

MODUL AJAR KURIKULUM MERDEKA
SMKN 1 Narmada — SEMESTER GANJIL
Modul Ajar: Menganalisis konsep dasar pemrograman terstruktur

I. INFORMASI UMUM & KOMPONEN INTI

Nama Penyusun	Candra Rusmana, S.T., M.Kom. (SMKN 1 Narmada)
Mata Pelajaran & Kelas	Kompetensi Keahlian Rekayasa Perangkat Lunak (XI RPL)
Elemen CP	pemrograman terstruktur dasar pemodelan perangkat lunak, dan pemrograman antar muka grafis; — Pemrograman berbasis teks, grafis, dan multimedia Menerapkan perintah eksekusi bahasa pemrograman yang mengimplementasikan pemrograman terstruktur, pemrograman berorientasi objek lanjutan, pemodelan perangkat lunak, pemrograman antarmuka GUI (Graphical User Interface), dan pemanfaatan pustaka (library).
Tujuan Pembelajaran (TP)	Menganalisis konsep dasar pemrograman terstruktur
Target Peserta Didik	Peserta didik reguler/tipikal (36 Siswa)
Sarana & Prasarana	Komputer dengan perangkat lunak pemrograman, proyektor, whiteboard, dan akses internet
Pemahaman Bermakna	Peserta didik dapat memahami dan menganalisis konsep dasar pemrograman terstruktur, termasuk pemodelan perangkat lunak dan pemrograman antar muka grafis, dengan memanfaatkan pengetahuan dari matematika, logika, dan komunikasi
Pertanyaan Pemantik	1. Apa yang dimaksud dengan pemrograman terstruktur dan bagaimana penerapannya dalam pengembangan perangkat lunak? 2. Bagaimana cara menerapkan logika dan matematika dalam pemrograman terstruktur?

MATERI PEMBELAJARAN

Tujuan Pembelajaran (TP): Menganalisis konsep dasar pemrograman terstruktur

A. Pendahuluan: Apa itu Pemrograman Terstruktur?

Pemrograman terstruktur adalah sebuah paradigma (sudut pandang) dalam rekayasa perangkat lunak yang berfokus pada penulisan kode yang jelas, terorganisasi, dan mudah

dipahami. Paradigma ini lahir pada akhir 1960-an sebagai solusi atas masalah "Spaghetti Code"—yaitu kode program yang alurnya melompat-lompat secara tidak beraturan menggunakan perintah GOTO, sehingga sangat sulit dilacak jika terjadi *bug* atau kesalahan.

Prinsip utama dari pemrograman terstruktur adalah memecah masalah yang besar dan kompleks menjadi modul-modul atau fungsi-fungsi yang lebih kecil (pendekatan *Top-Down*), serta mengatur alur logika program menggunakan tiga struktur kontrol dasar.

B. Tiga Struktur Kontrol Dasar

Untuk menganalisis dan membangun program terstruktur, kita hanya membutuhkan tiga blok bangunan dasar. Semua algoritma kompleks di dunia dapat disederhanakan menjadi kombinasi dari ketiga struktur ini:

1. Struktur Runtunan (*Sequence*)

Ini adalah struktur alur kerja yang paling dasar. Instruksi dieksekusi secara berurutan, baris demi baris, dari atas ke bawah. Instruksi kedua tidak akan dieksekusi sebelum instruksi pertama selesai.

- **Karakteristik:** Tidak ada percabangan, tidak ada pengulangan.
- **Analisis Penggunaan:** Digunakan untuk inisialisasi variabel, proses perhitungan matematika beruntun, atau pertukaran nilai data.

2. Struktur Percabangan (*Selection / Decision*)

Struktur ini memungkinkan program untuk membuat keputusan berdasarkan kondisi tertentu (bernilai *True* atau *False*). Alur eksekusi program dapat berbelok arah tergantung pada hasil evaluasi kondisi tersebut.

- **Karakteristik:** Menggunakan kata kunci seperti IF, IF-ELSE, ELIF, atau SWITCH-CASE.
- **Analisis Penggunaan:** Digunakan saat program harus merespons input yang berbeda, seperti menentukan kelulusan siswa berdasarkan nilai, atau memvalidasi hak akses login.

3. Struktur Perulangan (*Iteration / Looping*)

Struktur yang menginstruksikan program untuk mengeksekusi satu atau sekumpulan blok kode secara berulang-ulang selama kondisi tertentu masih terpenuhi.

- **Karakteristik:** Menggunakan kata kunci seperti FOR, WHILE, atau DO-WHILE.

- **Analisis Penggunaan:** Sangat krusial untuk memproses data dalam jumlah besar (seperti membaca isi *database*) tanpa harus menulis ulang kode yang sama.

