



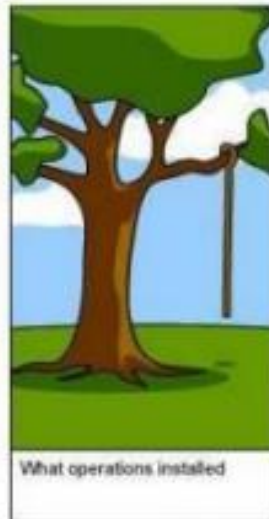
RPL – PERTEMUAN 11

PEMODELAN BERORIENTASI OBJEK

UNIFIED MODELING LANGUAGE(UML)

By : N. Nelis Febriani SM, M.Kom

APA YANG DIPIKIRKAN?



Tantangan Pengembangan Software



What the customer really
needed



What the customer got !

DEFINISI

Berfungsi :

- Untuk memodelkan perangkat lunak yang akan dibuat

Macam-macam Teknik :

- Pemrograman prosedural atau struktural (DFD) biasanya untuk skala yang lebih kecil (hanya 1 pengguna)
- Pemrograman berorientasi objek (UML) untuk skala yang lebih kompleks.

	Paradigm	Diagrams
1	Process-oriented Paradigm	Flowchart/Flowmap
2	Data-oriented Paradigm	DFD
3	Object-oriented Paradigm (data + process)	UML

PEMROGRAMAN BERORIENTASI OBJEK

- ✓ ***Object-oriented analysis (OOA)***— adalah sebuah pendekatan yang digunakan untuk :
 1. Menggunakan obyek yang sudah ada untuk digunakan kembali (*reuse*) atau diadaptasi untuk penggunaan baru
 2. Mendefinisikan obyek baru atau obyek yang dimodifikasi dan digabungkan dengan obyek yang sudah ada untuk membangun suatu aplikasi bisnis.

- ✓ ***Object modeling (Pemodelan Obyek)*** — Teknik untuk mengidentifikasi obyek yang ada dalam sistem dan relasi diantara obyek-obyek tersebut.

MENGAPA HARUS OO (OBJECT ORIENTED) ?

BEBERAPA ALASAN:

1. Pengembangan perangkat lunak itu sulit karena “perangkat lunak mudah diimpikan”
2. Kompleksitas pengembangan perangkat lunak yang terus bertumbuh, membutuhkan dukungan konsep yang lebih handal, guna ulang (*reusable*) dan natural
3. OO menawarkan tipe data abstrak, modularitas, pemodelan informasi, proses software untuk mengatasinya
4. Walaupun demikian, OO bukan jaminan sukses pengembangan perangkat lunak

KAPAN DAN DIMANA OO ?

BEBERAPA SITUASI UMUM:

1. Jika perangkat lunak (PL) yang dibangun cukup kompleks
2. Jika PL yang dibangun diperkirakan akan tumbuh makin kompleks di masa mendatang
3. Jika kita ingin membangun PL yang:
 - Berdasar pada komponen yang telah pernah ada sebelumnya (daur ulang)
 - Dapat dipergunakan kembali di masa mendatang (*reusable*)
4. Dan mungkin, kapanpun dan di manapun

PEMODELAN TERSTRUKTUR VS PEMODELAN BERORIENTASI OBJEK

- Pemrograman Terstruktur terdiri dari pemecahan masalah yang besar menjadi masalah yang lebih kecil dan seterusnya,
- pemrograman berorientasi objek terdiri dari pengkelompokan kode dengan data yang setiap objeknya berfungsi secara independen sehingga pada setiap perubahan kode tidak bergantung pada kode.
- Dengan menggunakan Pemrograman Berorientasi Objek maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut.
- Sedangkan untuk Pemrograman Terstruktur, menggunakan prosedur/ tata cara yang teratur untuk mengoperasikan data struktur.

Pemodelan Terstruktur vs Pemodelan Berorientasi Objek

Pemrograman Terstruktur	Pemrograman Berorientasi Objek
Memecah program dalam fungsi dan data	Menggabungkan fungsi dan data dalam kelas - kelas atau objek - objek
Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan)	Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat) dan Polymorphism (perbedaan bentuk dan perilaku)
Struktur program rumit karena berupa urutan proses dan fungsi-fungsi	Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.
Re-use kode program kurang	Kode program sangat re-usable. object dan class dapat digunakan berkali-kali, sehingga dapat menghemat space memori.
Efektif digunakan untuk menyelesaikan masalah kecil dan tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error	Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok - kelompok kecil, sesuai dengan fungsinya
Mudah diawal, namun kompleks diproses selanjutnya	Sulit diawal (karena harus membuat class) namun selanjutnya akan terasa mudah dan cepat
Eksekusi lebih lambat karena setiap perintah dikerjakan berurutan	Eksekusi lebih cepat karena dieksekusi bersamaan, program hanya mengatur Objek, properties dan method-nya saja

