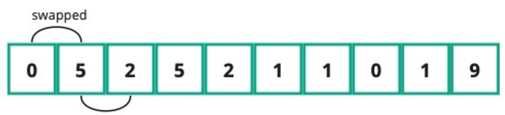
Bubble Sort Algorithm
input testcase : 5025211019

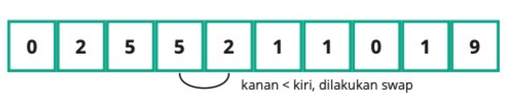
```c
//Bubble Sort in C
#include <stdio.h>//Adrian Karuna Soetikno
#include <stdbool.h>//NRP : 5025211019

void swap(int *X, int *Y)
{
    int temp = *X;
    *X = *Y;
    *Y = temp;
}

//function bubble sort
void bubbleSort(int arr[], int n)
{
  int i, j;
  bool flag;
  for (i = 0; i < n-1; i++)
  {
    flag = false;
    for (j = 0; j < n-i-1; j++)
    {
      if (arr[j] > arr[j+1])
      {
        swap(&arr[j], &arr[j+1]);
        flag = true;
      }
    }
    // IF no two elements were swapped by inner Loop, then break
    if (flag == false)
      break;
  }
}

int main(){
    int arr[] = {5, 0, 2, 5, 2, 1, 1, 0, 1, 9}; //input nrp
    int size = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, size);
    printf("Sorted array: ");

        for (int i=0; i < size; i++){
        printf("%d ", arr[i]);
        }

    return 0;
}
```

apabila array kanan lebih kecil dari pada kiri, maka akan di swap

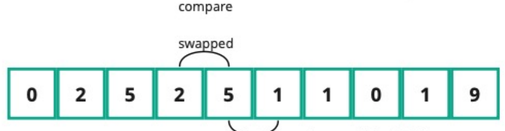


kanan < kiri, maka dilakukan swap

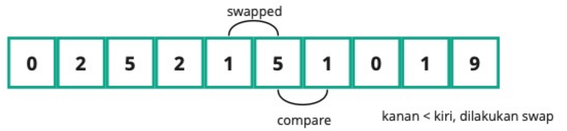
karena kanan == kiri, tidak dilakukan swap



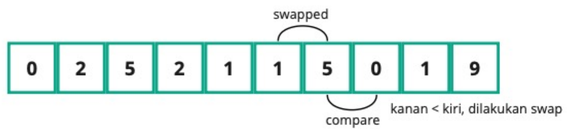
kanan < kiri, dilakukan swap



kanan < kiri, dilakukan swap



kanan < kiri, dilakukan swap

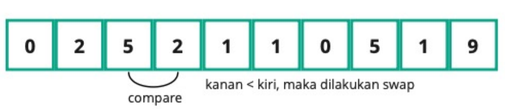


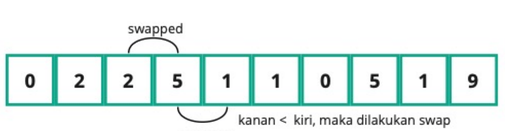
kanan < kiri, dilakukan swap

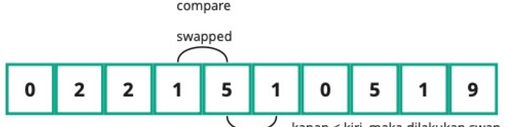
kanan < kiri, dilakukan swap

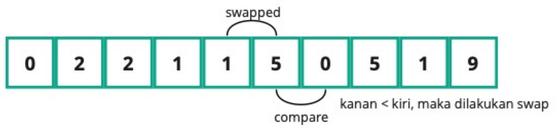
kanan > kiri, tidak dilakukan swap

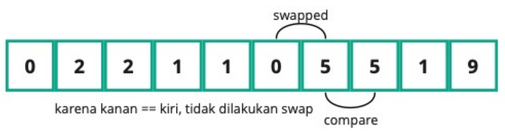
proses comparing dan swap yang pertama telah selesai, berikutnya akan dilakukan compare dan swap lagi, tetapi iterasi akan terhenti pada array[length-1]

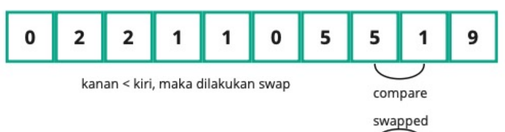


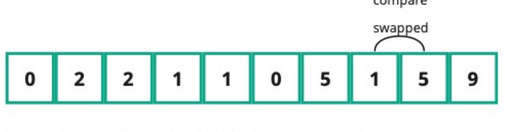








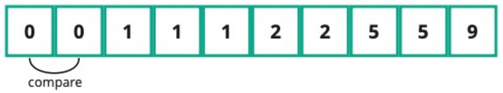




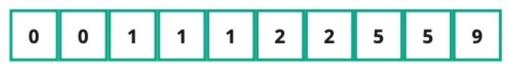
komparasi yang kedua telah selesai, jumlah iterasi yang dilakukan lebih kecil 1 daripada yang sebelumnya, karena dilakukan hanya hingga array[length -1]

berikutnya akan dilakukan proses yang sama dengan jumlah iterasi akan terus dikurangi 1 dari belakang array terus menerus, hingga isi dari array akan ter sort.