C. Analisis Kelebihan Pemrograman Terstruktur

Mengapa kita harus meninggalkan gaya penulisan kode bebas dan beralih ke pemrograman terstruktur? Berikut adalah hasil analisis keuntungannya:

1. **Keterbacaan (*Readability*):** Kode dibaca dari atas ke bawah secara logis. Hal ini memudahkan *programmer* lain untuk memahami alur berpikir penulis kode aslinya.
2. **Kemudahan *Debugging* & Pemeliharaan:** Karena alurnya jelas dan dibagi menjadi blok/modul (fungsi), melacak sumber *error* menjadi jauh lebih cepat.
3. **Efisiensi Waktu:** Modul kode yang sudah ditulis dapat digunakan kembali (*reusability*) tanpa harus menulis ulang.

D. Contoh Implementasi dalam Bahasa Pemrograman

Untuk memudahkan analisis, mari kita lihat bagaimana ketiga struktur dasar ini diterapkan dalam bahasa pemrograman yang sering digunakan dalam industri pengembangan web dan *software*, seperti **Python** dan **JavaScript**.

1. Contoh dalam Python Berikut adalah program sederhana untuk mencetak angka genap dari 1 hingga 10, yang menggabungkan struktur runtunan, perulangan, dan percabangan:

```
# 1. Runtunan (Deklarasi variabel batas)
batas_maksimal = 10
print("Deret Angka Genap:")

# 2. Perulangan (Loop dari 1 sampai 10)
for angka in range(1, batas_maksimal + 1):

    # 3. Percabangan (Cek apakah habis dibagi 2)
    if angka % 2 == 0:
        print(angka)
```

2. Contoh dalam JavaScript Logika yang sama persis jika dianalisis dan diterjemahkan ke dalam JavaScript untuk aplikasi berbasis web:

```
// 1. Runtunan
let batasMaksimal = 10;
console.log("Deret Angka Genap:");

// 2. Perulangan
for (let angka = 1; angka <= batasMaksimal; angka++) {
```

```

        // 3. Percabangan
        if (angka % 2 == 0) {
            console.log(angka);
        }
    }
}

```

Dalam bahasa C#, kita perlu mendefinisikan tipe data secara ketat (seperti int untuk bilangan bulat) dan memasukkannya ke dalam struktur *Class* dan *Method* utama (Main).

```
using System;
```

```

class Program
{
    static void Main()
    {
        // 1. RUNTUNAN (Sequence): Deklarasi variabel dan nilai awal
        int batasMaksimal = 10;
        Console.WriteLine("Deret Angka Genap:");

        // 2. PERULANGAN (Iteration): Mengulang dari 1 sampai 10
        for (int angka = 1; angka <= batasMaksimal; angka++)
        {
            // 3. PERCABANGAN (Selection): Mengecek apakah angka habis dibagi
            if (angka % 2 == 0)
            {
                Console.WriteLine(angka);
            }
        }
    }
}

```

Contoh kasus Diskon Belanja

```
using System;
```

```

class DiskonToko
{
    static void Main()
    {
        // Runtunan: Menyiapkan variabel
        double totalBelanja = 600000; // Contoh input belanja
        double totalBayar;

        Console.WriteLine("Total belanja Anda: Rp " + totalBelanja);

        // Percabangan: Cek syarat diskon
        if (totalBelanja > 500000)
        {
            // Jika lebih dari 500.000, hitung diskon 10%
            double diskon = totalBelanja * 0.10;

```

```
        totalBayar = totalBelanja - diskon;
        Console.WriteLine("Selamat! Anda mendapatkan diskon 10%.");
    }
    else
    {
        // Jika 500.000 atau kurang, tidak ada diskon
        totalBayar = totalBelanja;
        Console.WriteLine("Anda tidak mendapatkan diskon.");
    }

    // Runtunan: Menampilkan hasil akhir
    Console.WriteLine("Total yang harus dibayar: Rp " + totalBayar);
}
}
```

LEMBAR KERJA PESERTA DIDIK (LKPD)

Mata Pelajaran: Dasar-Dasar Pemrograman **Topik:** Analisis Konsep Dasar Pemrograman

Terstruktur **Waktu:** 45 Menit

IDENTITAS PESERTA DIDIK

- Nama Kelompok / Individu :
- Kelas :
- No. Absen :
- Hari, Tanggal :

A. TUJUAN PEMBELAJARAN

Melalui penyelesaian LKPD ini, peserta didik diharapkan mampu menganalisis dan mengidentifikasi penerapan struktur runtunan (*sequence*), percabangan (*selection*), dan perulangan (*iteration*) dalam algoritma serta kehidupan nyata.

