

Tugas Praktikum 4

Function

1. Anda bekerja sebagai seorang pengembang game di sebuah kasino virtual. Manajer kasino meminta Anda untuk membuat versi sederhana dari permainan Blackjack yang dapat dimainkan oleh satu pemain melawan dealer (komputer). Permainan ini harus mematuhi aturan dasar Blackjack:
 1. Pemain dan dealer masing-masing memulai dengan satu kartu.
 2. Pemain dapat terus menarik kartu tambahan hingga mereka memutuskan untuk berhenti atau total nilai kartunya melebihi 21.
 3. Dealer harus menarik kartu tambahan sampai total nilai kartunya mencapai 17 atau lebih.
 4. Pemain menang jika nilai kartunya lebih tinggi dari dealer tanpa melebihi 21, atau jika dealer melebihi 21.
 5. Jika nilai kartu pemain melebihi 21, pemain langsung kalah.
 6. Jika nilai kartu pemain dan dealer sama, permainan dianggap seri.

Note: Implementasikan permainan ini dengan memanfaatkan function.

Format Input: 'y' atau 'Y' untuk mengambil kartu tambahan, 'n' atau 'N' untuk berhenti mengambil kartu.

Format Output: "Welcome to Blackjack!". Output setelah pengambilan kartu: "Kartu anda sekarang adalah: <nilai_kartu>", "Apakah masih akan mengambil kartu? (y/n)". Output jika nilai kartu pemain melebihi 21: "Anda kalah!". Output dealer: "Kartu dealer adalah: <nilai_kartu>" (untuk kartu awal), "Kartu dealer sekarang adalah: <nilai_kartu>" (setelah pengambilan kartu). Output akhir permainan: "Anda kalah, kartu anda melebihi 21.", "Anda menang, dealer melebihi 21.", "Anda menang!", "Seri!", dan "Dealer menang!".

Contoh Test Case 1 (Input & Output):

```
Welcome to Blackjack!
Kartu anda sekarang adalah: 7
Apakah masih akan mengambil kartu? (y/n)
y
Kartu anda sekarang adalah: 17
Apakah masih akan mengambil kartu? (y/n)
y
Kartu anda sekarang adalah: 20
Apakah masih akan mengambil kartu? (y/n)
n
Kartu dealer adalah: 6
Kartu dealer sekarang adalah: 13
Kartu dealer sekarang adalah: 20
Seril!
```

Contoh Test Case 2 (Input & Output):

```
Welcome to Blackjack!
Kartu anda sekarang adalah: 3
Apakah masih akan mengambil kartu? (y/n)
y
Kartu anda sekarang adalah: 9
Apakah masih akan mengambil kartu? (y/n)
y
Kartu anda sekarang adalah: 17
Apakah masih akan mengambil kartu? (y/n)
y
Kartu anda sekarang adalah: 20
Apakah masih akan mengambil kartu? (y/n)
n
Kartu dealer adalah: 8
Kartu dealer sekarang adalah: 17
Anda menang!
```

2. Anda adalah seorang pemburu harta karun yang sedang menjelajahi sebuah pulau terpencil. Misi Anda adalah menemukan harta karun yang tersembunyi dengan berjalan sejauh 50 meter. Namun, ada bahaya yang harus diwaspadai. Jika Anda melangkah lebih dari 20 meter dalam satu langkah, ada kemungkinan Anda akan jatuh ke dalam jebakan berbahaya. Anda harus berhenti menggali jika Anda merasa situasinya tidak aman. Buatlah program yang akan membantu Anda menentukan apakah Anda harus menggali untuk menemukan harta karun atau tidak. Program ini harus memiliki fitur sebagai berikut:

1. Pemain dapat memasukkan jarak langkah yang ditempuh dalam meter.
2. Program harus menghitung total jarak yang telah ditempuh oleh pemain.
3. Program harus memeriksa setiap langkah untuk melihat apakah ada bahaya (langkah lebih dari 20 meter dianggap berbahaya).
4. Jika pemain merasa total jarak yang ditempuh adalah 50 meter, mereka dapat menggali untuk mencari harta karun.
5. Jika ada bahaya yang terdeteksi, program akan menyarankan pemain untuk tidak menggali.
6. Program akan berhenti jika pemain memasukkan nilai 0, dan akan memberikan keputusan apakah aman untuk menggali atau tidak.
7. Tangani inputan yang tidak valid

Note: Implementasikan permainan ini dengan memanfaatkan function.

