

SATUAN ACARA PERKULIAHAN (SAP)

Mata Kuliah : Struktur Data
 Kode : TIS3213
 Semester : III
 Waktu : 1 x 3 x 50 Menit
 Pertemuan : 9

A. Kompetensi

1. Utama

Mahasiswa dapat memahami tentang konsep pemrograman menggunakan struktur antrian.

2. Pendukung

Mahasiswa dapat mengetahui implementasi antrian.

B. Pokok Bahasan

Antrian

C. Sub Pokok Bahasan

- Pengertian Antrian
- Implementasi Antrian Dengan Larik

D. Kegiatan Belajar Mengajar

Tahapan Kegiatan	Kegiatan Pengajaran	Kegiatan Mahasiswa	Media & Alat Peraga
Pendahuluan	1. Menjelaskan perkuliahan yang akan dijalani dalam satu semester 2. Menjelaskan materi-materi perkuliahan dan buku-buku acuan yang akan dipergunakan dalam semester ini	Mendengarkan dan memberikan komentar	Notebook, LCD, Papan Tulis
Penyajian	1. Menjelaskan tentang pengertian antrian 2. Menjelaskan tentang implementasi antrian dengan menggunakan larik	Memperhatikan, mencatat, dan memberikan komentar. Mengajukan pertanyaan.	Notebook, LCD, Papan Tulis

Penutup	<ol style="list-style-type: none"> 1. Mengajukan pertanyaan kepada mahasiswa. 2. Memberikan kesimpulan. 3. Mengingatkan akan kewajiban untuk pertemuan selanjutnya. 	<p>Memberikan komentar. Mengajukan dan menjawab pertanyaan</p>	<p>Notebook, LCD, Papan Tulis</p>
---------	--	--	---------------------------------------

E. Evaluasi

Evaluasi dilakukan dengan cara memberikan pertanyaan langsung dan tidak langsung kepada mahasiswa.

F. Daftar Referensi

1. P. Insap Santosa, *Struktur Data Menggunakan Turbo Pascal 6.0*, Andi Offset, Yogyakarta, 2001
2. Wirth Niklaus, "Algorithms and Data Structure", Prentice Hall Int. Inc, 1986
3. Antonie Pranata, *Algoritma dan Pemrograman*, J&J Learning Yogyakarta, 2000
4. Dwi Sanjaya, *Bertualang dengan Struktur Data di Planet Pascal*, J&J Learning Yogyakarta, 2001
5. Materi – Materi dari Internet.

**RENCANA KEGIATAN BELAJAR MINGGUAN
(RKBM)**

Mata Kuliah : Struktur Data
Kode : TIS3213
Semester : III
Waktu : 1 x 3 x 50 Menit
Pertemuan : 9

Minggu Ke-	Topik (Pokok Bahasan)	Metode Pembelajaran	Estimasi Waktu (Menit)	Media
1	2	3	4	5
9	5.1 Pengertian Antrian 5.2 Implementasi Antrian Dengan Larik	Ceramah, Diskusi Kelas	1 x 3 x 50'	Notebook, LCD, Papan Tulis

BAB V

ANTRIAN

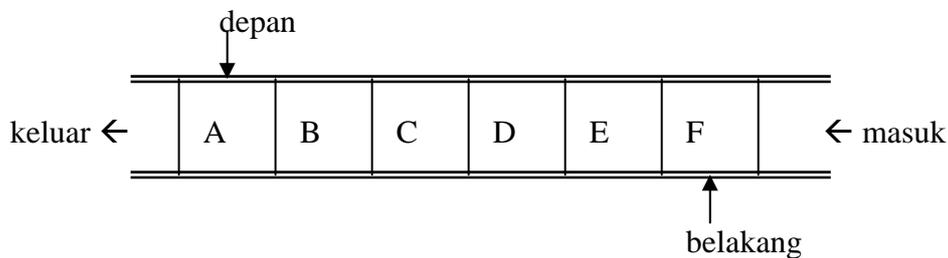
5.1 PENGERTIAN ANTRIAN

Antrian adalah suatu kumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung (disebut dengan sisi belakang atau *rear*), dan penghapusan (pengambilan elemen) dilakukan lewat ujung lain (disebut dengan sisi depan atau *front*).

