

Contoh Praktikum Algoritma dan Struktur Data



Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen Pengajar :

Dra.SRI REZEKI CANDRA NURSARI,M.Kom
Prak. Algoritma dan Struktur Data - I

**S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024**

contoh 1

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 typedef struct{
6     string npm;
7     string nama_mhs;
8     double nilai;
9 } nilaiMK;
10
11 typedef struct {
12     int first;
13     int last;
14     nilaiMK Dat[10];
15 } queue;
16
17 void buatqukosong(queue *Q){
18     (*Q).first = -1;
19     (*Q).last = -1;
20 }
21
22 bool iskosong(queue Q){
23     bool hasil = false;
24
25     if(Q.first == -1){
26         hasil = true;
27     }
28     return hasil;
29 }
30
31 bool ispenuhi(queue Q){
32     bool hasil = false;
33
34     if(Q.last == 9){
35         hasil = true;
36     }
37     return hasil;
38 }
39
40 void ADD (string npm, string nama_mhs, double nilai, queue *Q){
41     if(iskosong(*Q) == 1){
42         (*Q).first = 0;
43         (*Q).Dat[0].npm=npm;
44         (*Q).Dat[0].nama_mhs=nama_mhs;
45         (*Q).Dat[0].nilai=nilai;
46     }
47     else{
48         if(ispenuhi(*Q) != 1){
49             (*Q).last=(*Q).last+1;
50             (*Q).Dat[(*Q).last].npm = npm;
51             (*Q).Dat[(*Q).last].nama_mhs = nama_mhs;
52             (*Q).Dat[(*Q).last].nilai = nilai;
53         }
54         else{
55             cout<<"Queue Penuh"<<endl;
56         }
57     }
58 }
59
60 void del(queue *Q){
61     if((*Q).last != 0){
62         (*Q).first++;
63         (*Q).last--;
64     }
65     else{
66         int i;
67         for(i=(*Q).first+1;i<=(*Q).last;i++){
68             (*Q).Dat[i-1].npm=(*Q).Dat[i].npm;
69             (*Q).Dat[i-1].nama_mhs=(*Q).Dat[i].nama_mhs;
70             (*Q).Dat[i-1].nilai=(*Q).Dat[i].nilai;
71         }
72         (*Q).last=(*Q).last-1;
73     }
74 }
75
76 void cetakqueue(queue Q)
77 {
78     if(Q.first != -1)
79     {
80         cout<<"Menampilkan Queue"<<endl;
81         cout<<"-----"<<endl;
82         for(i=Q.first;i<=Q.last;i++)
83         {
84             cout<<"Elemen Ke :"<<i<<endl;
85             cout<<"Nomor Pokok Mahasiswa :"<<Q.Dat[i].npm<<endl;
86             cout<<"Nama Mahasiswa :"<<Q.Dat[i].nama_mhs<<endl;
87             cout<<"Nilai Mahasiswa :"<<Q.Dat[i].nilai<<endl;
88             cout<<"-----"<<endl;
89         }
90     }
91     else
92     {
93         cout<<"Queue Kosong"<<endl;
94     }
95 }
96
97 int main(){
98     queue Q;
99     buatqukosong(&Q);
100     cetakqueue(Q);
101     cout<<endl;
102     cout<<"-----"<<endl;
103     ADD("4523210666","Budi",88.75,&Q);
104     ADD("4523210777","Susi",78.85,&Q);
105     ADD("4523210888","Nuri",98.65,&Q);
106     ADD("4523210999","Bimo",68.55,&Q);
107     ADD("4523210555","Arif",78.85,&Q);
108     ADD("4523210444","Rido",98.65,&Q);
109     ADD("4523210222","Ella",68.55,&Q);
110     cetakqueue(Q);
111     cout<<"-----"<<endl;
112     del(&Q);
113     cetakqueue(Q);
114     cout<<"-----"<<endl;
115     return 0;
116 }
```

