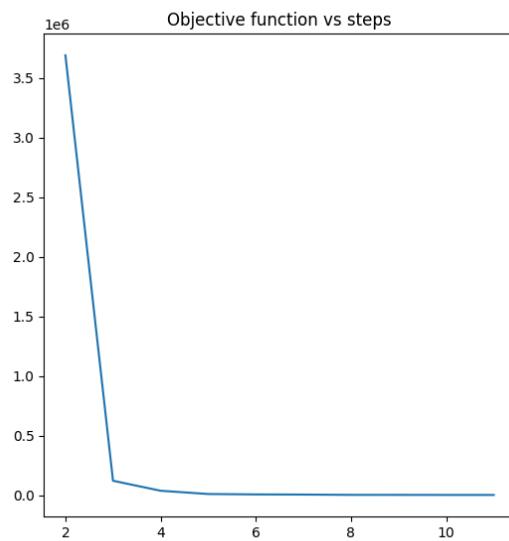


- State Akhir:

Final Cube



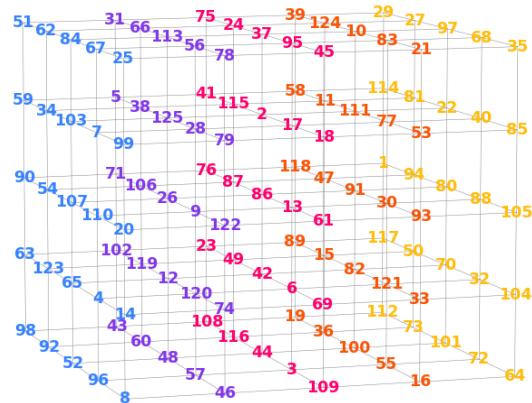
- Nilai objektif akhir yang dicapai: 3930
- Banyaknya iterasi: 11
- Plot nilai objektif terhadap banyaknya iterasi:



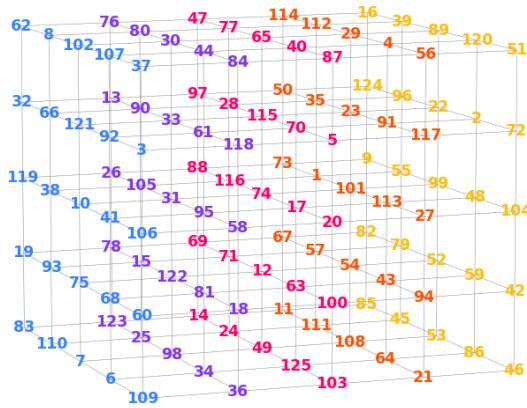
- Durasi pencarian: 0.904816 sekon

2.2.2 Simulated Annealing

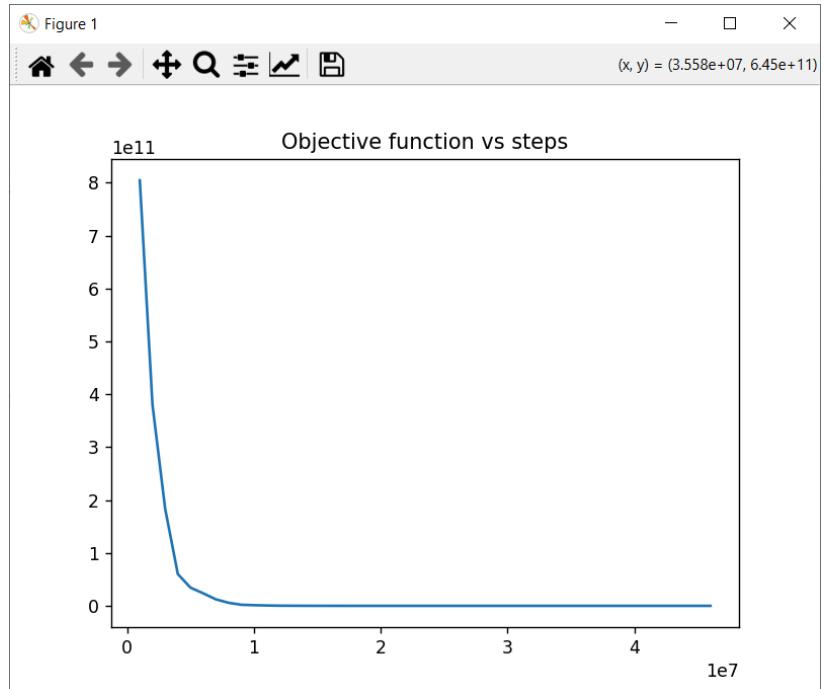
- Eksperimen Minimum Temperature
 - Variansi 1:
 - Temperatur Awal: 10^{11}
 - Cooling Rate = 0.9999993
 - Minimum Temperature: 10^{-3}
 - Sampel 1:
 - State Awal:



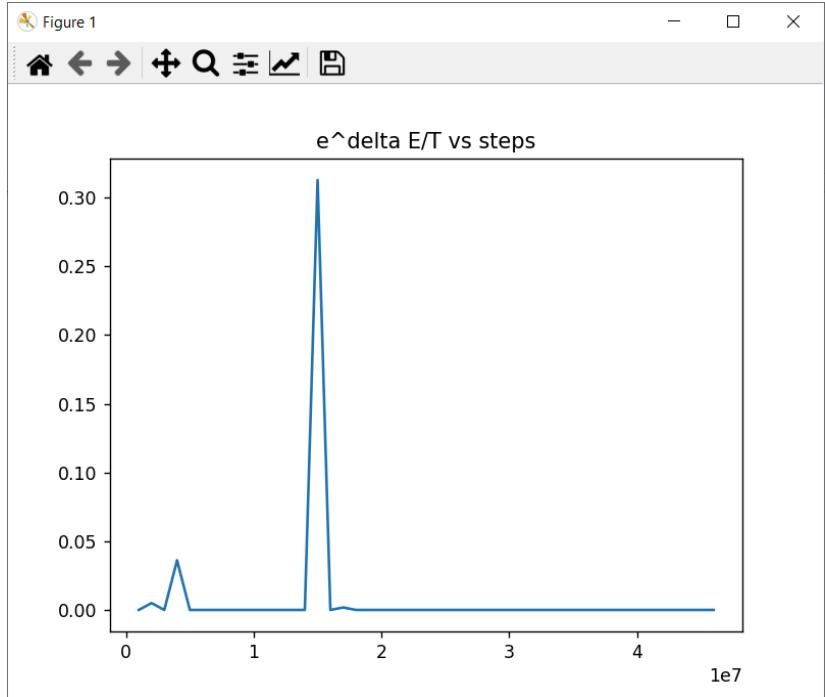
- State Akhir:



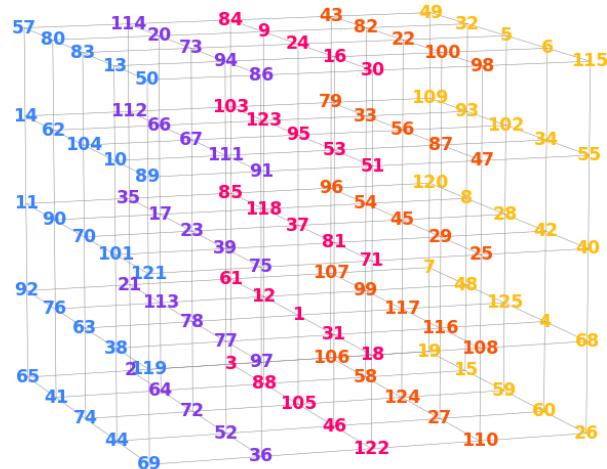
- Nilai objective function yang dicapai: **38**
- Plot objective function terhadap banyaknya iterasi:



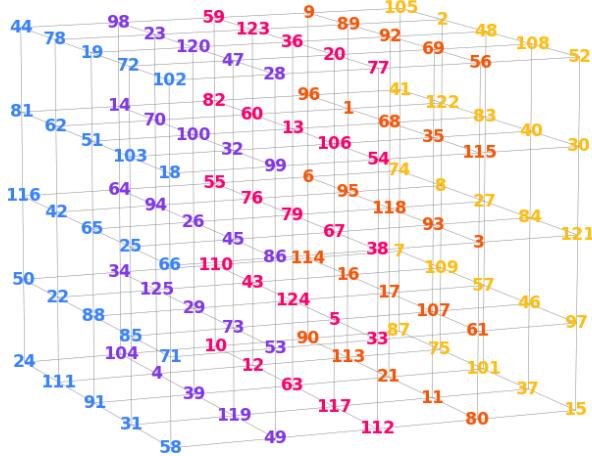
- Durasi pencarian: 24,03 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



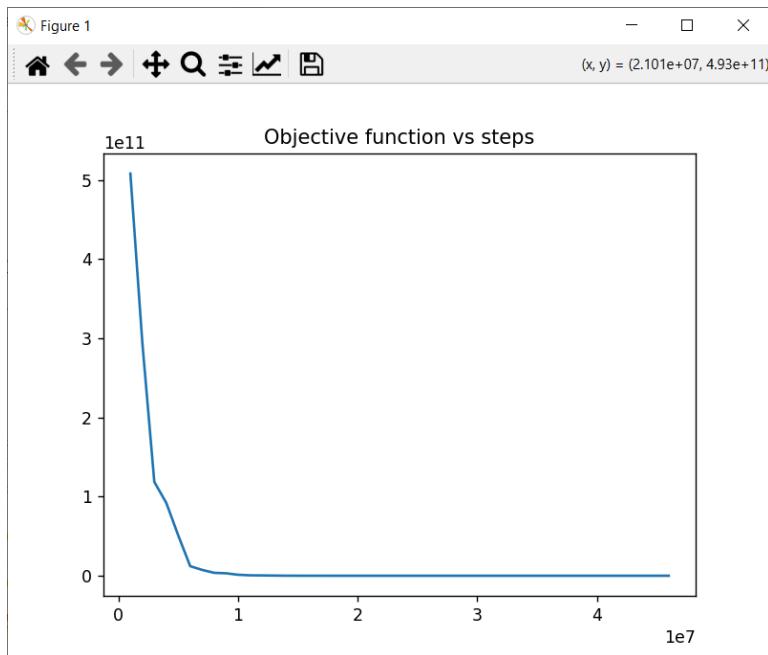
- Frekuensi stuck: 44813555 kali
- Sampel 2:
 - State Awal:



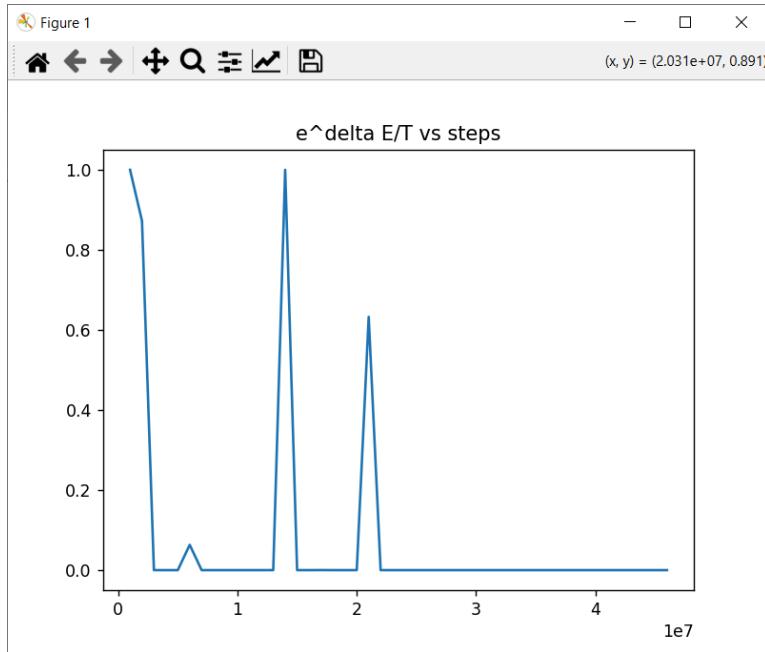
- State Akhir:



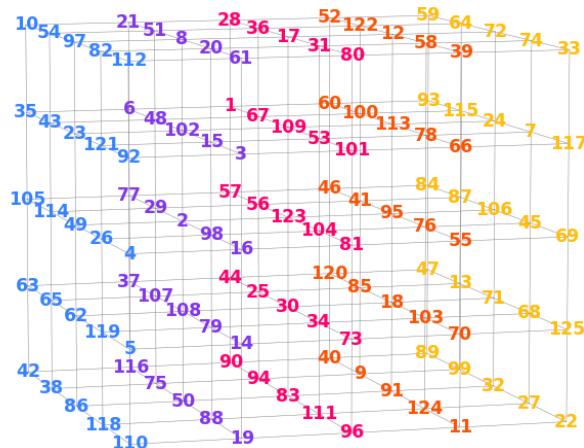
- Nilai objective function yang dicapai: **40**
- Plot objective function terhadap banyaknya iterasi:



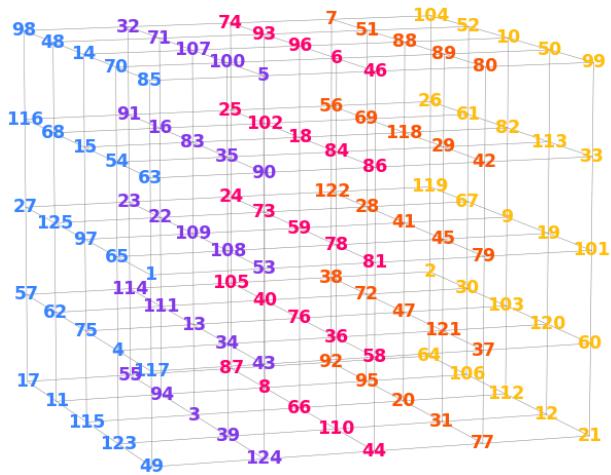
- Durasi pencarian: 21,73 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



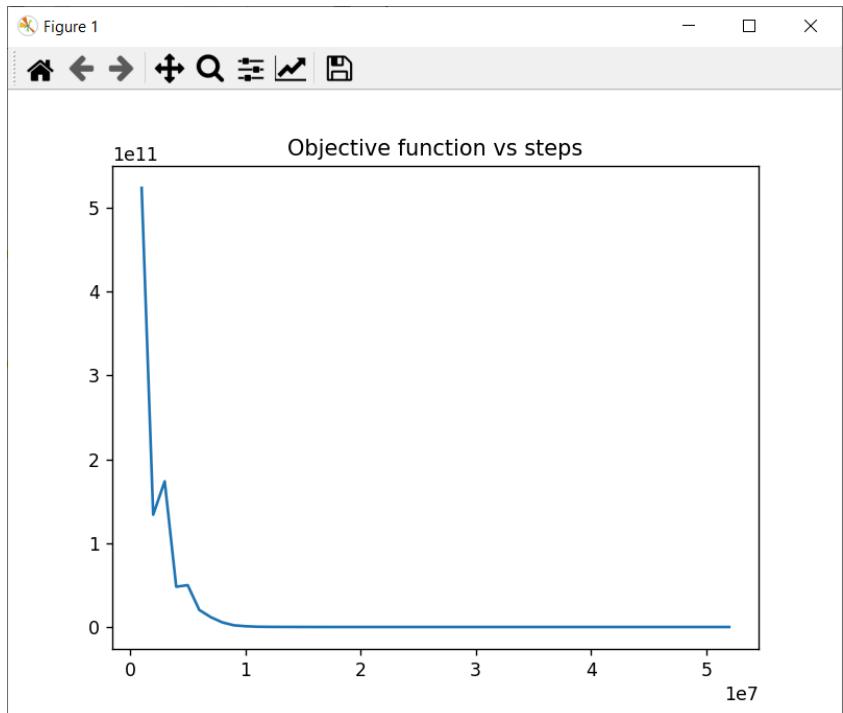
- Frekuensi stuck: 44813437 kali
- Variansi 2:
 - Temperatur Awal: 10^{11}
 - Cooling Rate = 0.9999993
 - Minimum Temperature: 10^{-5}
 - Sampel 1:
 - State Awal:



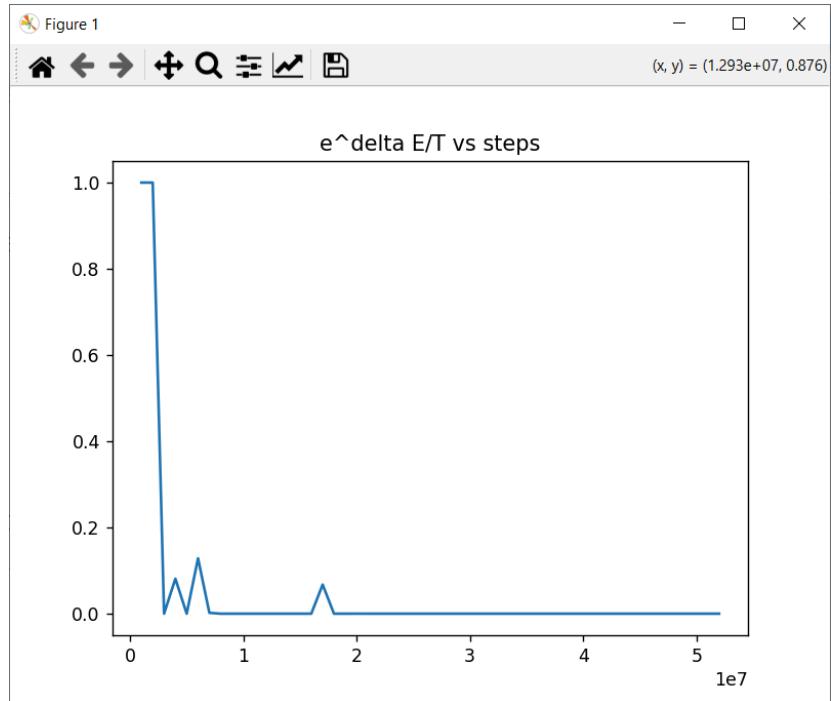
- State Akhir:



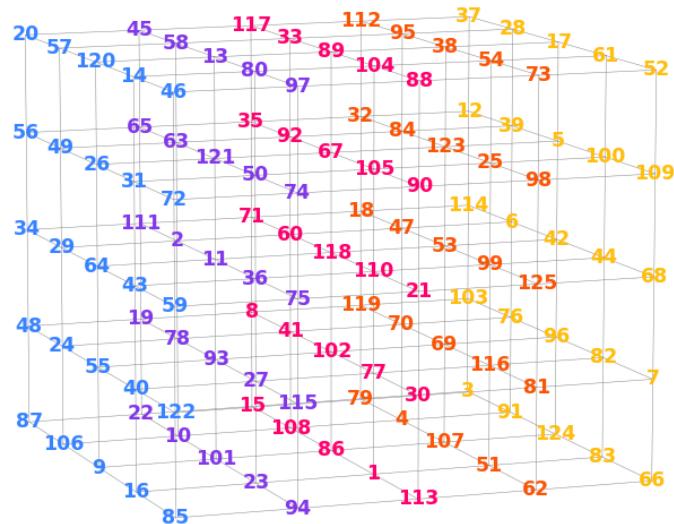
- Nilai objective function yang dicapai: **30**
- Plot objective function terhadap banyaknya iterasi:



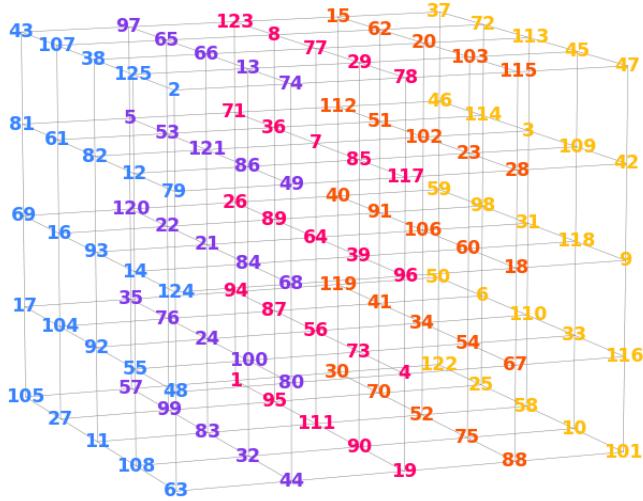
- Durasi pencarian: 37,20 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



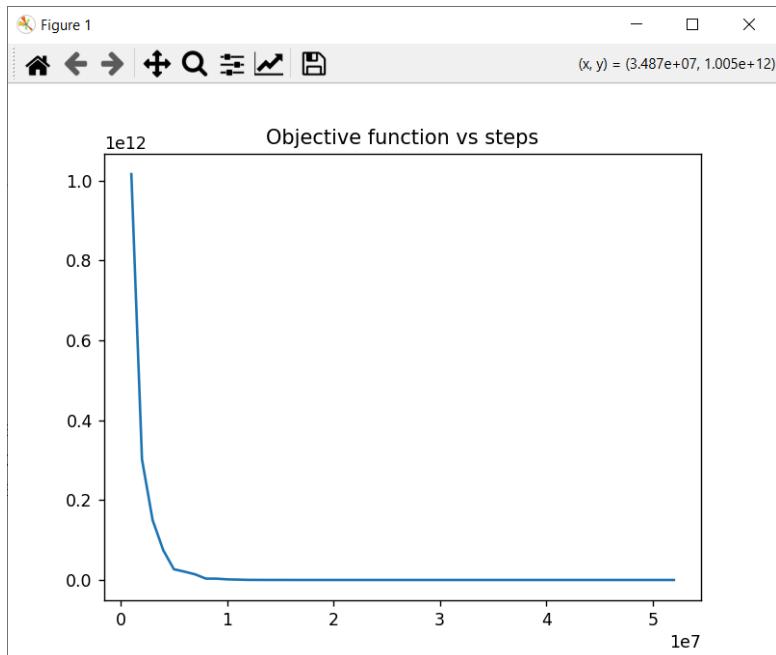
- Frekuensi stuck: 51386609 kali
- Sampel 2:
 - State Awal:



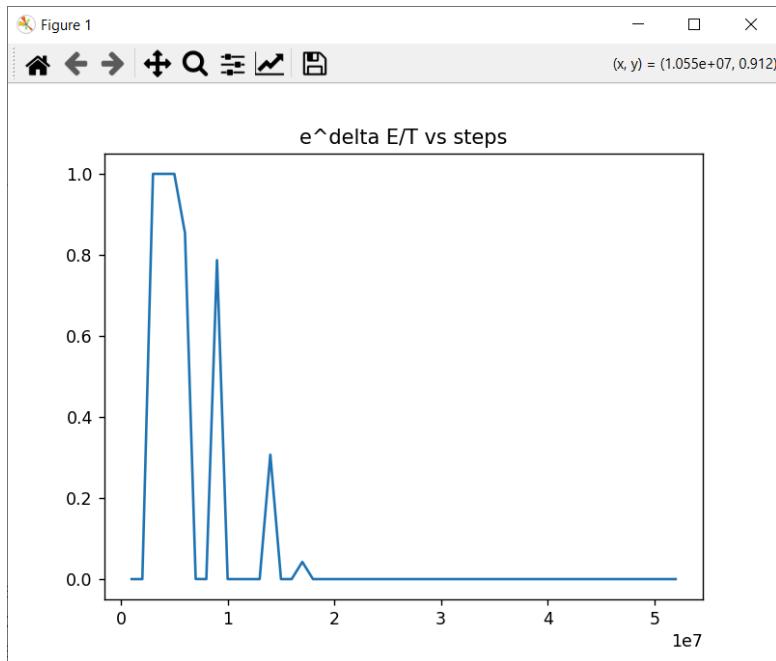
- State Akhir:



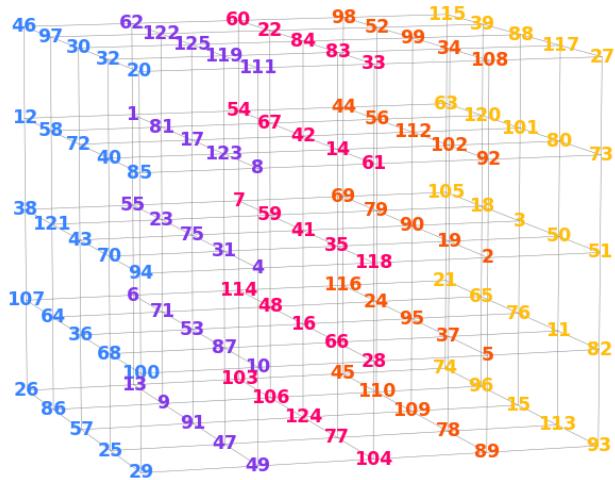
- Nilai objective function yang dicapai: **37**
- Plot objective function terhadap banyaknya iterasi:



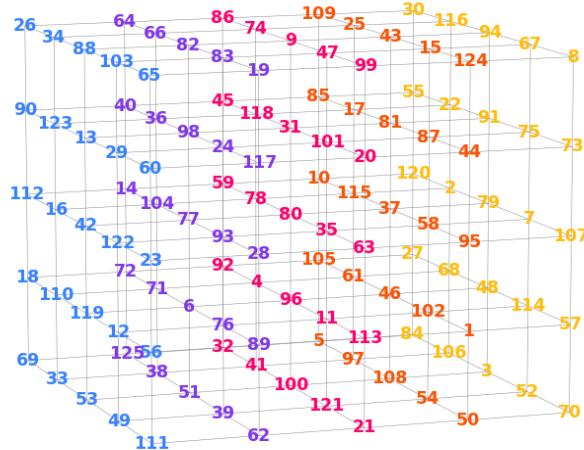
- Durasi pencarian: 26,64 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



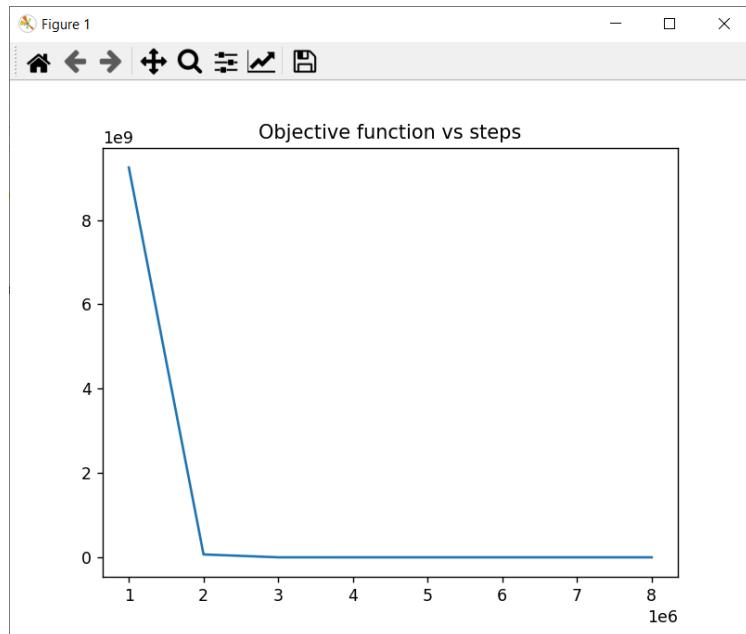
- Frekuensi stuck: 51390435 kali
- Eksperimen Cooling Rate
 - Variansi 1:
 - Temperatur Awal: 10^{11}
 - Cooling Rate = 0.999995
 - Minimum Temperature: 10^{-7}
 - Sampel 1:
 - State Awal:



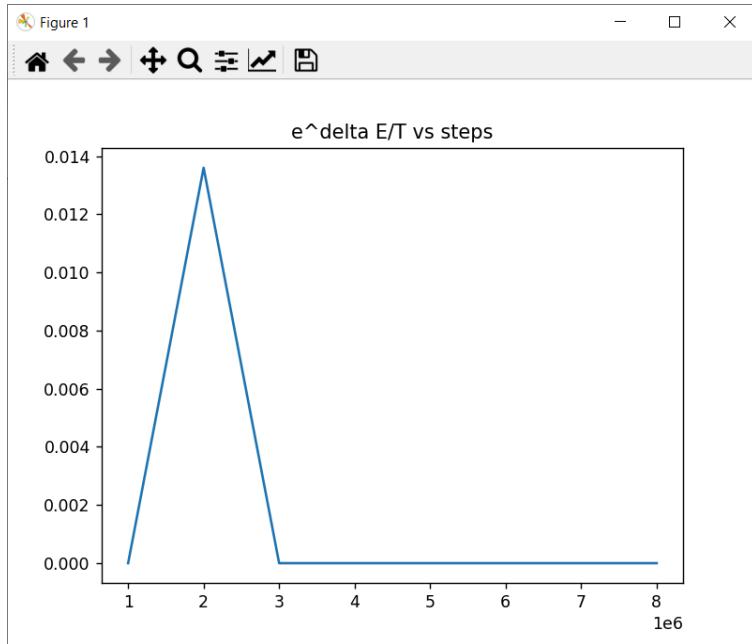
- State Akhir:



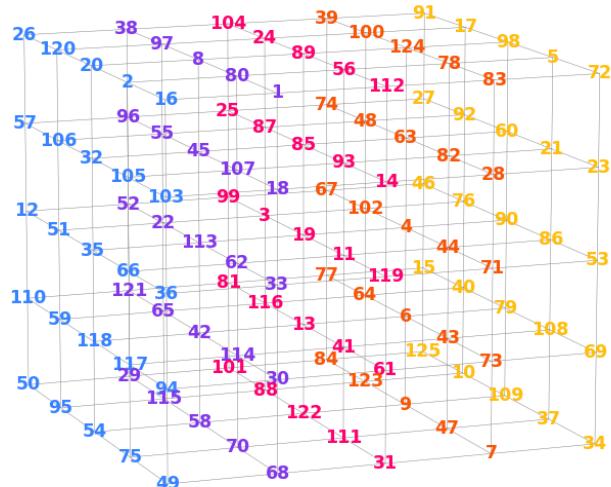
- Nilai objective function yang dicapai: **44**
- Plot objective function terhadap banyaknya iterasi:



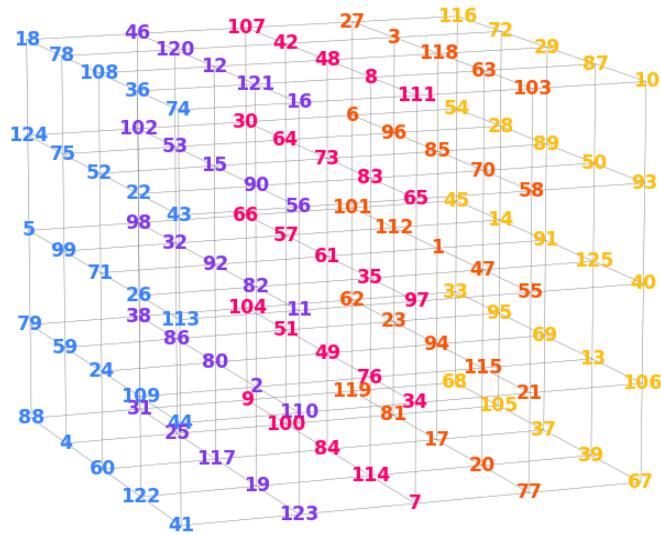
- Durasi pencarian: 3,85 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



