

# 4

## STUDI KASUS

### Objektif :

- **Memahami studi kasus Galeri VCD**
  - **Mampu mendesain sistem Galeri VCD dengan UML.**
  - **Mampu mengimplementasikan desain sistem Galeri VCD ke dalam Microsoft Visio**
- 
- 

### 4.1. Pemahaman kasus Galeri VCD

VCD galeri adalah sebuah galeri yang bergerak di bidang retail dengan menjual produk-produk hiburan VCD, DVD, CD Playstation, accessories dan lain-lain. Pada saat ini galeri tersebut ingin meningkatkan efisiensi kerja dari para pegawainya dan meningkatkan penjualan dengan cara mengotomatisasi sistem penjualan.

Berikut adalah sistem yang bersifat manual yang sedang berjalan di galeri tersebut. Proses penjualan dimulai saat seorang pelanggan menanyakan tentang suatu produk kepada petugas toko. Oleh petugas toko, permintaan tersebut akan ditulis dalam bentuk bon. Selanjutnya atas dasar bon tersebut, petugas toko akan mengecek dan mengambil barang yang dipesan. Jika barang tidak ada atau jumlah permintaan tidak sesuai maka transaksi akan dikonfirmasi untuk tetap dilakukan atau tidak. Jika transaksi akan tetap dilakukan dengan cara mengambil barang jenis lain atau mengubah jumlah barang yang dipesan, maka petugas akan melakukan perhitungan pembayaran. Pembayaran bisa dilakukan dengan cara tunai (cash), dengan kartu kredit atau dengan kartu debit. Jika pembayaran dilakukan dengan tunai, maka transaksi akan dicatat pada buku penjualan, namun jika dilakukan dengan kartu kredit atau kartu debit maka akan dilakukan otorisasi terlebih dahulu dengan mesin otorisasi yang disediakan oleh bank. Setelah semua proses selesai, barulah proses penjualan dicatat di buku penjualan.

Untuk memberikan pelayanan yang lebih baik kepada pelanggan, pelanggan diperbolehkan mengembalikan (return) barang yang tidak sesuai dengan spesifikasi dengan

barang yang sama atau dengan barang lain yang senilai. Petugas akan mencatat transaksi ini di buku return.

Untuk menghitung stok, petugas stok akan membaca buku inventori barang sebagai acuan untuk menghitung stok awal. Selanjutnya berdasarkan dokumen penjualan dan dokumen return akan bisa dihitung stok akhir.

#### 4.1.1. Rancangan Skenario

Merancang scenario merupakan langkah awal yang harus dilakukan dalam tahapan perancangan sistem menggunakan UML. Karena disinilah saatnya kita mendefinisikan bisnis proses – bisnis proses apa saja yang terdapat didalam sistem yang kita tentukan.

Berdasarkan pemahaman kasus pada bagian sebelumnya, maka kita dapat mendefinisikan skenario sebagai berikut ini:

1. Name : Request barang
  - a. Actor who initiates :
    - Pelanggan
  - b. Precondition
    - None
  - c. Steps
    - menanyakan item barang kepada penjaga toko
  - d. Postcondition
    - Entry permintaan
  - e. Actor who gets benefits
    - Pelanggan
  
2. Name : Entry Permintaan
  - a. Actor who initiates :
    - Petugas toko
  - b. Precondition
    - Menanyakan barang

- c. Steps
    - Memasukkan nama barang yang dicari atau memasukkan kode barang yang dicari ke form pencarian
  - d. Postcondition
    - cari barang
  - e. Actor who gets benefits
    - Petugas toko
3. Name : cari barang
- a. Actor who initiates :
    - Petugas toko
  - b. Precondition
    - Entry permintaan
  - c. Steps
    - program akan mencari data barang ke dalam database
  - d. Postcondition
    - view barang
  - e. Actor who gets benefits
    - Petugas toko
4. Name : view permintaan
- a. Actor who initiates :
    - Petugas toko
  - b. Precondition
    - Cari barang
  - c. Steps
    - Menampilkan hasil query permintaan dari database ke layar monitor dalam bentuk daftar.
  - d. Postcondition
    - Memilih barang

