

# **LAPORAN FINAL PROJECT**

## **STRUKTUR DATA**



### **Disusun oleh :**

- |                           |               |
|---------------------------|---------------|
| 1. Fatwa Sifaun Nahar     | (20081010073) |
| 2. Imam Maskuri           | (20081010074) |
| 3. Faris Syaifulloh       | (20081010078) |
| 4. Mohammad Syarifuz Zaim | (20081010081) |
| 5. Aldo Pradana Ariando   | (20081010110) |

### **Mata Kuliah :**

Struktur Data A081

### **Dosen Pengampu :**

Fawwaz Ali Akbar, S.Kom, M.Kom

## **PROGRAM STUDI INFORMATIKA**

## **FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR**

**JAWA TIMUR**

**2022**

## A. Deskripsi Singkat Final Project (Antrian Sunat)

Pada final project kami membuat sebuah program yang menerapkan dua jenis struktur data yaitu **single linked list** dan **queue** dengan menggunakan bahasa C. Program yang kami buat adalah program antrian sunat masal yang dioperasikan oleh perawat atau admin untuk mendata semua anak yang mengikuti sunat masal. Alasan kami menggunakan single linked list dan queue adalah karena single linked list merupakan struktur data yang terdiri atas sekumpulan data bertipe sama dan memperhatikan urutan. Sedangkan alasan kami menggunakan queue adalah karena jenis struktur data tersebut berupa linear list dimana data dimasukkan melalui sebuah ujung yang disebut end/rear dan dihapus dari ujung lain yang disebut front, alur input pada queue yaitu FIFO atau First In - First Out.

Berdasarkan hal tersebut, kami menerapkan dua jenis struktur data yang dimana queue untuk menyimpan seluruh urutan dan kerja antrian sunat masal yang meliputi ambil antrian, panggil antrian, tampilkan antrian yang tersisa atau belum terpanggil, hingga destroy antrian. Sedangkan untuk single linked list kami terapkan untuk menampilkan seluruh informasi terkait dengan data peserta sunat masal. Oleh karena itu, input dari program ini akan dinamis atau berdasarkan kebutuhan user dan output dari program ini juga akan sesuai dengan keinginan dari pilihan user pada program.

## B. Penjelasan Source Code Program

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 #include<conio.h>
5
6 #define MAX 5
7
8 struct nodeKhitan{
9     int nomorAntrian;
10    int idKhitan;
11    int usiaPeserta;
12    char namaPeserta[50];
13    char alamatPeserta[50];
14    char jenisKhitan[50];
15
16    struct nodeKhitan *next;
```

```

17 };
18 typedef struct nodeKhitan nodeKhitan;
19
20 struct queueKhitan{
21     int count;
22     nodeKhitan *front;
23     nodeKhitan *rear;
24 };
25 typedef struct queueKhitan queueKhitan;
26
27 //=====
=====

```

Pada line 1-27 ini, pertama kami import dulu fungsi - fungsi yang sudah didefinisikan pada empat header file, diantara lain yaitu `stdio.h`, `stdlib.h`, `string.h`, dan `conio.h`. Keempat header file tersebut berisi deklarasi fungsi - fungsi dasar yang kami butuhkan untuk membuat program antrian sunat ini. Selanjutnya, kami definisikan variabel `MAX` yang berisi nilai interger sebesar 5 untuk batas maksimal pada antrian sunat. Berikutnya kami membuat dua node untuk dua jenis struktur data yang kami terapkan yaitu `nodeKhitan` yang menyimpan enam data dengan satu pointer `next`, lalu untuk node `queueKhitan` menyimpan satu data dengan dua pointer dari `nodeKhitan` yaitu pointer `front` dan pointer `rear` untuk alur queue-nya.

