**NAMA: FITRI ARISKA**

**NIM : 20051397082**

**KELAS: D4 MI 2020B**

1. **Selection Sort**

Proses pengurutan dengan metode selection sort secara ascending dapat dijelaskan sebagai berikut:Contoh: Elemen array / larik dengan N=6 buah elemen dibawah ini.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 9 | 1 | 2 | 5 | 6 | 4 |

Langkah 1: Cari elemen maksimum di dalam larik L[1..6] maks = L[1] = 9

Tukar maks dengan L[N], hasil akhir langkah 1:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 2 | 5 | 6 | **9** |

Langkah 2: (berdasarkan susunan larik hasil langkah 1)

Cari elemen maksimum di dalam larik L[1..5] maks = L[5] = 6

Tukar maks dengan L[5],hasil akhir langkah 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 2 | 5 | **6** | **9** |

Langkah 3: (berdasarkan susunan larik hasil langkah 2)

Cari elemen maksimum di dalam larik L[1..4] maks = L[4] = 5

Tukar maks dengan L[4],hasil akhir langkah 3:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 2 | **5** | **6** | **9** |

Langkah 4: (berdasarkan susunan larik hasil langkah 3)

Cari elemen maksimum di dalam larik L[1..3] maks = L[3] = 4

Tukar maks dengan L[1],hasil akhir langkah 4:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 1 | **4** | **5** | **6** | **9** |

Langkah 5: (berdasarkan susunan larik hasil langkah 4)

Cari elemen maksimum di dalam larik L[1..2] maks = L[2] = 2

Tukar maks dengan L[2],hasil akhir langkah 5:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | **2** | **4** | **5** | **6** | **9** |

1. **Insertion Sort**

Proses pengurutan dengan metode penyisipan langsung (straight) dapat dijelaskan sebagai berikut : Data dicek satu per satu mulai dari yang kedua sampai dengan yang terakhir. Apabila ditemukan data yang lebih kecil daripada data sebelumnya, maka data tersebut disisipkan pada posisi yang sesuai. Akan lebih mudah apabila membayangkan pengurutan kartu. Pertama-tama anda meletakkan kartukartu tersebut di atas meja, kemudian melihatnya dari kiri ke kanan. Apabila kartu di sebelah kanan lebih kecil daripada kartu di sebelah kiri, maka ambil kartu tersebut dan sisipkan di tempat yang sesuai.

Contoh: Elemen array / larik dengan N=6 buah elemen dibawahini.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 9 | 1 | 2 | 5 | 6 | 4 |

Langkah 1: Elemen L[1] dianggap sudah terurut

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **9** | 1 | 2 | 5 | 6 | 4 |

Langkah 2: (berdasarkan susunan larik pada langkah 1)

Cari posisi yang tepat untuk L[2] pada L[1..2],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **9** | 2 | 5 | 6 | 4 |

Langkah 3: (berdasarkan susunan larik pada langkah 2)

Cari posisi yang tepat untuk L[3] pada L[1..3],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **9** | 5 | 6 | 4 |

Langkah 4: (berdasarkan susunan larik pada langkah 3)

Cari posisi yang tepat untuk L[4] pada L[1..4],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **5** | **9** | 6 | 4 |

Langkah 5: (berdasarkan susunan larik pada langkah 4)

Cari posisi yang tepat untuk L[5] pada L[1..5],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **5** | **6** | **9** | 4 |

Langkah 6: (berdasarkan susunan larik pada langkah 5)

Cari posisi yang tepat untuk L[6] pada L[1..6],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **5** | **6** | **4** | **9** |

Langkah 7: (berdasarkan susunan larik pada langkah 6)

Cari posisi yang tepat untuk L[5] pada L[1..5],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **5** | **4** | **6** | **9** |

Langkah 8: (berdasarkan susunan larik pada langkah 7)

Cari posisi yang tepat untuk L[4] pada L[1..4],diperoleh :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **4** | **5** | **6** | **9** |

**Kesimpulan** : Pengurutan data (sorting) didefinisikan sebagai suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu. Metode penyisipan (Insertion sort) bertujuan untuk menjadikan bagian sisi kiri array terurutkan sampai dengan seluruh array berhasil diurutkan. Metode ini mengurutkan bilangan-bilangan yang telah dibaca; dan berikutnya secara berulang akan menyisipkan bilangan-bilangan dalam array yang belum terbaca ke sisi kiri array yang telah terurut. Lalu untuk cara kerja algoritma selection sort adalah pada elemen acuan pada indeks ke-i, akan memilih elemen dengan nilai paling kecil pada indeks berikutnya (ke-j=i+1 s/d ke-n) dan menukar elemen acuan dengan elemen terkecil.