**TUGAS PRAKTIKUM ALGORITMA & STRUKTUR DATA**

**Jilid 12**



**Oleh :**

**Nama : Rosi Arif Mulyadi**

**NRP : 3121522021**

**Prodi : D3 Teknik Informatika PENS PSDKU Sumenep**

**Kelas : 1 ITA D3 Sumenep**

**Dosen :**

**LUSIANA AGUSTIEN M.Kom**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**MODUL 6.1**

**QUEUE / ANTRIAN ((implementasi Queue dengan Array Sirkular)**

**KEGIATAN PRAKTIKUM**

1. Lakukukan Analisa terhadap setiap listing program dari ujicoba dan implementasi yang sudah di lakukan pada modul diatas.

Jawab :

**Implementasi Queue Sirkular menggunakan Array**

Listing Program :

#include <stdio.h>

#include <stdlib.h>

#define MAKS 5

int sisip();

int hapus();

int tampil();

struct antrian

{

int q[MAKS];

int d, b;

}que;

int main()

{

int ch = 0;

que.d = 0;

que.b = 0;

while(ch!=4)

{

printf("\n\n\t\t MENU ");

printf("\n\n\t1. Penyisipan ");

printf("\n\t2. Penghapusan ");

printf("\n\t3. Tampilkan ");

printf("\n\t4. Keluar ");

printf("\n\n\t Masukkan pilihan anda : \t");

scanf("%d", &ch);

switch(ch)

{

case 1:

sisip();

break;

case 2:

hapus();

break;

case 3:

tampil();

break;

}

}

getch();

return 0;

}

int tampil()

{

int i;

printf("\n\n");

if(que.d==0 && que.b==0)

{

printf("\n\n\t Antrian kosong");

return 0;

}

printf("\n\n\t Element-element Antrian adalah : ");

if(que.b<que.d)

{

for(i=que.d+1; i<MAKS; i++)

printf("\n\t%d", que.q[i]);

for(i=0; i<=que.b; i++)

printf("\n\t%d", que.q[i]);

}else

for(i=que.d+1; i<=que.b; i++)

printf("\n\t%d", que.q[i]);

return 0;

}

int sisip()

{

int item;

if((que.b+1)%MAKS==que.d)

{

printf("\n\n\tAntrian penuh");

return 0;

}else

{

que.b=(que.b+1)%MAKS;

printf("\n\n\t Masukkan Element yang akan disisipkan : ");

scanf("%d", &item);

que.q[que.b]=item;

}

return 0;

}

int hapus()

{

int item;

if(que.d==que.b)

{

printf("\n\n\t Antrian kosong");

return 0;

}

que.d=(que.d+1)%MAKS;

item=que.q[que.d];

if(que.d==que.b)

{

que.b=0;

que.d=0;

