1. Atribut Q digunakan sebagai penyimpanan data queue dalam bentuk array 1 dimensi
2. Jawab : Max = digunakan sebagai atribut pendeklarasian jumlah data maksimal yang dapat disimpan di dalam queue Size = digunakan sebagai atribut pendeklarasian jumlah data yang sudah masuk dalam queue Front = digunakan sebagai atribut pendeklarasian indeks data yang paling depan dalam queue Rear = digunakan sebagai atribut pendeklarasian indeks data yang paling belakang dalam queue
3. Atribut front dan rear diinisialisasi dengan nilai -1 bukannya 0, karena data disimpan menggunakan tipe data array, yang dimana array selalu diawali dari indeks 0, ketika array kosong maka seharusnya indeksnya masih belum ada, bukannya 0.
4. Yang terjadi adalah data maksimal yang dapat dimasukkan dalam queue berkurang 1 dari yang dideklarasikan di awal. Misalnya maksimal data queue = 4, maka queue hanya dapat menyimpan sebanyak 4-1 = 3 data.
5. Untuk memeriksa rear sebelumnya, karena dalam queue, ketika data pada indeks ke 0 dikeluarkan, maka data baru akan dimasukkan ke dalam indeks 0, dan apakah rear sebelumnya berada pada posisi terakhir dalam indeks array, jika ya maka rear diubah ke indeks 0, dimana data tersebut adalah data yang baru
6. Data baru disimpan pada posisi terakhir dalam queue yaitu pada potongan program

[https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/raw/master/Pertemuan%2010/6.png](https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/blob/master/Pertemuan%2010/6.png)

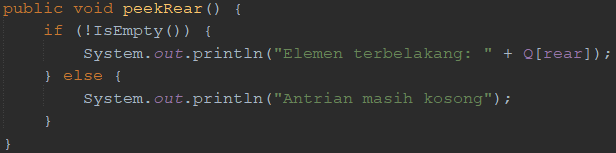
1. Karena pada method Dequeue() memerlukan nilai kembalian, yaitu data yang dikeluarkan dari queue, dan pada main ditampilkan data tersebut
2. Data yang dikeluarkan adalah data yang paling depan ditunjukkan pada potongan program

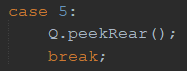
[https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/raw/master/Pertemuan%2010/8.png](https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/blob/master/Pertemuan%2010/8.png)

1. Untuk memeriksa front sebelumnya, karena dalam queue, ketika data pada indeks ke 0 dikeluarkan, maka data baru akan dimasukkan ke dalam indeks 0, dan apakah front sebelumnya berada pada posisi terakhir dalam indeks array, jika ya maka front diubah ke indeks 0, dimana data tersebut adalah data yang baru
2. Baris kode program yang menunjukkan bahwa data terlebih dahulu disimpan variabel lain sebelum diambil/dihapus yaitu pada

[https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/raw/master/Pertemuan%2010/10.png](https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/blob/master/Pertemuan%2010/10.png)

1. Ini dikarenakan data yang masuk dalam queue tidak terurut sesuai indeks arraynya, misalkan array sudah penuh dan perlu didequeue, maka data paling awal dan berada di indeks 0 adalah data yang dikeluarkan, dan setelah itu dienqueue lagi, maka data yang baru masuk akan diletakkan di indeks array 0, sehingga tidak mungkin data yang baru masuk akan didequeue selanjutnya. Jika variabel i selalu 0 maka front akan selalu di indeks ke 0
2. Potongan kode tersebut digunakan untuk mencari nilai i, yang dimana i merupakan indeks dari urutan queue
3. Modifikasi pada project minggu10, package minggu10, class ModifQueue.java dan class ModifQueueMain.java, berikut potongan kode program :

[](https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/blob/master/Pertemuan%2010/13.png)

[](https://github.com/pramudyawibowo/Algoritma-dan-Struktur-Data/blob/master/Pertemuan%2010/13_1.png)