

## ASSIGNMENT MODUL 8 STRUKTUR DATA

## 1. Queue.h

```

Queue.h x main.cpp x Queue.cpp x
1  #ifndef QUEUE_H_INCLUDED
2  #define QUEUE_H_INCLUDED
3
4  #include <iostream>
5
6  using namespace std;
7
8  struct Infotype {
9      string nama;
10     int usia;
11     string pekerjaan;
12     bool prioritas;
13     int nomor_antrean;
14     bool kondisi_darurat;
15 };
16
17 typedef struct ElemQ *Address;
18
19 struct ElemQ {
20     Infotype info;
21     Address next;
22 };
23
24 struct Queue{
25     Address head;
26     Address tail;
27 };
28
29 void createQueue_103032330095(Queue &Q);
30
31 bool isEmpty_103032330095(Queue Q);
32
33 Address createElemQueue_103032330095(string nama, int usia, string pekerjaan, int nomor_antrean);
34
35 void enqueue_103032330095(Queue &Q, Address P);
36

```

```

Queue.h x main.cpp x Queue.cpp x
36
37 void dequeue_103032330095(Queue &Q, Address &P);
38
39 Address front_103032330095(Queue Q);
40
41 Address back_103032330095(Queue Q);
42
43 int size_103032330095(Queue Q);
44
45 void printInfo_103032330095(Queue Q);
46
47 void serveQueue_103032330095(Queue &Q);
48
49 void reassignQueue_103032330095(Queue &Q);
50
51 void checkWaitingTime_103032330095(Queue &Q, int waktu_sekarang);
52
53 void emergencyHandle_103032330095(Queue &Q, int nomor_antrean);
54
55 void updatePriority_103032330095(Queue &Q);
56
57 Address findAndRemove_103032330095(Queue &Q, int nomor_antrean);
58
59 #endif // QUEUE_H_INCLUDED
60

```

## 2. Queue.cpp

```

Queue.h x main.cpp x Queue.cpp x
1  #include "Queue.h"
2
3  void createQueue_103032330095(Queue &Q){
4      Q.head = NULL;
5      Q.tail = NULL;
6  }
7
8  bool isEmpty_103032330095(Queue Q){
9      return Q.head == NULL;
10 }
11
12 Address createElemQueue_103032330095(string nama, int usia, string pekerjaan, int nomor_antrean){
13     Address P = new ElemQ;
14     P->info.nama = nama;
15     P->info.usia = usia;
16     P->info.pekerjaan = pekerjaan;
17     P->info.prioritas = (usia >= 60 || pekerjaan == "tenaga kesehatan");
18     P->info.nomor_antrean = nomor_antrean;
19     P->info.kondisi_darurat = false;
20     P->next = NULL;
21     return P;
22 }
23
24 void enqueue_103032330095(Queue &Q, Address P){
25     if (isEmpty_103032330095(Q)) {
26         Q.head = P;
27         Q.tail = P;
28     } else if (P->info.prioritas) {
29         if (!Q.head->info.prioritas) {
30             P->next = Q.head;
31             Q.head = P;
32         } else {

```

```

Queue.h x main.cpp x Queue.cpp x
31         Q.head = P;
32     } else {
33         Address temp = Q.head;
34         while (temp->next != NULL && temp->next->info.prioritas) {
35             temp = temp->next;
36         }
37         P->next = temp->next;
38         temp->next = P;
39         if (P->next == NULL) {
40             Q.tail = P;
41         }
42     }
43     } else {
44         Q.tail->next = P;
45         Q.tail = P;
46     }
47 }
48
49 void dequeue_103032330095(Queue &Q, Address &P){
50     if (isEmpty_103032330095(Q)) {
51         P = NULL;
52         cout << "Semua warga telah terlayani." << endl;
53     } else {
54         P = Q.head;
55         Q.head = Q.head->next;
56         if (Q.head == NULL){
57             Q.tail = NULL;
58         }
59         P->next = NULL;
60     }
61 }
62

```

```

Queue.h x main.cpp x Queue.cpp x
61 }
62
63 Address front_103032330095(Queue Q) {
64     return Q.head;
65 }
66
67 Address back_103032330095(Queue Q) {
68     return Q.tail;
69 }
70
71 int size_103032330095(Queue Q) {
72     int count = 0;
73     Address temp = Q.head;
74     while (temp != NULL) {
75         count++;
76         temp = temp->next;
77     }
78     return count;
79 }
80
81 void printInfo_103032330095(Queue Q) {
82     Address P;
83     if (isEmpty_103032330095(Q)) {
84         cout << "Antrean Kosong." << endl;
85     } else {
86         cout << "Daftar Antrean : " << endl;
87         P = Q.head;
88         while (P != NULL) {
89             cout << "Nama : " << P->info.nama << endl;
90             cout << "Usia : " << P->info.usia << endl;
91             cout << "Pekerjaan : " << P->info.pekerjaan << endl;
92             if (P->info.prioritas) {