TAHAPAN PEMODELAN BERORIENTASI OBJEK

- Memahami dan mendefinisikan konteks dan metode penggunaan sistem
- Merancang arsitektur sistem
- Mengidentifikasi objek utama sistem
- Mengembangkan model desain
- Menspesifikasikan interface objek

KEUNTUNGAN PENGGUNAAN PEMODELAN BERORIENTASI OBJEK

- ▶ Kemampuan untuk menangani tipe-tipe data dan masalah-masalah yang lebih kompleks dan lebih sulit.
- ▶ Memperbaiki komunikasi antara pengguna, analisis, perancangan dan pemrogram.
- ▶ Meningkatkan derajat konsistensi antara tahap analisis, perancangan, serta kegiatan pemrograman karena sama untuk setiap tahap itu.
- ▶ Ketangguhan dan ketegaran sistem (robustness).
- ▶ Kemampuan untuk menggunakan ulang hasil-hasil analisis, perancangan serta pemrograman (reusable component) pada suatu proyek ke proyek lainnya.
- ▶ Meningkatkan konsistensi antara model-model yang dikembangkan selama analisis, perancangan, serta pemrograman berorientasi objek.

UNIFIED MODELLING LANGUAGE (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem informasi atau perangkat lunak.

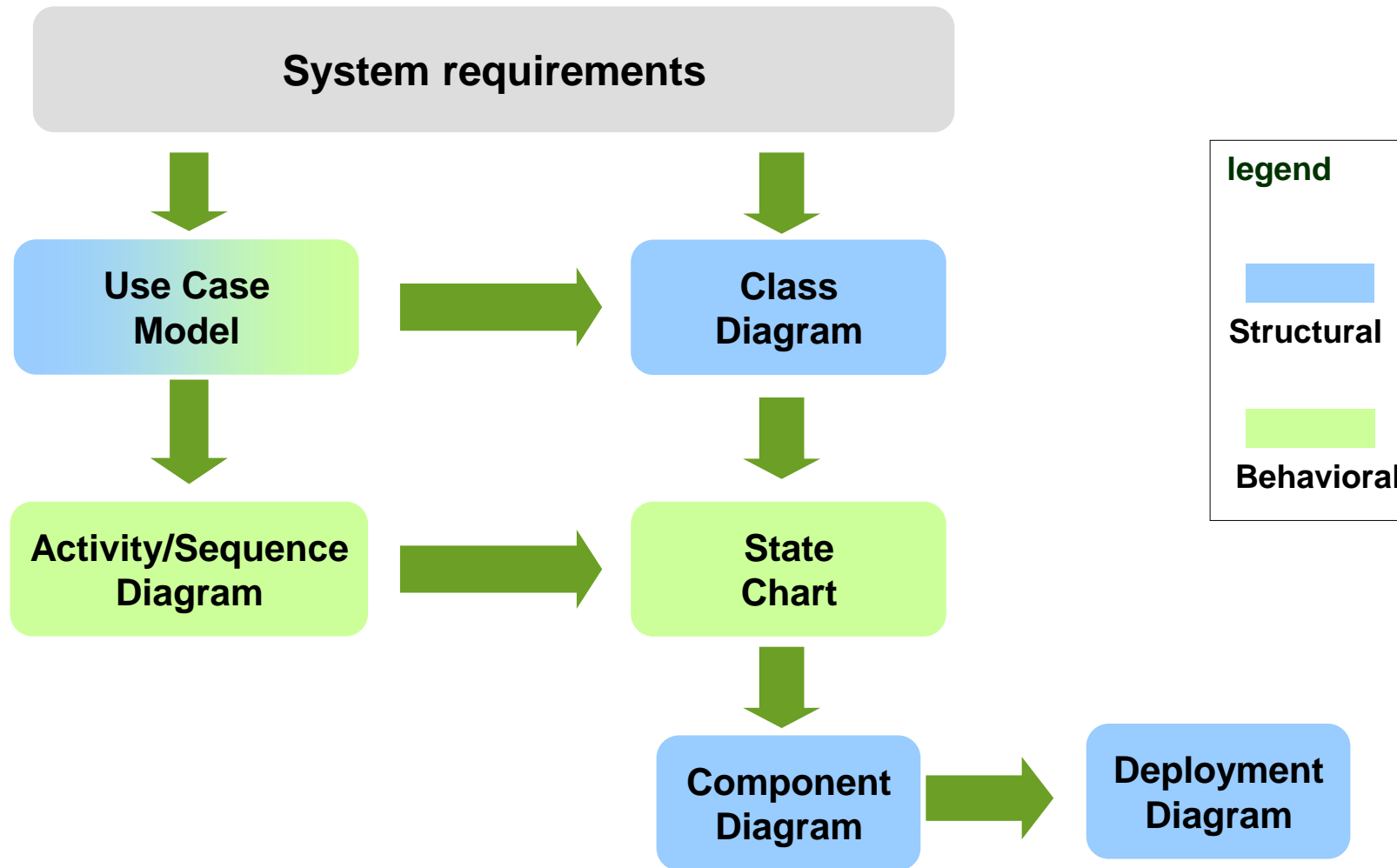
UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