```
40 void ADD (string npm, string nama_mhs, double nilai, queue *Q)
41 {
42     if(iskosong(*Q) == 1){
43         (*Q).first = 0;
44         (*Q).last = 0;
45         (*Q).Dat[0].npm=npm;
46         (*Q).Dat[0].nama_mhs=nama_mhs;
47         (*Q).Dat[0].nilai=nilai;
48     }
49     else{
50         if(ispenuhi(*Q) != 1){
51             (*Q).last=(*Q).last+1;
52             (*Q).Dat[(*Q).last].npm = npm;
53             (*Q).Dat[(*Q).last].nama_mhs = nama_mhs;
54             (*Q).Dat[(*Q).last].nilai = nilai;
55         }
56         else{
57             cout<<"Queue Penuh"<<endl;
58         }
59     }
60 }
61
62 void del(queue *Q){
63     if((*Q).last != 0){
64         (*Q).first++;
65         (*Q).last--;
66     }
67     else{
68         int i;
69         for(i=(*Q).first+1;i<=(*Q).last;i++){
70             (*Q).Dat[i-1].npm=(*Q).Dat[i].npm;
71             (*Q).Dat[i-1].nama_mhs=(*Q).Dat[i].nama_mhs;
72             (*Q).Dat[i-1].nilai=(*Q).Dat[i].nilai;
73         }
74         (*Q).last=(*Q).last-1;
75     }
76 }
77
78 void cetakqueue(queue Q)
79 {
80     if(Q.first != -1)
81     {
82         cout<<"Menampilkan Queue"<<endl;
83         cout<<"-----"<<endl;
84         for(i=Q.first;i<=Q.last;i++)
85         {
86             cout<<"Elemen Ke :"<<i<<endl;
87             cout<<"Nomor Pokok Mahasiswa :"<<Q.Dat[i].npm<<endl;
88             cout<<"Nama Mahasiswa :"<<Q.Dat[i].nama_mhs<<endl;
89             cout<<"Nilai Mahasiswa :"<<Q.Dat[i].nilai<<endl;
90             cout<<"-----"<<endl;
91         }
92     }
93     else
94     {
95         cout<<"Queue Kosong"<<endl;
96     }
97 }
98
99 int main(){
100     queue Q;
101     buatqukosong(&Q);
102     cetakqueue(Q);
103     cout<<endl;
104     cout<<"-----"<<endl;
105     ADD("4523210666","Budi",88.75,&Q);
106     ADD("4523210777","Susi",78.85,&Q);
107     ADD("4523210888","Nuri",98.65,&Q);
108     ADD("4523210999","Bimo",68.55,&Q);
109     ADD("4523210555","Arif",78.85,&Q);
110     ADD("4523210444","Rido",98.65,&Q);
111     ADD("4523210222","Ella",68.55,&Q);
112     cetakqueue(Q);
113     cout<<"-----"<<endl;
114     del(&Q);
115     cetakqueue(Q);
116     cout<<"-----"<<endl;
117     return 0;
118 }
```

```
76 void cetakqueue(queue Q)
77 {
78     if(Q.first != -1)
79     {
80         cout<<"Menampilkan Queue"<<endl;
81         cout<<"-----"<<endl;
82         int i;
83         for(i=Q.first;i<=Q.last;i++)
84         {
85             cout<<"Elemen Ke :"<<i<<endl;
86             cout<<"Nomor Pokok Mahasiswa :"<<Q.Dat[i].npm<<endl;
87             cout<<"Nama Mahasiswa :"<<Q.Dat[i].nama_mhs<<endl;
88             cout<<"Nilai Mahasiswa :"<<Q.Dat[i].nilai<<endl;
89             cout<<"-----"<<endl;
90         }
91     }
92     else
93     {
94         cout<<"Queue Kosong"<<endl;
95     }
96 }
97
98 int main(){
99     queue Q;
100     buatqukosong(&Q);
101     cetakqueue(Q);
102     cout<<endl;
103     cout<<"-----"<<endl;
104     ADD("4523210666","Budi",88.75,&Q);
105     ADD("4523210777","Susi",78.85,&Q);
106     ADD("4523210888","Nuri",98.65,&Q);
107     ADD("4523210999","Bimo",68.55,&Q);
108     ADD("4523210555","Arif",78.85,&Q);
109     ADD("4523210444","Rido",98.65,&Q);
110     ADD("4523210222","Ella",68.55,&Q);
111     cetakqueue(Q);
112     cout<<"-----"<<endl;
113     del(&Q);
114     cetakqueue(Q);
115     cout<<"-----"<<endl;
116     return 0;
117 }
```

