

[T3] Antrian Puskesmas Pak Dokter

Struktur Data dan Algoritma
Departemen Teknik Komputer ITS

1 Deskripsi Masalah

Pak Dokter baru saja membuka puskesmas kecil di desanya. Karena hanya ada satu dokter dan satu ruang periksa, ia perlu mengatur antrian pasien dengan baik. Puskesmas ini melayani dua jenis pasien: pasien darurat (emergency) dan pasien biasa (regular).

Pak Dokter menyadari bahwa pasien darurat harus didahulukan, tetapi di antara pasien dengan tingkat yang sama, mereka dilayani berdasarkan urutan kedatangan (first-come-first-served). Untuk membantu mengelola antrian ini, Pak Dokter meminta bantuan untuk membuat sistem antrian sederhana.

Sistem antrian memiliki aturan sebagai berikut:

1. **Pasien Emergency:** Selalu didahulukan, diurutkan berdasarkan waktu kedatangan
2. **Pasien Regular:** Dilayani setelah semua emergency selesai, berdasarkan waktu kedatangan
3. Setiap pasien memiliki nomor antrian, nama, dan jenis (emergency/regular)

2 Format Masukan dan Keluaran

2.1 Format Masukan

Masukan terdiri atas beberapa baris. Baris pertama berisi satu buah bilangan bulat positif N ($1 \leq N \leq 50$) yang merupakan banyak operasi. N baris berikutnya masing-masing berisi operasi yang akan dilakukan dengan format sebagai berikut:

- **DAFTAR nomor nama jenis** - Mendaftarkan pasien baru
- **PANGGIL** - Memanggil pasien berikutnya
- **LIHAT** - Melihat status antrian
- **CARI nomor** - Mencari posisi pasien dalam antrian

Batasan:

- $1 \leq nomor \leq 999$
- $1 \leq |nama| \leq 20$, tanpa spasi
- $jenis \in \{E, R\}$ dimana E = Emergency, R = Regular
- Maksimal 30 pasien dalam antrian

2.2 Format Keluaran

Untuk setiap operasi, keluarkan respons sesuai format berikut:

- **DAFTAR:**
 - Jika berhasil: TERDAFTAR
 - Jika nomor sudah ada: NOMOR SUDAH ADA
- **PANGGIL:**
 - Jika ada pasien: PANGGIL nomor nama
 - Jika antrian kosong: ANTRIAN KOSONG
- **LIHAT:** Tampilkan antrian dalam format:
 - EMERGENCY: [daftar emergency]
 - REGULAR: [daftar regular]
 - Jika kosong: ANTRIAN KOSONG
- **CARI:**
 - Jika ditemukan: POSISI nomor_posisi JENIS
 - Jika tidak ditemukan: TIDAK DITEMUKAN

3 Contoh Masukan/Keluaran

Masukan	Keluaran
12 DAFTAR 101 Budi R DAFTAR 102 Siti E DAFTAR 103 Ahmad R LIHAT PANGGIL LIHAT DAFTAR 104 Rina E CARI 103 CARI 104 PANGGIL PANGGIL DAFTAR 101 Doni R	TERDAFTAR TERDAFTAR TERDAFTAR EMERGENCY: 102-Siti REGULAR: 101-Budi 103-Ahmad PANGGIL 102 Siti EMERGENCY: kosong REGULAR: 101-Budi 103-Ahmad TERDAFTAR POSISI 2 REGULAR POSISI 1 EMERGENCY PANGGIL 104 Rina PANGGIL 101 Budi NOMOR SUDAH ADA

