
UJIAN TENGAH SEMESTER (UTS)
GANJILTAHUN AKADEMIK 2024/2025

<i>Mata Kuliah</i>	: Testing Dan QA Perangkat Lunak		
<i>Fakultas/Jurusan</i>	: Teknik / Informatika	<i>Hari/Tanggal</i>	: Selasa, 29 Oktober 2024
<i>Semester/Kelas</i>	: VII / 07TPLM004	<i>Waktu</i>	: 18:20 – 20:00
<i>Dosen</i>	: Ahmad Fauzi, S.Kom., M.Kom.	<i>Jam ke</i>	: 1 (Satu)
<i>Sifat</i>	: -	<i>Shift</i>	: Reguler B

Jawab pertanyaan di bawah ini dengan singkat dan jelas!

1. Buatlah perencanaan pengujian pada web berikut:

NIM Ganjil: <https://automationteststore.com/index.php?rt=account/login>

NIM Genap: <https://parabank.parasoft.com/parabank/index.htm>

2. Pada soal no. 2 buat tabel test case dengan mencakup 3 metode pengujian dan bagi kedalam tahapan-tahapan pengujian minimal 5 IDTest.
3. Sebuah fungsi di bawah ini bertugas untuk menghitung diskon berdasarkan harga dan jenis pelanggan. Kode program ditulis dalam bahasa Java sebagai berikut:

```
public double hitungDiskon(double harga, String tipePelanggan) {  
    double diskon = 0;  
    if (tipePelanggan.equals("VIP")) {  
        if (harga > 100000) {  
            diskon = harga * 0.1;  
        } else {  
            diskon = harga * 0.05;  
        }  
    } else if (tipePelanggan.equals("Reguler")) {  
        if (harga > 50000) {  
            diskon = harga * 0.07;  
        } else {  
            diskon = harga * 0.03;  
        }  
    }  
    return diskon;  
}
```

Pertanyaan Nim Ganjil: Branch Coverage Analysis

- a. Identifikasi jumlah cabang yang ada dalam kode di atas dan sebutkan semua jalur eksekusi yang mungkin.
- b. Tentukan minimal 2 set data uji yang dapat digunakan untuk mencapai 100% branch coverage. Jelaskan alasan pemilihan data tersebut.

Pertanyaan Nim Genap: Path Coverage Analysis

- a. Berapa jumlah jalur eksekusi (path) yang mungkin dalam kode tersebut?
 - b. Tentukan minimal 3 set data uji yang dapat digunakan untuk mencapai 100% path coverage. Jelaskan alasan di balik pemilihan data tersebut dan bagaimana setiap data uji menguji jalur yang berbeda.
4. Diketahui spesifikasi aplikasi e-commerce sederhana untuk menghitung total harga setelah penerapan diskon. Spesifikasi aplikasi sebagai berikut:
- a. Jika pelanggan adalah VIP dan total belanja lebih dari 1.000.000, maka diskon yang diberikan adalah 15%.
 - b. Jika pelanggan adalah Reguler dan total belanja lebih dari 500.000, maka diskon yang diberikan adalah 10%.
 - c. Selain itu, pelanggan akan mendapatkan diskon 5%.

Pertanyaan Nim Genap: Equivalence Partitioning

- a. Identifikasi kelas ekuivalensi yang berbeda untuk variabel tipe pelanggan dan total belanja.
- b. Buatlah minimal 3 set data uji yang mewakili tiap kelas ekuivalensi untuk pengujian aplikasi.

