

## TP MODUL 8

Nama : Ahmad Naufal Ramadhan  
NIM : 103012300239  
Kelas : IF-47-02  
Kode Asprak : YDA

### Header.h

```
1  #ifndef HEADER_H_INCLUDED
2  #define HEADER_H_INCLUDED
3
4  #include <iostream>
5  using namespace std;
6
7  #define head(Q) Q.head
8  #define tail(Q) Q.tail
9  #define next(P) P->next
10 #define info(P) P->info
11
12 struct infotype{
13     string nama;
14     int usia;
15     string pekerjaan;
16     bool prioritas;
17     int nomor_antrean;
18     bool kondisi_darurat;
19     int waktu_daftar;
20 };
21
22 struct ElemQ{
23     infotype info;
24     ElemQ *next;
25 };
26
27 struct Queue{
28     ElemQ *head;
29     ElemQ *tail;
30 };
31
32 void createQueue_103012300239(Queue &Q);
33 bool isEmpty_103012300239(Queue Q);
34 ElemQ* createElemQueue_103012300239(string nama, int usia, string pekerjaan, int nomor_antrean, int waktu_daftar);
35 void enqueue_103012300239(Queue &Q, ElemQ *P);
36 void dequeue_103012300239(Queue &Q, ElemQ *&P);
37 ElemQ* front_103012300239(Queue Q);
38 ElemQ* back_103012300239(Queue Q);
39 int size_103012300239(Queue Q);
40 void printInfo_103012300239(Queue Q);
41 void serveQueue_103012300239(Queue &Q);
42 void reassignQueue_103012300239(Queue &Q);
43 void checkWaitingTime_103012300239(Queue &Q, int waktu_sekarang);
44 void emergencyHandle_103012300239(Queue &Q, int nomor_antrean);
45 void updatePriority_103012300239(Queue &Q);
46 ElemQ* findAndRemove_103012300239(Queue Q, int nomor_antrean);
47
48
49 #endif // HEADER_H_INCLUDED
50
```

## Header.cpp

```
1  #include "header.h"
2  #include <iostream>
3  using namespace std;
4
5  void createQueue_103012300239(Queue &Q) {
6      head(Q) = NULL;
7      tail(Q) = NULL;
8  }
9
10 bool isEmpty_103012300239(Queue Q) {
11     return head(Q) == NULL;
12 }
13
14 ElemQ* createElemQueue_103012300239(string nama, int usia, string pekerjaan, int nomor_antrean, int waktu_daftar) {
15     ElemQ *P = new ElemQ;
16     info(P).nama = nama;
17     info(P).usia = usia;
18     info(P).pekerjaan = pekerjaan;
19     info(P).prioritas = (usia >= 60 || pekerjaan == "tenaga kesehatan");
20     info(P).nomor_antrean = nomor_antrean;
21     info(P).kondisi_darurat = false;
22 }
23
24 void enqueue_103012300239(Queue &Q, ElemQ *P) {
25     if(isEmpty_103012300239(Q)) {
26         head(Q) = P;
27         tail(Q) = P;
28     } else if(info(P).prioritas) {
29         if (!info(head(Q)).prioritas) {
30             next(P) = head(Q);
31             head(Q) = P;
32         } else {
33             ElemQ *temp = head(Q);
34             while (next(temp) != NULL && info(next(temp)).prioritas) {
35                 temp = next(temp);
36             }
37             next(P) = next(temp);
38             next(temp) = P;
39             if(next(P) == NULL) {
40                 tail(Q) = P;
41             }
42         }
43     } else {
44         next(tail(Q)) = P;
45         tail(Q) = P;
46     }
47 }
48
49 void dequeue_103012300239(Queue &Q, ElemQ *&P) {
50     if(isEmpty_103012300239(Q)) {
51         P = NULL;
52         cout << "Semua warga telah terlayani" << endl;
```

```

53     } else {
54         P = head(Q);
55         head(Q) = next(head(Q));
56         if (head(Q) == NULL) {
57             tail(Q) = NULL;
58         }
59         next(P) = NULL;
60     }
61 }
62
63 ElemQ* front_103012300239(Queue Q) {
64     return head(Q);
65 }
66
67 ElemQ* back_103012300239(Queue Q) {
68     return tail(Q);
69 }
70
71 int size_103012300239(Queue Q) {
72     int count = 0;
73     ElemQ *temp = head(Q);
74     while (temp != NULL) {
75         count++;
76         temp = next(temp);
77     }
78     return count;
79 }
80
81 void printInfo_103012300239(Queue Q) {
82     if (isEmpty_103012300239(Q)) {
83         cout << "Antrean KOSONG!!!" << endl;
84     } else {
85         cout << "Daftar Antrean:" << endl;
86         ElemQ *P = head(Q);
87         while (P != NULL) {
88             cout << "Nama: " << info(P).nama << endl;
89             cout << "Usia: " << info(P).usia << endl;
90             if (info(P).prioritas == true) {
91                 cout << "prioritas: Ya" << endl;
92             } else {
93                 cout << "prioritas: Tidak" << endl;
94             }
95             cout << "Nomor Antrean: " << info(P).nomor_antrean << endl;
96             cout << "-----" << endl;
97             P = next(P);
98         }
99     }
100 }
101
102 void serveQueue_103012300239(Queue &Q) {
103     int count = 0;
104     ElemQ *P;