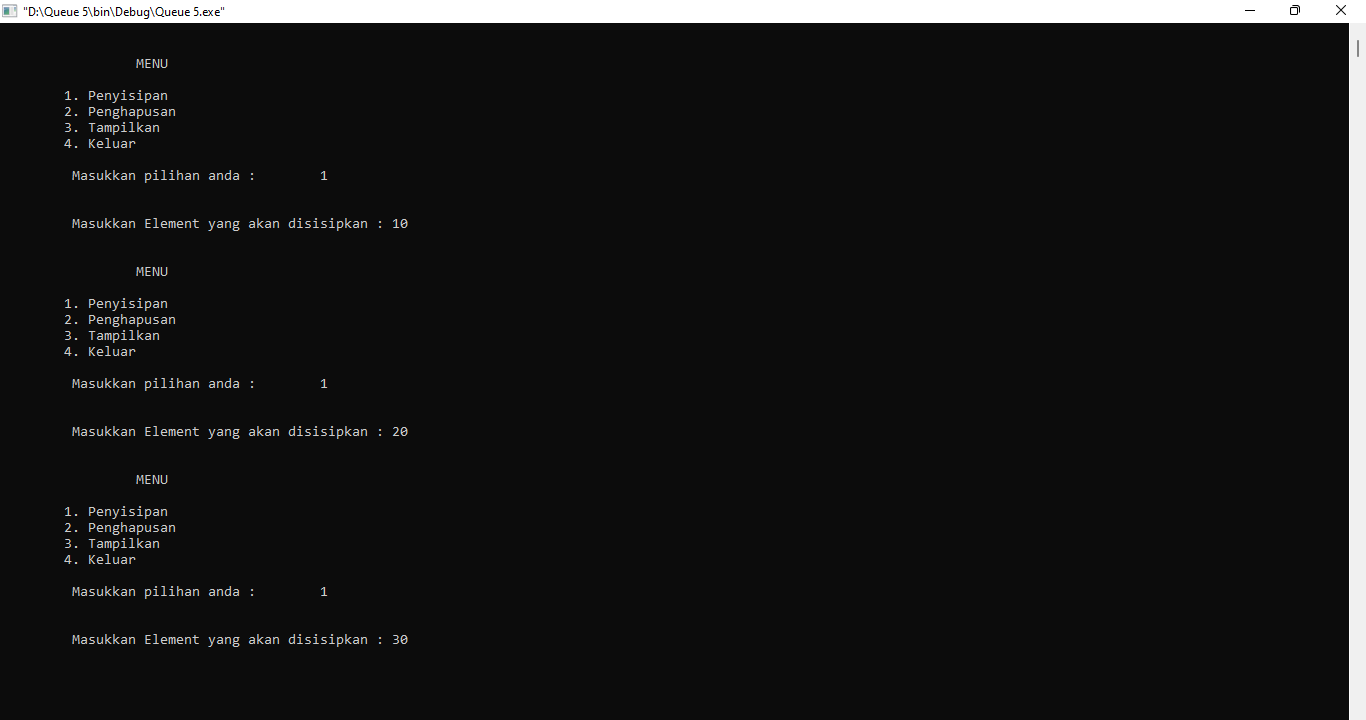
}

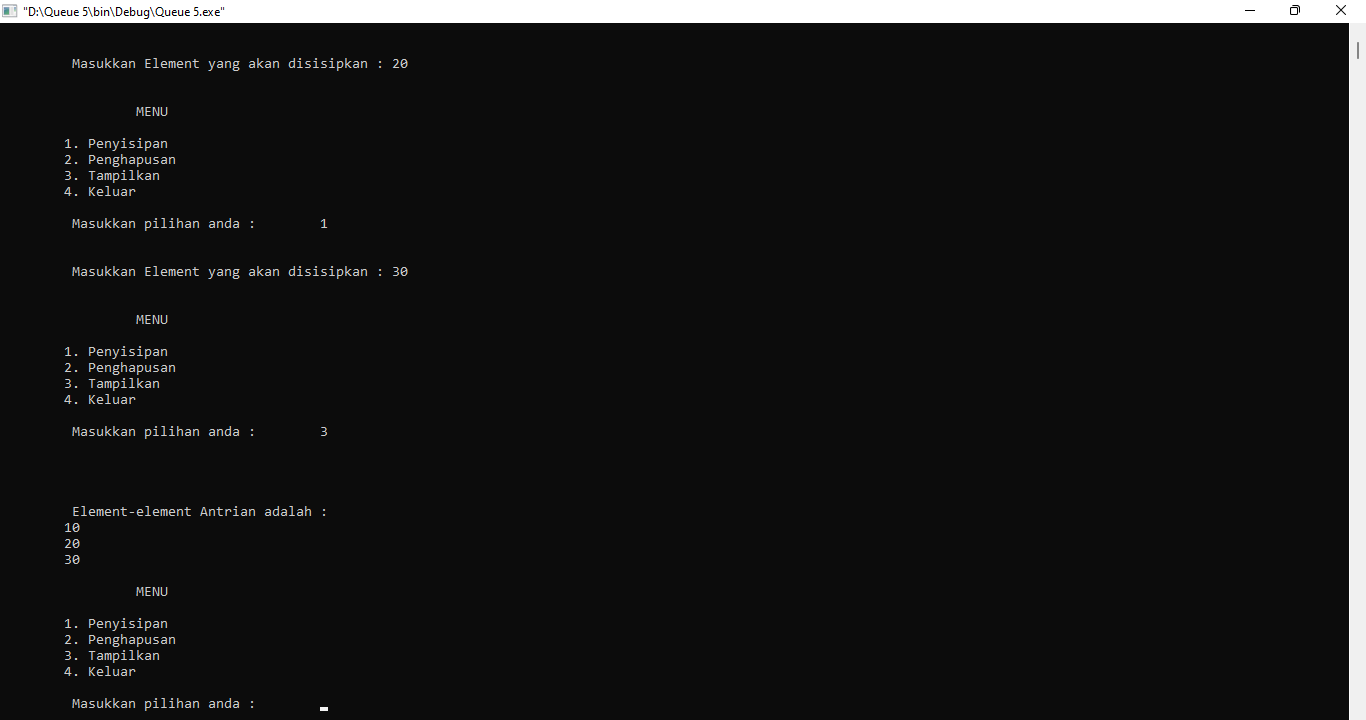
printf("\n\n\t Item yang dihapus adalah : %d", item);

