

Nomor 1

1. *Proses Bisnis:*

Pemesanan Online pada Toko Buku:

a) *Registrasi dan Login:*

- *Pelanggan baru mendaftar dengan email dan password*
- *Pelanggan yang sudah terdaftar melakukan login*

b) *Pencarian dan Pemilihan Buku:*

- *Fitur pencarian dengan filter (genre, penulis, harga)*
- *Halaman detail buku menampilkan sinopsis, review, dan rating*

c) *Manajemen Keranjang Belanja:*

- *Kemampuan menambah/mengurangi jumlah buku*
- *Fitur "Simpan untuk Nanti"*

d) *Proses Checkout:*

- *Pemilihan alamat pengiriman (bisa menyimpan multiple alamat)*
- *Pilihan pengiriman (reguler, ekspres) dengan estimasi waktu dan biaya*
- *Penerapan kode voucher atau diskon*

e) *Sistem Pembayaran:*

- *Integrasi dengan berbagai metode pembayaran (kartu kredit, transfer bank, e-wallet)*
- *Verifikasi pembayaran otomatis*

f) *Pemrosesan Order:*

- *Notifikasi ke admin untuk order baru*
- *Pengecekan ketersediaan stok*
- *Packing dan labeling*

g) *Pengiriman:*

- *Integrasi dengan jasa ekspedisi*
- *Pelacakan nomor resi*

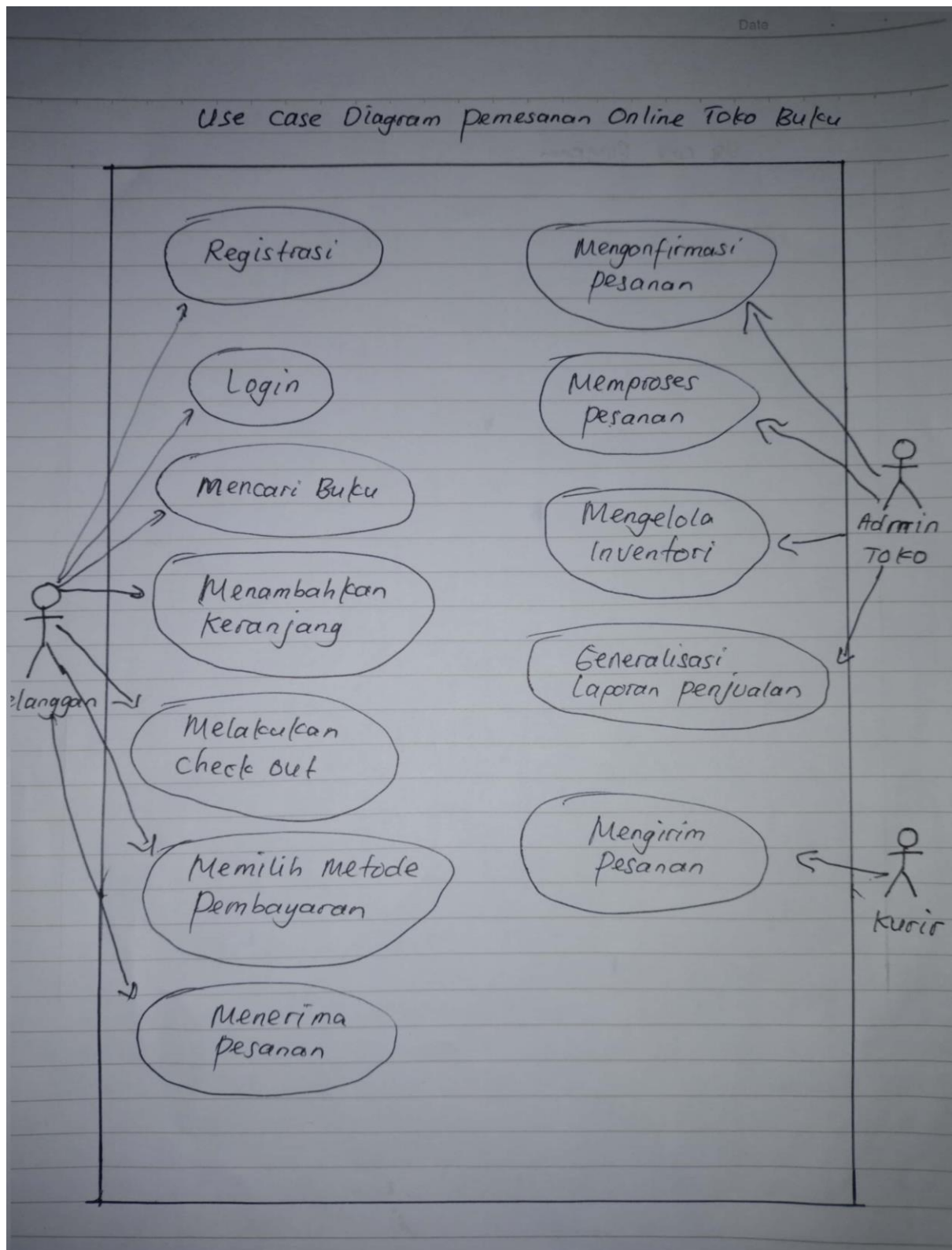
h) Layanan Pasca Pembelian:

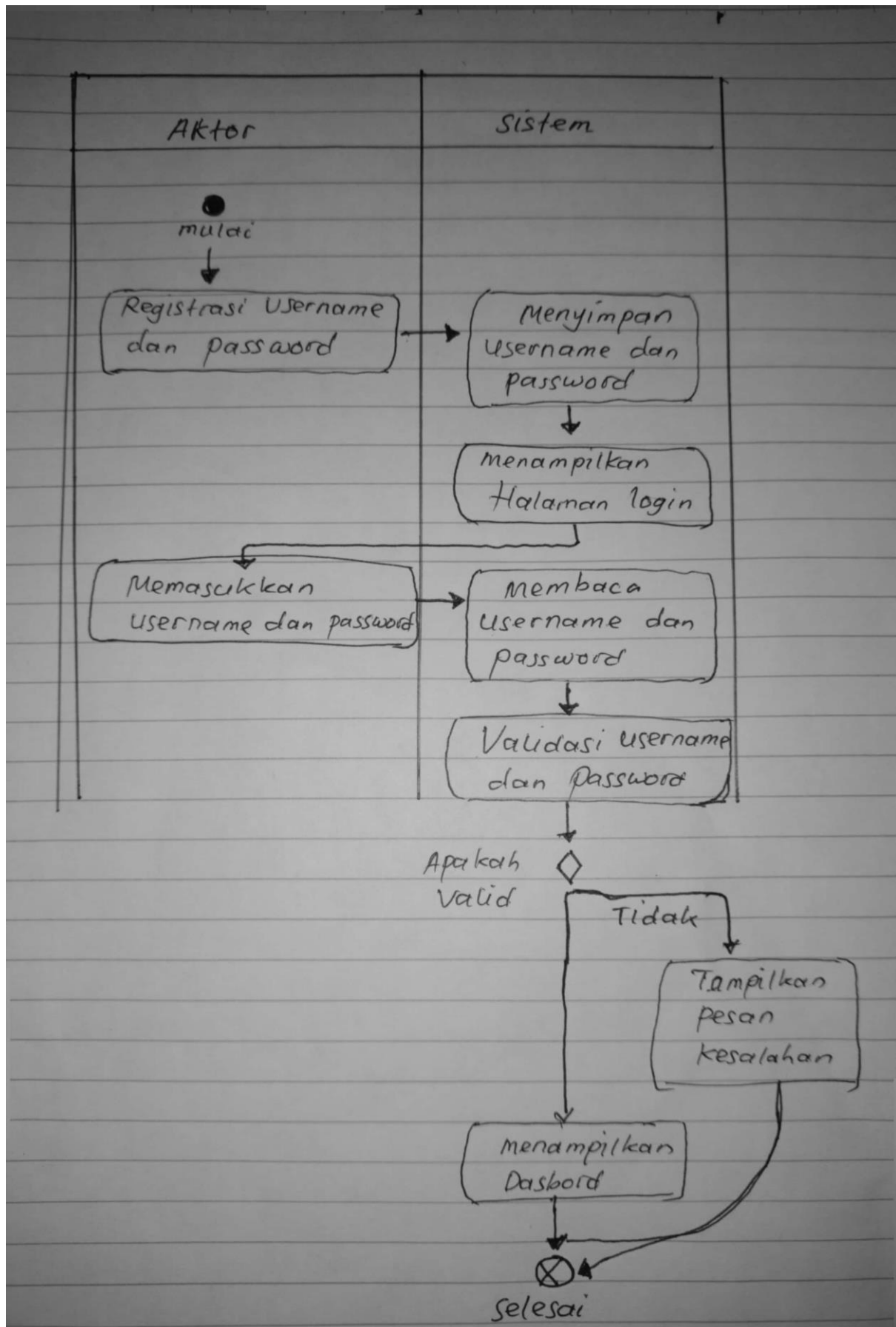
- Sistem rating dan review*
- Penanganan retur atau penukaran*