```

28 queueKhitan createQueueKhitan(void);
29 nodeKhitan *createNode(void);
30 int ambilAntrianKhitan(queueKhitan *myQueue, int nomorAntrian);
31 int panggilAntrianKhitan(queueKhitan *myQueue, int idKhitan,
    nodeKhitan **pHead);
32 void tampilkanAntrianKhitan(queueKhitan myQueue);
33 int destroyQueueKhitan(queueKhitan *myQueue);
34 void tampilkanDataPesertaKhitan(nodeKhitan *pHead);
35
36
37 int main(){
38     int nomorAntrian = 0, idKhitan = 1000;
39     char pilihMenu;
40
41     queueKhitan myQueue;
42     myQueue = createQueueKhitan();
43     nodeKhitan *pHead = NULL;
44
45     do{
46         fflush(stdin);
47         system("cls");
48         printf("===== PROGRAM ANTRIAN KHITAN DENGAN DATA PESERTA
    KHITAN =====\n\n");

```

49	printf("1. Ambil Antrian Khitan\n");
50	printf("2. Panggil Antrian Khitan\n");
51	printf("3. Tampilkan Antrian Khitan\n");
52	printf("4. Hapus Semua Antrian Khitan\n");
53	printf("5. Tampilkan Data Peserta Khitan\n");
54	printf("q. Keluar\n");
55	printf("\nPilih Menu : ");
56	scanf("%c", &pilihMenu);
57	

Pada line 28 - 34, kami membuat tujuh fungsi dengan parameternya masing - masing yang dimana isi dari fungsi - fungsi tersebut akan dibuat dibawah fungsi main. Semua paramater yang ada pada ketujuh fungsi tersebut kami definisikan berdasarkan kebutuhan dari pembuatan program ini. Selanjutnya, kami buat fungsi main agar semua fungsi yang dibuat sebelumnya dapat berjalan dan user dapat mengoperasikan program sesuai dengan keperluannya berdasarkan fungsi - fungsi yang telah didefinisikan pada fungsi main. Namun sebelum itu, kami definisikan terlebih dahulu tiga variabel seperti interger nomorAntrian, interger idKhitan, dan char pilihMenu, lalu panggil fungsi queueKhitan dengan pointer pHead yang berisikan NULL untuk penerapan queue pada antrian sunat. Setelah itu, kami membuat perintah printf untuk menampilkan teks pada program dengan maksud sebagai keterangan untuk user agar dapat menentukan pilihannya pada program.

58	switch(pilihMenu){
59	case '1':
60	nomorAntrian = ambilAntrianKhitan(&myQueue,
	nomorAntrian);
61	break;
62	
63	case '2':
64	idKhitan = panggilAntrianKhitan(&myQueue, idKhitan,
65	&pHead);
	break;
66	
67	case '3':
68	tampilkanAntrianKhitan(myQueue);
69	break;
70	
71	case '4':
72	nomorAntrian = destroyQueueKhitan(&myQueue);
73	break;
74	
75	case '5':
76	tampilkanDataPesertaKhitan(pHead);

77	break;
78	
79	case 'q':
80	printf("Terima Kasih Telah Menggunakan Program Ini\n");
81	break;
82	
83	default:
84	printf("Pilihan Menu Tidak Tersedia\n");
85	getch();
86	break;
87	}
88	} while(pilihMenu != 'q');
89	}
90	
91	
92	//=====
	=====

Pada line 58 - 92, kami buat percabangan swith case yang dimana setiap percabangan memanggil fungsinya masing - masing, seperti case 1 menetapkan variabel nomorAntrian sama dengan fungsi ambilAntrianKhitan, lalu case 2 menetapkan variabel idKhitan sama dengan fungsi panggilAntrianKhitan, selanjutnya case 3 memanggil fungsi tampilkanAntrianKhitan, berikutnya case 4 menetapkan variabel nomorAntrian sama dengan fungsi destroyQueueKhitan, dan case 5 memanggil fungsi tampilkanDataPesertaKhitan. Lalu ketika user menginputkan angka diluar case yang ditetapkan maka program akan memanggil percabangan deafult yang akan menampilkan teks berupa “Pilihan Menu Tidak Tersedia” dan program akan kembali menampilkan menu awal pada layar. Hal ini bertujuan untuk memanfaatkan cara kerja dari single linked list yang kami terapkan yaitu jumlah input dinamis yang berarti program dapat dioperasikan sesuai dengan keperluan atau kebutuhan user, namun dengan batas maksimal sebanyak 5 antrian. Pada percabangan yang kami buat, terdapat satu fungsi perulangan do while yang dimana program akan selalu melakukan perulangan percabangan ketika input dari user bukanlah ‘q’. Namun, ketika inputnya adalah ‘q’ maka program akan berhenti melakukan perulangan dan program akan menampilkan sebuah teks berupa “Terima Kasih Telah Menggunakan Program Ini” lalu exit dengan sendirinya.

93	
94	queueKhitan createQueueKhitan(void){
95	queueKhitan myQueue;

96	myQueue.count = 0;
97	myQueue.front = NULL;
98	myQueue.rear = NULL;
99	return myQueue;
100	}
101	
102	//=====
	=====

Pada line 94 - 100 merupakan fungsi createQueueKhitan yang memiliki parameter void dengan tipe struct queueKhitan. Isi dari fungsi ini adalah untuk mendefinisikan isi dari data dan pointer yang ada pada struct queueKhitan, seperti data count yang berisi 0, pointer front berisi NULL, dan pointer rear berisi NULL. Setelah itu, semua definisi tersebut akan dikembalikan ke myQueue untuk dapat dipanggil atau diproses pada fungsi - fungsi lainnya.

103	
104	nodeKhitan *createNode(void){
105	nodeKhitan *newNode = (nodeKhitan* )malloc(sizeof(nodeKhitan));
106	return newNode;
107	}
108	
109	//=====
	=====

Pada line 104 - 107 merupakan fungsi createNode yang memiliki parameter void dengan tipe struct nodeKhitan. Tujuan dari isi pada fungsi createNode adalah untuk mengalokasikan node baru secara dinamis dengan fungsi malloc yang berarti sebuah fungsi fasilitas untuk memesan tempat secara berurutan untuk tipe data pointer dengan jumlah data dinamis. Lalu dikembalikan ke newNode untuk dapat dipanggil atau diproses pada fungsi - fungsi lainnya.

110	
111	int ambilAntrianKhitan(queueKhitan *myQueue, int nomorAntrian){
112	system("cls");
113	printf("===== AMBIL ANTRIAN KHITAN =====\n\n");
114	if(myQueue->count == MAX){
115	printf("Antrian Penuh Datang Lagi Besok\n");
116	}
117	else {
118	nomorAntrian++;
119	nodeKhitan *newNode = createNode();
120	newNode->nomorAntrian = nomorAntrian;
121	
122	fflush(stdin);

