

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct InfoTugas {
6      char Pekerjaan;
7      int waktu;
8      int Dikerjakan;
9      string Status;
10 };
11
12 struct ElmTugas {
13     InfoTugas info;
14     ElmTugas* next;
15 };
16
17 struct AntrianTugas {
18     ElmTugas* Head;
19     ElmTugas* Tail;
20 };
21
22 void CreateQueue(AntrianTugas &Q) {
23     Q.Head = nullptr;
24     Q.Tail = nullptr;
25 }
26
27 bool IsEmpty(AntrianTugas Q) {
28     return (Q.Head == nullptr);
29 }
30
31 ElmTugas* Alokasi(char id, int waktu) {
32     ElmTugas* P = new ElmTugas;
33     P->info.Pekerjaan = id;
34     P->info.waktu = waktu;
35     P->info.Dikerjakan = 0;
36     P->info.Status = "unfinished";
37     P->next = nullptr;
38     return P;
39 }
40
41 void TambahTugas(AntrianTugas &Q, ElmTugas* P) {
42     if (IsEmpty(Q)) {
43         Q.Head = P;
44         Q.Tail = P;
45     } else {
46         Q.Tail->next = P;
47         Q.Tail = P;
48     }
49     Q.Tail->next = nullptr;
50 }
51
52 void HapusTugas(AntrianTugas &Q, ElmTugas* &P) {
53     if (!IsEmpty(Q)) {
54         P = Q.Head;
55         Q.Head = Q.Head->next;
56         if (Q.Head == nullptr)
57             Q.Tail = nullptr;
58         P->next = nullptr;
59     } else {
60         P = nullptr;
61     }
62 }
63
64 void TampilQueue(AntrianTugas Q) {
65     ElmTugas* P = Q.Head;
66     cout << "Daftar Tugas Saat Ini:\n";
67     while (P != nullptr) {
68         cout << "  Tugas " << P->info.Pekerjaan
69              << " | Waktu: " << P->info.waktu
70              << " | Dikerjakan: " << P->info.Dikerjakan
71              << " | Status: " << P->info.Status << endl;
72         P = P->next;
73     }
74     cout << endl;
75 }
76

```

```

77 void DailyTask(AntrianTugas &Q) {
78     int total = 0;
79     int jumlahTugas = 0;
80
81     ElmTugas* temp = Q.Head;
82     while (temp != nullptr) {
83         if (temp->info.Status == "unfinished") {
84             jumlahTugas++;
85         }
86         temp = temp->next;
87     }
88
89     TampilQueue(Q);
90
91     while (jumlahTugas > 0) {
92         ElmTugas* P;
93         HapusTugas(Q, P);
94
95         cout << "\nMemproses Tugas " << P->info.Pekerjaan << "... \n";
96         while (P->info.Dikerjakan < P->info.waktu) {
97             P->info.Dikerjakan++;
98             total++;
99         }
100
101         P->info.Status = "finished";
102         cout << "Tugas " << P->info.Pekerjaan << " -> finished.\n";
103
104         jumlahTugas--;
105
106         if (jumlahTugas > 0) {
107             cout << "\nSisa tugas:\n";
108             TampilQueue(Q);
109         }
110     }
111
112     cout << "\nSemua tugas selesai!\n";
113     cout << "Total waktu pengerjaan = " << total << " jam.\n";
114 }

```

● 115

```

116 int main() {
117     AntrianTugas Q;
118     CreateQueue(Q);
119
120     TambahTugas(Q, Alokasi('A', 9));
121     TambahTugas(Q, Alokasi('B', 7));
122     TambahTugas(Q, Alokasi('C', 5));
123     TambahTugas(Q, Alokasi('D', 1));
124
125     DailyTask(Q);
126
127     return 0;
128 }
129

```

```

● PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasQueue> cd "c:\shellyn\ku
liah\semester3\strukturData\teori\tugasQueue\" ; if ($?) { g++ queue.cpp -o que
ue } ; if ($?) { .\queue }
Daftar Tugas Saat Ini:
  Tugas A | Waktu: 9 | Dikerjakan: 0 | Status: unfinished
  Tugas B | Waktu: 7 | Dikerjakan: 0 | Status: unfinished
  Tugas C | Waktu: 5 | Dikerjakan: 0 | Status: unfinished
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas A...
Tugas A -> finished.

Sisa tugas:
Daftar Tugas Saat Ini:
  Tugas B | Waktu: 7 | Dikerjakan: 0 | Status: unfinished
  Tugas C | Waktu: 5 | Dikerjakan: 0 | Status: unfinished
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas B...
Tugas B -> finished.

Sisa tugas:
Daftar Tugas Saat Ini:
  Tugas C | Waktu: 5 | Dikerjakan: 0 | Status: unfinished
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas C...
Tugas C -> finished.

Sisa tugas:
Daftar Tugas Saat Ini:
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas D...
Tugas D -> finished.

