

Made Naradeon Handika Pramesta
103032300101

Main.cpp

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
1  #include <iostream>
2  #include "queue_vaksinasi.h"
3
4  using namespace std;
5
6  int main()
7  {
8      // Membuat antrean
9      Queue Q;
10     createQueue_103032300101(Q);
11     // Menambahkan beberapa elemen ke dalam queue
12     ElemQ* P1 = createElemQueue_103032300101("John Doe", 65, "lansia", 1);
13     ElemQ* P2 = createElemQueue_103032300101("Alice", 30, "tenaga kesehatan", 2);
14     ElemQ* P3 = createElemQueue_103032300101("Bob", 25, "pekerja", 3);
15     ElemQ* P4 = createElemQueue_103032300101("Charlie", 70, "pensiunan", 4);
16     ElemQ* P5 = createElemQueue_103032300101("David", 28, "pekerja", 5);
17     enqueue_103032300101(Q, P1);
18     enqueue_103032300101(Q, P2);
19     enqueue_103032300101(Q, P3);
20     enqueue_103032300101(Q, P4);
21     enqueue_103032300101(Q, P5);
22     // Menampilkan isi queue
23     cout << "Isi antrean awal:" << endl;
24     printInfo_103032300101(Q);
25     // Melayani antrean
26     cout << "\nMelakukan pelayanan pada antrean:" << endl;
27     serveQueue_103032300101(Q);
28     // Memeriksa antrean setelah pelayanan
29     cout << "\nIsi antrean setelah pelayanan:" << endl;
30     printInfo_103032300101(Q);
31     // Menambahkan elemen baru untuk simulasi pengaturan ulang prioritas
32     ElemQ* P6 = createElemQueue_103032300101("Edward", 22, "pekerja", 6);
```

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
31     // Menambahkan elemen baru untuk simulasi pengaturan ulang prioritas
32     ElemQ* P6 = createElemQueue_103032300101("Edward", 22, "pekerja", 6);
33     enqueue_103032300101(Q, P6);
34     // Simulasi reassignQueue untuk mengatur ulang antrean ke prioritas
35     cout << "\nMengatur ulang antrean berdasarkan prioritas:" << endl;
36     reassignQueue_103032300101(Q);
37     printInfo_103032300101(Q);
38     // Simulasi kondisi warga yang menunggu lebih dari 2 jam
39     cout << "\nMemeriksa waktu tunggu dan mengubah prioritas jika lebih dari 2 jam:" << endl;
40     checkWaitingTime_103032300101(Q, 130); // Asumsikan waktu sekarang 130 menit dari nomor antrean pertama
41     printInfo_103032300101(Q);
42     // Menangani kondisi darurat untuk seorang warga
43     cout << "\nMenangani kondisi darurat untuk warga dengan nomor antrean 5:" << endl;
44     emergencyHandle_103032300101(Q, 5);
45     printInfo_103032300101(Q);
46     // Update prioritas warga setiap jam
47     cout << "\nMengupdate prioritas antrean setiap jam:" << endl;
48     updatePriority_103032300101(Q);
49     printInfo_103032300101(Q);
50     // Mencari dan menghapus warga dengan nomor antrean tertentu
51     cout << "\nMenghapus warga dengan nomor antrean 3:" << endl;
52     ElemQ* removedElem = findAndRemove_103032300101(Q, 3);
53     if (removedElem) {
54         cout << "Warga yang dihapus: " << removedElem->info.nama << endl;
55     }
56     printInfo_103032300101(Q);
57     // Mengcek ukuran queue
58     cout << "\nUkuran antrean saat ini: " << size_103032300101(Q) << endl;
59     return 0;
60 }
61
```

Queue_vaksinasi.cpp

Made Naradeon Handika Pramesta
103032300101

```
queue_vaksinasi.h x queue_vaksinasi.cpp x main.cpp x
1  #include "queue_vaksinasi.h"
2
3  void createQueue_103032300101(Queue &Q) {
4      Q.head = NULL;
5      Q.tail = NULL;
6  }
7
8  bool isEmpty_103032300101(Queue Q) {
9      return Q.head == NULL;
10 }
11
12 ElemQ* createElemQueue_103032300101(string nama, int usia, string pekerjaan, int nomor_antrean) {
13     ElemQ* 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_103032300101(Queue &Q, ElemQ *P) {
25     if (isEmpty_103032300101(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 {
33             ElemQ *temp = Q.head;
34             while (temp->next != NULL && temp->next->info.prioritas) {
```

