

Sorting String Dinamis dalam Bahasa C



Oleh :

Jeffrey Hermanto Halimsetiawan

shadowz_029@yahoo.com.sg

tutorialpemrograman.wordpress.com

24 Desember 2009

Sorting String Dinamis dalam Bahasa C

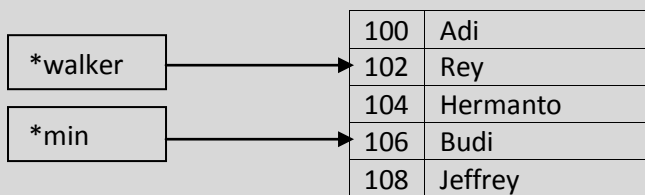
String dalam bahasa C direpresentasikan dalam bentuk array of character (`char string[30]`), nah bagaimana cara membentuk array of String? Tentu saja dengan membuat array of array of character, contohnya : `char arrString[30][100]` yang dapat dikatakan terdapat 30 string mulai dari indeks 0 – 29 yang setiap stringnya memiliki panjang tidak lebih dari 100 character. Kemudian muncul pertanyaan bagaimana cara mengurutkan sejumlah nama yang terdapat dalam array secara *ascending* ataupun *descending*. Caranya tentu mudah saja, sama seperti melakukan sorting dengan menggunakan salah satu metode sorting dalam array of integer namun dalam membandingkan antar string-nya dapat menggunakan salah satu fungsi dari library `string.h`, yaitu : `strcmp`. Bagaimana cara menukar / swap antar string? Jika dalam penukaran dua nilai integer biasanya digunakan :

```
int temp = a;
```

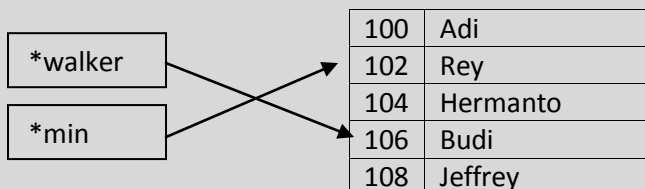
```
a = b;
```

```
b = temp;
```

Maka pada swap string hanya perlu menukar pointer antara dua buah string dalam array of string.



Diubah menjadi :



Dinamis yang dimaksud adalah program melakukan alokasi memori dinamis sehingga program hanya mengalokasikan memori seperti yang diinginkan oleh user. Untuk alokasi memori dinamis tersebut, digunakan fungsi `malloc` yang terdapat dalam library `stdlib.h`

Untuk lebih dapat memahami sorting string dinamis, di bawah ini disertakan source code program sederhana dengan menggunakan metode selection sort secara rekursif. Ingat **'Practice Makes Perfect'** . Semoga bermanfaat :D

Source Code

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

//Prototipe Fungsi
int getInput();
```

```

void createArray(char **aString,int n);
void printArray(char **aString,int n);
void sortingArray(char **aString,int n);
void swap(char **a,char **b);
void selectionSort(char **aString,int walker,int n);
void findMin(char **aString,int walker, int n,int *min);

int main(){
    char **aString;
    int n;
    n = getInput();
    //mengalokasikan memori array of string sebanyak n jumlah string
    aString = (char **) malloc(sizeof(char *) * n);
    createArray(aString,n);
    puts("\nInput String : ");
    printArray(aString,n);
    selectionSort(aString,0,n-1);
    //sortingArray(aString,n);
    puts("\nOutput String : ");
    printArray(aString,n);
    getch();
    return 0;
}
/**
 * Fungsi untuk meminta inputan jumlah string
 */
int getInput(){
    int n;
    while (true){
        printf("Jumlah String : ");
        if (scanf("%d",&n) == 1 && n > 0 ) return n;
    }
}
/**
 * Fungsi untuk mengalokasikan memori dari array of character sebanyak n
 */
void createArray(char **aString,int n){
    int i,j;

    for (i=0;i<n;i++){
        aString[i] = (char *) malloc (sizeof(char)* 50);
    }
    for (i=0;i<n;i++){
        fflush(stdin);

        printf("Input %3d : ",i+1);
        gets(aString[i]);
    }
}
/**
 * Fungsi untuk menampilkan isi array of String
 */
void printArray(char **aString,int n){
    for (int i=0;i<n;i++){
        printf("%s\n",aString[i]);
    }
}
/**
 * Fungsi untuk melakukan selection sort ascending secara iteratif
 */
void sortingArray(char **aString,int n){

```

```
int min;
for (int i=0;i<n-1;i++){
    min = i;
    for (int j=i+1;j<n;j++){
        if (strcmp(aString[min],aString[j]) > 0){
            min = j;
        }
    }
    if (min != i){
        swap(&aString[min],&aString[i]);
    }
}
}
/**
 * Fungsi untuk melakukan selection sort pada string secara rekursif
 */
void selectionSort(char **aString,int walker,int n){
    if (walker == n)
        return;
    else {
        int min = walker;
        findMin(aString,walker+1,n,&min);
        if (min != walker){
            swap(&aString[min],&aString[walker]);
        }
        selectionSort(aString,walker+1,n);
    }
}
/**
 * Fungsi untuk mencari nilai minimum pada setiap step Selection Sort
 * secara rekursif
 */
void findMin(char **aString,int walker, int n,int *min){
    if (walker > n)
        return;
    else {
        if (strcmp(aString[*min],aString[walker]) > 0){
            *min = walker;
        }
        findMin(aString,walker+1,n,&(*min));
    }
}
/**
 * Fungsi untuk melakukan swap array of character
 */
void swap(char **a,char **b){
    char *temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```