

Teknik Selection Sort dan Bubble Sort Menggunakan Borland C++

Fauzi Eko Saputro¹, Fata Nidaul Khasanah^{2,*}

¹Sistem Informasi; STMIK Bina Insani; Jl. Siliwangi No 6 Rawa Panjang Bekasi Timur 17114 Indonesia. Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 82400924; e-mail:

fauzioko@gmail.com

²Teknik Informatika; STMIK Bina Insani; Jl. Siliwangi No 6 Rawa Panjang Bekasi Timur 17114 Indonesia. Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 82400924; e-mail:

fatanidaul@gmail.com

* Korespondensi: e-mail: fatanidaul@gmail.com

Diterima: 05 Januari 2018; Review: 15 Januari 2018; Disetujui: 2 Februari 2018

Cara sitasi: Saputro FE, Khasanah FN. 2018. Teknik *Selection Sort* dan *Bubble Sort* Menggunakan Borland C++. Jurnal Mahasiswa Bina Insani. 2 (2): 136 – 145

Abstrak: Peran algoritma dalam perangkat lunak atau pemrograman begitu penting, sehingga perlu memahami konsep dasar algoritma. Hampir setiap saat dalam kehidupan sehari - hari sering dijumpai permasalahan - permasalahan yang harus diselesaikan dengan melibatkan operasi pengurutan data. Begitu banyak logika pemrograman yang telah diciptakan, untuk kasus yang umum dan juga khusus. Proses pengurutan (*sorting*) dilakukan menggunakan teknik *selection sort*, dan *bubble sort*. Aplikasi pengurutan dibangun menggunakan Borland C++. Algoritma dibutuhkan agar suatu permasalahan dapat diselesaikan dengan lebih efektif dengan sedikit sumber daya yang dilibatkan dan lebih efisien dengan durasi waktu yang dibutuhkan sedikit (cepat). *Pseudocode* C++ yang dibuat berhasil melakukan proses pengurutan dari 3 kelompok, pertama kelompok 1 – 100, kedua 100 – 1000, ketiga 1000 – 1000000 baik menggunakan teknik *selection sort* dan *bubble sort*.

Kata kunci: *Borland, Bubble Sort, C++, Selection Sort, Sorting*

Abstract: *The role of algorithms in the software or programming is so important, so the need to understand the basic concept of the algorithm. Almost every moment of everyday life often encountered problems which must be solved by involving data sorting operations. So many logic programming that has been created, for cases that are common and also special. The process of sorting (sorting) is done using the technique of selection sort, and bubble sort. Sorting applications built using Borland C++. The algorithm is needed so that a problem can be solved more effectively with fewer resources involved and more efficiently with the duration of time it takes a little bit fast. Pseudocode C++ created successfully perform the sorting process of three groups, the first group 1 – 100, 100 – 1000 second, third 1000 – 1 million using either engineering bubble sort and selection sort.*

Keywords: *Borland, Bubble Sort, C++, Selection Sort, Sorting*

1. Pendahuluan

Algoritma pada dasarnya adalah susunan logis dan sistematis yang digunakan untuk memecahkan atau pun menyelesaikan suatu permasalahan tertentu. Jadi, setiap susunan logis yang diurutkan berdasarkan sistematika tertentu yang dipakai untuk menyelesaikan permasalahan dapat digolongkan sebagai sebuah algoritma.

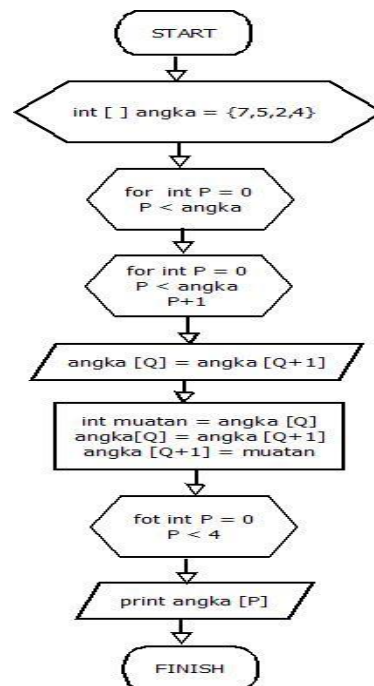
Pengurutan atau *sorting* merupakan suatu proses mengatur susunan data-data menurut syarat tertentu. Meskipun pengurutan ini sepertinya hanya sebuah masalah klasik dalam keinformatikaan, namun perannya tidak dapat dipisahkan terutama dalam pengolahan data. Suatu pengolahan data biasanya akan lebih efisien jika datanya telah terurut, seperti *Binary Search*. Mengingat pentingnya pengurutan dalam penggunaannya dalam hal keinformatikaan, maka perlu diketahui algoritma mana yang sebenarnya paling efisien untuk dipakai. Meskipun suatu algoritma pengurutan mempunyai kelebihan dan keterbatasan masing-masing, kompleksitas dan keefisiensannya tetap harus dipertimbangkan [Ezar and Rivan, 2017].