```

123     printf("Masukkan Nama Peserta : ");
124     gets(newNode->namaPeserta);
125     printf("Nomor Antrian Anda : %d\n", newNode->nomorAntrian);
126     newNode->next = NULL;
127
128     if(myQueue->front == NULL){
129         myQueue->front = newNode;
130     }
131     else {
132         myQueue->rear->next = newNode;
133     }
134     myQueue->rear = newNode;
135     myQueue->count++;
136     printf("\nAntrian Berhasil Dibuat\n");
137 }
138 getch();
139 return nomorAntrian;
140 }
141
142 //=====
=====

```

Pada line 111-142 merupakan fungsi ambil antrian khitan atau bisa dikatakan enqueue. Cara kerja dari fungsi ini yaitu pertama nomer antrian pada parameter akan dibuat increment sehingga akan terus bertambah setiap kali antrian ditambah. selanjutnya yaitu proses memasukkan data antrian ke dalam data queue pertama dibuat newnode untuk memasukkan data baru. selanjutnya semua data baru akan dimasukkan ke dalam newnode. setelah semua data sudah dimasukkan selanjutnya newnode akan dimasukkan kedalam queue dengan beberapa aturan jika queue front sudah terisi maka akan lanjut rear next sampai data pada queue rear null sehingga data newnode akan terisi pada data rear. selanjutnya yaitu return nomer antrian untuk mengirimkan nomer antrian sekarang dari fungsi ambil antrian kedalam fungsi main.

```

143
144 int panggilAntrianKhitan(queueKhitan *myQueue, int idKhitan,
nodeKhitan **pHead){
145     system("cls");
146     int jenisKhitan;
147
148     printf("===== PANGGIL ANTRIAN KHITAN =====\n\n");
149     if(myQueue->count == 0){
150         printf("Antrian Kosong\n");
151     }
152     else {
153         nodeKhitan *delAntrian = myQueue->front;

```

```

154     myQueue->front = myQueue->front->next;
155     myQueue->count--;
156     printf("Antrian %d Telah Dipanggil\n", delAntrian-
>nomorAntrian);
157     printf("Nama Peserta : %s\n", delAntrian->namaPeserta);
158
159     //INSERT DATA PESERTA KHITAN TO LINKED LIST
160     idKhitan++;
161     nodeKhitan *newNode = createNode();
162     newNode->idKhitan = idKhitan;
163     strcpy(newNode->namaPeserta, delAntrian->namaPeserta);
164
165     printf("\n\n===== DATA PESERTA KHITAN =====\n\n");
166     printf("Masukkan Usia Peserta : ");
167     scanf("%d", &newNode->usiaPeserta);
168     printf("Alamat Peserta      : ");
169     fflush(stdin);
170     gets(newNode->alamatPeserta);
171
172     do{
173         printf("\n== Masukkan Jenis Khitan ==\n");
174         printf("1. Khitan Laser\n");
175         printf("2. Khitan Gunting\n");
176         printf("3. Khitan Bambu\n");
177         printf("Pilih Jenis Khitan    : ");
178         scanf("%d", &jenisKhitan);
179         if(jenisKhitan == 1){
180             strcpy(newNode->jenisKhitan, "Khitan Laser");
181         }
182         else if(jenisKhitan == 2){
183             strcpy(newNode->jenisKhitan, "Khitan Gunting");
184         }
185         else if(jenisKhitan == 3){
186             strcpy(newNode->jenisKhitan, "Khitan Bambu");
187         }
188         else {
189             printf("Jenis Khitan Tidak Tersedia\n");
190         }
191     } while(jenisKhitan < 1 || jenisKhitan > 3);
192
193     newNode->next = NULL;
194     if(*pHead == NULL){
195         *pHead = newNode;
196     }
197     else {
198         nodeKhitan *pWalker = *pHead;
199         while(pWalker->next != NULL){
200             pWalker = pWalker->next;
201         }
202         pWalker->next = newNode;

```