- e. Actor who gets benefits
  - Petugas toko, pelanggan

5. Name : Memilih barang

- a. Actor who initiates :
  - Pelanggan
- b. Precondition
  - View permintaan
- c. Steps
  - Memilih barang yang diinginkan dari stok barang yang ada.
- d. Postcondition
  - Bayar barang
- e. Actor who gets benefits
  - Petugas toko, pelanggan

6. Name : Bayar barang

- a. Actor who initiates :
  - Pelanggan
- b. Precondition
  - Memilih barang
- c. Steps
  - Mendefinisikan jumlah harga yang harus dibayar
  - Menentukan tipe pembayaran
- d. Postcondition
  - Validasi pembayaran
- e. Actor who gets benefits
  - Penjaga toko, petugas keuangan

7. Name : Validasi pembayaran

- a. Actor who initiates :

- Pelanggan
  - b. Precondition
    - Bayar Barang
  - c. Steps
    - Validasi cara pembayaran
  - d. Postcondition
    - None
  - e. Actor who gets benefits
    - Penjaga toko, petugas keuangan
8. Name : Hitung stok barang
- a. Actor who initiates :
    - Petugas stok barang
  - b. Precondition
    - None
  - c. Steps
    - hitung jumlah barang yang sudah terjual
    - hitung jumlah barang yang tersisa
