**Tugas UAS**

**Laporan Penjelasan Aplikasi Sorting**

**Program Python**

**---------------------------------------------------------------------------------------------------------------------------------**

**Kelompok 4 :**

**- Rayhan Aqsyal Athallah (15200041) -**

**- Rizki Agung Sentosa (15200048) -**

**- Leonardus Alex Sandi Sugiarto Simbolon (15200064) -**

**- Fachry Fadillah (15200073) -**

**- Pramudya Widyastama (15200088) -**

**- Muhammad Kahfi Prayoga (15200105) -**

**---------------------------------------------------------------------------------------------------------------------------------**

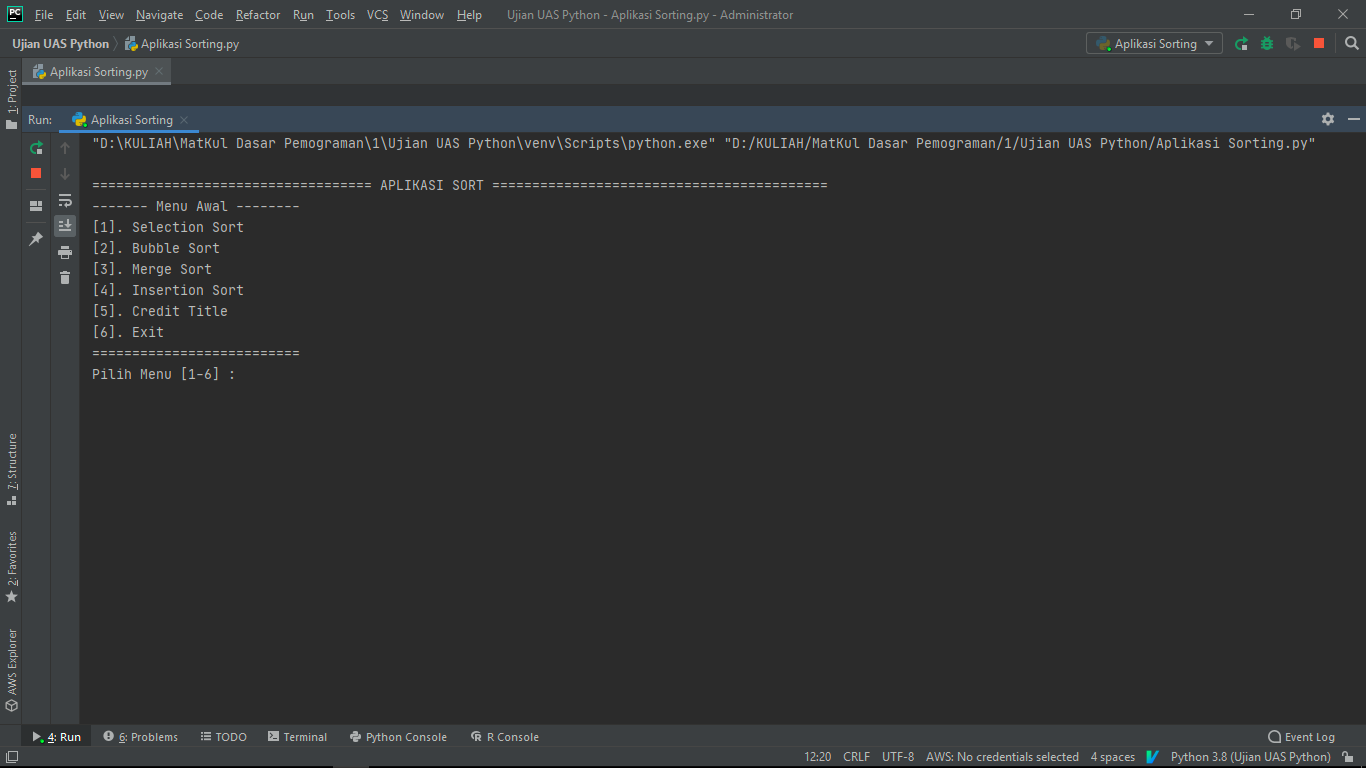
**- Penjelasan Aplikasi**

Jadi inti penjelasan dari semua sorting adalah pengurutan data berdasarkan keinginan baik itu dari yang terkecil maupun yang terbesar, sehingga data yang dihasilkan akan lebih terstruktur, teratur dan sesuai dengan kebutuhan kita.