```
87     cout<<"Elemen Ke :"<<i<<endl;
88     cout<<"Nomor Pokok Mahasiswa :"<<Q.Dat[i].npm<<endl;
89     cout<<"Nama Mahasiswa :"<<Q.Dat[i].nama_mhs<<endl;
90     cout<<"Nilai Mahasiswa :"<<Q.Dat[i].nilai<<endl;
91     cout<<"-----"<<endl;
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }
115 }
116 }
117 }
118 }
119 }
120 }
121 }
122 }
123 }
124 }
125 }
126 }
127 }
```

Pseudocode

Kamus/Deklarasi Variabel fungsi buatSTKosong

-

Algoritma/Dekripsi fungsi buatquKosong(queue *Q)
(*Q).first = -1
(*Q).last = -1

Kamus/Deklarasi Variabel fungsi isKosong
hasil = bool

Algoritma/Deskripsi fungsi isKosong(queue Q)
hasil = false
 if(Q.first == 0)
 hasil = true
 endif
return hasil

Kamus/Deklarasi Variabel fungsi ADD
npm, nama_mhs = char
nilai = double

Algoritma/Deskripsi fungsi ADD(npm, nama_mhs, nilai, queue *Q)
if(iskosong(*Q)==1)
 (*Q).first == 1
 (*Q).last == 1
 (*Q).Dat[10].npm == npm
 (*Q).Dat[10].nama_mhs == nama_mhs
 (*Q).Dat[10].nilai == nilai
 else
 if(ispenuh(*Q)!=1)
 (*Q).Dat[(*Q).last].npm=npm
 (*Q).Dat[(*Q).last].nama_mhs=nama_mhs
 (*Q).Dat[(*Q).last].nilai=nilai
 endif
 print("Queue Penuh")
 endif
endif

Kamus/Deklarasi Variabel fungsi DEL
i = int

Algoritma/Dekripsi fungsi DEL(queue *Q)
if((*Q).last != 0)
 (*Q).first = -1
 (*Q).last = -1
 else
 for (i=((*Q).first+1);i<=(*Q).last;i++)
 (*Q).Dat[i-1].npm=(*Q).Dat[i].npm
 (*Q).Dat[i-1].nama_mhs=(*Q).Dat[i].nama_mhs
 (*Q).Dat[i-1].nilai=(*Q).Dat[i].nilai
 endfor
 (*Q).last=(*Q).last+1

Kamus/Deklarasi Variabel fungsi cetakqueue
i = int

Algoritma/Dekripsi fungsi cetakqueue(queue Q)
if(Q.first != -1)
 print("Menampilkan Queue")
 for(i=Q.last;i>=Q.first;i-)
 print(i)
 print(Q.Dat[i].npm)
 print(Q.Dat[i].nama_mhs)
 print(Q.Dat[i].nilai)
 bantu = bantu->next
 else
 print("Queue Kosong")

Kamus/Deklarasi Variabel fungsi
npm, nama_mhs : char
nilai : double
first, last : int

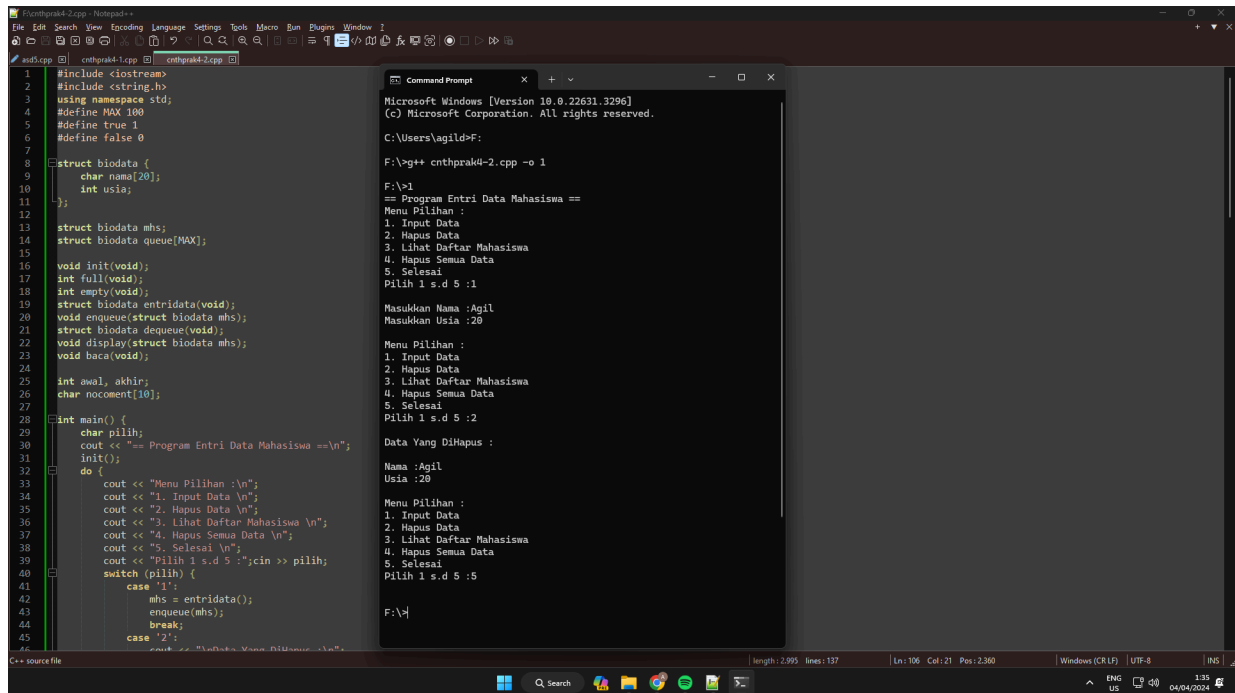
Algoritma/Dekripsi fungsi
typedef struktur (struct{npm,nama_mhs,nilai})
 type NilaiMK
typedef struktur (struct{first, last, NilaiMk Dat[10]})
 type queue
queue Q
buatquKosong(&Q) ; cetakqueue(Q)
 ADD("4523210666","Budi",88.75,&Q);
 ADD("4523210777","Susi",78.85,&Q);
 ADD("4523210888","Nuri",98.65,&Q);
 ADD("4523210999","Bimo",68.55,&Q);
 ADD("4523210555","Arif",78.85,&Q);
 ADD("4523210444","Rido",98.65,&Q);
cetakqueue(Q) ; DEL(&Q) ; cetakqueue(Q) ; DEL(&Q) ; cetakqueue(Q)
print (0)

Algoritma :