Dalam ilmu komputer, algoritma pengurutan atau *sorting* adalah algoritma yang meletakkan elemen-elemen suatu kumpulan data dalam urutan tertentu, proses pengurutan data yang sebelumnya disusun secara acak sehingga menjadi tersusun secara teratur menurut suatu aturan tertentu.

2. Metode Penelitian

Selection sort merupakan salah satu teknik pengurutan yang membandingkan elemen yang sekarang dengan elemen yang berikutnya samapai dengan elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari eleme sekarang maka dicatat posisinya dan kemudian ditukar. Dan begitu seterusnya. Algoritma *selection sort* dapat dirangkum sebagai berikut: a) Temukan nilai yang paling minimum (atau sesuai keinginan) di dalam struktur data. Jika ascending, maka yang harus ditemukan adalah nilai yang paling minimum. Jika descending, maka temukan nilai yang paling maksimum. b) Tukar nilai tersebut dengan nilai pada posisipertama di bagian struktur data yang belum diurutkan. c) Ulangi langkah di atas untuk bagian struktur datayang tersisa [Kumalasari, 2017].

Alur proses pengurutan menggunakan teknik *selection sort* ditunjukkan pada Gambar 1.

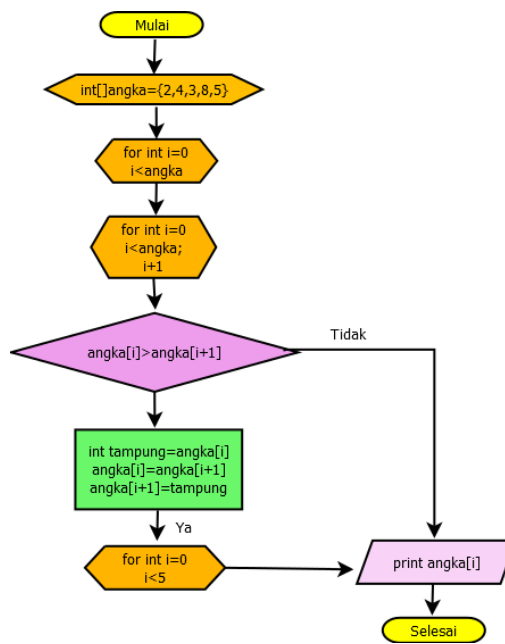


Sumber: Hasil Penelitian (2017)

Gambar 1. Flowchart Selection Sort

Bubble Sort adalah proses pengurutan sederhana yang bekerja dengan cara berulang kali membandingkan dua elemen data pada suatu saat dan menukar elemen data yang berurutannya salah. Ide dari *Bubble Sort* adalah gelembung air yang akan “mengapung” untuk table yang terurut menaik (ascending). Elemen bernilai kecil akan “diapungkan” (ke indeks terecil), artinya diangkat ke “atas” (indeks terkecil) melalui pertukaran. Karena algoritma ini melakukan pengurutan dengan cara membandingkan elemen – elemen data satu sama lain, maka *Bubble Sort* termasuk kedalam jenis algoritma comparison-based sorting [Rahayuningsih, 2016].

Alur proses pengurutan menggunakan teknik *bubble sort* ditunjukkan pada Gambar 2.