```

203     }
204
205     free(delAntrian);
206     printf("\nData Peserta Khitan Berhasil Ditambahkan\n");
207 }
208 getch();
209 return idKhitan;
210 }
211
212 //=====
=====

```

Pada line 144-210 digunakan untuk fungsi memanggil antrian dan dimasukkan kedalam linked list sunat. Jadi fungsi ini akan menghapus queue front yang akan dimasukkan kedalam linked list selanjutnya queue front akan dihapus dan diubah kedalam queue front next, sebelum dihapus data dari queue front akan disalin kedalam nodekhitan delantrian. Selanjutnya yaitu melengkapi data dari pasien khitan seperti umur, alamat, dan jenis khitan. setelah semuanya diisi data akan dimasukkan ke node delantrian dan dari node delantrian akan dimasukkan kedalam nodekhitan atau linked list. Cara memasukkan delantrian ke nodekhitan sama seperti insert pada linked list yaitu cek apakah head NULL, jika iya maka delantrian akan dimasukkan kedalam node head jika tidak maka akan dimasukkan kedalam node head next menggunakan bantuan pWalker.

```

213
214 int destroyQueueKhitan(queueKhitan *myQueue){
215     nodeKhitan *pWalker = myQueue->front;
216     while(pWalker != NULL){
217         nodeKhitan *temp = pWalker;
218         pWalker = pWalker->next;
219         free(temp);
220     }
221     myQueue->front = NULL;
222     myQueue->rear = NULL;
223     myQueue->count = 0;
224     system("cls");
225     printf("===== HAPUS ANTRIAN KHITAN =====\n\n");
226     printf("Antrian Berhasil Dihapus\n");
227     getch();
228     return 0;
229 }
230
231 //=====
=====

```

Pada line 214-213 digunakan untuk fungsi menghapus queue antrian khitan. Fungsi ini akan destroy atau menghapus semua antrian yang ada sehingga queue menjadi kosong. Cara juga cukup mudah pertama dicek dari queue front menggunakan pWalker apakah ada isinya, jika tidak ada maka pWalker next atau menuju node selanjutnya, namun jika ada isinya maka pWalker akan dimasukkan kedalam temp dan temp akan dihapus dengan free. selanjutnya jika semua data sudah dihapus akan dimasukkan data pada queue front dan rear berupa NULL dan count menjadi 0.

```

232
233 void tampilkanAntrianKhitan(queueKhitan myQueue){
234     system("cls");
235     printf("===== TAMPILKAN ANTRIAN KHITAN =====\n\n");
236
237     if(myQueue.count == 0){
238         printf("Antrian Kosong\n");
239     }
240     else {
241         nodeKhitan *pWalker = myQueue.front;
242
243         while(pWalker != NULL){
244             printf("%d -> %s\n", pWalker->nomorAntrian, pWalker->
namaPeserta);
245             pWalker = pWalker->next;
246         }
247     }
248     getch();
249 }
250
251 //=====
=====

```

Pada line 232-248 digunakan untuk menampilkan data antrian pada queue. caranya yaitu dengan menggunakan pWalker untuk queue front selanjutnya pWalker akan menampilkan datanya dan setelah ditampilkan pWalker akan diubah menjadi pWalker next sehingga menunjuk queue next. proses akan diulang sampai pWalker berupa data NULL.

```

252
253 void tampilkanDataPesertaKhitan(nodeKhitan *pHead){
254     system("cls");
255     printf("===== TAMPILKAN DATA PESERTA KHITAN =====\n\n");
256
257     if(pHead == NULL){

```

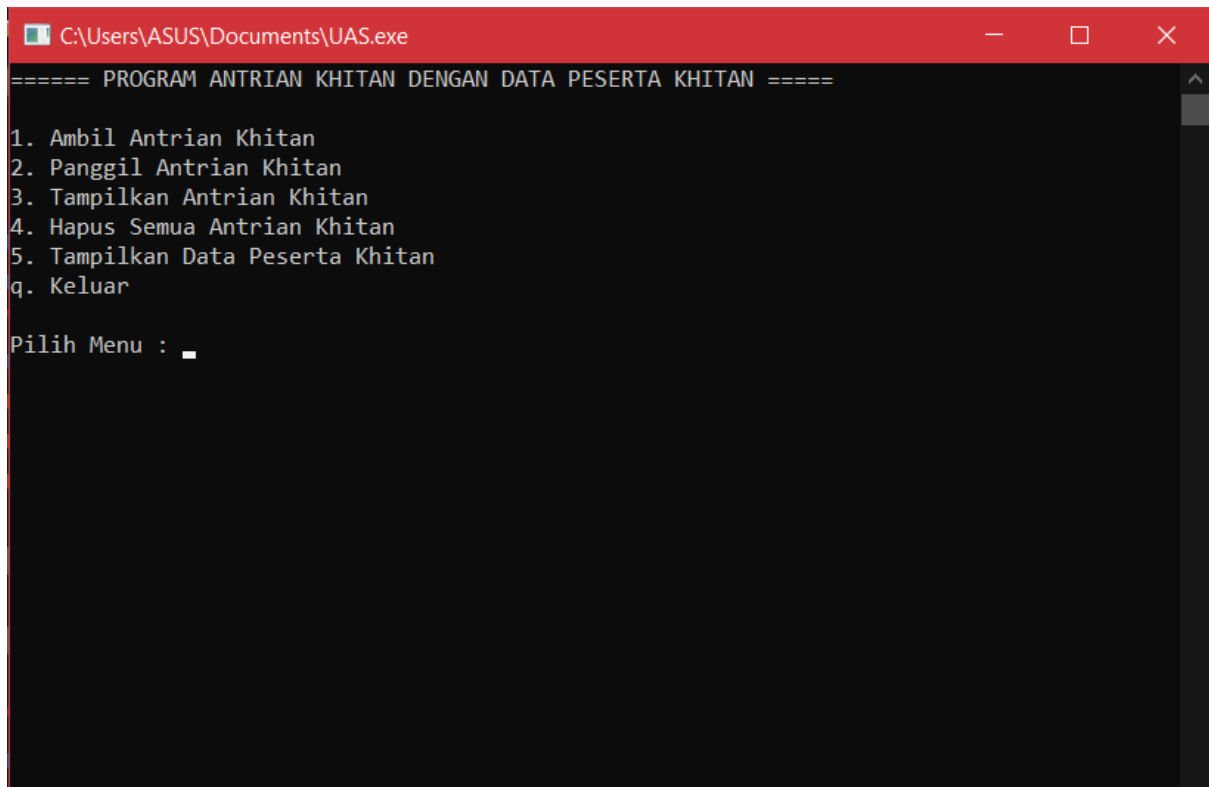
```

