Jurnal Matematika UNAND
Vol. **3** No. **4** Hal. 147 – 151
ISSN : 2303–291X
©Jurusan Matematika FMIPA UNAND

IMPLEMENTASI ANTRIAN DENGAN MENGGUNAKAN *ARRAY*

NELLI HINDRIANI, NARWEN, HAZMIRA YOZZA
Program Studi Matematika,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas,
Kampus UNAND Limau Manis Padang, Indonesia.
email : nellihindriani@gmail.com

Abstract. In a multiprogramming computer, the queues are a data set where addition of elements could only done on a tip on the back side is called rear and the elements had done through the end of the front sides which is called front. Due the queue is a data set, then the appropriate data type for presenting the queue is to use the array. The advantage of using an array as a data type is ability to which can collect multiple data of the same type within a single variable, thus within making the program consists from the several of similar types does not require a lot of variables. In a queues system there are some models of queues that may could be done include queuing model by shifting the position of the element to the previous element, the queues front models were occupied by a new element entered, the queue model in which the move is made by random element, and a queuing model random element position.

Kata Kunci: Queues, Array, Multiprogramming Computer, Pascal Program

1. Pendahuluan

Antrian atau *queue* adalah suatu garis tunggu *(waiting lines)* dari *customer* yang memerlukan layanan dari satu atau lebih fasilitas pelayanan. Definisi lain dari antrian adalah himpunan *customer*, *server* beserta aturan yang mengatur proses pelayanan antara kedatangan *customer* dan pelayanannya. Sistem antrian adalah kedatangan pelanggan untuk mendapatkan pelayanan, menunggu untuk dilayani jika fasilitas pelayanan masih sibuk, mendapatkan pelayanan dan kemudian meninggalkan sistem setelah menerima pelayanan. Sedangkan teori antrian adalah teori yang menyangkut studi probabilistik kejadian garis tunggu *(waiting lines)*, yakni suatu garis tunggu dari *customer* yang memerlukan layanan dari sistem yang ada.

Dalam *multiprogramming computer*, antrian adalah suatu kumpulan data dimana penambahan elemen hanya bisa dilakukan pada suatu ujung di sisi belakang yang disebut *rear* dan pengambilan elemen yang dilakukan melalui ujung sisi depan yang disebut $front$.

2. *Array*

Array atau larik adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Menurut definisinya, *array* adalah sebuah variabel yang dapat menyimpan lebih dari satu nilai sejenis dengan memiliki

tipe data yang sama. Suatu *array* mempunyai jumlah komponen yang banyaknya tetap. Banyaknya komponen dalam suatu *array* ditunjukan oleh suatu indeks yang disebut tipe indeks.

Array merupakan struktur data yang diproses menurut indeksnya. *Array* terdiri dari *array* satu dimensi *(array linier)*, dua dimensi (matriks) dan multidimensi. *Array* dapat bertipe data sederhana *(byte, word, integer, real, boolean, char atau string)* dan tipe data *scalar*. Tipe *array* ini artinya isi dari *array* atau komponen-komponennya atau elemen-elemennya mempunyai nilai dengan tipe data tersebut.

3. Operasi-Operasi Pada Antrian

Dalam antrian terdapat dua operasi dasar yaitu menambahkan elemen (*insert*) dan menghapus elemen (*remove*) dari antrian.

(1) *Insert*
Insert merupakan operasi pada antrian yang berguna untuk menambahkan atau memasukan suatu elemen baru ke dalam sebuah sistem antrian. Dengan menggunakan operasi *insert*, maka dengan sendirinya jumlah elemen dalam sistem juga akan bertambah. Agar tidak terjadi kesalahan, maka sebuah sistem antrian harus diberikan batas maksimum jumlah elemen yang dapat ditampung oleh sistem antrian tersebut.

Jika operasi *insert* tetap dilakukan terhadap antrian yang sudah mencapai batas maksimal, maka pada antrian tersebut akan terjadi kondisi kesalahan yang disebut dengan kondisi *queues overflow* atau kondisi antrian sudah penuh dan melimpah sehingga penambahan elemen baru tidak mungkin dilakukan.

(2) *Remove*
Remove merupakan operasi untuk menghapus elemen dalam suatu sistem antrian. Dengan menggunakan operasi *remove*, maka elemen-elemen yang sudah ada pada antrian dengan sendirinya akan berkurang. Jika sebuah antrian hanya memiliki satu elemen, dan dilakukan operasi *remove* maka sistem tersebut akan menghasilkan antrian kosong atau antrian yang tidak memiliki elemen di dalamnya atau yang dikenal dengan *empty queues*.