```
1.  Membuat fungsi buatQKosong (queue *Q)
2.  (*Q).first = -1
3.  (*Q).last = -1
4.  Membuat fungsi isKosong (queue Q)
5.  hasil = false
6.  Jika (Q.first == -1), maka kerjakan baris 6 s.d 7
7.  Hasil = true
8.  Memanggil fungsi hasil
9.  Membuat fungsi ADD (npm, nama_mhs, nilai, queue *Q)
10. jika (isKosong(*Q)==1)maka kerjakan baris 11 s.d 15
11.     (*Q).first == 1
12.     (*Q).last == 1
13.     (*Q).Dat[10].npm == npm
14.     (*Q).Dat[10].nama_mhs == nama_mhs
15.     (*Q).Dat[10].nilai == nilai
16. jika (isKosong(*Q)), maka kerjakan baris 17 s.d 19 kalau
    tidak baris 20
17.     (*Q).Dat[(*Q).last].npm=npm
18.     (*Q).Dat[(*Q).last].nama_mhs=nama_mhs
19.     (*Q).Dat[(*Q).last].nilai=nilai
20. Menampilkan/mencetak "Queue Penuh"
21. Membuat fungsi Del(queue *Q)
22. Jika ((*Q).last != 0), maka kerjakan baris 23 s.d. 24,
    kalau tidak baris 25
23.     (*Q).first = -1
24.     (*Q).last = -1
25. Selama(i=((*Q).first+1) maka kerjakan baris
26.     (*Q).Dat[i-1].npm=(*Q).Dat[i].npm
27.     (*Q).Dat[i-1].nama_mhs=(*Q).Dat[i].nama_mhs
28.     (*Q).Dat[i-1].nilai=(*Q).Dat[i].nilai
29.     i++
30.     (*Q).last=(*Q).last+1

