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

2. Use Case dan Activity Diagram:

   Use Case Diagram:

   Aktor Tambahan:

   - Sistem Pembayaran
   - Kurir

   Use Cases Tambahan:

   - Menulis Review
   - Melacak Pesanan
   - Mengelola Inventori (Admin)
   - Generasi Laporan Penjualan (Admin)

   Relasi:

   - "Melacak Pesanan" extends "Melakukan Checkout"
   - "Menulis Review" requires "Login"
   - "Mengelola Inventori" includes "Melihat Stok" dan "Memperbarui Stok"

   Activity Diagram:

   - Tambahkan decision point setelah verifikasi pembayaran
   - Jika pembayaran gagal, kembali ke langkah pemilihan metode pembayaran
   - Tambahkan swim lane untuk membedakan aktivitas Pelanggan, Sistem, dan Admin
   - Masukkan aktivitas paralel, misalnya sistem secara bersamaan memperbarui inventori dan mengirim konfirmasi ke pelanggan

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());
}
```

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

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 kritikal 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

No 2

Saya memahami bahwa Anda ingin gambar untuk use case diagram dan activity diagram. Saya mohon maaf karena saya tidak dapat menghasilkan atau menampilkan gambar secara langsung. Namun, saya bisa memberikan panduan detail tentang bagaimana menggambar diagram-diagram tersebut.

Untuk Use Case Diagram:

1. Gambar sebuah kotak besar yang merepresentasikan sistem.
2. Di luar kotak, gambar figur stick-man untuk merepresentasikan aktor:
    - Pelanggan
    - Admin Toko
    - Sistem Pembayaran
    - Kurir
3. Di dalam kotak, gambar oval-oval yang merepresentasikan use case:
    - Mendaftar Akun
    - Login
    - Mencari Buku
    - Menambahkan ke Keranjang
    - Melakukan Checkout
    - Memilih Metode Pembayaran

- Mengonfirmasi Pesanan
- Memproses Pesanan
- Mengirim Pesanan
- Menulis Review
- Melacak Pesanan
- Mengelola Inventori
- Generasi Laporan Penjualan

4. Hubungkan aktor dengan use case yang relevan menggunakan garis lurus.
5. Gunakan garis putus-putus dengan panah dan label "extends" untuk menghubungkan "Melacak Pesanan" ke "Melakukan Checkout".
6. Gunakan garis putus-putus dengan panah dan label "includes" untuk menghubungkan "Mengelola Inventori" ke "Melihat Stok" dan "Memperbarui Stok".

Untuk Activity Diagram:

1. Mulai dengan titik awal (filled circle).
2. Gambar kotak-kotak aktivitas yang dihubungkan dengan panah, mengikuti urutan:
    - Pelanggan mencari buku
    - Pelanggan menambahkan buku ke keranjang
    - Pelanggan melakukan checkout
    - Sistem menampilkan rincian pesanan
    - Pelanggan memilih metode pembayaran
    - Pelanggan mengonfirmasi pesanan
    - Sistem memverifikasi pembayaran
3. Setelah verifikasi pembayaran, tambahkan decision point (diamond):
    - Jika berhasil, lanjut ke proses selanjutnya
    - Jika gagal, kembali ke pemilihan metode pembayaran
4. Lanjutkan dengan aktivitas:
    - Admin memproses pesanan
    - Admin mengirim pesanan
    - Pelanggan menerima pesanan
5. Akhiri dengan titik akhir (filled circle inside a larger circle).
6. Bagi diagram menjadi tiga swim lane vertikal untuk Pelanggan, Sistem, dan Admin.
7. Tambahkan aktivitas paralel setelah konfirmasi pembayaran:

- Sistem memperbarui inventori
- Sistem mengirim konfirmasi ke pelanggan