NAMA : Alfi Syahrin

NIM : 2021573010042

KELAS : TI – 3C

TUGAS BAB I

TEST FORMATIF

a. Algoritma adalah urutan langkah-langkah sistematis yang digunakan untuk menyelesaikan masalah atau melakukan tugas tertentu, algoritma sering digunakan dalam pemrograman komputer untuk mengotomatiskan tugas tertentu atau menjalankan perhitungan matematika. Dan dasar dari pemrograman ilmu komputer secara umum.

b. - Struktur Urutan (Squence Proses) Struktur urut adalah struktur yang digunakan untuk mengerjakan jenis program yang pernyataannya sequential atau berurutan. Pada struktur ini, perintah yang diberikan secara beruntun atau berurutan baris per baris mulai dari awal hingga

   akhir. Struktur urut tidak memuat lompatan atau pengulangan didalamnya

   - Struktur Pemilihan (Selection Proses) Struktur Pemilihan adalah struktur yang digunakan pada program yang memerlukan proses pengujian kondisi untuk mengambil suatu keputusan apakah suatu baris perintah akan diproses atau tidak. Pengujian kondisi ini dilakukan untuk memilih salah satu dari beberapa alternatif yang tersedia. Tidak semua baris program akan dikerjakan pada struktur ini, melainkan hanya baris yang memenuhi syarat saja.
   - Struktur Pengulangan (Repetation Proses) Struktur Pengulangan adalah struktur yang melakukan pengulangan beberapa kali terhadap satu baris atau satu blok baris program. Pengulangan akan dilakukan sesuai dengan persyaratan yang diberikan. Beberapa intruksi atau perintah yang dapat digunakan untuk membuat program dengan struktur pengulangan, antara lain perintah for, while, atau do while. Perintah ini digunakan sesuai dengan jenis struktur pengulangan yang akan dibuat.

c. - Efisiensi > mengacu pada seberapa cepat dan sedikit sumber daya yang diperlukan algoritma
   - Ketepatan > algoritma harus memberikan hasil yang benar dan akurat sesuai tujuan
   - Keterbacaan > algoritma harus mudah dimengerti
   - Keandalan > algoritma harus dapat diandalkan dalam berbagai situasi
   - Kemudahan pemeliharaan > mudah dimodifikasi atau diperbaiki
   - Stabilitas > mampu menangani volume data yang berbeda
   - Kebenaran matematis > memberikan hasil yang benar dalam semua kasus
   - Kemudahan implementasi dan pemahaman

d. - Inisialisasi variable max dengan nilai negative tak terhingga
   - Inisialisasi variabel min dengan nilai positif tak terhingga
   - Masukkan angka yang diuji
   - Jika angka lebih besar dari nilai max, perbarui max dengan angka tersebut
     Jika angka lebih kecil dari nilai min, perbarui min
   - Setelah angka diuji nilai max dan min akan berisi nilai max dan min

e. Kebenaran (Correctness) :
   - Pemecahan Masalah yang Akurat > Algoritma harus memberikan hasil yang benar atau akurat sesuai dengan spesifikasi masalah yang sedang diselesaikan. Kesalahan dalam pemecahan masalah dapat mengakibatkan kerugian finansial, bahaya bagi keamanan, atau masalah lainnya
   - Keandalan > Algoritma yang salah dapat menyebabkan dampak serius, terutama dalam konteks sistem kritis seperti perangkat medis atau kendaraan otonom. Oleh karena itu, mengonfirmasi kebenaran algoritma adalah kunci untuk mencegah kesalahan yang berpotensi berbahaya.
   - Pemeliharaan yang Efisien > Algoritma yang benar lebih mudah dipelihara karena Anda memiliki pemahaman yang jelas tentang bagaimana algoritma seharusnya berperilaku. Pemeliharaan yang baik adalah bagian penting dari pengembangan perangkat lunak jangka panjang.

Efisiensi :

- Penggunaan Sumber Daya yang Efisien Algoritma yang efisien dapat menghemat waktu komputasi, memori, dan sumber daya lainnya. Ini menjadi sangat penting ketika bekerja dengan data besar atau dalam lingkungan dengan keterbatasan sumber daya, seperti perangkat mobile atau sistem tertanam.
- Peningkatan Kinerja > Algoritma yang efisien dapat meningkatkan kinerja sistem secara keseluruhan. Ini berarti aplikasi akan berjalan lebih cepat dan merespons lebih baik terhadap permintaan pengguna.
- Menghemat Biaya > Efisiensi algoritma juga dapat menghemat biaya, terutama dalam pengembangan perangkat lunak yang besar dan kompleks, di mana infrastruktur dan sumber daya dapat menjadi faktor biaya yang signifikan.

Maka pada saat menganalisis algoritma, kita sangat perlu mempertimbangkan keduanya. Seperti mengutamakan kebenaran hal yang paling penting, tapi begitu yakin algoritma telah benar, efisiensi juga perlu dipertimbangkan. Oleh karena itu, analisis algoritma mencakup pengujian kebenaran (testing) dan evaluasi efisiensi untuk memastikan keduanya seimbang dan sesuai dengan kebutuhan spesifik aplikasi atau masalah yang sedang diselesaikan.

f. 1. Masukkan Kartu ATM
2. Tempatkan kartu ATM Anda ke dalam slot kartu ATM yang sesuai
3. Masukkan PIN menggunakan keypad ATM.
4. Pilih bahasa yang diinginkan (hanya tersedia Bahasa Inggris & Bahasa Indonesia)
5. Pilih jenis transaksi yang diinginkan, dalam hal ini "Penarikan Tunai"
6. Masukkan jumlah uang yang ingin ditarik dari akun kita
7. Pilih jenis akun yang akan digunakan untuk penarikan ("Tabungan" atau "Giro")
8. ATM akan memeriksa terlebih dahulu apakah PIN yang dimasukkan sesuai dengan kartu yang kita masukkan dan apakah saldo mencukupi.