31. Membuat fungsi CetakQueue(queue Q)
32. Jika(Q.first != NULL)
33. Menampilkan/mencetak "Menampilkan Queue"
34. Selama (i = Q), maka kerjakan baris 35 s.d 37
35. Menampilkan/mencetak nilai variabel i
36. Menampilkan/mencetak nilai variabel Q.Dat[i].npm
37. Menampilkan/mencetak nilai variabel Q.Dat[i].nama_mhs
38. Menampilkan/mencetak nilai variabel Q.Dat[i].nilai
39. i++
40. Menampilkan/mencetak ("Queue Kosong")
41. Membuat fungsi utama
42. Membuat type struktur (npm, nama_mhs, nilai)
43. Membuat objek NilaiMk dari type struktur
44. Membuat Type struktur (first, last)
45. Membuat objek queue dari objek NilaiMK
46. Membuat objek queue dari type struktur
47. Objek Q dari objek queue
48. Memanggil fungsi CetakQueue(Q)
49. Memasukkan isi elemen queue("4523210666","Budi",88.75,&Q)
50. Memasukkan isi elemen queue("4523210777","Susi",78.85,&Q);
51. Memasukkan isi elemen queue("4523210888","Nuri",98.65,&Q);
52. Memasukkan isi elemen queue("4523210999","Bimo",68.55,&Q);
53. Memasukkan isi elemen queue("4523210555","Ari",78.85,&Q);
54. Memasukkan isi elemen queue("4523210444","Rido",98.65,&Q);
55. Memasukkan isi elemen queue("4523210222","Ella",68.55,&Q);
56. Memanggil fungsi CetakQueue(Q)
57. Memanggil fungsi Del(&Q)
58. Memanggil fungsi CetakQueue(Q)
59. Memanggil fungsi Del(&Q)
60. Memanggil fungsi CetakQueue(Q)
61. Selesai
```

contoh 2