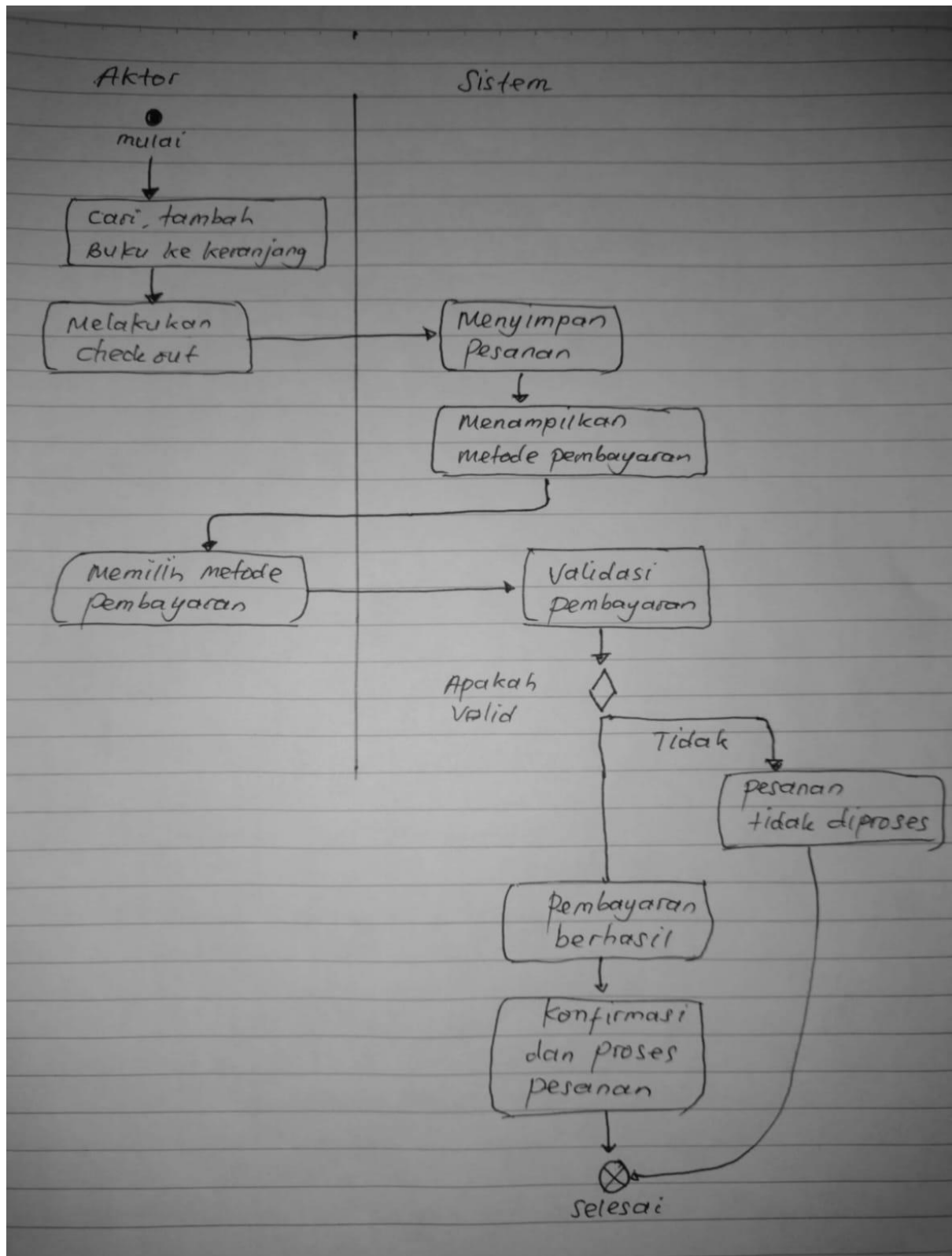
i) Analisis Data:

- Pencatatan preferensi pelanggan untuk rekomendasi di masa depan*
- Analisis penjualan untuk manajemen inventori*

Nomor 2







Nomor 3

3. Pengujian Perangkat Lunak Berorientasi Objek:

a) Unit Testing untuk kelas Buku:

@Test

```
public void testKonstruktor() {  
    Buku buku = new Buku("Java Programming", "John Doe",  
        100.0);  
    assertEquals("Java Programming", buku.getJudul());  
    assertEquals("John Doe", buku.getPenulis());  
    assertEquals(100.0, buku.getHarga(), 0.001);  
}
```

@Test

```
public void testGetHargaDiskonMaksimum() {  
    Buku buku = new Buku("Java Programming", "John Doe",  
        100.0);  
    assertEquals(0.0, buku.getHargaDiskon(100), 0.001);  
}
```

