

PERTEMUAN 6

ASESMEN AKHIR : Menerapkan Pemrograman Berbasis Teks

Bentuk asesmen : Tes Praktik

Asesmen akhir ini dilakukan untuk menilai kemampuan siswa dalam menerapkan pemrograman berbasis teks.

"Bubble Tea Shop Simulator"

:

```
PROGRAM BubbleTeaShop
  DEKLARASI
    MENU: {"Original Milk Tea": 15000, ...}
    TOPPINGS: {"Boba": 3000, ...}
    harga_dasar, total_topping, total: integer
    pilihan_rasa, ukuran: string
    ID_pesanan: integer

  MULAI
    TAMPILKAN "✨ SELAMAT DATANG DI BUBBLE TEA SHOP! ✨"
    TAMPILKAN_MENU()

    ULANGI:
      BACA pilihan_rasa
      SAMPAI pilihan_rasa ADA DI MENU

      ukuran = PILIH_UKURAN()
      total_topping = PILIH_TOPPING()

      JIKA ukuran = "Large" MAKA
        harga_dasar = MENU[pilihan_rasa] + 5000
      LAINNYA
        harga_dasar = MENU[pilihan_rasa]

      total = harga_dasar + total_topping
      ID_pesanan = GENERATE_ANGKA_ACAK(1000, 9999)

      TAMPILKAN_STRUK(pilihan_rasa, ukuran, total_topping, total, ID_pesanan)
  SELESAI
```

Sample Output

```
✨ SELAMAT DATANG DI BUBBLE TEA SHOP! ✨

=== MENU BUBBLE TEA ===
Original Milk Tea: Rp15,000
Taro Milk Tea: Rp18,000
Matcha Milk Tea: Rp20,000
Chocolate Milk Tea: Rp17,000
=====
Pilih rasa: matcha
Pilih ukuran (R/L): L

=== TOPPING ===
Boba: Rp3,000
Pudding: Rp4,000
Jelly: Rp2,500
Cheese Foam: Rp5,000
Pilih topping (atau 'selesai'): boba
Ditambahkan: Boba
Pilih topping (atau 'selesai'): cheese foam
Ditambahkan: Cheese Foam
Pilih topping (atau 'selesai'): selesai

=== STRUK PESANAN ===
Rasa      : Matcha Milk Tea (Large)
Topping   : Rp8,000
TOTAL     : Rp33,000
ID Pesanan: #7284
Terima kasih! 😊
```

Instruksi Implementasi untuk Siswa

Langkah 1: Setup Variabel

- Buat dictionary untuk menu dan topping

Langkah 2: Fungsi Tampilkan Menu

- Gunakan loop untuk menampilkan item menu

Langkah 3: Validasi Input Ukuran

- Gunakan while loop sampai input valid

Langkah 4: Hitung Total Topping

- Gunakan *accumulator pattern*

Catatan untuk Siswa

1. Python:

- Gunakan title() untuk format input
- Manfaatkan random.randint() untuk ID pesanan

2. JavaScript:

- Simpan data di object literals
- Gunakan toLocaleString() untuk format mata uang

3. C:

- Hindari string kompleks dengan menggunakan array index
- Gunakan toupper() untuk normalisasi input

Challenge: Tambahkan fitur diskon 10% untuk pembelian di atas Rp50.000!

Tabel Skenario Uji Coba

ID	Deskripsi	Input	Output yang Diharapkan	Tujuan
TC-1	Pesan valid dengan ukuran Large dan topping	Rasa: Matcha Milk Tea Ukuran: L Topping: Boba, Pudding Durasi: -	Total: Rp20.000 (harga dasar) + Rp5.000 (Large) + Rp7.000 (topping) = Rp32.000 ID Pesanan: 4 digit angka	Verifikasi perhitungan harga dan format output.