iterasi sorting akan terus berulang hingga hanya akan mencompare array[0] dan [1]



setelah itu, maka proses sorting dapat dinyatakan selesai, dan hasil sorting telah didapat pada array ini :

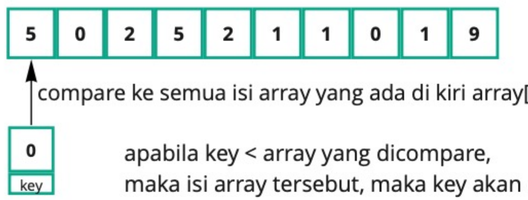


Insertion Sort Algorithm
input testcase : 5025211019

```c
// Insertion sort in C

#include <stdio.h>

void insertionSort(int array[], int size) {
  for (int step = 1; step < size; step++) {
    int key = array[step];
    int j = step - 1;

    // Compare key with each element on the left of it until an element smaller than
    // it is found.
    // For descending order, change key<array[j] to key>array[j].
    while (key < array[j] && j >= 0) {
      array[j + 1] = array[j];
      --j;
    }
    array[j + 1] = key;
  }
}

// Driver code
int main() {

  int arr[] = {5, 0, 2, 5, 2, 1, 1, 0, 1, 9};
  int size = sizeof(arr) / sizeof(arr[0]);
  insertionSort(arr, size);
  printf("Sorted array : ");
    for (int i = 0; i < size; i++) {
    printf("%d ", arr[i]);
  }
}
```

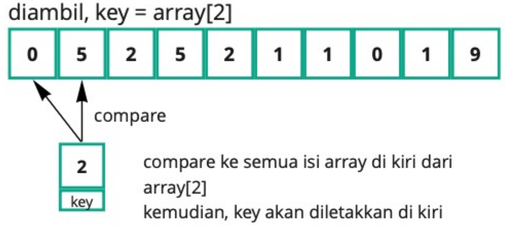
key pertama dianggap sudah ter sort

kemudian, diambil key, pada array[1]



apabila key < array yang dicompare,
maka isi array tersebut, maka key akan digeser ke kiri dari array tersebut

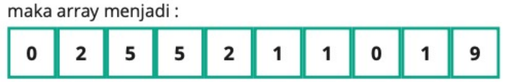
diambil, key = array[2]



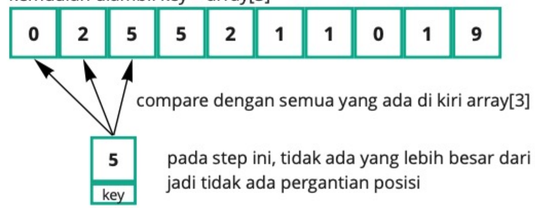
compare ke semua isi array di kiri dari array[2]
kemudian, key akan diletakkan di kiri isi array yang lebih besar dari key tersebut

karena key < 5 dan key > 0, maka diletakkan di tengah" 5 dan 0

maka array menjadi :

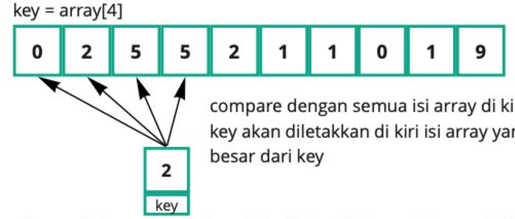


kemudian diambil key = array[3]



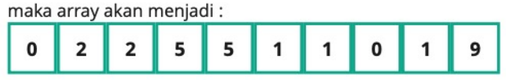
pada step ini, tidak ada yang lebih besar dari key
jadi tidak ada pergantian posisi

key = array[4]



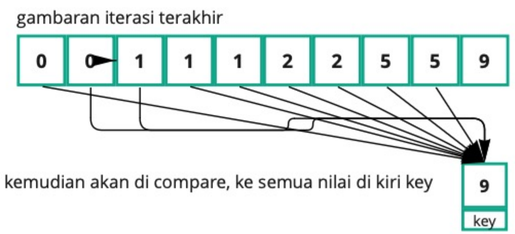
pada step ini, maka key akan dipindah ke kiri array[2] atau di kiri 5

maka array akan menjadi :