Sumber: Hasil Penelitian (2017)

Gambar 2. Flowchart Bubble Sort

3. Hasil dan Pembahasan

Pseudocode yang digunakan untuk melakukan *sorting* atau pengurutan dengan teknik *selection sort*. Pada sistem ini menggunakan bahasa C++.

```

#include <iostream.h>
#include <conio.h>
#include <stdio.h>
#include <iomanip.h>
#include <time.h>
#include <stdlib.h>

void garis()
{
cout<<"-----" <<endl;
}
main()
{
int x; //tipe data
int i;
int temp;
  
```

```

int nilai2;
int nilai['x'];
int minindex;
int j;
char range;
char jwb;
int start_s;

cout<<"\t NAMA KELOMPOK :\n";
cout<<"\t 1. Sahara Ramadhan\n";
cout<<"\t 2. Fauzi Eko Saputro\n";
cout<<"\t 3. Tia Monisya Afrianti\n" <<endl;
garis();
atas:
cout<<endl;
cout<<"\t>>> PROGRAM SELECTION SORT <<<< \n" <<endl;

cout<<" Banyak data = ";cin>>x;
cout<<endl;
cout<<" Daftar Range Data :\n\n";
cout<<" 1. 10-100\n";
cout<<" 2. 100-1.000\n";
cout<<" 3. 1.000-1.000.000\n"<<endl;
cout<<" Silakan pilih Range Datanya = ";cin>>range;
cout<<endl;

/*for(i=1; i<21; i++)
{
cout<<" S [data ke-"<<i<<" = ";cin>>x[i];
}*/
start_s=clock();
if(range=='1'){
cout<<"\n";
cout<<" ---- Data Range 10 - 100 ----\n"<<endl;
cout<<"\n Data sebelum di sort (Acak) :\n";
for(i=1; i<=x;i++)
{
nilai2=10+(rand()%91);
nilai[i]=nilai2;
cout<<setw(4)<<nilai[i];
}}
else if(range=='2'){
cout<<"\n";
cout<<" ---- Data Range 100 - 1000 ----\n"<<endl;
cout<<"\n Data sebelum di sort (Acak) :\n";
for(i=1; i<=x;i++)
{
nilai2=100+(rand()%915);
nilai[i]=nilai2;
cout<<setw(4)<<nilai[i];
}}
else if(range=='3'){
cout<<"\n";
cout<<" ---- Data Range 1000 - 1000000 ----\n"<<endl;
cout<<"\n Data sebelum di sort (Acak) :\n";
for(i=1; i<=x;i++)
{
nilai2=1000+(rand()%900000);

```

```
nilai[i]=nilai2;
cout<<setw(4)<<nilai[i];
}}
else
{
cout<<"\n ---- Data range tidak tersedia!!! ----"<<endl;
}

for(i=1; i<=x-1; i++) //perulangan iterasi
{
minindex=i;
for(j=i+1; j<=x; j++) //perulangan membandingkan data
{
if(nilai[minindex]>nilai[j])
{
minindex=j;
}
}
temp=nilai[i];
nilai[i]=nilai[minindex];
nilai[minindex]=temp;
}

cout<<"\n\n Data setelah di sort (Ascending) :\n";
for(i=1; i<=x; i++)
{
cout<<setw(4)<<nilai[i];
}
int stop_s=clock(); //waktu
cout<<endl<<"\n Waktu selesai : " <<(stop_s - start_s)/double(CLOCKS_PER_SEC) << "s"
<< endl;

garis();
cout<<"\t Input data lagi [Y/N]??? ";cin>>jwb;
jwb==getch();
garis();
if (jwb=='y'||jwb=='Y')
goto atas;
getch();
}
```

Pada hasil pengujian teknik *selection sort* digunakan untuk mengurutkan tiga kelompok, kelompok pertama untuk mengurutkan range 1-100, 100-1000, dan 1000-1000000.

```

C:\Users\TIA\Documents\Sorting\Selection2.exe
NAMA KELOMPOK :
1. Sahara Ramadhan
2. Fauzi Eko Saputro
3. Lia Monisya Afrianti

---
>>> PROGRAM SELECTION SORT <<<

Banyak data = 20
Daftar Range Data :
1. 10-100
2. 100-1.000
3. 1.000-1.000.000

Silakan pilih Range Datanya = 1

---- Data Range 10 - 100 ----

Data sebelum di sort (Acak) :
83 49 72 99 18 29 42 55 26 14 96 99 32 48 29 96 53 68 87 32

Data setelah di sort (Ascending) :
14 18 26 29 29 32 32 42 48 49 53 55 68 72 83 87 96 96 99 99

Waktu selesai : 0.015s

Input data lagi [Y/N]???
    
```