258         printf("Data Peserta Khitan Kosong\n");
259     }
260     else {
261         nodeKhitan *pWalker = pHead;
262
263         while(pWalker != NULL){
264             printf("ID Khitan          : %d\n", pWalker->idKhitan);
265             printf("Nama Peserta      : %s\n", pWalker-
>namaPeserta);
266             printf("Usia Peserta       : %d\n", pWalker-
>usiaPeserta);
267             printf("Alamat Peserta    : %s\n", pWalker-
>alamatPeserta);
268             printf("Jenis Khitan      : %s\n", pWalker-
>jenisKhitan);
269             printf("=====\n");
270             pWalker = pWalker->next;
271         }
272     }
273     getch();
274 }

```

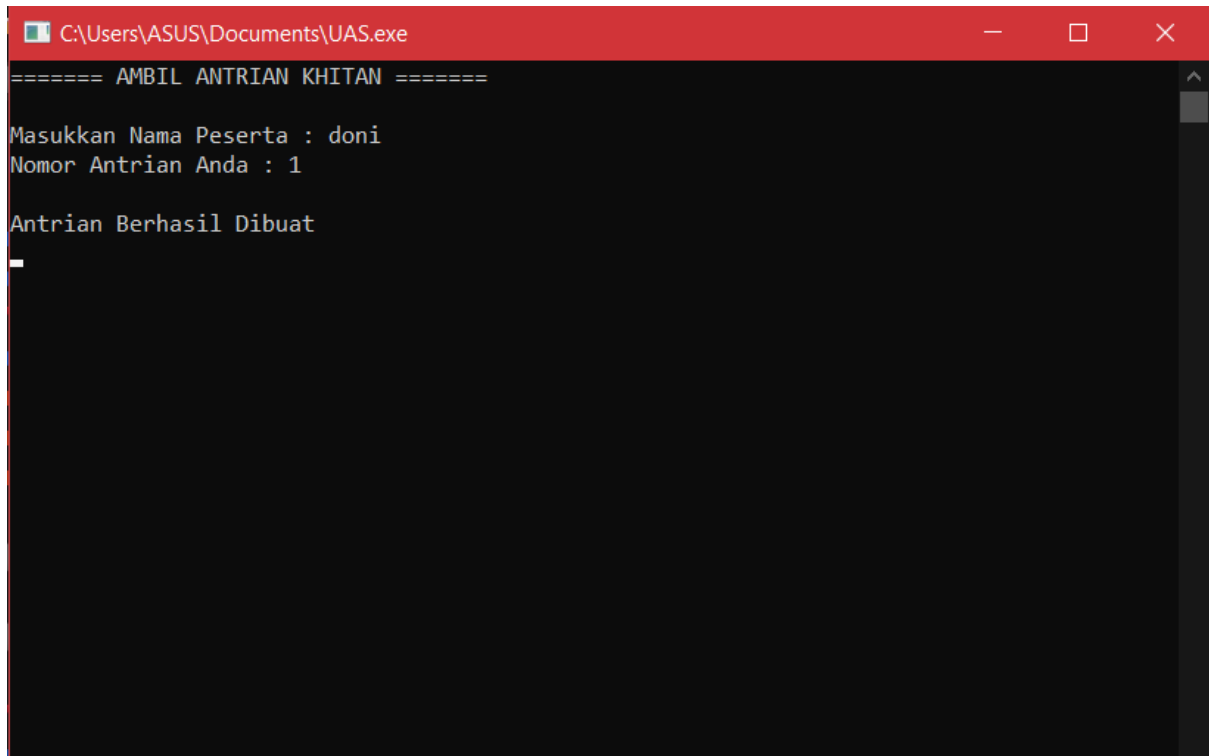
Pada line 253-270 akan digunakan untuk menampilkan peserta yang sudah dikhitan dengan cara menampilkan linked list yang sudah ada. Caranya juga hampir sama dengan menampilkan antrian yaitu dengan bantuan Pwalker untuk traversal semua node dan menampilkan masing masing node sampai node akhir dengan syarat bahwa data pada pWalker adalah NULL atau tidak ada isinya.

### C. HASIL RUNNING PROGRAM

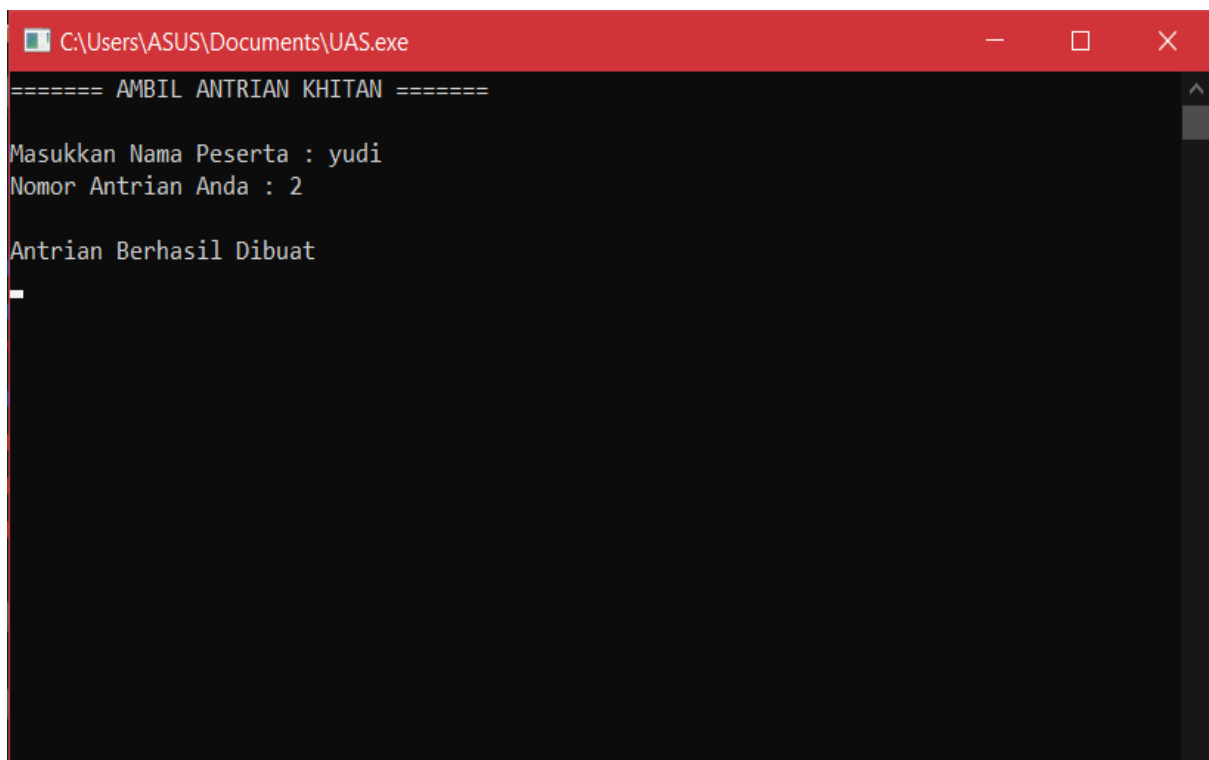