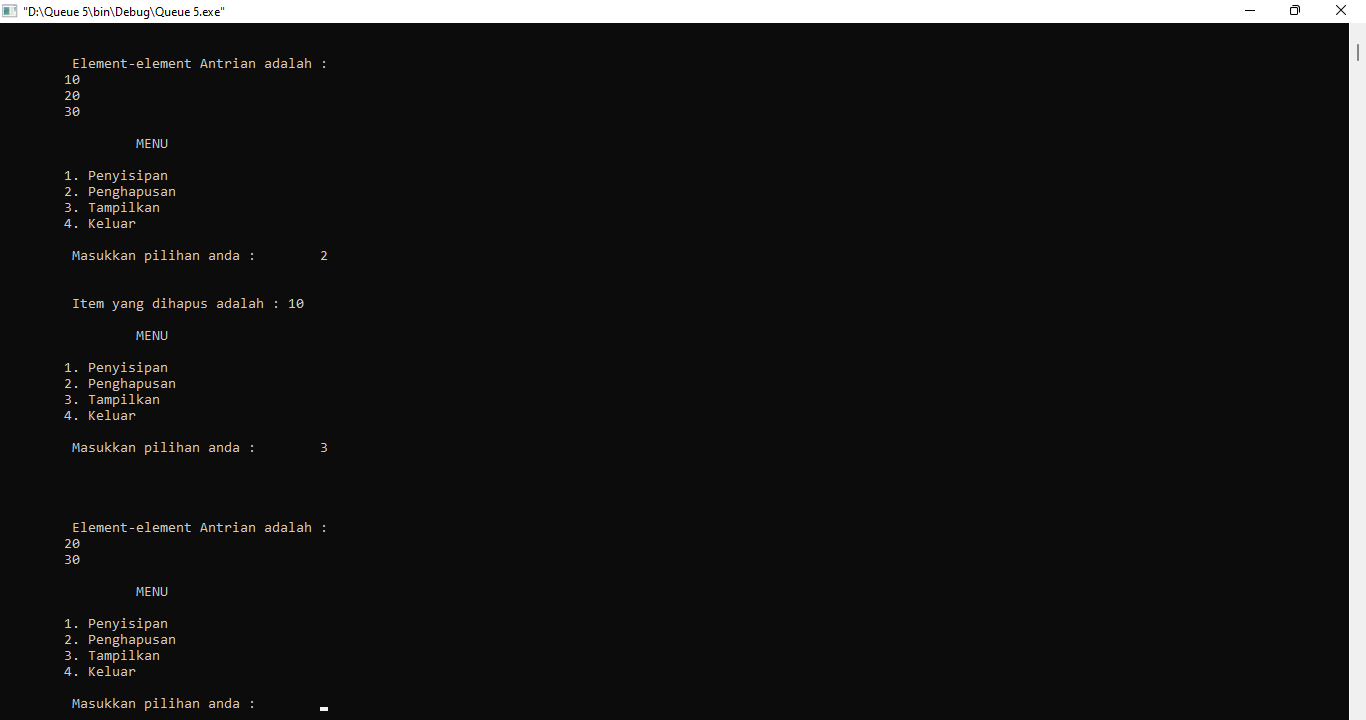
return 0;

