

LEMBAR KERJA MURID (LKM)

BERPIKIR KOMPUTASIONAL: FLOWCHART & PSEUDOCODE

Satuan Pendidikan: SMK Wijaya Putra Surabaya | Mata Pelajaran: Informatika | Fase / Kelas
/Semester: E / X / Ganjil | Kode TP: X.BK.3 |Alokasi Waktu: 16 JP (4 Pertemuan × 4 JP)

IDENTITAS MANDIRI

Nama Lengkap :
Kelas / No. Absen : /
Konsentrasi Kehalian :

TUJUAN PEMBELAJARAN

Merancang representasi solusi menggunakan algoritma, flowchart, dan pseudocode secara logis, sistematis, dan efisien.

KEGIATAN 1: KONSEP ALGORITMA DAN SIMBOL FLOWCHART

A. Orientasi Masalah

Amati alur kerja berikut dan jawablah pertanyaan di bawahnya:

"Seorang murid ingin menyalakan komputer di laboratorium, tetapi layar monitor tidak menampilkan gambar (gelap). Langkah apa yang harus dilakukan secara terstruktur untuk mengecek masalah tersebut?"

B. Identifikasi Simbol Flowchart

Jodohkan nama simbol dan fungsi berikut dengan menarik garis atau menuliskan huruf yang tepat:

No	Nama Simbol	Fungsi Utama
1	<i>Terminal</i>	A. Mengambil keputusan/kondisi (<i>Decision</i>)
2	<i>Process</i>	B. Menandai awal (<i>Start</i>) atau akhir (<i>End</i>) proses
3	<i>Input/Output</i>	C. Menjalankan proses perhitungan atau tindakan
4	<i>Decision</i>	D. Menerima masukan data atau menampilkan hasil

C. Latihan Menyusun Flowchart Sederhana

Gambarkan *flowchart* sederhana untuk proses **"Menyalakan Komputer di Laboratorium"** berdasarkan langkah logis!

(Gunakan area di bawah ini untuk menggambar)

[Area Gambar Flowchart]

KEGIATAN 2: MENYUSUN FLOWCHART BERDASARKAN STUDI KASUS

A. Studi Kasus

Studi Kasus: System Transaksi Kasir Sederhana

"Toko Sederhana memberikan diskon sebesar 10% jika total belanjaan pelanggan melebihi Rp100.000. Jika total belanjaan kurang dari atau sama dengan Rp100.000, pelanggan tidak mendapatkan diskon. Susunlah alur logika untuk menghitung total bayar yang harus dibayar pelanggan!"

B. Analisis Urutan Proses

1. **Input:**

.....

2. **Kondisi/Keputusan:**

.....

3. **Proses/Perhitungan:**

.....

4. **Output:**

.....

C. Desain Flowchart Studi Kasus

Buatlah *flowchart* lengkap dan terstruktur menggunakan simbol standar (dapat dikerjakan secara digital menggunakan draw.io / Canva atau digambar manual)!

[Lembar Kerja Gambar Flowchart]

KEGIATAN 3: MENYUSUN PSEUDOCODE SEBAGAI REPRESENTASI ALGORITMA

A. Konsep Pseudocode

Berdasarkan *flowchart* yang telah dibuat pada Pertemuan 2, terjemahkan setiap langkah menjadi bentuk **Pseudocode** dengan struktur penulisan yang benar (*Header, Declaration, Algorithm*)!

B. Lembar Kerja Penulisan Pseudocode

PROGRAM Hitung_Total_Bayar

DEKLARASI:

total_belanja : float
diskon : float
total_bayar : float

ALGORITMA:

READ total_belanja

IF total_belanja > 100000 THEN

 diskon <- total_belanja * 0.10

ELSE

 diskon <- 0

ENDIF

total_bayar <- total_belanja - diskon

WRITE total_bayar

(Sesuaikan pseudocode di atas dengan studi kasus yang dikerjakan)

KEGIATAN 4: INTEGRASI & PRESENTASI SOLUSI

A. Penggabungan & Verifikasi Algoritma

Lakukan pengujian manual (*Trace Table / Dry Run*) terhadap *flowchart* dan *pseudocode* yang telah dibuat untuk memastikan tidak ada kesalahan logika (*logic error*).

Uji Coba	Nilai Input (total_belanja)	Kondisi Terpenuhi?	Perhitungan Diskon	Total Bayar Sesuai? (Ya/Tidak)
Kasus 1	Rp150.000	Ya	Rp15.000	Rp135.000 (.....)
Kasus 2	Rp80.000	Tidak	Rp0	Rp80.000 (.....)

B. Lembar Refleksi Diri

Jawablah pertanyaan refleksi berikut:

1. Apakah *flowchart* dan *pseudocode* yang kamu buat sudah saling sesuai dan logis?
2. Kesulitan terbesar apa yang kalian hadapi saat mengubah *flowchart* menjadi *pseudocode*?

