

Sorting (Lanjutan)

Gravitation Sort & Bubble Sort

Pengurutan Gravitasi (Gravitation Sort)

- Algoritma GSA (Gravitational Search Algorithm) adalah salah satu algoritma optimasi yang dapat digunakan untuk pengambilan keputusan.
- Algoritma ini terinspirasi dari sebuah teori hukum gravitasi yaitu teori Newton. Inti dari teori tersebut adalah “Setiap partikel yang ada di dunia akan saling menarik satu sama lain dengan kekuatan yang berbanding lurus dengan massa partikel dan berbanding terbalik dengan jarak antar partikel tersebut”.
- Mirip dengan Bubble Sort tetapi dimulai dari elemen pertama (paling kiri) dan dibandingkan dengan elemen di belakangnya (sebelah kanannya), sehingga pada akhir langkah pertama diperoleh elemen terakhir sudah dalam posisi terurut. Demikian seterusnya.

Pengurutan Gravitasi (Gravitation Sort)

- Contoh : Urutkan naik elemen-elemen array A = [6, 2, 9, 3, 7, 4]

Lokasi	1	2	3	4	5	6
Data	6	2	9	3	7	4

Langkah 1

Lokasi	1	2	3	4	5	6
Awal	6*	2*	9	3	7	4
1	2	6*	9*	3	7	4
2	2	6	9*	3*	7	4
3	2	6	3	9*	7*	4
4	2	6	3	7	9*	4*
5	2	6	3	7	4	<u>9</u>

Langkah 2

Lokasi	1	2	3	4	5	6
Awal	2*	6*	3	7	4	<u>9</u>
1	2	6*	3*	7	4	<u>9</u>
2	2	3	6*	7*	4	<u>9</u>
3	2	3	6	7*	4*	<u>9</u>
4	2	3	6	4	<u>7</u>	<u>9</u>

Langkah 3

Lokasi	1	2	3	4	5	6
Awal	2*	3*	6	4	<u>7</u>	<u>9</u>
1	2	3*	6*	4	<u>7</u>	<u>9</u>
2	2	3	6*	4*	<u>7</u>	<u>9</u>
3	2	3	4	<u>6</u>	<u>7</u>	<u>9</u>

Pengurutan Gravitasi (Gravitation Sort)

Langkah 4

Lokasi	1	2	3	4	5	6
Awal	2*	3*	4	<u>6</u>	<u>7</u>	<u>9</u>
1	2	3*	4*	<u>6</u>	<u>7</u>	<u>9</u>
2	2	3	<u>4</u>	<u>6</u>	<u>7</u>	<u>9</u>

Langkah 5

Lokasi	1	2	3	4	5	6
Awal	2*	3*	<u>4</u>	<u>6</u>	<u>7</u>	<u>9</u>
1	2	<u>3</u>	<u>4</u>	<u>6</u>	<u>7</u>	<u>9</u>

Pengurutan Gelembung (*Bubble sort*)

- Diinspirasi oleh gelembung sabun yang ada dipermukaan air, dimana benda yang berat akan terbenam dan yang ringan akan terapung
- Bila pengurutan dengan acuan ascending : elemen yang bernilai besar akan “dibenamkan” melalui proses perbandingan antar elemen yang bersebelahan dan proses pertukaran
 - Proses pertukaran dilakukan sebanyak $N-1$ langkah, dimana N adalah ukuran array
 - Pada akhir setiap langkah ke l , array $L[1..N]$ akan terdiri atas dua bagian yaitu :
 - Yang sudah terurut
 - Yang belum terurut
 - Setelah langkah terakhir diperoleh array $L[1..N]$ yang terurut ascending

Pengurutan Gelembung (*Bubble sort*)

- Contoh : diurutkan secara ascending, $N = 6$

Lokasi	1	2	3	4	5	6
Data	25	27	10	8	76	21

Langkah 5

Lokasi	1	2	3	4	5	6
Awal	25*	27*	10	8	76	21
1	25	27*	10*	8	76	21
2	25	10	27*	8*	76	21
3	25	10	8	27*	76*	21
4	25	10	8	27	76*	21*
5	25	10	8	27	21	<u>76</u>

Langkah 4

Lokasi	1	2	3	4	5	6
Awal	25*	10*	8	27	21	<u>76</u>
1	10	25*	8*	27	21	<u>76</u>
2	10	8	25*	27*	21	<u>76</u>
3	10	8	25	27*	21*	<u>76</u>
4	10	8	25	21	<u>27</u>	<u>76</u>

Langkah 3

Lokasi	1	2	3	4	5	6
Awal	10*	8*	25	21	<u>27</u>	<u>76</u>
1	8	10*	25*	21	<u>27</u>	<u>76</u>
2	8	10	25*	21*	<u>27</u>	<u>76</u>
3	8	10	21	<u>25</u>	<u>27</u>	<u>76</u>

Pengurutan Gelembung (*Bubble sort*)

Langkah 2

Lokasi	1	2	3	4	5	6
Awal	8*	10*	21	<u>25</u>	<u>27</u>	<u>76</u>
1	8	10*	21*	<u>25</u>	<u>27</u>	<u>76</u>
2	8	10	<u>21</u>	<u>25</u>	<u>27</u>	<u>76</u>

Langkah 1

Lokasi	1	2	3	4	5	6
Awal	8*	10*	21	25	<u>27</u>	<u>76</u>
1	8	<u>10</u>	<u>21</u>	<u>25</u>	<u>27</u>	<u>76</u>

Pengurutan Gelembung (*Bubble sort*)

Algoritma:

Deklarasi

I : bilangan bulat {untuk langkah}
J : bilangan bulat {indek}
Temp : bilangan bulat {untuk penampung sementara}
L : Array [1 ..N]
N : bilangan bulat {jumlah elemen array}

Deskripsi

```
For I ← (N-1) downto 1 do
  For J ← 1 to I do
    If L[J] > L [J+1] then
      Temp ← L[J]
      L[J] ← L[J+1]
      L[J+1] ← temp
    Endif
  Endfor
Endfor
```