

## Main.cpp:

```
main.cpp X QUEUE.cpp X QUEUE.h X
1  #include "QUEUE.h"
2  int main()
3  {
4      Queue Q;
5      createQueue_103032330054(Q);
6
7      ElemQ* P1 = createElemQueue_103032330054("John Doe", 65, "lansia", 1);
8      ElemQ* P2 = createElemQueue_103032330054("Alice", 30, "tenaga kesehatan", 2);
9      ElemQ* P3 = createElemQueue_103032330054("Bob", 25, "pekerja", 3);
10     ElemQ* P4 = createElemQueue_103032330054("Charlie", 70, "pensiunan", 4);
11     ElemQ* P5 = createElemQueue_103032330054("David", 28, "pekerja", 5);
12
13     enqueue_103032330054(Q, P1);
14     enqueue_103032330054(Q, P2);
15     enqueue_103032330054(Q, P3);
16     enqueue_103032330054(Q, P4);
17     enqueue_103032330054(Q, P5);
18
19     cout << "Isi antrean awal:" << endl;
20     printInfo_103032330054(Q);
21
22     cout << "\nMelakukan pelayanan pada antrean:" << endl;
23     serveQueue_103032330054(Q);
24
25     cout << "\nIsi antrean setelah pelayanan:" << endl;
26     printInfo_103032330054(Q);
27
28     ElemQ* P6 = createElemQueue_103032330054("Edward", 22, "pekerja", 6);
29     enqueue_103032330054(Q, P6);
30
31     cout << "\nMengatur ulang antrean berdasarkan prioritas:" << endl;
32     reassignQueue_103032330054(Q);
33     printInfo_103032330054(Q);
34
35     cout << "\nMemeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:" << endl;
36     checkWaitingTime_103032330054(Q, 130);
37     printInfo_103032330054(Q);
38
39     cout << "\nMenangani kondisi darurat untuk warga dengan nomor antrean 5:" << endl;
40     emergencyHandle_103032330054(Q, 5);
41     printInfo_103032330054(Q);
42
43     cout << "\nMengupdate prioritas antrean setiap jam:" << endl;
44     updatePriority_103032330054(Q);
45     printInfo_103032330054(Q);
46
47     cout << "\nMenghapus warga dengan nomor antrean 3:" << endl;
48     ElemQ* removedElem = findAndRemove_103032330054(Q, 3);
49
50     if (removedElem != NULL) {
51         cout << "Warga yang dihapus: " << removedElem->info.nama << endl;
52     } else {
53         cout << "Warga dengan nomor antrean 3 tidak ditemukan" << endl;
54     }
55     printInfo_103032330054(Q);
56     cout << "\nUkuran antrean saat ini: " << size_103032330054(Q) << endl;
57     return 0;
58 }
59
```