```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4 #define MAX 100
5 #define true 1
6 #define false 0
7
8 struct biodata {
9     char nama[20];
10    int usia;
11};
12
13 struct biodata mhs;
14 struct biodata queue[MAX];
15
16 void init(void);
17 int full(void);
18 int empty(void);
19 struct biodata enqueue(struct biodata mhs);
20 void enqueue(struct biodata mhs);
21 struct biodata dequeue(void);
22 void display(struct biodata mhs);
23 void baca(void);
24
25 int awal, akhir;
26 char nocoment[10];
27
28 int main() {
29     char pilih;
30     cout << "== Program Entri Data Mahasiswa ==\n";
31     init();
32     do {
33         cout << "Menu Pilihan :\n";
34         cout << "1. Input Data\n";
35         cout << "2. Hapus Data\n";
36         cout << "3. Lihat Daftar Mahasiswa\n";
37         cout << "4. Hapus Semua Data\n";
38         cout << "5. Selesai\n";
39         cout << "Pilih 1 s.d 5 :"; cin >> pilih;
40         switch (pilih) {
41             case '1':
42                 mhs = enqueue(mhs);
43                 enqueue(mhs);
44                 break;
45             case '2':
46                 mhs = dequeue(mhs);
47                 dequeue(mhs);
48                 break;
49             case '3':
50                 display(mhs);
51                 break;
52             case '4':
53                 baca();
54                 break;
55             case '5':
56                 return 0;
57             default:
58                 cout << "Pilih 1 s.d 5 !";
59                 break;
60         }
61     } while (pilih != '5');
62     return 0;
63 }
64
65 void init(void) {
66     awal = -1;
67     akhir = -1;
68 }
69
70 void enqueue(struct biodata mhs) {
71     if (empty() == true) {
72         awal = 0;
73         akhir = 0;
74         queue[awal] = mhs;
75     } else if (full() != true) {
76         akhir++;
77         queue[akhir] = mhs;
78     } else {
79         cout << "Queue sudah penuh... \n";
80     }
81 }
82
83 struct biodata dequeue(void) {
84     int i;
85     if (empty() != true) {
86         mhs = queue[awal];
87         for (i = awal; i <= akhir; i++)
88             queue[i] = queue[i + 1];
89         akhir--;
90         return mhs;
91     } else {
92         cout << "Queue Kosong... \n";
93     }
94 }
95
96 int full(void) {
97     if (akhir == MAX - 1)
98         return true;
99     else
100         return false;
101 }
102
103 int empty(void) {
104     if (akhir == -1)
105         return true;
106     else
107         return false;
108 }
109
110 void baca(void) {
111     int i;
112     cout << "Isi Queue :\n";
113     if (empty() != true) {
114         for (i = awal; i <= akhir; i++) {
115             display(queue[i]);
116         }
117     } else {
118         cout << "Data Kosong.";
119         cout << "\n";
120     }
121 }
122
123 struct biodata enqueue(struct biodata mhs) {
124     struct biodata mhs;
125     cout << "\n";
126     cout << "Masukkan Nama :";
127     cin >> mhs.nama;
128     cout << "Masukkan Usia :";
129     cin >> mhs.usia;
130     return mhs;
131 }
132
133 void display(struct biodata mhs) {
134     cout << "\n";
135     cout << "Nama : " << mhs.nama << "\n";
136     cout << "Usia : " << mhs.usia << "\n";
137 }
```

```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\agild>F:

F:>g++ cnthprak4-2.cpp -o 1

F:>1
== Program Entri Data Mahasiswa ==
Menu Pilihan :
1. Input Data
2. Hapus Data
3. Lihat Daftar Mahasiswa
4. Hapus Semua Data
5. Selesai
Pilih 1 s.d 5 :1