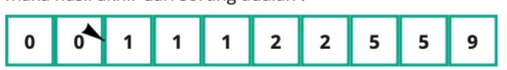
kemudian proses ini akan terulang terus menerus, hingga key mencapai akhir dari array,
kemudian loop akan berakhir, dan array akan ter sort

gambaran iterasi terakhir



karena tidak ada yang lebih besar dari nilai key, maka tidak ada terjadi pergantian/swap

maka hasil akhir dari sorting adalah :



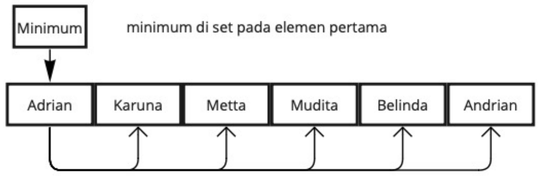
Selection Sort Algorithm for Strings

input testcase : Adrian, Karuna, Metta, Mudita, Belinda, Andrian

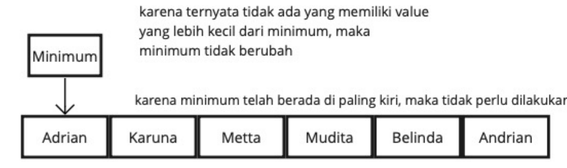
untuk melakukan sorting strings, maka perlu dilakukan beberapa perubahan pada program.

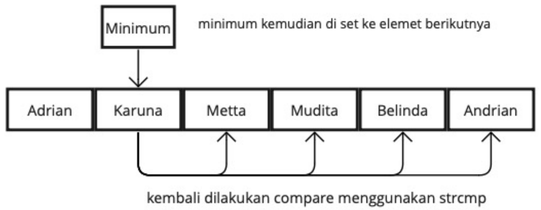
array input

minimum di set pada elemen pertama



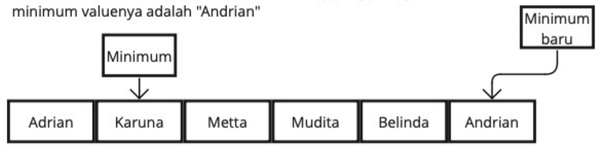
kemudian, dilakukan compare menggunakan strcmp apabila terdapat string yang lebih kecil dari string pada "minimum", maka minimum diubah ke string tersebut



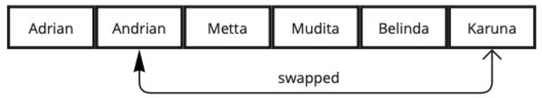


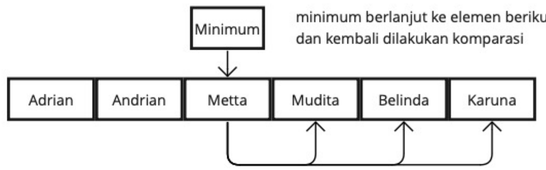
kembali dilakukan compare menggunakan strcmp dengan elemen dikanan minimum, tujuannya untuk mencari string yang memimiliki value paling kecil

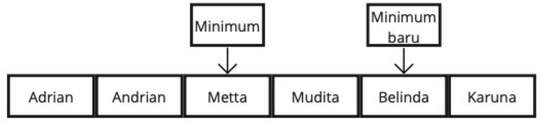
setelah dilakukan komparasi, ditemukan string yang paling minimum valuenya adalah "Andrian"



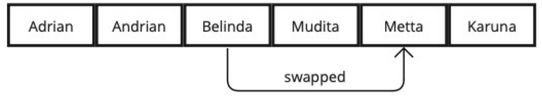
maka kedua string tersebut di swap posisinya

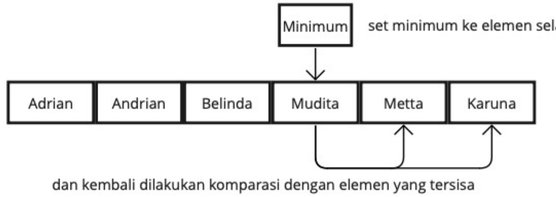






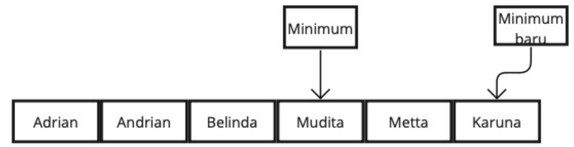
didapatkan minimum yang baru, yaitu "Belinda"