Terdapat beberapa sorting yang cukup popular untuk mengurutkan data, seperti ***Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort*** dan lain sebagainya. Dibawah ini kami hanya menjelaskan 4 metode sorting mudah mudahan dapat dipahami

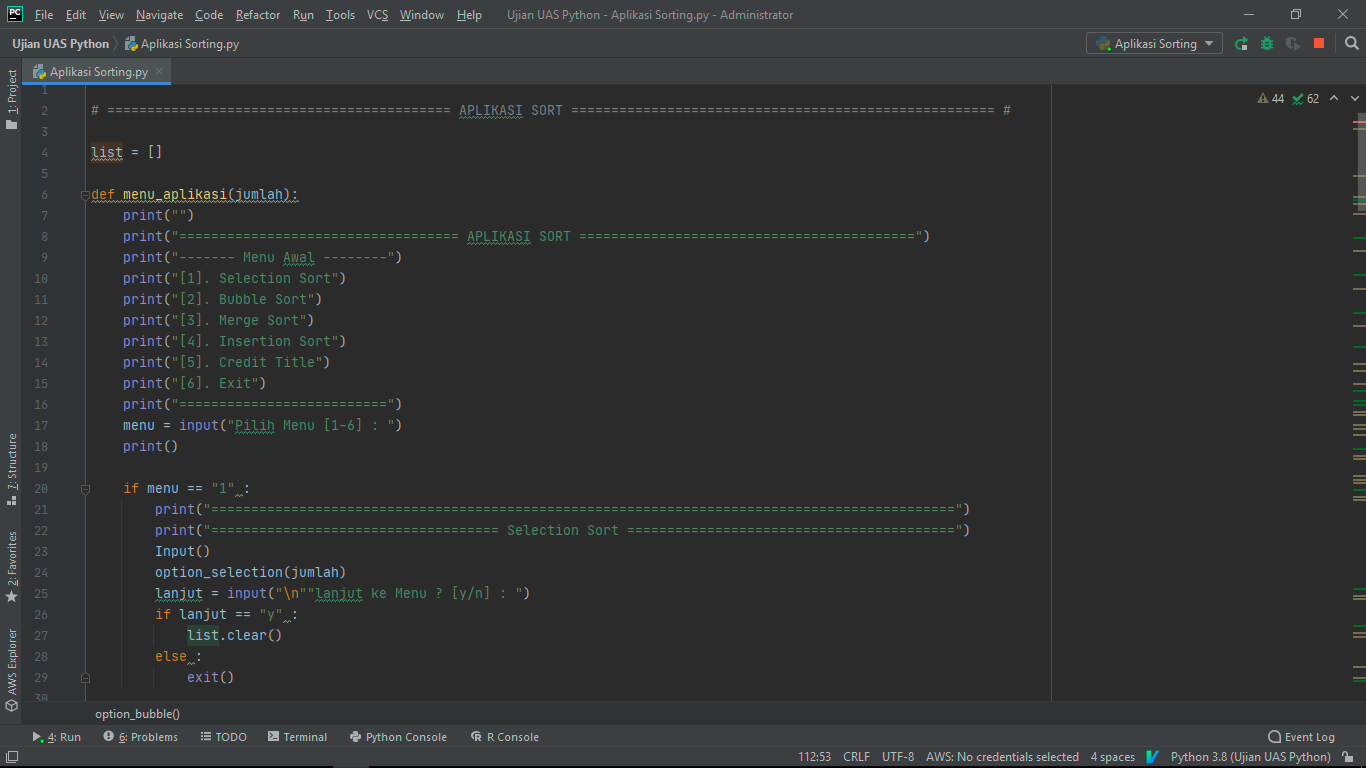
**----------------------------------------------------------------------------------------------------------------------------------**