Sumber: Hasil Penelitian (2017)

Gambar 3. Hasil Pengurutan Teknik *Selection Sort* 1-100

```

Input data lagi [Y/N]??? y

---
>>> PROGRAM SELECTION SORT <<<

Banyak data = 50
Daftar Range Data :
1. 10-100
2. 100-1.000
3. 1.000-1.000.000

Silakan pilih Range Datanya = 2

---- Data Range 100 - 1000 ----

Data sebelum di sort (Acak) :
684 431 885 360 204 159 157 136 239 701 736 536 710 818 174 600 419 135 106 637
289 408 201 529 502 238 906 889 354 206 397 200 275 677 568 871 552 842 659 119
199 863 616 976 680 658 855 964 735 308

Data setelah di sort (Ascending) :
106 119 135 136 157 159 174 199 200 201 204 206 238 239 275 289 308 354 360 397
408 419 431 502 529 536 552 568 600 616 637 658 659 677 680 684 701 710 735 736
818 842 855 863 871 885 889 906 964 976

Waktu selesai : 0.031s

Input data lagi [Y/N]???
    
```

Sumber: Hasil Penelitian (2017)

Gambar 4. Hasil Pengurutan Teknik *Selection Sort* 100-1000

```

Input data lagi [Y/N]??? y

---
>>> PROGRAM SELECTION SORT <<<

Banyak data = 100
Daftar Range Data :
1. 10-100
2. 100-1.000
3. 1.000-1.000.000

Silakan pilih Range Datanya = 3

---- Data Range 1000 - 1000000 ----

Data sebelum di sort (Acak) :
1053144293285342293332937126361426874607458793691602584230142306672511516116184
18215652791600816859205619297475523981222753004029690240119137177352134326267931
28111873311993195545353211262018320067970274842161424431249991908619730650411891
29492280152114332246334843318025168177041067924528103652196617135674019323135802
63781373622327516457481202831113224452224924243224802267224625336911479919422133
44332311480318701582187761890317205215222419220113260701517211212830124461

Data setelah di sort (Ascending) :
10531121148020182156240132853422475551645279535357486504674074587460793679708111
87333776916892979312933310365106791189111993120201263613344135801373614268144291
47991517215821168081611616859171351729517704177351841818903190861913719323194221
95541973020113205612112621143213432152221614219662224922275223272244522480226722
39012419224243244312446124528246252499925115251682584226267263702687027484200152
83812949229690300403014230667311133187032086322463293733180332313348433691

Waktu selesai : 0.047s
    
```

Sumber: Hasil Penelitian (2017)

