

Nama: Sheren Aulia Azahra

NIM: 103032400036

Kelas: IT-48-05

File queue.h

```
main.cpp X | queue.h X | queue.cpp X
1 #ifndef QUEUE_H_INCLUDED
2 #define QUEUE_H_INCLUDED
3
4 #include <iostream>
5 using namespace std;
6
7 typedef struct elmqueue *address;
8 struct infotype{
9     string nama;
10    int usia;
11    string pekerjaan;
12    bool prioritas;
13    int nomorAntrean;
14    int waktuDaftar;
15 } ;
16
17 struct Queue{
18     address head;
19     address tail;
20 };
21
22 struct elmqueue{
23     infotype info;
24     address next;
25 };
26
27 void createQueue_103032400036(Queue &Q);
28 bool isEmpty_103032400036(Queue Q);
29 address allocate_103032400036(string nama,int usia,string pekerjaan,int nomorAntrean, int waktuDaftar);
30 address front_103032400036(Queue Q);
31 address back_103032400036(Queue Q);
32 void enqueue_103032400036(Queue &Q, address p);
33 void dequeue_103032400036(Queue &Q, address &p);
34 int size_103032400036(Queue Q);
35 void printPatient_103032400036(address p);
36 void serveQueue_103032400036(Queue &Q);
37 void reassignQueue_103032400036(Queue &Q);
38 void emergencyHandle_103032400036(Queue &Q, int nomorAntrian);
39 #endif // QUEUE_H_INCLUDED
40
```

File queue.cpp

```
1 #include <iostream>
2 #include "queue.h"
3
4 using namespace std;
5
6 void createQueue_103032400036(Queue &Q) {
7     Q.head = nullptr;
8     Q.tail = nullptr;
9 }
10
11 bool isEmpty_103032400036(Queue Q) {
12     if (Q.head == nullptr && Q.tail == nullptr){
13         return true;
14     } else {
15         return false;
16     }
17 }
18
19 address allocate_103032400036(string nama, int usia, string pekerjaan, int nomorAntrean, int waktuDaftar) {
20     address p;
21     infotype x;
22
23     p = new elmqueue;
24     x.nama = nama;
25     x.usia = usia;
26     x.pekerjaan = pekerjaan;
27     x.prioritas = usia >= 60 || pekerjaan == "tenaga_kesehatan";
28     x.nomorAntrean = nomorAntrean;
29     x.waktuDaftar = waktuDaftar;
30
31     p->info = x;
32     p->next = nullptr;
33     return p;
34 }
35
36 address front_103032400036(Queue Q) {
37     return Q.head;
38 }
39
40 address back_103032400036(Queue Q) {
41     return Q.tail;
42 }
43
```

```

44 void enqueue_103032400036(Queue &Q, address p) {
45     address temp;
46     bool inputPatientPrioritas;
47     bool semuaPasienPrioritas;
48     bool tidakAdaPasienPrioritas;
49
50     if (!isEmpty_103032400036(Q)) {
51         inputPatientPrioritas = !p->info.prioritas;
52         semuaPasienPrioritas = Q.tail->info.prioritas;
53         tidakAdaPasienPrioritas = !Q.head->info.prioritas;
54     }
55     if (isEmpty_103032400036(Q)) {
56         Q.head = p;
57         Q.tail = p;
58     } else if (inputPatientPrioritas || semuaPasienPrioritas) {
59         Q.tail->next = p;
60         Q.tail = p;
61     } else if (tidakAdaPasienPrioritas) {
62         p->next = Q.head;
63         Q.head = p;
64     } else {
65         temp = Q.head;
66         while (temp->next != nullptr && temp->next->info.prioritas) {
67             temp = temp->next;
68         }
69         p->next = temp->next;
70         temp->next = p;
71     }
72 }
73
74 void dequeue_103032400036(Queue &Q, address &p) {
75     if (!isEmpty_103032400036(Q)) {
76         p = Q.head;
77         Q.head = p->next;
78
79         if (Q.head == nullptr) {
80             Q.tail = nullptr;
81         }
82     } else {
83         p = nullptr;
84     }
85 }
86
87 int size_103032400036(Queue Q) {
88     int count;
89     address p;
90
91     count = 0;
92     p = Q.head;
93
94     while (p!=nullptr) {
95         count++;
96         p = p->next;
97     }
98     return count;
99 }
```

```

100
101 void printPatient_103032400036(address p) {
102     if (p != nullptr) {
103         cout << "\n-----" << endl;
104         cout << "nama: " << p->info.nama << endl;
105         cout << "usia: " << p->info.usia << endl;
106         cout << "pekerjaan: " << p->info.pekerjaan << endl;
107         cout << "prioritas: " << p->info.prioritas << endl;
108         cout << "nomor antrean: " << p->info.nomorAntrean << endl;
109     }
110 }
111
112 void serveQueue_103032400036(Queue &Q) {
113     address p;
114     int jumPasien = 5;
115     while (!isEmpty_103032400036(Q) && jumPasien > 0) {
116         dequeue_103032400036(Q, p);
117         printPatient_103032400036(p);
118         cout << "Vaksinasi berhasil.";
119
120         p->next = nullptr;
121         jumPasien--;
122     }
123 }
124
125 void reassignQueue_103032400036(Queue &Q) {
126     address P;
127
128     P = Q.head;
129     while (P != nullptr) {
130         if (P->info.prioritas == false) {
131             P->info.prioritas = true;
132         }
133         P = P->next;
134     }
135 }
136
137 void emergencyHandle_103032400036(Queue &Q, int nomorAntrean) {
138     address P = Q.head;
139     address Prev = nullptr;
140
141     while (P != nullptr && P->info.nomorAntrean != nomorAntrean) {
142         Prev = P;
143         P = P->next;
144     }
145     if (P != nullptr) {
146         P->info.prioritas = true;
147         if (P != Q.head) {
148             Prev->next = P->next;
149             if (P == Q.tail) {
150                 Q.tail = Prev;
151             }
152             P->next = Q.head;
153             Q.head = P;
154         }
155     }
156 }
157