## Queue.cpp:

```
main.cpp x QUEUE.cpp x QUEUE.h x
1  #include "QUEUE.h"
2  void createQueue_103032330054(Queue &Q){
3      Q.head = NULL;
4      Q.tail = NULL;
5  }
6  bool isEmpty_103032330054(Queue Q){
7      return Q.head == NULL;
8  }
9  ElemQ* createElemQueue_103032330054(string nama, int usia, string pekerjaan, int nomor_antrean){
10     ElemQ *P = new ElemQ;
11     P->info.nama = nama;
12     P->info.usia = usia;
13     P->info.pekerjaan = pekerjaan;
14     P->info.prioritas = usia >= 60 || pekerjaan == "tenaga kesehatan";
15     P->info.nomor_antrean = nomor_antrean;
16     P->info.kondisi_darurat = false;
17     P->next = NULL;
18     return P;
19 }
20 void enqueue_103032330054(Queue &Q, ElemQ *P){
21     if (isEmpty_103032330054(Q)){
22         Q.head = P;
23         Q.tail = P;
24     }else if (P->info.prioritas){
25         if (!Q.head->info.prioritas){
26             P->next = Q.head;
27             Q.head = P;
28         }else{
29             ElemQ *temp = Q.head;
30             while (temp->next != NULL && temp->next->info.prioritas){
31                 temp = temp->next;
32             }
33             P->next = temp->next;
34             temp->next = P;
35             if (P->next == NULL){
36                 Q.tail = P;
37             }
38         }
39     }else{
40         Q.tail->next = P;
41         Q.tail = P;
42     }
43 }
```

```
44 void dequeue_103032330054 (Queue &Q, ElemQ *&P) {
45     if (isEmpty_103032330054 (Q)) {
46         P = NULL;
47         cout << "Semua warga telah terlayani." << endl;
48     } else {
49         P = Q.head;
50         Q.head = Q.head->next;
51         if (Q.head == NULL) {
52             Q.tail = NULL;
53         }
54         P->next = NULL;
55     }
56 }
57 ElemQ* front_103032330054 (Queue Q) {
58     return Q.head;
59 }
60 ElemQ* back_103032330054 (Queue Q) {
61     return Q.tail;
62 }
63 int size_103032330054 (Queue Q) {
64     int count = 0;
65     ElemQ *temp = Q.head;
66     while (temp != NULL) {
67         count++;
68         temp = temp->next;
69     }
70     return count;
71 }
```

```
void printInfo_103032330054(Queue Q){
    ElemQ *P = Q.head;
    while (P!=NULL){
        cout << endl << "-----" << endl;
        cout << "Nama: " << P->info.nama << endl;
        cout << "Usia: " << P->info.usia << endl;
        cout << "Pekerjaan: " << P->info.pekerjaan << endl;
        if (P->info.prioritas){
            cout << "Prioritas: Ya" << endl;
        }else{
            cout << "Prioritas: Tidak" << endl;
        }
        cout << "Nomor Antrean: " << P->info.nomor_antrean << endl;
        cout << "-----" << endl;
        P=P->next;
    }
}

void serveQueue_103032330054(Queue &Q){
    if (isEmpty_103032330054(Q)){
        // Empty queue handling
    }else{
        int kouta = 100;
        ElemQ *P;
        cout << endl << "Melakukan pelayanan pada antrean:" << endl;
        while (!isEmpty_103032330054(Q) && kouta != 0){
            dequeue_103032330054(Q,P);
            cout << "Melayanin warga:" << endl;
            cout << "-----" << endl;
            cout << "Nama: " << P->info.nama << endl;
            cout << "Usia: " << P->info.usia << endl;
            cout << "Pekerjaan: " << P->info.pekerjaan << endl;
            if (P->info.prioritas){
                cout << "Prioritas: Ya" << endl;
            }else{
                cout << "Prioritas: Tidak" << endl;
            }
            cout << "Vaksinasi berhasil." << endl;
            cout << "-----" << endl;
            kouta--;
        }
        if (kouta == 0){
            cout << endl << "-----KAPASITAS TELAH PENUH-----" << endl;
        }
        if (!isEmpty_103032330054(Q)){
            cout << endl << "-----SILAKAN DATANG KEMBALI BESOK-----" << endl;
        }
    }
}
```

```
void reassignQueue_103032330054(Queue &Q){
    Queue S;
    createQueue_103032330054(S);
    int P = size_103032330054(Q);
    ElemQ *R;
    while (P != 0 || !isEmpty_103032330054(Q)){
        dequeue_103032330054(Q,R);
        enqueue_103032330054(S,R);
        P--;
    }
    Q = S;
}

void checkWaitingTime_103032330054(Queue &Q, int waktu_sekarang){
    ElemQ *P = Q.head;
    if (waktu_sekarang > 120){
        P = Q.head;
        while (P!=NULL){
            P->info.prioritas = true;
            P=P->next;
        }
        reassignQueue_103032330054(Q);
    }
}

void emergencyHandle_103032330054(Queue &Q, int nomor_antrean){
    ElemQ *P = Q.head;
    while (P!=NULL){
        if (P->info.nomor_antrean == nomor_antrean){
            P->info.kondisi_darurat = true;
            P->info.prioritas = true;
            reassignQueue_103032330054(Q);
            return;
        }
        P=P->next;
    }
    cout << endl << "-----NOMOR ANTREAN TIDAK DITEMUKAN-----" << endl;
}

void updatePriority_103032330054(Queue &Q){
    ElemQ *P = Q.head;
    ElemQ *R;
    checkWaitingTime_103032330054(Q, 130);
    while (P != NULL){
        if (P->info.kondisi_darurat){
            reassignQueue_103032330054(Q);
        }
        P = P->next;
    }
}
```

```

167 ElemQ* findAndRemove_103032330054(Queue &Q, int nomor_antrean){
168     ElemQ *P, *S, *R;
169     P = Q.head;
170     S = Q.head;
171     while (P->next != NULL){
172         if (nomor_antrean == P->info.nomor_antrean){
173             if (P == Q.head){
174                 dequeue_103032330054(Q,R);
175                 return R;
176             }else if (P == Q.tail){
177                 Q.tail = S;
178                 Q.tail->next = NULL;
179                 return P;
180             }else{
181                 S->next = P->next;
182                 P->next = NULL;
183                 return P;
184             }
185         }
186         S = P;
187         P = P->next;
188     }
189     return NULL;
190 }
191

