
Simulasi Pengurutan Data Dengan Metode Seleksi

Indra Gunawan

STMIK IBBI

J Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548

E-mail: indragwn@yahoo.com

Abstrak

Pengurutan data atau *sorting* merupakan suatu proses dimana suatu susunan data yang semula dalam kondisi acak dapat menjadi urut, baik dari data terkecil sampai dengan data yang terbesar, atau sebaliknya dari data terbesar sampai dengan data terkecil. Terdapat beberapa metode yang cukup populer dalam proses pengurutan data tersebut dan salah satunya adalah metode seleksi. Pengurutan data dengan metode seleksi dapat dijelaskan dengan cara sederhana yaitu dengan cara mencari data terkecil dari susunan data awal dari data pertama sampai data terakhir. Setelah data terkecil ditemukan kemudian tukarkan data dengan data pertama. Setelah itu dicari lagi data terkecil tetapi dimulai dari data kedua sampai data terakhir dan setelah didapatkan kemudian tukarkan dengan data kedua. Begitu seterusnya hingga diperoleh data urut dari awal sampai akhir.

Kata kunci : Pengurutan data, seleksi, algoritma

Abstract

Sorting of data is a processing which an arrangement of the original data in the random condition can be sequential, the data either from the smallest to the biggest data, or otherwise of the largest data until the data is the smallest. There are several methods that are quite popular in the process of sorting the data and one of them is the method of selection. Sorting the data by the method of selection can be explained with a simple way of searching for data that is the smallest of the composition of the initial data from the first data to recent data. Once the data is the smallest discovered later exchanged data with the first data. After the data was sought again the smallest but starting from the second data until after the last data and then exchanged with the data obtained both. And so on until the data obtained sequence from beginning to end.

Keywords : Pengurutan data, seleksi, algoritma

1. Pendahuluan

Dalam pengurutan data, yang disimpan dalam pengingat utama komputer, ada aspek ekonomis yang perlu dipertimbangkan. Aspek ini antara lain menyangkut kapasitas pengingat yang tersedia. Aspek lain adalah dalam hal waktu, yaitu waktu yang diperlukan untuk melakukan permutasi sehingga semua data akhirnya menjadi terurut. Ukuran efisiensi yang baik bisa diperoleh dari banyaknya perbandingan dan perpindahan yang harus dilakukan. Angka – angka ini merupakan fungsi dari N yaitu banyaknya elemen yang akan diurutkan. Algoritma yang baik memerlukan perbandingan sebanyak $N \log N$ kali. Meskipun demikian kita akan melihat beberapa metode yang disebut dengan metode langsung (*straight method*), yang seluruhnya memerlukan N^2 perbandingan. Metode langsung ini bisa dikelompokkan menjadi tiga metode, yaitu penyisipan (*insertion*), seleksi (*selection*), dan penukaran (*exchange*).

Dari tiga metode tersebut, metode seleksi merupakan salah satu metode pengurutan yang cukup sederhana karena tidak melibatkan proses yang rumit dalam implementasinya. Untuk mempelajari metode tersebut diperlukan suatu cara yang dapat mempermudah kita dalam mempelajari metode seleksi tersebut dan salah satu cara yang paling efektif adalah dengan mensimulasikan cara kerja metode seleksi tersebut.

2. Metode Penelitian

Adapun tahapan-tahapan yang dilaksanakan dalam melakukan penelitian adalah sebagai berikut :

1. Penelitian kepustakaan (*library research*), yaitu membaca buku – buku yang berhubungan dengan metode – metode pengurutan data khususnya metode seleksi dan juga membaca buku – buku yang berhubungan dengan bahasa pemrograman yang dipergunakan untuk mengetahui fasilitas – fasilitas apa saja yang disediakan oleh bahasa pemrograman tersebut sehingga akan mempermudah dalam merancang aplikasi simulasi yang diinginkan.
-

2. Merancang aplikasi simulasi berdasarkan informasi – informasi yang diperoleh dari penelitian kepustakaan.
3. Menguji rancangan aplikasi untuk menemukan kesalahan – kesalahan yang mungkin terjadi untuk diperbaiki sampai tidak ditemukan lagi kesalahan.