Antrian menggunakan prinsip "masuk pertama keluar pertama" atau FIFO (*First In First Out*). Dengan kata lain, urutan keluar elemen akan sama dengan urutan masuknya.

5.2 IMPLEMENTASI ANTRIAN DENGAN LARIK

Tipe data yang sesuai untuk menyajikan antrian adalah menggunakan larik dan senarai berantai. Berikut contoh penyajian antrian dengan menggunakan larik :

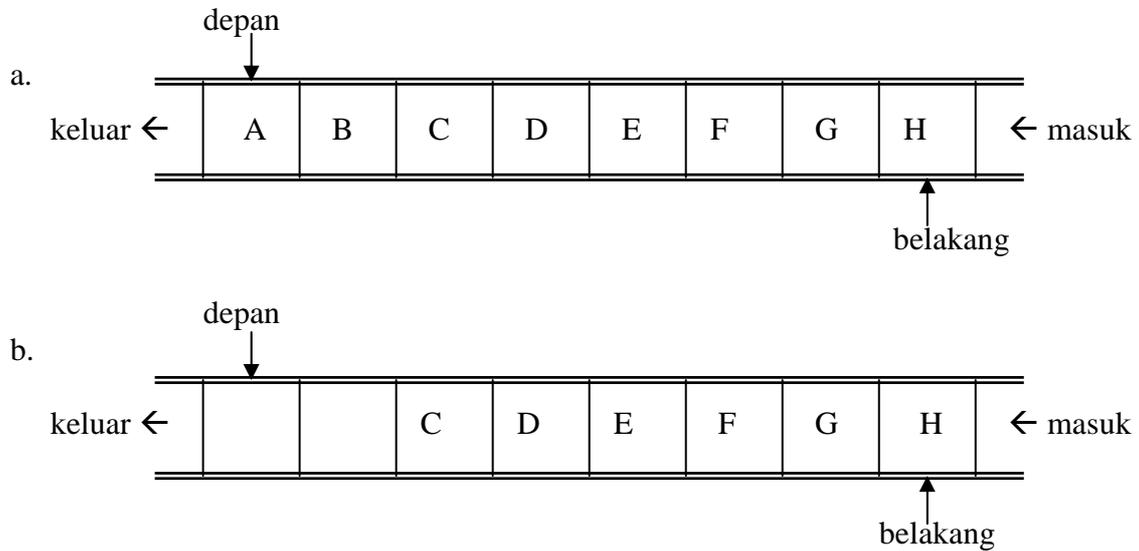


Gambar 5.1 Contoh antrian dengan 6 elemen

Antrian diatas berisi 6 elemen, yaitu A, B, C, D, E dan F. Elemen A terletak dibagian depan antrian elemen F terletak di bagian belakang antrian. Dengan demikian, jika ada elemen baru yang akan masuk maka akan diletakkan disebelah kanan F. Jika ada elemen yang akan dihapus, maka A akan dihapus lebih dahulu.

Seperti halnya pada tumpukan, maka dalam antrian juga mengenal dua operasi dasar, yaitu menambah elemen baru yang akan ditempatkan pada bagian belakang antrian dan menghapus elemen yang terletak dibagian depan antrian. Disamping itu seringkali juga perlu dilihat apakah antrian mempunyai isi atau dalam keadaan kosong.

Contoh penambahan dan penghapusan elemen pada antrian terlihat pada Gambar 5.2 berikut :