```

```

104 ElemQ *P;
105 while(isEmpty_103012300239(Q) != true && count <= 100){
106     dequeue_103012300239(Q, P);
107     cout << "Melayani warga:" << endl;
108     cout << "Nama: " << info(P).nama << endl;
109     cout << "Usia: " << info(P).usia << endl;
110     cout << "Pekerjaan: " << info(P).pekerjaan << endl;
111     if(info(P).prioritas == true){
112         cout << "Prioritas: Ya" << endl;
113     }else{
114         cout << "Prioritas: Tidak" << endl;
115     }
116     cout << "Vaksinasi berhasil." << endl;
117     cout << "-----" << endl;
118     count++;
119 }
120
121 if(size_103012300239(Q) > 100){
122     cout << "Kapasitas harian telah penuh." << endl;
123     if(isEmpty_103012300239(Q) == false){
124         cout << "Warga yang belum terlayani diminta kembali besok"
125     }
126 }
127 }
128
129 void reassignQueue_103012300239(Queue &Q){
130     Queue priorityQ, normalQ, waitingQ;
131     createQueue_103012300239(priorityQ);
132     createQueue_103012300239(normalQ);
133     createQueue_103012300239(waitingQ);
134     ElemQ *P;
135     while(isEmpty_103012300239(Q) != true){ // Mengeluarkan isi queue
136         dequeue_103012300239(Q, P);
137         if(info(P).prioritas){
138             enqueue_103012300239(priorityQ, P);
139         }else{
140             enqueue_103012300239(normalQ, P);
141         }
142     }
143     while(isEmpty_103012300239(priorityQ) != true){ // Memasukkan que
144         dequeue_103012300239(priorityQ, P);
145         enqueue_103012300239(waitingQ, P);
146     }
147     while(isEmpty_103012300239(normalQ) != true){ // Memasukkan queue
148         dequeue_103012300239(normalQ, P);
149         enqueue_103012300239(waitingQ, P);
150     }
151     Q = waitingQ;
152 }
153
154 void checkWaitingTime_103012300239(Queue &Q, int waktu_sekarang){
155     Queue priorityQ, tempQ;

