

Modul O6 Terbimbing

REVIEW-SUBPROGRAM

1. **[Fibofiboan Rekursif]** Buatlah program dengan implementasi **Rekursif** berdasarkan persamaan berikut!

$$\begin{cases} 1 & , n = 1 \\ 1 & , n = 2 \\ 4 & , n = 3 \\ \text{fibofiboan}(n-1) + \text{fibofiboan}(n-2) + \text{fibofiboan}(n-3) & , n > 3 \end{cases}$$

```

program utama
kamus
    n: integer
algoritma
    input(n)
    output(fibofiboan(n))
endprogram

```

```

function fibofiboan(n: integer) -> integer
{
    Implementasi persamaan
}

```

Contoh Masukan dan Keluaran

Masukan	Keluaran
5	11
12	802

2. **[Cetak Baris Rekursif]** Buatlah program dengan implementasi **Rekursif** untuk mencetak barisan bilangan sesuai contoh input output

Contoh Masukan dan Keluaran

Input	Output
4	1 + 2 - 3 + 4
8	1 + 2 - 3 + 4 - 5 + 6 - 7 + 8

```
program utama
kamus
    n: integer
algoritma
    input(n)
    cetakBaris(n)
endprogram
```

```
procedure cetakBaris(n: integer)
{
    Mencetak barisan bilangan dengan memperhatikan ganjil dan genap
}
```

Modul O6 Mandiri

REVIEW-SUBPROGRAM

1. **[Vcorp Rewarding]** Dalam rangka perayaan lebaran, Vei, sebagai CEO Vcorp yang baik hati, ingin memberikan hadiah berupa parcel kepada para karyawannya. Dia mempertimbangkan untuk memberikan **bigParcel** kepada pegawai dengan jabatan **Manager** dan **Supervisor**, sedangkan pegawai dengan jabatan **Officer, Staff, dan Intern** akan mendapatkan **smallParcel**.

Setiap parcel membutuhkan kertas pembungkus dengan ukuran tertentu:

- **bigParcel** membutuhkan kertas berukuran **60 cm × 80 cm**
- **smallParcel** membutuhkan kertas berukuran **40 cm × 60 cm**

Vcorp membeli kertas dalam ukuran **80 cm × 100 cm** per lembar. Setiap lembar kertas memiliki harga **Rp. 750**. Jika jumlah total kertas yang dibutuhkan tidak pas dengan ukuran lembar kertas yang tersedia, maka Vcorp harus membeli 1 lembar tambahan.

Biaya lain yang harus diperhitungkan adalah harga parcel:

- **bigParcel** seharga **Rp. 75.000**
- **smallParcel** seharga **Rp. 45.000**

Bantulah Vcorp untuk menghitung:

1. **Jumlah total pegawai yang akan menerima parcel.**
2. **Jumlah total bigParcel dan smallParcel yang dibutuhkan.**
3. **Jumlah total lembar kertas pembungkus yang harus dibeli.**
4. **Total biaya yang diperlukan untuk membeli parcel dan kertas pembungkus.**

Format Input

- Serangkaian input berupa posisi pegawai (string) yang diberikan satu per satu.
- Input berakhir ketika pengguna memasukkan simbol "#" (tanpa tanda kutip).

Format Output

- Vcorp membutuhkan anggaran sebesar: Rp. <total_biaya>, untuk memberikan parcel kepada <total_pegawai> pegawai.
Yang terdiri dari <jumlah_bigParcel> bigParcel dan <jumlah_smallParcel> smallParcel.