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
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_103032300101(Queue &Q, ElemQ *&P) {
50     if (isEmpty_103032300101(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
63 ElemQ* front_103032300101(Queue Q) {
64     return Q.head;
65 }
66
67 ElemQ* back_103032300101(Queue Q) {
```

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
67 ElemQ* back_103032300101(Queue Q) {
68     return Q.tail;
69 }
70
71 int size_103032300101(Queue Q) {
72     int count = 0;
73     ElemQ* temp = Q.head;
74     while (temp != NULL) {
75         count++;
76         temp = temp->next;
77     }
78     return count;
79 }
80
81 void printInfo_103032300101(Queue Q) {
82     ElemQ* p = Q.head;
83     if (Q.head == NULL) {
84         cout << "Antrean Kosong" << endl;
85     } else {
86         while (p != NULL) {
87             cout << "Nama: " << p->info.nama << endl;
88             cout << "Usia: " << p->info.usia << endl;
89             cout << "Pekerjaan: " << p->info.pekerjaan << endl;
90             if (p->info.prioritas) {
91                 cout << "Prioritas: Ya" << endl;
92             } else {
93                 cout << "Prioritas: Tidak" << endl;
94             };
95             cout << "Nomor Antrian: " << p->info.nomor_antrean << endl;
96             cout << "-----" << endl;
97             p = p->next;
98         }
99     }
100 }
```

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
101
102 void serveQueue_103032300101(Queue &Q) {
103     if (!isEmpty_103032300101(Q)) {
104         int kouta = 100;
105         ElemQ *P;
106         while (!isEmpty_103032300101(Q) && kouta != 0) {
107             dequeue_103032300101(Q, P);
108             cout << "Melayanin warga:" << endl;
109             cout << "-----" << endl;
110             cout << "Nama: " << P->info.nama << endl;
111             cout << "Usia: " << P->info.usia << endl;
112             cout << "Pekerjaan: " << P->info.pekerjaan << endl;
113             if (P->info.prioritas) {
114                 cout << "Prioritas: Ya" << endl;
115             } else {
116                 cout << "Prioritas: Tidak" << endl;
117             }
118             cout << "Vaksinasi berhasil." << endl;
119             cout << "-----" << endl;
120             kouta--;
121         }
122         if (kouta == 0) {
123             cout << endl << "-----KAPASITAS TELAH PENUH-----" << endl;
124         }
125         if (!isEmpty_103032300101(Q)) {
126             cout << endl << "-----SILAKAN DATANG KEMBALI BESOK-----" << endl;
127         }
128     }
129 }
130
131 void reassignQueue_103032300101(Queue &Q) {
132     Queue S;
133     createQueue_103032300101(S);
134     int P = size_103032300101(Q);
```

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
134     int P = size_103032300101(Q);
135     ElemQ *R;
136     while (P != 0 || !isEmpty_103032300101(Q)) {
137         dequeue_103032300101(Q,R);
138         enqueue_103032300101(S,R);
139         P--;
140     }
141     Q = S;
142 }
143
144 void checkWaitingTime_103032300101(Queue &Q, int waktu_sekarang) {
145     ElemQ *P = Q.head;
146     if (waktu_sekarang > 120) {
147         P = Q.head;
148         while (P!=NULL) {
149             P->info.prioritas = true;
150             P=P->next;
151         }
152     }
153     reassignQueue_103032300101(Q);
154 }
155
156 void emergencyHandle_103032300101(Queue &Q, int nomor_antrean) {
157     ElemQ *P = Q.head;
158
159     while (P != NULL && P->info.nomor_antrean != nomor_antrean) {
160         P=P->next; #define NULL __null
161     };
162
163     if (P == NULL) {
164         printf("Nomor Antrean %d tidak ditemukan \n", nomor_antrean);
165     } else {
166         P->info.kondisi_darurat = true;
167         P->info.prioritas = true;
```

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
167         P->info.prioritas = true;
168         reassignQueue_103032300101(Q);
169     }
170 }
171
172 void updatePriority_103032300101(Queue &Q) {
173     ElemQ *P = Q.head;
174     ElemQ *R;
175     checkWaitingTime_103032300101(Q, 130);
176     while (P != NULL) {
177         if (P->info.kondisi_darurat) {
178             reassignQueue_103032300101(Q);
179         }
180         P = P->next;
181     }
182 }
183 ElemQ* findAndRemove_103032300101(Queue &Q, int nomor_antrean) {
184     ElemQ *P, *S, *R;
185     P = Q.head;
186     S = Q.head;
187     while (P->next != NULL) {
188         if (nomor_antrean == P->info.nomor_antrean) {
189             if (P == Q.head) {
190                 dequeue_103032300101(Q, R);
191                 return R;
192             } else if (P == Q.tail) {
193                 Q.tail = S;
194                 Q.tail->next = NULL;
195                 return P;
196             } else {
197                 S->next = P->next;
198                 P->next = NULL;
199                 return P;
200             }
201         }
202         S = P;
203         P = P->next;
204     }
205     return NULL;
206 }
207
```

Queue_vaksinasi.h