```

```

156 createQueue_103012300239(priorityQ);
157 createQueue_103012300239(tempQ);
158 ElemQ *P;
159 while(isEmpty_103012300239(Q) != true){
160     dequeue_103012300239(Q, P);
161     if((waktu_sekarang - info(P).waktu_daftar) > 120){
162         info(P).prioritas = true;
163         enqueue_103012300239(priorityQ, P);
164     }else{
165         enqueue_103012300239(tempQ, P);
166     }
167 }
168 while(isEmpty_103012300239(priorityQ) != true){ // Memasukkan queue priorityQ ke Q
169     dequeue_103012300239(priorityQ, P);
170     enqueue_103012300239(Q, P);
171 }
172 while(isEmpty_103012300239(tempQ) != true){ // Memasukkan queue tempQ ke Q
173     dequeue_103012300239(tempQ, P);
174     enqueue_103012300239(Q, P);
175 }
176 }
177
178 void emergencyHandle_103012300239(Queue &Q, int nomor_antrean){
179     Queue priorityQ, tempQ;
180     createQueue_103012300239(priorityQ);
181     createQueue_103012300239(tempQ);
182     ElemQ *P;
183     bool isFound = false;
184     while(isEmpty_103012300239(Q) != true){
185         dequeue_103012300239(Q, P);
186         if(info(P).nomor_antrean == nomor_antrean){
187             info(P).kondisi_darurat = true;
188             isFound = true;
189             enqueue_103012300239(priorityQ, P);
190         }else{
191             enqueue_103012300239(tempQ, P);
192         }
193     }
194     if(isFound == false){
195         cout << "Warga dengan nomor antrean " << nomor_antrean << " tidak ditemukan." << endl;
196     }
197     while(isEmpty_103012300239(priorityQ) != true){ // Memasukkan queue priorityQ ke Q
198         dequeue_103012300239(priorityQ, P);
199         enqueue_103012300239(Q, P);
200     }
201     while(isEmpty_103012300239(tempQ) != true){ // Memasukkan queue tempQ ke Q
202         dequeue_103012300239(tempQ, P);
203         enqueue_103012300239(Q, P);
204     }
205 }

```

```

207 void updatePriority_103012300239(Queue &Q){
208     int waktu;
209     ElemQ *P;
210     Queue tmp, tmpdarurat, tmpprio;
211     createQueue_103012300239(tmpdarurat);
212     createQueue_103012300239(tmp);
213     createQueue_103012300239(tmpprio);
214     while (head(Q) != NULL){
215         dequeue_103012300239(Q, P);
216         if (info(P).kondisi_darurat){
217             enqueue_103012300239(tmpdarurat, P);
218         } else if (info(P).prioritas || info(P).waktu_daftar > 120){
219             enqueue_103012300239(tmpprio, P);
220         } else {
221             enqueue_103012300239(tmpdarurat, P);
222         }
223     }
224     while (head(tmpdarurat) != NULL){
225         dequeue_103012300239(tmpdarurat, P);
226         enqueue_103012300239(Q, P);
227     }
228     while (head(tmpprio) != NULL){
229         dequeue_103012300239(tmpprio, P);
230         enqueue_103012300239(Q, P);
231     }
232     while (head(tmp) != NULL){
233         dequeue_103012300239(tmp, P);
234         enqueue_103012300239(Q, P);
235     }
236 }
237
238 ElemQ* findAndRemove_103012300239(Queue Q, int nomor_antrean){
239     Queue tempQ;
240     createQueue_103012300239(tempQ);
241     ElemQ *P;
242     ElemQ *Pketemu = NULL;
243     while (isEmpty_103012300239(Q) != true){
244         dequeue_103012300239(Q, P);
245         if (info(P).nomor_antrean == nomor_antrean){
246             Pketemu = P;
247         } else {
248             enqueue_103012300239(tempQ, P);
249         }
250     }
251     while (isEmpty_103012300239(tempQ) != true){ // Memasukkan queue tempQ ke Q
252         dequeue_103012300239(tempQ, P);
253         enqueue_103012300239(Q, P);
254     }
255     if (Pketemu == NULL){
256         cout << "Warga dengan nomor antrean " << nomor_antrean << " tidak ditemukan dalam antrean.";
257     }
258     return Pketemu;
259 }