```

```

Queue.h x main.cpp x Queue.cpp x
91     cout << "Pekerjaan : " << P->info.pekerjaan << endl;
92     if (P->info.prioritas) {
93         cout << "Prioritas : Ya" << endl;
94     } else {
95         cout << "Prioritas : Tidak" << endl;
96     }
97     cout << "Nomor Antrean : " << P->info.nomor_antrean << endl;
98     cout << "-----" << endl;
99     P = P->next;
100 }
101 }
102 }
103
104 void serveQueue_103032330095(Queue &Q) {
105     int maks = 0;
106     Address P;
107     if (isEmpty_103032330095(Q)) {
108         cout << "Antrean kosong" << endl;
109     } else {
110         while (!isEmpty_103032330095(Q) && maks < 100) {
111             dequeue_103032330095(Q, P);
112             cout << "Melayani warga : " << endl;
113             cout << "Nama : " << P->info.nama << endl;
114             cout << "Usia : " << P->info.usia << endl;
115             cout << "Pekerjaan : " << P->info.pekerjaan << endl;
116             if (P->info.prioritas) {
117                 cout << "Prioritas : Ya" << endl;
118             } else {
119                 cout << "Prioritas : Tidak" << endl;
120             }
121             cout << "Vaksinasi berhasil." << endl;
122             cout << "-----" << endl;

```

```

Queue.h x main.cpp x Queue.cpp x
121         cout << "Vaksinasi berhasil." << endl;
122         cout << "-----" << endl;
123         maks++;
124     }
125     if (maks == 100) {
126         cout << "Kapasitas pelayanan harian telah penuh!" << endl;
127     }
128     if (!isEmpty_103032330095(Q)){
129         cout << "Bagi warga yang belum terlayani, harap hadir kembali besok." << endl;
130     }
131 }
132 }
133
134 void reassignQueue_103032330095(Queue &Q){
135     Queue R;
136     int sizeQueue, i;
137     Address P;
138     createQueue_103032330095(R);
139     sizeQueue = size_103032330095(Q);
140     for (i = 1; i <= sizeQueue; i++){
141         dequeue_103032330095(Q, P);
142         enqueue_103032330095(R, P);
143     }
144     Q = R;
145 }
146
147 void checkWaitingTime_103032330095(Queue &Q, int waktu_sekarang){
148     Address P;
149     if (waktu_sekarang > 120) {
150         P = Q.head;
151         while (P != NULL){
152             P->info.prioritas = true;
153             P = P->next;

```

```

Queue.h x main.cpp x Queue.cpp x
151         while (P != NULL){
152             P->info.prioritas = true;
153             P = P->next;
154         }
155     }
156     reassignQueue_103032330095(Q);
157 }
158
159 void emergencyHandle_103032330095(Queue &Q, int nomor_antrean){
160     Address P;
161     P = findAndRemove_103032330095(Q, nomor_antrean);
162     if (P != NULL) {
163         P->info.kondisi_darurat = true;
164         P->info.prioritas = true;
165         P->next = Q.head;
166         Q.head = P;
167     }
168 }
169
170 void updatePriority_103032330095(Queue &Q){
171     Address P, R;
172     checkWaitingTime_103032330095(Q, 130);
173     P = Q.head;
174     while (P != NULL){
175         if (P->info.kondisi_darurat){
176             R = P->next;
177             emergencyHandle_103032330095(Q, P->info.nomor_antrean);
178             P = R;
179         } else {
180             P = P->next;
181         }
182     }
183 }