```
queue_vaksinasi.h X queue_vaksinasi.cpp X main.cpp X
1  #ifndef QUEUE_VAKSINASI_H_INCLUDED
2  #define QUEUE_VAKSINASI_H_INCLUDED
3  #include <iostream>
4
5  using namespace std;
6
7  struct Infotype {
8      string nama;
9      int usia;
10     string pekerjaan;
11     bool prioritas;
12     int waktu_daftar;
13     int nomor_antrean;
14     bool kondisi_darurat;
15 };
16
17 struct ElemQ {
18     Infotype info;
19     ElemQ *next;
20 };
21
22 struct Queue {
23     ElemQ *head;
24     ElemQ *tail;
25 };
26
27
28
29 void createQueue_103032300101(Queue &Q);
30 bool isEmpty_103032300101(Queue Q);
```

Made Naradeon Handika Pramesta
103032300101

```
queue_vaksinasi.h  X  queue_vaksinasi.cpp  X  main.cpp  X
18     Infotype info;
19     ElemQ *next;
20 };
21
22 struct Queue {
23     ElemQ *head;
24     ElemQ *tail;
25 };
26
27
28
29 void createQueue_103032300101(Queue &Q);
30 bool isEmpty_103032300101(Queue Q);
31 ElemQ* createElemQueue_103032300101(string nama, int usia, string pekerjaan, int nomor_antrean);
32 void enqueue_103032300101(Queue &Q, ElemQ *P);
33 void dequeue_103032300101(Queue &Q, ElemQ *&P);
34 ElemQ* front_103032300101(Queue Q);
35 ElemQ* back_103032300101(Queue Q);
36 int size_103032300101(Queue Q);
37 void printInfo_103032300101(Queue Q);
38 void serveQueue_103032300101(Queue &Q);
39 void reassignQueue_103032300101(Queue &Q);
40 void checkWaitingTime_103032300101(Queue &Q, int waktu_sekarang);
41 void emergencyHandle_103032300101(Queue &Q, int nomor_antrean);
42 void updatePriority_103032300101(Queue &Q);
43 ElemQ* findAndRemove_103032300101(Queue &Q, int nomor_antrean);
44
45
46 #endif // QUEUE_VAKSINASI_H_INCLUDED
47
```

Hasil :


```
Isi antrean awal:
Nama: John Doe
Usia: 65
Pekerjaan: lansia
Prioritas: Ya
Nomor Antrian: 1
-----
Nama: Alice
Usia: 30
Pekerjaan: tenaga kesehatan
Prioritas: Ya
Nomor Antrian: 2
-----
Nama: Charlie
Usia: 70
Pekerjaan: pensiunan
Prioritas: Ya
Nomor Antrian: 4
-----
Nama: Bob
Usia: 25
Pekerjaan: pekerja
Prioritas: Tidak
Nomor Antrian: 3
-----
Nama: David
Usia: 28
Pekerjaan: pekerja
Prioritas: Tidak
Nomor Antrian: 5
-----
9 Melakukan pelayanan pada antrean:
Melayanin warga:
```

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.

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

Isi antrean setelah pelayanan:
Antrean Kosong

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

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

Menangani kondisi darurat untuk warga dengan nomor antrean 5:
Nomor Antrean 5 tidak ditemukan
Nama: Edward
Usia: 22
```

```
Menangani kondisi darurat untuk warga dengan nomor antrean 5:
```

```
Nomor Antrean 5 tidak ditemukan
```

```
Nama: Edward
```

```
Usia: 22
```

```
Pekerjaan: pekerja
```

```
Prioritas: Ya
```

```
Nomor Antrian: 6
```

```
-----
```

```
Mengupdate prioritas antrean setiap jam:
```

```
Nama: Edward
```

```
Usia: 22
```

```
Pekerjaan: pekerja
```

```
Prioritas: Ya
```

```
Nomor Antrian: 6
```

```
-----
```

```
Menghapus warga dengan nomor antrean 3:
```

```
Nama: Edward
```

```
Usia: 22
```

```
Pekerjaan: pekerja
```

```
Prioritas: Ya
```

```
Nomor Antrian: 6
```

```
-----
```

```
Ukuran antrean saat ini: 1
```

```
Process returned 0 (0x0)   execution time : 0.118 s
```

```
Press any key to continue.
```