```

## Main.cpp

```
1 | #include "header.h"
2 | #include <iostream>
3 |
4 | using namespace std;
5 |
6 | int main() {
7 |     Queue Q;
8 |     createQueue_103012300239(Q);
9 |
10 |    ElemQ* P1 = createElemQueue_103012300239("John Doe", 65, "lansia", 1, 0);
11 |    ElemQ* P2 = createElemQueue_103012300239("Alice", 30, "tenaga kesehatan", 2, 1);
12 |    ElemQ* P3 = createElemQueue_103012300239("Bob", 25, "pekerja", 3, 4);
13 |    ElemQ* P4 = createElemQueue_103012300239("Charlie", 70, "pensiunan", 4, 4);
14 |    ElemQ* P5 = createElemQueue_103012300239("David", 28, "pekerja", 5, 7);
15 |
16 |    enqueue_103012300239(Q, P1);
17 |    enqueue_103012300239(Q, P2);
18 |    enqueue_103012300239(Q, P3);
19 |    enqueue_103012300239(Q, P4);
20 |    enqueue_103012300239(Q, P5);
21 |
22 |    cout << "Isi antrean awal:" << endl;
23 |    printInfo_103012300239(Q);
24 |
25 |    cout << "\nMelakukan pelayanan pada antrean:" << endl;
26 |    serveQueue_103012300239(Q);
27 |
28 |    cout << "\nIsi antrean setelah pelayanan:" << endl;
29 |    printInfo_103012300239(Q);
30 |
31 |    ElemQ* P6 = createElemQueue_103012300239("Edward", 22, "pekerja", 6, 9);
32 |    enqueue_103012300239(Q, P6);
33 |
34 |    cout << "\nMengatur ulang antrean berdasarkan prioritas:" << endl;
35 |    reassignQueue_103012300239(Q);
36 |    printInfo_103012300239(Q);
37 |
38 |    cout << "\nMemeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:" << endl;
39 |    checkWaitingTime_103012300239(Q, 130);
40 |
41 |    printInfo_103012300239(Q);
42 |
43 |    cout << "\nMenangani kondisi darurat untuk warga dengan nomor antrean 5:" << endl;
44 |    emergencyHandle_103012300239(Q, 5);
45 |    printInfo_103012300239(Q);
46 |
47 |    cout << "\nMengupdate prioritas antrean setiap jam:" << endl;
48 |    updatePriority_103012300239(Q);
49 |    printInfo_103012300239(Q);
50 |
51 |    cout << "\nMenghapus warga dengan nomor antrean 3:" << endl;
52 |    ElemQ* removedElem = findAndRemove_103012300239(Q, 3);
53 |
54 |    if (removedElem) {
55 |
56 |        if (removedElem) {
57 |            cout << "Warga yang dihapus: " << info(removedElem).nama << endl;
58 |        }
59 |
60 |        printInfo_103012300239(Q);
61 |
62 |        cout << "\nUkuran antrean saat ini: " << size_103012300239(Q) << endl;
63 |
64 |        return 0;
65 |    }
66 | }
```

## Running

```
Isi antrean awal:
Daftar Antrean:
Nama: John Doe
Usia: 65
prioritas: Ya
Nomor Antrean: 1
-----
Nama: Charlie
Usia: 70
prioritas: Ya
Nomor Antrean: 4
-----
Nama: Bob
Usia: 25
prioritas: Tidak
Nomor Antrean: 3
-----
Nama: David
Usia: 28
prioritas: Tidak
Nomor Antrean: 5
-----

Melakukan pelayanan pada antrean:
Melayani warga:
Nama: John Doe
Usia: 65
Pekerjaan: lansia
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!!!

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

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

Menangani kondisi darurat untuk warga dengan nomor antrean 5:
Warga dengan nomor antrean 5 tidak ditemukan.
Daftar Antrean:
Nama: Edward
Usia: 22
prioritas: Ya
Nomor Antrean: 6
-----

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

Ukuran antrean saat ini: 1

Process returned 0 (0x0)   execution time : 0.095 s
Press any key to continue.
|
```