Operasi *remove* tidak dapat dilakukan terhadap antrian yang kosong. Oleh karena itu sebelum operasi *remove* akan dilakukan, maka terlebih dahulu harus dilakukan pemeriksaan antrian. Apakah antrian tersebut berada dalam kondisi kosong atau penuh. Jika operasi *remove* tetap dilakukan pada antrian yang kosong, maka hal ini akan menyebabkan terjadinya kondisi kesalahan yang dinamakan *queues underflow*.

4. Implementasi Antrian dengan Menggunakan *Array*

Pada awalnya antrian berada dalam keadaan kosong atau tidak ada satupun elemen di dalamnya. Pada keadaan ini tidak dapat dilakukan penghapusan elemen karena akan meyebabkan kesalahan yang disebut $underflow$. Elemen baru dapat ditambahkan ketika antrian berada dalam keadaan tidak penuh. Jika antrian dalam keadaan penuh maka elemen baru tidak dapat ditambahkan, karena jika dilakukan

penambahan elemen baru maka akan terjadi kesalahan atau yang disebut *overflow*. Dengan demikian elemen tersebut harus menunggu sampai antrian berada dalam keadaan bisa menambahkan elemen baru.

Dalam sebuah sistem antrian terdapat beberapa model antrian yang mungkin bisa dilakukan antara lain adalah model antrian dengan melakukan penggeseran elemen ke posisi elemen sebelumnya, model antrian *front* ditempati oleh elemen yang baru masuk, model antrian dimana perpindahan dilakukan oleh sembarang elemen, dan model antrian posisi elemen sembarang.

Prosedur-prosedur yang dilakukan untuk menggambarkan beberapa model antrian tersebut adalah:

- Prosedur awal yang dilakukan adalah menggambarkan keadaan awal antrian, yaitu dalam keadaan kosong sehingga penambahan elemen dapat dilakukan. Prosedur yang dilakukan untuk menggambarkan keadaan awal antrian :

```
Program Antrian;
Uses crt; Const maxqueue = 20;
Type    queue = record item       : array [1..maxqueue] of string[5];
                        front,rear : 1.. maxqueue; end;

var q  : queue; j : integer;
    posisi : byte;
    pilih,jawab,jawab1 : char;
    elemen, ambil : string[5];


procedure kotak(urut,n,m : integer;tulisan:string);
var i,x,y : integer;
begin
  case urut of
  0  : begin x :=  5; y :=  2;  end; 1  : begin x :=  3; y :=  7; end;
  2  : begin x := 13; y :=  7; end;  3  : begin x := 23; y :=  7; end;
  4  : begin x := 33; y :=  7; end;  5  : begin x := 43; y :=  7; end;
  6  : begin x := 43; y := 12; end;  7  : begin x := 33; y := 12; end;
  8  : begin x := 23; y := 12; end;  9  : begin x := 13; y := 12; end;
  10 : begin x :=  3; y := 12; end; 11 : begin x :=  3; y := 17; end;
  12 : begin x := 13; y := 17; end; 13 : begin x := 23; y := 17; end;
  14 : begin x := 33; y := 17; end; 15 : begin x := 43; y := 17; end;
  16 : begin x := 43; y := 22; end; 17 : begin x := 33; y := 22; end;
  18 : begin x := 23; y := 22; end; 19 : begin x := 13; y := 22; end;
  20 : begin x :=  3; y := 22; end; 21 : begin x := 53; y :=  2; end;
  end;
  gotoxy(x,y); write(chr(218));
  for i := 1 to n do write(chr(196)); write(chr(191));
  for i :=1 to m do begin
     gotoxy(x,y+i);write(chr(179));gotoxy(x+n+1,y+i);write(chr(179));
     gotoxy(x+2,y+i);write(tulisan); end;
  gotoxy(x,y+i+1);write(chr(192));for i := 1 to n do write(chr(196));
  write(chr(217));
  if urut = 0 then begin
     gotoxy(x+2,y+3);
     write('Pelayanan');
  end end;
```

Gambar 1. Prosedur yang dilakukan untuk menggambarkan keadaan awal antrian

- Yang dilakukan pertama kali sebelum memasukan elemen ke dalam antrian adalah memeriksa keadaan awal antrian, apakah berada dalam keadaan kosong atau penuh. Prosedur yang dilakukan untuk memeriksa apakah antrian dalam keadaan kosong adalah:

- Jika antrian dalam keadaan kosong, maka dapat dilakukan penambahan elemen ke dalam antrian. Prosedur yang dilakukan untuk menambahkan elemen ke

```
Function empty( q : queue )  : boolean; Begin
 with q do if front < 1 then empty := true else empty := false;
End;
```