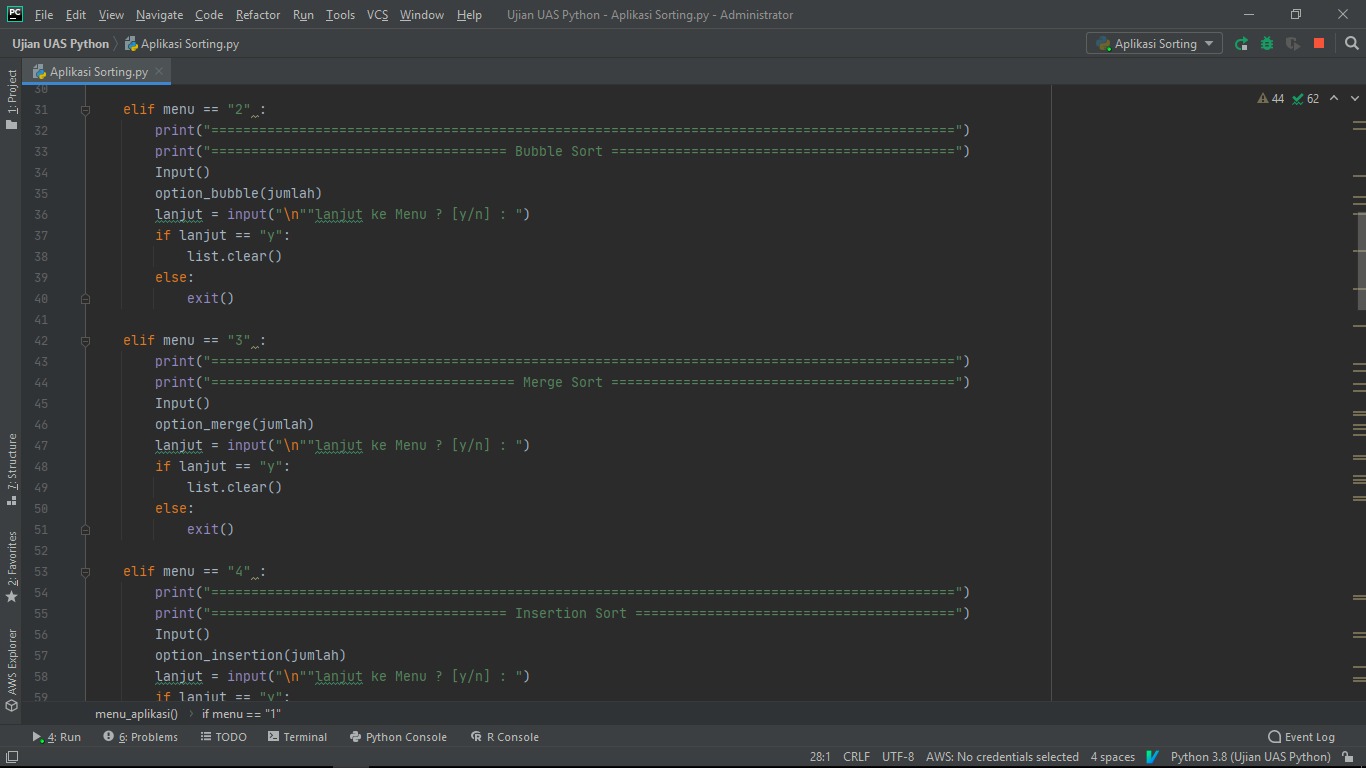
**\*Tampilan Aplikasi**

****

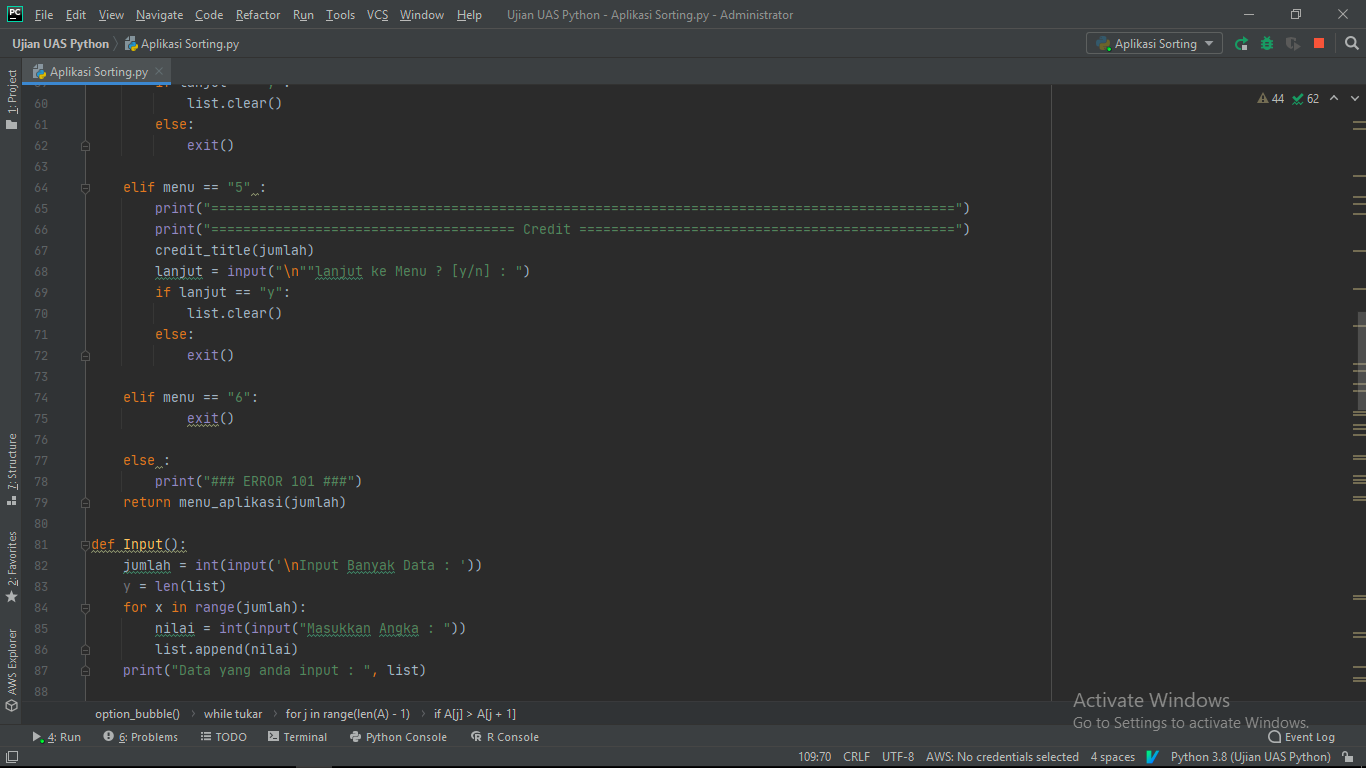
**---------------------------------------------------------------------------------------------------------------------------------**

**\*Program Aplikasi Sorting**

****

****

**\*Lanjut dibawah**



**---------------------------------------------------------------------------------------------------------------------------------**