INPUT	OUTPUT
#	
supervisor manager manager	Vcorp membutuhkan anggaran sebesar: Rp. 1161750, untuk memberikan parcel kepada 23 pegawai. Yang terdiri dari 4 bigParcel dan 19 smallParcel

manager officer officer officer staff staff staff staff staff staff staff staff staff intern intern intern intern intern #	
supervisor manager manager manager supervisor officer manager staff staff manager intern manager intern staff #	Vcorp membutuhkan anggaran sebesar: Rp. 875250, untuk memberikan parcel kepada 14 pegawai. Yang terdiri dari 8 bigParcel dan 6 smallParcel

GLOBAL VARIABLES

```

{ukuran paper yang dibutuhkan bigParcel}
constant bigParcelPaper: integer = 60 * 80

{ukuran paper yang dibutuhkan smallParcel}
constant smallParcelPaper: integer = 40 * 60

{ukuran satuan paper yang tersedia oleh penjual}
constant paperSize: integer = 80 * 100

{cost bigParcel}
constant bigParcelCost: integer = 75000

{cost smallParcel}
constant smallParcelCost: integer = 45000

{cost per paper}
constant paperCost: integer = 750

```

```

program utama

kamus
    employee: string
    highLevel, entryLevel, totalEmployeesVal: integer
    totalPaperNeededSizeVal, totalPaperNeededVal, totalCostVal: integer

algoritma
    {input data pertama}
    —

    {loop untuk menerima inputan yang berhenti bila inputnya adalah "#"}
    —
    {panggil prosedur untuk menghitung jumlah pegawai berdasarkan posisi}
    —

    {panggil fungsi untuk menghitung ukuran total paper}
    —

    {panggil fungsi untuk menghitung jumlah total paper yang dibutuhkan}
    —

    {panggil fungsi untuk menghitung jumlah total employees yang dimiliki Vcorp}
    —

    {panggil fungsi untuk melakukan pembaharuan nilai jumlah total paper yang
    dibutuhkan}
    —

    {panggil fungsi untuk melakukan perhitungan total cost yang dibutuhkan Vcorp}
    —

    {print sesuai contoh output}
    —

endprogram

```

```

procedure checkEmployees(in position: string, in/out highLevel, entryLevel: integer)
{
    I.S. position adalah string posisi pegawai, highLevel dan entryLevel adalah jumlah pegawai di
    setiap level
    F.S. highLevel bertambah jika posisi adalah "manager" atau "supervisor",
    entryLevel bertambah jika posisi adalah "officer", "staff", atau "intern"
}

```

```

function totalEmployees(highLevel, entryLevel: integer) -> integer
{
    I.S. highLevel dan entryLevel adalah jumlah pegawai di setiap level
    F.S. mengembalikan jumlah seluruh pegawai
}

```

```

function totalPaperNeededSize(highLevel, entryLevel: integer) -> integer
{
    I.S. highLevel dan entryLevel adalah jumlah pegawai di setiap level
    F.S. mengembalikan total ukuran paper yang dibutuhkan dalam cm2
}

```

```

function totalPaperNeeded(totalPaperNeededSize: integer) -> integer
{
    I.S. totalPaperNeededSize adalah total ukuran paper yang dibutuhkan
    F.S. mengembalikan jumlah paper yang dibutuhkan dalam satuan lembar
}

```

```
function needMore(totalPaperNeededSize: integer) -> boolean
{
  I.S. totalPaperNeededSize adalah total ukuran paper yang dibutuhkan
  F.S. Mengembalikan true jika masih ada sisa ukuran paper yang belum tercover
}
```

```
procedure updateTotalPaperNeeded(in/out totalPaperNeeded : integer, needMore: boolean)
{
  I.S. totalPaperNeeded adalah jumlah lembar paper yang dibutuhkan, needMore adalah boolean
  F.S. Jika needMore bernilai true, totalPaperNeeded bertambah satu
}
```

```
function totalCost(totalPaperNeeded, highLevel, entryLevel: integer) -> integer
{
  I.S. totalPaperNeeded adalah jumlah lembar paper yang dibutuhkan, highLevel dan entryLevel
  adalah jumlah pegawai di setiap level
  F.S. mengembalikan total biaya yang dibutuhkan untuk seluruh paper dan parcel yang merupakan
  akumulasi dari paper yang dibutuhkan dan biaya bigParcel serta smallParcel
}
```