

Tugas



• **Soal :**

1. Buatlah bash script yang dapat menghitung jumlah kata "bash" dalam suatu kalimat

contoh :

apabila kita belajar bash, maka bash akan mengerti kita

hasil :

jumlah kata bash ada : 2

2. Buatlah bash script yang dapat menghitung jumlah angka genap dalam suatu array, namun apabila terdapat angka lebih dari sama dengan 100 maka hasilnya adalah 0

contoh :

(5 10 7 8 9)

hasilnya : 2

(10 8 9 6 8)

hasilnya : 4

(5 6 100 8 10)

hasilnya : 0



1. Buat file dengan nama jawaban1.sh

```
Activities Terminal Feb 23 18:22
imsalyunus@imsalVM: ~
GNU nano 4.8 jawaban1.sh Modified
#!/bin/bash
# Kalimat yang akan dihitung
kalimat="apabila kita belajar bash, maka bash akan mengerti kita"
# Menghitung jumlah kata "bash" dalam kalimat
jumlah_bash=$(echo "$kalimat" | grep -o -i 'bash' | wc -l)
# Menampilkan hasil
echo "Jumlah kata 'bash' dalam kalimat adalah: $jumlah_bash"

root@imsalVM:/home/imsalyunus# chmod +x jawaban1.sh
root@imsalVM:/home/imsalyunus# ./jawaban1.sh
Jumlah kata 'bash' dalam kalimat adalah: 2
root@imsalVM:/home/imsalyunus# nano jawaban1.sh
root@imsalVM:/home/imsalyunus#
```

Penjelasan:

#!/bin/bash: Ini adalah shebang yang memberitahu sistem bahwa ini adalah script bash dan harus dijalankan menggunakan bash.

kalimat="apabila kita belajar bash, maka bash akan mengerti kita": Variabel kalimat berisi kalimat yang akan dihitung jumlah kata "bash"-nya.

jumlah_bash=\$(echo "\$kalimat" | grep -o -i 'bash' | wc -l): Ini adalah baris yang melakukan perhitungan. echo "\$kalimat" mencetak kalimat ke output, kemudian grep -o -i 'bash' digunakan untuk mencari kata "bash" (baik huruf besar maupun kecil) dalam kalimat dan

menampilkan setiap kecocokan pada baris yang berbeda. Flag -o digunakan agar hanya bagian yang cocok yang dicetak. Flag -i digunakan agar pencarian bersifat case-insensitive. Akhirnya, wc -l digunakan untuk menghitung jumlah baris, yang pada kasus ini sama dengan jumlah kata "bash". Hasilnya disimpan dalam variabel jumlah_bash. echo "Jumlah kata 'bash' dalam kalimat adalah: \$jumlah_bash": Ini akan mencetak jumlah kata "bash" yang ditemukan dalam kalimat. Variabel jumlah_bash diambil dengan menggunakan \$jumlah_bash.

2. Buat file dengan nama jawaban2.sh

```
GNU nano 4.8 jawaban2.sh
#!/bin/bash

# Mendefinisikan array
array=(5 10 7 8 9)
# Inisialisasi variabel jumlah_angka_genap
jumlah_angka_genap=0

# Menghitung jumlah angka genap
for angka in "${array[@]"; do
    # Periksa apakah angka lebih dari atau sama dengan 100
    if (( angka >= 100 )); then
        jumlah_angka_genap=0
        break
    fi

    # Periksa apakah angka genap
    if (( angka % 2 == 0 )); then
        ((jumlah_angka_genap++))
    fi
done

# Menampilkan hasil
echo "${array[@]} hasilnya: $jumlah_angka_genap"

root@imsalVM:/home/imsalyunus# chmod +x jawaban2.sh
root@imsalVM:/home/imsalyunus# ./jawaban2.sh
(5 10 7 8 9) hasilnya: 2
root@imsalVM:/home/imsalyunus#
```

Penjelasan:

#!/bin/bash: Shebang yang menunjukkan bahwa script ini adalah script bash.

array=(5 10 7 8 9): Mendefinisikan array dengan elemen yang diberikan.

jumlah_angka_genap=0: Inisialisasi variabel jumlah_angka_genap untuk menyimpan jumlah angka genap.

Loop for: Looping melalui setiap elemen dalam array.

Pemeriksaan angka: Dalam loop, terdapat dua kondisi yang diperiksa. Pertama, apakah angka lebih besar dari atau sama dengan 100. Jika ya, jumlah_angka_genap diatur menjadi 0 dan loop dihentikan menggunakan break. Kedua, apakah angka genap (modulus 2 sama dengan 0). Jika ya, jumlah_angka_genap akan bertambah satu.

echo "\${array[@]} hasilnya: \$jumlah_angka_genap": Menampilkan hasil jumlah angka genap dalam array. Penggunaan "\${array[@]}" akan menggabungkan semua elemen dalam array untuk ditampilkan. Kemudian, \$jumlah_angka_genap digunakan untuk menampilkan jumlah angka genap yang telah dihitung sebelumnya.