2.1. Perancangan Sistem

2.1.1. Mekanisme Metode Seleksi

Cara kerja metode seleksi didasarkan pada pencarian elemen dengan nilai terkecil, kemudian dilakukan penukaran dengan elemen ke I. Secara singkat, metode ini bisa dijelaskan sebagai berikut. Pada langkah pertama, dicari data yang terkecil dari data pertama sampai data terakhir. Kemudian data terkecil tersebut kita tukar dengan data pertama. Dengan demikian, data pertama sekarang mempunyai nilai paling kecil dibandingkan dengan data yang lain. Pada langkah kedua, data terkecil kita cari mulai data kedua sampai data terakhir. Data terkecil yang kita peroleh ditukar dengan data kedua. Dengan demikian seterusnya sampai seluruh vector dalam keadaan terurutkan.

Dengan kata lain mekanisme seleksi dapat dituliskan sebagai berikut. Masukan dinyatakan sebagai vektor A (belum terurutkan), dan N (banyak elemen). Keluaran adalah vektor A yang sudah dalam keadaan terurutkan.

Langkah 0	Baca vektor yang akan diurutkan (dalam program utama).
Langkah 1	Kerjakan langkah 2 sampai 4 untuk I = 1 sampai N – 1
Langkah 2	Tentukan : Lok = 1 Kerjakan langkah 3 untuk J = I + 1 sampai N
Langkah 3	(Mencari data terkecil) Test : apakah A[Lok] > A[J] ? Jika ya, tentukan : Lok = J.
Langkah 4	Tukarkan nilai A[Lok] dengan A[I].
Langkah 5	Selesai

Metode ini secara garis besar merupakan kebalikan dari metode penyisipan langsung. Dalam setiap langkah pada metode penyisipan langsung kita hanya memperhatikan satu elemen dari sumber dan semua elemen dari larik tujuan untuk menentukan posisinya yang tepat; sehingga sering disebut dengan *one source-multiple destinations*. Dalam metode seleksi terjadi sebaliknya yaitu kita memperhatikan semua elemen dalam larik sumber untuk menentukan elemen terkecil yang akan kita tempatkan pada tujuan; sehingga sering disebut dengan *multiple sources-one destination*. Dalam metode ini banyaknya perbandingan (untuk mencari elemen dengan nilai terkecil) besarnya tak gayut terhadap urutan semula. Banyaknya perbandingan adalah sebesar :

$$C = (N^2 - N) / 2$$

2.1.2. Rancangan Form Aplikasi

Aplikasi rancangan memiliki 2 buah *form* yakni *form* simulasi dan *form* masuk data. Kedua form tersebut akan dijelaskan sebagai berikut :

1. Form Simulasi

Form simulasi (seperti Gambar 1) adalah *form* yang tampil pertama sekali sewaktu program dijalankan. Menu – menu yang terdapat pada *form* ini adalah “Data” dan “Selesai”. Menu “Data” digunakan untuk menampilkan form untuk memasukkan data yang akan diurutkan. Menu “Selesai” digunakan untuk terminasi dari program. Bidang yang terdapat label “**POSISI**” dan “**DATA**” digunakan sebagai bidang untuk mem-visualisasi-kan data – data yang sedang diurutkan. Label “**POSISI**” menunjukkan posisi atau nomor urut data. Sedangkan label “**DATA**” menunjukkan data – data yang sedang diurutkan. Tabel Pengamatan Susunan Data digunakan untuk mencatat susunan data dari awal simulasi hingga akhir simulasi. Option “Menaik” dan “Menurun” digunakan untuk memilih apakah data akan diurut secara menaik atau menurun. Label “**Jumlah Perbandingan**” menyatakan banyaknya perbandingan yang harus dilakukan untuk mengurutkan sejumlah data yang kita masukkan. Label “**Jumlah Pertukaran Data**” menyatakan banyaknya pertukaran data yang terjadi dari awal hingga akhir pengurutan. *Scroll bar* pada label “**Kecepatan Simulasi**” digunakan untuk mengatur kecepatan simulasi. Tombol “**Simulasikan**” digunakan untuk menjalankan simulasi setelah data dimasukkan. Tombol “**Simulasikan**” akan berubah menjadi tombol “**Batal**” setelah diklik. Tombol “**Batal**” ini berfungsi untuk membatalkan simulasi yang sedang berjalan. Tombol

“Berhenti” digunakan untuk menghentikan simulasi sejenak untuk kemudian dapat dilanjutkan kembali. Tombol “Berhenti” hanya akan aktif sewaktu simulasi dijalankan.