RUBRIK PENILAIAN RINGKAS (ASESMEN FORMATIF/SUMATIF)

Kriteria	Skor 4 (Sangat Baik)	Skor 3 (Baik)	Skor 2 (Cukup)	Skor 1 (Kurang)
Simbol Flowchart	Penggunaan simbol 100% sesuai standar baku.	Ada 1-2 kesalahan penggunaan simbol.	Penggunaan simbol kurang sesuai.	Simbol tidak sesuai standar.
Logika Algoritma	Alur logika sangat runtut, tepat, dan efisien.	Alur logika runtut dan tepat.	Terdapat kesalahan logika minor.	Alur logika tidak terstruktur.
Pseudocode	Penulisan terstruktur dan sesuai <i>flowchart</i> .	Penulisan terstruktur, ada sedikit perbedaan dengan <i>flowchart</i> .	Struktur <i>pseudocode</i> kurang tepat.	<i>Pseudocode</i> tidak sesuai dengan <i>flowchart</i> .

LEMBAR KERJA MURID (LKM) MANDIRI

TUGAS BESAR / PROYEK AKHIR SEMESTER: PERANCANGAN SISTEM POS TOKO MODERN

Satuan Pendidikan : SMK Wijaya Putra Surabaya
Mata Pelajaran : Informatika
Fase / Kelas / Semester : E / X / Ganjil
Kode TP : X.BK.3
Sifat Tugas : Mandiri (Individual)
Produk Akhir : Laporan Perancangan Algoritma Aplikasi POS Toko Modern

IDENTITAS MURID

- Nama Lengkap :
- Kelas / Nomer Absen : /
- Konsentrasi Keahlian :

A. DESKRIPSI TUGAS BESAR

Kamu ditunjuk sebagai seorang *Junior System Analyst* untuk merancang alur logika aplikasi **Point of Sale (POS) Toko Modern**. Aplikasi ini digunakan kasir untuk memproses transaksi penjualan barang, menghitung total bayar, diskon bertingkat, serta mencetak struk belanja sederhana.

B. SPESIFIKASI DAN KETENTUAN LOGIKA SISTEM

1. Input Data:

- Nama Barang, Harga Satuan, dan Jumlah Beli (*Quantity*).
- Status Keanggotaan Pelanggan (*Member / Non-Member*).
- Uang Pembayaran dari Pembeli.

2. Aturan Diskon & Transaksi:

- **Diskon Member:** Jika Pelanggan adalah *Member*, dapatkan diskon langsung sebesar **5%** dari subtotal.
- **Diskon Belanja (Bertingkat):**
 - Subtotal > Rp500.000 → Tambahan Diskon **10%**.
 - Subtotal Rp200.000 – Rp500.000 → Tambahan Diskon **5%**.
 - Subtotal < Rp200.000 → Tidak ada tambahan diskon belanja.
- **Total Bayar:** (Subtotal - Total Diskon).
- **Kembalian:** (Uang Pembayaran - Total Bayar).

3. Validasi Pembayaran:

- Jika Uang Pembayaran < Total Bayar, tampilkan pesan: "Uang Kurang, Transaksi Gagal".
- Jika Uang Pembayaran ≥ Total Bayar, tampilkan kembalian dan cetak struk transaksi.

C. TAHAP Pengerjaan pada LKM

TAHAP 1: ANALISIS VARIABEL DAN TIPE DATA

Isilah tabel variabel yang akan kamu gunakan dalam pembuatan algoritma ini:

Nama Variabel	Tipe Data	Keterangan / Fungsi
nama_barang	String	Menyimpan nama barang yang dibeli
harga_satuan	Real / Integer	Menyimpan harga per unit barang
jumlah_beli	Integer	Menyimpan kuantitas barang
is_member	Boolean / String	Menyimpan status member (Ya/Tidak)
Subtotal	Real / Float	Hasil dari harga_satuan $\times$ jumlah_beli
diskon_member	Real / Float	Besaran diskon member
diskon_belanja	Real / Float	Besaran diskon berdasarkan nominal belanja
total_diskon	Real / Float	diskon_member + diskon_belanja
total_bayar	Real / Float	Nominal akhir yang harus dibayar
uang_bayar	Real / Float	Nominal uang dari pelanggan
Kembalian	Real / Float	uang_bayar - total_bayar

TAHAP 2: DESAIN FLOWCHART SISTEM (MANDIRI)

Buatlah *flowchart* lengkap yang menggambarkan alur aplikasi POS Toko Modern di atas secara runtut dan menggunakan simbol baku!

(Gunakan aplikasi digital seperti draw.io / Canva / Lucidchart atau digambar rapi di lembar ini)

TAHAP 3: PENULISAN PSEUDOCODE (MANDIRI)

Tuliskan *pseudocode* terstruktur berdasarkan *flowchart* yang telah kamu buat di atas! Gunakan format penulisan baku (*Header, Declaration, Algorithm*).