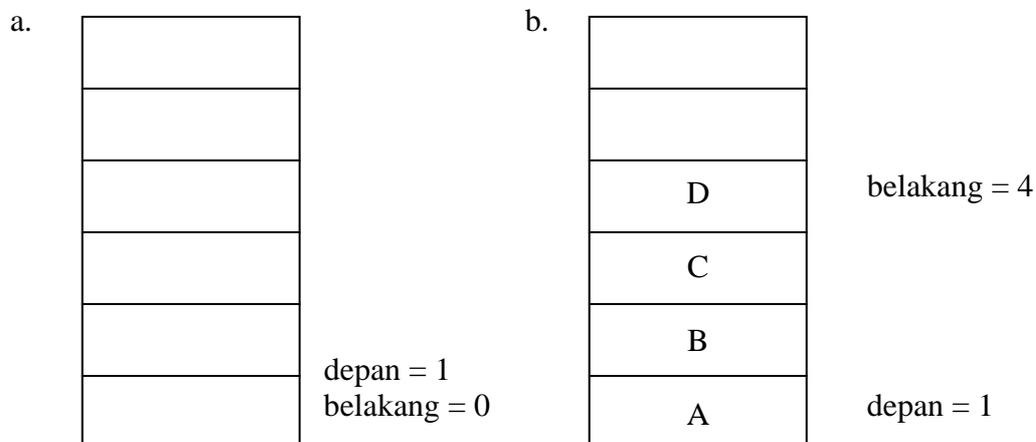


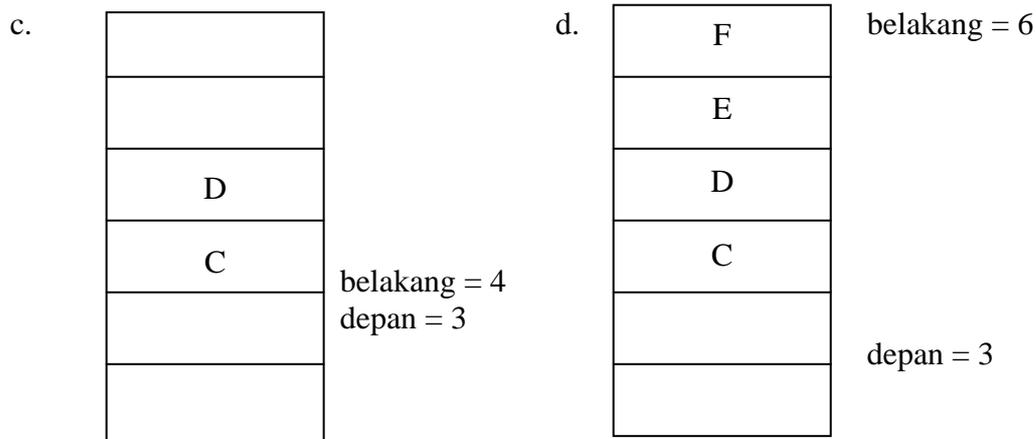
Gambar 5.2 Contoh penambahan dan penghapusan elemen pada suatu antrian

Untuk menyajikan antrian menggunakan larik, maka dibutuhkan deklarasi antrian. Misalnya :

```
const Max_Elemen = 100;
type Antri = Array [ 1 .. Max_Elemen ] of integer;
var Antrian : Antri;
    Depan, belakang : integer;
```

Dengan menggunakan larik, maka kejadian *overflow* sangat mungkin, yakni jika antrian telah penuh, sementara masih ada data yang akan dimasukkan. Contoh :



