

Dasar-Dasar Pemrograman 2

Lab 02

Methods & Recursion

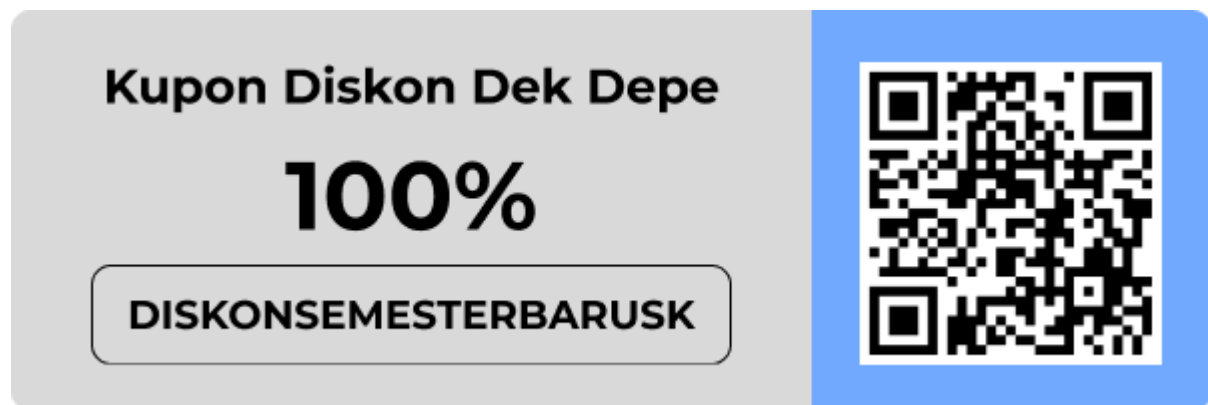


FAKULTAS
ILMU
KOMPUTER

Riwayat Versi

Versi	Timestamp	Keterangan	Warna
1	23-02-2023; 14.00 WIB	Rilis Pertama	-
2	24-02-2023; 14.50 WIB	Keterangan penalti tidak menggunakan rekursi	Kuning

Promo Fotokopi Dek Depe



Deskripsi

Setelah Dek Depe membuka toko fotokopi, ternyata usahanya sangat laris di kalangan mahasiswa! Dengan keuntungan yang Dek Depe telah dapatkan, Dek Depe berniat untuk membuat sistem kupon diskon yang mahasiswa bisa gunakan untuk menambahkan diskon. Namun, Dek Depe sadar bahwa sistem tersebut dapat saja disalahgunakan sehingga Dek Depe ingin memiliki cara untuk memverifikasi apakah kupon tersebut valid atau tidak. Sebagai teman baik Dek Depe, tentu saja kamu akan membantu Dek Depe untuk **mengimplementasikan *checksum* untuk generasi dan validasi kupon** tersebut!

Ketentuan Program

1. Program akan meminta apakah pengguna ingin membuat kupon, validasi kode kupon, atau keluar
2. Apabila membuat kupon:
 - Program akan meminta String yang menjadi kode kupon
 - Kode kupon merupakan String yang hanya terdiri dari karakter kapital
 - Program akan melakukan kalkulasi *checksum*
 - Tambahkan *checksum* karakter ke akhir String
 - Ulangi kembali langkah 2-3 dengan String yang telah ditambah *checksum*
 - Cetak String hasil akhir dengan tambahan **2 karakter *checksum*** di akhir
3. Apabila validasi kupon:
 - Program akan meminta String yang sudah memiliki *checksum*
 - 2 huruf terakhir adalah *checksum* dari kupon tersebut
 - Nama kupon adalah semua karakter yang muncul sebelum *checksum*

- Program melakukan validasi apakah 2 karakter terakhir merupakan *checksum* yang benar atau bukan
 - Apabila benar, program akan mencetak: "Kupon yang diberikan valid"
 - Apabila tidak benar, program akan mencetak: "Kupon yang diberikan tidak valid"
4. Apabila pengguna memilih keluar, maka program akan keluar tanpa pesan apapun
5. **Semua input dijamin valid**
6. **Program harus memiliki setidaknya satu method rekursif. Solusi yang tidak menggunakan rekursi sama sekali akan diberikan penalti -80 poin.**
-

Operasi *checksum* didefinisikan sebagai berikut:

- Jumlahkan nilai dari setiap huruf yang ada
 - Nilai dari suatu huruf adalah posisi huruf tersebut dalam abjad (0-indexing)
Sebagai contoh, huruf A bernilai 0, huruf B bernilai 1, ..., huruf Y bernilai 24, dan huruf Z bernilai 25
- Lakukan modulo 26 ke jumlah yang didapat
- Hasil dikembalikan menjadi huruf, dengan definisi yang sama dengan poin nomor 1



Tips mengubah ke/dari posisi abjad

- Kurangi tipe char dengan 'A' untuk mendapatkan posisi huruf di abjad
Contoh: 'Z' - 'A' akan menghasilkan 25
- Tambah tipe int dengan 'A' dan *casting* dengan (char) untuk mengubah posisi abjad menjadi karakter yang tepat
Contoh: (char)(5 + 'A') akan menghasilkan F

Test Case

```
Halo! Apa yang ingin kamu lakukan?  
[1] Buat kupon  
[2] Validasi kupon  
[3] Keluar  
Pilihan: 1  
Nama kupon: ABC  
Kode kupon adalah: ABCDG  
  
Halo! Apa yang ingin kamu lakukan?
```

```
[1] Buat kupon
[2] Validasi kupon
[3] Keluar
Pilihan: 2
Kupon: ABCDG
Kupon yang diberikan valid

Halo! Apa yang ingin kamu lakukan?
[1] Buat kupon
[2] Validasi kupon
[3] Keluar
Pilihan: 2
Kupon: ABCED
Kupon yang diberikan tidak valid

Halo! Apa yang ingin kamu lakukan?
[1] Buat kupon
[2] Validasi kupon
[3] Keluar
Pilihan: 3
```

Penjelasan:

- Saat membuat kupon:
 - Menentukan nilai huruf:
 - Nilai huruf A adalah 0
 - Nilai huruf B adalah 1
 - Nilai huruf C adalah 2
 - Maka, program akan melakukan:
 $(0 + 1 + 2) \% 26 = 3$, huruf dengan nilai 3 adalah D.
 - String baru adalah ABCD, lakukan checksum sekali lagi, sehingga:
 $(0 + 1 + 2 + 3) \% 26 = 6$, huruf dengan nilai 6 adalah G.
 - String akhir adalah ABCDG.
- Saat validasi kupon:
 - String ABCDG:
 - ABC adalah nama kupon, DG adalah checksum
 - Checksum dari ABC adalah DG, maka kupon valid
 - String ABCED:
 - ABC adalah nama kupon, ED adalah checksum
 - Checksum dari ABC adalah DG, maka kupon tidak valid

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: **1**

Nama kupon: **DDP**

Kode kupon adalah: DDPVQ

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: **2**

Kupon: **DDPVQ**

Kupon yang diberikan valid

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: **2**

Kupon: **DDPOK**

Kupon yang diberikan tidak valid

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: **3**

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: **1**

Nama kupon: **DISKONSEMESTERBARU**

Kode kupon adalah: DISKONSEMESTERBARUSK

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: **1**

Nama kupon: **DEKDEPEKERENBANGET**

Kode kupon adalah: DEKDEPEKERENBANGETEI

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: 3

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: 2

Kupon: REKURSIMEMUTAROTAKNA

Kupon yang diberikan valid

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: 2

Kupon: REKURSIMEMUTAROTAKKU

Kupon yang diberikan tidak valid

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: 2

Kupon: DEKDEPEDIASUKSESMY

Kupon yang diberikan valid

Halo! Apa yang ingin kamu lakukan?

- [1] Buat kupon
- [2] Validasi kupon
- [3] Keluar

Pilihan: 2

Kupon: DEKDEPEDIASUKSESYA

Kupon yang diberikan tidak valid



Input dijamin **valid**, sehingga tidak perlu melakukan validasi input.

Komponen Penilaian

- 50% Kebenaran generasi kupon
- 30% Kebenaran validasi kupon
- 10% Standar penulisan kode
- 10% Dokumentasi

Pengumpulan Berkas

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.
[KodeAsdos]_[Kelas]_[NPM]_[NamaLengkap]_Lab02.zip

Contoh:

DDP_A_1234567890_DekDePe_Lab02.zip