

UTS Algoritma & Struktur Data

Eri Tri Haryanto

Informatika

232102020

1. Kasus: Anda bekerja sebagai pengembang perangkat lunak di sebuah startup fintech. Salah satu fitur yang sedang dikembangkan adalah sistem pelunasan pinjaman dengan skema bunga menurun setiap bulan hingga lunas
 - a. Jelaskan bagaimana Anda bisa menggunakan perulangan while atau for untuk mensimulasikan pembayaran cicilan tiap bulan, di mana bunga berkurang setiap kali pembayaran dilakukan.

- Pokok pinjaman berkurang sesuai jumlah cicilan.
- Bunga dihitung dari sisa pinjaman dan berkurang karena sisa pokok makin kecil.
- Proses berlanjut hingga pinjaman lunas ($\text{sisa_pinjaman} \leq 0$).

- b. Psudocode dan flowchart

Input: pinjaman_awal, bunga_awal, cicilan_pokok

Set bulan = 0

Set sisa_pinjaman = pinjaman_awal

While sisa_pinjaman > 0:

 bunga = sisa_pinjaman * bunga_awal

 total_cicilan = cicilan_pokok + bunga

 sisa_pinjaman = sisa_pinjaman - cicilan_pokok

 bulan = bulan + 1

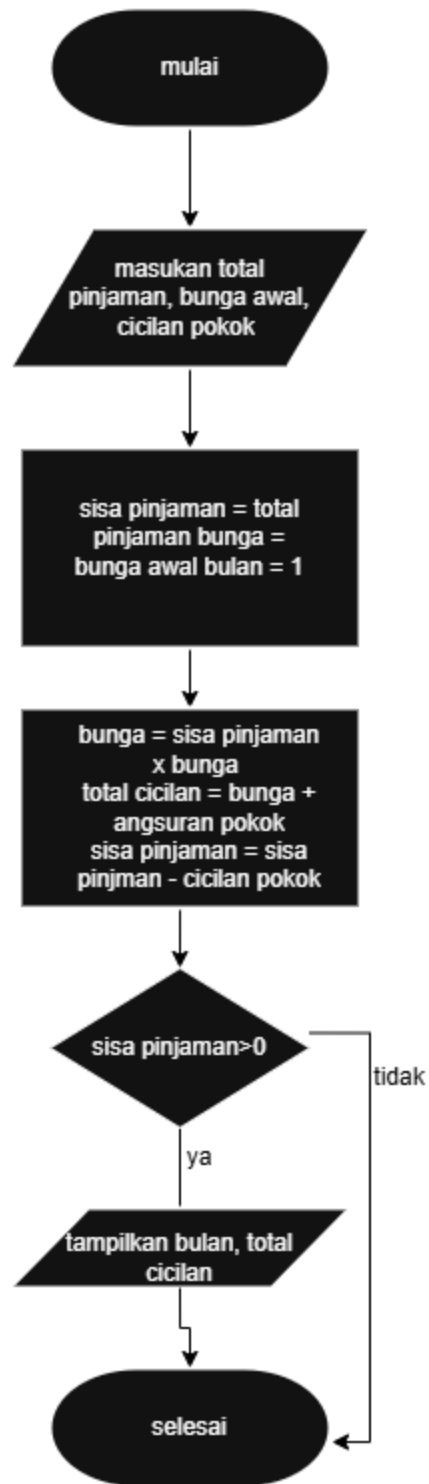
 Tampilkan("Bulan", bulan, "Cicilan:", total_cicilan)

Tampilkan("Pinjaman lunas dalam", bulan, "bulan")

Flowchart :

- c. Modifikasi algoritma

- Pinjaman tidak akan pernah lunas (bunga bertambah terus).
- Harus dicek setiap iterasi.



While sisa_pinjaman > 0:

bunga = sisa_pinjaman * bunga_awal

If cicilan_pokok <= bunga:

Tampilkan("Cicilan tidak mencukupi untuk menutup bunga. Perlu dinaikkan.")