Gambar 5. Hasil Pengurutan Teknik *Selection Sort* 1000-1000000

Pseuocode yang digunakan untuk melakukan *sorting* atau pengurutan dengan teknik *bubble sort*. Pada sistem ini menggunakan bahasa C++.

```
#include <iostream.h>
#include <conio.h>
#include <time.h>
#include <stdlib.h>

void garis()
{
cout<<"-----" <<endl;
}
main(){
int nilai['n'];
int nilai2;
int temp;
int n;
char range;
char jwb;
int start_s;

cout<<"\t NAMA KELOMPOK :\n";
cout<<"\t 1. Sahara Ramadhan\n";
cout<<"\t 2. Fauzi Eko Saputro\n";
cout<<"\t 3. Tia Monisya Afrianti\n" <<endl;
garis();
atas:
cout<<endl;
cout<<"\t>>> PROGRAM BUBBLE SORT <<< \n" <<endl;

cout<<" Banyak data = ";cin>>n;
cout<<endl;
cout<<" Daftar Range Data :\n\n";
cout<<" 1. 10-100\n";
cout<<" 2. 100-1.000\n";
cout<<" 3. 1.000-1.000.000\n"<<endl;
cout<<" Silakan pilih Range Datanya = ";cin>>range;
cout<<endl;
/*for (int a=1; a<=n; a++){
cout<<" B [data ke-"<<a<<"]: ";
cin>>nilai[a];
}*/
start_s=clock();
if(range=='1')
{
cout<<"\n";
cout<<" ---- Data Range 10 - 100 ----\n"<<endl;
cout<<" Data Sebelum diurutkan (Acak) :"<<endl;
for(int a=1; a<=n; a++){
nilai2=10+(rand()%91);
nilai[a]=nilai2;
cout<<nilai[a]<<" ";
}}
else if(range=='2')
{
cout<<"\n";
```

```

cout<<" ---- Data Range 100 - 1000 ----\n"<<endl;
cout<<" Data Sebelum diurutkan (Acak) : "<<endl;
for(int a=1; a<=n; a++){
    nilai2=100+(rand()%915);
    nilai[a]=nilai2;
    cout<<nilai[a]<<" ";
}
else if(range=='3')
{
    cout<<"\n";
    cout<<" ---- Data Range 1000 - 1000000 ----\n"<<endl;
    cout<<" Data Sebelum diurutkan (Acak) : "<<endl;
    for(int a=1; a<=n; a++){
        nilai2=1000+(rand()%900000);
        nilai[a]=nilai2;
        cout<<nilai[a]<<" ";
    }
    else
    {
        cout<<"\n ---- Data range tidak tersedia!!! ----"<<endl;
    }

    for(int a=n-1; a>=1; a--){
        for(int b=1; b<=a; b++){
            if(nilai[b]>nilai[b+1]){
                temp=nilai[b+1];
                nilai[b+1]=nilai[b];
                nilai[b]=temp;
            }
        }
    }

    cout<<"\n\n Data setelah di sort (Ascending) : "<<endl;
    for (int a=1; a<=n; a++){
        cout<<nilai[a]<<" ";}

    cout<<"\n";
    int stop_s=clock();
    //cout << "\n\nJumlah dari bilangan genap = " << jumlah << endl;
    cout<<endl<<"\n Waktu selesai : " <<(stop_s - start_s) /double(CLOCKS_PER_SEC) << "s
\n" << endl;

    garis();
    cout<<"\t Input data lagi [Y/N]??? ";cin>>jwb;
    jwb==getch();
    garis();
    if (jwb=='y'||jwb=='Y')
        goto atas;
    getch();
}

```

Pada hasil pengujian teknik *bubble sort* digunakan untuk mengurutkan tiga kelompok, kelompok pertama untuk mengurutkan range 1-100, kelompok kedua 100-1000, dan kelompok ketiga 1000-1000000.


```
C:\Users\TIA\Documents\Sorting\Bubble.exe
NAMA KELOMPOK :
1. Sahara Ramadhan
2. Fauzi Eko Saputro
3. Tia Monisya Afrianti

>>> PROGRAM BUBBLE SORT <<<

Banyak data = 20
Daftar Range Data :
1. 10-100
2. 100-1.000
3. 1.000-1.000.000
Silakan pilih Range Datanya = 1

---- Data Range 10 - 100 ----
Data Sebelum diurutkan (Acak) :
83 49 72 99 18 29 42 55 26 14 96 99 32 48 29 96 53 68 87 32
Data setelah di sort (Ascending) :
14 18 26 29 29 32 32 42 48 49 53 55 68 72 83 87 96 96 99 99
Waktu selesai : 0.031s

Input data lagi [Y/N]???
```

Sumber: Hasil Penelitian (2017)

Gambar 6. Hasil Pengurutan Teknik *Bubble Sort* 1 – 100

```
>>> PROGRAM BUBBLE SORT <<<

Banyak data = 50
Daftar Range Data :
1. 10-100
2. 100-1.000
3. 1.000-1.000.000
Silakan pilih Range Datanya = 2

---- Data Range 100 - 1000 ----
Data Sebelum diurutkan (Acak) :
684 431 885 360 204 157 157 136 239 701 736 536 710 818 174 600 419 135 106 637
289 400 201 529 502 230 906 089 354 206 397 200 275 677 560 871 552 842 659 119
199 863 616 976 680 658 855 964 735 308
Data setelah di sort (Ascending) :
106 119 135 136 157 159 174 199 200 201 204 206 238 239 275 289 308 354 360 397
408 419 431 502 529 536 552 568 600 616 637 658 659 677 680 684 701 710 735 736
818 842 855 863 871 885 889 906 964 976
Waktu selesai : 0.031s
```

Sumber: Hasil Penelitian (2017)