  - d. Postcondition
    - Buat laporan
  - e. Actor who gets benefits
    - Kepala toko
9. Name : Hitung penjualan
- a. Actor who initiates :
    - Petugas keuangan
  - b. Precondition
    - None
  - c. Steps
    - hitung total transaksi penjualan

- d. Postcondition
  - Buat laporan
- e. Actor who gets benefits
  - kepala toko

10. Name : buat laporan

- a. Actor who initiates :
  - Kepala toko
- b. Precondition
  - Hitung stok penjualan, hitung stok barang
- c. Steps
  - buat laporan keuangan dan stok barang
- d. Postcondition
  - None
- e. Actor who gets benefits
  - Pimpinan

#### **4.1.2. Rancangan Use Case**

Setelah kita mendefinisikan skenario sistem, maka kita menuangkan rancangan skenario tersebut ke dalam bentuk diagram agar dapat dengan mudah dibaca oleh programmer ataupun pihak yang berkepentingan lainnya dan juga agar memudahkan pembuatan dokumentasinya. Diagram yang dimaksud disini adalah Use Case Diagram.

Berikut adalah Use case Diagramnya :



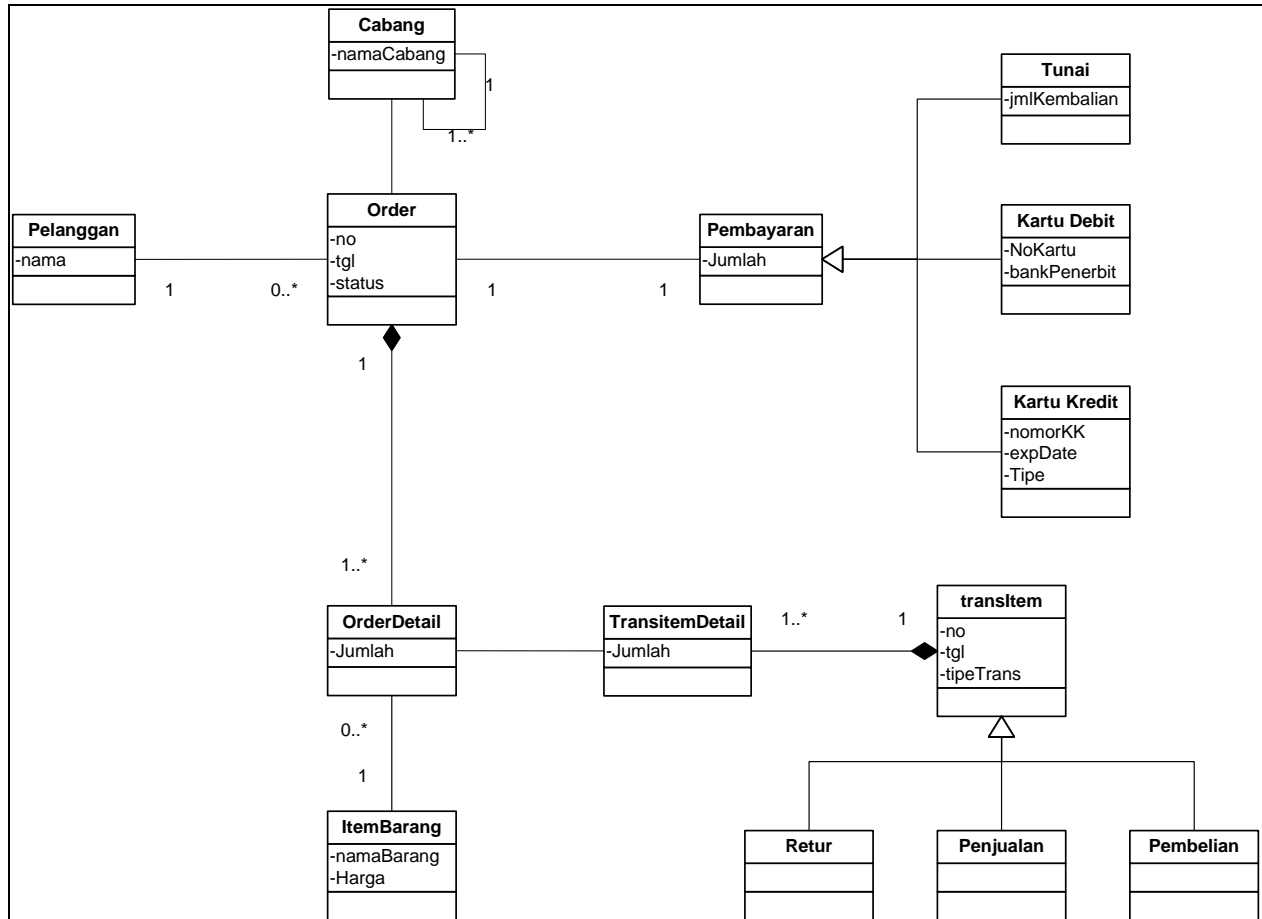
Gambar 4.1. Use Case diagram sistem galeri VCD

