

Tugas 1: Implementasi Operasi Dasar Pada Array

Dwi Cahya Ramadani 5024201025

September 9, 2021

1 Program Operasi Dasar Pada Array

Program pada Listing 1 adalah implementasi operasi Dasar Pada Array dengan menggunakan Bahasa C

Listing 1: Operasi dasar Dalam Array

```
#include <stdio.h>
void Cetak(float L[100], int M)
{
    for(int i=0; i<=M; i++)
    { printf("%f", L[i]);
    }
}

void Insert(float L[100], int *M, int i, float X)
{ int N=100; //Panjang array
  int BACK;
  if(*M<N-1)
  { BACK=*M;
    //Menggeser Elemen Ke kanan
    while (BACK>i)
    { L[BACK]= L[BACK-1];
      BACK=BACK-1;
    }
    //Menyisipkan Nilai Pada posisi ke-i
    L[i]=X;
    //Menambah Jumlah elemen
    *M=*M+1;
  }
}

void Delete(float L[100], int *M, int i)
{ int BACK;
  //Jika nilai i<0 atau i>posisi elemen maka tida tidak valid.
  if (i<0) || (i>*M) return;
  BACK=i;
  for (BACK=i; BACK<*M; BACK++)
```

```

    { L[BACK]= L[BACK+1];
    }
    //Mengurangi posisi elemen terakhir
    *M=*M-1;
}

int Search(float L[100], int M, float X)
{ // Parameter Input
  //L[100]:Array, panjang 100
  //M : Posisi Elemen terakhir
  //X : Nilai yang di cari
  //Output : Posisi elemen.
  // pos>=0 bila ditemukan
  // pos=-1 bila tidak ditemukan
  int C=0,pos=-1;
  for (C=0;C<=M;C++)
  { if (L[C]==X)
    { //Jika ditemukan maka
      // Simpan nilai C dan pencarian diakhiri.
      pos=C;
      break;
    }
  }
  return pos;
}

void Update(float L[100], int i, float X)
{ // Parameter Input
  //L=Array panjang 10;
  //i:Posisi Elemen yang diperarui
  //X:Nilai yang baru
  L[i]=X;
}

void Append(float L[100],int *M, float X)
{int N=100;//Panjang array
  if(*M<N-1)
  { *M=*M+1;
    L[*M]=X;
  }
}

int main()
{
  //Index array pada C dimulai dari 0.
  float L[100];
  //M=-1 Array kosong
  int M;

```

```

M=-1;
//Menambah lima Item pada array.
Append(L,&M,4);
Append(L,&M,5);
Append(L,&M,6);
Append(L,&M,5);
Append(L,&M,1);
Cetak(L,M);
//Mencari Posisi item yang bernilai 6.
int pos;
pos = Search(L,M,6);
printf("Posisi: %d",pos);
//Hapus elemen yang telah disimpan di variabel pos.
Delete(L,&M,pos);
Cetak(L,M);
//Menyisipkan nilai 8 pada posisi index 4;
Insert(L,&M,4,8);
Cetak(L,M);
//Perbarui isi dari

return 0;
}

```

2 Tugas

Dengan menggunakan fungsi-fungsi operasi dasar pada array seperti yang terdapat pada Listing 1

1. Buatlah fungsi untuk menghapus seluruh isi nilai yang pada Array !.
2. Buatlah Fungsi untuk mencari posisi nilai tertinggi yang terdapat pada Array dan cetaklah hasilnya !.
3. Buatlah fungsi untuk menghapus nilai tertinggi tersebut !.

Tugas Dikumpulkan dengan ketentuan sbb :

- Tugas dikumpulkan dalam bentuk Laporan dan dikumpulkan ke google class room dengan kode kelas :
 - Kelas A : 5b4a7kl
 - Kelas B : qoi54js
- Nama yang muncul di google class room harus nama yang sesuai dengan absen pada My ITS Presensi.
- Tugas disimpan dengan Nama File :Tugas1-Nrp-NamaMahasiswa.pdf.
- Format Laporan:
 - Identitas Mahasiswa : Tugas-1, NRP, Nama.
 - Laporan Berisi Listing dan penjelasan program yang telah di buat.
- Laporan dikirim paling lambat pada :
Hari/Tanggal : Minggu/ 12 September 2021

3 Jawaban

Berikut adalah Listing 2 yang merupakan jawaban dari ketiga pertanyaan di atas. Pertanyaan nomor 1 terjawab pada fungsi DeleteAll, Pertanyaan Nomor 2 terjawab pada fungsi Biggest, dan pertanyaan nomor 3 terjawab pada fungsi DeleteBiggest.