**Penjelasan Program Aplikasi**

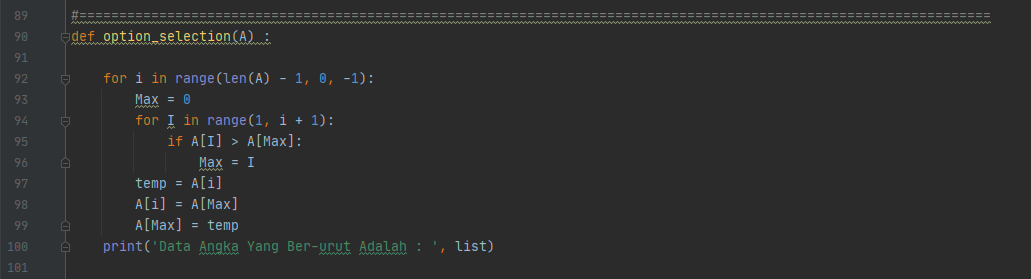
Tampilan/Halaman utama dari aplikasi sorting yang menampilkan menu-menu fungsi dari berbagai jenis sorting. Pada menu ini terdiri dari percabangan if sebagai pemilih fungsi sorting yang akan kita coba gunakan. Penggunaan input di fungsi def() menu aplikasi awal ini yang akan menentukan sorting apa yang akan kita gunakan. Dalam percabangan if kita akan memanggil fungsi input lain yang digunakan sebagai isi data di setiap pilihan menu sorting. Dan juga fungsi def() yang berisi program sorting dan akan langsung di jalankan. Dan kita menambahkan fitur “Lanjutkan Aplikasi” sebagai fungsi jika kita selesai menggunakan metode sorting 1 maka kita bisa melanjutkan ke menu aplikasi Kembali tanpa harus membuka atau me-run program, kecuali pengguna tidak ingin melanjutkan dengan memilih “n” dengan begitu aplikasi akan tertutup/berhenti. Sehingga menjadikan aplikasi ini lebih fleksibel.

**---------------------------------------------------------------------------------------------------------------------------------**

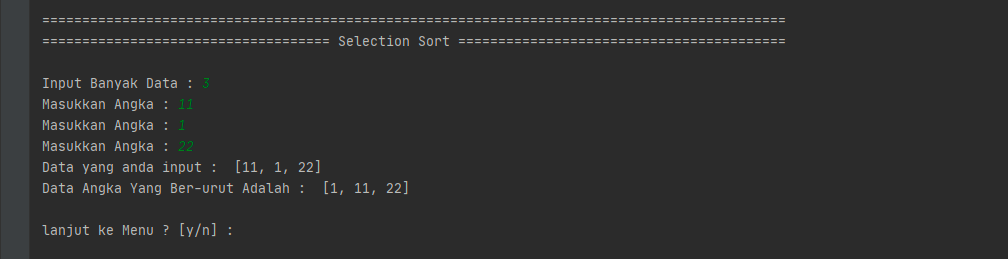
**1. Selection Sort**

Teknik pengurutan dengan cara pemilihan elemen atau proses kerja dengan memilih elemen data terkecil untuk kemudian dibandingkan & ditukarkan dengan elemen pada data awal, dst s/d seluruh elemen sehingga menghasilkan pola data yang telah disort.

**\*Program Selection Sort**

****

**\*Run Program**

****

**Selection Sort**

1. Pengecekan dimulai dari data ke-1 sampai data ke-n.

2. Tentukan bilangan dengan index terkecil dari data bilangan tersebut.

3. Tukar bilangan index terkecil tersebut dengan bilangan pertama (i= 1).

4. Ulangi langkah 2 dan 3 dengan bilangan index (i= i+1) hingga data terurut semuanya

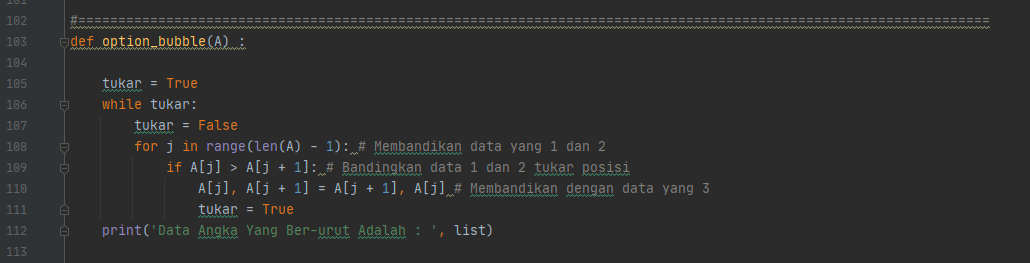
**---------------------------------------------------------------------------------------------------------------------------------**