### 4.1.3. Rancangan Class

Setelah kita membuat use case diagram, langkah selanjutnya adalah membuat Class Diagram berdasarkan use case diagram tersebut. Class diagram ini harus berisikan objek-objek yang terdapat di dalam sistem toko VCD ini. Merujuk kepada kasus toko VCD tersebut, focus utama pada kasus ini adalah penjualan (order). **Order** diasosiasikan dengan **Pelanggan** yang melakukan **Pembelian** dan **Pembayaran**. **Pembayaran** dapat dilakukan dengan 3 cara : **Tunai**, **Kartu Debit** dan **Kartu Kredit**. **Order** mengandung **OrderDetail**, dimana OrderDetail

memperlihatkan barang apa saja yang dibeli oleh pelanggan. Oleh karenanya, **OrderDetail** diasosiasikan dengan **ItemBarang**.

Berikut adalah diagramnya :



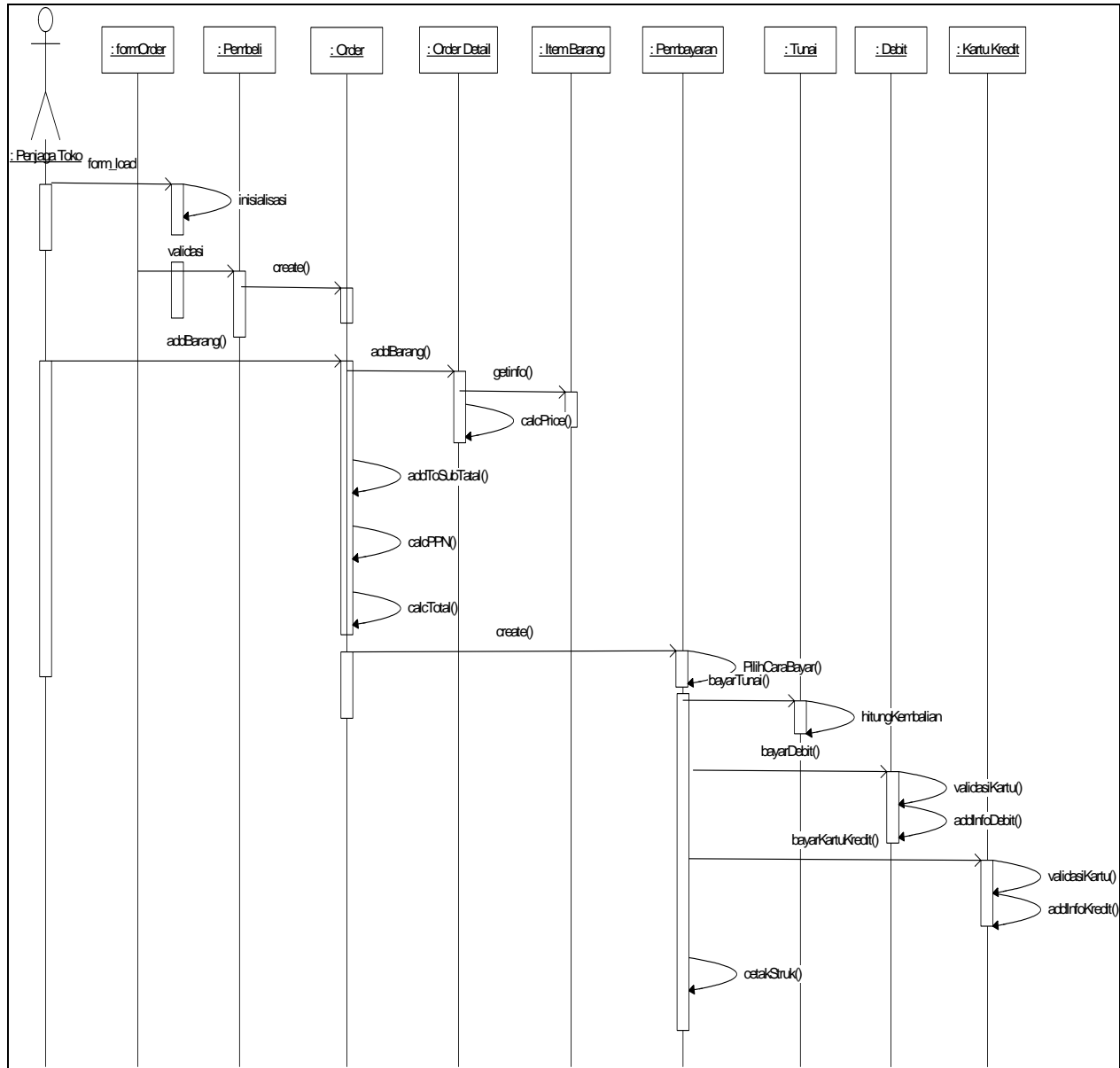
Gambar 4.2. Class diagram sistem galeri VCD

#### 4.1.4. Rancangan Sequence

Sequence diagram menjelaskan interaksi antar obyek yang disusun dalam suatu urutan waktu yaitu urutan kejadian yang dilakukan oleh seorang actor dalam menjalankan sistem. Diagram ini secara khusus berasosiasi dengan use case. Diagram ini menunjukkan bagaimana detail operasi dilakukan – pesan apa yang dikirim dan kapan terjadinya.



Berikut adalah rancangan sequence diagram yang dipetakan dari obyek-obyek yang ada pada class diagram sebelumnya.