```

#### Queue.h:

```

1  #ifndef QUEUE_H_INCLUDED
2  #define QUEUE_H_INCLUDED
3  #include <iostream>
4  using namespace std;
5  struct Infotype{
6      string nama;
7      int usia;
8      string pekerjaan;
9      bool prioritas;
10     int nomor_antrean;
11     bool kondisi_darurat;
12     int waktu_daftar;
13 };
14 struct ElemQ{
15     Infotype info;
16     ElemQ *next;
17 };
18 struct Queue{
19     ElemQ *head;
20     ElemQ *tail;
21 };
22
23 void createQueue_103032330054(Queue &Q);
24 bool isEmpty_103032330054(Queue Q);
25 ElemQ* createElemQueue_103032330054(string nama, int usia, string pekerjaan, int nomor_antrean);
26 void enqueue_103032330054(Queue &Q, ElemQ *P);
27 void dequeue_103032330054(Queue &Q, ElemQ *&P);
28 ElemQ* front_103032330054(Queue Q);
29 ElemQ* back_103032330054(Queue Q);
30 int size_103032330054(Queue Q);
31 void printInfo_103032330054(Queue Q);
32 void serveQueue_103032330054(Queue &Q);
33 void reassignQueue_103032330054(Queue &Q);
34 void checkWaitingTime_103032330054(Queue &Q, int waktu_sekarang);
35 void emergencyHandle_103032330054(Queue &Q, int nomor_antrean);
36 void updatePriority_103032330054(Queue &Q);
37 ElemQ* findAndRemove_103032330054(Queue &Q, int nomor_antrean);
38 #endif // QUEUE_H_INCLUDED
39

```

Output:

```
"D:\OneDrive - Telkom University\Backup I
Isi antrean awal:
-----
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
-----
```

Melakukan pelayanan pada antrean:

Melakukan pelayanan pada antrean:

Melayanin warga:

```
-----
Nama: John Doe
Usia: 65
Pekerjaan: lansia
Prioritas: Ya
Vaksinasi berhasil.
```

Melayanin warga:

```
-----
Nama: Alice
Usia: 30
Pekerjaan: tenaga kesehatan
Prioritas: Ya
Vaksinasi berhasil.
```

Melayanin warga:

```
-----
Nama: Charlie
Usia: 70
Pekerjaan: pensiunan
Prioritas: Ya
Vaksinasi berhasil.
```

Melayanin warga:

```
-----
Nama: Bob
Usia: 25
Pekerjaan: pekerja
Prioritas: Tidak
Vaksinasi berhasil.
```

Melayanin warga:

```
-----
Nama: David
Usia: 28
Pekerjaan: pekerja
Prioritas: Tidak
Vaksinasi berhasil.
```

Isi antrean setelah pelayanan:

Mengatur ulang antrean berdasarkan prioritas:

-----  
Nama: Edward  
Usia: 22  
Pekerjaan: pekerja  
Prioritas: Tidak  
Nomor Antrean: 6  
-----

Memeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:

-----  
Nama: Edward  
Usia: 22  
Pekerjaan: pekerja  
Prioritas: Ya  
Nomor Antrean: 6  
-----

Menangani kondisi darurat untuk warga dengan nomor antrean 5:

-----NOMOR ANTREAN TIDAK DITEMUKAN-----

-----  
Nama: Edward  
Usia: 22  
Pekerjaan: pekerja  
Prioritas: Ya  
Nomor Antrean: 6  
-----

Mengupdate prioritas antrean setiap jam:

-----  
Nama: Edward  
Usia: 22  
Pekerjaan: pekerja  
Prioritas: Ya  
Nomor Antrean: 6  
-----

Menghapus warga dengan nomor antrean 3:  
Warga dengan nomor antrean 3 tidak ditemukan

-----  
Nama: Edward  
Usia: 22