Queue

```
Procedure CrerateListMakanan(In/Out Q : ListMakanan)
Algoritma

Kamus
        Q.first ← NULL
endprocedure
```

```
Function isEmpty(Out Q : ListMakanan) → bool
Kamus

Algoritma
        return Q.first = NULL
endfunction
```

```
Procedure CreateElemen(In/Out Q : ListMakanan, In/Out P : adr)
Kamus

Algoritma
        P ← new Elemen
        Info(P) ←X
        Info(P) ←NULL
endprocedure
```

```
Procedure Enqueue(Im/Out Q : LisrMakanan, In P : adr)
Kamus

Algoritma
        if isEmpty(Q) then
                Q.first ← P
        else
                address last ← Q.first
                while next(last) != NULL do
                        last ← next(last)
                endwhile
                next(last) ← P
        endif
endprocedure
```

```
Procedure Dequeue(In/Out Q : ListMakanan, In/Out P : adr)
Kamus

Algoritma
    If isEmpty(Q) then
        P ← NULL
    else
        P ← Q.first
        Q.first ← next(Q.first)
    endif
endprocedure
```

```
Procedure ShowSemuaMakanan(In Q : ListMakanan)
Kamus

Algoritma
    if isEmpty(Q) then
        output (“Belum Ada Pesanan”)
        return
    endif

    output(“pemesanan”)
    address P ← Q.first
    while P != NULL do
        output(“Pesanan Ke-“, info.ID(P))
        output(“Makanan :”, info.Makanan(P))
        output(“Porsi : “, info.Porsi(P))
        P ← next(P)
    endwhile
endprocedure
```

Stack

```
Procedure CreateStack(In/Out S : stack)
Kamus

Algoritma
        S.top ← -1
endprocedure
```

```
function isEmpty(in S : stack)
Kamus

Algoritma
        Return S.top = -1
endfunnction
```

```
functionn isFull(in S : stack)
Kamus

Algoritma
        Return S.top = maxE1 – 1
endfunnction
```

```
procedure Push(In/Out S : stack, X : infotype)
Kamus

Algoritma
        If isFull(S) then
                Output(“Tidak Ada Piring”)
                Return
        endif
        S.top++
        S.data[S.top] ← X
Endprocedure
```

```
Function Pop(In/Out S : stack) → int
Kamus

Algoritma
    If  isEmpty(S) then
        Output(“Stack is empty”)
        Return -1
    Endif
    Infotype X ← S.data[S.top]
    S.top—
    Return X.ID
Endfunction
```

```
Procedure PrintStack(In S : stack)
Kamus

Algoritma
    If IsEmpty(S)
        Output(“Stack is empty”)
        Return
    endif
    output(“Piring : “)
    for int i ← S.top I down to 0 do
        output(“Piring Ke ”, S.data[i].ID, “ “, “Porsi : ”, S.data[i].Porsi )
    endfor
endprocedure
```

```
function Menu() → int
Kamus
        int choice
Algoritma
        Output("Menu:")
        Output("1. Masukkan Pesanan")
        Output("2. Hapus Pesanan")
        Output("3. List Pesanan")
        Output("4. Tambah Piring")
        Output("5. Hapus Piring")
        Output("6. List Piring")
        Output("7. Exit")
        Output("Enter your choice: ")
        Input(choice)

        Return choice
endprocedure
```