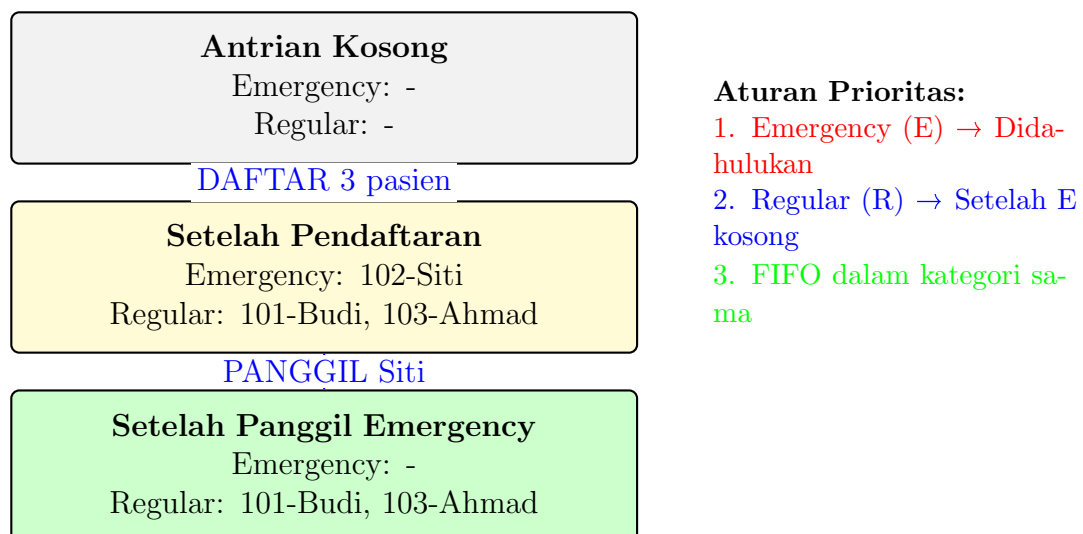
4 Penjelasan

Mari kita ikuti alur operasi pada contoh di atas:

1. **DAFTAR 101 Budi R:** Mendaftarkan Budi sebagai pasien regular dengan nomor 101
2. **DAFTAR 102 Siti E:** Mendaftarkan Siti sebagai pasien emergency dengan nomor 102
3. **DAFTAR 103 Ahmad R:** Mendaftarkan Ahmad sebagai pasien regular dengan nomor 103
4. **LIHAT:** Menampilkan status antrian:
 - Emergency queue: 102-Siti
 - Regular queue: 101-Budi, 103-Ahmad (berdasarkan urutan kedatangan)
5. **PANGGIL:** Memanggil pasien berikutnya. Karena ada emergency (Siti), maka Siti dipanggil terlebih dahulu
6. **LIHAT:** Setelah Siti dipanggil, emergency queue kosong, regular queue masih ada Budi dan Ahmad
7. **DAFTAR 104 Rina E:** Mendaftarkan Rina sebagai pasien emergency
8. **CARI 103:** Mencari Ahmad. Ia ada di posisi 2 dalam antrian regular
9. **CARI 104:** Mencari Rina. Ia ada di posisi 1 dalam antrian emergency
10. **PANGGIL:** Karena ada emergency (Rina), Rina dipanggil
11. **PANGGIL:** Emergency kosong, jadi panggil dari regular queue (Budi)
12. **DAFTAR 101 Doni R:** Mencoba mendaftar dengan nomor 101 yang sudah digunakan Budi (meski Budi sudah dipanggil, nomor tetap reserved)

5 Ilustrasi Sistem

Sistem Antrian Puskesmas Pak Dokter



6 Strategi Implementasi

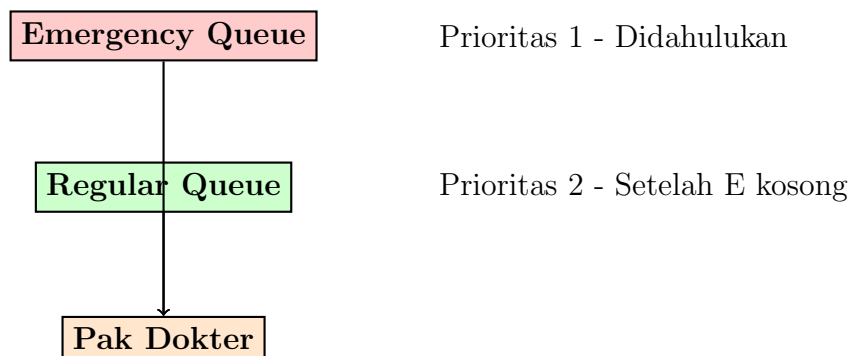
6.1 Struktur Data yang Digunakan

1. **Queue Emergency:** Menyimpan pasien darurat (FIFO)
2. **Queue Regular:** Menyimpan pasien biasa (FIFO)
3. **Array/List:** Untuk tracking nomor yang sudah digunakan

6.2 Algoritma Kunci

1. **Enqueue:** Masukkan ke queue yang sesuai berdasarkan jenis
2. **Dequeue:** Prioritas emergency dulu, baru regular
3. **Linear search:** Untuk operasi CARI
4. **Validation:** Cek duplikasi nomor antrian

7 Diagram Alur Queue



8 Catatan Implementasi

- Gunakan dua queue terpisah: satu untuk emergency, satu untuk regular
- Implementasikan queue dengan array circular atau linked list
- Operasi PANGGIL selalu cek emergency queue dulu
- Untuk CARI, hitung posisi dalam kategori yang sesuai
- Validasi nomor antrian untuk mencegah duplikasi
- Kompleksitas waktu yang diharapkan: $O(1)$ untuk enqueue/dequeue, $O(n)$ untuk pencarian
- Handle edge cases: antrian kosong, nomor tidak ditemukan, queue penuh