Gambar 7. Hasil Pengurutan Teknik *Bubble Sort* 100 – 1000

```
C:\Users\TIA\Documents\Sorting\Bubble.exe
NAMA KELOMPOK :
1. Sahara Ramadhan
2. Fauzi Eko Saputro
3. Tia Monisya Afrianti

>>> PROGRAM BUBBLE SORT <<<

Banyak data = 100
Daftar Range Data :
1. 10-100
2. 100-1.000
3. 1.000-1.000.000
Silakan pilih Range Datanya = 3

---- Data Range 1000 - 1000000 ----
Data Sebelum diurutkan (Acak) :
1346 1130 1592 2999 15656 8117 18595 7415 23940 32126 10004 15558 4571 23079 19
492 2360 6412 27721 23463 26047 28119 32441 8190 14985 32214 28509 31252 27571 1
5779 20816 22631 20651 18925 24593 4924 14310 4929 22995 16561 17092 19480 12280
29466 9664 6892 14863 23766 6364 18639 22151 21427 1100 26795 9812 16108 13666
13347 20842 20774 18163 6589 27303 18666 11941 14398 8870 14565 2779 17190 32232
1053 14429 3285 3422 9333 32937 12636 14268 7460 7458 7936 9169 25842 30142 306
67 25115 16116 10410 2156 5279 16008 16859 20561 9297 4755 23981 22275 30040 296
90 2401
Data setelah di sort (Ascending) :
8953 1100 1130 1346 2090 2156 2360 2401 2779 3285 3422 4571 4734 4755 4979 5279
5384 6412 6589 6892 7415 7458 7460 7936 8117 8190 8870 9169 9297 9333 9664 9812
10004 10169 10666 11941 11982 12288 12636 12656 13347 13666 14268 14310 14390 14
429 14565 14863 14905 15558 15779 16008 16108 16116 16561 16859 17092 17190 1841
8 18595 18639 18995 19489 19492 20042 20561 20774 20816 21427 22151 22275
22681 22995 23463 23766 23879 23940 23981 24593 25115 25842 26047 26795 27303 27
571 27721 28119 28509 29466 29690 30040 30142 30667 31252 32126 32214 32441 3293
7 33233
Waktu selesai : 0.094s

Input data lagi [Y/N]???
```

Sumber: Hasil Penelitian (2017)

Gambar 8. Hasil Pengurutan Teknik *Bubble Sort* 1000 – 1000000

Setelah program pengurutan berhasil dijalankan. Proses selanjutnya dilakukan analisa waktu terhadap waktu yang dibutuhkan untuk teknik *selection sort* dan *bubble sort* berhasil melakukan proses pengurutan terhadap tiga kelompok bilangan. Hasil analisa waktu ditampilkan dalam Tabel 1.

Tabel 1 Hasil Analisa Proses Pengurutan

No.	Nama Proses	Jumlah Data yang di Sorting	Lama Proses (Per-Detik)
1	Selection Sort	20	0,015 s
		50	0,031 s
		100	0,047 s
2	Bubble Sort	20	0,031 s
		50	0,031 s
		100	0,094 s

4. Kesimpulan

Penelitian ini berhasil melakukan proses pengurutan (*sorting*) terhadap tiga kelompok bilangan, kelompok pertama rentang 1 – 100, kelompok kedua rentang 100 – 1000, dan kelompok ketiga rentang 1000 – 1000000 dengan menggunakan bahasa pemrograman C++. Berdasarkan hasil analisa waktu yang telah dilakukan menunjukkan bahwa teknik *selection sort* membutuhkan waktu yang lebih singkat dibandingkan dengan teknik *bubble sort*. *Selection sort* membutuhkan rata-rata waktu sekitar 0.031 detik, sedangkan *bubble sort* membutuhkan rata-rata waktu sekitar 0.078 detik.

Referensi

- Ezar M, Rivian A. 2017. Perbandingan Kecepatan Gabungan Algoritma Utama Quick Sort dan Merge Sort dengan Algoritma Tambahan Insertion Sort , Bubble Sort dan Selection Sort. J. Tek. Inform. dan Sist. Inf. 3: 319–331.
- Kumalasari D. 2017. Analisa Perbandingan Kompleksitas Algoritma Bubble Sort , Cocktail Sort Dan Comb Sort Dengan Bahasa Pemrograman C ++. J. Speed 9: 1–7.
- Rahayuningsih PA. 2016. Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting). J. Evolusi 4: 64–75.