@Test(expected = IllegalArgumentException.class)

```
public void testGetHargaDiskonNegatif() {  
    Buku buku = new Buku("Java Programming", "John Doe",  
        100.0);  
    buku.getHargaDiskon(-10);  
}
```

b) Mocking dan Dependency Injection:

```
public class PesananService {  
    private BukuRepository bukuRepo;  
  
    public PesananService(BukuRepository bukuRepo) {
```



```
        this.bukuRepo = bukuRepo;
    }

    public double hitungTotalHarga(List<String> idBuku) {
        return idBuku.stream()
            .map(id -> bukuRepo.findById(id))
            .mapToDouble(Buku::getHarga)
            .sum();
    }
}

@Test
public void testHitungTotalHarga() {
    BukuRepository mockRepo = mock(BukuRepository.class);
    when(mockRepo.findById("1"))
        .thenReturn(new Buku("Book1", "Author1", 50.0));
    when(mockRepo.findById("2"))
        .thenReturn(new Buku("Book2", "Author2", 30.0));

    PesananService service = new PesananService(mockRepo);
    double total = service.hitungTotalHarga(
        Arrays.asList("1", "2")
    );

    assertEquals(80.0, total, 0.001);
    verify(mockRepo, times(2)).findById(anyString());
}
```

Nomor 4

4. Studi Kasus dan Rencana Pengujian:

Rencana pengujian untuk Aplikasi Mobile Manajemen Tugas Kuliah:

a) Pengujian Unit:

- Test-Driven Development (TDD) untuk fungsi-fungsi utama
- Pengujian boundary cases (misalnya, tanggal invalid)
- Pengujian exception handling

b) Pengujian Integrasi:

- Pengujian integrasi antara modul database lokal dan sinkronisasi cloud
- Pengujian callback dan event handling antar komponen

c) Pengujian Sistem:

- Stress testing dengan simulasi ratusan tugas
- Pengujian konsumsi baterai dan penggunaan memori
- Pengujian skenario offline dan sinkronisasi setelah kembali online

d) Pengujian Penerimaan Pengguna (UAT):

- Beta testing dengan grup mahasiswa dari berbagai jurusan
- A/B testing untuk desain UI alternatif
- Pengumpulan metrik penggunaan dan analisis

e) Pengujian Keamanan:

- Penetration testing untuk menguji keamanan data
- Pengujian sanitasi input untuk mencegah SQL injection
- Audit kode untuk kerentanan keamanan umum

f) Pengujian Usability:

- Pengujian aksesibilitas untuk pengguna dengan keterbatasan
- Analisis heatmap untuk interaksi pengguna
- Pengujian responsivitas UI pada berbagai ukuran layar

g) *Pengujian Kinerja:*

- *Pengujian waktu respon untuk operasi CRUD*
- *Pengujian skalabilitas dengan dataset besar*
- *Pengujian kinerja sinkronisasi data*

h) *Pengujian Kompatibilitas:*

- *Pengujian pada berbagai versi OS (Android/iOS)*
- *Pengujian pada berbagai perangkat dengan spesifikasi berbeda*

Nomor 5

5. Pentingnya Pengujian Perangkat Lunak:

a) Jaminan Kualitas:

- Memastikan konsistensi fungsionalitas di seluruh bagian aplikasi
- Validasi bahwa semua requirement telah dipenuhi

b) Deteksi Bug Dini:

- Mengidentifikasi masalah logika dan algoritma
- Menemukan edge cases yang mungkin terlewatkan selama pengembangan

c) Peningkatan Keandalan:

- Menguji ketahanan aplikasi terhadap input yang tidak valid
- Memastikan stabilitas dalam berbagai kondisi jaringan dan perangkat keras

d) Optimasi Kinerja:

- Identifikasi bottleneck dan area yang memerlukan optimasi
- Memastikan aplikasi responsive bahkan dengan beban tinggi

e) Keamanan:

- Mencegah kebocoran data sensitif
- Menguji ketahanan terhadap serangan umum seperti XSS dan CSRF

f) Compliance:

- Memenuhi standar industri seperti ISO/IEC 25010
- Memastikan kepatuhan terhadap regulasi seperti GDPR untuk privasi data

g) Efisiensi Biaya:

- Mengurangi biaya maintenance jangka panjang
- Mencegah kerugian finansial akibat bug kritical pasca-rilis

h) Kepuasan Pengguna:

- Memastikan user experience yang mulus dan intuitif
- Mengurangi frustrasi pengguna akibat crash atau bug

i) Dokumentasi:

- Pengujian berfungsi sebagai dokumentasi hidup dari perilaku yang diharapkan
- Membantu dalam onboarding developer baru ke proyek

j) Confidence dalam Perubahan:

- Memungkinkan refactoring dan peningkatan dengan risiko minimal
- Memberikan kepercayaan diri dalam melakukan deployment

k) Feedback Loop:

- Menyediakan umpan balik cepat kepada developer tentang perubahan kode
- Mendorong praktik pengembangan yang lebih baik dan lebih hati-hati