

Pembahasan Lotre Star

Deskripsi Singkat

Diberikan N perintah yang akan melakukan operasi pada sebuah mesin lotre. Mesin lotre memiliki X buah pipa yang tersambung pada kotak utama. Setiap pipa dapat diisi bola yang memiliki nilai K.

- Query A (format A x K) : Memasukkan bola bernilai ke dalam pipa ke-x
- Query B (format B x P) : Mengeluarkan bola sebanyak P buah dari pipa ke-x ke kotak utama
- Query C (format C Q) : Mengeluarkan bola sebanyak Q dari semua pipa
- Query D (format D M) : Mengosongkan kotak utama jika total nilai dari bola-bola yang terkumpul di kotak utama sama dengan atau lebih dari M

Ide

Buat array of queues dengan panjang 3001 sesuai dengan batasan yang ditentukan di soal. Buat juga suatu variabel untuk menyimpan nilai dari bola-bola yang sudah keluar dari pipa. Selanjutnya, lengkapi operasi-operasi query pada fungsi processCommand.

Berikut ini adalah potongan program untuk ide tersebut.

```
// Di luar main
private static Queue<Integer>[] queues = new Queue[3001];
private static int mainBox = 0;

// Di dalam processCommand
private static void processCommand(String[] query) throws IOException{
    if(query[0].equals("A")){
        int index = Integer.parseInt(query[1]);
        int value = Integer.parseInt(query[2]);

        queues[index].add(value);
    }
    else if(query[0].equals("B")){
        int index = Integer.parseInt(query[1]);
        int count = Integer.parseInt(query[2]);

        int counter = 0;
        while (!queues[index].isEmpty() && counter < count){
            mainBox += queues[index].poll();
            counter++;
        }
    }
    else if (query[0].equals("C")){
        int count = Integer.parseInt(query[1]);
        for(int i=0;i<queues.length;i++){
            int counter = 0;
            while (!queues[i].isEmpty() && counter < count){
                mainBox += queues[i].poll();
                counter++;
            }
        }
    }
}
```

```

    }
}
else if (query[0].equals("D")){
    int limit = Integer.parseInt(query[1]);
    if (mainBox >= limit){
        int result = mainBox;
        printOutput(String.format("RESET %d", result));
        mainBox = 0;
    }
    else {
        printOutput("GAGAL");
    }
}
}

```

- Query A melakukan add pada queue ke-index sesuai dengan input.
- Query B mengambil queue ke-index dan melakukan poll sebanyak count. Agar tidak terjadi error saat melakukan poll pada queue kosong, gunakan while dengan kondisi queue tidak boleh kosong dan counter kurang dari count.
- Query C mengiterasi semua queues pada array dan melakukan poll sebanyak count pada setiap queue. Agar tidak terjadi error saat melakukan poll pada queue kosong, gunakan while dengan kondisi queue tidak boleh kosong dan counter kurang dari count.

Pada query B dan C, setiap kali pop, jumlahkan hasil pop ke mainBox.

- Query D mengecek nilai mainBox. Jika lebih dari atau sama dengan limit, print "RESET " + nilai mainBox lalu set mainBox menjadi 0 (reset). Sebaliknya, print "GAGAL"