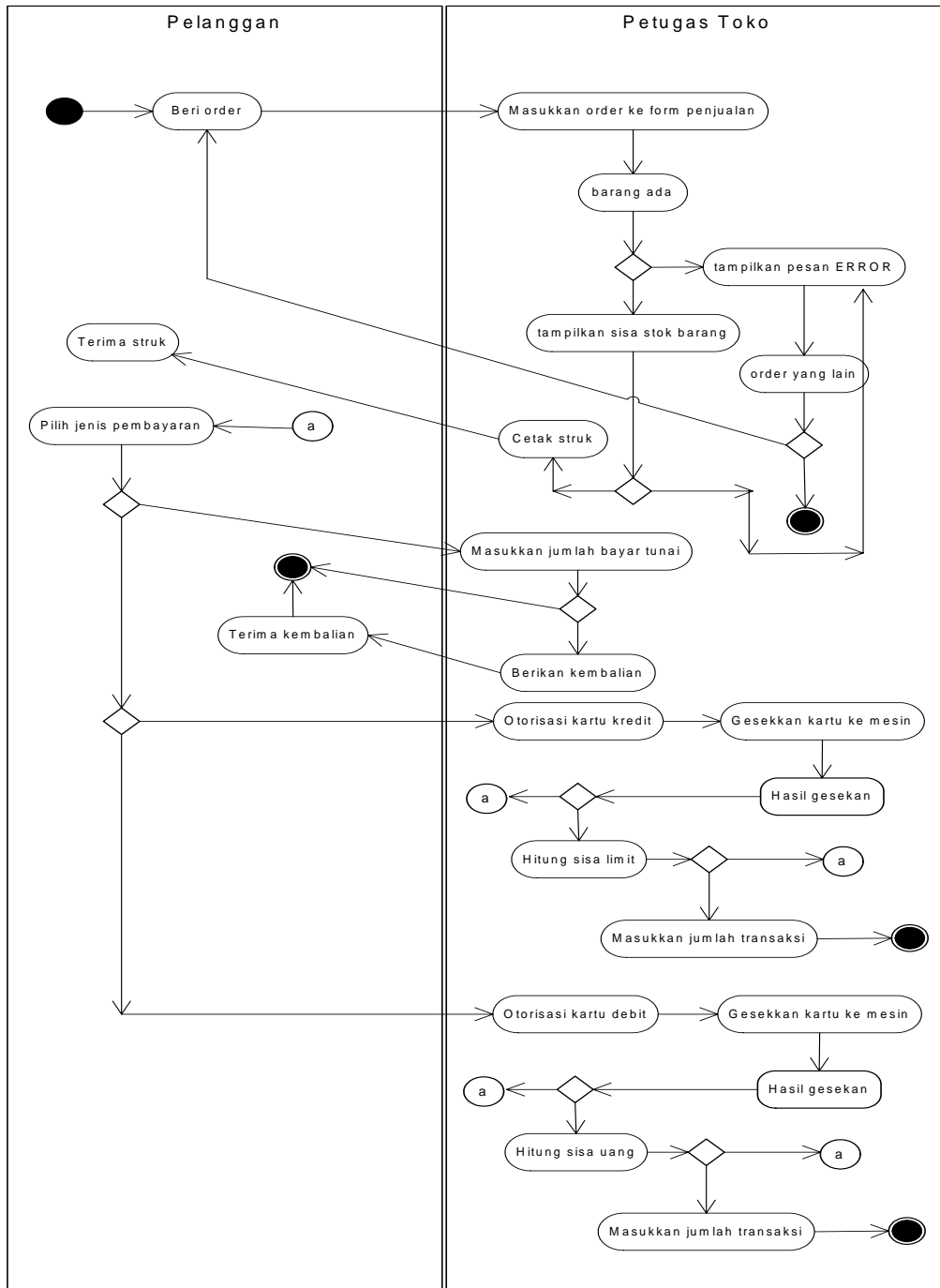


Gambar 4.3. Sequence diagram sistem galeri VCD

#### 4.1.5. Rancangan Activity

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas pada suatu proses. Diagram ini sangat mirip dengan *flowchart* karena kita dapat memodelkan prosedur logika, proses bisnis alur kerja. Perbedaan utamanya adalah flowchart digunakan untuk menggambarkan alur kerja dari sebuah sistem, sedangkan activity diagram dibuat untuk menggambarkan aktivitas dari actor.

Berikut adalah gambaran activity diagram dari aktivitas para actor-aktor yang ada pada sistem galeri toko VCD :