PROGRAM Aplikasi_POS_Toko_Modern

DEKLARASI:

```
nama_barang      : string
harga_satuan, subtotal : real
jumlah_beli      : integer
status_member    : string
diskon_member, diskon_belanja : real
total_diskon, total_bayar : real
uang_bayar, kembalian : real
```

ALGORITMA:

```
// 1. Input Data Belanja
READ nama_barang, harga_satuan, jumlah_beli, status_member
```

```
// 2. Hitung Subtotal
subtotal <- harga_satuan * jumlah_beli
```

```
// 3. Cek Diskon Member
IF status_member = "Ya" OR status_member = "YA" THEN
    diskon_member <- subtotal * 0.05
ELSE
    diskon_member <- 0
ENDIF
```

```
// 4. Cek Diskon Belanja Bertingkat
IF subtotal > 500000 THEN
    diskon_belanja <- subtotal * 0.10
ELSE IF subtotal >= 200000 AND subtotal <= 500000 THEN
    diskon_belanja <- subtotal * 0.05
ELSE
    diskon_belanja <- 0
ENDIF
```

```
// 5. Hitung Total Bayar
total_diskon <- diskon_member + diskon_belanja
total_bayar <- subtotal - total_diskon
```

```
WRITE "Total Pembayaran: ", total_bayar
```

```
// 6. Input Pembayaran & Validasi Kembalian
READ uang_bayar
```

```
IF uang_bayar >= total_bayar THEN
    kembalian <- uang_bayar - total_bayar
    WRITE "Transaksi Berhasil! Kembalian: ", kembalian
ELSE
    WRITE "Uang Pembayaran Kurang! Transaksi Gagal."
ENDIF
```

TAHAP 4: TRACE TABLE / DRY RUN (UJI COBA LOGIKA MANDIRI)

Lakukan uji coba (*Trace Table*) mandiri menggunakan 2 skenario data berikut untuk memastikan tidak terjadi kesalahan logika:

Skenario 1 (Member, Belanja > Rp500.000):

harga_satuan = Rp150.000, jumlah_beli = 4, status_member = "Ya", uang_bayar = Rp600.000

Langkah / Variabel	Hasil Perhitungan / Nilai
Subtotal	Rp600.000
diskon_member (5%)	Rp30.000
diskon_belanja (10%)	Rp60.000
total_diskon	Rp90.000
total_bayar	Rp510.000
Status Pembayaran	Berhasil / Kembalian: Rp90.000

Skenario 2 (Non-Member, Belanja Rp150.000):

harga_satuan = Rp50.000, jumlah_beli = 3, status_member = "Tidak", uang_bayar = Rp150.000

Langkah / Variabel	Hasil Perhitungan / Nilai
Subtotal	
diskon_member	
diskon_belanja	
total_diskon	
total_bayar	
Status Pembayaran	

TAHAP 5: REFLEKSI MANDIRI

Jawablah pertanyaan refleksi secara jujur:

1. Apa bagian tersulit saat merancang algoritma percabangan bertingkat pada aplikasi POS ini?

o *Jawaban:*

.....
.....

2. Mengapa pengujian *Trace Table* penting dilakukan sebelum algoritma diubah menjadi kode program yang sebenarnya?

○ *Jawaban:*

.....
.....

RUBRIK PENILAIAN TUGAS BESAR MANDIRI

Kriteria	Skor 4 (Sangat Baik)	Skor 3 (Baik)	Skor 2 (Cukup)	Skor 1 (Kurang)
Kelengkapan Analisis Variabel	Variabel, tipe data, dan deskripsi dianalisis lengkap & tepat.	Variabel dan tipe data lengkap, deskripsi kurang detail.	Terdapat beberapa variabel yang hilang/salah tipe data.	Analisis variabel tidak lengkap.
Kebenaran Flowchart	Simbol 100% baku, percabangan runtut & teruji secara logika.	Simbol baku, ada kesalahan logika minor pada percabangan.	Penggunaan simbol kurang tepat/alur kurang jelas.	Flowchart tidak lengkap / tidak sesuai kriteria.
Kebenaran Pseudocode	Struktur penulisan sangat rapi, sinkron dengan <i>flowchart</i> .	Struktur penulisan rapi, ada sedikit ketidaksesuaian.	Format pseudocode kurang rapi & banyak kesalahan syntax.	Pseudocode tidak sesuai dengan aturan.
Akurasi Uji Coba (Trace Table)	Perhitungan pada kedua skenario 100% tepat & teliti.	Perhitungan tepat hanya pada 1 skenario.	Terdapat kesalahan perhitungan pada kedua skenario.	Trace table tidak diisi.