**2. Bubble Sort**

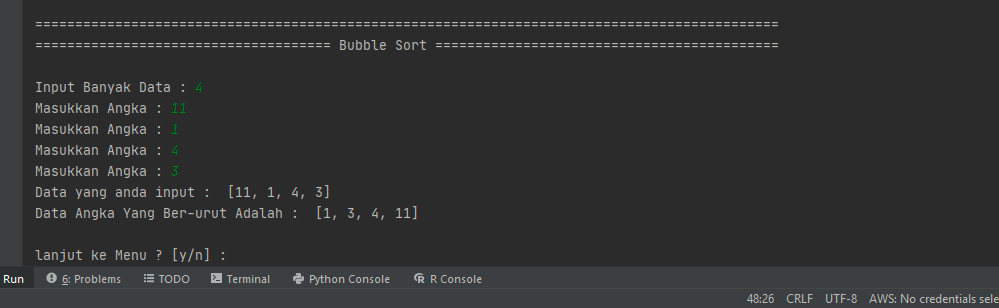
Metode pengurutan dengan membandingkan data nilai elemen yang sekarang dengan data nilai elemen-elemen berikutnya.

Pembandingan elemen dapat dimulai dari awal atau mulai dari paling akhir. Apabila elemen yang sekarang lebih besar (untuk urut menaik) atau lebih kecil (untuk urut menaik) dari elemen berikutnya, maka posisinya ditukar, tapi jika tidak maka posisinya tetap

**\*Program Selection Sort**



**\*Run Program**



**Bubble Sort**

1. Bandingkan nilai pada data ke-1 dengan data ke-2.

2. Jika nilai data ke-1 lebih besar dari data ke-2 maka tukar posisinya.

3. Kemudian data yang lebih besar tersebut dibandingkan lagi dengan data ke-3.

4. Jika data ke-3 lebih kecil dari data ke-2 maka tukar posisinya, dan begitu seterusnya sampai

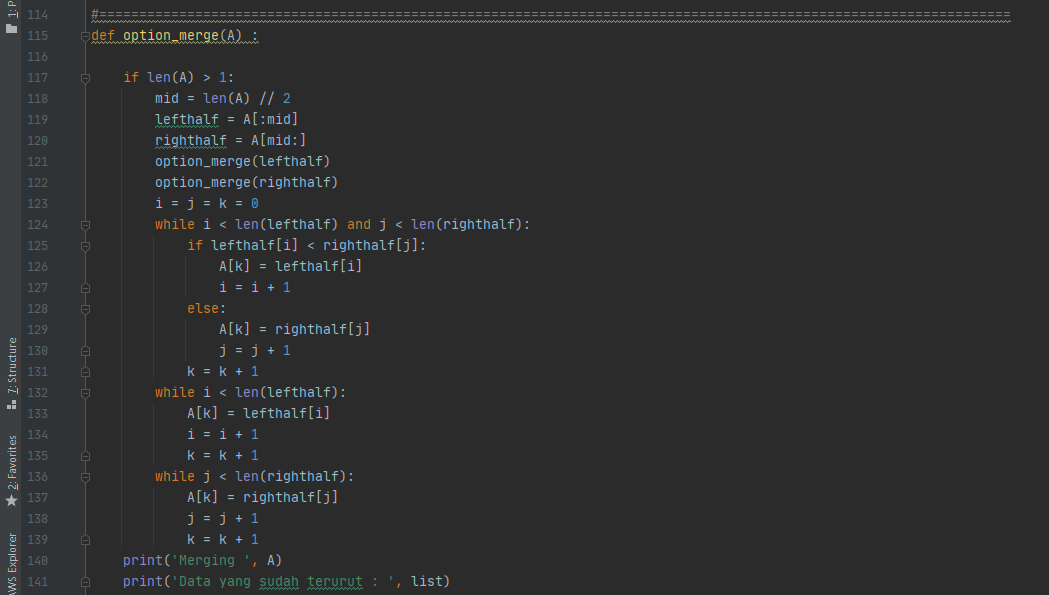
semua data yang ada jadi terurut.