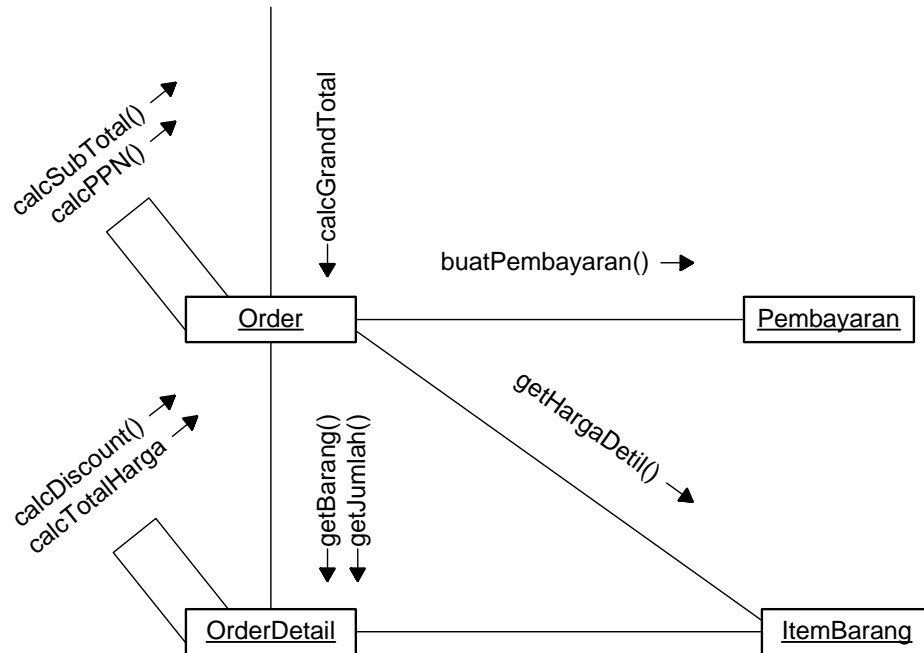


Gambar 4.4. Activity diagram sistem galeri VCD

#### 4.1.6. Rancangan Collaboration

Collaboration Diagram dibuat agar setiap proses transaksi yang terjadi akan terlihat lebih jelas setiap alur pesan antar obyek. Dasar dari pembuatan collaboration diagram ini adalah sequence diagram.

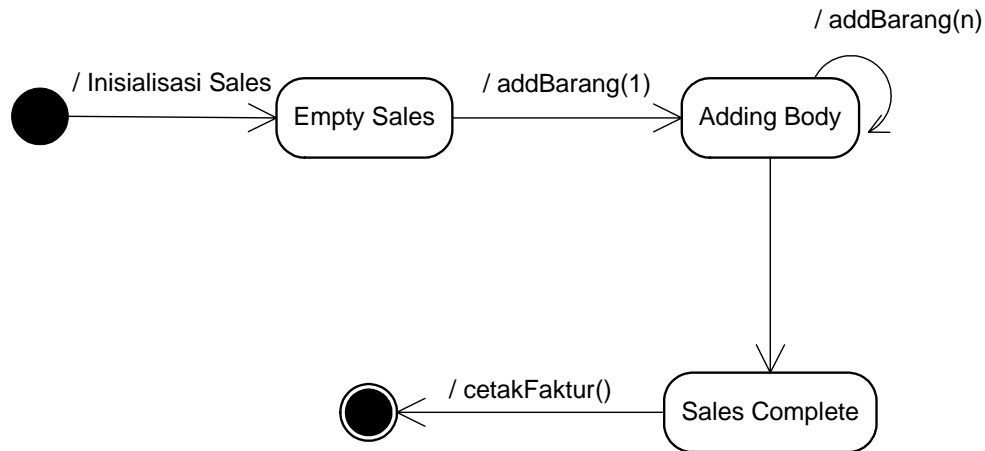
Berikut adalah rancangan collaboration diagram untuk sistem galeri toko VCD :



Gambar 4.5. Collaboration diagram sistem galeri VCD

#### 4.1.7. Rancangan State

State machine diagram biasanya digunakan untuk memodelkan perilaku dinamis suatu kelas atau obyek. Diagram ini memperlihatkan urutan state yang dilalui sebuah obyek, kejadian yang menyebabkan sebuah transisi dari suatu state atau aktivitas ke state atau aktivitas yang lain, dan aksi yang menyebabkan perubahan state atau aktivitas. Perlu diperhatikan bahwa tidak semua kelas akan mempunyai state yang menarik untuk dibahas. sebagai contoh pada sistem galeri toko VCD kita, state yang bisa digunakan untuk pelanggan mungkin terdiri dari prospek, aktif dan non-aktif. Hal ini tidak penting untuk dibicarakan karena sifatnya yang kaku dan akan berlaku seperti itu seterusnya. Sedangkan untuk kelas *Penjualan* akan butuh beberapa state selama daur hidupnya. Berikut adalah state diagram tersebut :



Gambar 4.6. Statechart diagram sistem galeri VCD

#### 4.1.8. Rancangan Component

Component diagram menggambarkan alokasi semua class dan obyek ke dalam komponen-komponen fisik di sebuah sistem software. Diagram ini memperlihatkan pengaturan dan kebergantungan di antara komponen-komponen software seperti dynamic library link (DLL), executable component dan lain-lain.