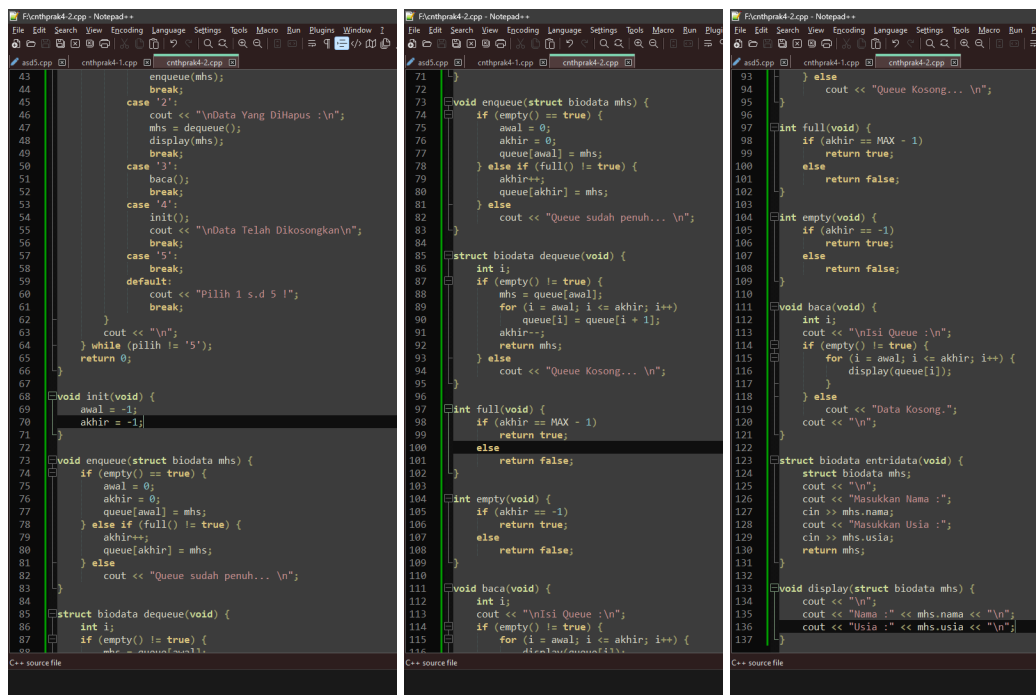
Masukkan Nama :Agil
Masukkan Usia :20

Menu Pilihan :
1. Input Data
2. Hapus Data
3. Lihat Daftar Mahasiswa
4. Hapus Semua Data
5. Selesai
Pilih 1 s.d 5 :2

Data Yang DiHapus :

Nama :Agil
Usia :20

Menu Pilihan :
1. Input Data
2. Hapus Data
3. Lihat Daftar Mahasiswa
4. Hapus Semua Data
5. Selesai
Pilih 1 s.d 5 :5
```



```
43 enqueue(mhs);
44 break;
45 case '2':
46     cout << "\nData Yang DiHapus :\n";
47     mhs = dequeue(mhs);
48     display(mhs);
49     break;
50 case '3':
51     baca();
52     break;
53 case '4':
54     init();
55     cout << "\nData Telah Dikongkan\n";
56     break;
57 case '5':
58     break;
59 default:
60     cout << "Pilih 1 s.d 5 !";
61     break;
62 }
63 cout << "\n";
64 while (pilih != '5');
65 return 0;
66 }
67
68 void init(void) {
69     awal = -1;
70     akhir = -1;
71 }
72
73 void enqueue(struct biodata mhs) {
74     if (empty() == true) {
75         awal = 0;
76         akhir = 0;
77         queue[awal] = mhs;
78     } else if (full() != true) {
79         akhir++;
80         queue[akhir] = mhs;
81     } else {
82         cout << "Queue sudah penuh... \n";
83     }
84 }
85
86 struct biodata dequeue(void) {
87     int i;
88     if (empty() != true) {
89         mhs = queue[awal];
90         for (i = awal; i <= akhir; i++)
91             queue[i] = queue[i + 1];
92         akhir--;
93         return mhs;
94     } else {
95         cout << "Queue Kosong... \n";
96     }
97 }
98
99 int full(void) {
100     if (akhir == MAX - 1)
101         return true;
102     else
103         return false;
104 }
105
106 int empty(void) {
107     if (akhir == -1)
108         return true;
109     else
110         return false;
111 }
112
113 void baca(void) {
114     int i;
115     cout << "Isi Queue :\n";
116     if (empty() != true) {
117         for (i = awal; i <= akhir; i++) {
118             display(queue[i]);
119         }
120     } else {
121         cout << "Data Kosong.";
122         cout << "\n";
123     }
124 }
125
126 struct biodata enqueue(struct biodata mhs) {
127     struct biodata mhs;
128     cout << "\n";
129     cout << "Masukkan Nama :";
130     cin >> mhs.nama;
131     cout << "Masukkan Usia :";
132     cin >> mhs.usia;
133     return mhs;
134 }
135
136 void display(struct biodata mhs) {
137     cout << "\n";
138     cout << "Nama : " << mhs.nama << "\n";
139     cout << "Usia : " << mhs.usia << "\n";
140 }
```

Pseudocode

Kamus/Deklarasi Variabel fungsi
pilih = char

Algoritma/Dekripsi fungsi

```
init()
input ("pilih")
switch (pilih)
    case '1'
        mhs = entridata()
        enqueue(mhs)
        break
    case '2'
        print("Data Yang Dihapus :")
        mhs = dequeue()
        display(mhs)
        break
    case '3'
        baca()
        break
    case '4'
        init()
        print ("Data Telah Dikосongkan")
        break
    case '5'
        break
    default
        print("Pilih 1 s.d 5 !")
        break
while(pilih != '5')
    return 0
```

Kamus/Deklarasi Variabel fungsi init

-

Algoritma/Dekripsi fungsi DEL(queue *Q)