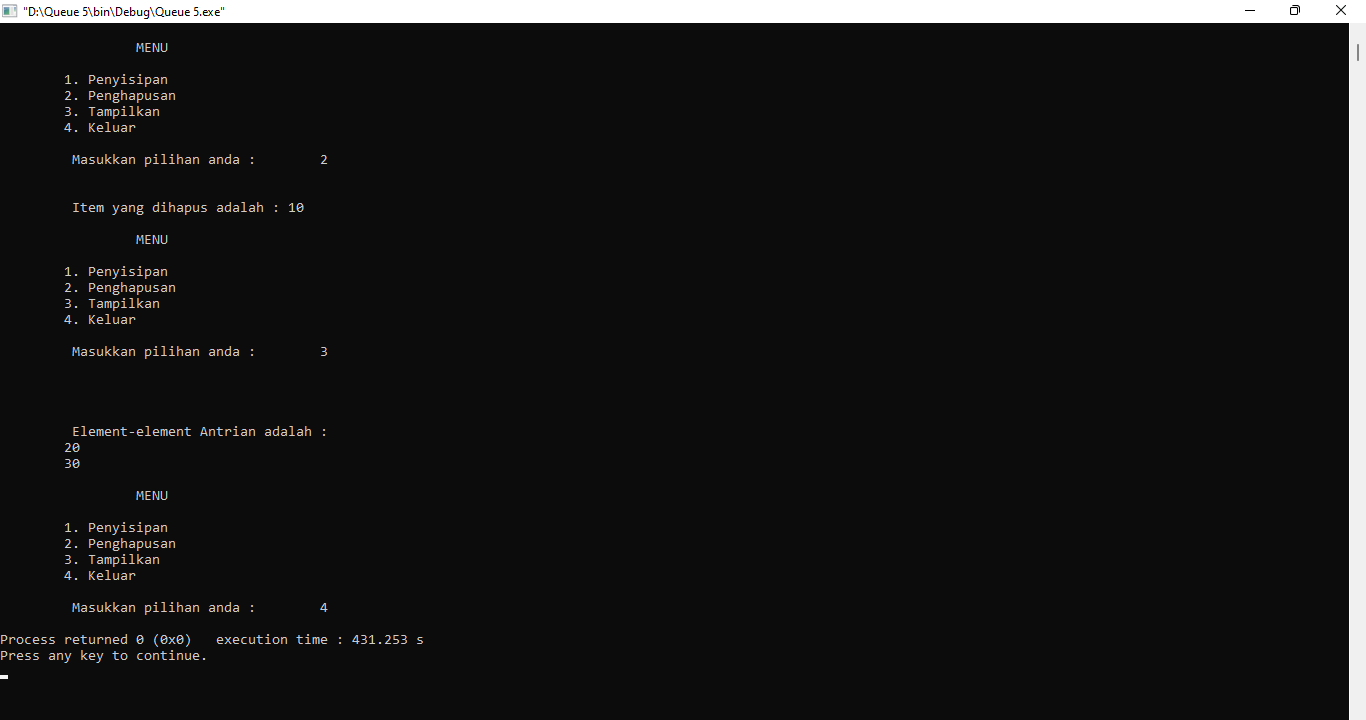
}

Output :