Gambar 1. Form Simulasi

2. Form Masuk Data

Form (seperti Gambar 2) ini digunakan untuk memasukkan data yang akan diurutkan. Form ini dimunculkan dengan cara memilih menu “Data” pada form simulasi. Jumlah data yang dapat dimasukkan adalah antara 2 buah sampai dengan 14 buah data. Jumlah data yang akan dimasukkan dapat ditentukan dengan cara memilih pada *combo box* “Jumlah Data”. Setelah memilih jumlah data yang akan dimasukkan maka tahap selanjutnya adalah memasukkan data – data yang akan diurutkan melalui *text box* “Data”. Setelah selesai klik tombol “Selesai” atau klik tombol “Batal” jika data tidak jadi dimasukkan.

Gambar 2. Form Masuk Data

3. Pembahasan dan Hasil

3.1. Pembahasan

3.1.1. Algoritma Metode Seleksi

Algoritma metode seleksi terbagi menjadi 2 jenis yaitu :

1. Algoritma Metode Seleksi Pengurutan Naik

Algoritma metode seleksi untuk pengurutan naik adalah sebagai berikut :

```
For I = 1 To (N - 1)
  Lok = I
  For J = (I + 1) To N
    If A(Lok) > A(J) Then
      Lok = J
    End If
  Next J
  Swap (A(I), A(Lok))
Next I
```

2. Algoritma Metode Seleksi Pengurutan Turun

Algoritma metode seleksi untuk pengurutan turun adalah sebagai berikut :

```
For I = 1 To (N - 1)
  Lok = I
  For J = (I + 1) To N
    If A(Lok) < A(J) Then
      Lok = J
    End If
  Next J
  Swap (A(I), A(Lok))
Next I
```

3.1.2. Algoritma Simulasi Metode Seleksi

Algoritma simulasi metode seleksi terbagi menjadi dua yaitu :

1. Algoritma Simulasi Urut Naik

Algoritma untuk simulasi urut naik adalah sebagai berikut :

```
Tampilkan susunan data mula - mula sebelum terurut.
I = 1
JlhDataTukar = 0
Do While (I <= N - 1) And Not Cancel
  Lok = I
  Geser gambar panah pertama menunjuk data ke-I dan
  Tampilkan nilai I dengan posisi dibawah gambar panah pertama.
  J = I + 1
  Do While J <= N And Not Cancel
    Geser gambar panah kedua menunjuk data ke-J dan
    Tampilkan nilai J dengan posisi dibawah gambar panah kedua.
    Tampilkan gambar panah ketiga pada posisi(1200 * (1.6 + 0.6 * (Lok - 2)),
    1080).
    Tampilkan info berikut : "Data ke-" & Lok & "(" & Data(Lok - 1) & ") > Data
    ke-" & J & "(" & Data(J - 1) & ") ?".
    If Data(Lok - 1) > Data(J - 1) Then
      Lok = J
      Geser gambar panah ketiga pada posisi(1200 * (1.6 + 0.6 * (Lok - 2)), 1935)
      Tampilkan Nilai Lok dengan posisi dibawah gambar panah ketiga.
      Tampilkan info berikut"-> Ya. Maka Lok = "& Lok
    Else
      Tampilkan info berikut:" -> Tidak".
    End If
    J = J + 1
  Loop
  If Lok <> I Then
    JlhDataTukar = JlhDataTukar + 1
    Tampilkan JlhPertukaranData.
    Tampilkan info berikut:"Data ke-" & I & " dan Data ke-" & Lok & " saling
    dipertukarkan.".
    Tampilkan gambar tangan pertama pada posisi(1200* (1.6 + 0.6 * (Lok - 2)),
    1080).
    Geser data ke-Lok ke bawah seiring dengan gambar tangan pertama sampai pada
    posisi(1200 * (1.6 + 0.6 * (Lok - 2)),1560).
    Geser gambar tangan pertama ke posisi dibawah data ke-I.
    Geser data ke-I ke bawah seiring dengan gambar tangan pertama sampai pada
    posisi Y = 2520.
    Tampilkan gambar tangan kedua disamping kanan data ke-Lok.
```

```

Geser data ke-Lok ke kiri sampai sejajar dengandata ke-I.
Tampilkan gambar tangan ketiga dibawah data ke-Lok.
Geser data ke-Lok ke atas seiring dengan gambar tangan ketiga sampai nilai Y
= 960.
Tampilkan gambar tangan keempat disamping kiridata ke-I.
Geser data ke-I ke kanan seiring dengan gambar tangan keempat hingga posisi
data ke-Lok sebelum digeser.
Tampilkan gambar tangan ketiga di bawah data ke-I.
Geser data ke-I ke atas seiring dengan gambar tangan ketiga hingga nilai Y =
1080.
Temp = Data(I - 1)
Data(I - 1) = Data(Lok - 1)
Data(Lok - 1) = Temp
Pertukarkan nilai antara Label ke-(I-1) dengan Label ke-(Lok-1).
Pertukarkan letak/posisi antara Label ke-(I-1)dengan Label ke-(Lok-1).
End If
Tampilkan susunan data untuk iterasi ke-I
I = I + 1