B. PETUNJUK Pengerjaan

1. Bacalah setiap studi kasus dengan teliti.
2. Diskusikan dengan teman sebangku/kelompok Anda (jika dikerjakan berkelompok).
3. Tuliskan hasil analisis Anda langsung pada kolom yang telah disediakan.
4. Tanyakan kepada guru jika ada instruksi yang kurang jelas.

AKTIVITAS 1: Menganalisis Logika Aplikasi Sehari-hari

Studi Kasus: Bayangkan Anda sedang menggunakan aplikasi dompet digital (seperti GoPay atau mobile banking) untuk membeli token listrik (PLN). Proses yang terjadi di dalam sistem aplikasi tersebut kurang lebih seperti ini:

1. Pengguna memasukkan ID Pelanggan PLN.
2. Sistem mengecek apakah ID Pelanggan tersebut valid atau tidak. Jika valid, sistem menampilkan nama pelanggan. Jika tidak valid, sistem memunculkan pesan *error*.
3. Pengguna memilih nominal token yang ingin dibeli (misal: Rp50.000).
4. Pengguna memasukkan PIN keamanan.
5. Sistem mengecek PIN. Jika salah, sistem akan meminta pengguna memasukkan PIN kembali (maksimal 3 kali percobaan sebelum akun diblokir).
6. Sistem memotong saldo dan menampilkan kode token listrik.

Tugas: Dari 6 langkah di atas, identifikasilah mana bagian yang merupakan *Sequence* (Runtunan), *Selection* (Percabangan), dan *Iteration* (Perulangan). Jelaskan alasan Anda!

- **Runtunan (Sequence):**

.....
.....
Alasan:

- **Percabangan (Selection):**

.....

.....
Alasan:

- **Perulangan (Iteration):**

.....
.....
Alasan:

AKTIVITAS 2: Bedah *Pseudocode* (Kode Semu)

Tugas: Di bawah ini terdapat sebuah algoritma sederhana untuk sistem penyortiran barang di gudang. Analisislah baris kode tersebut dengan menarik garis atau memberikan keterangan di sebelahnya: mana blok Runtunan, mana blok Percabangan, dan mana blok Perulangan.

Plaintext

1. MULAI
2. Siapkan keranjang kosong
3. Jumlah_barang = 100
- 4.
5. SELAMA (Jumlah_barang > 0) LAKUKAN:
6. Ambil satu barang dari kotak masuk
- 7.
8. JIKA (Kondisi_barang == "Rusak") MAKA:
9. Buang barang ke tempat sampah
10. JIKA TIDAK MAKA:
11. Masukkan barang ke keranjang
- 12.
13. Jumlah_barang = Jumlah_barang - 1
14. SELESAI SELAMA
- 15.
16. Cetak "Penyortiran Selesai"
17. SELESAI

Hasil Analisis:

1. Baris nomor berapa yang menunjukkan **Perulangan**? Jawaban:
.....
2. Baris nomor berapa yang menunjukkan **Percabangan**? Jawaban:
.....
3. Apa yang akan terjadi pada sistem jika baris nomor 13 dihapus? (Analisis kritis) Jawaban:
.....
.....

AKTIVITAS 3: Merancang Solusi

Tugas: Sebuah toko alat elektronik sedang mengadakan diskon. Aturannya adalah:

- Jika total belanja pembeli lebih dari Rp500.000, maka mendapatkan diskon 10%.
- Jika total belanja kurang dari atau sama dengan Rp500.000, tidak mendapatkan diskon.

Berdasarkan struktur **Percabangan**, tuliskan urutan logika (algoritma/kalimat sederhana) untuk menghitung total yang harus dibayar oleh pembeli!

Jawaban (Rancangan Algoritma):

1.
2.
3.
4.
5.

NILAI & PARAF GURU (Diisi oleh guru) Nilai: Catatan Guru:

..... **Paraf:**

Kunci Jawaban Singkat untuk Guru (Pegangan Anda):

- **Aktivitas 1:**
 - *Sequence*: Langkah 1, 3, 4, 6 (Karena instruksi berjalan lurus ke bawah tanpa syarat).
 - *Selection*: Langkah 2 dan 5 awal (Karena ada kata "Jika valid/salah").
 - *Iteration*: Langkah 5 akhir (Meminta memasukkan PIN *kembali/diulang* maksimal 3 kali).
- **Aktivitas 2:**
 1. Baris 5 sampai 14 (Loop).
 2. Baris 8 sampai 11 (If-Else).
 3. Jika baris 13 dihapus, akan terjadi *Infinite Loop* (perulangan tanpa henti) karena Jumlah_barang selalu 100 dan tidak pernah berkurang, sehingga program akan *crash* atau *hang*.