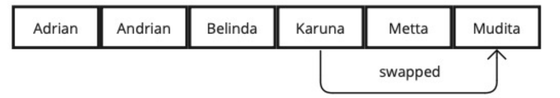


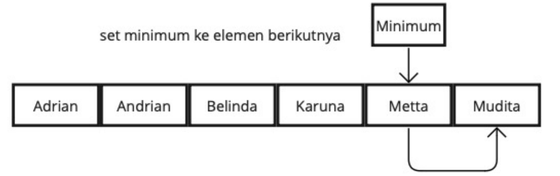


dan kembali dilakukan komparasi dengan elemen yang tersisa

didapatkan minimum yang baru "Karuna"







karena tidak ditemukan yang memiliki value lebih kecil, maka tidak dilakukan swap

maka array of strings ini telah ter sort, dengan urutan : Adrian, Andrian, Belinda, Karuna Metta, Mudita

Insertion Sort Algorithm for Strings
input testcase : Adrian, Karuna,
Metta, Mudita, Belinda, Andrian

```c
//insertion sort for strings in c
//Adrian Karuna Soetikno
//5025211019

#include <stdio.h>
#include <string.h>

int main(int argc, char const *argv[])
{
    char strings[6][100]= {
         "Adrian",
         "Karuna",
         "Metta", //contoh input, menggunakan 6 nama
         "Mudita",
         "Belinda",
         "Andrian"
     };

    int size = 6;//jumlah input tadi,  6 nama

    //insertionsort(strings, size);
    for (int step = 1; step < size; step++) {
    char temp[100];
    strcpy(temp, strings[step]);
    int j = step - 1;

    while (strcmp(temp, strings[j]) < 0 && j >= 0) {
        strcpy(strings[j+1], strings[j]);
      //strings[j + 1] = strings[j];
      --j;
    }
    strcpy(strings[j+1],temp);
  }
    printf("Sorted array : \n");
    for (int i = 0; i < size; i++) {
    printf("%s\n", strings[i]);
  }

    return 0;
}
```

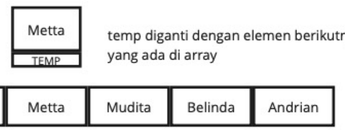
array input

step pertama adalah, set elemen ke 1 pada array ke variabel temp

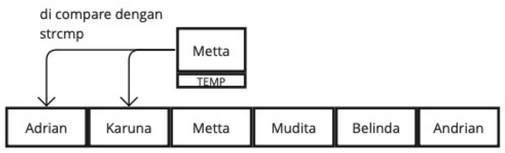
string "karuna" di strcpy ke dalam variable temp

kemudian temp akan di compare menggunakan strcmp dengan elemen sebelumnya (step - 1)

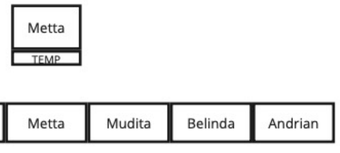
karena "Adrian" memiliki value lebih kecil dari "Karuna", maka tidak dilakukan swap atau pergantian posisi string pada array



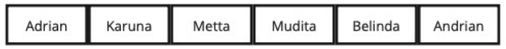
temp diganti dengan elemen berikutnya yang ada di array



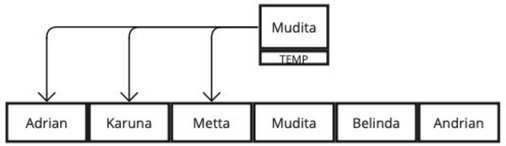
setelah di compare, "Metta" diletakkan tepat di kiri string yang memiliki value lebih besar dari metta. Atau di kanan string yang memiliki value lebih kecil dari "Metta"



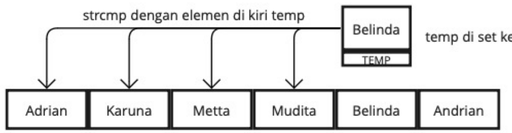
karena "Metta" memiliki value lebih besar dari elemen di kirinya, maka posisi metta tidak berubah



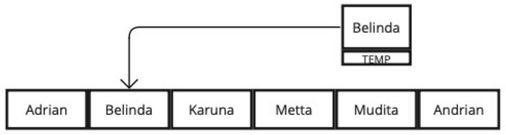
kemudian temp kembali di compare menggunakan strcmp dengan elemen elemen di kirinya

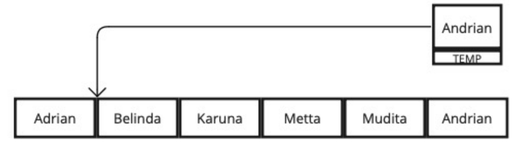


mudita memiliki value terbesar, maka tidak terjadi swap



karena value dari "Belinda" adalah lebih besar dari "Adrian" maka diletakkan di kiri "Adrian"





karena value "Andrian" lebih besar dari "Adrian" dan lebih kecil dari "Belinda", maka diletakkan ditengah"nya

karena iterasi sudah selesai semua, maka array sudah ter sort dengan urutan sebagai berikut :