

**Kasus :**

---

**B. Petunjuk Pengerjaan Soal Essay**

1. *Upload source code* yang Anda buat ke akun github masing-masing dan tuliskan link-nya di kulino!
2. Tuliskan penjelasan mengenai *source code* tersebut ke dalam dokumen pdf dan di-*upload* di kulino!

Buatlah program yang dapat menentukan apakah string input adalah persamaan yang valid dalam representasi unary dengan penjumlahan. Contoh:

- a. " $111 + 1111 = 11111 + 1 + 1$ " adalah persamaan yang valid karena  $3 + 4 = 5 + 1 + 1$
- b. " $11 + 1 = 1111$ " tidak valid karena  $2 + 1$  tidak sama dengan 4.
- c. " $11 + 11 + 1 = 11 = 111$ " tidak valid karena hanya satu tanda sama dengan yang diijinkan.
- d. " $111-1 = 11$ " tidak valid karena karakter yang diijinkan hanya 1, +, dan =.

Anda memiliki satu batasan kecil. Anda tidak diizinkan untuk menggunakan aritmatika bawaan komputer, tetapi sebagai gantinya hanya diperbolehkan untuk mencocokkan karakter. Selanjutnya, Anda hanya diperbolehkan satu kali melakukan *scanning* pada *string*. Untuk menyelesaikan soal ini, dapat menggunakan satu *stack*. Ide umumnya adalah lakukan *scanning* pada *string* dan *push* '1' sampai ketemu '='. Setelah itu cocokkan '1' dengan *pop* nilai dari *stack*. Anda dapat menentukan kevalidan string jika *stack* kosong. Jika *stack* kosong dan masih ada angka '1' pada inputan, maka tidak valid. Jika semua inputan telah digunakan, tetapi *stack* tidak kosong, maka tidak valid.

**Solusi:**

Menurut pemahaman saya, program tersebut akan memeriksa apakah element yang dimasukkan ke dalam stack memiliki kesamaan dengan element yang akan dihapus dari stack.

Untuk melewati langkah-langkah yang saya berikan di bawah ini, cukup klik link ini :

<https://github.com/DianAdhiUtama/UTS-PBO-PRAKTIKUM/blob/main/PersamaanDenganUnary.java>

## Langkah-langkah :

1. Buat *class* (yang berarti nama *file* juga sama dengan nama *class* ).  
Kemudian, panggil *library* *arraylist*, *list*, dan *iterator*.

```
import java.util.ArrayList;
import java.util.List;
import java.util.Iterator;

public class PersamaanDenganUnary {
    public static void main(String[] args) {
```

2. karena saya masih di level mencoba dengan *defined input* (input yang bukan berasal dari *user*, saya deklarasikan sebuah persamaan bertipe *string*.

```
// deklarasi persamaan yang akan dicek
String persamaan1 = "11+1=1+1+1";
```

3. kemudian saya membuat sebuah *arraylist* bernama “listDariChars”. Stack yang diminta tadi akan menggunakan *arraylist* ini. *Arraylist* ini akan menyimpan huruf dan karakter milik *string* tadi.

```
// membuat list (array yang dinamis)
//bertipe arraylist(nama lain dari 'list')
// kita beri nama list tersebut 'listDariChars'
List<Character> listDariChars = new ArrayList<>
();
```

4. selanjutnya, saya *push* tiap huruf dan karakter milik *string* “persamaan1” ke dalam *arraylist* tersebut. *if-argument* di dalam perulangan *for* , saya gunakan untuk mengecek apakah *string* mengandung tanda sama dengan (=)

```
// mengubah string ke list berisi char
// kemudian push ke stack 'listDariChars' bertipe Char
acters

for (

char sementara : persamaan1.toCharArray()) {
    listDariChars.add(sembentara);

    if (persamaan1.contains("=")) {
        continue;
    }

}
```

5. membuat *iterator* untuk mengakses tiap anggota *stack* lalu mencetak/menampilkan anggota *stack*.

```
// perulangan while untuk mencetak anggota dari stack
while (aksesAnggota.hasNext()) {
    System.out.println("anggota dari stack yang ditamb
ahkan: " + aksesAnggota.next());

}
System.out.println();
```

6. terakhir adalah memeriksa apakah *arraylist* bernama “listDariChars” sudah kosong (tidak anggota satupun) atau masih berisi (minimal ada 1 anggota di dalamnya). jika terbukti benar, akan *pop* anggota dari stack.

```
// perulangan untuk mengecek ketikan
// stack tidak kosong.
// jika tidak kosong, akan pop anggota dari stack
while (!listDariChars.isEmpty()) {
    char anggotaHilang;
    anggotaHilang = listDariChars.remove(listDariChars
.size() - 1);
    System.out.println("anggota dari stack yang terhap
us : " + anggotaHilang);
}
```

**Berikut hasil dari program tersebut:**

anggota dari stack yang ditambahkan: 1  
anggota dari stack yang ditambahkan: 1  
anggota dari stack yang ditambahkan: +  
anggota dari stack yang ditambahkan: 1  
anggota dari stack yang ditambahkan: =  
anggota dari stack yang ditambahkan: 1  
anggota dari stack yang ditambahkan: +  
anggota dari stack yang ditambahkan: 1  
anggota dari stack yang ditambahkan: +  
anggota dari stack yang ditambahkan: 1

anggota dari stack yang terhapus : 1  
anggota dari stack yang terhapus : +  
anggota dari stack yang terhapus : 1

anggota dari stack yang terhapus : +

anggota dari stack yang terhapus : 1

anggota dari stack yang terhapus : =

anggota dari stack yang terhapus : 1

anggota dari stack yang terhapus : +

anggota dari stack yang terhapus : 1

anggota dari stack yang terhapus : 1