**---------------------------------------------------------------------------------------------------------------------------------**

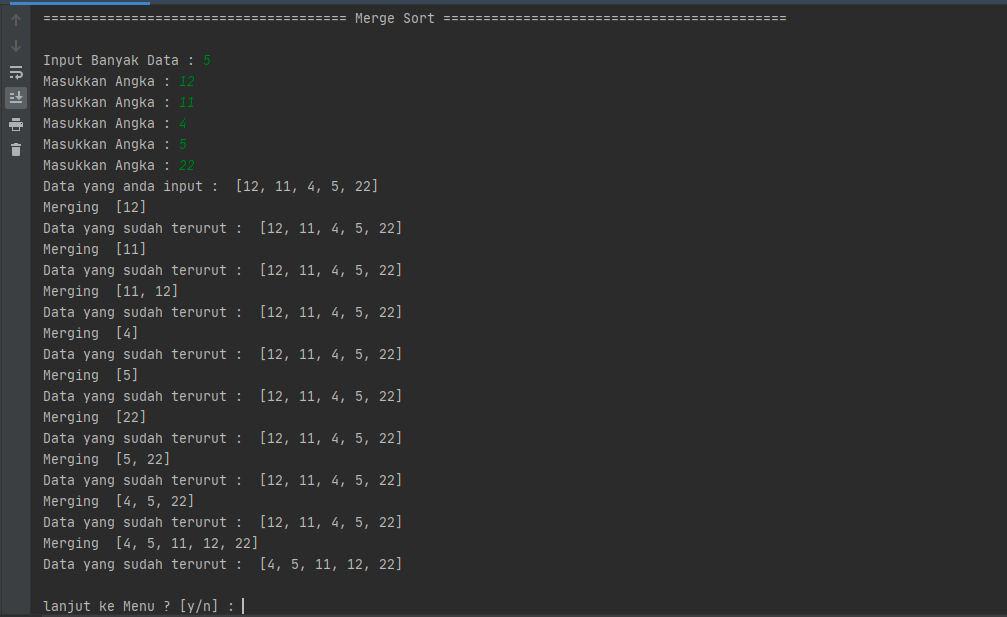
**3. Merge Sort**

Menggabungkan dua array yang sudah terurut

**\*Program Selection Sort**



**\*Run Program**



**Merge Sort**

1.Pertama data dipecah menjadi 2 bagian dimana bagian pertama merupakan setengah (jika data

genap) atau setengah minus satu (jika data ganjil) dari seluruh data, kemudian dilakukan

pemecahan kembali untuk masing-masing blok sampai hanya ada satu data dalam satu blok.

2. Setelah digabungkan kembali dengan membandingkan pada blok yang sama apakah data

pertama lebih besar dari pada data ke-tengah+1 , jika iya maka data ke-tengah+1 dipindah

menjadi data pertama. Kemudian data pertama tadi sampai data ke-tengah dipindah menjadi

data ke-dua sampai data ke-tengah+1.

3. Begitu seterusnya hingga membentuk data yg terurut dalam satu blok yang utuh seperti

awalnya. Sehingga merge sort membutuhkan fungsi rekursif dalam penyelesainnya.