```

Loop

2. Algoritma Simulasi Urut Turun

Algoritma untuk simulasi urut turun adalah sebagai berikut :

```

Tampilkan susunan data mula - mula sebelum terurut.
I = 1
JlhDataTukar = 0
Do While (I <= N - 1) And Not Cancel
    Lok = I
    Geser gambar panah pertama menunjuk data ke-I dan
    Tampilkan nilai I dengan posisi dibawah gambar panahpertama.
    J = I + 1
    Do While J <= N And Not Cancel
        Geser gambar panah kedua menunjuk data ke-J dan
        Tampilkan nilai J dengan posisi dibawah gambar panah kedua.
        Tampilkan gambar panah ketiga pada posisi(1200 * (1.6 + 0.6 * (Lok - 2)), 1080).
        Tampilkan info berikut : "Data ke-" & Lok & "(" &Data(Lok -
        1) & ") < Data ke-" & J & "(" &Data(J - 1) & ") ?".
        If Data(Lok - 1) < Data(J - 1) Then
            Lok = J
            Geser gambar panah ketiga pada posisi(1200 * (1.6 + 0.6 * (Lok - 2)), 1935)
            Tampilkan Nilai Lok dengan posisi dibawah gambar panah ketiga.
            Tampilkan info berikut"-> Ya. Maka Lok = "& Lok
        Else
            Tampilkan info berikut:" -> Tidak".
        End If
        J = J + 1
    Loop
    If Lok <> I Then
        JlhDataTukar = JlhDataTukar + 1
        Tampilkan JlhPertukaranData.
        Tampilkan info berikut:"Data ke-" & I & " dan Data ke-" & Lok & " saling
        dipertukarkan.".
        Tampilkan gambar tangan pertama pada posisi(1200* (1.6 + 0.6 * (Lok - 2)),
        1080).
        Geser data ke-Lok ke bawah seiring dengan gambar tangan pertama sampai pada
        posisi(1200 * (1.6 + 0.6 * (Lok - 2)),1560).
        Geser gambar tangan pertama ke posisi dibawah data ke-I.
        Geser data ke-I ke bawah seiring dengan gambar tangan pertama sampai pada
        posisi Y = 2520.
        Tampilkan gambar tangan kedua disamping kanan data ke-Lok.
        Geser data ke-Lok ke kiri sampai sejajar dengan data ke-I.
        Tampilkan gambar tangan ketiga dibawah data ke-Lok.
        Geser data ke-Lok ke atas seiring dengan gambar tangan ketiga sampai nilai Y =
        960.
        Tampilkan gambar tangan keempat disamping kiridata ke-I.
        Geser data ke-I ke kanan seiring dengan gambar tangan keempat hingga posisi
        data ke-Lok sebelum digeser.
        Tampilkan gambar tangan ketiga di bawah data ke-I.
        Geser data ke-I ke atas seiring dengan gambar tangan ketiga hingga nilai Y =
        1080.
        Temp = Data(I - 1)
        Data(I - 1) = Data(Lok - 1)
        Data(Lok - 1) = Temp
        Pertukarkan nilai antara Label ke-(I-1) dengan Label ke-(Lok-1).
        Pertukarkan letak/posisi antara Label ke-(I-1)dengan Label ke-(Lok-1).
    
```