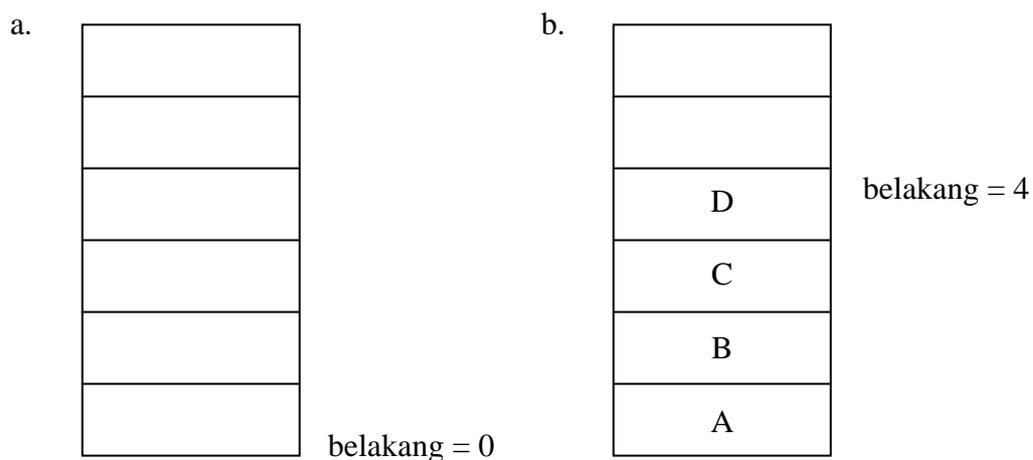


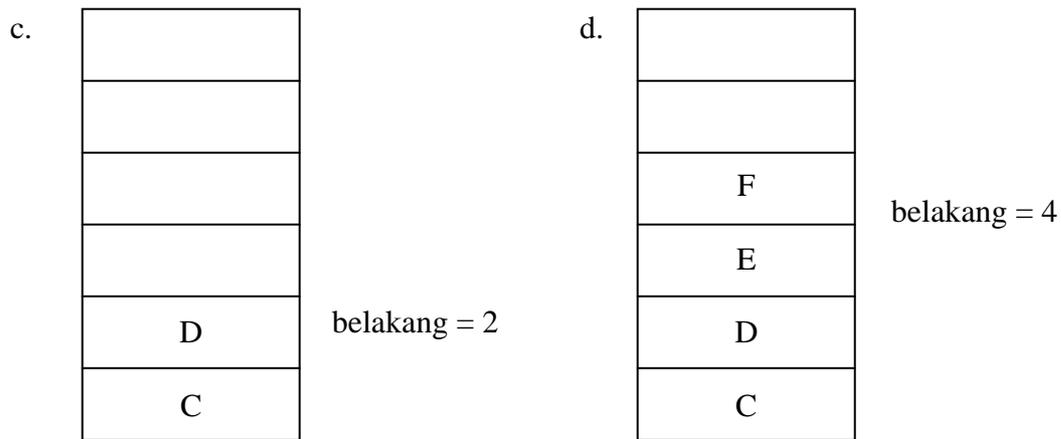
Gambar 5.3 Ilustrasi penambahan dan penghapusan elemen pada sebuah antrian.

Pada saat permulaan (Gambar 5.3.a), antrian dalam keadaan kosong. Pada Gambar 5.3.b terdapat empat buah elemen yang akan ditambahkan. Dalam hal ini depan = 1 dan belakang = 4. Gambar 5.3.c menunjukkan antrian setelah dua elemen dihapus. Gambar 5.3.d menunjukkan antrian setelah dua elemen baru ditambahkan.

Pada ilustrasi diatas, tidak ada elemen yang bisa ditambahkan lagi, padahal antrian masih adan yang kosong,. Salah satu penyelesaiannya adalah dengan mengubah prosedur menghapus elemen, jika ada elemen yang dihapus maka semua elemen lain digeser, sehingga antrian selalu dimulai dari depan = 1.

Contoh :