```
awal = -1
akhir = -1
```

Kamus/Deklarasi Variabel fungsi enqueue

-

Algoritma/Dekripsi fungsi enqueue(biodata mhs)

```
if (empty() == true)
    awal = ;
    akhir = 0
    queue[awal] = mhs
else
    if (full() != true)
        akhir++
        queue[akhir] = mhs
    else
        print("Queue sudah penuh... ")
```

Kamus/Deklarasi Variabel fungsi biodata dequeue

i = int

Algoritma/Dekripsi fungsi biodata dequeue

```
if (empty() != true)
    mhs = queue[awal]
    for (i = awal; i <= akhir; i++)
        queue[i] = queue[i + 1]
    akhir--
    return mhs
else
    print("Queue Kosong")
```

Kamus/Deklarasi Variabel fungsi biodata entridata

-

Algoritma/Dekripsi fungsi biodata entridata

```
typedef struktur (struct{nama,usia})
    type biodata mhs
    type biodata queue[MAX]
input (mhs.nama)
input (mhs.usia)
return mhs
```

Kamus/Deklarasi Variabel fungsi display

-

Algoritma/Dekripsi fungsi display (biodata mhs)

```
print (mhs.nama)
print (mhs.usia)
```

Algoritma :

```
1.  Membuat fungsi utama
2.  Memanggil fungsi init()
3.  Memasukkan Nilai Variabel pilih
4.  case '1'
5.      mhs = entridata()
6.      enqueue(mhs)
7.      break
8.  case '2'
9.      Mencetak/Menampilkan ("Data Yang Dihapus :")
10.     mhs = dequeue()
11.     display(mhs);
12.     break
13. case '3'
14.     baca()
15.     break
16. case '4'
17.     init()
18.     Mencetak/Menampilkan Data Telah Dikосongkan/n";
19.     break
20. case '5'
21.     break
22. default
23.     Mencetak/Menampilkan "Pilih 1 s.d 5 !"
24.     break
25.     while (pilih != '5');
26.     Mencetak 0
27.     Membuat fungsi init
28.     awal=-1
29.     akhir=-1

30. Membuat fungsi enqueue(biodata mhs)
31. jika (empty() == true) maka kerjakan baris 32 s.d 34
32. awal=0
33. akhir=0
34. queue[awal]=mhs
35. jika (full() !=true)
36.     akhir++
37.     queue[akhir]=mhs
38. Mencetak "Queue sudah penuh"
39. Membuat biodata dequeue
40. jika (empty() != true) maka kerjakan baris 41 s.d 46 kalau tidak
    baris 47
41. mhs=queue[awal]
42. selama(i=awal) maka kerjakan baris 43 s.d 45
43. queue[i] = queue[i+1]
44. akhir-
45. i++
46. Mencetak mhs
47. Mencetak "Queue Kosong"
48. Membuat fungsi full
49. Jika (akhir==MAX-1)
50. Mencetak true
51. Mencetak false
52. Membuat empty
53. jika (akhir== -1)
54. Mencetak true
55. Mencetak false
56. Membuat fungsi baca
57. jika (empty() != true) maka kerjakan baris 68 s.d 71
58. selama(i=awal) maka kerjakan baris 69 s.d 70
59. display(queue[i])
60. i++
61. Mencetak "Data Kosong"

72. Membuat fungsi biodata entridata
73. Membuat type struktur(nama,usia)
74. Membuat objek biodata mhs
75. Membuat objek biodata queue[MAX]
76. Memasukkan Nilai Variabel mhs.nama
77. Memasukkan Nilai Variabel mhs.usia
78. Mencetak mhs
79. Membuat fungsi display(biodata mhs)
80. Menampilkan/Mencetak Nilai Variabel mhs.nama
81. Menampilkan/Mencetak Nilai Variabel mhs.usia
```