Pada kasus galeri toko VCD ini, diasumsikan akan memanfaatkan teknologi COM atau DCOM melalui DTS (Data Translation Service). DTS ini mempunyai layer-layer antara lain :

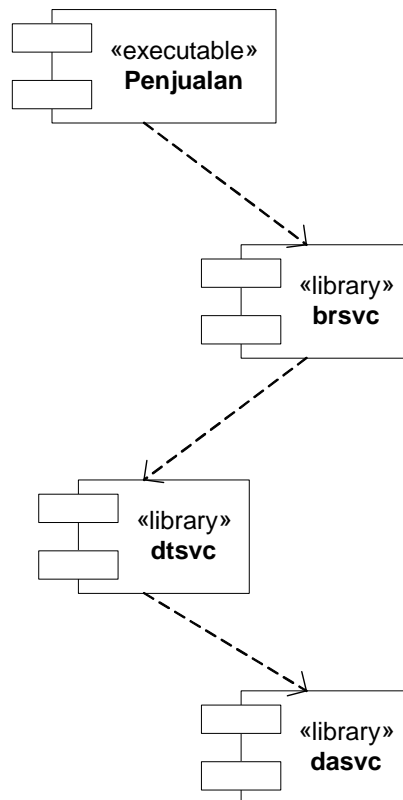
- Business Rule Services (BRSVC.DLL)
- Data Translation Services (DTSVC.DLL)
- Data Access Services (DASVC.DLL)

Selanjutnya kita bisa membuat infrastruktur aplikasi yang disebut dengan component diagram dimana componentnya antara lain adalah component yang menggambarkan logika presentasi (presentation logic) dan business context services. Component ini (Penjualan.EXE) mengandung semua *form* dan *user interface controller*. Penjualan.EXE sangat bergantung pada component BRSVC.DLL.

Component berikutnya memetakan logika aturan bisnis (*Logical Business Rule*). Component ini (BRSVC.DLL) akan menjadi ActiveX DLL dan mengandung semua class entity. BRSVC.DLL sangat bergantung pada component DTSVC.DLL.

Component berikutnya memetakan logika pengaksesan data (Logical Data Access). DASVC.DLL akan menjadi ActiveX DLL dan mengandung logika API database (seperti ADO ataupun DAO)

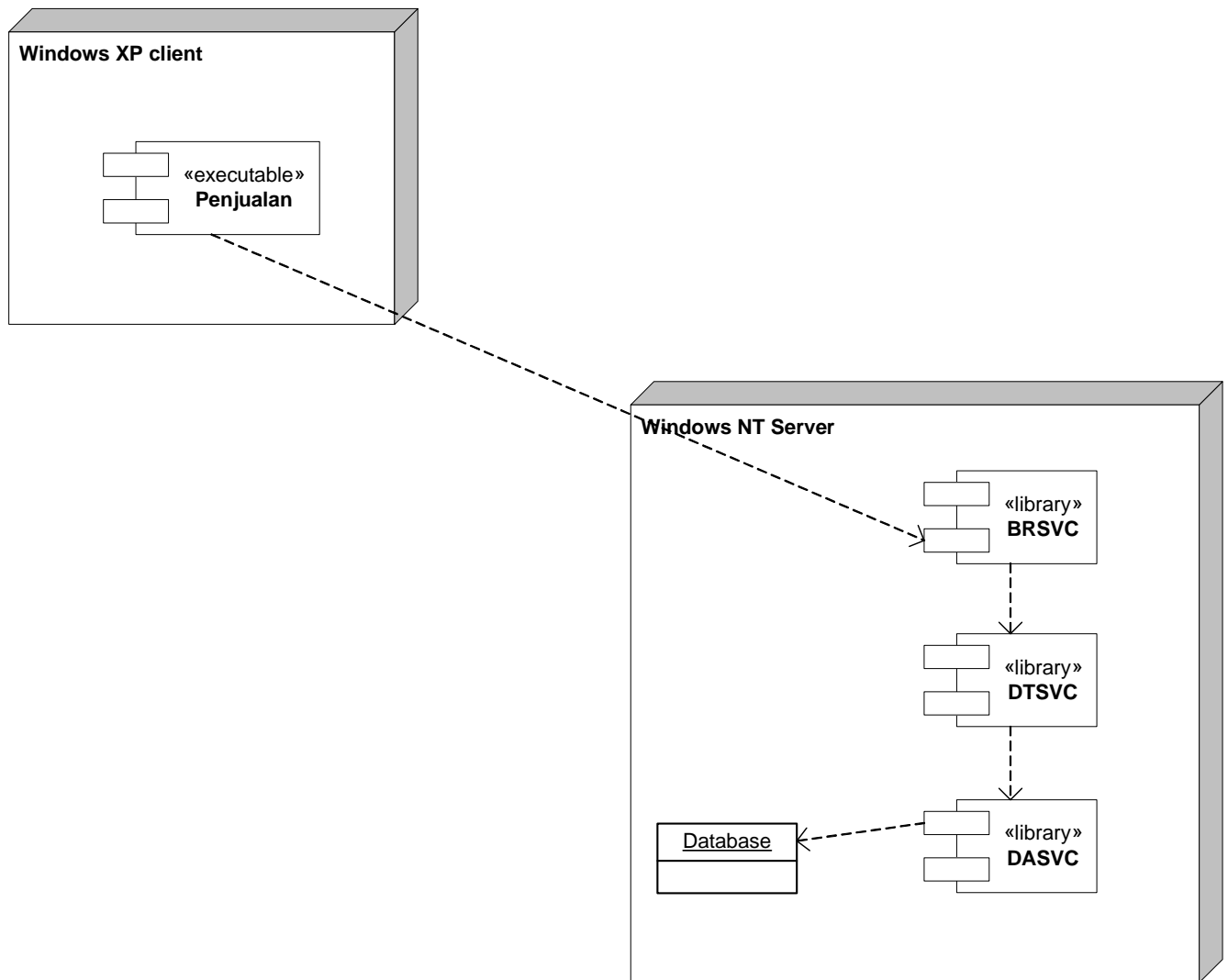
Dengan demikian bisa kita gambarkan Component diagramnya adalah seperti berikut :