Gambar 2. Prosedur untuk Memeriksa Keadaan Antrian

dalam antrian adalah sebagai berikut:

```
Function remove( var  q : queue ) : string;Begin
  if empty (q) then begin
             gotoxy(30,21);write('Antrian underflow, ');
             gotoxy(20,22);writeln('Anda seharusnya melakukan Insert');end
    else with q do begin
             front := 1; remove := item [front];
         end; end;
```

Gambar 3. Prosedur untuk Menambah Isi Antrian

- Jika elemen meninggalkan antrian, maka dilakukan penghapusan elemen. Prosedur yang dilakukan untuk menghapus isi antrian seperti berikut ini :

```
Procedure insert( var q : queue ; x : string);

Var     sim : integer; Begin
 with q do Begin
          sim := rear;if rear = maxqueue then rear := 1
          else  rear := rear + 1;item[rear] := x;
      End; End;
```

Gambar 4. Prosedur untuk Menghapus Isi Antrian

- Pada program utama ada tiga tahap yang akan dilakukan terhadap antrian, ketiga tahap tersebut adalah memasukkan objek antrian, objek keluar antrian dan perintah keluar. Ketiga langkah tersebut terlebih dahulu dideklarasikan di awal program utama seperti dibawah ini:

Pada tahap pertama yaitu untuk memasukkan objek antrian, dilakukan operasi insert. Operasi insert merupakan operasi untuk menambahkan elemen ke dalam antrian. Operasi ini hanya bisa dilakukan jika antrian tidak berada dalam keadaan penuh, jika operasi insert tetap dilakukan pada antrian yang sudah penuh maka akan terjadi error yang disebut overflow.

Prosedur kedua yang dilakukan adalah prosedur untuk menghapus elemen. Prosedur ini dilakukan dengan menggunakan operasi remove dengan syarat bahwa antrian tidak boleh dalam keadaan kosong, karena ketika antrian dalam keadaan

```
begin
 clrscr;
 { ---- Keadaaan Antrian Mula-mula ---}

 q.front := 1;
 q.rear  := maxqueue;

 kotak(0,10,1,' ');
 for j := 1 to 20 do kotak(j,7,1,' ');
 kotak(21,25,21,' ');


 gotoxy(55,9); write('1.Masukkan Objek Antrian');
 gotoxy(55,11);write('2.Objek Keluar Antrian');
 gotoxy(55,13);write('3.Keluar');
 gotoxy(53,6);write(chr(195));gotoxy(53,15);write(chr(195));
 gotoxy(79,6);write(chr(180));gotoxy(79,15);write(chr(180));
 pilih := '1';
 while (pilih = '1') or (pilih = '2') do begin
  repeat
   gotoxy(55,17);write('Pilihan Anda[1/2/3]:    ');
   gotoxy(55,17);write('Pilihan Anda[1/2/3]: ');  readln(pilih);
  until pilih in ['1','2','3'];

end;
```

Gambar 5. Prosedur Program Utama

kosong dan dilakukan operasi remove maka akan terjadi kesalahan yang disebut underflow. Tahap ketiga merupakan tahap keluar dari model antrian.

5. Ucapan Terima kasih

Penulis mengucapkan terima kasih kepada Ibu Izzati Rahmi H.G, M.Si, Bapak Budi Rudianto, M.Si dan Bapak Dr. Admi Nazra yang telah memberikan masukan dan saran dalam penyempurnaan penulisan artikel ini.

Daftar Pustaka

[1] Faisal, Fachri. 2005. Pendekatan Teori Antrian : Kasus Nasabah Bank pada Pukul 08.00-11.00 WIB di Bank BNI 46 Cabang Bengkulu. Jurnal Gradien Vol 1 Nomor 2. Juli 2005. Universitas Bengkulu. Hal 90 – 97.

[2] Handoko, T Hani dkk. *Dasar-dasar Operation Research*, Yogyakarta. PT. BPFE-Yogyakarta.

[3] Hartono, Jogiyanto. 1988. *Bahasa Turbo Pascal Teori dan Aplikasi Program Komputer Jilid II*. Andy Yogyakarta.

[4] Hillier, Frederick S. dan Gerald J. Lieberman. 2008. *Introduction to Operation Research*. Eighth Edition. Andy.

[5] Munir, Rinaldi. 2007. *Algoritma dan Pemograman dalam Bahasa Pascal dan C+*. Informatika Bandung

[6] Siagian, P. 2001. *Penelitian Operasional Teori dan Praktik*. Universitas Indonesia. Jakarta.

[7] Wospakrik, H.J. 1996. *Teori dan Soal-soal Operation Research*. Bandung. Erlangga

[8] Zakaria, Tedddy Marcus dan Agus Prijono. 2005. *Konsep dan Implementasi Struktur Data*. Informatika.