Semua tugas selesai!
Total waktu pengerjaan = 22 jam.
○ PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasQueue>

```

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct InfoTugas {
6      char Pekerjaan;
7      int waktu;
8      int Dikerjakan;
9      string Status;
10 };
11
12 struct ElmTugas {
13     InfoTugas info;
14     ElmTugas* next;
15 };
16
17 struct AntrianTugas {
18     ElmTugas* Head;
19     ElmTugas* Tail;
20 };
21
22 void CreateQueue(AntrianTugas &Q) {
23     Q.Head = nullptr;
24     Q.Tail = nullptr;
25 }
26
27 bool IsEmpty(AntrianTugas Q) {
28     return (Q.Head == nullptr);
29 }
30
31 ElmTugas* Alokasi(char id, int waktu) {
32     ElmTugas* P = new ElmTugas;
33     P->info.Pekerjaan = id;
34     P->info.waktu = waktu;
35     P->info.Dikerjakan = 0;
36     P->info.Status = "unfinished";
37     P->next = nullptr;
38     return P;
39 }
40
41 void TambahTugas(AntrianTugas &Q, ElmTugas* P) {
42     if (IsEmpty(Q)) {
43         Q.Head = P;
44         Q.Tail = P;
45     } else {
46         Q.Tail->next = P;
47         Q.Tail = P;
48     }
49     Q.Tail->next = nullptr;
50 }
51
52 void HapusTugas(AntrianTugas &Q, ElmTugas* &P) {
53     if (!IsEmpty(Q)) {
54         P = Q.Head;
55         Q.Head = Q.Head->next;
56         if (Q.Head == nullptr)
57             Q.Tail = nullptr;
58         P->next = nullptr;
59     } else {
60         P = nullptr;
61     }
62 }
63
64 void TampilQueue(AntrianTugas Q) {
65     ElmTugas* P = Q.Head;
66     cout << "Daftar Tugas Saat Ini:\n";
67     while (P != nullptr) {
68         cout << "  Tugas " << P->info.Pekerjaan
69              << " | Waktu: " << P->info.waktu
70              << " | Dikerjakan: " << P->info.Dikerjakan
71              << " | Status: " << P->info.Status << endl;
72         P = P->next;
73     }
74     cout << endl;
75 }
76

```

```

77 void DailyTask(AntrianTugas &Q) {
78     int total = 0;
79     int jumlahTugas = 0;
80
81     ElmTugas* temp = Q.Head;
82     while (temp != nullptr) {
83         if (temp->info.Status == "unfinished") {
84             jumlahTugas++;
85         }
86         temp = temp->next;
87     }
88
89     TampilQueue(Q);
90
91     while (jumlahTugas > 0) {
92         ElmTugas* P;
93         HapusTugas(Q, P);
94
95         cout << "\nMemproses Tugas " << P->info.Pekerjaan << "...\\n";
96         while (P->info.Dikerjakan < P->info.waktu) {
97             P->info.Dikerjakan++;
98             total++;
99         }
100
101         P->info.Status = "finished";
102         cout << "Tugas " << P->info.Pekerjaan << " -> finished.\\n";
103
104         TambahTugas(Q, P);
105         jumlahTugas--;
106
107         if (jumlahTugas > 0) {
108             cout << "\\nSisa tugas:\\n";
109             TampilQueue(Q);
110         }
111     }
112
113     cout << "\\nSemua tugas selesai!\\n";
114     cout << "Total waktu pengerjaan = " << total << " jam.\\n";
115 }

```

```

115
116 v int main() {
117     AntrianTugas Q;
118     CreateQueue(Q);
119
120     TambahTugas(Q, Alokasi('A', 9));
121     TambahTugas(Q, Alokasi('B', 7));
122     TambahTugas(Q, Alokasi('C', 5));
123     TambahTugas(Q, Alokasi('D', 1));
124
125     DailyTask(Q);
126
127     return 0;
128 }
129

```

```

● PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasQueue> cd "c:\shellyn\ku
lah\semester3\strukturData\teori\tugasQueue\" ; if ($?) { g++ queue.cpp -o que
ue } ; if ($?) { .\queue }
Daftar Tugas Saat Ini:
  Tugas A | Waktu: 9 | Dikerjakan: 0 | Status: unfinished
  Tugas B | Waktu: 7 | Dikerjakan: 0 | Status: unfinished
  Tugas C | Waktu: 5 | Dikerjakan: 0 | Status: unfinished
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas A...
Tugas A -> finished.

Sisa tugas:
Daftar Tugas Saat Ini:
  Tugas B | Waktu: 7 | Dikerjakan: 0 | Status: unfinished
  Tugas C | Waktu: 5 | Dikerjakan: 0 | Status: unfinished
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas B...
Tugas B -> finished.

Sisa tugas:
Daftar Tugas Saat Ini:
  Tugas C | Waktu: 5 | Dikerjakan: 0 | Status: unfinished
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas C...
Tugas C -> finished.

Sisa tugas:
Daftar Tugas Saat Ini:
  Tugas D | Waktu: 1 | Dikerjakan: 0 | Status: unfinished

Memproses Tugas D...
Tugas D -> finished.

Semua tugas selesai!
Total waktu pengerjaan = 22 jam.
○ PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasQueue>

```