```
C:\Users\ASUS\Documents\UAS.exe
===== PROGRAM ANTRIAN KHITAN DENGAN DATA PESERTA KHITAN =====
1. Ambil Antrian Khitan
2. Panggil Antrian Khitan
3. Tampilkan Antrian Khitan
4. Hapus Semua Antrian Khitan
5. Tampilkan Data Peserta Khitan
q. Keluar
Pilih Menu : _
```

Gambar 1. Menu awal



Gambar 2. Input antrian pertama



Gambar 3. Input antrian kedua

```
C:\Users\ASUS\Documents\UAS.exe
===== PANGGIL ANTRIAN KHITAN =====

Antrian 1 Telah Dipanggil
Nama Peserta : doni

===== DATA PESERTA KHITAN =====

Masukkan Usia Peserta : 13
Alamat Peserta      : banyu urip

== Masukkan Jenis Khitan ==
1. Khitan Laser
2. Khitan Gunting
3. Khitan Bambu
Pilih Jenis Khitan   : 1

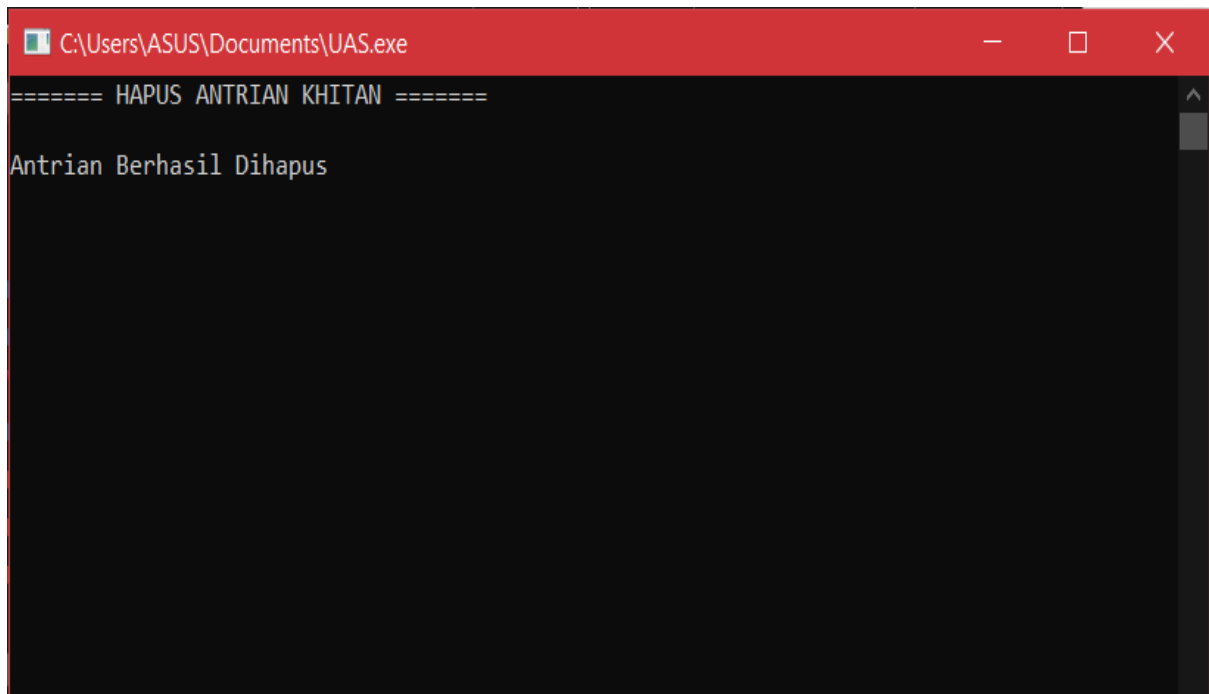
Data Peserta Khitan Berhasil Ditambahkan
```

Gambar 4. Panggil antrian pertama

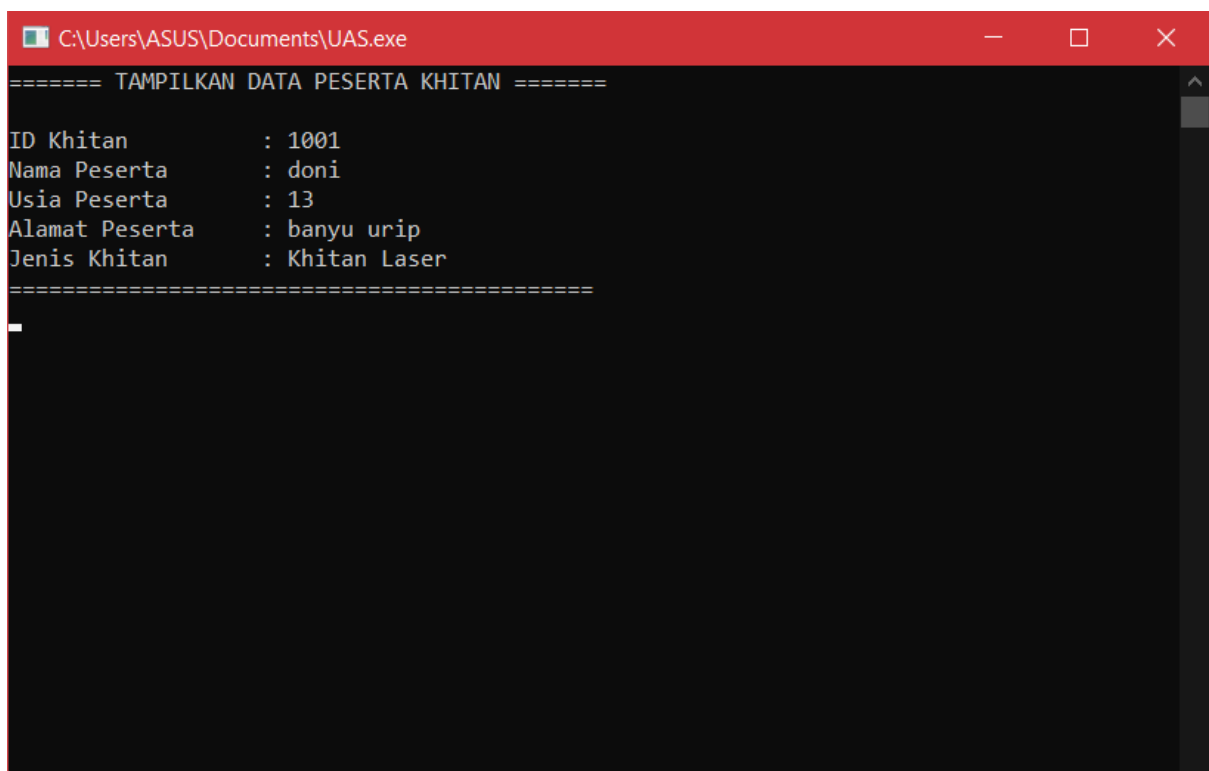
```
C:\Users\ASUS\Documents\UAS.exe
===== TAMPILKAN ANTRIAN KHITAN =====