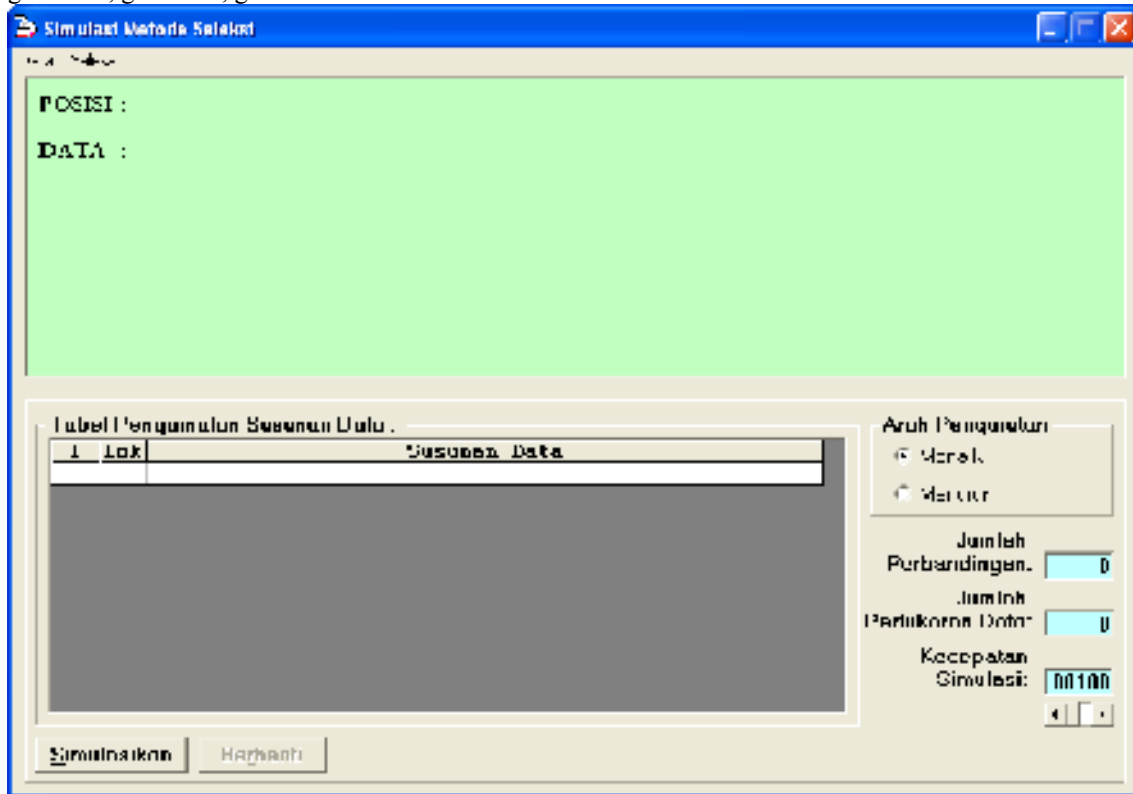
```

End If
Tampilkan susunan data untuk iterasi ke-I
I = I + 1
Loop

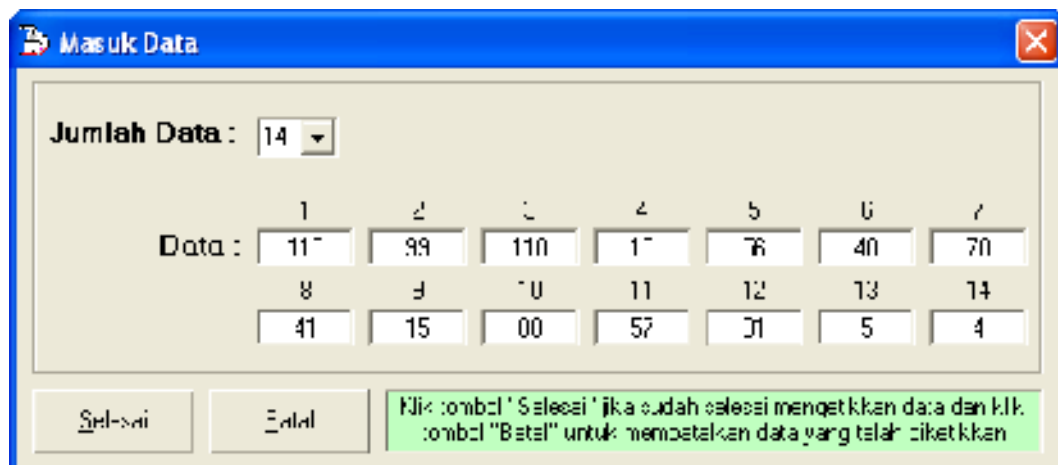
```