```

```
Queue.h X main.cpp X Queue.cpp X
1 #include "Queue.h"
2
3 int main()
4 {
5     // Membuat antrean
6     Queue Q;
7     createQueue_103032330095(Q);
8     Address P1, P2, P3, P4, P5;
9
10    // Menambahkan beberapa elemen ke dalam queue
11    P1 = createElemQueue_103032330095("John Doe", 65, "lansia", 1);
12    P2 = createElemQueue_103032330095("Alice", 30, "tenaga kesehatan", 2);
13    P3 = createElemQueue_103032330095("Bob", 25, "pekerja", 3);
14    P4 = createElemQueue_103032330095("Charlie", 70, "pensiunan", 4);
15    P5 = createElemQueue_103032330095("David", 28, "pekerja", 5);
16
17    enqueue_103032330095(Q, P1);
18    enqueue_103032330095(Q, P2);
19    enqueue_103032330095(Q, P3);
20    enqueue_103032330095(Q, P4);
21    enqueue_103032330095(Q, P5);
22
23    // Menampilkan isi queue
24    cout << "Isi antrean awal:" << endl;
25    printInfo_103032330095(Q);
26
27    // Melayani antrean
28    cout << "\nMelakukan pelayanan pada antrean:" << endl;
29    serveQueue_103032330095(Q);
30
31    // Memeriksa antrean setelah pelayanan
32    cout << "\nIsi antrean setelah pelayanan:" << endl;
33    printInfo_103032330095(Q);
34
35    // Menambahkan elemen baru untuk simulasi pengaturan ulang prioritas
```

```
Queue.h x main.cpp x Queue.cpp x
35 // Menambahkan elemen baru untuk simulasi pengaturan ulang prioritas
36 Address P6 = createElemQueue_103032330095("Edward", 22, "pekerja", 6);
37 enqueue_103032330095(Q, P6);
38
39 // Simulasi reassignQueue untuk mengatur ulang antrian ke prioritas
40 cout << "\nMengatur ulang antrian berdasarkan prioritas:" << endl;
41 reassignQueue_103032330095(Q);
42 printInfo_103032330095(Q);
43
44 // Simulasi kondisi warga yang menunggu lebih dari 2 jam
45 cout << "\nMemeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:" << endl;
46 checkWaitingTime_103032330095(Q, 130); // Asumsikan waktu sekarang 130 menit dari nomor antrian pertama
47 printInfo_103032330095(Q);
48
49 // Menangani kondisi darurat untuk seorang warga
50 cout << "\nMenangani kondisi darurat untuk warga dengan nomor antrian 5:" << endl;
51 emergencyHandle_103032330095(Q, 5);
52 printInfo_103032330095(Q);
53
54 // Update prioritas warga setiap jam
55 cout << "\nMengupdate prioritas antrian setiap jam:" << endl;
56 updatePriority_103032330095(Q);
57 printInfo_103032330095(Q);
58
59
60 // Mencari dan menghapus warga dengan nomor antrian tertentu
61 cout << "\nMenghapus warga dengan nomor antrian 3:" << endl;
62 Address removedElem = findAndRemove_103032330095(Q, 3);
63 if (removedElem) {
64     cout << "Warga yang dihapus: " << removedElem->info.nama << endl;
65 }
66 printInfo_103032330095(Q);
67
68 // Mengecek ukuran queue
69 cout << "\nUkuran antrian saat ini: " << size_103032330095(Q) << endl;
70
71 return 0;
72 }
```

#### 4. Output

```
"D:\KuliahGena\SMT 3\Strukt" × + v

Isi antrean awal:
Daftar Antrean :
Nama : John Doe
Usia : 65
Pekerjaan : lansia
Prioritas : Ya
Nomor Antrean : 1
-----
Nama : Alice
Usia : 30
Pekerjaan : tenaga kesehatan
Prioritas : Ya
Nomor Antrean : 2
-----
Nama : Charlie
Usia : 70
Pekerjaan : pensiunan
Prioritas : Ya
Nomor Antrean : 4
-----
Nama : Bob
Usia : 25
Pekerjaan : pekerja
Prioritas : Tidak
Nomor Antrean : 3
-----
Nama : David
Usia : 28
Pekerjaan : pekerja
Prioritas : Tidak
Nomor Antrean : 5
-----
```

```
"D:\KuliahGena\SMT 3\Struktur" X + v

Melakukan pelayanan pada antrean:
Melayani warga :
Nama : John Doe
Usia : 65
Pekerjaan : lansia
Prioritas : Ya
Vaksinasi berhasil.
-----
Melayani warga :
Nama : Alice
Usia : 30
Pekerjaan : tenaga kesehatan
Prioritas : Ya
Vaksinasi berhasil.
-----
Melayani warga :
Nama : Charlie
Usia : 70
Pekerjaan : pensiunan
Prioritas : Ya
Vaksinasi berhasil.
-----
Melayani warga :
Nama : Bob
Usia : 25
Pekerjaan : pekerja
Prioritas : Tidak
Vaksinasi berhasil.
-----
Melayani warga :
Nama : David
Usia : 28
Pekerjaan : pekerja
Prioritas : Tidak
Vaksinasi berhasil.
-----

Isi antrean setelah pelayanan:
Antrean Kosong.
```



```
"D:\KuliahGena\SMT 3\Strukt  ×  +  v

Mengatur ulang antrean berdasarkan prioritas:
Daftar Antrean :
Nama : Edward
Usia : 22
Pekerjaan : pekerja
Prioritas : Tidak
Nomor Antrean : 6
-----

Memeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:
Daftar Antrean :
Nama : Edward
Usia : 22
Pekerjaan : pekerja
Prioritas : Ya
Nomor Antrean : 6
-----

Menangani kondisi darurat untuk warga dengan nomor antrean 5:
Warga dengan nomor antrean 5 tidak ditemukan dalam antrean
Daftar Antrean :
Nama : Edward
Usia : 22
Pekerjaan : pekerja
Prioritas : Ya
Nomor Antrean : 6
-----

Mengupdate prioritas antrean setiap jam:
Daftar Antrean :
Nama : Edward
Usia : 22
Pekerjaan : pekerja
Prioritas : Ya
Nomor Antrean : 6
-----

Menghapus warga dengan nomor antrean 3:
Warga dengan nomor antrean 3 tidak ditemukan dalam antrean
Daftar Antrean :
Nama : Edward
Usia : 22
Pekerjaan : pekerja
Prioritas : Ya
Nomor Antrean : 6
-----

Ukuran antrean saat ini: 1

Process returned 0 (0x0)    execution time : 0.157 s
Press any key to continue.
```