Pertanyaan Nim Ganjil: Boundary Value Analysis (BVA)

- a. Identifikasi nilai batas untuk variabel total belanja.
 - b. Buatlah minimal 3 set data uji yang menguji batas-batas ini dan jelaskan kenapa data tersebut penting untuk diuji.
5. Sebuah perusahaan pengembang perangkat lunak telah menyelesaikan pengembangan aplikasi pemesanan tiket online. Aplikasi ini memungkinkan pengguna untuk melakukan pemesanan tiket pesawat, kereta api, dan bus secara online. Fitur-fitur utama dalam aplikasi meliputi **Pencarian tiket, Pemesanan tiket, Pembayaran, E-ticket**. Selama fase UAT (User Acceptance Testing), beberapa pengguna yang mewakili calon pengguna akhir diundang untuk menguji aplikasi ini. Tim QA telah mengidentifikasi berbagai bug yang muncul selama pengujian ini, dan sekarang sedang mengevaluasi serta mengkategorikan bug tersebut untuk diperbaiki. Berikut adalah tabel test case pelaporan bug:

ID Test Case	Deskripsi Bug	Langkah Reproduksi	Hasil yang Diharapkan	Hasil Sebenarnya	Severity	Priority	Status
TC-001	Harga tiket tidak konsisten antara pencarian dan checkout	1. Buka aplikasi.	Harga tiket di pencarian dan checkout harus sama.	Harga di pencarian Rp 500.000, tetapi di checkout berubah menjadi Rp 700.000.			Open
		2. Cari tiket pesawat Jakarta ke Bali.					
		3. Periksa harga tiket di halaman pencarian.					
		4. Lanjutkan ke halaman checkout.					
TC-002	Gagal melanjutkan pembayaran menggunakan dompet digital	1. Cari dan pilih tiket.	Pembayaran berhasil diproses dengan dompet digital.	Muncul pesan "Pembayaran gagal" saat menggunakan dompet digital, metode lain berfungsi			Open
		2. Lanjutkan ke pembayaran.					
		3. Pilih metode pembayaran dompet digital.					
		4. Klik tombol "Bayar".					
TC-003	Tampilan tidak responsif di perangkat mobile	1. Akses aplikasi di perangkat mobile dengan resolusi tertentu.	Semua elemen UI terlihat dan dapat diakses dengan benar.	Beberapa elemen UI hilang, dan tombol "Cari" tidak terlihat di layar.			Open
		2. Buka halaman pencarian tiket.					
		3. Perhatikan layout dan elemen UI di halaman tersebut.					
TC-004	E-ticket tidak diterima melalui email	1. Selesaikan pemesanan dan pembayaran tiket.	Pengguna menerima e-ticket di email setelah pembayaran berhasil.	Pengguna tidak menerima e-ticket di email, tetapi bisa diakses melalui			Open
		2. Periksa email untuk e-ticket yang dikirimkan.					
TC-005	Aplikasi crash setelah pembayaran berhasil	1. Lakukan pemesanan tiket.	Halaman konfirmasi muncul setelah pembayaran selesai.	Aplikasi crash setiap kali pembayaran selesai dan mencoba memuat halaman			Open
		2. Lanjutkan pembayaran hingga selesai.					
		3. Tunggu halaman konfirmasi.					
TC-006	Pengguna bisa memesan tiket untuk tanggal di masa lalu	1. Cari tiket dengan memilih tanggal di masa lalu.	Aplikasi menolak pemesanan tiket untuk tanggal di masa lalu.	Aplikasi tetap memproses pemesanan tiket untuk tanggal yang sudah			Open
		2. Lanjutkan ke pemesanan tiket.					
		3. Selesaikan proses pembayaran.					
TC-007	Tiket tidak muncul di riwayat setelah pembayaran berhasil	1. Selesaikan pemesanan dan pembayaran tiket.	Tiket muncul di riwayat pemesanan setelah pembayaran.	Tiket tidak muncul di riwayat pemesanan setelah pembayaran berhasil.			Open
		2. Cek riwayat pemesanan di aplikasi setelah pembayaran.					

Pertanyaan:

- Identifikasikan dan lengkapi kategori bug apa yang sesuai untuk setiap bug yang ditemukan pada studi kasus di atas.
- Jelaskan bagaimana kamu akan memprioritaskan bug yang ditemukan untuk diperbaiki (misalnya, dari yang paling kritis hingga minor) dan mengapa.

“Selamat Mengerjakan”