3.2. Hasil

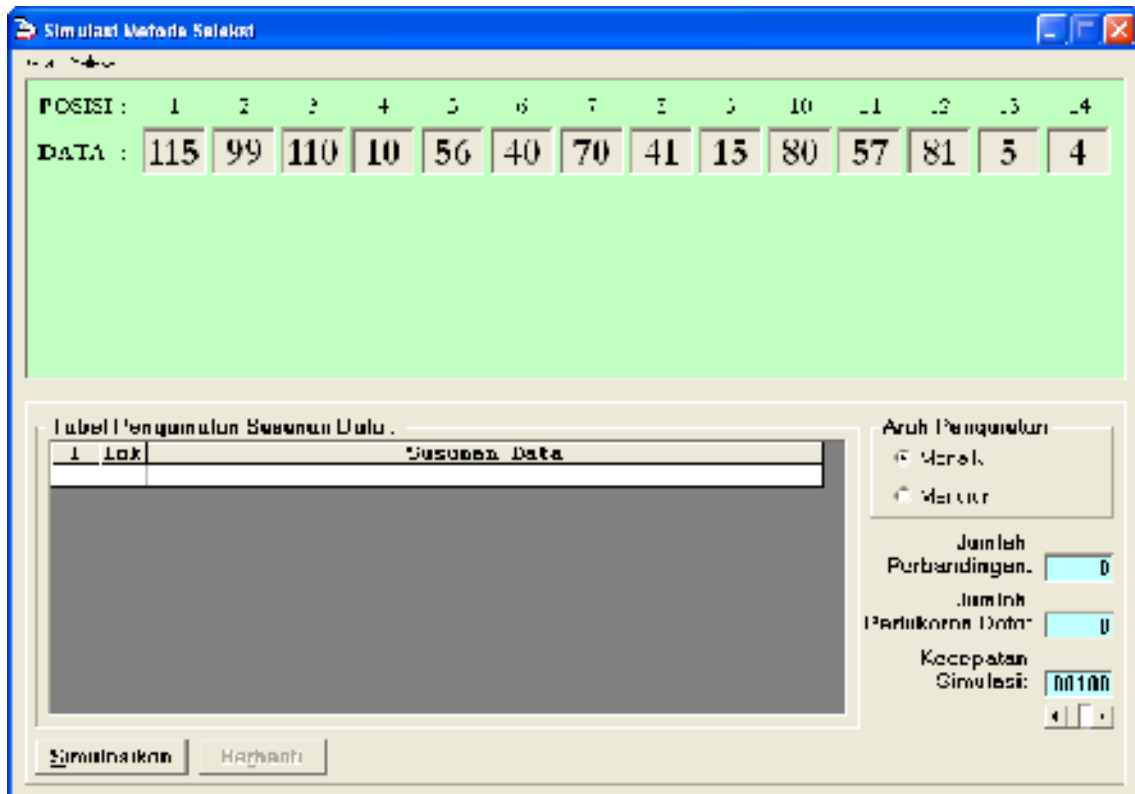
Hasil dari sistem simulasi yang dirancang dapat dilihat tampilan pada gambar 3, gambar 4, gambar 5, gambar 6, gambar 7 berikut ini :