Analisa :

* Pada ujicoba program diatas dapat disimpulkan bahwa Program yang menggunakan **Queue Sirkular** dengan **Array** itu dapat mengurutkan data yang dimasukan dengan cara mengantri.
* Dalam uji coba program tersebut **Queue Sirkular** berperan untuk membuat data yang kita inputkan mengantri dengan rapi.
* Bedanya dengan **Linked list** pada **Array** ini bukan data yang kita inputkan yang berurutan tapi value dari data tersebut dan pada **Queue Sirkular** menggunakan **Array** itu lokasi dari data yang diinputkan akan berbeda dan apabila dihapus secara berurutan maka yang urutan setelahnya tidak bisa menempati lokasi data sebelumnya.

2. Kerjakan kegiatan praktikum yang ada pada poin F.

**ADA DI BAWAH SOAL DAN JAWABAN**

**TUGAS PRAKTIKUM**

1. Buatlah sebuah algoritma dan program untuk mengimplementasikan antrian sirkular menggunakan linkedlist.

Jawab :

**Implementasi Queue Sirkular menggunakan Linked list**

Algoritma :

* Gunakan rumus seperti menggunakan **Linked list.**
* Masukkan rumus-rumus tersebut dan gabungkan dengan rumus dari **Queue.**
* Masukkan if((a==NULL)&&(b==NULL)) (pada bagian Tampil)

Masukkan n->q = nil;

n->d = NULL;

n->b = NULL;

n->next = NULL; (pada bagian Sisipan)

* Masukkan if((a==NULL)&&(b==NULL)) yang juga bisa digunakan untuk menghapus (pada bagian Hapus).
* Pada rumus **Queue Sirkular** yang berbasis **Linked list** disaat akan menampilkan itu akan mengurutkan (Antrian) dan apabila dihapus data inputan yang pertama, maka data inputan berikutnya dapat mengisi data inputan sebelumnya.

Listing Program :

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

int sisip(int nil);

int hapus();

int tampil();

struct Antrian

{

int q;

int d, b;

struct Antrian\* next;

};

struct Antrian \*a = NULL;

struct Antrian \*b = NULL;

int main()

{

int ch = 0;

que.d = 0;

que.b = 0;

while(ch!=4)

{

printf("\n\n\t\t MENU ");

printf("\n\t1. Penyisipan ");

printf("\n\t2. Penghapusan ");

printf("\n\t3. Tampilkan ");

printf("\n\t4. Keluar ");

printf("\n\n\t Masukkan pilihan anda : \t");

scanf("%d", &ch);

switch(ch)

{

case 1:

sisip();

break;

case 2:

hapus();

break;

case 3:

tampil();

break;

}

}

getch();

return 0;

}

int tampil()

{

struct node\* t;

t = a;

if((a==NULL)&&(b==NULL))

printf("Antrian Kosong");

else{

do{

printf("\n%d",t->q, t->d, t->b, t->next);

t = t->q, t->d, t->b, t->next;

}while(t != a);

}

}

int sisip(int nil)

{

struct Antrian\* n;

n = (struct Antrian\*)malloc(sizeof(struct Antrian));

n->q = nil;

n->d = NULL;

n->b = NULL;

n->next = NULL;

if((b==NULL)&&(a==NULL))

{

a = b = n;

b->next = a;

}

else

{

b->next = n;

b= n;

n->next = a;

}

}

int hapus()

{

struct Antrian\* t;

t = a;

if((a==NULL)&&(b==NULL))

printf("Antrian Kosong");

else if(a == b){

a = b = NULL;

free(t);

}

else{

a = a->next;

b->next = a;

free(t);

}

}

Output :