Format Input: Pemain memasukkan jarak langkah yang ditempuh dalam meter (int). Pemain dapat memasukkan 0 untuk menyelesaikan permainan dan menerima keputusan akhir. Program akan berhenti jika pemain memasukkan nilai negatif atau 0.

Format Output: Tampilkan total jarak yang telah ditempuh: "Total jarak: <total_distance> meter". Jika ada langkah berbahaya: "Tidak aman untuk menggali harta karun. Coba lagi!". Jika total jarak tepat 50 meter dan tidak ada bahaya: "Aman! Kamu tepat menemukan harta karun dan menang!". Jika total jarak bukan 50 meter: "Tidak menemukan harta karun. Coba lagi!".

Contoh Test Case 1 (Input & Output):

```
Masukkan langkah (meter) atau 0 untuk selesai: 10
Masukkan langkah (meter) atau 0 untuk selesai: 15
Masukkan langkah (meter) atau 0 untuk selesai: -2str
Input tidak valid. Masukkan bilangan bulat.
Masukkan langkah (meter) atau 0 untuk selesai: 30
Masukkan langkah (meter) atau 0 untuk selesai: 0
Total jarak: 55 meter
Ada bahaya: Ya
Keputusan: Tidak aman untuk menggali harta karun. Coba lagi!
```

Contoh Test Case 2 (Input & Output):

```
Masukkan langkah (meter) atau 0 untuk selesai: 10
Masukkan langkah (meter) atau 0 untuk selesai: 20
Masukkan langkah (meter) atau 0 untuk selesai: 20
Masukkan langkah (meter) atau 0 untuk selesai: 0
Total jarak: 50 meter
Ada bahaya: Tidak
Keputusan: Aman! Kamu tepat menemukan harta karun dan menang!
```

3. Budi memiliki ketertarikan dengan sebuah teka-teki matematika yang mengharuskan dia untuk mengubah sebuah bilangan bulat positif dengan aturan tertentu hingga bilangan tersebut menjadi 1. Namun, Budi sering kali tidak yakin apakah input yang dia masukkan valid atau tidak. Untuk itu, Budi meminta bantuanmu untuk membuat sebuah program yang dapat membantu menghitung jumlah langkah yang diperlukan, dan juga memastikan bahwa input yang dimasukkan adalah valid.

Aturan permainannya adalah sebagai berikut:

1. Budi akan memasukkan sebuah bilangan bulat positif n
2. Jika n adalah bilangan genap, bagi n dengan 2.
3. Jika n adalah bilangan ganjil, kalikan n dengan 3 lalu tambahkan 1.
4. Ulangi proses ini hingga nilai n menjadi 1.
5. Program harus menghitung dan menampilkan jumlah langkah yang diperlukan untuk mencapai angka 1.
6. Jika input yang dimasukkan bukan bilangan bulat positif, program harus menampilkan pesan kesalahan "Input tidak Valid".

Format Input: Program menerima satu input berupa bilangan bulat positif n .

Format Output: Menampilkan hasil perhitungan dari bilangan n hingga mencapai angka 1. Jumlah langkah yang diperlukan untuk mencapai angka 1: "Jumlah langkah: <step>". Jika input tidak valid, tampilkan pesan kesalahan: "Input tidak Valid".

Contoh Test Case 1 (Input & Output):

```
Masukan angka: 12
6.0
3.0
10.0
5.0
16.0
8.0
4.0
2.0
1.0
Jumlah langkah: 9
```

Contoh Test Case 2 (Input & Output):

```
Masukan angka: str
Input tidak Valid
```

4. Seorang programmer sedang mengembangkan kalkulator sederhana yang dapat melakukan operasi dasar: penjumlahan, pengurangan, perkalian, dan pembagian. Program harus meminta pengguna untuk memasukkan dua angka dan operasi yang diinginkan.

Contoh input 1

```
Selamat datang di Kalkulator Sederhana!
Angka pertama: 10
Angka kedua: 5
Operasi (+, -, *, /): *
```

Contoh output 1

```
Hasil: 15
```

Contoh input 2

```
Selamat datang di Kalkulator Sederhana!
Angka pertama: a
```

Contoh output 2

```
Input tidak valid: could not convert string to float: 'a'
```

Contoh input 3

```
Angka pertama: 15  
Angka kedua: 0  
Operasi (+, -, *, /): /
```

Contoh output 3

```
Pembagian dengan nol tidak diperbolehkan.
```

Contoh input 4

```
Angka pertama: 14  
Angka kedua: 12  
Operasi (+, -, *, /): .
```

Contoh output 4

```
Operasi tidak valid. Gunakan +, -, *, atau /.
```