Pembuktian Queue

[Panggil function create()]

[antrian.head=antrian.tail =]

[Cetak Menu:]

==========

PROGRAM QUEUE

==========

1. ENQUEUE

2. DEQUEUE

3. TAMPIL

4. CLEAR

5. EXIT

[Input pil:]

Masukan Pilihan :1

1

[Apakah pil = 1? Ya]

[Input data:]

Masukan Data :9

9

[Panggil function Enqueue(9):]

[Panggi fungsi IsEmpty()]

[Apakah antrian.tail==-1 ? Ya, IsEmpty() = 1]

[Apakah IsEmpty() = 1 ?

[antrian.head = antrian.tail = 0]

[antrian.data[antrian.tail] = data]

[antrian.data[0] = 9]

[Cetak:]

9 sudah dimasukan

[Cetak Menu:]

==========

PROGRAM QUEUE

==========

1. ENQUEUE

2. DEQUEUE

3. TAMPIL

4. CLEAR

5. EXIT

[Input pil:]

Masukan Pilihan :1

1

[Apakah pil = 1? Ya]

[Input data:]

Masukan Data :7

7

[Panggil function Enqueue(7):]

[Panggi fungsi IsEmpty()]

[Apakah antrian.tail==-1 ? Tidak, IsEmpty() 0]

[Apakah IsEmpty() = 1 ?

[Panggi fungsi IsFull()]

[MAX = 4, MAX-1 = 3]

[Apakah antrian.tail = MAX-1 = 3? Tidak, IsFull() = 0]

[Apakah IsFull() 1 ?

[antrian.tail++ -> antrian.tail = 1]

[antrian.data[antrian.tail] = data]

[antrian.data[1] = 7]

[Cetak:]

7 sudah dimasukan

[Cetak Menu:]

==========

PROGRAM QUEUE

==========

1. ENQUEUE

2. DEQUEUE

3. TAMPIL

4. CLEAR

5. EXIT

[Input pil:]

Masukan Pilihan :2

2

[Apakah pil = 1? Tidak]

[Apakah pil = 2? Ya]

[Panggil function Dequeue():]

[e = 9]

[i = 0, apakah 0 <= 0? Ya]

[antrian.data[i]=antrian.data[i+1]]

[antrian.data[0]=antrian.data[1]]

[antrian.data[0]=7]

[i++ -> i = 1]

[i = 1, apakah 1 <= 0? Tidak]

[antrian.tail-- -> antrian.tail = 0]

[9 sudah dihapus]

[Cetak Menu:]

==========

PROGRAM QUEUE

==========

1. ENQUEUE

2. DEQUEUE

3. TAMPIL

4. CLEAR

5. EXIT

[Input pil:]

Masukan Pilihan :1

1

[Apakah pil = 1? Ya]

[Input data:]

Masukan Data :4

4

[Panggil function Enqueue(4):]

[Panggi fungsi IsEmpty()]

[Apakah antrian.tail==-1 ? Tidak, IsEmpty() 0]

[Apakah IsEmpty() = 1 ?

[Panggi fungsi IsFull()]

[MAX = 4, MAX-1 = 3]

[Apakah antrian.tail = MAX-1 = 3? Tidak, IsFull() = 0]

[Apakah IsFull() 1 ?

[antrian.tail++ -> antrian.tail = 1]

[antrian.data[antrian.tail] = data]

[antrian.data[1] = 4]

[Cetak:]

4 sudah dimasukan

[Cetak Menu:]

==========

PROGRAM QUEUE

==========

1. ENQUEUE

2. DEQUEUE

3. TAMPIL

4. CLEAR

5. EXIT

[Input pil:]

Masukan Pilihan :1

1

[Apakah pil = 1? Ya]

[Input data:]

Masukan Data :8

8

[Panggil function Enqueue(8):]

[Panggi fungsi IsEmpty()]

[Apakah antrian.tail==-1 ? Tidak, IsEmpty() 0]

[Apakah IsEmpty() = 1 ?

[Panggi fungsi IsFull()]

[MAX = 4, MAX-1 = 3]

[Apakah antrian.tail = MAX-1 = 3? Tidak, IsFull() = 0]

[Apakah IsFull() 1 ?

[antrian.tail++ -> antrian.tail = 2]

[antrian.data[antrian.tail] = data]

[antrian.data[2] = 8]

[Cetak:]

8 sudah dimasukan

[Cetak Menu:]

==========

PROGRAM QUEUE

==========

1. ENQUEUE

2. DEQUEUE

3. TAMPIL

4. CLEAR

5. EXIT

[Input pil:]

Masukan Pilihan :3

3

[Apakah pil = 1? Tidak]

[Apakah pil = 2? Tidak]

[Apakah pil = 3? Ya]

[Panggil function Tampil():]

[Panggil function IsEmpty()]

[Apakah antrian.tail==-1 ? Tidak, IsEmpty() 0]

[Apakah IsEmpty() = 1 ? Ya]

[i=0, Apakah i <= 2? Ya]

7

[i++ -> i = 1]

[i=1, Apakah i <= 2? Ya]

4

[i++ -> i = 2]

[i=2, Apakah i <= 2? Ya]

8

[i++ -> i = 3]

[i=3, Apakah i <= 2? Tidak]