Listing 2: JAWABAN

```
#include <iostream>
#include <stdio.h>

void Cetak ( float L [100] , int M ){
    for(int i =0; i <= M ; i ++){
        printf ( "%.2f\n",L [ i ] );
    }
}

void Insert ( float L [100] , int *M , int i , float X ){
    // Panjang array
    int N =100;
    int BACK ;
    if (* M <N -1){
        BACK =* M+1 ;
        // Menggeser Elemen Ke kanan
        while ( BACK > i ){
            L [ BACK ]= L [ BACK -1];
            BACK = BACK -1;
        }
        // Menyisipkan Nilai Pada posisi ke -i
        L [ i ]= X ;
        // Menambah Jumlah elemen
        *M = *M + 1;
    }
}

void Delete ( float L [100] , int *M , int i ){
    int BACK ;
    // Jika nilai i <0 atau i > posisi elemen maka tida tidak valid .
    if ((i <0)|| ( i >* M )) return ;
    BACK = i ;
    //Proses Menghapus dengan Menggeser ke Kiri
    for ( BACK = i ; BACK <* M ; BACK ++){
        L [ BACK ]= L [ BACK +1];
    }
    // Mengurangi posisi elemen terakhir
    *M = *M - 1;
```

```

}

int Search ( float L [100] , int M , float X ){
    // Parameter Input
    //L [100]: Array , panjang 100
    //M : Posisi Elemen terakhir
    //X : Nilai yang di cari
    // Output : Posisi elemen .
    // pos >=0 bila ditemukan
    // pos = -1 bila tidak ditemukan

    int C =0 , pos = -1;
    for ( C =0; C <= M ; C ++){
        if ( L [ C ]== X ){
            // Jika ditemukan maka
            // Simpan nilai C dan pencarian diakhiri .
            pos = C ;
            break ;
        }
    }
    return pos ;
}

void Update ( float L [100] , int i , float X ){
    // Parameter Input
    //L= Array panjang 10;
    //i: Posisi Elemen yang diperarui
    //X: Nilai yang baru

    L [ i ]= X ;
}

void Append ( float L [100] , int *M , float X ){
    // Definisikan Panjang array
    int N =100;
    if (*M <N -1){
        // Menambah Jumlah Elemen
        *M = *M + 1;
        // Masukkan X sebagai nilai index terakhir
        L [*M ]= X ;
    }
}

void DeleteAll (float L[100], int *M){
    //Definisikan Akhir Array
    int BACK = *M;

```

```

        //Proses penghapusan
        for (int i=BACK ; i>=0; i--){
            Delete(L,&BACK,i);
        }
        *M = BACK;
    }

float Biggest (float L[100], int M){
    float temp=0;
    // Traversal untuk mencari nilai terbesar
    for (int i=0; i<=M; i++){
        if (L[i]>temp){
            temp=L[i];
        }
    }
    return temp;
}

void DeleteBiggest (float L[100], int *M, float big){
    //Mendefinisikan Akhir Array
    int BACK = *M;
    //Mencari elemen yang memiliki nilai sama dengan nilai maksimum
    for (int i=BACK; i>=0; i--){
        if (L[i]==big){
            //Proses Menghapus
            Delete(L,&BACK,i);
        }
    }
    *M = BACK;
}

int main (){
    // Index array pada C dimulai dari 0.
    float L [100];

    //M= -1 Array kosong
    int M ;
    M = -1;

    // Menambah lima Item pada array .
    printf("Item_Telah_Ditambahkan!\n");
    Append (L ,& M ,4);
    Append (L ,& M ,5);
    Append (L ,& M ,6);
    Append (L ,& M ,5);
    Append (L ,& M ,1);
}

```

```

Cetak (L , M );
system("PAUSE");

// Mencari Posisi item yang bernilai 6.
int pos ;
int x=6;
printf("\nMencari_item_bernilai%d...\n",x);
pos = Search (L ,M ,x);
printf ("Posisi_item_di_index_ke-%d!\n", pos );
system("PAUSE");

// Hapus elemen yang telah disimpan di variabel pos .
printf("\nMenghapus_item_di_index_ke%d...\n", pos);
Delete (L ,& M , pos );
printf("Item_Telah_Dihapus!\n");
Cetak (L , M );
system("PAUSE");

// Menyisipkan nilai 8 pada posisi index 4;
float a=15;
int b=2;
printf("\nMenyisipkan_nilai%.f_pada_index_ke%d...\n",a,b);
Insert (L ,& M ,b, a);
printf("Item_Telah_Ditambahkan!\n");
Cetak (L , M );
system("PAUSE");

//Menghapus Semua Nilai dari Array
printf("\nMenghapus_semua_item...\n");
DeleteAll(L, &M);
printf("Item_Telah_Dihapus!\n");
system("PAUSE");

//Menambahkan Item Baru
system("CLS");
printf("Membuat_Array_Baru...\n");
Append (L ,& M ,4);
Append (L ,& M ,10);
Append (L ,& M ,1);
Append (L ,& M ,10);
Append (L ,& M ,9);
printf("Array_Baru_Telah_Dibuat!\n");
Cetak (L , M );
system("PAUSE");

```



```

//Mengubah Nilai
printf("\nMengubah_nilai_item_index_ke-%d_menjadi%.f\n",b,a);
Update(L, b, a);
printf("Nilai_telah_diubah!\n");
Cetak (L, M);
system("PAUSE");

//Mencari Nilai Maksimum
printf("\nMencari_nilai_maksimum...\n");
float big;
big=Biggest(L,M);
printf("Nilai_maksimumnya_adalah%.f\n",big);
system("PAUSE");

//Menghapus Nilai Maksimum
printf("\nMenghapus_nilai_maksimum...\n");
DeleteBiggest(L,&M,big);
printf("Data_Dengan_Nilai%.f_Telah_Dihapus!\n",big);
Cetak (L , M );

return 0;
}

```