9. ATM akan menampilkan jumlah uang yang akan ditarik. Pastikan jumlahnya benar

sebelum melanjutkan.

10. Jika jumlah dan informasi sudah benar, pilih "Ya" atau "OK" untuk menarik uang.
11. ATM akan mengeluarkan uang dalam denominasi yang sesuai, dan juga akan mencetak struk transaksi.
12. Ambil uang yang telah ditarik beserta struknya, dan pastikan untuk menutup sesi dengan benar dan pastikan kita telah mengambil semua uang dan kartu ATM sebelum meninggalkan mesin ATM.

g.

```
function hitungNilai(skor) {
  if (skor > 90) return 'A';
  else if (skor >= 86 && skor <= 90) return 'A-';
  else if (skor >= 81 && skor <= 85) return 'B+';
  else if (skor >= 76 && skor <= 80) return 'B';
  else if (skor >= 71 && skor <= 75) return 'B-';
  else if (skor >= 66 && skor <= 70) return 'C+';
  else if (skor >= 61 && skor <= 65) return 'C';
  else if (skor >= 56 && skor <= 60) return 'C-';
  else if (skor >= 46 && skor <= 55) return 'D';
  else return 'E';
}
const nilai = hitungNilai(85);
console.log(`Nilai Anda ${nilai}`);
```

```
Nilai Anda B+                                    g_nilai.js:15
```

h.

```
function hitungR(nim) {
  var r1 = (2 / 5) % nim;
  var r2 = 3 / 7;
  var r3 = 3;

  return 1 / (1 / r1 + 1 / r2 + 1 / r3);
}
console.log(`Nilai R adalah ${hitungR(42)}`);
```

```
Nilai R adalah 0.1935483870967742
```

i. Algoritma

1. Membaca input gaji pokok, jumlah anak, masa kerja (dalam tahun), dan masuk kerja (dalam hari).

2. Menghitung tunjangan istri/suami = 10% dari gaji pokok.

3. Menghitung tunjangan anak = 5% dari gaji pokok untuk setiap anak.

4. Menghitung THR = Rp 5000 kali masa kerja (tahun).

5. Menghitung pajak = 15% dari total gaji pokok, tunjangan istri/suami, dan tunjangan anak.

6. Menghitung bantuan transport = Rp 3000 kali masuk kerja (hari).

7. Menghitung polis asuransi = Rp 20000.

8. Menghitung total pendapatan bulanan = (gaji pokok + tunjangan istri/suami + tunjangan anak + THR

+ bantuan transport) - (pajak + polis asuransi).

9. Menampilkan total pendapatan bulanan.

Program

```python
# Program
# Membaca input dari pengguna
gaji_pokok = float(input("Masukkan gaji pokok: "))
jumlah_anak = int(input("Masukkan jumlah anak: "))
masa_kerja_tahun = int(input("Masukkan masa kerja (dalam tahun): "))
masuk_kerja_hari = int(input("Masukkan jumlah masuk kerja (dalam hari): "))
# Menghitung tunjangan istri/suami
tunjangan_istri_suami = 0.1 * gaji_pokok
# Menghitung tunjangan anak
tunjangan_anak = 0.05 * gaji_pokok * jumlah_anak
# Menghitung THR
THR = 5000 * masa_kerja_tahun
# Menghitung pajak
total_pendapatan_sebelum_pajak = gaji_pokok + tunjangan_istri_suami + tunjangan_anak + THR
pajak = 0.15 * total_pendapatan_sebelum_pajak
# Menghitung bantuan transport
bantuan_transport = 3000 * masuk_kerja_hari
# Menghitung polis asuransi
polis_asuransi = 20000
# Menghitung total pendapatan bulanan
total_pendapatan_bulanan = (gaji_pokok + tunjangan_istri_suami + tunjangan_anak + THR + bantuan_transport)
# Menampilkan total pendapatan bulanan
print(f"Total pendapatan bulanan adalah: Rp {total_pendapatan_bulanan:.2f}")
```

```
Masukkan gaji pokok: 100.000
Masukkan jumlah anak: 5
Masukkan masa kerja (dalam tahun): 300
Masukkan jumlah masuk kerja (dalam hari): 5
Total pendapatan bulanan adalah: Rp 1270114.75
```

j.

```javascript
function floorAkar(n) {
  let i = 1;
  while (i * i < n) {
    i++;
  }
  if (n % i === 0) {
    console.log(i);
  } else {
    console.log(i - 1);
  }
}
const n = 25;
floorAkar(n);
```

```
5
```

k.

```
function cariJarakTerdekat(arr) {
  let dmin = Infinity;

  for (let i = 0; i < arr.length; i++) {
    for (let j = i + 1; j < arr.length; j++) {
      const currentDistance = Math.abs(arr[i] - arr[j]);
      if (currentDistance === 0) {
        return 0;
      } else if (currentDistance < dmin) {
        dmin = currentDistance;
      }
    }
  }
  return dmin;
}
const array = [1, 5, 3, 19, 18, 25, 7];
const jarakTerdekat = cariJarakTerdekat(array);
console.log('Jarak terdekat antara dua elemen:', jarakTerdekat);
```

```
Jarak terdekat antara dua elemen: 1
```