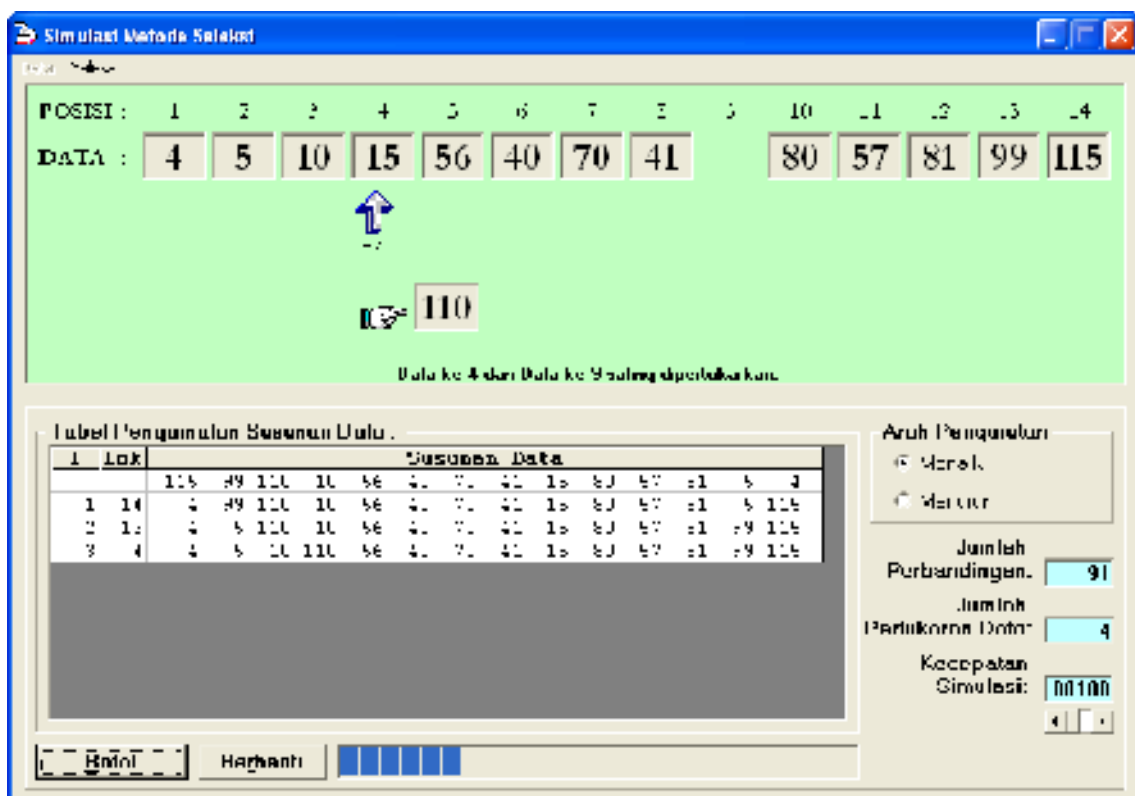
Gambar 3. Tampilan *Form* Simulasi Awal



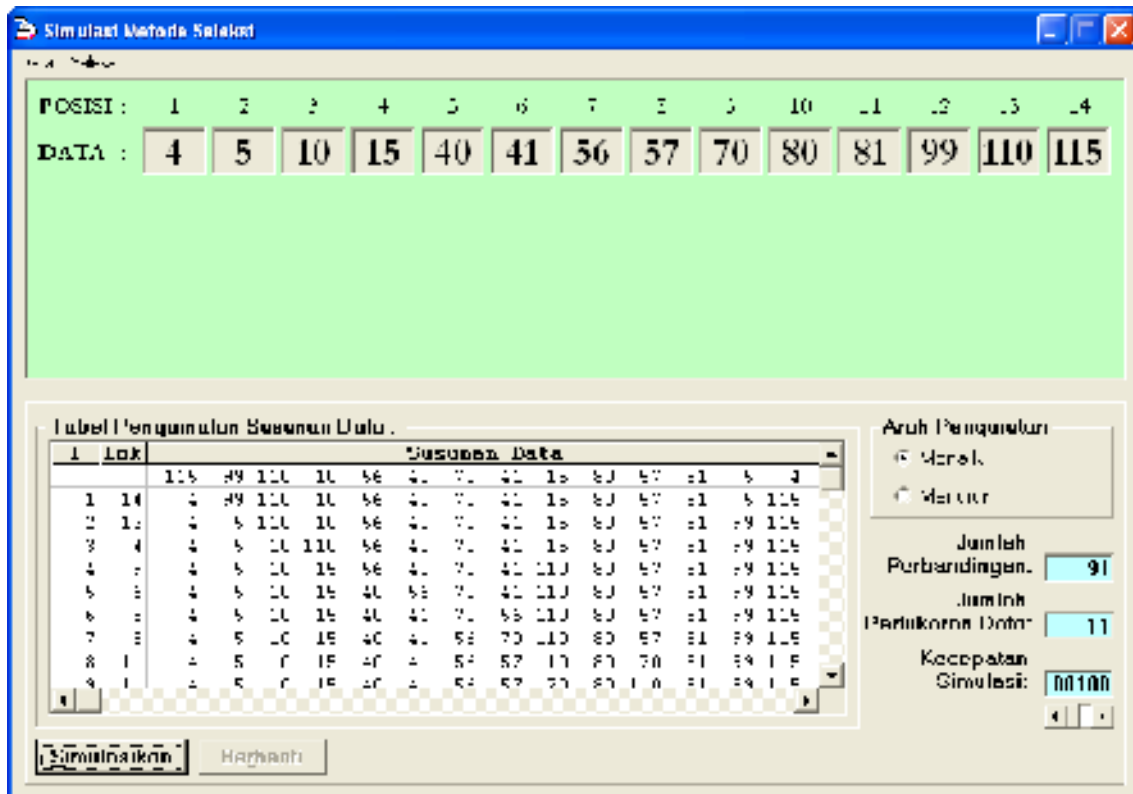
Gambar 4. Tampilan *Form* Masuk Data



Gambar 5. Tampilan *Form* Simulasi Saat Data Selesai Dimasukkan



Gambar 6. Tampilan *Form* Simulasi Saat Simulasi Berlangsung



Gambar 7. Tampilan Form Simulasi Setelah Simulasi Selesai

3. Kesimpulan

Adapun kesimpulan yang dapat diambil dari simulasi yang telah dilakukan adalah sebagai berikut:

1. Jumlah pertukaran data yang terjadi tergantung kepada susunan data yang akan diurutkan. Semakin acak susunan data maka semakin banyak terjadi proses pertukaran data.
2. Jumlah perbandingan data tidak tergantung kepada susunan data, jumlah perbandingan data hanya tergantung kepada banyaknya data yang akan diurutkan.
Jumlah perbandingan data yang terjadi dapat dihitung dengan rumus : $(N^2 - N) / 2$
3. Dilihat dari rumus jumlah perbandingan data tersebut, maka metode seleksi bukanlah metode pengurutan yang efisien karena akan memakan banyak waktu terutama jika digunakan untuk mengurut data dalam jumlah yang banyak.
4. Dengan adanya simulasi metode seleksi maka cara kerja metode tersebut jelas terlihat dan lebih mudah dipahami.

4. Saran

Adapun saran yang dapat diberikan adalah bahwa tulisan maupun aplikasi simulasi ini dapat dijadikan sebagai salah satu acuan untuk membahas ataupun merancang simulasi untuk metode – metode pengurutan lainnya. Dengan demikian akan semakin memperluas wawasan kita terhadap metode – metode pengurutan yang ada.

Daftar Pustaka

- [1] Halvorson, Michael, *Microsoft Visual Basic 6.0 Professional*, Terjemahan Adi Kurniadi, P.T. Elex Media Komputindo, Jakarta, 2000
- [2] P. Insap Santosa Ir, M.Sc., *Struktur Data Menggunakan Turbo Pascal 6.0*, ANDI OFFSET, Yogyakarta, 2000