2 -> yudi
```

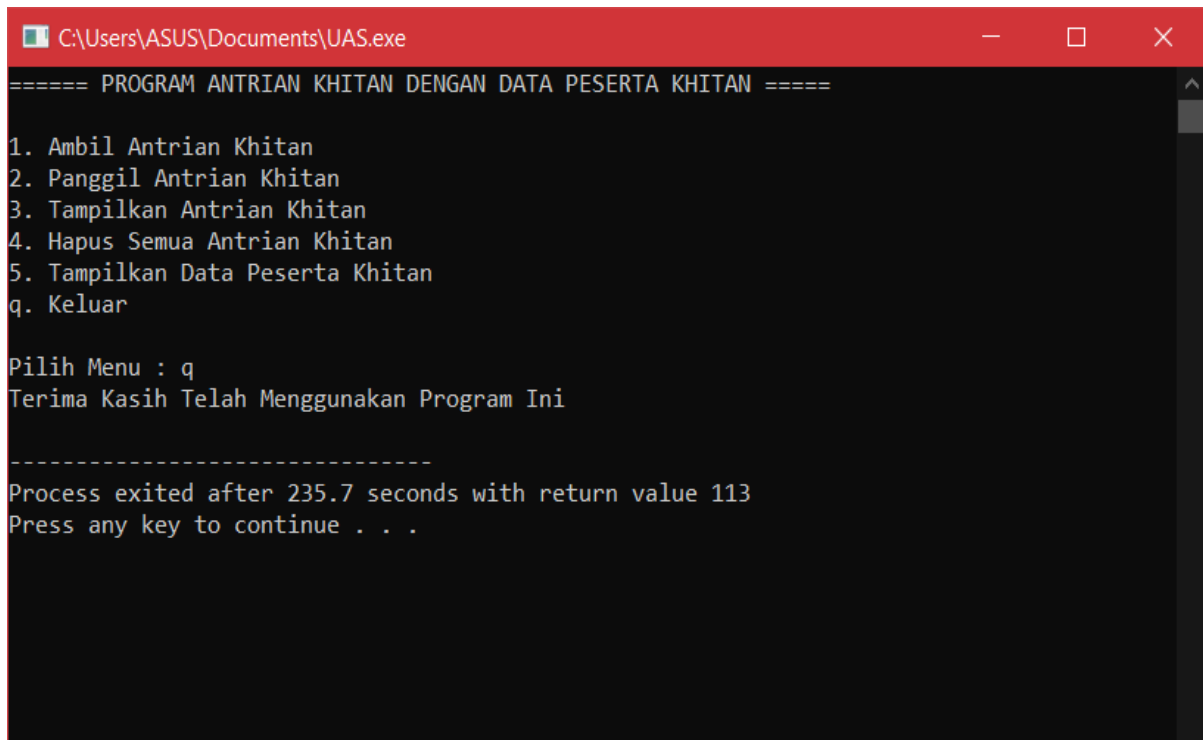
Gambar 5. Tampilan antrian yang belum dipanggil



Gambar 6. Hapus semua antrian tersisa



Gambar 7. Tampilan informasi antrian yang telah selesai



```
C:\Users\ASUS\Documents\UAS.exe
===== PROGRAM ANTRIAN KHITAN DENGAN DATA PESERTA KHITAN =====
1. Ambil Antrian Khitan
2. Panggil Antrian Khitan
3. Tampilkan Antrian Khitan
4. Hapus Semua Antrian Khitan
5. Tampilkan Data Peserta Khitan
q. Keluar

Pilih Menu : q
Terima Kasih Telah Menggunakan Program Ini

-----
Process exited after 235.7 seconds with return value 113
Press any key to continue . . .
```

Gambar 8. Keluar program