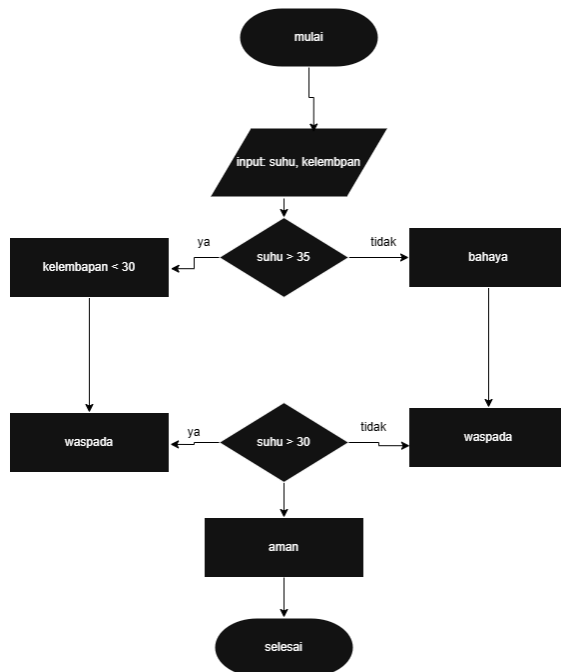
Break

2. Kasus: Anda sedang membangun sistem deteksi suhu untuk peringatan dini kebakaran hutan. Sistem ini harus menentukan level bahaya berdasarkan suhu dan kelembaban lingkungan.

- a. Buat algoritma menggunakan percabangan bersarang (nested if) untuk mengklasifikasikan kondisi sebagai "Aman", "Waspada", "Bahaya", atau "Darurat" berdasarkan suhu dan kelembaban yang diberikan.

- if suhu < 30:
 - if kelembaban > 60:
 - status = "Aman"
 - else:
 - status = "Waspada"
- elif suhu <= 40:
 - if 40 <= kelembaban <= 60:
 - status = "Waspada"
 - else:
 - status = "Bahaya"
- elif suhu <= 50:
 - if 20 <= kelembaban < 40:
 - status = "Bahaya"
 - else:
 - status = "Darurat"
- else:
 - status = "Darurat"

- b. Flowchart

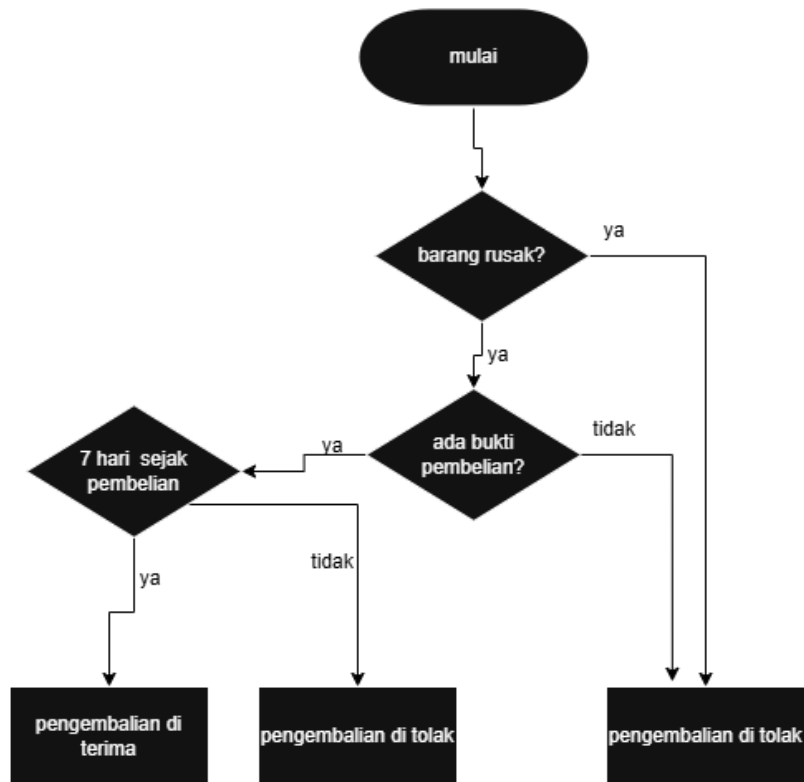


c. Jelaskan strategi yang dapat digunakan, seperti lookup table atau binary decision trees.

- Lookup table, Buat tabel dari kombinasi suhu & kelembaban → langsung akses status tanpa banyak if-else.
- Binary decision tree, Gunakan pohon keputusan yang meminimalkan jumlah komparasi.
- Paralelisasi, Jalankan proses klasifikasi secara paralel untuk setiap sensor
- Preprocessing, Sensor bisa mengirim data dalam batch → klasifikasi sekaligus (bulk processing)

3. Kasus: Sebuah perusahaan e-commerce ingin mengotomatiskan proses pengembalian barang. Algoritma yang dibuat harus memeriksa apakah barang rusak, apakah pembeli memiliki bukti pembelian, dan apakah pengembalian dilakukan dalam waktu yang ditentukan.