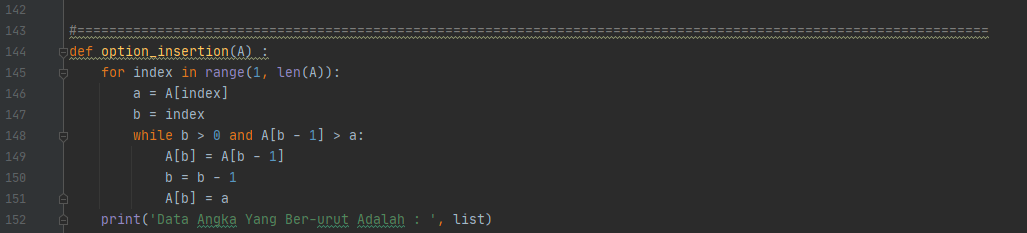
**---------------------------------------------------------------------------------------------------------------------------------**

**4. Instertion Sort**

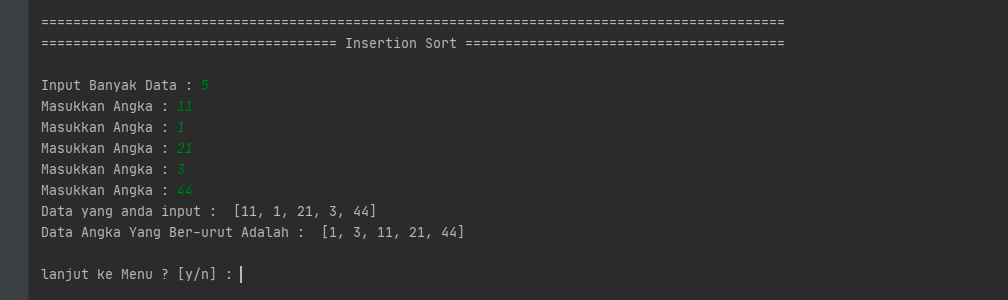
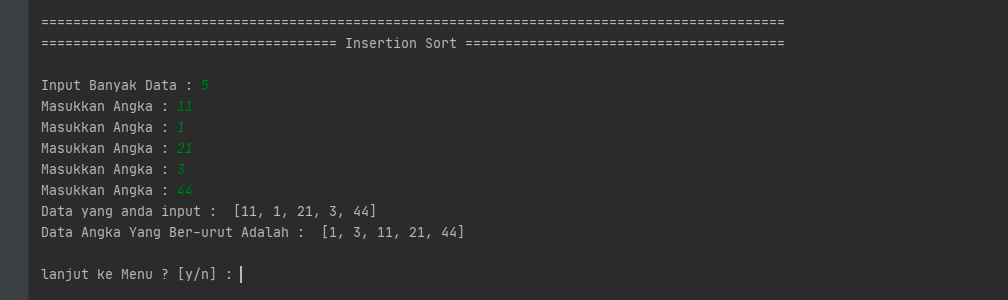
Pengurutan data yang membandingkan data dengan dua elemen data pertama, kemudian membandingkan elemen-elemen data yang sudah diurutkan, kemudian perbandingan atara data tersebut akan terus diulang hingga tidak ada elemen data yang tersisa (Rahayuningsih, 2016).

Mirip dengan cara mengurutkan kartu, perlembar yang diambil & disisipkan (insert) ke tempat yang seharusnya.

**\*Program Selection Sort**



**\*Run Program**



**Insertion Sort**

1. Bandingkan data ke-2 dengan data ke-1, jika data ke-2 lebih kecil maka tukar posisinya, jika

tidak biarkan aja.

2. Data ke-3 dibandingkan dengan data ke-1 dan ke-2, jika data ke-3 lebih kecil kemudian tukar

lagi posisinya.

3. Data ke-4 dibandingkan dengan data ke-3, ke-2, dan ke-1, jika data ke-4 lebih kecil dari

ketiganya maka letakkan data ke-4 ke posisi paling depan. Begitu seterusnya sampai tidak ada

lagi data yang bisa dipindahkan.

Link Video Youtube :

https://youtu.be/Oq0XuInQwRE