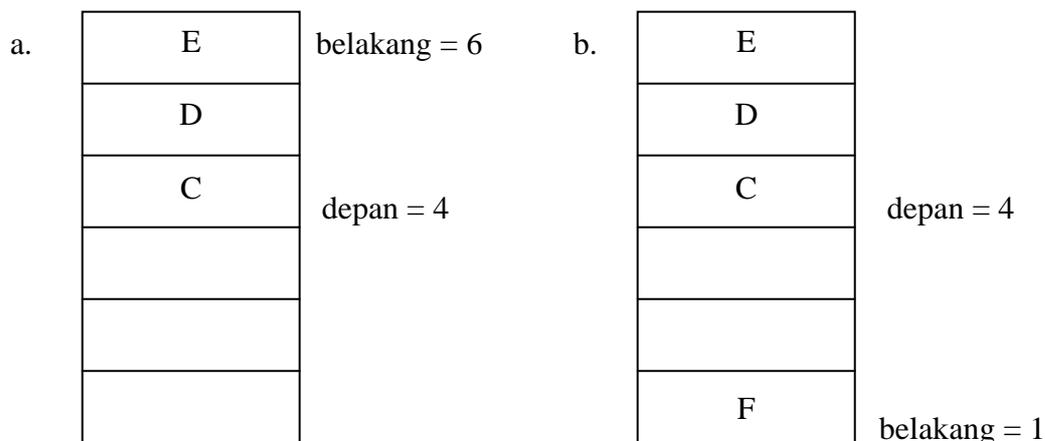


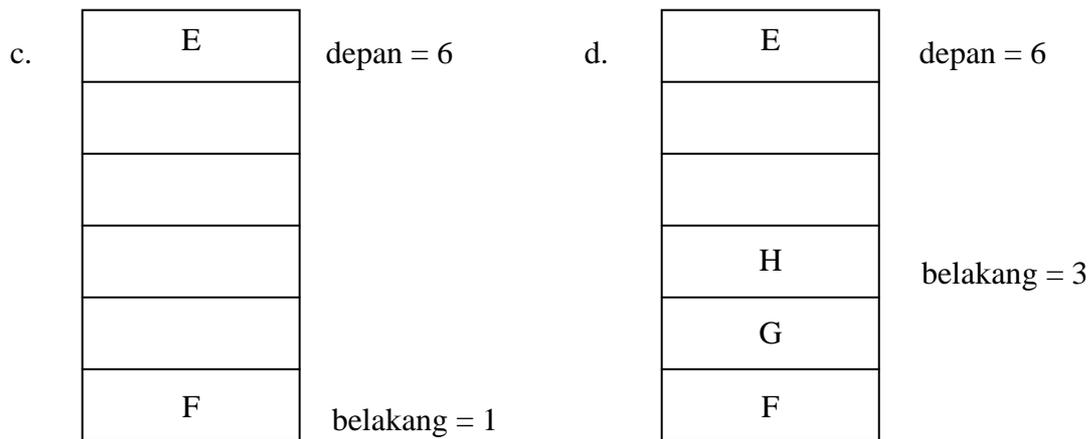
Gambar 5.4 Ilustrasi penambahan dan penghapusan elemen pada sebuah antrian dengan pergeseran

Cara ini kelihatannya sudah memecahkan persoalan yang ada, tetapi jika antrian mempunyai 1000 elemen maka waktu banyak dihabiskan hanya untuk melakukan proses penggeseran. Hal ini tentu saja tidak sangat tidak efisien.

Pemecahannya adalah dengan memperlakukan larik yang menyimpan elemen antrian sebagai larik yang memutar (*circular*), bukan lurus (*straight*), yakni dengan membuat elemen pertama larik terletak berdekatan langsung dengan elemen terakhir larik. Dengan cara ini, meskipun elemen terakhir telah terpakai, elemen baru masih tetap bisa ditambahkan pada elemen pertama, sejauh elemen pertama dalam keadaan kosong.

Contoh :





Gambar 5.5 Ilustrasi penambahan dan penghapusan elemen pada antrian yang diimplementasikan dengan larik yang memutar

Dengan melihat ilustrasi diatas, bisa disusun prosedur untuk menambah elemen baru kedalam antrian, sebagai berikut :

```

procedure TAMBAH (var Q : Antri; X : char);
begin
  if Belakang := Max_Elemen then Belakang := 1
  else Belakang := Belakang + 1;

  if Depan = Belakang then
    writeln ('ANTRIAN SUDAH PENUH')
  else Q [ Belakang ] := X
end

```

Program 5.1 Prosedur untuk menambah elemen baru pada sebuah antrian

Untuk menghapus elemen dari antrian, terlebih dahulu harus dilihat apakah antrian dalam keadaan kosong atau tidak. Berikut disajikan fungsi untuk mengecek keadaan antrian :

```

function KOSONG (Q : Antri) : boolean;
begin
  KOSONG := (Depan = Belakang);
end;

```

Program 5.2 Fungsi untuk mengecek keadaan antrian

Dengan memperhatikan fungsi diatas, maka bisa disusun fungsi lain untuk menghapus elemen, yaitu :

```
function HAPUS (var Q : Antri)      : char;
begin
  if KOSONG(Q) then
    writeln ('ANTRIAN KOSONG...')
  else
    begin
      HAPUS := Q [Depan];
      if Depan := Max_Elemen then
        Depan := 1
      else
        Depan := Depan + 1;
    end;
end;
```

Program 5.3 Fungsi untuk menghapus elemen dari antrian

--ooOOOoo--

Soal & Pembahasan :

Soal :

1. Apakah yang dimaksud dengan antrian (*queue*).
2. Sebutkan prinsip antrian.

Pembahasan :

1. Antrian adalah suatu kumpulan data yang mana penambahan elemen hanya bisa dilakukan pada suatu ujung, dan penghapusan (pengambilan elemen) dilakukan lewat ujung lain.
2. Antrian menggunakan prinsip "masuk pertama keluar pertama" atau FIFO (*First In First Out*). Dengan kata lain, urutan keluar elemen akan sama dengan urutan masuknya.