Nama : Abdillah Mufki Auzan Mubin

NPM : 40621100046

**ALGORITMA DAN PEMROGRAMAN I**

Tugas Pendahuluan V

1. Buat algoritma untuk penghitungan gaji karyawan berikut, sehingga bisa menangani lebih dari 1 karyawan tanpa perlu keluar dari program. Buat 2 versi algoritma menggunakan looping WHILE- DO dan REPEAT-UNTIL!.

Sebuah perusahaan meminta anda sebagai programmer yang sudah sangat terkenal untuk membuatkan program penghitungan gaji karyawan di perusahaan tersebut, dengan ketentuan:

* Gaji Pokok berdasarkan golongan

|  |  |
| --- | --- |
| Golongan | Gaji Pokok |
| 1 | 500 |
| 2 | 1000 |
| 3 | 1500 |
| 4 | 2000 |

* Bonus diberikan kepada pegawai berdasarkan masa kerja

|  |  |
| --- | --- |
| Masa Kerja | Bonus |
| 3 | 50 |
| 6 | 75 |
| >7 | 100 |

* Jika sudah menikah maka tunjangan keluarganya sebesar 10% dari gaji pokok
* Jika sudah menikah dan mempunyai anak (maksimal 3 orang) akan mendapat tunjangan anak sebesar 12.5% per anak Rumus perhitungan gajinya menjadi: Gaji = Gaji Pokok + Bonus + tunjangan keluarga + tunjangan anak

**Solusi:**

|  |  |
| --- | --- |
| **Algoritma WHILE-DO** | **Algoritma REPEAT-UNTIL** |
| Program hitung\_gaji | Program hitung\_gaji |
| Kamus Data:  sts, lanjut : char gpokok : integer  gol, masa\_kerja, bonus, janak :integer t\_keluarga, t\_anak, thp: real | Kamus Data:  sts, lanjut : char gpokok : integer  gol, masa\_kerja, bonus, janak :integer t\_keluarga, t\_anak, thp: real |
| Algoritma:  lanjut ‘Y’  WHILE (lanjut!=’N’) OR (lanjut!=’n’) DO INPUT(gol)  INPUT(masa\_kerja) INPUT(sts)  CASE (gol) of   1. : gpokok 500 2. : gpokok 1000 3. : gpokok 1500   4 : gpokok 2000  ENDCASE  IF masa\_kerja <= 3 THEN bonus 50  ELSEIF (masa\_kerja>= 3) AND (masa\_kerja<= 7) THEN  bonus 75  ELSEIF masa\_kerja >7 THEN bonus 100  ENDIF  CASE (sts) of  ‘Y’, ‘y’ : t\_keluarga 0.1\*gpokok INPUT(janak)  IF janak > 3 THEN Janak  ENDIF  t\_anak 0.125\*janak\*gpokok  ENDCASE  thp gpokok+bonus+t\_keluarga+t\_anak OUTPUT(thp)  OUTPUT(‘Lanjut Program(Y/N)’) INPUT(lanjut)  END WHILE | Algoritma:  lanjut ‘Y’ REPEAT  INPUT(gol) INPUT(masa\_kerja) INPUT(sts)  CASE (gol) of  1 : gpokok 500  2 : gpokok 1000  3 : gpokok 1500  4 : gpokok 2000  ENDCASE  IF masa\_kerja <= 3 THEN bonus 50  ELSEIF (masa\_kerja>= 3) AND (masa\_kerja<= 7) THEN  bonus 75  ELSEIF masa\_kerja >7 THEN bonus 100  ENDIF  CASE (sts) of  ‘Y’, ‘y’ : t\_keluarga 0.1\*gpokok INPUT(janak)  IF janak > 3 THEN Janak  ENDIF  t\_anak 0.125\*janak\*gpokok  ENDCASE  thp gpokok+bonus+t\_keluarga+t\_anak OUTPUT(thp)  OUTPUT(‘Lanjut Program(Y/N)’) INPUT(lanjut)  UNTIL (lanjut=’N’) OR (lanjut=’n’) |

2. Buat algoritma untuk kalkulator keliling bidang berikut, sehingga bisa menangani lebih dari 1 perhitungan tanpa perlu keluar dari program. Buat 2 versi algoritma menggunakan looping WHILE-DO dan REPEAT-UNTIL! Buatlah program untuk membuat kalkulator keliling bidang berdimensi dua, yaitu persegi, persegi panjang, dan lingkaran. Input yang digunakan bergantung kebutuhan menu yang dipilih

|  |  |
| --- | --- |
| INPUT / OUTPUT | Kalkulator Keliling   1. Keliling Persegi 2. Keliling Persegi Panjang 3. Keliling Lingkaran 4. Keluar   Masukkan nomor menu : 1 Menu Hitung Keliling Persegi Masukkan sisi : 5  Keliling persegi : 20 Kalkulator Keliling   1. Keliling Persegi 2. Keliling Persegi Panjang 3. Keliling Lingkaran 4. Keluar   Masukkan nomor menu : 3 Menu Hitung Keliling Lingkaran Masukkan jari-jari : 7  Keliling lingkaran : 44 |