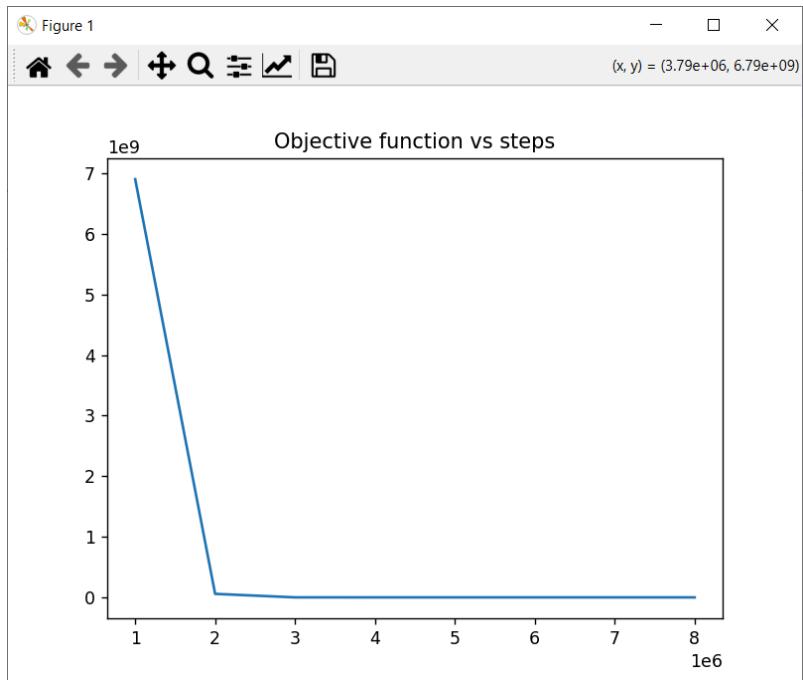
- Frekuensi stuck: 8115020 kali
- Sampel 2:
 - State Awal:



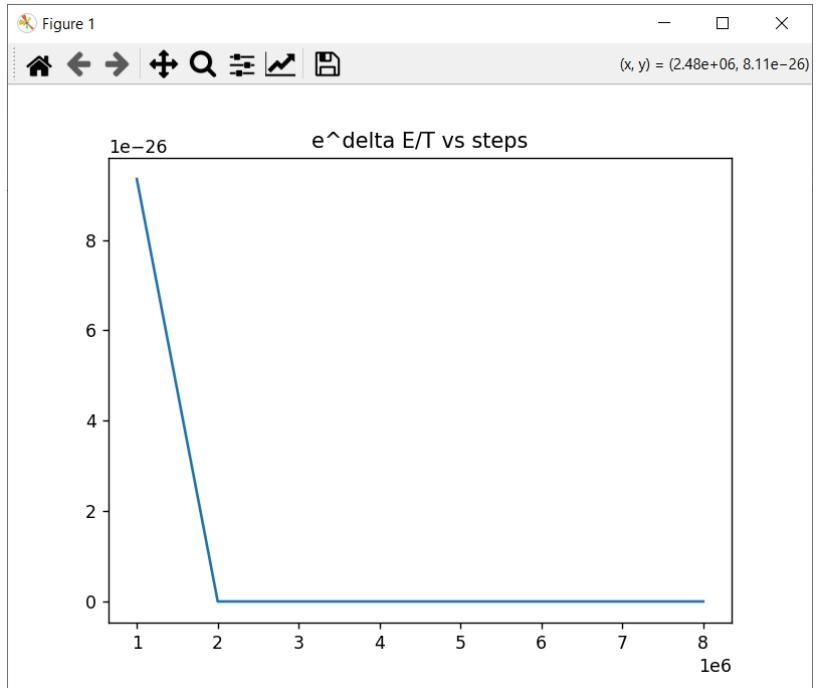
- State Akhir:



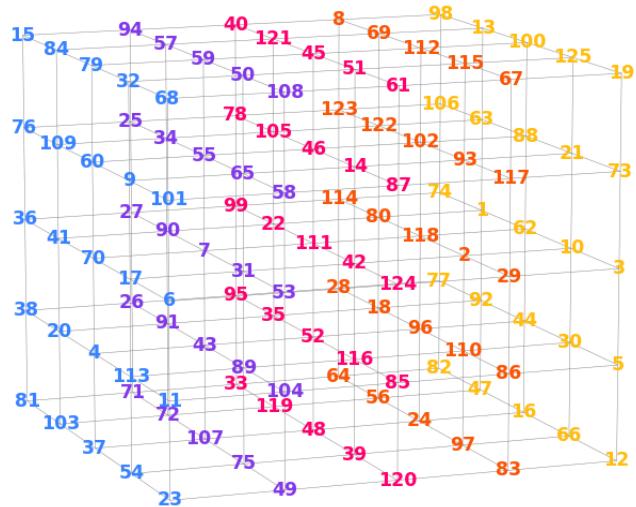
- Nilai objective function yang dicapai: **56**
- Plot objective function terhadap banyaknya iterasi:



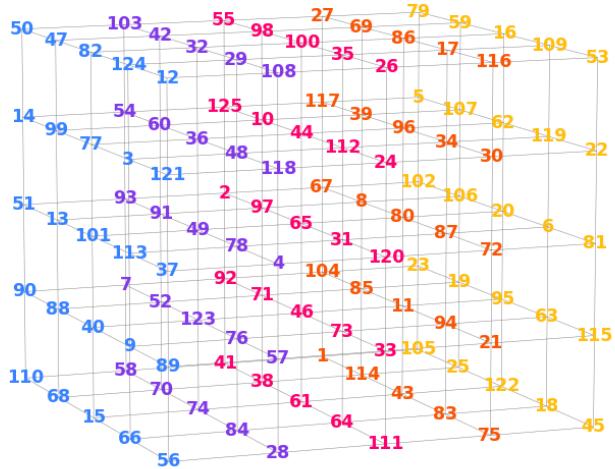
- Durasi pencarian: 3,69 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



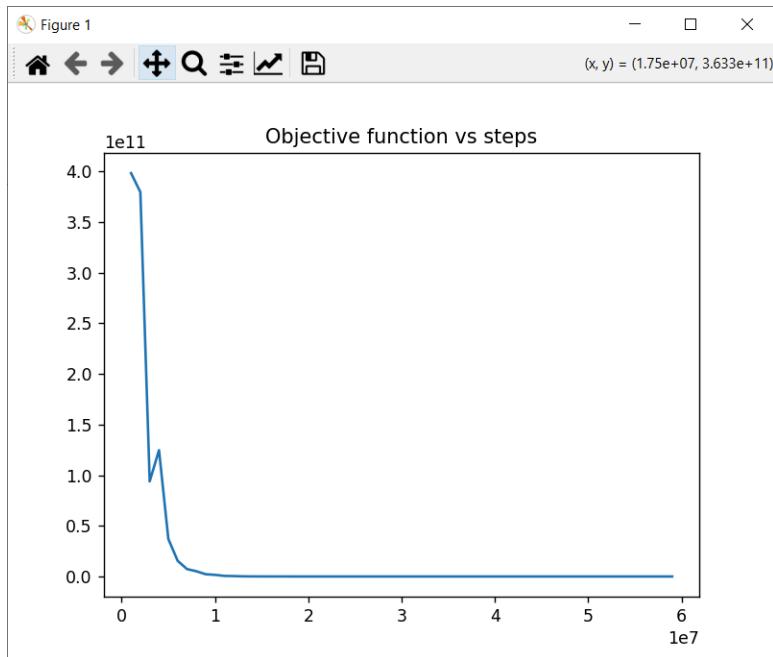
- Frekuensi stuck: 8115182 kali
- Variansi 2:
 - Temperatur Awal: 10^{11}
 - Cooling Rate = 0.9999993
 - Minimum Temperature: 10^{-7}
 - Sampel 1:
 - State Awal:



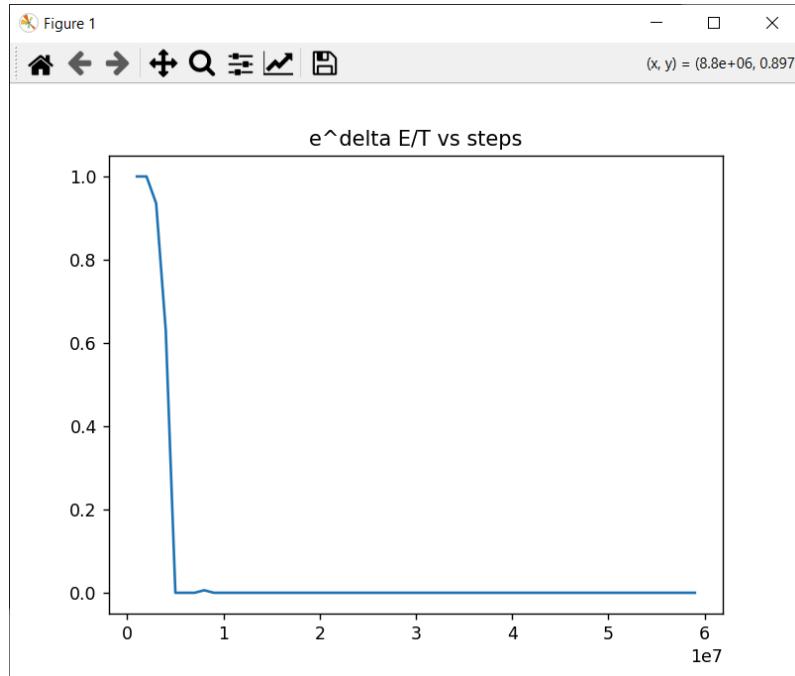
- State Akhir:



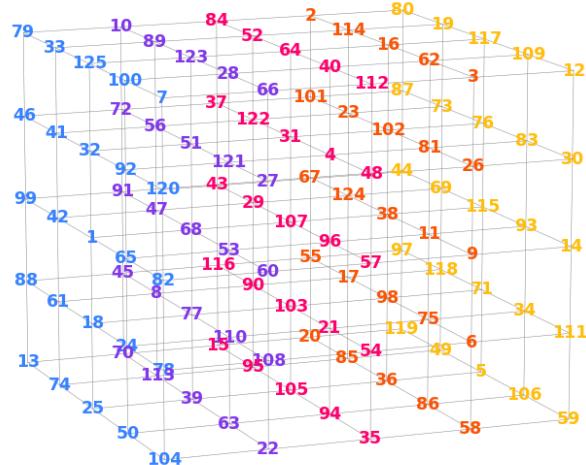
- Nilai objective function yang dicapai: **39**
- Plot objective function terhadap banyaknya iterasi:



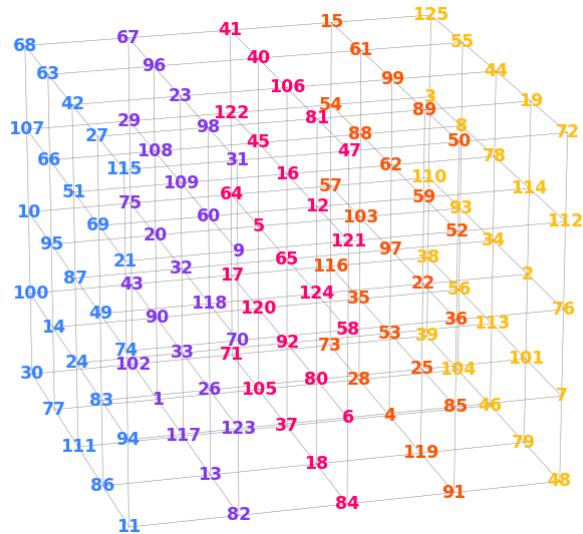
- Durasi pencarian: 41,22 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



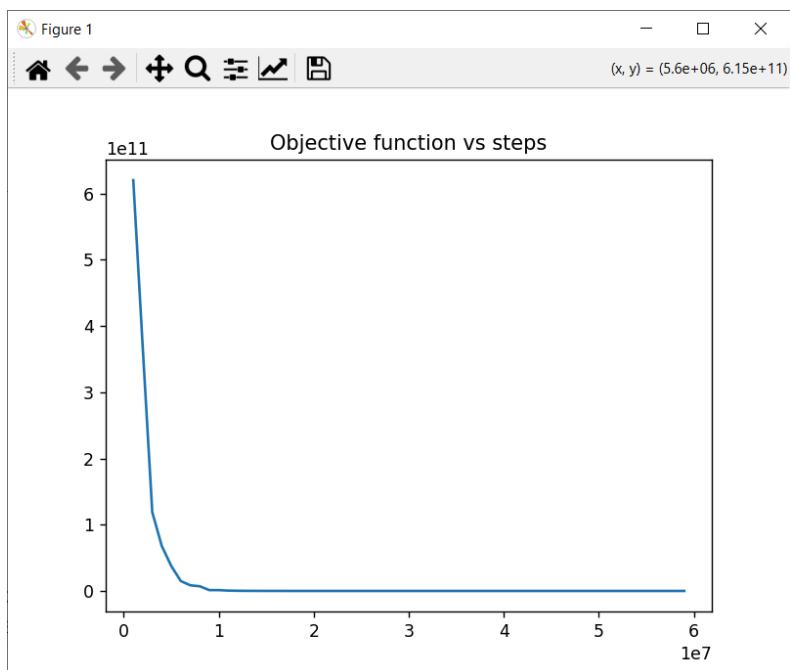
- Frekuensi stuck: 57970272 kali
- Sampel 2:
 - State Awal:



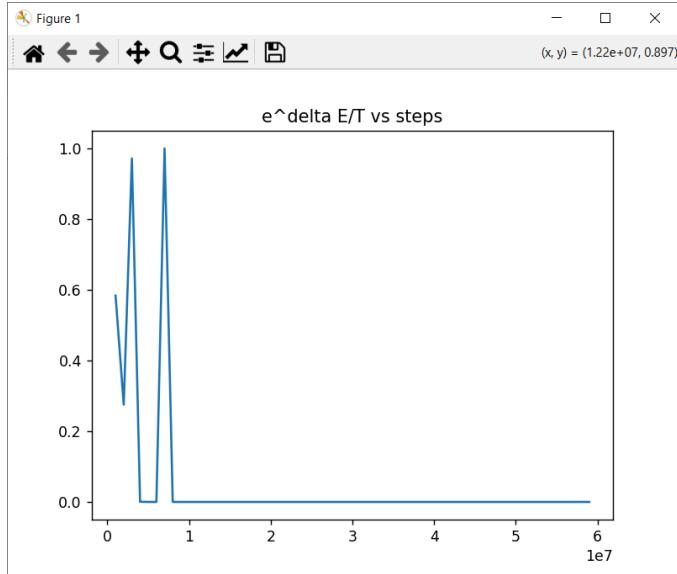
- State Akhir:



- Nilai objective function yang dicapai: **34**
- Plot objective function terhadap banyaknya iterasi:



- Durasi pencarian: 32,65 s
- Plot $e^{\Delta E/T}$ terhadap banyak iterasi:



- Frekuensi stuck: 57968392 kali

2.2.3 Genetic Algorithm

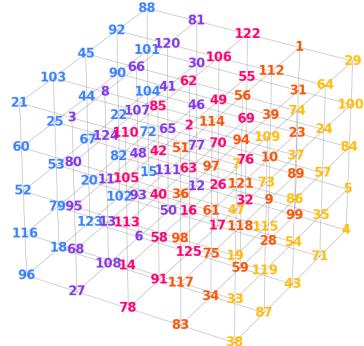
- #### • Eksperimen Jumlah Populasi

- ## ○ Variansi 1:

- Jumlah Populasi: **10**
 - Banyak Iterasi: **100**
 - Sampel 1:

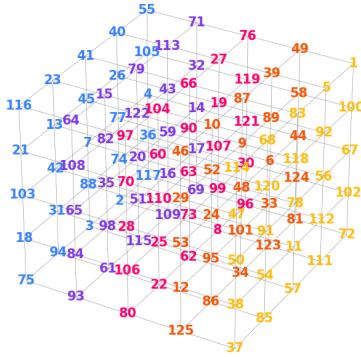
- State Awal:

- Cube 1:



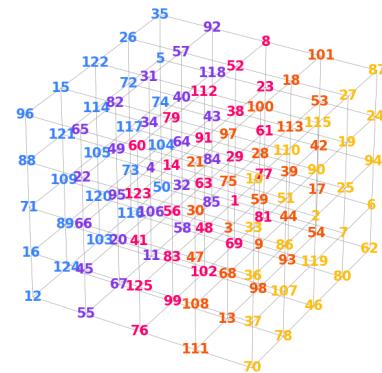
- Cube 2:

Start Population Cube 2:



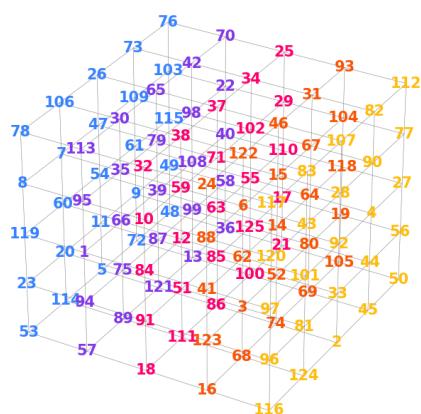
- Cube 3:

Start Population Cube 3:



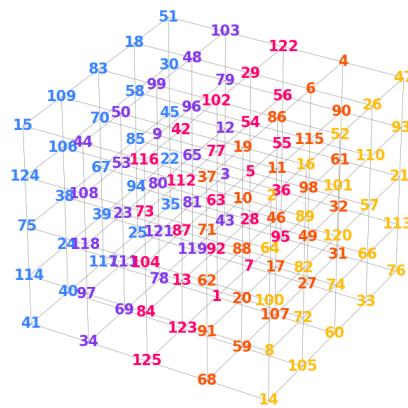
- Cube 4:

Start Population Cube 4:



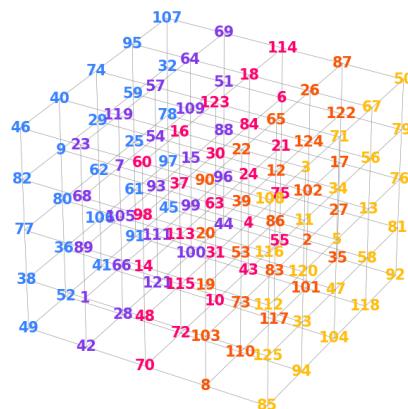
- Cube 5:

Start Population Cube 5:



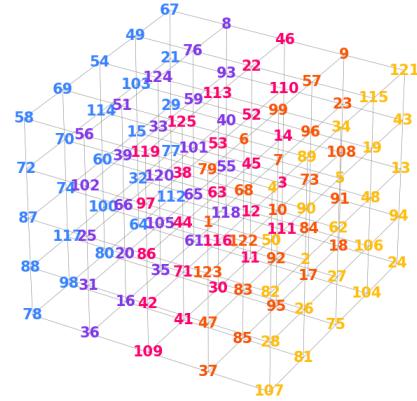
- Cube 6:

Start Population Cube 6:



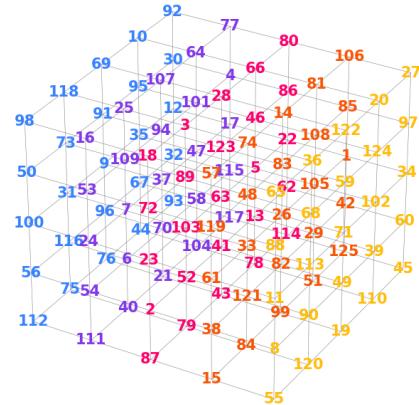
- Cube 7:

Start Population Cube 7:



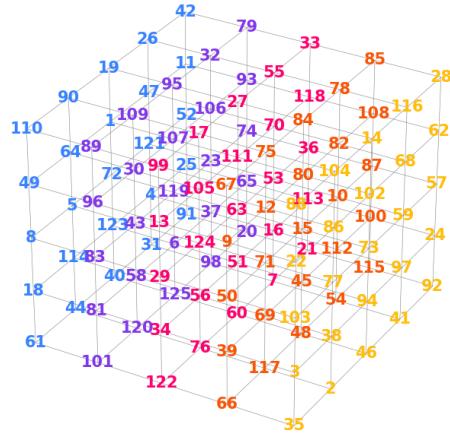
- Cube 8:

Start Population Cube 8:



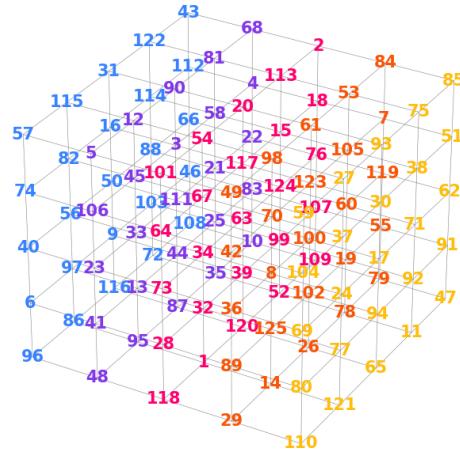
- Cube 9:

Start Population Cube 9:



- Cube 10:

Start Population Cube 10:



- State Akhir:

- Cube 1:

```
11 34 59 23 63 79 81 109 52 123 111 77 54 14 13  
24 36 82 64 103 125 45 49 31 18 120 71 108 47 16  
93 90 98 106 44 51 80 101 116 38 19 124 21 70 39  
9 86 10 61 102 84 26 48 89 53 12 105 85 87 32 72  
119 41 7 50 30 83 68 58 28 95 118 122 8 55 88 97  
92 100 91 94 4 74 107 117 20 5 69 29 42 2 67 27 76
```

75 37 99 73 65 66 6 104 17 25 22 3 112 57 96 1 115
15 121 35 43 33 110 114 113 46 40 62 56 78 60

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

- Cube 6:

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

- Cube 7:

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

- Cube 8:

13 56 59 6 22 102 29 109 81 123 111 69 108 26 120
52 76 82 67 46 66 97 113 110 54 37 71 3 47 27 93
106 98 99 44 51 11 101 116 38 19 124 21 70 39 91

86 10 61 79 84 65 48 85 53 12 105 34 87 32 50 119
41 7 20 30 83 68 58 28 95 4 122 8 55 88 118 90 96
33 43 35 18 107 117 115 5 103 40 100 78 36 14 49
25 114 60 31 92 89 15 104 17 73 77 64 23 1 94 2 42
80 63 125 57 9 75 62 112 16 72 74 24 45 121

- Cube 9:

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

- Cube 10:

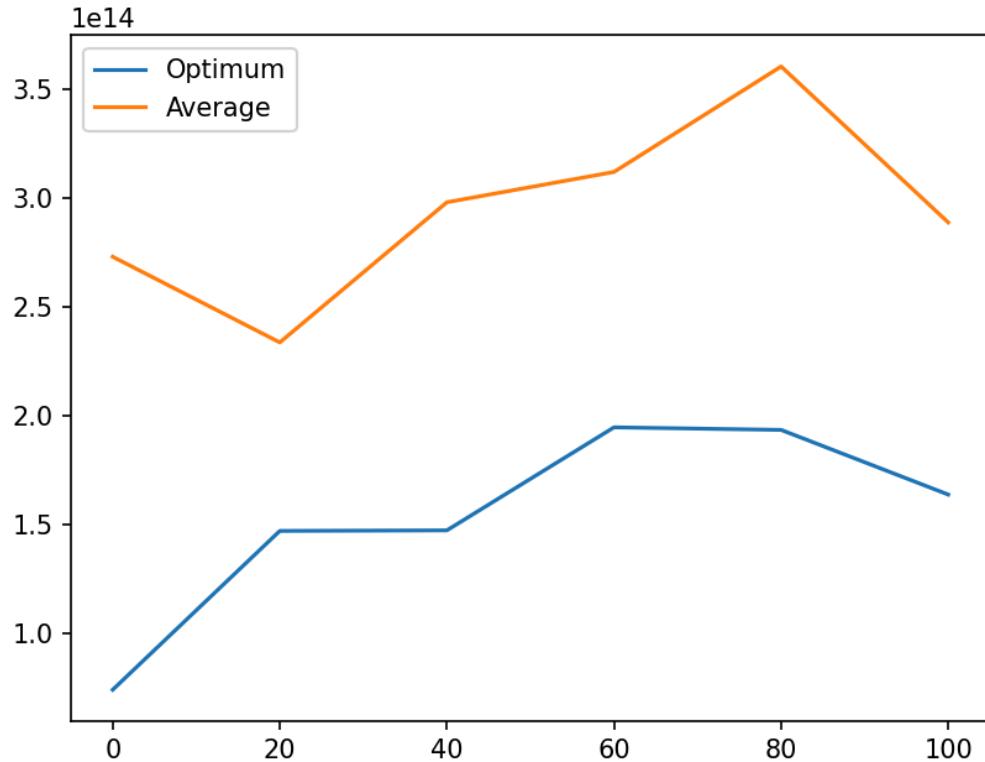
106 53 70 123 20 27 63 29 38 37 68 120 90 46 62
80 61 55 13 108 19 16 39 51 75 99 76 118 66 96
114 124 86 11 113 56 40 101 9 102 10 49 28 117 94
43 89 95 115 79 84 78 91 15 47 67 105 85 107 31
35 5 119 41 64 87 17 83 116 93 58 125 18 26 82
103 81 54 100 73 50 98 110 24 14 6 32 109 121 88
45 77 111 72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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 235247155390570
- Fitness Cube 2: 235247155390570
- Fitness Cube 3: 230700418792584
- Fitness Cube 4: 286079055861674
- Fitness Cube 5: 254555502563911
- Fitness Cube 6: 294896347128189

- Fitness Cube 7: 305945865259700
- Fitness Cube 8: 163716548046924
- Fitness Cube 9: 305609657655240
- Fitness Cube 10: 575047906646795
- Average Fitness: 288704561273615
- Optimum: 163716548046924

■ Plot:



■ Durasi: 1030150 microsekon

■ Sampel 2:

• State Awal:

○ Cube 1:

```

29 25 117 90 106 35 121 84 119 14 46 48 31 12 87
17 56 44 38 2 83 11 125 49 19 64 109 85 24 113 5
71 6 3 104 77 124 89 47 72 79 33 20 27 23 60 111
40 30 100 9 22 105 65 81 66 41 96 8 39 32 78 63 67
73 108 59 116 118 70 37 91 54 76 43 57 122 93 52

```

28 69 80 68 94 1 95 74 112 15 7 75 53 82 62 36 88
4 102 13 21 18 114 26 61 86 103 98 50 16 55 115
97 123 101 110 92 99 120 42 107 45 51 34 10 58

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

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

67 34 54 24 123 100 112 86 70 87 78 2 31 77 93 95

63 48 80 113 38 4 83 105 8 121 114 20 89 43 6 7 9

28 91 40 3 39 94 125 56 122 25 57 82 30 76 64 75

14 19 65 50 66 22 41 36 108 60 49 98 53 16 26 45

118 5 102 111 97 103 35 106 15 21 61 88 99 117

- Cube 9:

61 111 104 2 100 64 74 90 51 122 110 16 81 87 83

123 58 115 9 57 120 80 15 47 88 121 6 43 14 29

108 118 49 93 39 117 92 45 96 30 41 62 67 109 27

105 89 59 50 56 5 113 38 107 44 13 112 97 102 75

77 85 63 103 4 8 48 7 114 19 21 94 124 20 60 17 42

54 101 84 11 98 65 71 86 22 3 12 31 55 32 24 95 33

1 119 106 52 28 125 46 53 82 70 69 37 73 79 26 18

34 99 91 68 76 36 40 23 78 116 66 25 72 10 35

- Cube 10:

92 11 15 90 70 48 38 39 109 55 65 73 49 67 61 56

10 51 21 14 5 112 66 121 84 35 123 54 6 87 29 101

52 8 16 76 97 53 23 18 94 80 41 1 36 95 2 30 72 96

79 118 60 47 32 115 13 125 104 75 117 74 63 83 78

27 88 20 108 77 19 9 100 113 50 28 34 120 25 105

59 102 114 110 31 62 22 82 43 107 45 26 98 103 4

37 46 69 116 111 85 71 106 3 33 86 44 68 58 40 99

7 89 119 81 122 42 91 124 17 93 57 64 12 24

- State Akhir:

- Cube 1:

2 123 49 83 89 20 18 7 81 13 90 80 61 107 78 44 98

101 42 108 34 82 119 70 19 46 1 32 8 110 53 50 66

12 111 86 26 85 6 96 88 87 24 63 115 64 36 121 33

94 28 79 38 117 48 113 73 16 112 45 91 15 104 72

4 102 58 76 62 3 120 5 75 47 30 14 74 95 56 92 57

35 109 23 67 52 10 21 40 25 27 60 103 59 122 37

22 43 100 41 65 77 97 39 29 71 84 106 9 99 11 55
31 116 105 54 69 93 17 68 125 118 114 124 51

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

- Cube 6:

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

- Cube 7:

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

- Cube 8:

2 104 87 106 45 7 112 67 6 93 60 77 111 51 72 41
61 120 102 5 25 69 65 78 115 117 39 20 37 22 62
95 4 81 98 48 94 30 73 38 34 1 109 76 101 63 64 27

26 118 9 110 29 113 92 18 42 66 15 125 32 79 16
59 119 114 107 46 70 24 68 36 49 23 40 8 82 19 14
57 33 116 13 12 53 3 54 21 89 96 17 31 121 100
103 83 90 86 75 99 84 35 47 88 123 50 52 10 124
44 91 58 55 56 43 71 108 105 28 80 97 74 85 122
11

- Cube 9:

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

- Cube 10:

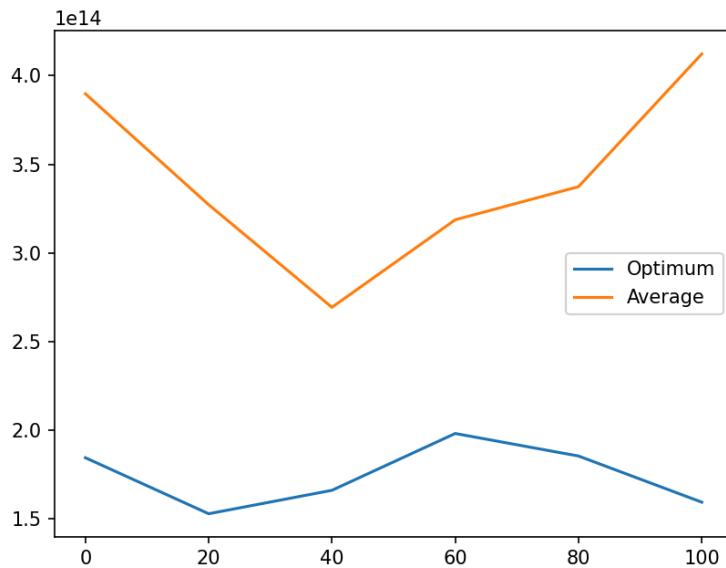
2 87 45 48 6 125 64 108 102 10 44 80 117 57 62 51
67 34 1 109 69 66 7 92 40 29 43 63 42 101 46 56 30
85 99 111 116 12 72 94 93 59 89 32 84 83 19 81 70
113 28 110 11 77 107 35 76 25 90 17 27 91 36 26
106 33 118 13 23 105 22 103 24 98 8 120 115 53 18
37 104 52 82 60 123 9 112 14 15 114 38 124 122 4
58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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 516672300804979
- Fitness Cube 2: 655318302230533
- Fitness Cube 3: 565481299713575
- Fitness Cube 4: 159404594856785
- Fitness Cube 5: 477060517593507

- Fitness Cube 6: 477060517593507
- Fitness Cube 7: 237534362243079
- Fitness Cube 8: 237534362243079
- Fitness Cube 9: 298158059714426
- Fitness Cube 10: 498170563417267
- Average Fitness: 412239488041073
- Optimum: 159404594856785

● Plot:



● Durasi: 1148998 microsekon

■ Sampel 3:

● State Awal:

○ Cube 1:

```

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

```

75 102 103 53 19 44 78 71 111 50 83 94 41 15 34

17

- Cube 2:

40 97 111 94 26 114 89 117 16 3 17 15 46 31 109

18 11 60 29 100 81 44 73 64 125 23 108 2 107 121

51 69 30 1 118 32 47 116 33 82 13 112 12 66 113

77 54 105 25 19 85 78 84 14 83 34 88 106 8 119 74

36 63 96 86 58 99 124 65 45 102 101 120 57 115 38

41 93 28 37 104 49 75 43 70 80 103 6 7 27 20 55 9

59 48 24 56 39 90 50 22 62 61 71 92 35 72 87 98 42

53 76 122 123 52 68 21 79 110 67 91 95 4 5 10

- Cube 3:

49 87 97 47 8 77 101 115 83 78 50 105 116 40 120

16 122 76 25 98 54 23 31 30 84 6 56 48 3 38 11 32

62 110 69 21 52 39 111 20 1 100 45 73 2 12 82 24

109 27 15 67 81 14 118 125 35 53 28 95 113 65 63

117 85 37 70 92 59 58 112 86 5 66 68 61 102 104

64 7 55 99 42 4 22 88 106 114 36 43 51 46 103 19

75 29 71 79 34 44 72 60 119 91 89 93 9 108 80 18

121 96 10 107 17 123 33 13 94 90 26 124 41 74 57

- Cube 4:

124 35 58 95 48 28 100 122 1 59 105 32 30 10 113

103 51 70 8 99 88 13 101 39 14 71 81 79 49 37 104

54 118 4 2 16 60 31 56 7 84 57 69 91 42 43 41 97

47 29 55 11 45 116 34 119 74 23 19 64 33 92 63 53

5 87 102 114 22 66 21 24 18 46 85 121 111 36 117

77 15 12 96 98 3 76 50 108 6 112 67 52 90 20 73 26

75 109 78 86 107 68 17 125 120 38 72 83 9 82 62

123 110 115 89 27 93 40 44 25 80 61 106 65 94

- Cube 5:

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

- Cube 6:

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

- Cube 7:

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

- Cube 8:

47 69 55 108 24 114 97 43 12 83 72 101 70 32 79
100 99 18 49 42 10 118 125 2 16 9 51 7 14 23 87
109 75 94 29 115 80 17 67 33 82 3 1 120 71 22 20
44 65 39 91 107 96 13 50 41 6 110 68 27 64 60 63

38 25 95 66 40 53 8 85 73 124 77 98 88 15 76 84
104 26 4 92 61 59 48 58 35 30 121 57 93 106 89
123 45 62 102 28 122 81 86 52 78 21 105 34 46 5
11 119 117 116 112 19 90 56 31 111 74 54 103 37
36 113

- Cube 9:

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

- Cube 10:

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

- State Akhir:

- Cube 1:

25 61 80 13 85 73 115 35 108 20 17 31 89 98 116
51 124 87 19 82 93 103 8 90 58 15 94 38 79 81 119
45 30 54 102 99 7 123 71 70 83 107 44 67 68 10 97
33 59 11 125 75 66 46 120 9 114 48 1 117 23 112
64 3 29 101 14 52 16 50 26 43 27 91 105 118 56 60
69 5 100 47 96 21 42 36 55 88 37 62 41 76 65 113

12 77 24 95 78 32 72 104 110 34 40 49 63 121 22
122 86 18 57 6 109 92 111 2 106 53 84 4 74 28 39

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

- Cube 6:

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

- Cube 7:

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

- Cube 8:

115 112 103 94 76 24 85 113 61 124 34 91 119 14
43 8 65 7 52 50 62 10 90 51 60 15 89 67 47 97 116
18 20 39 87 82 125 104 9 19 71 2 31 46 27 59 56 32
69 42 110 40 11 6 16 111 96 13 25 109 30 17 84 29

120 58 117 100 44 26 41 38 68 12 48 83 3 35 72 88
36 80 5 49 105 99 75 23 70 102 81 92 4 53 1 73 98
114 79 93 107 123 95 21 64 66 77 108 55 54 101 33
57 78 63 37 74 86 118 121 22 122 106 28 45

- Cube 9:

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

- Cube 10:

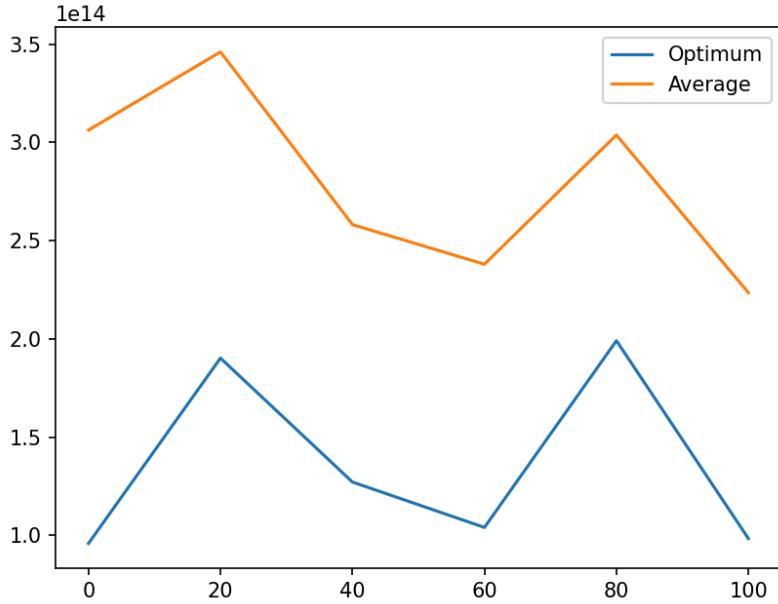
69 96 16 4 55 32 12 95 104 56 17 31 89 29 116 51
124 87 19 82 93 103 8 90 58 15 94 38 79 64 119 22
30 54 102 99 7 123 71 65 83 107 44 67 68 10 42 33
59 78 125 75 11 6 74 101 3 121 34 1 92 112 113 63
24 40 23 52 111 50 117 25 27 72 26 48 57 106 91
105 81 66 98 118 60 21 85 100 18 62 41 86 84 120
46 114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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 98378010008368
- Fitness Cube 2: 98378010008368
- Fitness Cube 3: 356567718267241
- Fitness Cube 4: 356409105172779
- Fitness Cube 5: 151640832689363
- Fitness Cube 6: 151640832689363
- Fitness Cube 7: 199302206011229
- Fitness Cube 8: 199302206011229

- Fitness Cube 9: 397392269227224
- Fitness Cube 10: 225904153376887
- Average Fitness: 223491534346205
- Optimum: 98378010008368

- Plot:



- Durasi: 1158916 microsekon

- Variansi 2:

- Jumlah Populasi: **4**
- Banyak Iterasi: **100**
- Sampel 1:

- State Awal:

- Cube 1:

```

47 105 20 33 73 82 12 83 123 64 43 45 84 54 117 1
4 29 80 28 6 99 26 107 121 22 17 102 14 10 91 32
13 106 7 48 124 77 66 61 96 25 23 9 89 95 74 55 78
46 50 90 71 70 122 2 81 76 87 98 118 113 63 30 58
92 108 24 37 51 40 94 85 72 41 88 112 34 79 59
103 35 42 109 60 101 39 27 38 68 52 19 116 53 5

```

110 86 56 36 57 11 69 16 104 44 100 21 119 18 120
97 62 125 111 65 15 8 75 115 3 67 49 31 114 93

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

- State Akhir:

- Cube 1:

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

- Cube 2:

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

- Cube 3:

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

- Cube 4:

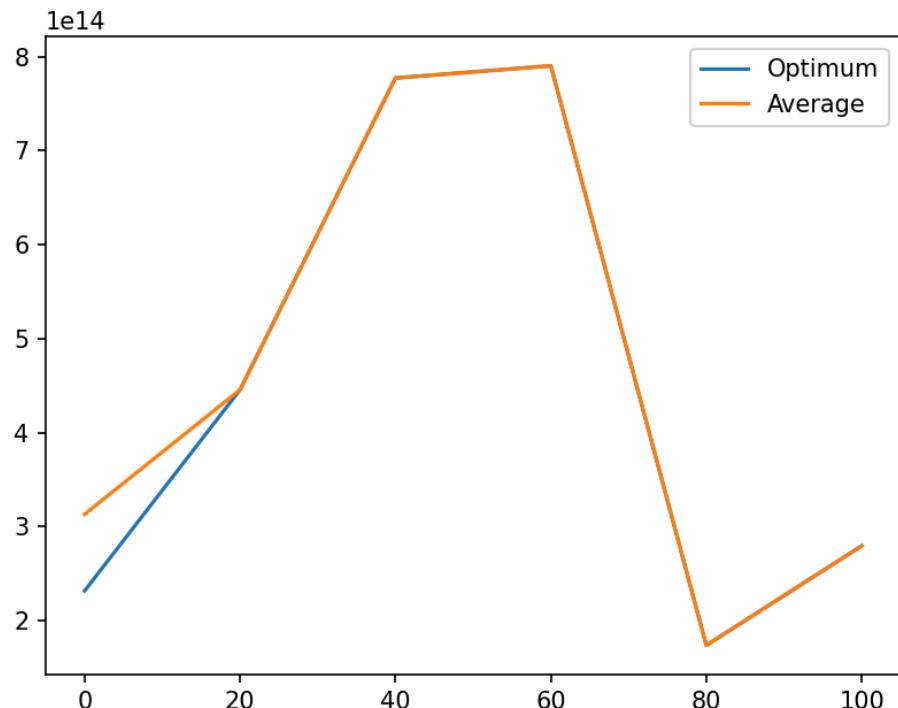
51 125 58 35 23 54 61 42 71 49 50 78 81 107 83
111 85 25 79 94 104 100 124 32 9 41 123 63 6 99 3
39 122 52 22 36 88 109 114 84 82 87 120 73 69 118

45 30 106 121 33 55 92 98 103 105 34 7 2 44 24 17
13 113 15 97 38 16 46 4 65 86 116 119 18 89 77 10
5 93 66 21 74 48 1 59 27 96 64 115 95 68 62 53 67
26 70 102 90 72 31 40 43 11 47 112 12 108 20 19
14 8 76 117 80 56 91 60 57 29 37 110 101 28 75

■ Nilai Objective Function Akhir

- Fitness Cube 1: 279471589745242
- Fitness Cube 2: 279471589745242
- Fitness Cube 3: 279471589745242
- Fitness Cube 4: 279471589745242
- Average Fitness: 279471589745242
- Optimum: 279471589745242

■ Plot:



■ Durasi: 434253 microsekon

■ Sampel 2:

- State Awal:

○ Cube 1:

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

- Cube 2:

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

- Cube 3:

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

- Cube 4:

14 121 22 80 38 90 84 82 70 92 17 61 50 7 81 43 68
27 51 58 4 28 26 75 20 69 105 36 114 56 109 15 29
12 49 124 125 122 118 62 31 64 18 57 107 9 97 65

40 10 33 85 74 16 66 35 91 2 93 96 99 100 63 88 73
101 48 53 25 78 6 32 102 54 76 30 115 94 52 95 39
5 59 55 106 83 111 67 87 108 89 24 104 19 45 79 8
120 72 1 119 86 71 112 123 117 47 44 11 34 23 3
41 46 21 103 77 110 116 60 13 42 113 37 98

- State Akhir:

- Cube 1:

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

- Cube 2:

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

- Cube 3:

97 2 51 120 117 54 64 26 32 14 8 52 71 67 30 1 122
119 20 100 24 53 101 49 55 60 121 109 13 3 83 124
95 65 21 102 81 35 110 11 104 106 125 27 118 114
34 37 25 69 23 43 108 123 66 91 36 107 18 47 79 6
63 80 56 89 86 57 48 72 87 22 16 15 50 98 59 73
113 29 90 111 70 105 84 12 33 41 5 74 94 78 39 45

58 38 68 115 92 10 76 7 103 96 46 93 19 77 17 28

85 62 99 9 88 75 4 44 82 112 40 116 42 61 31

- Cube 4:

97 2 51 120 117 54 64 26 32 14 8 52 71 67 30 1 122

119 20 100 24 53 101 49 55 60 121 109 13 3 83 124

95 65 21 102 81 35 110 11 104 106 125 27 118 114

34 37 25 69 23 43 108 123 66 91 36 107 18 47 79 6

63 80 56 89 86 57 48 72 87 22 16 15 50 98 59 73

113 29 90 111 70 105 84 12 33 41 5 74 94 78 39 45

58 38 68 115 92 10 76 7 103 96 46 93 19 77 17 28

85 62 99 9 88 75 4 44 82 112 40 116 42 61 31

- Nilai Objective Function Akhir:

- Fitness Cube 1: 148226743258066

- Fitness Cube 2: 148226743258066

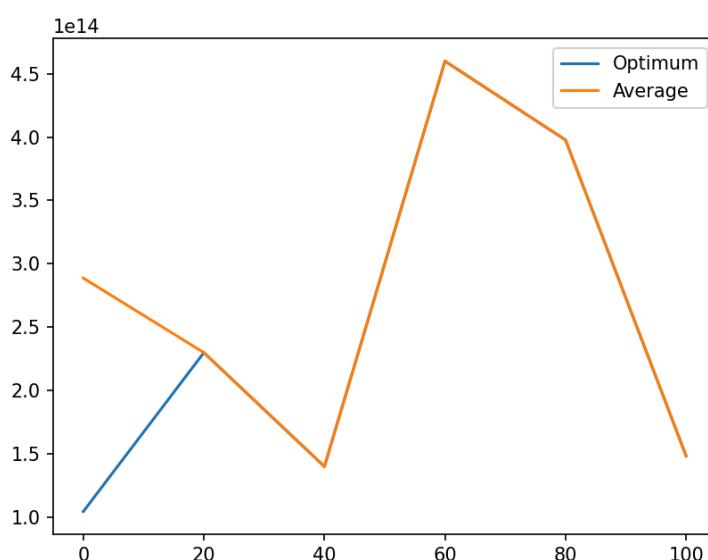
- Fitness Cube 3: 148226743258066

- Fitness Cube 4: 148226743258066

- Average Fitness: 148226743258066

- Optimum: 148226743258066

- Plot:



- Durasi: 476031 microsekon

■ Sampel 3:

- State Awal:

- Cube 1:

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

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

- State Akhir:

- Cube 1:

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

- Cube 2:

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

- Cube 3:

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

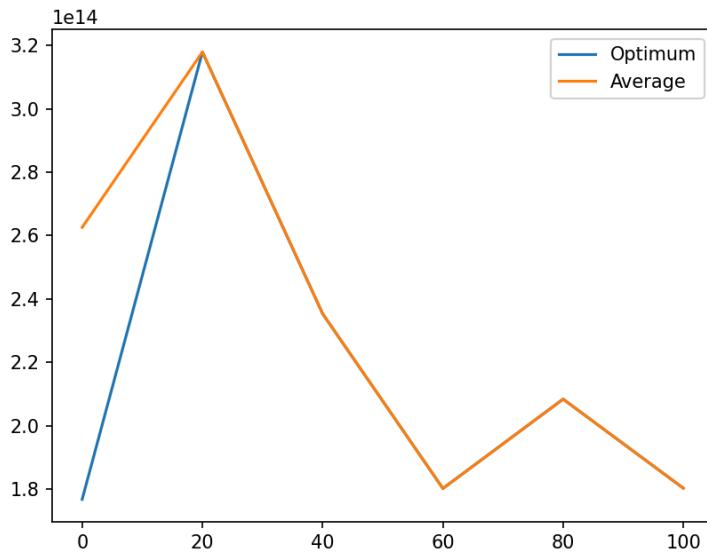
- Cube 4:

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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 180239054037780
- Fitness Cube 2: 180239054037780
- Fitness Cube 3: 180239054037780
- Fitness Cube 4: 180239054037780
- Average Fitness: 180239054037780
- Optimum: 180239054037780

- Plot:



- Durasi: 453207 microsekon
- Variansi 3
 - Jumlah Populasi: **20**
 - Banyak Iterasi: **100**
 - Sampel 1:
 - State Awal:
 - Cube 1:
 - 54 75 78 19 25 93 83 82 92 5 65 112 66 71 77 108
34 117 109 107 1 94 38 62 14 85 125 95 90 91 8 48
22 101 111 104 61 88 72 7 20 110 105 43 29 26 122
120 124 64 39 42 56 32 57 123 40 74 4 12 100 84
63 98 36 76 103 89 2 47 118 21 59 113 6 9 115 73
18 121 86 16 79 24 55 69 44 81 11 15 45 17 23 67
37 27 96 10 31 80 68 50 41 28 33 46 116 49 13 114
53 58 3 52 70 106 30 102 35 87 97 60 99 119 51
 - Cube 2:
 - 4 116 121 84 104 34 107 77 9 20 51 119 76 79 61
14 118 73 81 102 75 110 112 92 55 70 35 108 98
114 23 43 124 7 117 26 3 60 87 25 37 101 59 21 41
48 5 50 12 72 38 42 109 49 19 83 22 56 71 47 2 44

63 125 103 24 95 115 54 39 11 30 88 68 32 6 111
13 27 100 10 64 15 105 78 28 40 85 52 94 120 82
97 123 96 57 29 89 8 33 65 69 86 122 62 58 80 17
16 31 67 36 53 113 1 93 66 106 99 90 91 18 46 74
45

- Cube 3:

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

- Cube 4:

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

- Cube 5:

54 67 1 2 14 96 42 75 121 125 97 122 48 99 34 71
114 88 56 50 4 118 18 100 80 72 47 91 65 110 84
123 7 25 52 5 64 81 22 28 11 115 9 49 69 109 90 89
45 3 35 120 15 27 79 39 104 105 102 92 10 38 63
95 24 112 61 87 44 70 106 51 40 53 26 21 43 86 12
23 36 8 29 33 66 78 57 60 6 98 101 124 119 108

116 74 93 73 31 113 68 30 19 55 111 13 59 83 103
37 85 58 20 76 41 117 17 16 82 107 77 94 62 32 46

- Cube 6:

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

- Cube 7:

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

- Cube 8:

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

- Cube 9:

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

- Cube 10:

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

- Cube 11:

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

- Cube 12:

54 53 124 20 105 93 52 22 46 89 95 36 79 7 41 30
58 120 62 69 29 40 43 78 21 2 72 13 74 92 76 8 66
73 39 109 103 38 84 81 34 4 18 59 37 70 26 64 121
3 114 1 57 51 112 117 67 96 11 83 108 60 63 75

113 122 35 80 118 15 115 19 101 90 85 28 71 106
98 33 17 111 82 88 44 123 65 48 24 6 55 97 116 14
50 10 91 5 61 100 99 68 107 27 16 110 86 94 23 77
87 102 125 31 56 12 104 42 32 45 9 49 47 119 25

- Cube 13:

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

- Cube 14:

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

- Cube 15:

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

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

108 81 119 103 72 123 54 59 31 8 64 69 109 43 4
11 63 73 107 67 89 41 35 46 80 77 88 118 70 95 97
58 44 94 57 34 5 9 42 101 48 12 50 62 71 37 79 56
120 33 25 45 110 51 116 115 18 93 40 61 21 75 26
60 124 83 87 16 22 104 121 76 82 30 13 10 86 85
28

- Cube 20:

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

- State Akhir:

- Cube 1:

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

- Cube 2:

37 9 96 121 57 1 53 33 24 99 13 109 125 69 47 30
45 49 23 81 35 34 79 51 2 95 18 80 25 31 83 113
106 122 74 73 104 115 117 36 54 56 63 116 119 3
85 84 52 60 98 11 20 28 22 12 118 8 108 4 68 88 21
7 62 19 48 70 14 59 114 71 16 105 77 103 123 72

39 100 17 58 26 64 50 43 92 65 97 91 38 78 75 82

94 101 107 10 112 15 90 41 89 61 110 40 6 111 67

27 29 93 32 124 86 55 87 76 66 44 120 46 102 5 42

- Cube 3:

68 119 14 110 121 51 40 71 98 28 77 116 8 22 48

57 27 94 30 11 104 3 122 93 36 44 108 107 96 62

32 89 75 100 43 111 90 59 78 106 37 26 33 47 9 69

91 39 42 53 17 112 54 7 66 82 64 70 25 5 21 20 124

97 113 120 88 102 15 101 92 79 103 50 23 29 117

19 55 35 58 81 38 118 80 123 83 86 61 105 12 49

76 16 52 114 67 72 34 6 31 84 56 60 18 13 125 109

1 74 41 99 45 73 87 115 95 2 24 46 85 4 63 65 10

- Cube 4:

81 91 122 87 77 85 28 108 98 40 55 123 115 57 29

121 110 111 72 119 63 86 27 25 14 62 125 75 71 36

114 43 54 58 101 1 69 68 56 103 2 53 50 64 21 30

52 35 74 124 97 113 118 39 18 10 37 9 49 120 26

93 5 88 17 47 20 34 116 94 13 84 8 73 48 109 66 61

95 100 70 45 38 96 15 117 19 31 11 90 32 82 99 16

89 41 112 80 51 107 83 92 6 79 4 102 44 42 65 23

104 105 59 60 106 22 12 78 46 76 33 24 67 3 7

- Cube 5:

99 26 104 123 78 65 38 122 124 109 57 91 52 110

27 55 103 14 18 75 111 32 56 44 20 93 70 33 85 69

105 74 31 2 23 107 92 84 36 76 51 48 6 73 13 100

19 7 125 87 12 117 60 71 77 9 115 29 98 79 97 49

116 16 15 17 1 21 88 106 96 101 59 35 86 34 58 30

108 68 43 113 54 25 102 82 63 28 8 47 119 24 41 3

80 121 5 11 61 10 45 40 95 64 120 72 53 114 37 66

39 94 83 89 22 118 50 4 62 42 112 90 46 67 81

- Cube 6:

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

- Cube 7:

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

- Cube 8:

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

- Cube 9:

83 20 65 118 43 93 18 37 82 1 15 33 39 77 63 90 56
4 52 91 30 62 11 5 104 112 31 124 53 86 125 64 3
122 51 57 72 8 38 76 87 100 7 96 28 32 59 78 41 60

40 98 99 109 21 16 89 36 103 9 85 69 114 71 22
101 111 115 94 123 61 88 23 49 25 92 45 50 95 19
119 35 6 44 2 120 116 75 66 74 48 108 29 17 81 14
13 106 113 34 97 67 79 12 55 68 107 46 58 42 47
84 80 26 121 54 70 24 73 10 105 27 102 110 117

- Cube 10:

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

- Cube 11:

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

- Cube 12:

37 9 24 99 66 8 58 98 52 79 31 33 122 73 20 100 12
77 39 108 80 16 46 6 23 105 45 50 124 17 112 81
107 74 87 68 28 44 1 36 2 119 101 64 60 34 94 83
88 30 21 25 93 19 110 62 125 89 76 106 59 121 47
103 71 90 63 111 26 69 56 54 11 49 55 82 109 91
38 78 70 118 42 104 27 123 13 61 53 43 40 114 97

120 117 75 65 116 96 41 18 67 85 22 10 35 4 92 72
57 3 51 14 102 84 86 48 113 15 32 29 5 115 95 7

- Cube 13:

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

- Cube 14:

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

- Cube 15:

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

- Cube 16:

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

- Cube 17:

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

- Cube 18:

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

- Cube 19:

96 83 44 26 98 79 51 102 16 123 78 57 88 52 38 73
64 124 43 106 21 81 18 25 32 125 95 60 62 14 110
104 31 89 10 116 118 68 117 76 8 47 37 119 59 19

48 12 108 3 80 105 122 15 53 5 90 85 11 33 4 72 91
107 36 77 41 22 93 61 56 54 23 92 20 45 1 17 40 71
6 30 34 63 7 75 94 115 82 65 66 101 97 2 13 50 55
109 46 114 28 27 121 111 39 29 103 70 35 58 67 24
100 120 74 99 84 69 87 112 86 49 113 42 9

- Cube 20:

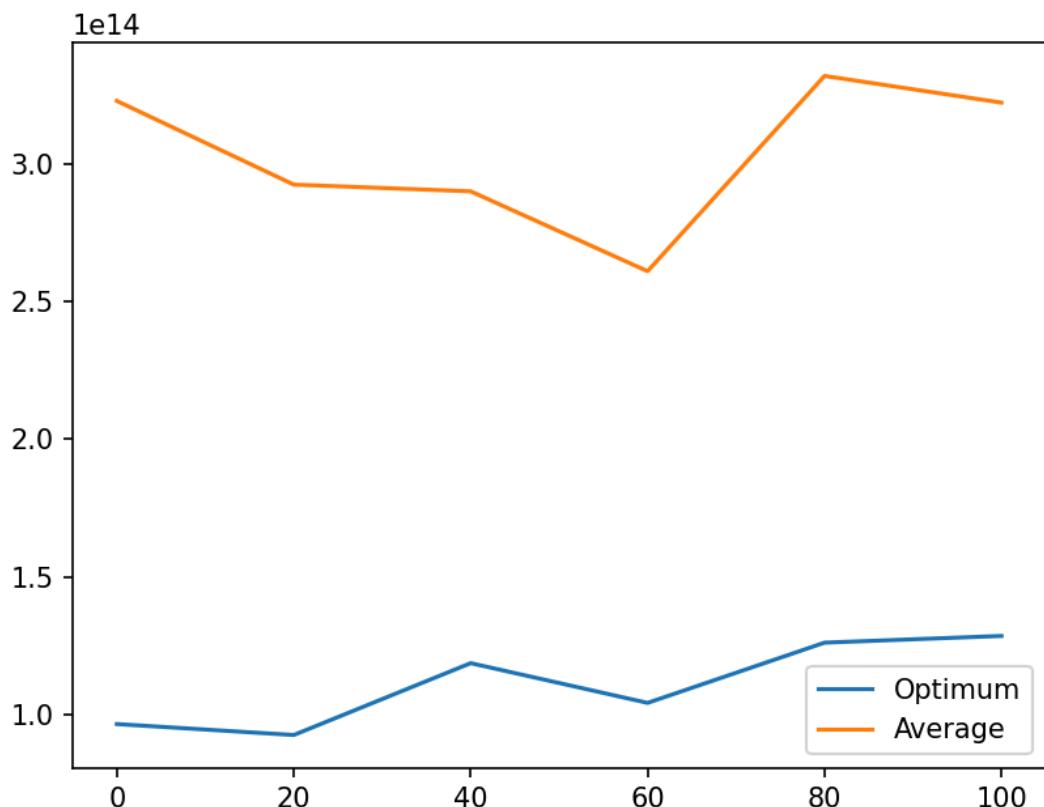
104 88 114 122 106 69 48 78 101 54 50 13 1 56 125
20 37 16 7 107 11 45 49 72 10 28 44 82 27 60 25 29
65 111 31 18 117 97 17 76 119 61 120 63 38 53 62
39 109 40 5 103 33 3 9 108 98 6 21 51 86 14 100 19
91 35 75 85 23 32 22 42 80 2 113 95 84 26 115 79
87 71 94 52 66 41 89 105 92 112 67 110 123 70 93
74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

■ Nilai Objective Function Akhir:

- Fitness Cube 1: 128709870682117
- Fitness Cube 2: 272175318076699
- Fitness Cube 3: 323026136045687
- Fitness Cube 4: 380263567924976
- Fitness Cube 5: 387357757936627
- Fitness Cube 6: 201966142652799
- Fitness Cube 7: 272082037292742
- Fitness Cube 8: 217872234421462
- Fitness Cube 9: 410414288671860
- Fitness Cube 10: 260394602112221
- Fitness Cube 11: 372635757657933
- Fitness Cube 12: 690696764083455
- Fitness Cube 13: 286813736519395
- Fitness Cube 14: 146712159718927
- Fitness Cube 15: 353923433421946
- Fitness Cube 16: 329288329198367

- Fitness Cube 17: 154288147107640
- Fitness Cube 18: 534537083446713
- Fitness Cube 19: 257512982499186
- Fitness Cube 20: 460905659367227
- Average Fitness: 322078800441898
- Optimum: 128709870682117

■ Plot:



- Durasi: 2211867 microsekon
- Sampel 2:
- State Awal:
 - Cube 1:


```
74 36 27 31 52 5 22 99 9 35 85 20 61 107 120 60 48
          112 72 92 45 68 65 103 4 81 75 125 53 78 29 106
          83 104 58 10 95 98 49 123 94 37 6 15 34 116 102
          113 44 87 2 124 91 62 89 51 13 70 88 101 80 82 63
          79 108 43 47 38 1 105 18 39 100 50 55 19 121 26
```

97 64 76 32 40 54 118 96 24 16 110 56 57 7 77 73
21 69 17 41 117 33 30 14 122 86 93 23 28 46 12
109 11 8 114 66 71 84 119 3 67 25 115 59 90 42
111

- Cube 2:

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

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

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

31 91 60 58 97 74 57 82 40 69 27 62 11 111 22 99
48 19 125 24 5 114 13 68 81 52 77 6 103 63 8 53
106 76 12 39 108 45 30 105 61 101 107 46 23 80 26
120 90 20 44 113 89 117 56 37 83 16 124 116 4 18
96 79 110 102 95 73 33 122 84 17 71 49 66 119 34
85 94 115 54 32 92 98 65 41 43 104 1 38 118 10