a. Flowchart



b. Pseudocode

barang = [1, 2, 3, 1, 4, 2, 5]

for i in range(len(barang)):

 kondisi = barang[i]

 if kondisi == 1:

 print(f"Barang ke-{i+1}: Diterima (Masih segel & tidak rusak)")

 elif kondisi == 2:

 print(f"Barang ke-{i+1}: Perlu Verifikasi (Tidak segel, tidak rusak)")

 elif kondisi == 3:

 print(f"Barang ke-{i+1}: Ditolak (Rusak)")

 elif kondisi == 4:

 print(f"Barang ke-{i+1}: Diterima (Salah kirim)")

 else:

 print(f"Barang ke-{i+1}: Kode tidak dikenal")

- Penjelasan:

- Daftar kondisi barang disimpan dalam list barang, yang berisi kode status barang (1–5).
- Dilakukan perulangan (looping) menggunakan for i in range(len(barang)) untuk memproses setiap item satu per satu.
- Nilai kode status dari barang ke-i disimpan dalam variabel kondisi.
- Kemudian dicek nilainya menggunakan struktur if-elif-else:
- Jika kondisi == 1, berarti barang masih segel dan tidak rusak → diterima.

- Jika kondisi == 2, barang tidak segel tapi tidak rusak → perlu verifikasi.
- Jika kondisi == 3, barang rusak → ditolak.
- Jika kondisi == 4, barang salah kirim → diterima juga, tapi kasus khusus.
- Jika nilai kode tidak termasuk 1–4 → dianggap kode tidak dikenal.

c. Modifikasi algoritma

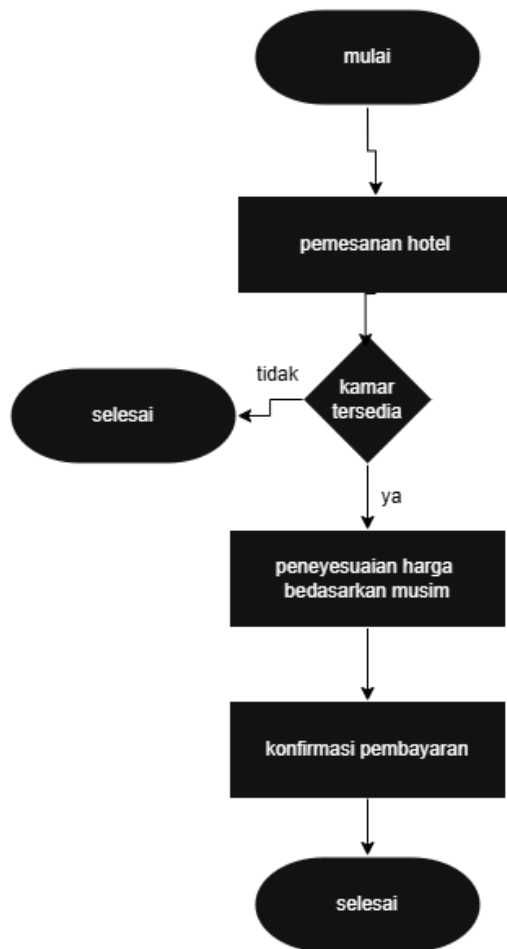
```
barang = {
    "jenis": "elektronik",    # atau "pakaian", "buku", dll
    "rusak": False,
    "punya_bukti": True,
    "hari_sejak_beli": 10
}
```

```
batas_pengembalian = {
    "elektronik": 14,
    "lainnya": 7
}
```

```
if barang["jenis"] == "elektronik":
    batas = batas_pengembalian["elektronik"]
else:
    batas = batas_pengembalian["lainnya"]
```

```
if barang["rusak"]:
    if barang["punya_bukti"]:
        print("Pengembalian Diterima")
    else:
        print("Pengembalian Ditolak (tidak ada bukti pembelian)")
else:
    if barang["hari_sejak_beli"] <= batas:
        print("Pengembalian Diterima")
    else:
        print("Pengembalian Ditolak (melebihi batas hari)")
```

4. Kasus: Anda bekerja di tim pengembangan sistem reservasi hotel otomatis berbasis AI. Salah satu tantangan utama adalah membuat sistem yang dapat menyesuaikan harga kamar berdasarkan permintaan dan musim.
 - a. Flowmap



b. Algoritma

- Data Historis: Kumpulkan data harga, okupansi, dan faktor eksternal
- Feature Engineering: Musim, hari libur, acara lokal, dll.
- Model Training: Gunakan algoritma seperti Random Forest atau Neural Network
- Deployment: Integrasikan model ke sistem reservasi

c. Update flowmap