TC-2	Pesan tanpa topping	Rasa: Taro Milk Tea Ukuran: R Topping: Selesai	Total: Rp18.000 (harga dasar) + Rp0 = Rp18.000	Memastikan program menangani input tanpa topping.
TC-3	Input rasa tidak valid	Rasa: "Coklat"	Pesan error: " <i>Rasa tidak tersedia!</i> "	Validasi input rasa.
TC-4	Input ukuran tidak valid	Ukuran: "XL"	Pesan error: " <i>Pilihan tidak valid!</i> "	Validasi input ukuran.
TC-5	Input topping tidak valid	Topping: "Keju"	Topping diabaikan/tidak menambah harga	Validasi input topping.
TC-6	Total harga \geq Rp50.000	Rasa: Matcha Milk Tea (Large) + 4 topping (misal: Boba, Pudding, Jelly, Cheese Foam)	Total: Rp25.000 (harga dasar) + Rp14.000 = Rp39.000 (jika tanpa diskon)	Uji kasus tepi untuk fitur diskon (opsional).

Rubrik Penilaian

Kriteria Penilaian (Skala: 1-4, 4 = Sangat Baik):

Aspek	4	3	2	1
Fungsionalitas	Semua skenario uji berjalan sempurna.	1-2 skenario error.	3-4 skenario error.	Program tidak berjalan.
Variabel & Tipe Data	Variabel deskriptif, tipe data tepat, dan konstanta digunakan.	Ada 1-2 kesalahan penamaan/tipe data.	Variabel ambigu/tipe data salah.	Variabel tidak relevan.
Struktur Kontrol	Loop dan percabangan digunakan optimal untuk validasi input.	Ada 1-2 redundansi logika.	Logika tidak efisien.	Tidak ada validasi input.

Fungsi	Program modular dengan ≥ 4 fungsi yang jelas.	Menggunakan 3 fungsi.	Menggunakan 2 fungsi.	Tidak menggunakan fungsi.
Library Eksternal	Menggunakan random untuk ID pesanan dan format mata uang (jika ada).	Hanya menggunakan 1 fitur library.	Tidak menggunakan library.	-
Keterbacaan Kode	Kode rapi, indentasi konsisten, komentar jelas.	Kode rapi tetapi komentar kurang.	Indentasi tidak konsisten.	Kode berantakan.
Penanganan Error	Menangani semua input tidak valid dengan pesan jelas.	Menangani 75% kasus error.	Menangani 50% kasus error.	Tidak ada penanganan error.
Kreativitas	Menambahkan fitur unik (diskon, loyalitas, dll).	Menambahkan 1 fitur tambahan.	Hanya mengikuti contoh.	Tidak ada inovasi.

Pedoman Penilaian

1. Skor Total:

- Setiap aspek dinilai 1-4.
- Total skor = Jumlah skor semua aspek.
- Maksimal: 32 poin (8 aspek \times 4).

2. Konversi Nilai Akhir:

- A (Sangat Baik): 28-32 poin
- B (Baik): 21-27 poin
- C (Cukup): 14-20 poin
- D (Perlu Perbaikan): ≤ 13 poin

3. Ketentuan Tambahan:

- Bonus 5 poin untuk fitur kreatif yang relevan (misal: sistem diskon).

- Pengurangan 2 poin untuk keterlambatan pengumpulan.
- Plagiarisme: Langsung mendapat nilai E.

4. Umpan Balik:

- Berikan komentar spesifik untuk setiap aspek (contoh: "Validasi input ukuran sudah baik, tetapi penanganan topping tidak valid perlu diperbaiki").
- Soroti bagian kode yang inovatif.

Contoh Penilaian

Nama Siswa: Budi

- Fungsionalitas: 4
 - Variabel & Tipe Data: 3
 - Struktur Kontrol: 4
 - Fungsi: 4
 - Library: 3
 - Keterbacaan: 4
 - Error Handling: 3
 - Kreativitas: 4 (+5 bonus)
- Total: 29 + 5 = 34 (Nilai A)

Catatan: "Program Budi sangat kreatif dengan tambahan fitur diskon, tetapi perlu memperbaiki penanganan input topping!"