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

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

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

- Cube 20:

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

- State Akhir:

- Cube 1:

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

- Cube 2:

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

- Cube 3:

57 16 45 80 102 62 23 31 41 72 11 65 56 63 90 91 3
78 13 24 50 84 74 104 109 61 122 15 116 18 64 19

94 123 4 12 93 100 36 95 92 97 110 121 27 34 25
87 108 42 28 17 111 58 26 76 106 75 60 77 43 14
101 6 89 40 69 29 32 37 88 114 70 1 30 71 59 107
98 44 48 81 96 21 51 117 10 86 9 103 115 8 5 46 52
7 55 112 124 54 105 20 83 47 79 85 49 66 120 35
73 39 118 82 113 22 68 67 99 38 53 119 125 2 33

- Cube 4:

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

- Cube 5:

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

- Cube 6:

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

- Cube 7:

70 112 97 37 20 27 12 3 123 93 31 122 85 78 33
106 109 32 110 55 125 76 49 26 118 50 115 113 69
29 86 91 17 92 11 22 59 6 56 8 24 71 1 25 105 74
44 63 14 107 34 88 16 19 54 119 114 60 64 21 57
67 43 116 41 73 52 15 83 35 30 40 5 75 48 62 61 7
94 87 45 77 117 65 90 51 95 79 66 89 18 9 81 10 28

4 47 39 102 104 38 13 103 23 82 98 120 53 58 99
101 84 68 96 124 42 111 108 72 2 80 46 100 36 121

- Cube 8:

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

- Cube 9:

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

- Cube 10:

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

- Cube 11:

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

- Cube 12:

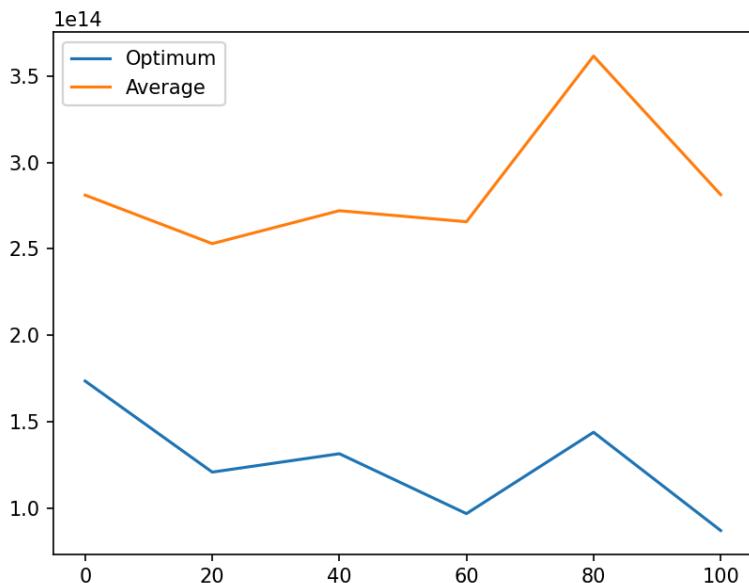
65 27 13 23 93 85 106 97 32 110 52 39 26 122 115
113 34 91 17 92 22 125 87 8 6 38 1 18 33 107 84 19
105 114 75 103 43 28 60 118 51 11 67 68 30 2 83
41 20 70 16 74 123 77 7 3 31 109 57 63 101 5 73 66
62 55 59 44 71 82 45 24 10 21 48 56 54 69 108 100
37 79 94 50 35 29 40 36 111 117 4 112 42 14 64 96
12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

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

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

- Nilai Objective Function Akhir:
 - Fitness Cube 1: 169912515053304
 - Fitness Cube 2: 448471808953192
 - Fitness Cube 3: 290829753112149
 - Fitness Cube 4: 358791273247460
 - Fitness Cube 5: 213859999198790
 - Fitness Cube 6: 243326056593045
 - Fitness Cube 7: 199135137847544
 - Fitness Cube 8: 252863141334745
 - Fitness Cube 9: 362546734595505
 - Fitness Cube 10: 181610254325268
 - Fitness Cube 11: 349485152227757
 - Fitness Cube 12: 450836402531986
 - Fitness Cube 13: 515351981839962
 - Fitness Cube 14: 283318366279248
 - Fitness Cube 15: 282771982353838
 - Fitness Cube 16: 87089141330096
 - Fitness Cube 17: 111515518786379
 - Fitness Cube 18: 111515518786379
 - Fitness Cube 19: 264044382342267
 - Fitness Cube 20: 449186987554448
 - Average Fitness: 281323105414668
 - Optimum: 87089141330096

- Plot:



- Durasi: 2135925 microsekon
- Sampel 3:

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

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

- Cube 6:

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

- Cube 7:

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

- Cube 8:

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

- Cube 9:

47 116 39 91 53 94 99 10 26 29 54 104 92 97 113
106 20 119 56 41 3 88 50 65 81 82 16 83 44 33 46
48 107 6 43 28 21 76 84 110 64 108 68 71 96 57 58
5 95 112 80 14 101 55 120 118 13 51 75 66 87 60

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

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

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

30 87 49 5 51 1 15 101 6 113 38 106 69 41 66 29 75
120 93 62 77 124 98 100 57 44 96 111 33 22 110 17
32 12 46 115 82 13 85 31 23 72 68 97 56 61 48 36
123 83 9 105 24 73 70 3 116 108 104 16 8 53 88 25
119 103 89 27 64 2 95 118 125 86 59 19 39

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

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

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

- Cube 12:

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

- Cube 13:

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

- Cube 14:

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

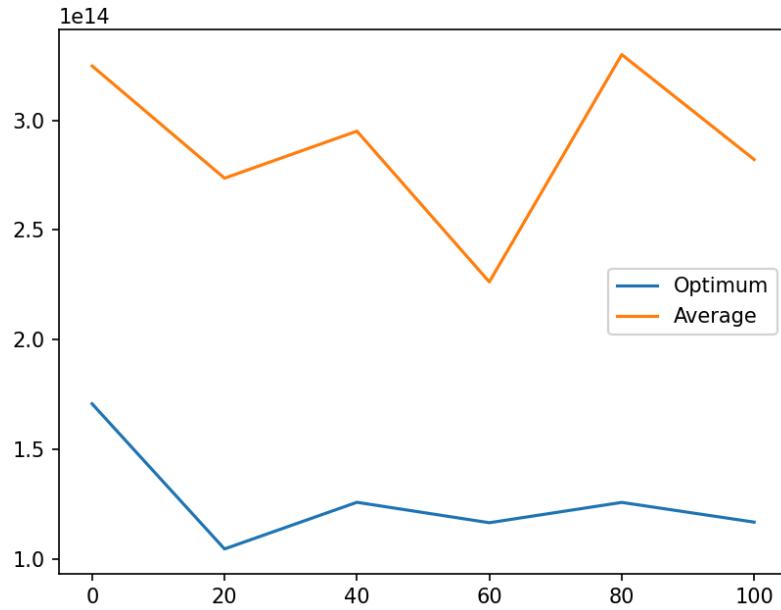
- Cube 15:

2 18 110 28 60 68 112 47 15 81 66 113 46 36 93
106 71 19 56 116 59 91 32 27 86 11 42 52 78 41
108 39 97 98 20 31 10 12 17 14 51 16 69 120 124
23 119 72 53 7 37 92 104 50 54 90 103 73 70 58 80

102 24 43 123 107 35 82 114 94 64 49 48 30 3 83
40 95 29 122 111 76 121 115 1 84 85 33 87 89 45
74 9 8 44 25 62 125 6 77 26 75 13 63 57 4 22 88
109 5 105 117 79 118 101 99 100 55 65 67 96 38 21
34 61

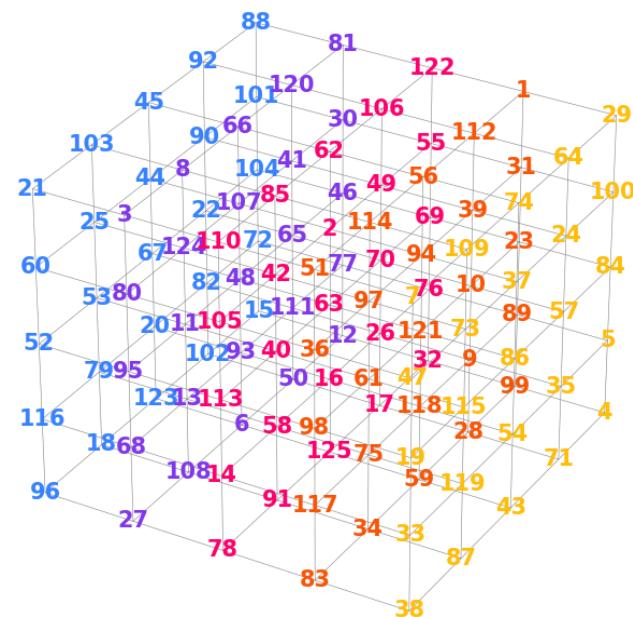
- Cube 16:
40 36 18 114 23 28 93 97 73 58 39 45 32 7 22 24 75
33 5 77 84 88 105 1 42 15 86 26 12 95 31 113 101
98 102 44 115 8 21 3 85 118 19 41 60 38 51 72 104
87 111 11 100 67 121 52 64 91 90 74 46 108 6 81
35 96 29 4 109 122 10 117 57 83 76 125 106 65 119
92 94 89 53 82 62 59 37 80 123 120 14 68 27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

- 8 113 117 36 40 72 100 62 42 10 33 3 7 78 45 85 64
84 5 98 109 24 23 31 101 94 1 34 6 80 57 102
- Cube 20:
108 9 116 37 121 85 12 77 97 56 65 18 36 83 84 17
98 76 14 112 26 81 87 8 60 123 53 38 90 92 114 74
21 79 73 118 103 44 64 1 40 62 75 15 31 55 69 109
105 51 91 13 115 27 28 19 68 72 16 82 101 20 100
5 41 30 3 120 102 106 2 10 110 52 57 4 88 7 67 58
47 70 104 119 117 95 80 11 32 125 89 35 34 124 54
59 78 96 49 113 86 46 33 6 24 45 42 122 50 71 63
107 43 66 48 29 93 94 25 22 99 23 61 39 111
- Nilai Objective Function Akhir:
 - Fitness Cube 1: 348571611740895
 - Fitness Cube 2: 383874108798380
 - Fitness Cube 3: 411117002855953
 - Fitness Cube 4: 171653741937023
 - Fitness Cube 5: 235295715198205
 - Fitness Cube 6: 346762357978231
 - Fitness Cube 7: 172157170211854
 - Fitness Cube 8: 279538918144821
 - Fitness Cube 9: 165336889705743
 - Fitness Cube 10: 116887687825357
 - Fitness Cube 11: 351195652687444
 - Fitness Cube 12: 564836399344154
 - Fitness Cube 13: 162118498520796
 - Fitness Cube 14: 312874332489237
 - Fitness Cube 15: 400758141469667
 - Fitness Cube 16: 220361975577872
 - Fitness Cube 17: 134255155429047
 - Fitness Cube 18: 372241046143820
 - Fitness Cube 19: 152785886005726
 - Fitness Cube 20: 339828176784057
 - Average Fitness: 282122523442414
 - Optimum: 116887687825357
- Plot:



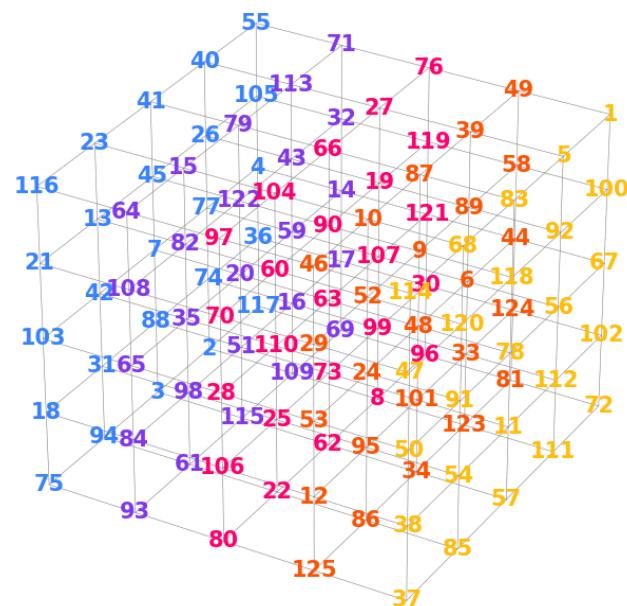
- Durasi: 2599661 microsekron
- Eksperimen Jumlah Iterasi
 - Variansi 1
 - Jumlah Populasi: **10**
 - Banyak Iterasi: **100**
 - Sampel 1:
 - State Awal:
 - Cube 1:

Start Population Cube 1:



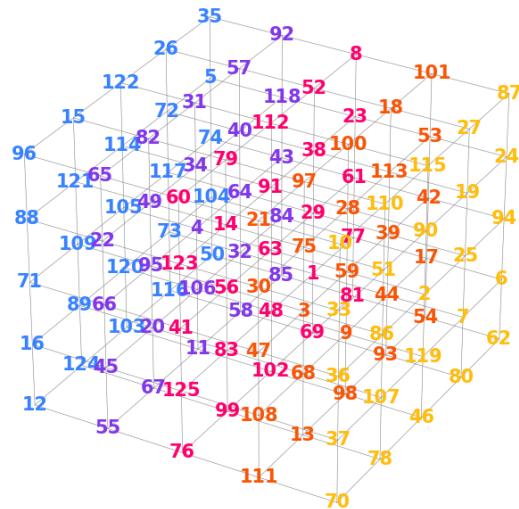
• Cube 2:

Start Population Cube 2:



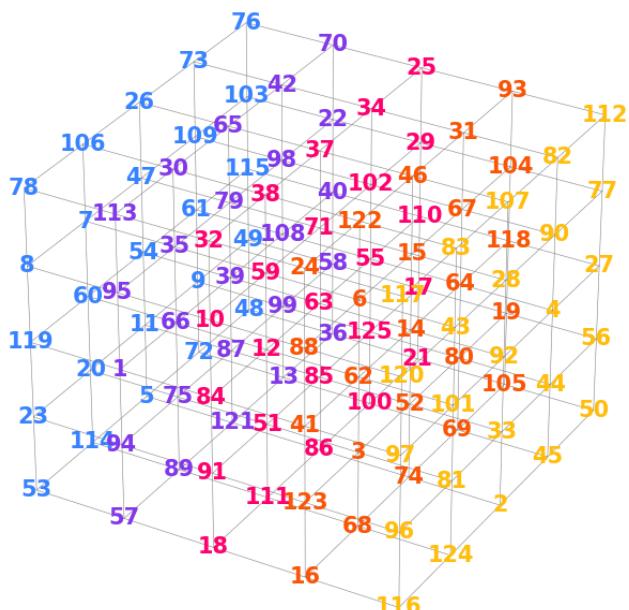
- Cube 3:

Start Population Cube 3:



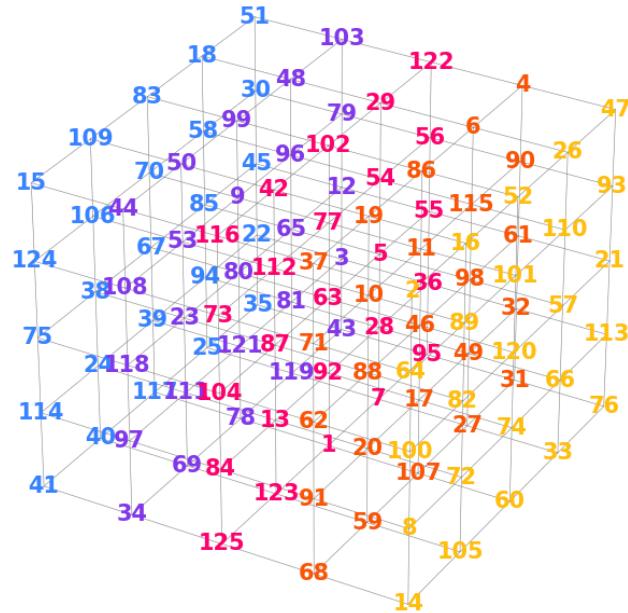
- Cube 4:

Start Population Cube 4:



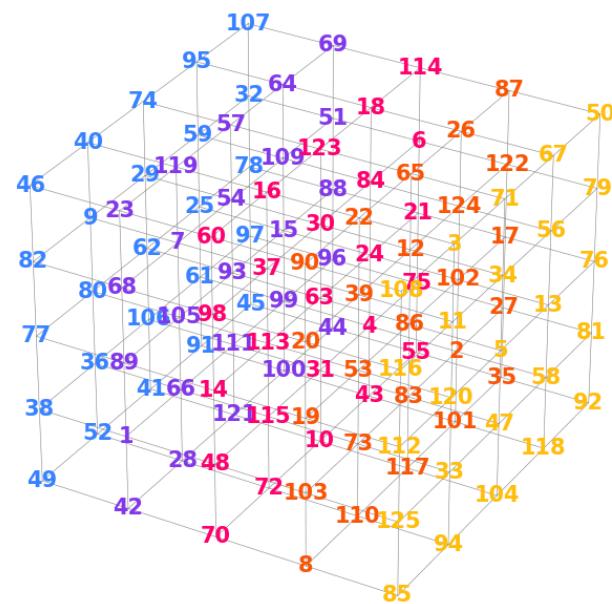
- Cube 5:

Start Population Cube 5:



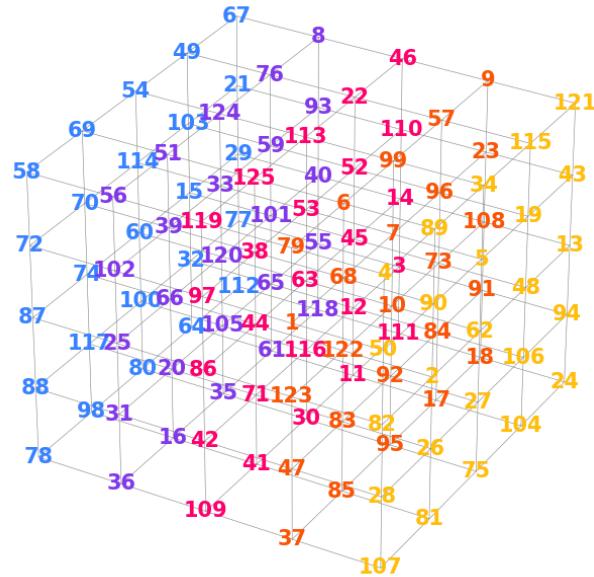
• Cube 6:

Start Population Cube 6:



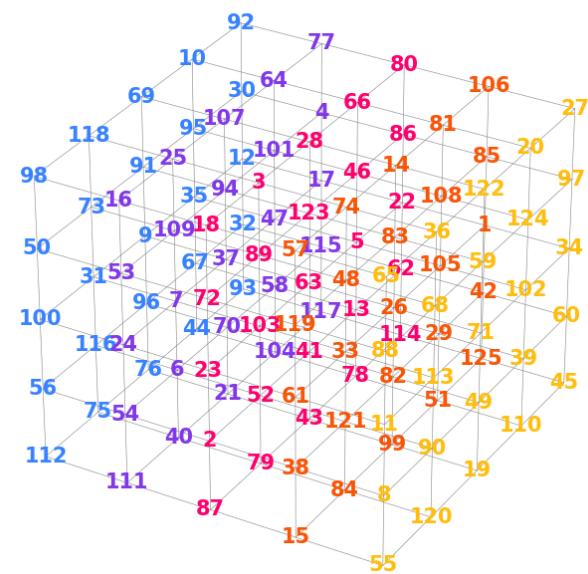
- Cube 7:

Start Population Cube 7:



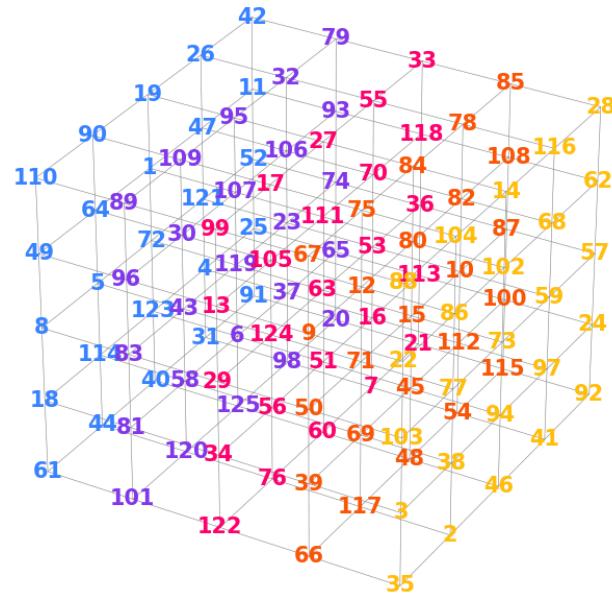
- Cube 8:

Start Population Cube 8:



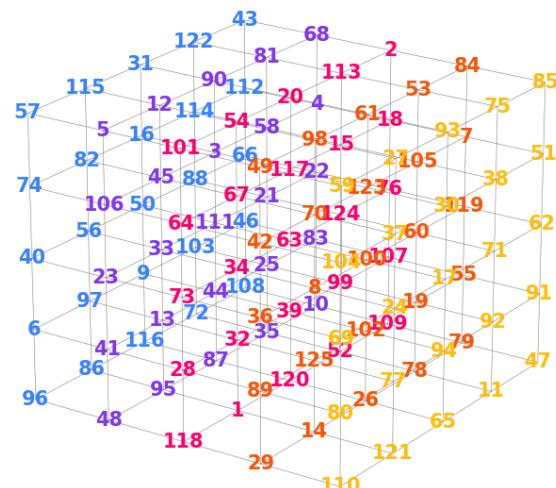
- Cube 9:

Start Population Cube 9:



- Cube 10:

Start Population Cube 10:

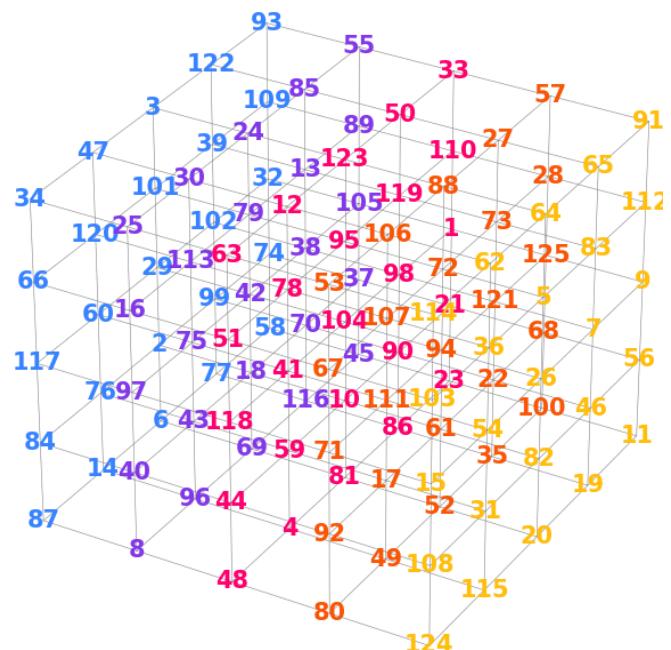


- Nilai Objective Function Awal:

- Fitness Cube 1: 263560646550665

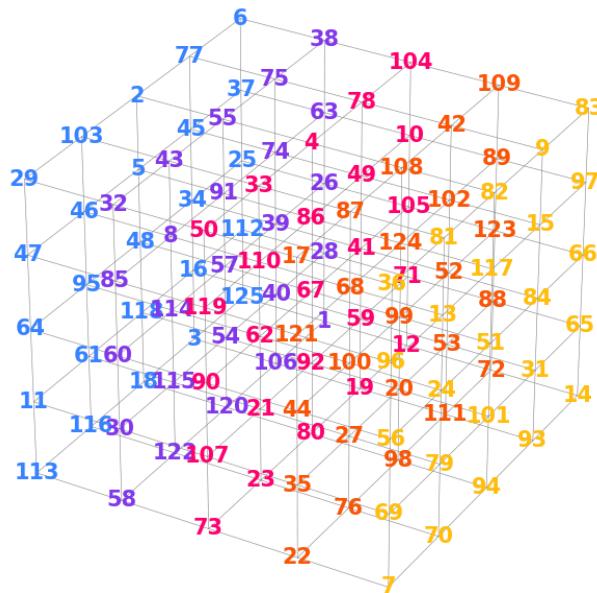
- Fitness Cube 2: 236023422769202
 - Fitness Cube 3: 121841637860615
 - Fitness Cube 4: 284678844305931
 - Fitness Cube 5: 298789572963916
 - Fitness Cube 6: 280154047305725
 - Fitness Cube 7: 183183771262428
 - Fitness Cube 8: 404310869968088
 - Fitness Cube 9: 348104434359665
 - Fitness Cube 10: 170822492435573
 - Average Fitness: 259146973978180
 - Optimum: 121841637860615
- State Akhir:
- Cube 1:

End Population Cube 1:



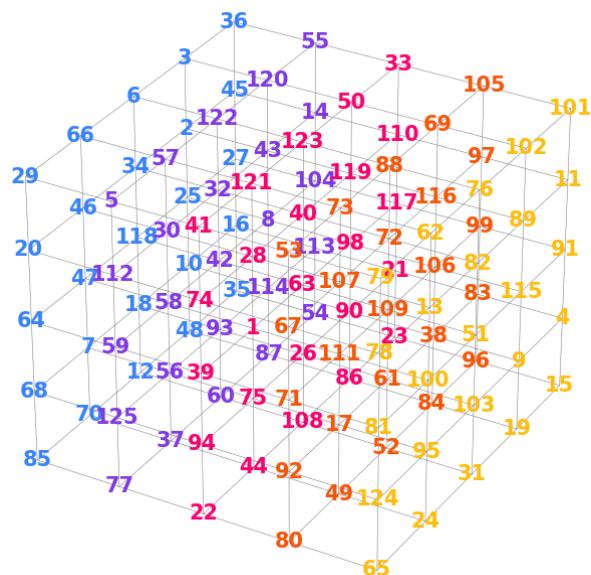
- Cube 2:

End Population Cube 2:



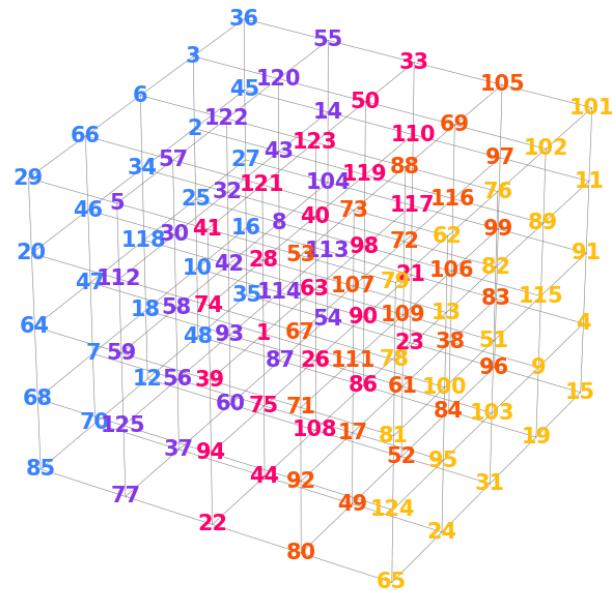
- Cube 3:

End Population Cube 3:



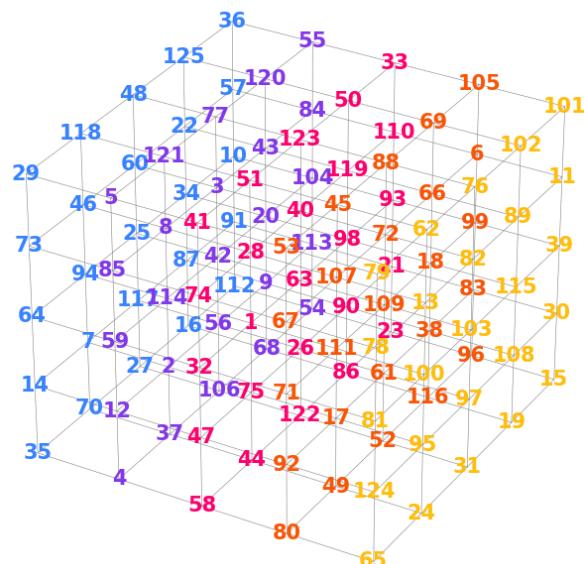
- Cube 4:

End Population Cube 4:



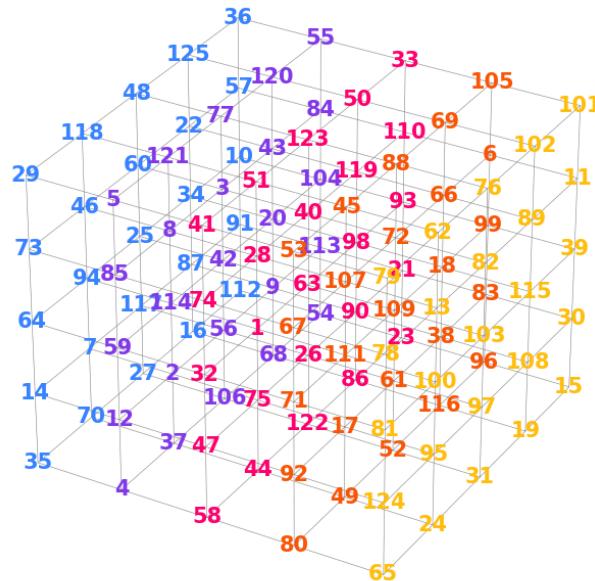
- Cube 5:

End Population Cube 5:



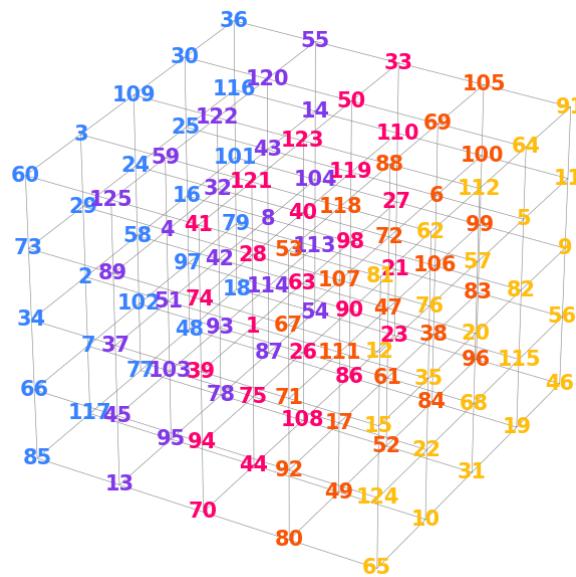
- Cube 6:

End Population Cube 6:



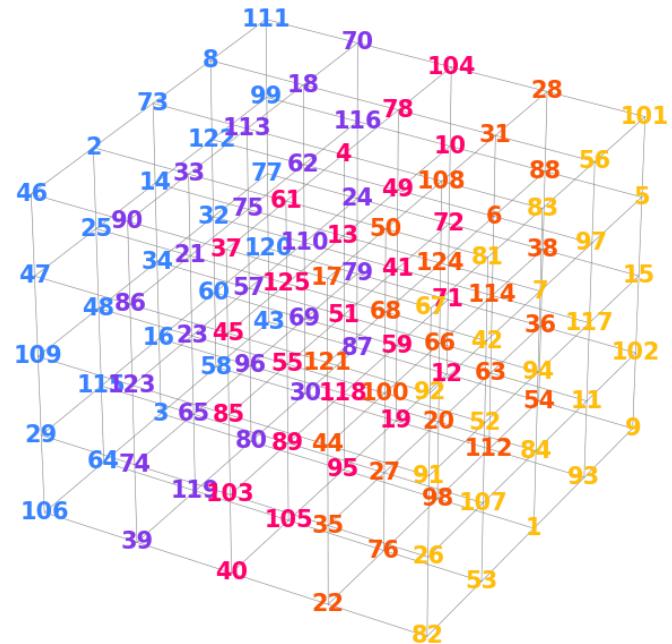
- Cube 7:

End Population Cube 7:



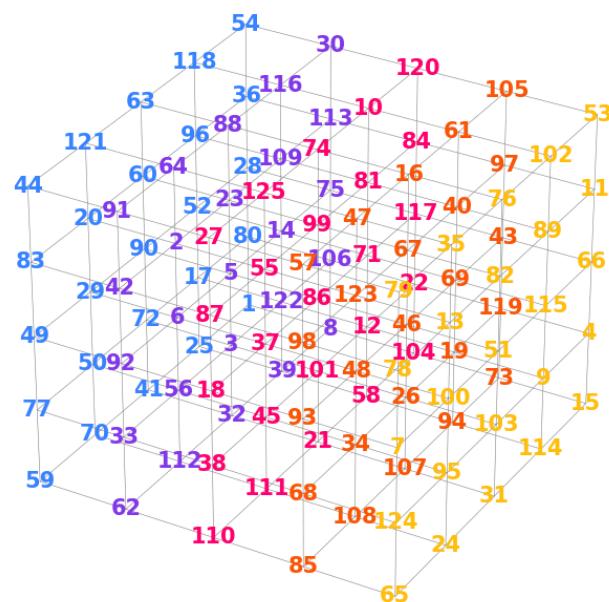
- Cube 8:

End Population Cube 8:



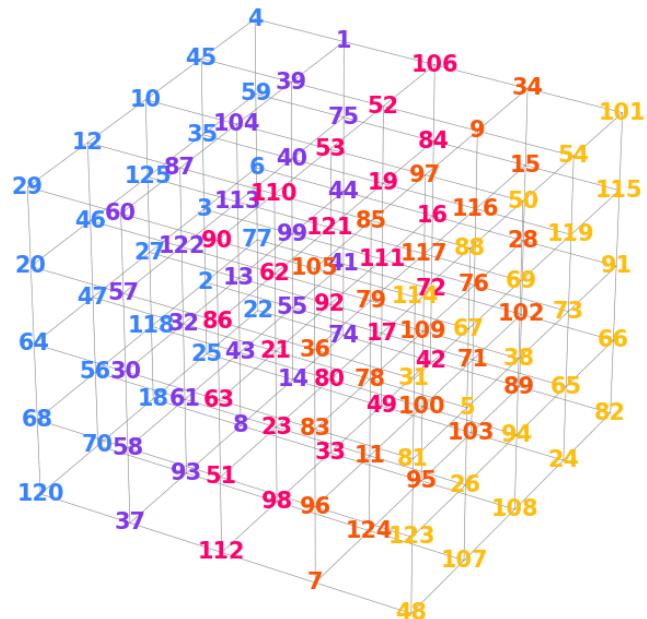
- Cube 9:

End Population Cube 9:



- Cube 10:

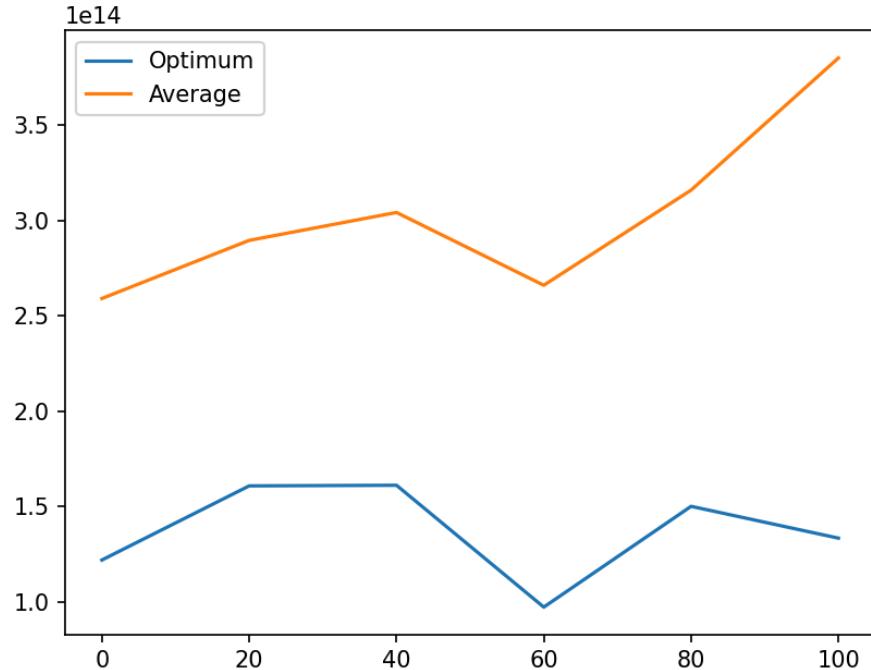
End Population Cube 10:



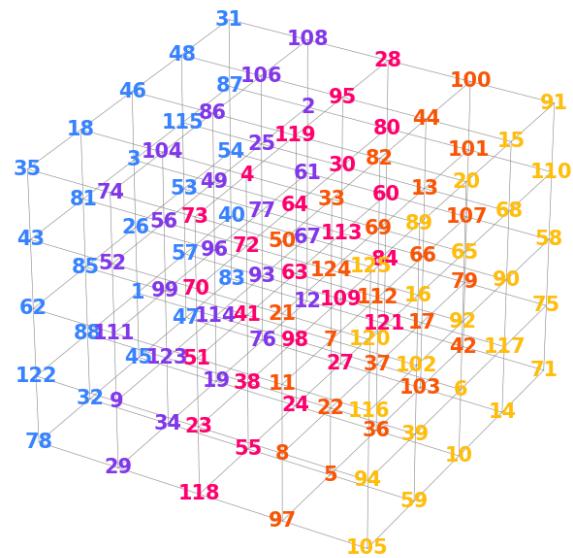
■ Nilai Objective Function Akhir:

- Fitness Cube 1: 275424005145169
- Fitness Cube 2: 253279970867903
- Fitness Cube 3: 542516936512125
- Fitness Cube 4: 542516936512125
- Fitness Cube 5: 254018007287458
- Fitness Cube 6: 254018007287458
- Fitness Cube 7: 826018210901531
- Fitness Cube 8: 168985305593638
- Fitness Cube 9: 133329287303128
- Fitness Cube 10: 603649530697511
- Average Fitness: 385375619810804
- Optimum: 133329287303128

■ Plot:

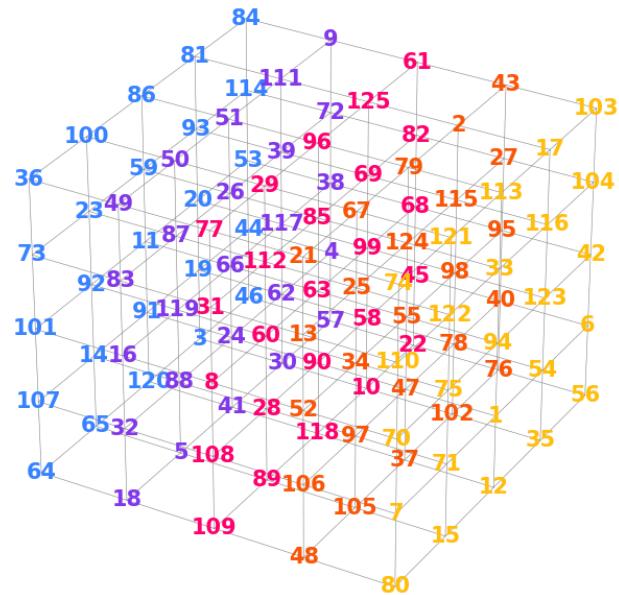


- Durasi: 1019070 microsekon
 - Sampel 2:
 - State Awal:
 - Cube 1:
- Start Population Cube 1:



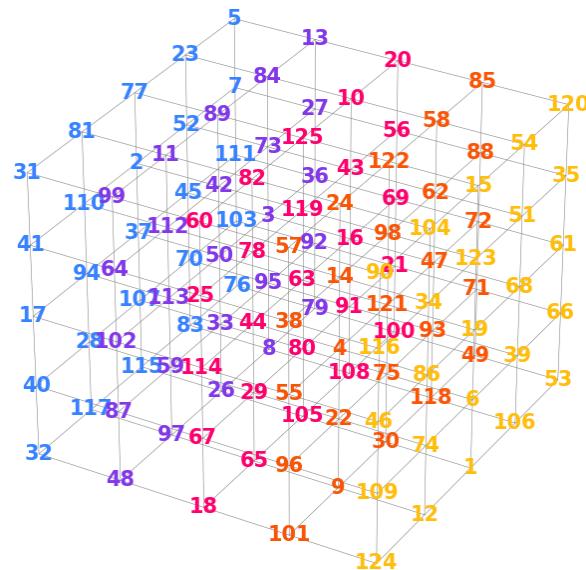
○ Cube 2:

Start Population Cube 2:



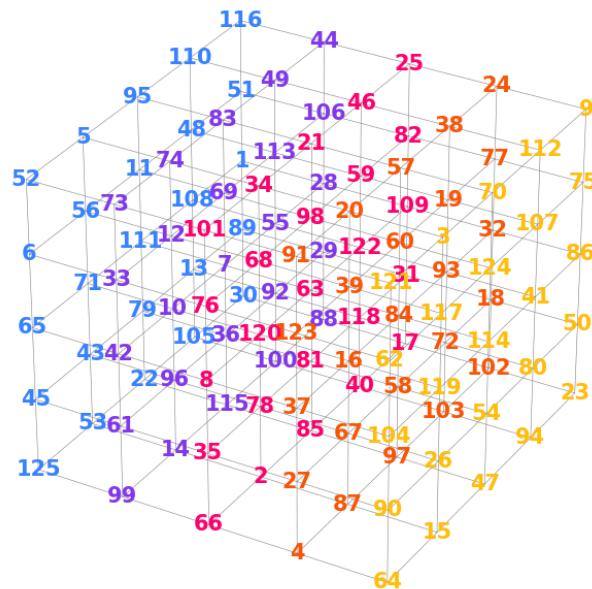
○ Cube 3:

Start Population Cube 3:



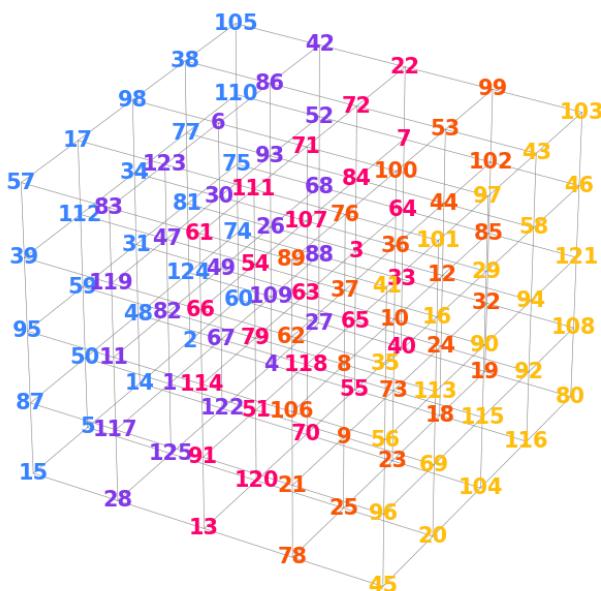
○ Cube 4:

Start Population Cube 4:



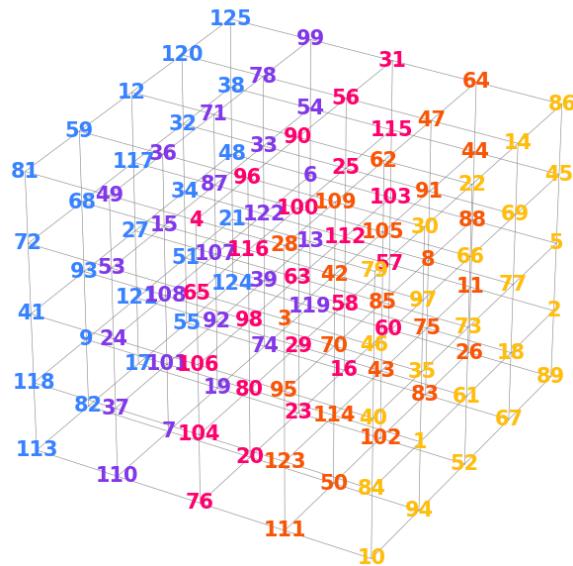
○ Cube 5:

Start Population Cube 5:



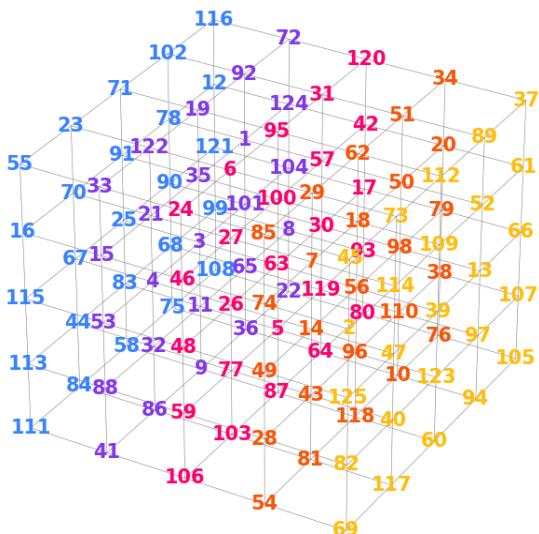
○ Cube 6:

Start Population Cube 6:

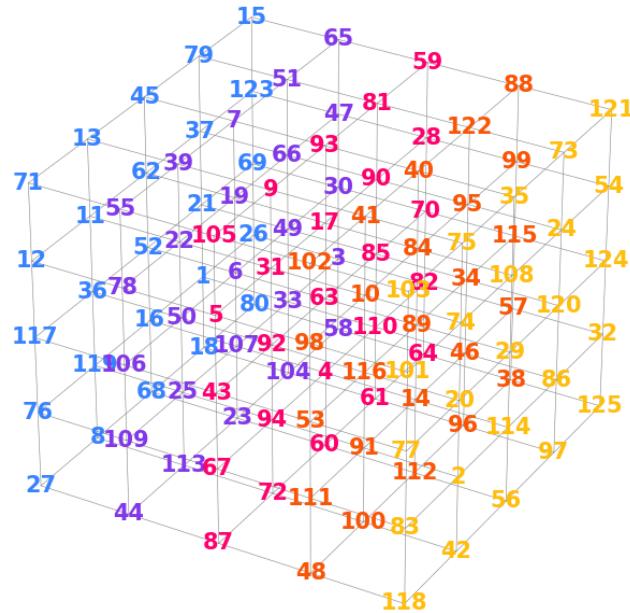


○ Cube 7:

Start Population Cube 7:

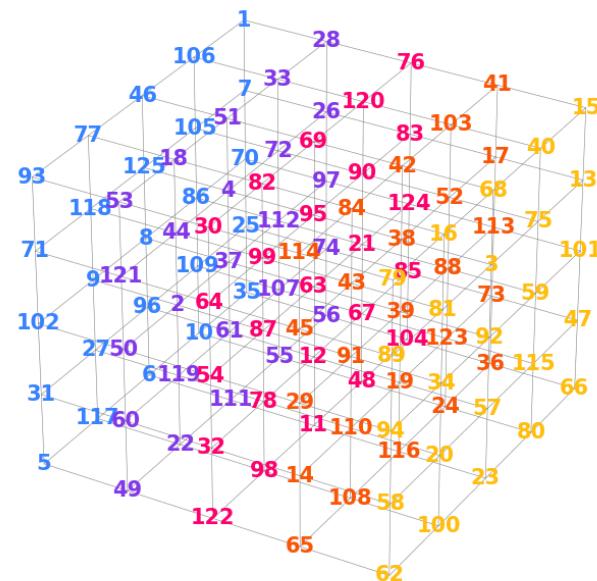


Start Population Cube 8:



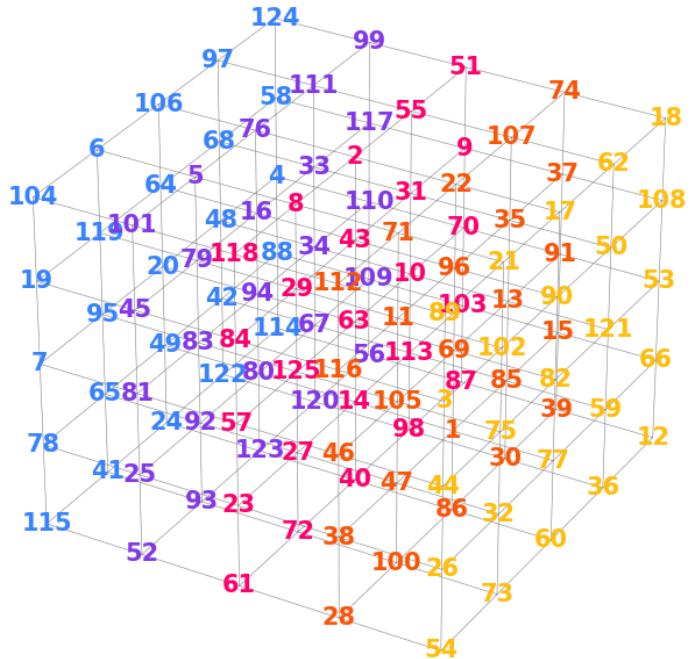
○ Cube 9:

Start Population Cube 9:



- Cube 10:

Start Population Cube 10:



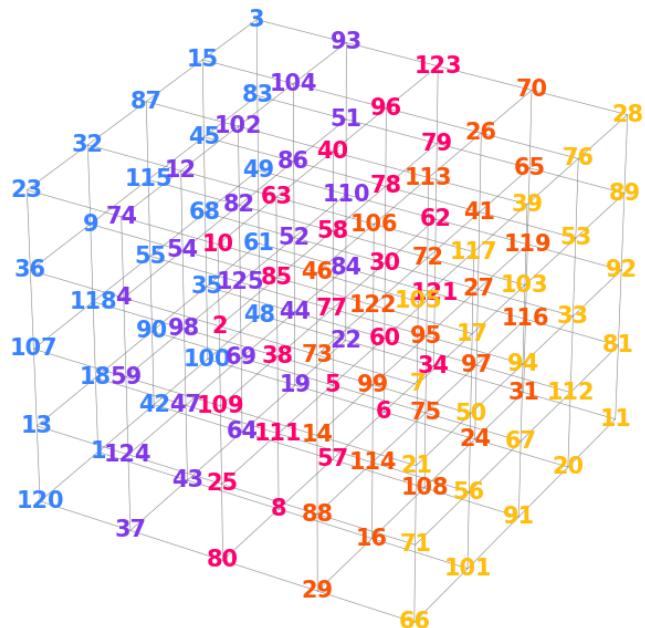
- Nilai Objective Function Awal:

- Fitness Cube 1: 434595791046297
- Fitness Cube 2: 206740039873117
- Fitness Cube 3: 344067342667166
- Fitness Cube 4: 229167141990339
- Fitness Cube 5: 290450923719563
- Fitness Cube 6: 190762439146451
- Fitness Cube 7: 363698248988346
- Fitness Cube 8: 149232639075747
- Fitness Cube 9: 299116559485115
- Fitness Cube 10: 262131876117322
- Average Fitness: 276996300210946
- Optimum: 149232639075747

- State Akhir:

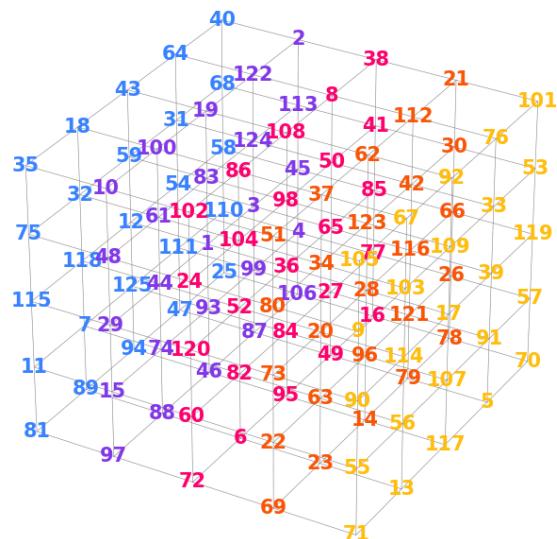
○ Cube 1:

End Population Cube 1:



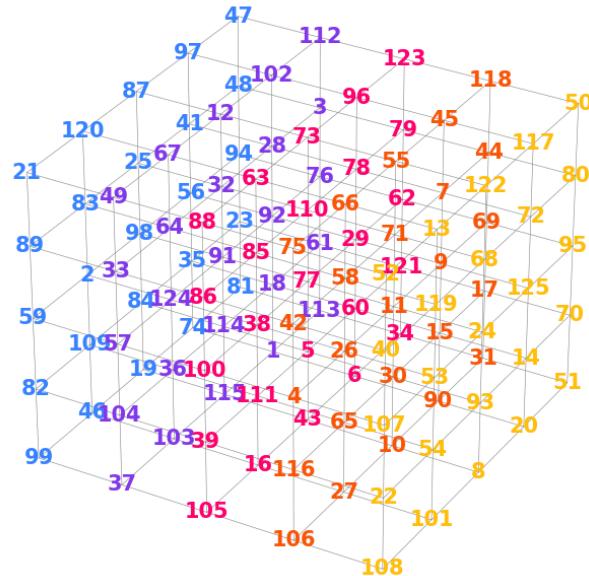
○ Cube 2:

End Population Cube 2:



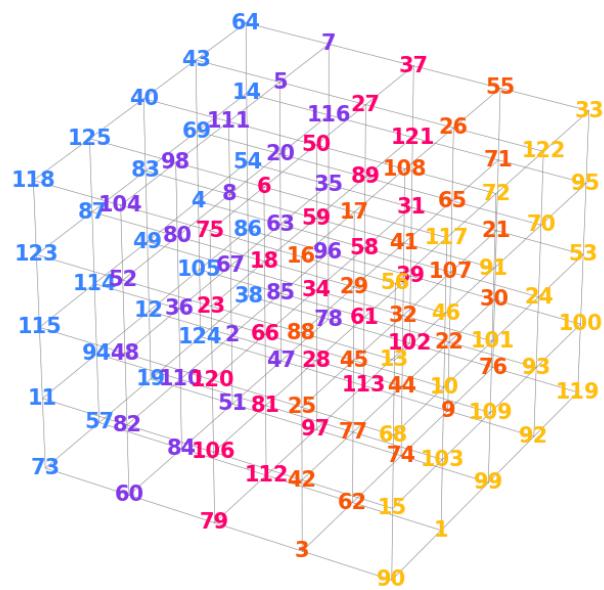
○ Cube 3:

End Population Cube 3:



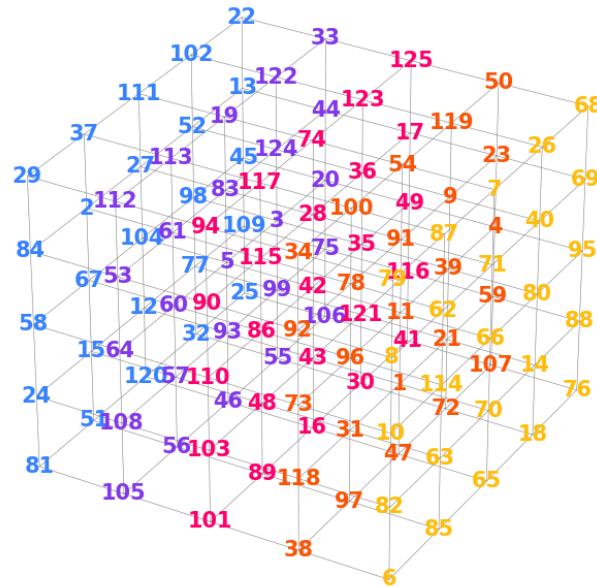
○ Cube 4:

End Population Cube 4:



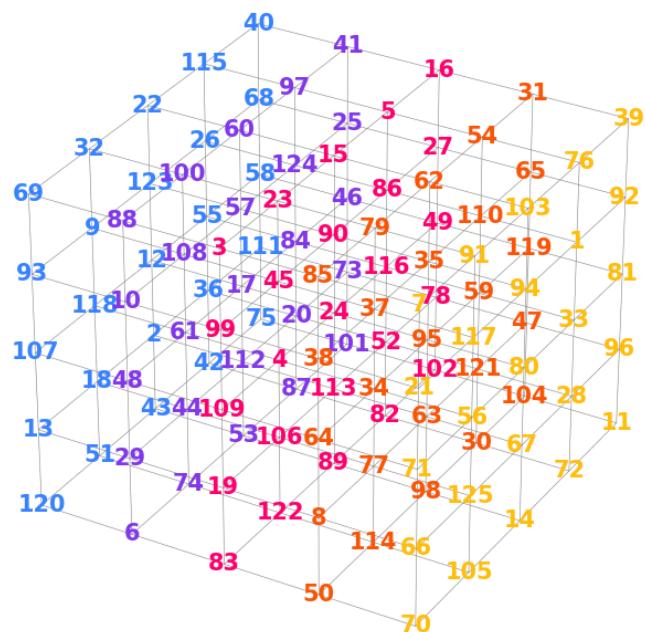
○ Cube 5:

End Population Cube 5:



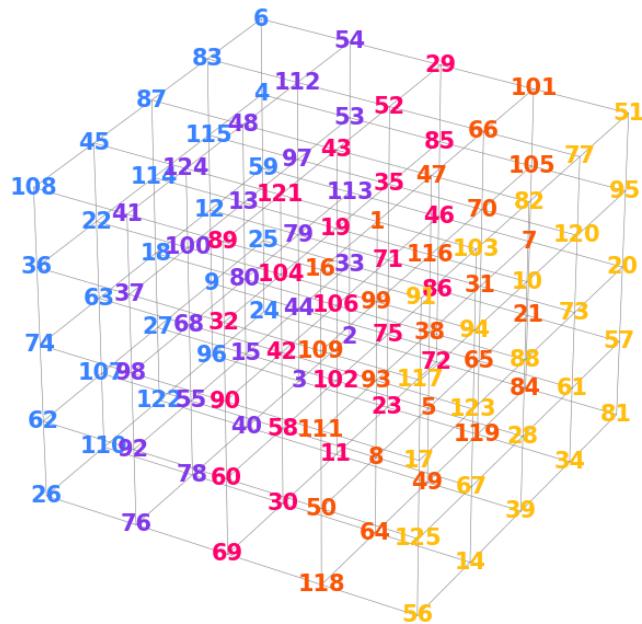
○ Cube 6:

End Population Cube 6:



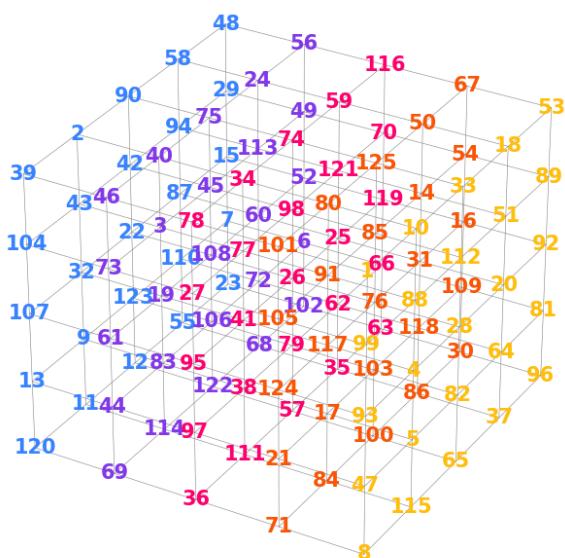
○ Cube 7:

End Population Cube 7:



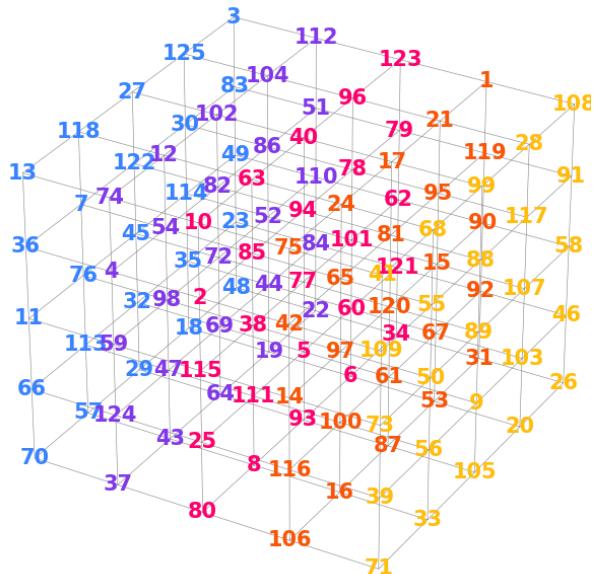
○ Cube 8:

End Population Cube 8:



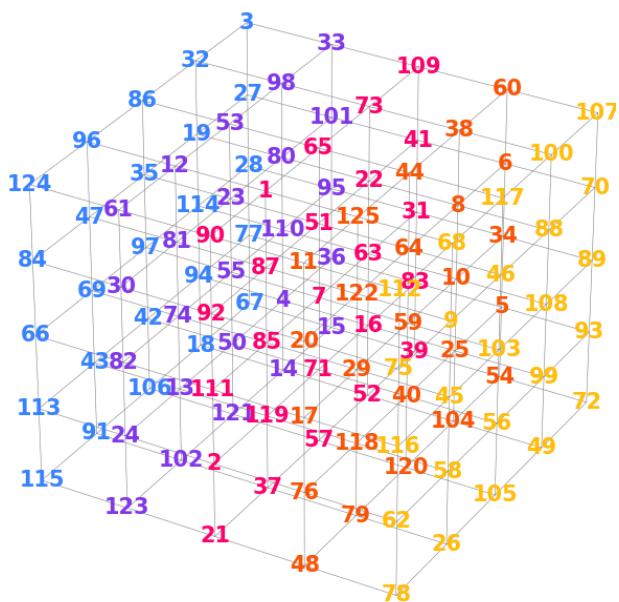
○ Cube 9:

End Population Cube 9:



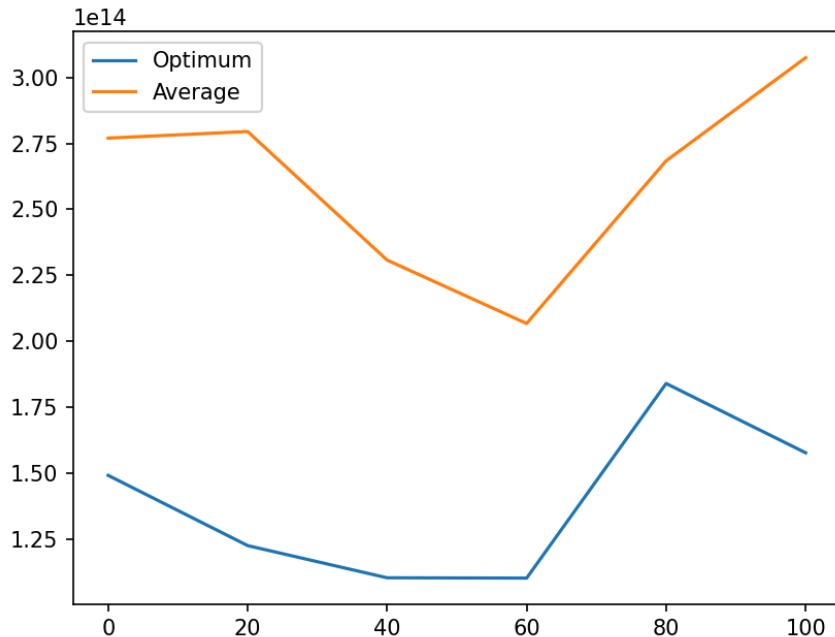
○ Cube 10:

End Population Cube 10:



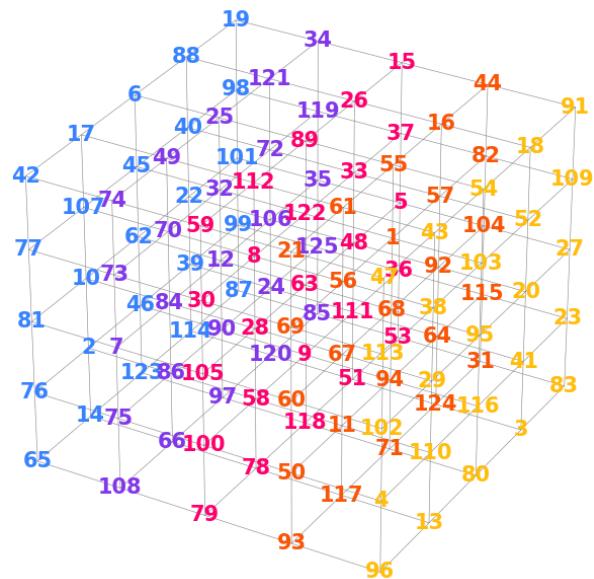
- Nilai Objective Function Akhir:
 - Fitness Cube 1: 336350542965455
 - Fitness Cube 2: 316459336742391
 - Fitness Cube 3: 215957973614571
 - Fitness Cube 4: 157794757637168
 - Fitness Cube 5: 328377703901518
 - Fitness Cube 6: 586290358823071
 - Fitness Cube 7: 184100317874422
 - Fitness Cube 8: 428148969162124
 - Fitness Cube 9: 221047520723109
 - Fitness Cube 10: 299994665321589
 - Average Fitness: 307452214676541
 - Optimum: 157794757637168

- Plot:



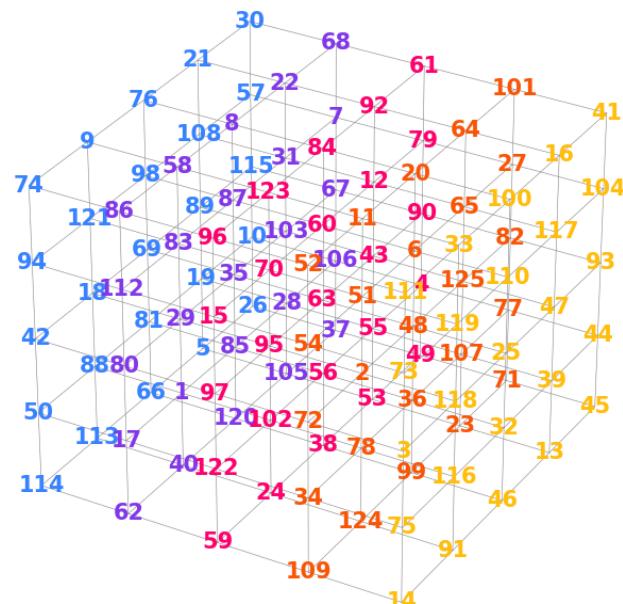
- Durasi: 1120396 microsekon
- Sampel 3:
 - State Awal:
 - Cube 1:

Start Population Cube 1:



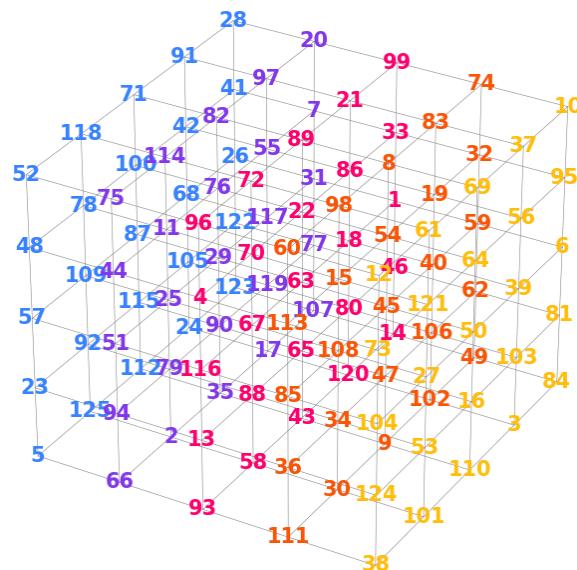
o Cube 2:

Start Population Cube 2:



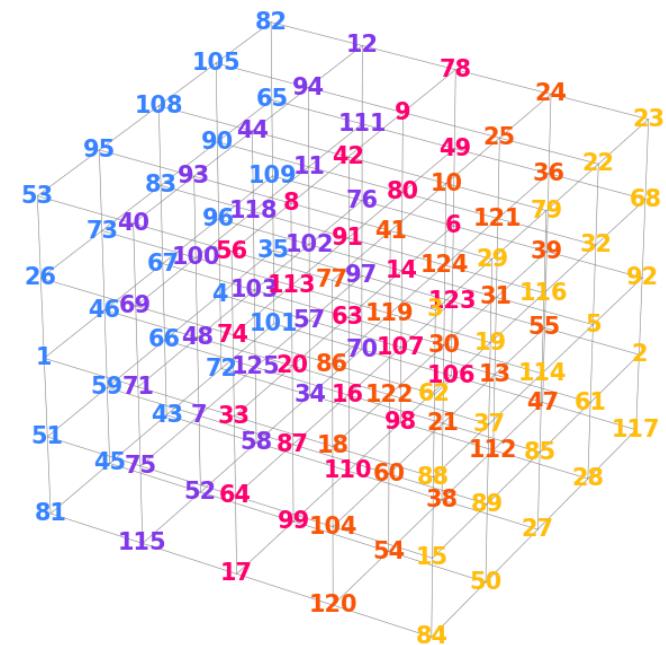
o Cube 3:

Start Population Cube 3:



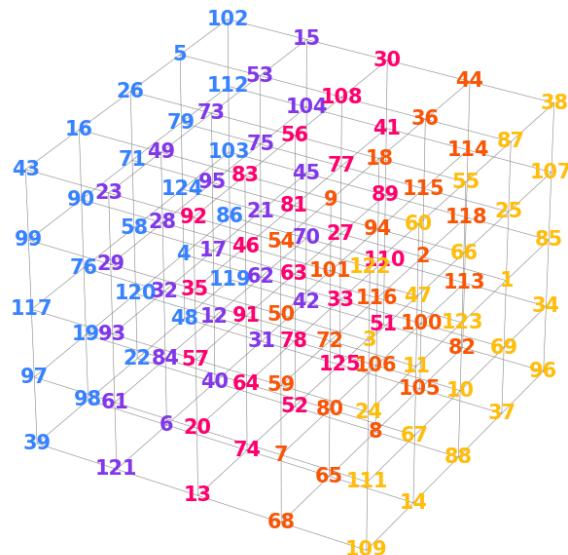
o Cube 4:

Start Population Cube 4:



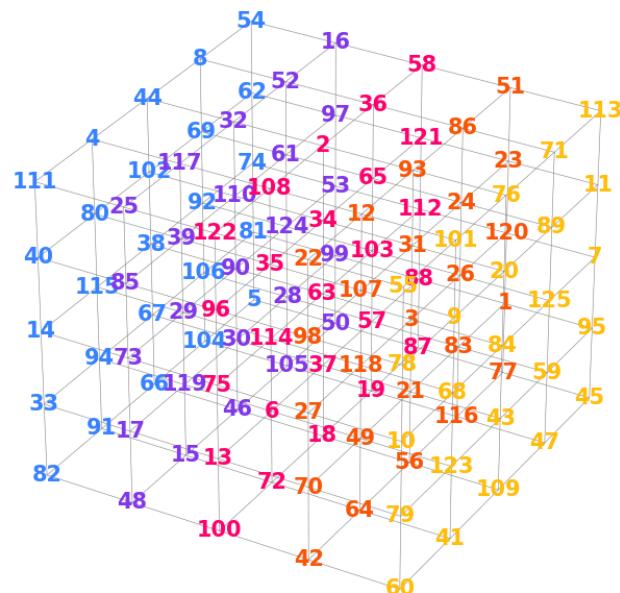
○ Cube 5:

Start Population Cube 5:



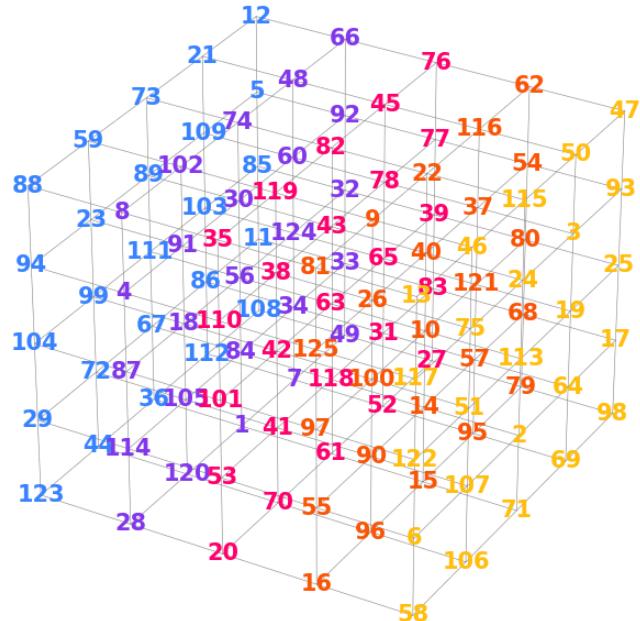
○ Cube 6:

Start Population Cube 6:



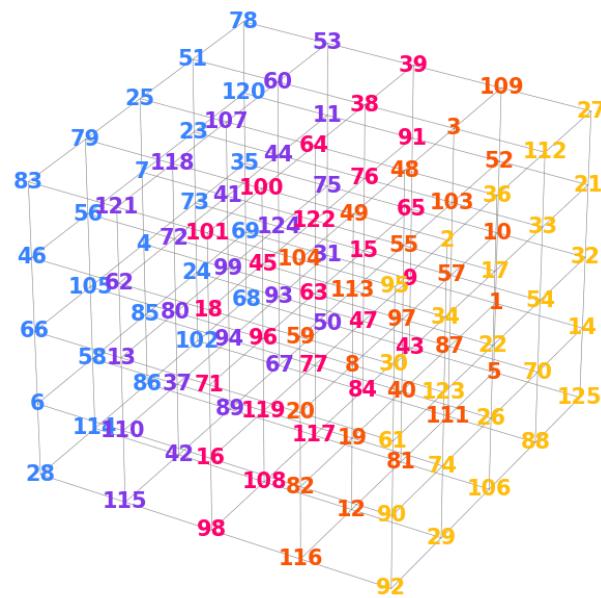
○ Cube 7:

Start Population Cube 7:



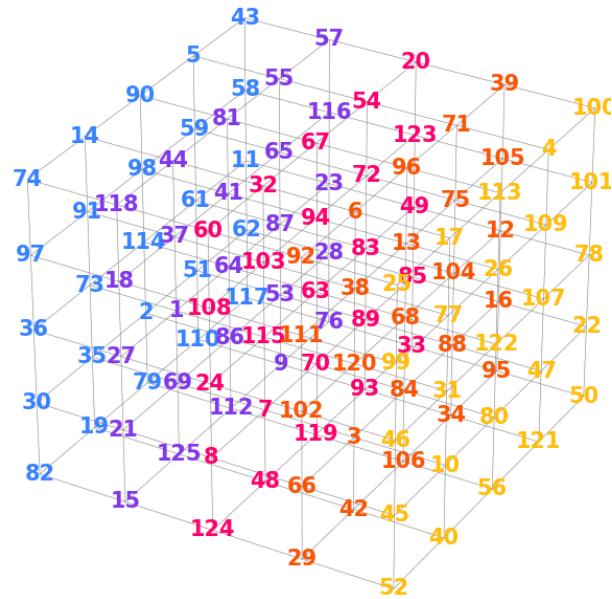
○ Cube 8:

Start Population Cube 8:



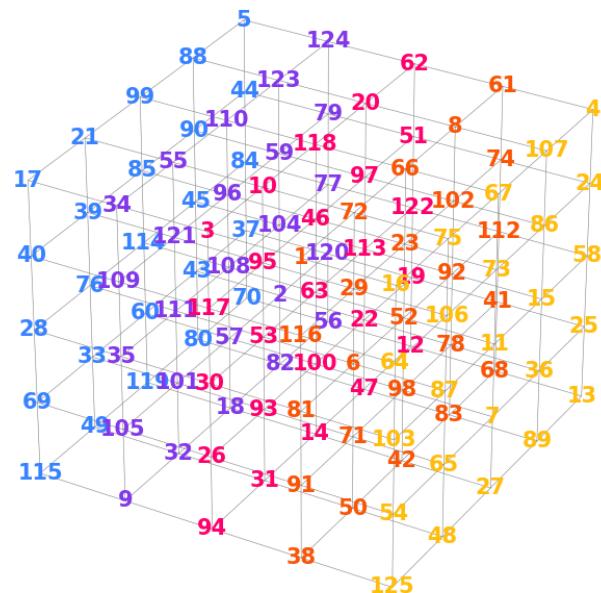
○ Cube 9:

Start Population Cube 9:



○ Cube 10:

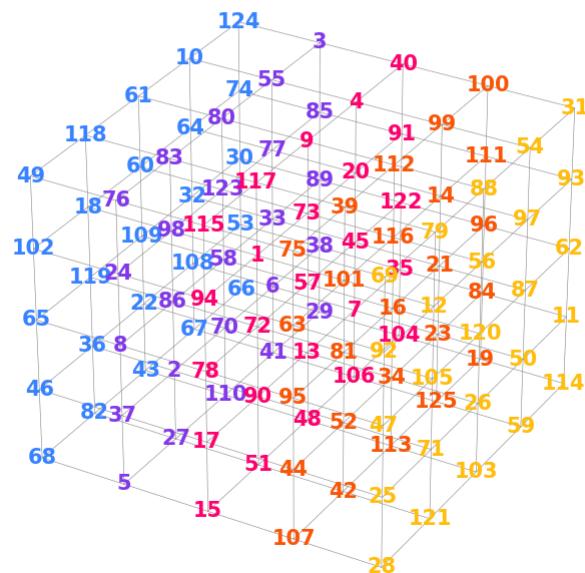
Start Population Cube 10:



- Nilai Objective Function Awal:

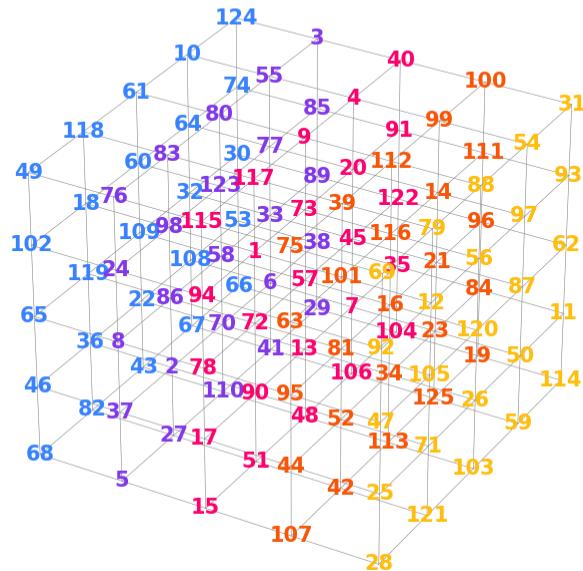
- Fitness Cube 1: 205661291303893
- Fitness Cube 2: 177132657364271
- Fitness Cube 3: 300374466439880
- Fitness Cube 4: 260409816495628
- Fitness Cube 5: 289624218119933
- Fitness Cube 6: 205931725730879
- Fitness Cube 7: 263108767492920
- Fitness Cube 8: 243040892599027
- Fitness Cube 9: 291442742073040
- Fitness Cube 10: 381606484134241
- Average Fitness: 261833306175371
- Optimum: 177132657364271
- State Akhir:
- Cube 1:

End Population Cube 1:



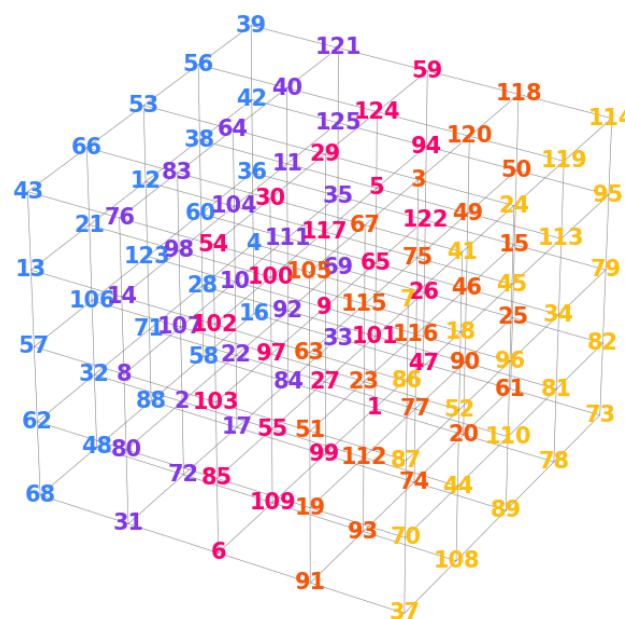
- Cube 2:

End Population Cube 2:



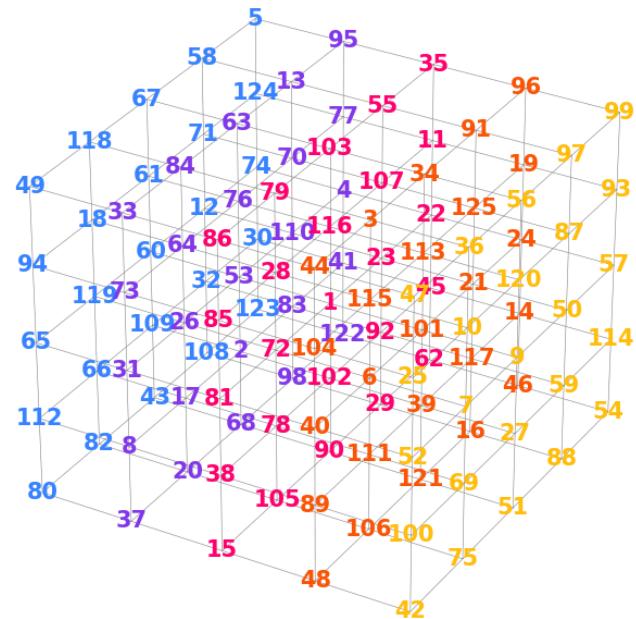
- Cube 3:

End Population Cube 3:



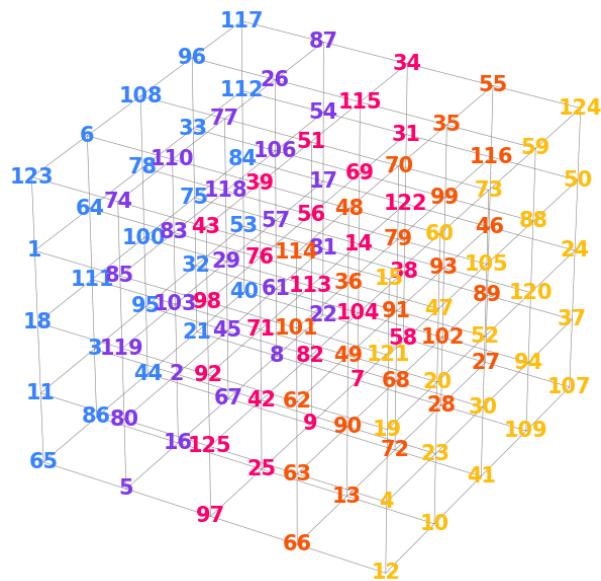
- Cube 4:

End Population Cube 4:



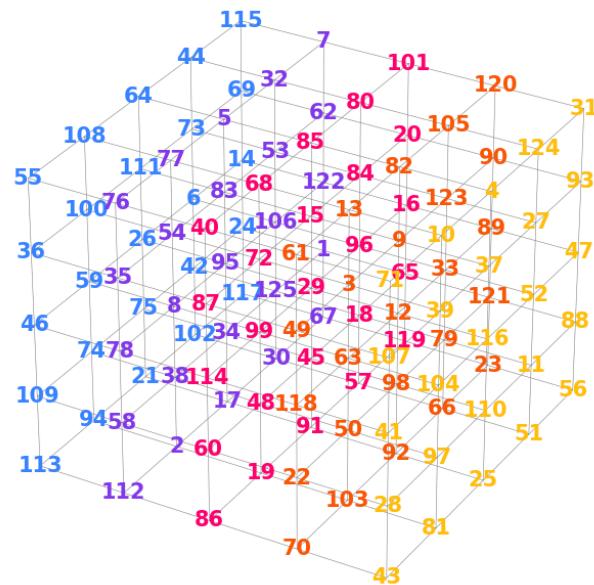
o Cube 5:

End Population Cube 5:



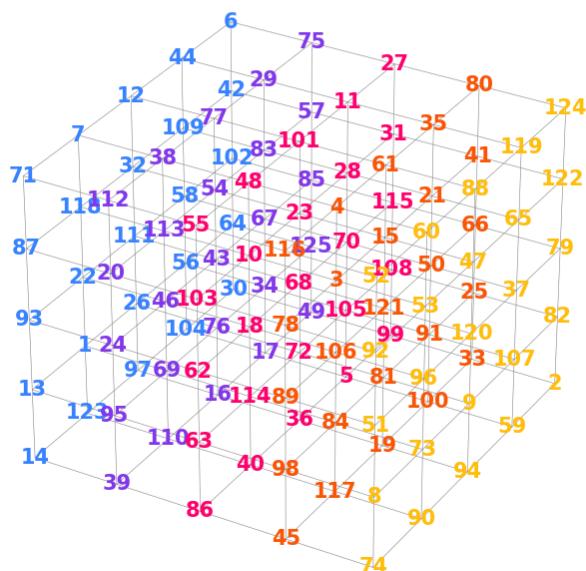
○ Cube 6:

End Population Cube 6:



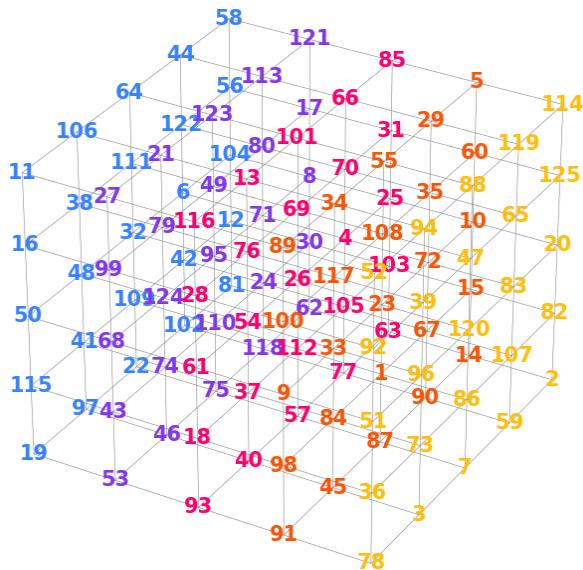
○ Cube 7:

End Population Cube 7:



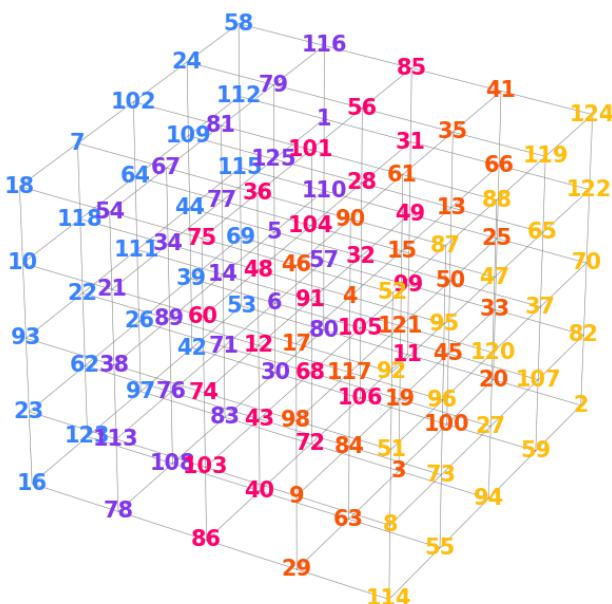
○ Cube 8:

End Population Cube 8:



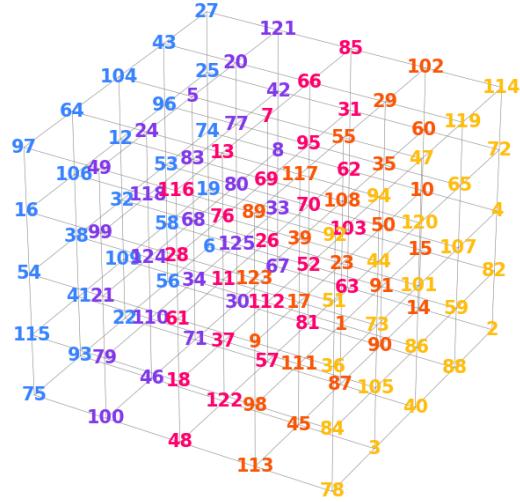
- Cube 9:

End Population Cube 9:

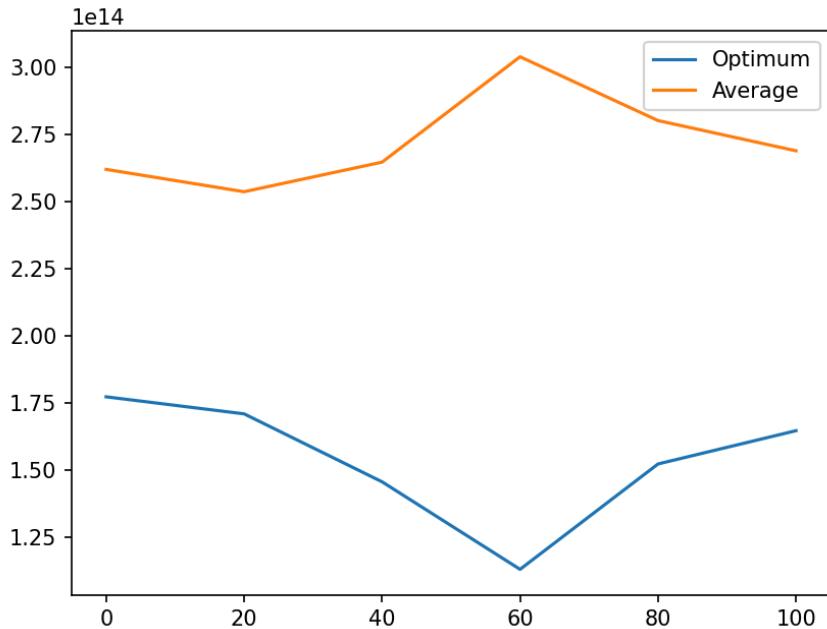


- Cube 10:

End Population Cube 10:



- Nilai Objective Function Akhir:
 - Fitness Cube 1: 277945725937736
 - Fitness Cube 2: 277945725937736
 - Fitness Cube 3: 242662914214163
 - Fitness Cube 4: 167611646942667
 - Fitness Cube 5: 340064654170927
 - Fitness Cube 6: 438731483838641
 - Fitness Cube 7: 164522816785678
 - Fitness Cube 8: 290894082111006
 - Fitness Cube 9: 253274183468779
 - Fitness Cube 10: 233908595859246
 - Average Fitness: 268756182926657
 - Optimum: 164522816785678
- Plot:



- Durasi: 1045246 microsekon
- Variansi 2:
 - Jumlah Populasi: **10**
 - Banyak Iterasi: **200**
 - Sampel 1:
 - State Awal:
 - Cube 1:
 - 54 89 97 111 22 7 107 122 43 38 58 91 5 110 125
 - 53 18 100 26 13 25 29 79 50 12 68 31 69 1 3 8 41
 - 119 45 52 28 114 109 87 117 44 60 105 73 33 59 86
 - 14 95 40 47 4 30 51 80 67 88 85 23 17 72 66 63 9
 - 121 77 78 102 96 56 112 76 99 65 120 116 39 46 84
 - 35 32 55 62 6 16 71 123 49 10 104 11 42 124 113
 - 82 106 118 57 94 74 64 34 90 108 115 15 37 27 92
 - 24 48 81 21 2 103 19 101 98 36 75 93 70 20 83 61
 - Cube 2:
 - 70 47 27 54 85 89 100 97 33 26 93 112 13 23 77
 - 118 99 68 102 106 39 88 37 55 78 11 120 61 94 116
 - 111 75 40 17 104 96 45 50 86 109 56 18 16 67 71

30 46 28 113 64 52 31 72 82 6 124 121 119 21 62
107 79 63 29 42 115 81 51 80 2 74 24 12 35 36 87
98 92 125 7 83 10 103 1 3 44 84 14 4 43 9 90 73 65
69 114 8 41 123 60 38 32 20 48 19 5 53 15 59 108
66 91 22 76 101 49 58 117 57 122 105 95 34 110 25

- Cube 3:

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

- Cube 4:

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

- Cube 5:

113 115 54 53 16 124 86 106 20 108 37 49 117 60
46 111 57 32 104 102 4 31 51 17 3 73 95 13 121 8
79 48 69 38 55 56 1 50 29 61 24 44 75 77 33 68 98
84 78 40 35 92 21 123 90 64 116 91 83 23 45 74 63
94 70 10 43 66 39 25 7 2 30 58 97 22 5 19 67 110
28 82 122 81 76 18 72 112 41 71 9 118 27 11 65 88

103 59 87 105 42 109 12 89 47 26 114 93 80 100 15
62 120 119 96 101 52 6 34 125 85 99 107 36 14

- Cube 6:

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

- Cube 7:

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

- Cube 8:

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

- Cube 9:

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

- Cube 10:

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

- Nilai Objective Function Awal:

- Fitness Cube 1: 633637836350742
- Fitness Cube 2: 249682186367923
- Fitness Cube 3: 308160571619490
- Fitness Cube 4: 167631867739739
- Fitness Cube 5: 327894355720238
- Fitness Cube 6: 192207494614301
- Fitness Cube 7: 298764234133462
- Fitness Cube 8: 386354330671312
- Fitness Cube 9: 336663567375164
- Fitness Cube 10: 206885522683322
- Average Fitness: 310788196727569
- Optimum: 167631867739739

- State Akhir:

- Cube 1:

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

- Cube 2:

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

- Cube 3:

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

- Cube 4:

107 79 88 103 45 29 84 28 30 42 122 68 38 96 89
 37 61 31 60 114 73 12 66 10 32 55 17 57 93 39 4 50
 86 81 18 99 44 76 41 9 74 22 47 109 105 54 26 101

110 58 116 123 6 23 75 20 64 106 113 115 112 53 8
104 118 91 90 87 65 124 34 77 2 95 83 24 63 70 35
13 49 67 36 98 21 1 119 71 11 16 48 7 5 59 125 92
51 14 25 62 85 80 43 52 102 100 97 19 120 3 117
40 69 33 27 78 15 111 82 46 72 121 108 56 94

- Cube 5:

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

- Cube 6:

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

- Cube 7:

46 15 42 58 65 70 12 103 74 76 101 106 40 49 13
114 73 116 57 10 34 110 27 122 113 66 97 79 4 61
107 90 111 67 53 118 8 120 84 43 69 25 60 62 56
38 52 75 51 63 119 24 77 105 71 99 108 2 100 44
32 41 104 28 36 91 31 54 98 5 93 64 125 117 92 19
33 37 20 9 85 81 83 109 16 80 11 3 47 72 45 95 14

29 78 102 124 112 50 121 1 30 6 94 22 88 96 35 86
18 39 55 59 115 87 21 68 123 82 89 23 17 7 26 48

- Cube 8:

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

- Cube 9:

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

- Cube 10:

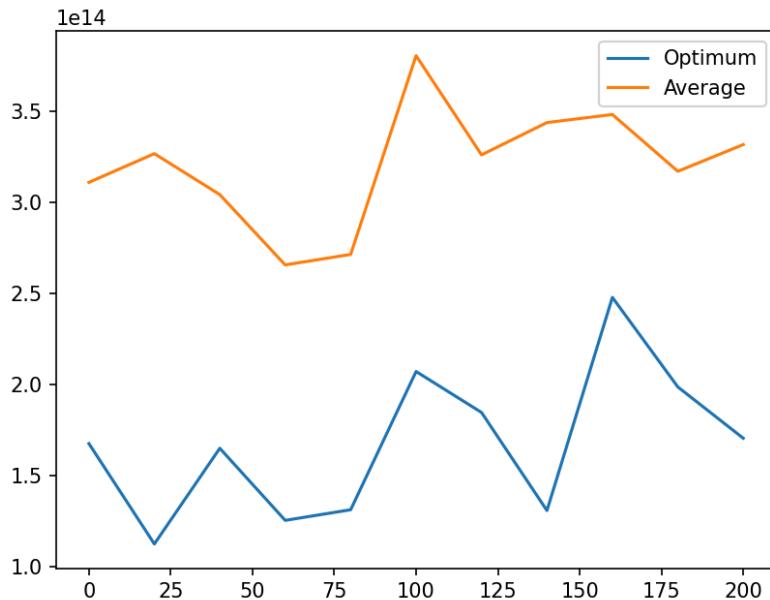
61 79 17 53 18 123 112 38 116 10 21 20 6 73 72 85
113 43 13 117 125 103 71 4 118 91 33 99 29 89 114
64 122 19 100 77 26 28 119 93 40 58 109 45 80 95
49 86 120 82 62 104 68 23 27 41 96 50 59 66 101
105 115 25 30 78 34 15 110 102 74 16 37 22 83 24
87 70 35 54 3 92 8 98 42 55 88 97 14 32 46 39 106
11 81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

■ Nilai Objective Function Akhir:

- Fitness Cube 1: 462739608260196

- Fitness Cube 2: 395947516693048
- Fitness Cube 3: 170569964144814
- Fitness Cube 4: 207601583712516
- Fitness Cube 5: 497364206629067
- Fitness Cube 6: 234742675464514
- Fitness Cube 7: 345916463251234
- Fitness Cube 8: 307477764724348
- Fitness Cube 9: 505483070765645
- Fitness Cube 10: 187027915959968
- Average Fitness: 331487076960535
- Optimum: 170569964144814

■ Plot:



■ Durasi: 1982430 microsekon

■ Sampel 2:

- State Awal:

- Cube 1:

```

111 46 6 87 97 1 42 47 38 100 106 90 59 72 4 10 8
69 14 33 30 9 11 43 34 21 117 121 20 19 36 40 107
81 74 18 58 61 50 125 123 17 96 32 57 116 95 102

```

112 29 22 71 64 62 85 66 41 84 67 86 26 68 63 120
65 118 15 124 75 54 70 49 77 12 45 52 99 2 105 93
16 37 108 89 7 13 104 56 78 60 31 103 48 28 88
122 113 109 94 114 83 51 25 55 5 119 39 98 24 115
91 92 110 23 76 27 80 53 35 79 44 101 82 3 73

- Cube 2:

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

- Cube 3:

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

- Cube 4:

88 37 66 124 43 120 51 68 39 45 105 2 9 52 34 92
36 41 33 80 117 64 15 17 109 53 55 71 58 18 118
112 1 108 60 96 30 76 81 97 42 27 20 93 67 57 23
119 69 79 3 87 121 10 49 116 89 82 8 113 11 28 63
7 24 19 84 4 115 125 95 107 98 21 61 100 40 31 77
25 83 59 73 104 78 102 114 47 26 29 111 6 22 70

106 72 103 94 90 12 44 48 122 16 13 110 5 54 123
35 50 56 101 14 46 38 32 65 62 85 86 91 75 74 99

○ Cube 5:

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

○ Cube 6:

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

○ Cube 7:

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

○ Cube 8:

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

- Cube 9:

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

- Cube 10:

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

- Nilai Objective Function Awal:

- Fitness Cube 1: 371909960200944
- Fitness Cube 2: 134607859672249
- Fitness Cube 3: 443394766394049
- Fitness Cube 4: 403062411373155

- Fitness Cube 5: 149278322320453
 - Fitness Cube 6: 313499637070098
 - Fitness Cube 7: 563812675618001
 - Fitness Cube 8: 478554381895462
 - Fitness Cube 9: 352888798321004
 - Fitness Cube 10: 144343192616843
 - Average Fitness: 335535200548225
 - Optimum: 134607859672249
- State Akhir:
 - Cube 1:
 79 64 72 123 74 73 3 89 65 91 121 18 71 114 1 116
 87 69 59 52 81 43 124 19 32 34 63 77 22 113 88 24
 7 61 119 48 11 25 76 67 68 86 9 99 93 110 101 36
 112 85 111 117 106 108 66 84 58 107 29 16 42 95
 45 75 2 103 44 51 70 96 125 35 40 31 53 39 17 47
 50 118 20 98 122 12 60 10 97 6 82 26 55 15 92 13
 83 104 57 28 8 49 54 78 27 23 21 80 14 5 109 115
 105 94 56 37 33 90 120 46 62 102 4 41 100 38 30
 - Cube 2:
 101 62 58 37 113 34 87 75 115 45 19 46 44 63 110
 121 54 65 124 43 125 80 69 4 76 3 117 86 25 72 53
 114 73 41 120 28 11 29 18 112 95 33 51 108 70 96
 31 81 17 47 118 71 97 61 100 88 24 7 23 38 82 20
 119 30 104 67 52 99 6 122 50 68 15 103 102 91 16
 13 74 79 1 55 93 8 26 83 36 42 98 9 85 111 12 107
 77 35 92 39 106 14 10 2 40 66 27 105 94 32 48 64
 60 57 84 49 78 21 90 109 123 89 116 59 22 56 5
 - Cube 3:
 101 64 109 37 34 115 49 124 21 15 58 73 3 123 110
 60 65 95 86 121 125 102 108 114 80 83 78 116 25
 72 96 87 69 59 52 90 11 120 84 112 19 45 51 76 57

63 77 29 113 17 44 75 12 61 28 88 24 7 23 32 82 20
119 31 104 67 70 99 6 122 50 68 118 103 18 91 16
13 74 79 1 55 93 8 26 97 36 81 98 9 85 111 10 107
89 35 92 39 106 14 54 2 40 66 27 105 94 42 48 5 33
100 53 43 46 62 71 47 4 38 41 117 22 56 30

- Cube 4:

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

- Cube 5:

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

- Cube 6:

39 64 46 108 24 30 12 21 45 73 26 123 61 103 92
90 85 79 14 8 80 43 74 62 49 81 83 75 17 98 55 72
96 120 118 59 20 18 115 95 57 44 3 121 125 71 28
41 29 84 37 97 82 87 104 105 10 31 106 77 67 122
111 117 6 93 19 13 50 65 101 78 86 53 109 100 23
119 4 60 48 22 2 35 91 36 68 34 66 7 51 58 1 11

114 107 32 9 25 63 54 27 113 110 102 76 16 42 99
5 15 69 112 88 47 33 116 70 40 56 52 38 94 89 124

- Cube 7:

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

- Cube 8:

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

- Cube 9:

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

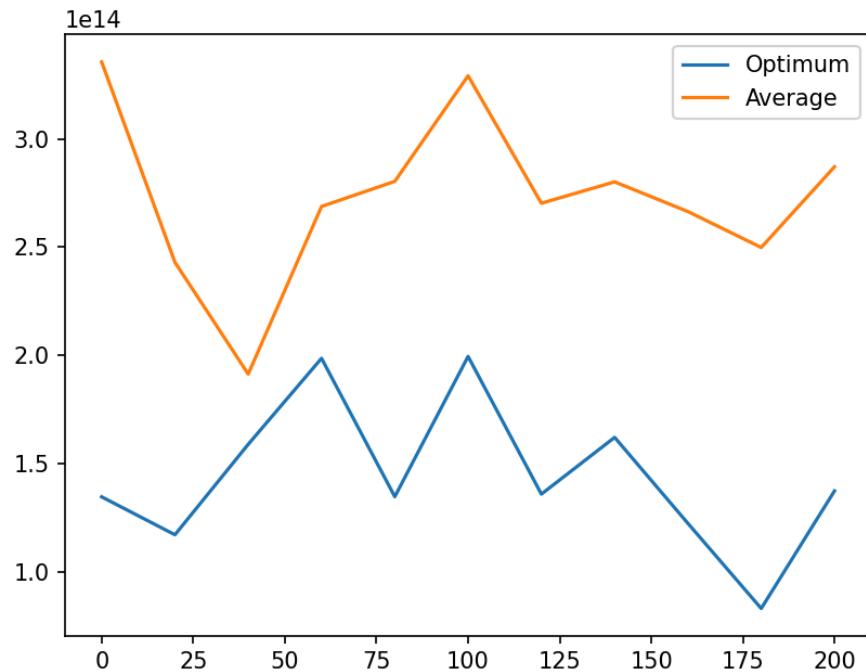
- Cube 10:

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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 206547927795593
- Fitness Cube 2: 156169949523001
- Fitness Cube 3: 331245404812497
- Fitness Cube 4: 331245404812497
- Fitness Cube 5: 283484735508079
- Fitness Cube 6: 283484735508079
- Fitness Cube 7: 488463342603918
- Fitness Cube 8: 245890366348068
- Fitness Cube 9: 137353861339624
- Fitness Cube 10: 406859422338839
- Average Fitness: 287074515059019
- Optimum: 137353861339624

- Plot:



- Durasi: 2106282 microsekon
- Sampel 3:
 - State Awal:
 - Cube 1:

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

124 39 117 34 54 21 17 1 89 9 104 20 119 101 98
 32 93 92 122 23 35 91 118 36 45 51 73 107 113 25
 69 41 114 57 2 88 85 27 44 24 103 55 48 6 61 30
 102 86 123 108 16 15 87 14 19 67 81 71 110 74 106
 60 63 75 8 12 70 64 111 31 26 112 95 120 62 58 37

38 66 3 49 52 59 13 97 5 84 22 65 43 33 72 83 7 99
77 78 42 28 18 94 76 100 109 121 125 80 56 116 82
105 4 96 10 115 40 90 50 68 29 79 46 47 53 11

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

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

26 2 108 106 59 42 24 11 117 27 94 93 21 125 63
105 69 12 60 114 50 16 57 111 48 30 82 1 43 9 119
70 52 49 101 102 25 5 17 3 19 88 107 115 85 110
31 87 56 10 18 29 23 83 75 51 97 37 34 47 7 96 14
61 45 55 92 44 122 113 112 121 64 67 120 28 54

- Cube 10:

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

- Nilai Objective Function Awal:

- Fitness Cube 1: 237023720381333
- Fitness Cube 2: 243960045600219
- Fitness Cube 3: 336617011136248
- Fitness Cube 4: 322589945292076
- Fitness Cube 5: 494764064747517
- Fitness Cube 6: 143463663539174
- Fitness Cube 7: 357540865812094
- Fitness Cube 8: 191408002153203
- Fitness Cube 9: 454850580102567
- Fitness Cube 10: 300925883892321
- Average Fitness: 308314378265675
- Optimum: 143463663539174

- State Akhir:

- Cube 1:

51 71 48 59 43 65 33 107 36 90 7 120 46 22 73 25
10 44 77 40 89 52 53 5 99 9 50 14 29 38 67 78 4 69

94 86 1 83 57 104 42 16 24 74 82 62 23 121 30 66
19 105 117 39 95 3 116 97 80 34 85 98 2 112 75 56
61 91 88 70 63 32 60 21 35 58 37 96 119 87 101
106 76 102 84 100 45 41 17 26 8 113 118 6 123 110
27 79 20 115 55 108 64 92 49 124 28 122 11 13 81
68 114 15 12 93 47 109 111 31 54 18 72 103 125

- Cube 2:

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

- Cube 3:

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

- Cube 4:

40 69 20 77 92 88 79 18 42 44 19 65 63 117 8 36
123 34 124 98 108 29 23 41 84 86 16 100 51 55 78
90 103 4 114 75 85 13 50 122 14 61 94 24 112 93
74 121 102 22 45 53 97 1 10 110 105 91 37 104 46
115 111 89 26 27 62 120 2 6 82 106 49 113 58 12
17 70 116 31 107 25 7 101 83 59 125 60 80 118 56

47 43 87 72 68 11 119 3 57 66 76 21 33 28 39 38 54
48 71 15 99 81 30 9 95 32 109 73 64 35 67 52 5 96

○ Cube 5:

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

○ Cube 6:

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

○ Cube 7:

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

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

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

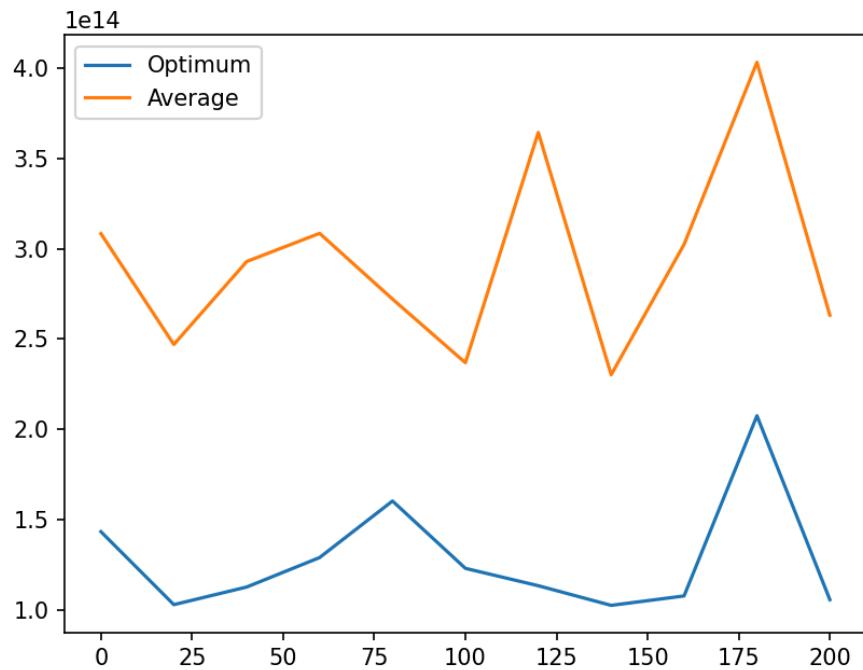
- Cube 10:
 ○ 51 77 40 17 46 90 67 43 121 81 35 102 100 21 52
 70 83 122 50 120 26 10 117 9 71 4 36 55 59 30 86
 23 29 32 6 76 72 68 87 7 101 57 79 34 111 82 62
 108 85 56 2 45 28 53 24 105 14 12 88 1 25 124 123
 96 119 41 15 49 48 74 73 98 112 95 89 18 22 106
 66 13 104 19 38 84 125 109 44 97 39 116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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 610076518649331
- Fitness Cube 2: 133488462392642
- Fitness Cube 3: 261408797655652

- Fitness Cube 4: 105882732648092
- Fitness Cube 5: 184643419692172
- Fitness Cube 6: 232818336218117
- Fitness Cube 7: 219555332913528
- Fitness Cube 8: 219555332913528
- Fitness Cube 9: 296427923029464
- Fitness Cube 10: 368418275536220
- Average Fitness: 263227513164874
- Optimum: 105882732648092

- Plot:



- Durasi: 2143345 microsekon

- Variansi 3:
 - Jumlah Populasi: **10**
 - Banyak Iterasi: **500**
 - Sampel 1:
 - State Awal:
 - Cube 1:

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

- Cube 2:

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

- Cube 3:

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

- Cube 4:

49 125 41 38 91 55 73 39 35 44 31 94 28 79 85 104
68 59 122 99 70 81 51 27 67 75 113 97 42 82 72 25
5 83 18 45 60 30 120 8 46 92 23 43 16 52 100 80 40
106 66 116 56 74 21 37 117 110 123 90 3 12 63 98

33 93 20 62 101 29 9 48 10 7 69 1 115 14 19 22 58
78 77 121 50 87 111 118 64 105 61 76 103 11 57
107 95 112 109 32 15 17 4 88 119 65 108 24 84 53
13 2 54 102 86 124 96 26 89 71 114 34 47 36 6

- Cube 5:

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

- Cube 6:

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

- Cube 7:

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

108 76 35 69 103 97 12 72 46 124 95 87 15 82 48
107

- Cube 8:

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

- Cube 9:

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

- Cube 10:

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

■ Nilai Objective Function Awal:

- Fitness Cube 1: 359187052355186

- Fitness Cube 2: 513990353196849
- Fitness Cube 3: 178130547650185
- Fitness Cube 4: 244004678550098
- Fitness Cube 5: 310842134780125
- Fitness Cube 6: 615223457946028
- Fitness Cube 7: 327465924495305
- Fitness Cube 8: 125290558873928
- Fitness Cube 9: 185766476889071
- Fitness Cube 10: 243628111693986
- Average Fitness: 310352929643076
- Optimum: 125290558873928

■ State Akhir:

- Cube 1:

```

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

```

- Cube 2:

```

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

```

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

- Cube 6:

18 21 5 4 37 78 51 19 102 46 56 105 62 54 87 73 57
60 30 124 59 48 74 72 106 119 35 11 82 14 95 71
65 31 111 93 117 34 9 90 98 89 29 69 120 110 49
61 63 40 70 27 6 100 8 16 36 1 92 114 23 28 22 77

64 32 68 20 7 96 24 66 75 116 103 80 97 47 44 107
83 104 12 13 41 38 58 109 15 17 53 101 113 112 91
26 33 84 85 123 121 108 55 122 94 42 88 76 39 118
67 81 86 115 52 25 45 50 3 10 125 43 79 99 2

- Cube 7:

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

- Cube 8:

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

- Cube 9:

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

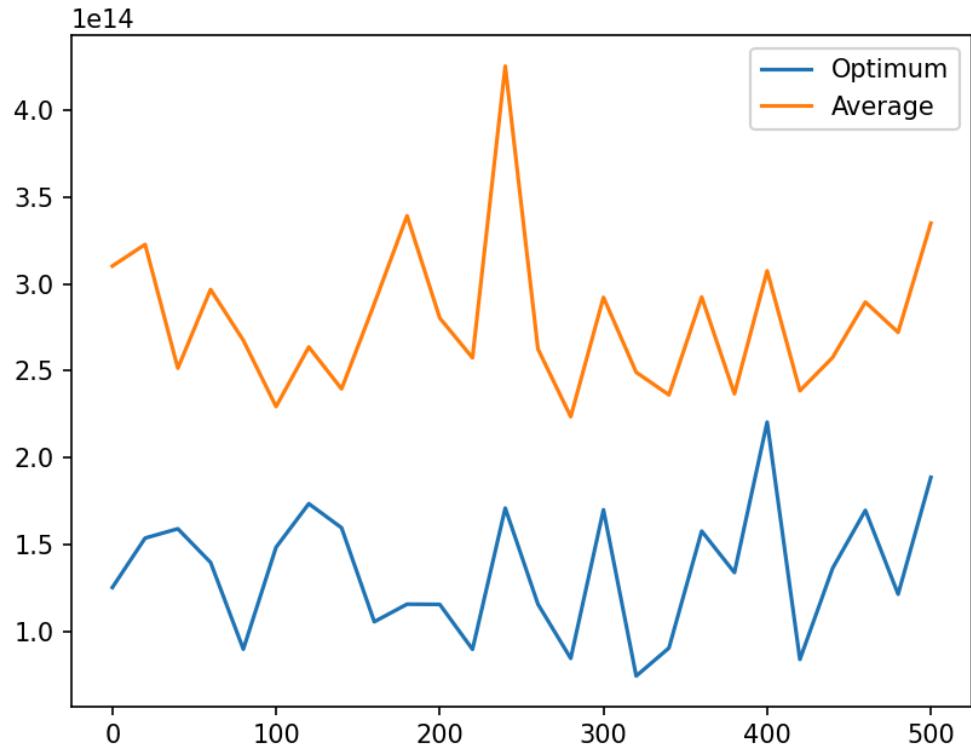
- Cube 10:

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

■ Nilai Objective Function Akhir:

- Fitness Cube 1: 311718095655637
- Fitness Cube 2: 188642145970384
- Fitness Cube 3: 402170215040151
- Fitness Cube 4: 397352245849049
- Fitness Cube 5: 272433958866704
- Fitness Cube 6: 540243985754265
- Fitness Cube 7: 221569523400557
- Fitness Cube 8: 221569523400557
- Fitness Cube 9: 549113376194091
- Fitness Cube 10: 245071949704802
- Average Fitness: 334988501983619
- Optimum: 188642145970384

■ Plot:



- Durasi: 5049579 microsekon
 - Sampel 2:
 - State Awal:
 - Cube 1:
- 58 117 44 43 35 29 88 79 95 104 56 85 86 122 4
 105 82 21 73 24 116 96 5 37 100 71 109 64 25 111
 72 80 40 36 19 55 62 38 45 65 12 30 52 23 124 94
 70 78 26 121 91 34 119 68 32 67 113 99 51 2 50 77
 63 11 59 42 8 69 74 57 123 28 20 18 125 15 54 93 9
 114 53 98 3 41 14 102 17 106 103 33 120 89 48 47
 110 61 90 13 92 84 75 115 76 10 112 22 27 66 49
 97 108 118 87 1 6 16 46 31 81 101 39 7 83 60 107
- Cube 2:
- 9 29 82 26 3 55 7 61 46 54 60 92 50 89 117 103 106
 67 68 93 65 21 112 22 2 80 58 71 102 94 101 85 73
 79 33 24 6 12 31 16 124 15 100 86 43 40 66 32 35
 114 72 45 88 99 38 91 70 10 74 123 109 64 63 28

96 125 49 98 42 14 36 8 105 84 97 87 122 120 59 5
95 20 23 69 104 18 17 34 47 13 53 4 19 118 48 116
56 39 44 57 37 121 111 108 77 11 25 41 1 75 62 27
81 52 119 107 30 76 51 83 90 115 113 78 110

- Cube 3:

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

- Cube 4:

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

- Cube 5:

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

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

66 101 96 95 53 125 92 12 84 11 1 106 4 90 22 103
63 69 44 76 102 117 28 110 72 81 40 78 124 14 36
7 41 85 2 98 32 80 93 119 111 60 8 21 56 51 109 42
55 83 86 64 47 45 123 39 31 6 37 3 67 29 121 74 33
18 73 10 107 88 5 120 77 75 70 79 61 59 94

- Cube 10:

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

- Nilai Objective Function Awal:

- Fitness Cube 1: 216328511765339
- Fitness Cube 2: 420102733264802
- Fitness Cube 3: 328635218081604
- Fitness Cube 4: 285523853803017
- Fitness Cube 5: 425045274049603
- Fitness Cube 6: 248046010565887
- Fitness Cube 7: 227682457789111
- Fitness Cube 8: 330020146573567
- Fitness Cube 9: 389646806684073
- Fitness Cube 10: 138114006484075
- Average Fitness: 300914501906107
- Optimum: 138114006484075

- State Akhir:

- Cube 1:

52 97 96 7 57 85 80 49 15 25 84 51 10 6 66 53 90
123 13 101 39 88 34 44 100 30 121 111 94 56 29 48

55 93 16 114 72 46 89 37 91 5 124 125 113 71 106
58 77 92 107 118 50 26 68 27 59 4 19 38 8 31 115
22 61 67 54 12 64 102 35 87 98 47 65 23 117 83 42
78 82 45 18 95 2 120 112 75 122 24 108 76 9 81 79
11 3 86 14 109 41 110 21 32 17 105 62 20 103 70
116 40 43 1 99 28 69 119 104 63 60 33 73 36 74

- Cube 2:

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

- Cube 3:

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

- Cube 4:

73 39 86 21 59 90 83 122 22 69 10 65 30 8 14 2 95
62 97 67 31 82 23 72 102 78 1 87 44 94 6 68 28 47
33 99 64 101 121 42 103 43 88 85 92 108 114 15
111 84 4 27 57 60 98 35 34 24 76 16 112 41 123 25
52 12 26 75 107 9 71 13 55 32 11 17 61 3 120 77 79
119 49 105 48 54 18 116 7 81 96 117 46 53 36 93

70 124 66 80 109 58 106 115 40 125 100 19 45 89

37 110 113 50 74 104 91 56 51 118 63 20 38 29 5

○ Cube 5:

18 39 80 85 46 28 97 102 95 71 73 99 96 45 66 53

90 123 13 67 14 88 56 117 100 30 121 111 27 37 79

48 55 74 16 23 72 10 105 44 91 6 38 41 92 36 109

20 9 57 29 87 106 115 40 54 7 113 119 89 77 35 15

78 19 64 47 112 108 62 17 120 116 60 51 12 114 63

24 101 125 107 33 83 61 11 32 75 52 104 103 124

68 42 3 26 94 86 4 31 118 49 21 84 8 1 25 122 93

70 98 59 43 76 81 50 65 22 5 58 69 2 34 110 82

○ Cube 6:

59 14 26 45 66 11 67 73 107 100 9 85 23 34 121 39

86 16 80 90 91 75 52 57 62 97 13 108 114 32 105

56 102 93 1 17 89 3 25 31 28 10 33 83 124 70 87 44

64 58 82 53 40 74 103 88 55 96 7 99 94 116 4 79 15

42 47 104 113 123 118 6 117 120 51 77 8 69 49 101

50 60 27 98 35 109 112 95 122 24 48 111 119 37 30

43 72 41 38 92 36 20 46 115 54 84 22 78 68 106 61

18 71 76 125 110 65 19 81 12 5 2 21 63 29

○ Cube 7:

83 39 63 21 97 92 112 34 73 28 71 78 77 90 98 52

36 13 114 47 124 104 54 93 123 94 24 120 14 7 62

25 59 5 106 91 49 76 31 117 16 23 72 44 67 53 38

84 125 32 10 42 111 87 109 26 41 102 45 82 96 74

64 85 11 9 3 121 15 58 33 81 88 68 107 113 4 99 35

116 122 55 56 27 110 115 46 1 20 30 19 119 101 51

118 22 65 86 69 37 6 40 79 50 89 18 80 105 75 103

2 66 95 8 100 43 60 70 57 61 12 17 108 29 48

○ Cube 8:

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

- Cube 9:

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

- Cube 10:

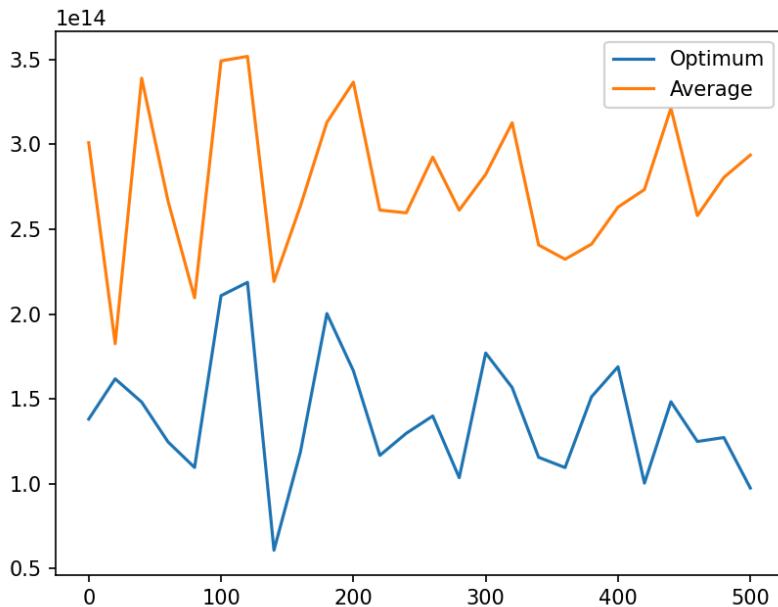
85 21 122 82 8 5 97 67 9 116 12 57 53 87 28 99 124
43 108 111 102 78 19 75 34 79 76 22 47 52 112 90
123 20 24 120 62 72 54 31 25 59 15 106 91 49 104
86 64 70 44 16 3 45 117 101 92 33 35 10 88 68 107
37 4 42 61 23 94 2 32 50 56 74 27 41 83 17 55 39
109 30 84 118 119 73 51 60 77 65 40 69 115 6 81 1
93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

- Nilai Objective Function Akhir:

- Fitness Cube 1: 586858543374579
- Fitness Cube 2: 261861488301712
- Fitness Cube 3: 178764027636266
- Fitness Cube 4: 342028685729185

- Fitness Cube 5: 166845360113452
- Fitness Cube 6: 276421159611139
- Fitness Cube 7: 414130226193216
- Fitness Cube 8: 235682238349805
- Fitness Cube 9: 376262396083752
- Fitness Cube 10: 97515478573561
- Average Fitness: 293636960396666
- Optimum: 97515478573561

- Plot:



- Durasi: 5294440 microsekon
- Sampel 3:
 - State Awal:
 - Cube 1:


```
40 64 54 90 85 77 91 27 38 76 1 65 15 99 111 6 100
86 42 4 37 92 45 12 56 110 16 113 67 94 10 101 79
96 2 17 103 41 120 66 68 11 73 119 36 61 44 19 25
72 114 75 7 49 60 62 69 48 78 106 51 3 63 32 70 93
123 57 102 52 84 24 34 117 30 105 115 33 8 55 95
26 97 14 47 23 20 59 21 116 50 81 43 98 118 124
```

28 88 125 104 39 89 22 121 29 53 112 82 108 18 83
87 13 35 109 107 74 46 9 5 31 71 80 122 58

○ Cube 2:

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

○ Cube 3:

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

○ Cube 4:

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

○ Cube 5:

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

- Cube 6:

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

- Cube 7:

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

- Cube 8:

70 104 116 99 97 113 48 114 6 55 122 18 73 119 37
94 100 32 56 101 1 13 29 15 121 17 42 11 26 43 4
21 60 115 62 5 72 80 123 39 75 110 9 111 2 98 66
65 88 90 47 31 44 82 102 93 14 68 81 107 77 12 63

50 117 79 25 58 51 95 86 69 27 105 22 84 45 124
41 67 64 61 34 91 83 103 125 3 76 49 108 28 96 40
52 33 10 38 36 16 78 71 74 46 7 24 54 19 92 8 112
59 35 57 118 87 120 85 20 23 30 53 89 109 106

- Cube 9:

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

- Cube 10:

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

- Nilai Objective Function Awal:

- Fitness Cube 1: 211213047599272
- Fitness Cube 2: 721557912191531
- Fitness Cube 3: 246298118977449
- Fitness Cube 4: 393887034419174
- Fitness Cube 5: 140370194920474
- Fitness Cube 6: 144508693099988
- Fitness Cube 7: 431067934790151
- Fitness Cube 8: 183020785755675

- Fitness Cube 9: 172850003023415
 - Fitness Cube 10: 127033187439526
 - Average Fitness: 277180691221665
 - Optimum: 127033187439526
- State Akhir:
 - Cube 1:
 24 93 116 68 70 29 14 88 11 106 73 71 103 48 87
 41 27 102 114 64 4 47 46 97 96 19 80 49 5 90 39 40
 81 23 105 16 89 53 63 61 125 9 100 17 118 95 31
 54 34 55 74 15 65 3 109 123 86 77 2 104 28 42 1 26
 121 33 120 124 20 13 78 62 94 32 84 99 101 112 45
 72 67 76 38 37 57 25 52 35 30 83 119 36 108 92 56
 60 75 111 12 113 122 117 91 44 79 66 8 6 82 21
 115 22 110 10 51 69 18 50 43 59 7 58 85 107 98
 - Cube 2:
 42 33 77 23 7 110 49 121 120 19 47 85 108 107 58
 88 72 94 3 104 96 46 123 16 106 17 113 9 43 13 12
 11 92 6 28 8 114 103 65 44 41 87 101 45 54 93 57
 14 22 21 80 69 38 95 67 53 76 62 125 98 100 74 89
 59 55 124 118 50 99 111 82 27 90 71 56 63 2 40 29
 32 116 83 30 26 35 122 117 15 78 112 79 4 66 73
 37 52 64 34 20 10 105 18 36 25 119 5 84 68 75 97
 91 31 61 86 109 51 81 24 48 70 39 102 115 60 1
 - Cube 3:
 58 44 77 35 1 85 120 11 7 45 46 14 103 40 73 80
 100 48 75 96 9 29 5 59 69 89 81 36 90 30 118 37 32
 50 56 93 21 114 124 119 112 99 101 57 8 78 76 28
 23 52 84 95 25 97 116 51 63 15 2 67 27 55 121 12
 87 49 88 13 54 4 117 6 38 68 66 42 92 65 123 91 74
 64 72 107 31 86 109 110 18 102 108 115 39 20 71

98 24 53 113 94 34 82 43 16 19 122 70 79 60 41 47

22 125 33 10 104 111 17 26 106 62 105 83 3 61

- Cube 4:

59 15 30 57 125 37 31 72 29 6 77 85 124 76 67 14

87 93 64 56 66 26 81 69 22 5 94 118 96 73 12 17 55

113 102 92 48 47 2 43 112 1 62 74 42 21 110 104

25 3 52 18 78 100 101 20 38 88 40 97 120 123 63

45 121 27 23 109 11 54 71 98 9 53 34 82 89 16 4 24

106 95 49 10 51 86 13 103 68 116 122 70 117 115

91 8 50 28 39 7 119 36 108 65 79 60 75 19 33 99

107 80 46 41 105 114 83 32 61 58 84 35 111 90 44

- Cube 5:

59 33 62 57 124 27 90 93 81 45 43 73 79 112 30

111 123 106 55 23 122 75 56 77 40 83 29 63 80 16

51 42 86 66 119 61 117 76 89 32 47 14 105 3 8 54

107 94 28 38 121 68 22 5 11 36 84 21 6 67 35 101

26 85 19 31 102 65 116 49 114 7 17 9 72 12 120 97

64 48 2 18 87 110 109 50 104 92 118 69 108 115 39

125 71 103 24 82 113 88 34 20 1 53 100 41 46 52

96 10 15 95 25 91 99 13 70 74 37 60 78 4 58 44 98

- Cube 6:

59 58 25 57 125 66 43 33 105 63 85 56 2 87 10 60

35 116 107 21 53 109 67 18 69 3 47 113 94 90 34

19 84 6 92 121 55 124 48 16 32 41 62 89 24 7 75 4

79 12 100 74 65 76 13 8 120 15 5 119 39 38 114 96

98 9 97 88 118 31 22 28 99 17 54 72 81 101 86 123

52 11 115 61 45 71 103 44 73 102 50 104 36 93 51

20 1 117 108 37 122 82 106 111 14 23 70 30 64 112

80 46 110 78 95 42 29 83 26 27 40 49 77 68 91

- Cube 7:

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

- Cube 8:

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

- Cube 9:

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

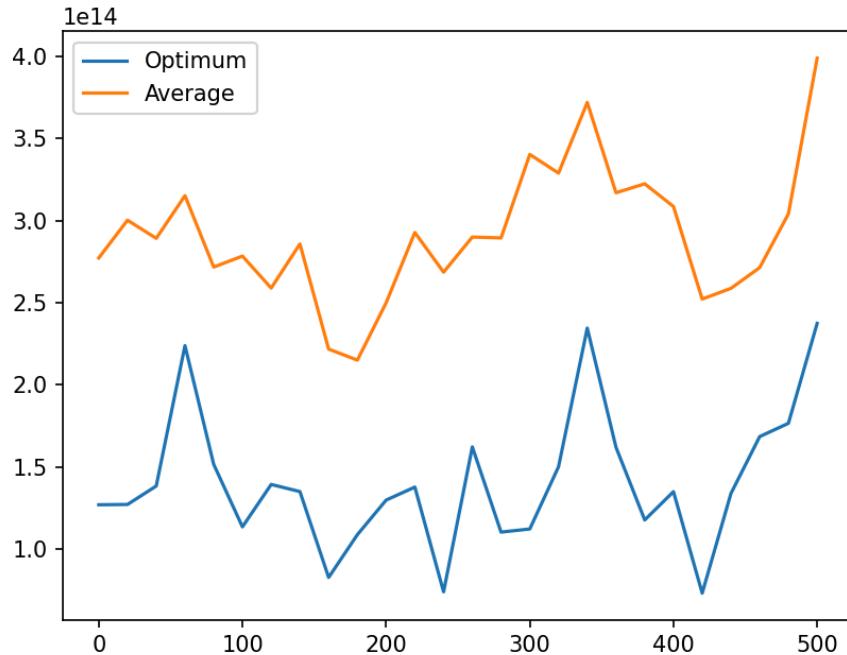
- Cube 10:

59 15 7 57 125 66 119 18 120 45 61 77 60 50 68 76
67 14 107 91 9 93 99 56 28 26 49 36 81 73 118 5 94
88 96 37 84 47 55 29 53 89 62 48 8 32 75 2 102 112
25 78 40 100 12 51 63 41 58 104 27 1 3 6 92 65 42

105 54 52 101 30 38 124 23 97 117 11 123 106 21
 95 72 121 31 86 109 103 85 116 108 115 39 114 71
 98 24 4 113 20 34 82 43 16 19 122 70 33 64 87 80
 46 110 79 74 13 83 17 10 69 22 35 111 90 44

- Nilai Objective Function Akhir:
 - Fitness Cube 1: 377780255809725
 - Fitness Cube 2: 349046135407257
 - Fitness Cube 3: 577057987029426
 - Fitness Cube 4: 354105552460321
 - Fitness Cube 5: 438183422975652
 - Fitness Cube 6: 237358877401221
 - Fitness Cube 7: 259793564056224
 - Fitness Cube 8: 314991253260650
 - Fitness Cube 9: 539623328488898
 - Fitness Cube 10: 539623328488898
 - Average Fitness: 398756370537827
 - Optimum: 237358877401221

- Plot:



- Durasi: 4824108 microsekon

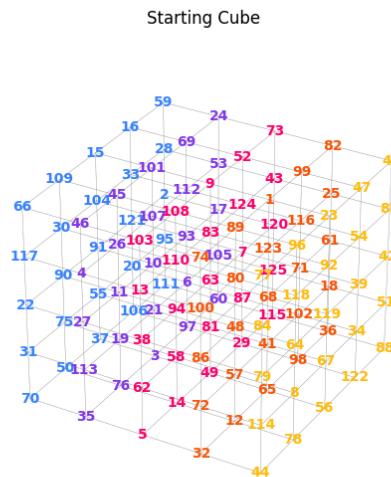
2.2.4 Hill-Climbing with Sideways Move

- Eksperimen Jumlah Iterasi

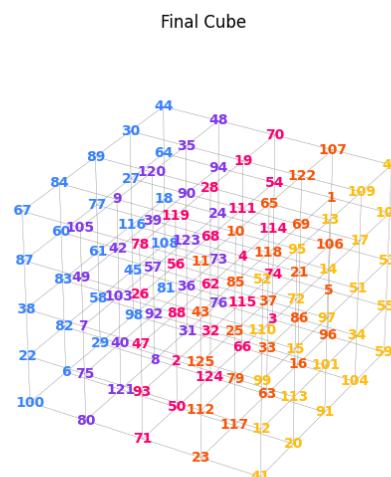
- Variansi 1:

- Sampel 1:

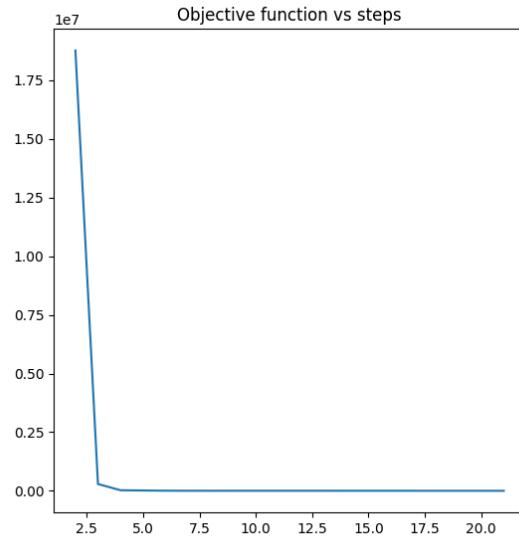
- State Awal:



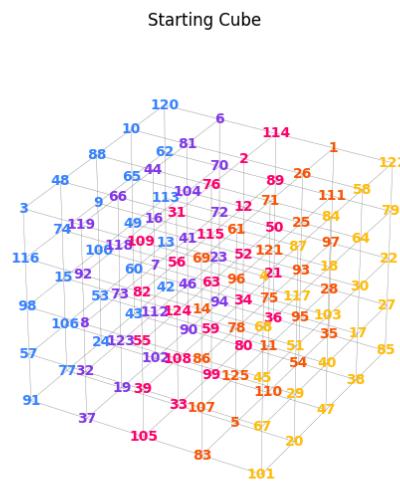
- State Akhir:



- Nilai objektif akhir yang dicapai: 435
- Banyaknya iterasi: 21
- Plot nilai objektif terhadap banyaknya iterasi:

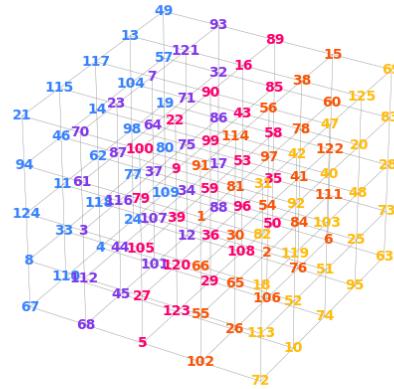


- Durasi pencarian: 1.656254 sekon
- Sampel 2:
- State Awal:

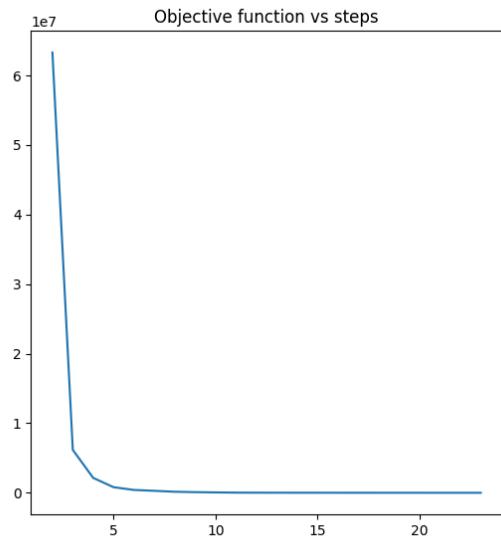


- State Akhir:

Final Cube

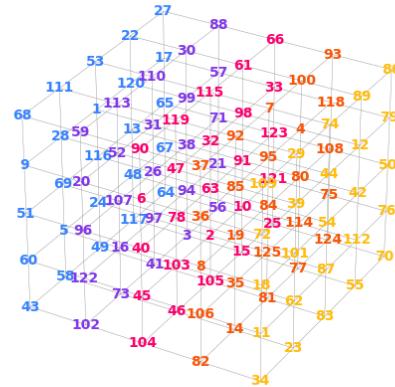


- Nilai objektif akhir yang dicapai: 1464
- Banyaknya iterasi: 23
- Plot nilai objektif terhadap banyaknya iterasi:



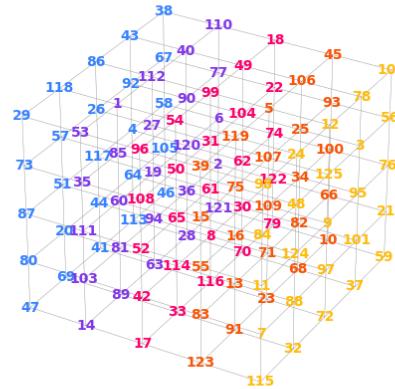
- Durasi pencarian: 2.181519 sekon
- Sampel 3:
 - State Awal:

Starting Cube

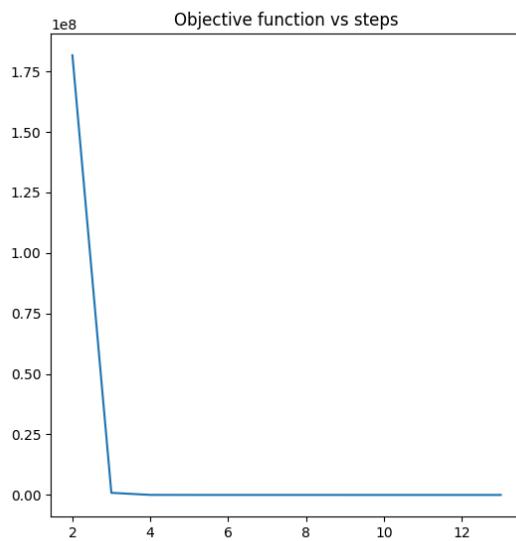


- State Akhir:

Final Cube



- Nilai objektif akhir yang dicapai: 695
- Banyaknya iterasi: 13
- Plot nilai objektif terhadap banyaknya iterasi:



- Durasi pencarian: 1.204370 sekon

2.2.5 Analisis Hasil Eksperimen

- Seberapa dekat tiap-tiap algoritma bisa mendekati global optima dan mengapa hasilnya demikian?
 - Genetic Algorithm:
Genetic Algorithm masih seringkali sangat jauh dari global optima. Hal ini dikarenakan fungsi pembangun successor Genetic Algorithm (Selection, Cross Over, Mutation) masih terlalu acak/random. Akibatnya, successor tidak cenderung membaik menuju optima layaknya hill climbing. Penggunaan *ordered-cross-over* untuk memenuhi *constraint* angka unik juga membatasi kelebihan Genetic Algorithm, yaitu penukaran informasi-informasi penting. Akibatnya, informasi dari *cross over* iterasi sebelumnya dapat hilang akibat *cross over* berikutnya.
 - Simulated Annealing:
Simulated Annealing relatif dekat dengan global optima. Hal ini dikarenakan cara kerja algoritma simulated annealing yang awalnya mengeksplor hampir seluruh tetangga (mengaproksimasi *complete search*), namun pada akhirnya mengaproksimasi *steepest-ascent hill*

climbing. Untuk suksesor dengan nilai *objective function* yang lebih buruk, algoritma akan berpindah ke *state successor* dengan suatu probabilitas yang telah dijelaskan sebelumnya. Cara kerja ini memungkinkan algoritma untuk mempertahankan informasi penting sambil berpindah ke *state* yang lebih baik. Selain itu, cara kerja algoritma ini didesain untuk menghindari lokal optima, yang turut membantu kondisi hasil algoritma yang dekat dengan global optima.

- Steepest Ascent Hill-Climbing:

Steepest Ascent Hill-Climbing lebih sering terjebak pada local optima dan flat local optima, sehingga jarang sekali untuk menemukan solusi yang merupakan global optimum. Hal ini dikarenakan cara kerja dalam pemilihan neighbor pada algoritma Steepest Ascent Hill-Climbing yang mirip dengan algoritma Greedy pada umumnya.

- Hill-Climbing with Sideways Move:

Mirip seperti Steepest Ascent Hill-Climbing, dengan perbedaan yang terletak pada kemampuannya untuk tidak terjebak di flat local optima. Jika menemukan flat local optima, algoritma ini akan tetap memilih neighbor tersebut dengan harapan akan menemukan global optima. Algoritma ini relatif lebih mendekati global optima jika dibandingkan dengan Steepest Ascent Hill-Climbing. Jika dibandingkan dengan algoritma lain seperti Genetic Algorithm dan Simulated Annealing, algoritma ini masih relatif lebih jauh dari global optima.

- Bagaimana perbandingan hasil pencarian tiap-tiap algoritma dengan algoritma local search yang lain?

- Genetic Algorithm:

Dibandingkan algoritma yang lain, algoritma Genetic Algorithm menghasilkan hasil yang sangat buruk. Hal ini karena pembentukan *child* pada Genetic Algorithm terlalu acak sehingga cenderung memperburuk hasil. Genetic Algorithm juga dapat *stuck* apabila populasi terlalu kecil.

- Simulated Annealing:

Dibandingkan algoritma yang lain, algoritma *simulated annealing* menghasilkan hasil yang relatif baik. Hal ini karena cara kerja algoritma yang menghindari lokal optima.

- Steepest Ascent Hill-Climbing:

Jika hasil eksperimen algoritma ini dibandingkan dengan algoritma lain, dapat terlihat hasil pencarian yang ditemukan oleh algoritma ini tidak bagus. Hal ini dikarenakan penentuan pengambilan neighbor yang logikanya mirip dengan algoritma Greedy, sehingga seringkali terjebak pada local optima.

- Hill-Climbing with Sideways Move:

Jika hasil eksperimen algoritma ini dibandingkan dengan algoritma saudaranya, Steepest Ascent Hill-Climbing, hasilnya akan lebih baik. Tetapi jika dibandingkan dengan algoritma Simulated Annealing, lebih buruk. Alasannya adalah algoritma ini berhasil melampaui flat local optima, tetapi tidak mampu melampaui valley untuk mencapai global optima teoritis di seberangnya.

- Bagaimana perbandingan durasi proses pencarian tiap algoritma relatif terhadap algoritma lainnya?

- Genetic Algorithm:

Dibandingkan dengan algoritma Simulated Annealing, Genetic Algorithm memiliki durasi yang lebih lama. Hal ini karena Genetic Algorithm memiliki beberapa proses tambahan, yaitu Selection, Cross Over, dan Mutation, sehingga Genetic Algorithm membutuhkan waktu yang lebih lama. Dibandingkan dengan algoritma Hill Climbing, algoritma Genetic Algorithm juga lebih lambat karena algoritma akan berjalan sebanyak jumlah iterasi, sedangkan algoritma *hill-climbing* akan berhenti ketika sudah berada di lokal optima. Namun, relatif terhadap perbedaan jumlah iterasi, Genetic Algorithm dapat lebih cepat. Hal ini karena untuk fungsi pembangun successor, Hill Climbing harus membangun seluruh kemungkinan kubus tetangga, sedangkan Genetic Algorithm hanya perlu

menerapkan Selection, Cross Over, dan Mutation sebanyak jumlah populasi.

- Simulated Annealing:

Dibandingkan dengan algoritma *hill-climbing*, algoritma ini lebih lambat. Hal ini karena algoritma akan berjalan hingga temperatur kurang dari minimum temperatur, sedangkan algoritma *hill-climbing* akan berhenti ketika sudah berada di lokal optima. Namun, jika dibandingkan algoritma *genetic algorithm*, algoritma ini cenderung lebih cepat.

- Steepest Ascent Hill-Climbing:

Jika dibandingkan dengan algoritma lainnya, algoritma ini adalah yang paling cepat dalam menemukan solusi (walaupun solusi yang ditemukan bukan global optima, melainkan local optima).

- Hill-Climbing with Sideways Move:

Jika dibandingkan dengan algoritma Steepest Ascent Hill-Climbing, algoritma ini lebih lambat, karena masih mau mengambil neighbor yang memiliki nilai yang sama dengan current state untuk keluar dari flat local optima.

- Seberapa konsisten hasil akhir yang didapatkan dari tiap-tiap eksperimen yang dilakukan?

- Genetic Algorithm:

Dari hasil eksperimen, dapat dilihat bahwa hasil akhir Genetic Algorithm tidak konsisten. Hal ini dikarenakan fungsi pembangun Genetic Algorithm yang terlalu acak. Populasi yang terlalu sedikit juga dapat menimbulkan Genetic Algorithm untuk *stuck* karena menghasilkan *child* yang sama.

- Simulated Annealing:

Dari hasil eksperimen, dapat dilihat bahwa algoritma *simulated annealing* konsisten mendapatkan nilai *objective function* dalam kisaran 30-50. Ini artinya algoritma *simulated annealing* selain mampu mendapatkan hasil yang baik, juga mampu untuk mendapatkan hasil konsisten. Hal ini tercapai karena cara kerja algoritma yang menghindari terjebak dalam lokal optima.

- Steepest Ascent Hill-Climbing:

Dari hasil eksperimen, dapat dilihat bahwa algoritma Steepest Ascent Hill-Climbing konsisten mendapatkan nilai fungsi objektif di kisaran 4000. Hal ini mengindikasikan seringnya algoritma ini terjebak di local optima.
 - Hill-Climbing with Sideways Move:

Berbeda dengan Steepest Ascent Hill-Climbing, dari hasil eksperimen dapat dilihat hasil yang lebih bervariasi dari algoritma ini, yaitu dari 435 sampai 1464. Hal ini mengindikasikan kemampuan algoritma ini dalam membebaskan diri dari jebakan flat local optima. Hasil yang bervariasi ini dapat dijelaskan dengan ketidakmampuan algoritma ini untuk mengambil risiko demi mencapai global optima.
- Bagaimana pengaruh banyak iterasi dan jumlah populasi terhadap hasil akhir pencarian pada Genetic Algorithm?
 - Pengaruh Jumlah Populasi:

Dalam Genetic Algorithm, populasi mempengaruhi potensi terjadinya *stuck*. *Stuck* yang dimaksud adalah ketika algoritma menghasilkan *child* yang sama semua. Hal ini terjadi ketika proses *selection* memilih *parent* yang sama sehingga menghasilkan *child* yang sama. Seiring berjalannya iterasi, jumlah individu yang sama semakin meningkat sampai semua individu sama dengan satu sama lain. Semakin kecil populasi, semakin besar kemungkinan algoritma *stuck* karena menghasilkan *child* yang sama. Sebaliknya, semakin besar populasi, semakin kecil kemungkinan algoritma *stuck*. Hal ini karena dengan meningkatnya jumlah populasi, semakin besar kandidat untuk *selection* sehingga lebih kecil kemungkinan algoritma untuk memilih *parent* yang sama. Dengan banyaknya kandidat, semakin besar juga algoritma untuk memperbaiki populasi yang awalnya banyak individu yang sama menjadi kembali berbeda.
 - Pengaruh Banyak Iterasi:

Dalam Genetic Algorithm, banyak iterasi tidak terlalu berpengaruh pada kinerja algoritma. Hal ini dikarenakan Genetic Algorithm tidak

“mencondong” menuju *state optimum* layaknya hill climbing atau simulated annealing. Sebaliknya, fungsi pembangun successor Genetic Algorithm (Selection, Cross Over, Mutation) masih terlalu acak/random sehingga Genetic Algorithm masih terlalu sering memperburuk *state* yang awalnya baik. Dalam hal ini, Genetic Algorithm lebih serupa dengan *Random Walk*. Akibatnya penambahan iterasi tidak efektif dalam meningkatkan kinerja Genetic Algorithm. Penggunaan *ordered-cross-over* untuk memenuhi *constraint* angka unik juga berpengaruh terhadap keacakan fungsi pembangun successor Genetic Algorithm.

- Bagaimana pengaruh minimum *temperature* dan *cooling rate* pada algoritma *simulated annealing*?

- Pengaruh minimum *temperature*:

Dari hasil eksperimen terlihat bahwa semakin kecil minimum *temperature*, hasil algoritma akan semakin baik. Selain itu, semakin kecil minimum *temperature*, semakin lama durasi algoritma. Hal ini terjadi karena dengan minimum *temperature* yang lebih kecil, algoritma memiliki waktu yang lebih banyak untuk meng-*consider* suksesor dari suatu *state*.

- Pengaruh *cooling rate*:

Semakin besar *cooling rate*, semakin baik hasil algoritma. Namun, semakin besar *cooling rate*, algoritma membutuhkan waktu semakin lama. Hal ini karena dengan nilai *cooling rate* yang lebih besar, algoritma memiliki lebih banyak waktu untuk mengeksplorasi ruang fungsi objektif, yang akan berdampak pada hasil yang lebih baik.

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Untuk mengatasi keterbatasan algoritma *Steepest Ascent Hill-Climbing* yang cenderung terjebak di solusi *local optimum* dan *flat local optimum*, dan keterbatasan algoritma *Hill-Climbing with Sideways Move* yang tidak mampu mengambil risiko untuk mencapai global optima, penggunaan metode seperti *Simulated Annealing* atau *Genetic Algorithm* dapat dipertimbangkan. Kedua algoritma ini dapat membantu menghindari jebakan *local optimum* dan meningkatkan peluang menemukan solusi *optimal global*.

3.2 Saran

Agar proses pencarian solusi lebih efektif, perlu dilakukan efisiensi kode pada fungsi-fungsi perhitungan seperti *distanceEff* dan *distanceMat*, sehingga proses pencarian solusi bisa lebih cepat. Pengoptimalan fungsi-fungsi ini akan sangat berguna ketika jumlah data yang diproses semakin besar.

PEMBAGIAN TUGAS

Nama/NIM	Pembagian Tugas
Ariel Herfrison/13522002	Implementasi dan Eksperimen Genetic Algorithm.
Andhika Tantyo Anugrah/13522094	Implementasi dan Eksperimen Steepest Ascent Hill-Climbing Algorithm dan Hill-Climbing with Sideways Move.
Kristo Anugrah/13522024	Implementasi dan Eksperimen Simulated Annealing Algorithm.
Chelvadinda/13522154	Membuat laporan.

REFERENSI

1. <https://www.magischvierkant.com/three-dimensional-eng/magic-features/>
2. Russell, S. J., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Pearson.
3. Saeidi, S. (2018). Solving the Rubik's cube using simulated annealing and genetic algorithm.