Gambar 4.7. Component diagram sistem galeri VCD

#### 4.1.9. Rancangan Deployment

Rancangan deployment diagram tidak terlalu jauh berbeda karena deployment diagram pada intinya hanya menggambarkan lokasi atau dimana komponen-komponen berada. Secara fisik. Perhatikan gambar diagram tersebut dibawah ini :

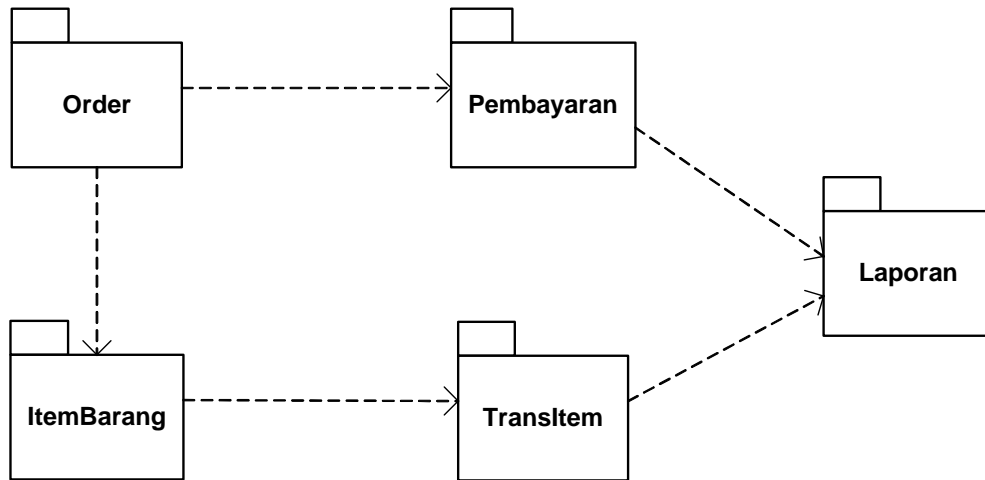


Gambar 4.8. Deployment diagram sistem galeri VCD

Terlihat jelas disini bahwa komponen executable terletak di client. Begitu juga pada server, kita bisa melihat komponen-komponen apa saja yang digunakan.

#### 4.1.10. Rancangan Package

Pada package diagram, setiap package memiliki ketergantungan terhadap package yang lainnya. Package diagram sangat berguna untuk mendisain sistem yang berskala besar supaya ketergantungan antar package dapat terlihat jelas. Berikut adalah package diagram untuk toko VCD :



Gambar 4.9. Package diagram sistem galeri VCD