```

File main.cpp

```
main.cpp X queue.h X queue.cpp X
1 #include <iostream>
2 #include "queue.h"
3
4 using namespace std;
5
6 int main() {
7     Queue Q;
8     address p;
9     string nama, pekerjaan;
10    int usia, nomor;
11    int i, jumlah;
12
13    createQueue_103032400036(Q);
14    i = 0;
15
16    cout << "masukkan banyak pasien yang ingin di input: ";
17    cin >> jumlah;
18    cout << endl;
19
20    for(i = 0; i < jumlah; i++) {
21        cout << "pasien ke- " << (i+1) << endl;
22        cout << "nama: ";
23        cin >> nama;
24        cout << "usia: ";
25        cin >> usia;
26        cout << "pekerjaan: ";
27        cin >> pekerjaan;
28        cout << "nomor antrean: ";
29        cin >> nomor;
30        cout << endl;
31        p = allocate_103032400036(nama, usia, pekerjaan, nomor, i);
32        enqueue_103032400036(Q, p);
33    }
34
35    cout << "banyak pasien dalam antrean: " << size_103032400036(Q) << endl;
36    serveQueue_103032400036(Q);
37    cout << endl;
38
39    int darurat;
40    cout << "\nmasukkan nomor antrean yang ingin di prioritaskan: ";
41    cin >> darurat;
42    emergencyHandle_103032400036(Q, darurat);
43    serveQueue_103032400036(Q);
44
45    reassignQueue_103032400036(Q);
46    cout << "\nndata prioritas pasien telah di update." << endl;
47    serveQueue_103032400036(Q);
48
49    return 0;
50 }
```

Output

```
C:\Users\shere\OneDrive\TELKOM UNIVERSITY\SEMESTER 3\STRUKTUR DATA\MODUL 8\QUE
masukkan banyak pasien yang ingin di input: 11

pasien ke- 1
nama: andi
usia: 25
pekerjaan: guru
nomor antrean: 1

pasien ke- 2
nama: bela
usia: 42
pekerjaan: tenaga_kesehatan
nomor antrean: 2

pasien ke- 3
nama: citra
usia: 33
pekerjaan: mahasiswa
nomor antrean: 3

pasien ke- 4
nama: doni
usia: 45
pekerjaan: petani
nomor antrean: 4

pasien ke- 5
nama: eka
usia: 70
pekerjaan: pensiunan
nomor antrean: 5

pasien ke- 6
nama: faisal
usia: 29
pekerjaan: karyawan_swasta
nomor antrean: 6

pasien ke- 7
nama: gina
usia: 54
pekerjaan: pedagang
nomor antrean: 7

pasien ke- 8
nama: heri
usia: 66
pekerjaan: tenaga_kesehatan
nomor antrean: 8
```

```
pasien ke- 9  
nama: indah  
usia: 31  
pekerjaan: wirausaha  
nomor antrean: 9
```

```
pasien ke- 10  
nama: joni  
usia: 40  
pekerjaan: supir  
nomor antrean: 10
```

```
pasien ke- 11  
nama: kiki  
usia: 58  
pekerjaan: ibu_rumah_tangga  
nomor antrean: 11
```

```
banyak pasien dalam antrean: 11
```

```
-----  
nama: bela  
usia: 42  
pekerjaan: tenaga_kesehatan  
prioritas: 1  
nomor antrean: 2  
Vaksinasi berhasil.  
-----
```

```
nama: eka  
usia: 70  
pekerjaan: pensiunan  
prioritas: 1  
nomor antrean: 5  
Vaksinasi berhasil.  
-----
```

```
nama: heri  
usia: 66  
pekerjaan: tenaga_kesehatan  
prioritas: 1  
nomor antrean: 8  
Vaksinasi berhasil.  
-----
```

```
nama: andi  
usia: 25  
pekerjaan: guru  
prioritas: 0  
nomor antrean: 1  
Vaksinasi berhasil.  
-----
```

```
nama: citra  
usia: 33  
pekerjaan: mahasiswa  
prioritas: 0  
nomor antrean: 3  
Vaksinasi berhasil.
```

```
masukkan nomor antrean yang ingin di prioritaskan: 4
```

```
-----  
nama: doni  
usia: 45  
pekerjaan: petani  
prioritas: 1  
nomor antrean: 4  
Vaksinasi berhasil.
```

```
-----  
nama: faisal  
usia: 29  
pekerjaan: karyawan_swasta  
prioritas: 0  
nomor antrean: 6  
Vaksinasi berhasil.
```

```
-----  
nama: gina  
usia: 54  
pekerjaan: pedagang  
prioritas: 0  
nomor antrean: 7  
Vaksinasi berhasil.
```

```
-----  
nama: indah  
usia: 31  
pekerjaan: wirausaha  
prioritas: 0  
nomor antrean: 9  
Vaksinasi berhasil.
```

```
-----  
nama: joni  
usia: 40  
pekerjaan: supir  
prioritas: 0  
nomor antrean: 10  
Vaksinasi berhasil.
```

```
data prioritas pasien telah di update.  
-----  
nama: kiki  
usia: 58  
pekerjaan: ibu_rumah_tangga  
prioritas: 1  
nomor antrean: 11  
Vaksinasi berhasil.  
Process returned 0 (0x0)  execution time : 6.046 s  
Press any key to continue.
```

v