# Solusi:

|  |  |
| --- | --- |
| **Algoritma WHILE-DO** | **Algoritma REPEAT-UNTIL** |
| Program keliling\_bangunan | Program keliling\_bangunan |
| Kamus Data: menu : integer  s, p, l, r, keliling : real const  pie = 3.14 | Kamus Data: menu : integer  s, p, l, r, keliling : real const  pie = 3.14 |
| Algoritma:  WHILE menu!=4 DO  INPUT(menu) CASE (menu) of  1 : INPUT(s)  keliling s\*4 2 : INPUT(p)  INPUT(l)  keliling (2\*p)+(2\*l) 3 : INPUT(r)  keliling 2\*pie\*r 4 : OUTPUT(‘Keluar...’)  ENDCASE  OUTPUT(keliling) END WHILE | Algoritma:  REPEAT  INPUT(menu) CASE (menu) of  1 : INPUT(s)  keliling s\*4 2 : INPUT(p)  INPUT(l)  keliling (2\*p)+(2\*l) 3 : INPUT(r)  keliling 2\*pie\*r 4 : OUTPUT(‘Keluar...’)  ENDCASE  OUTPUT(keliling) UNTIL menu=4 |

3. Buatlah algoritma untuk menghitung dan menampilkan deret berikut: Sum = 1 + 5 + 25 + 125 + 625 + ...

Input dari user: N yaitu jumlah suku. Buat 2 versi jawaban dengan menggunakan skema looping FOR-DO dan WHILE-DO!

Contoh:

|  |
| --- |
| Input : N = 3  Sum = 1 + 5 + 25 = 31 |
| Input : N = 4  Sum = 1 + 5 + 25 + 125 = 156 |

Solusi:

|  |  |
| --- | --- |
| Algoritma FOR-DO | Algoritma WHILE-DO |
| Program deret | Program deret |
| Kamus Data: i, n, ta, na : integer | Kamus Data: i, n, ta, na : integer |
| Algoritma:  na 1  ta 0  INPUT(n) OUTPUT(‘Sum = ’) FOR i 1 TO n DO OUPUT(na, ‘ + ’)  ta ta + na na na\*5 END FOR OUTPUT(‘=’, ta) | Algoritma:  i 1  na 1  ta 0 INPUT(n) OUTPUT(‘Sum = ’) WHILE i <= n DO OUPUT(na, ‘ + ’)  ta ta + na na na\*5  i i + 1 END WHILE OUTPUT(‘=’, ta) |

4. Buatlah algoritma untuk menghitung dan menampilkan deret berikut: Sum = 5 + 10 + 15 + 20 + 25 + ...

Input dari user: N yaitu jumlah suku. Buat 2 versi jawaban dengan menggunakan skema looping FOR-DO dan REPEAT-UNTIL!

Contoh:

|  |
| --- |
| Input : N = 3  Sum = 5 + 10 + 15 = 30 |
| Input : N = 5  Sum = 5 + 10 + 15 + 20 + 25 = 75 |

Solusi :

|  |  |
| --- | --- |
| **Algoritma FOR-DO** | **Algoritma WHILE-DO** |
| Program deret | Program deret |
| Kamus Data:  i, n, ta, na : integer | Kamus Data:  i, n, ta, na : integer |
| Algoritma:  na 5  ta 0 INPUT(n)  OUTPUT(‘Sum = ’)  FOR i 1 TO n DO  OUPUT(na, ‘ + ’)  ta ta + na na na+5  END FOR OUTPUT(‘=’, ta) | Algoritma:  i 1  na 5  ta 0 INPUT(n)  OUTPUT(‘Sum = ’) REPEAT  OUPUT(na, ‘ + ’)  ta ta + na na na+5  i i + 1 UNTIL i > n OUTPUT(‘=’, ta) |

5. Buatlah algoritma untuk menampilkan deret berikut:

1 1 1 2 2 2 3 3 3 4 4 4 5 5 5

Input dari user: N yaitu banyaknya angka, M yaitu banyaknya pengulangan. Petunjuk: Gunakan skema looping bersarang.

Contoh :

|  |
| --- |
| Input : N = 3, M = 2 1 1 2 2 3 3 |
| Input : N = 4, M = 5  1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 |

Solusi :

|  |  |
| --- | --- |
| **Algoritma FOR** | **Algoritma WHILE** |
| Program deret | Program deret |
| Kamus Data:  m, n, i, j : integer | Kamus Data:  m, n, i, j : integer |
| Algoritma: INPUT(n) INPUT(m)  FOR i 1 to n DO FOR j 1 to m DO  OUPUT(i) ENDFOR  ENDFOR | Algoritma: INPUT(n) INPUT(m)  i 1 j 1  FOR i to n DO j 1  FOR j to m DO OUPUT(i)  j j+1 ENDFOR  i i+1 ENDFOR |