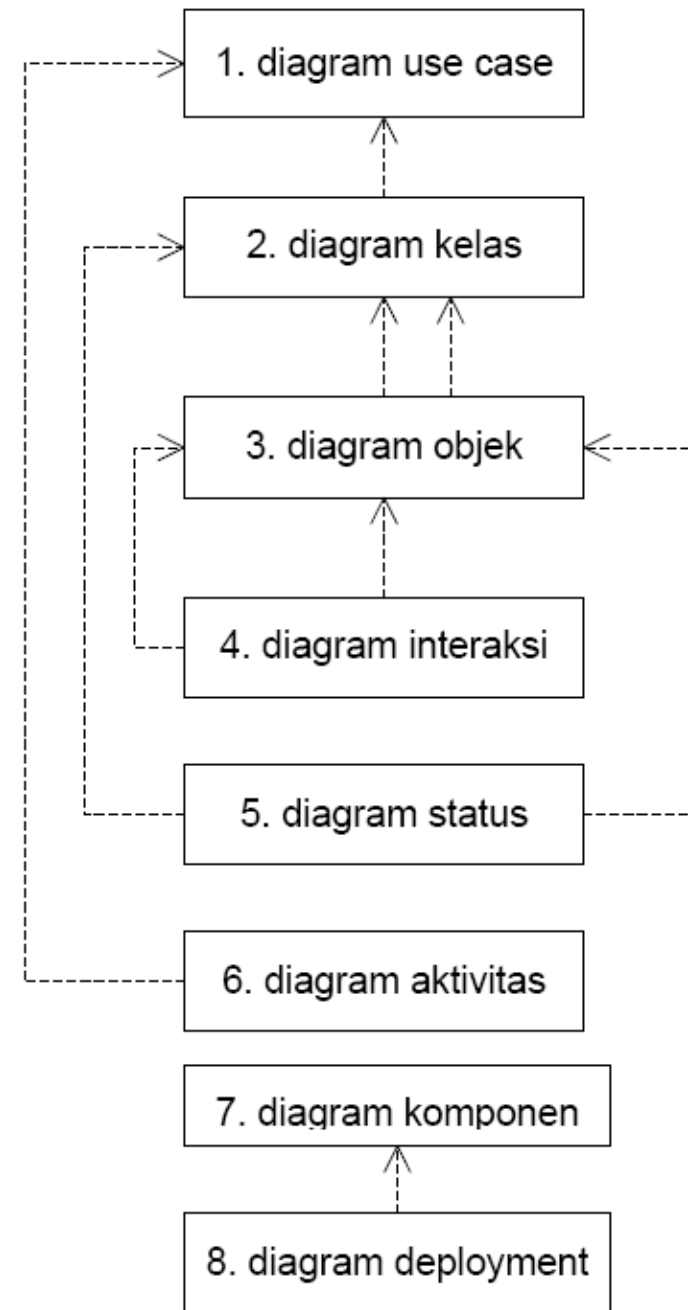
ANALYSIS AND DESIGN PROCESS

Zachman Framework



- *Structural UML diagram* untuk menganalisis dan menggambarkan struktur dari suatu sistem atau proses
- *Behavioral diagram* menunjukkan apa yang harus terjadi dalam suatu sistem dan menggambarkan bagaimana objek berinteraksi satu sama lain untuk menciptakan sistem yang berfungsi.

LANGKAH-LANGKAH PEMBUATAN UML



USE CASE

Pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat.



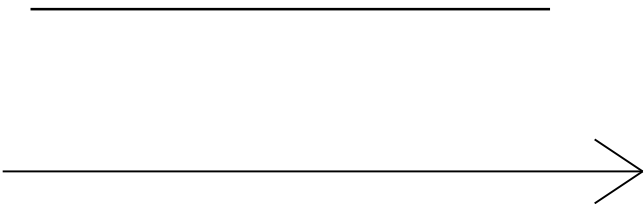
Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat

Terdapat beberapa simbol dalam menggambarkan diagram use case, yaitu *use cases*, aktor dan relasi.

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

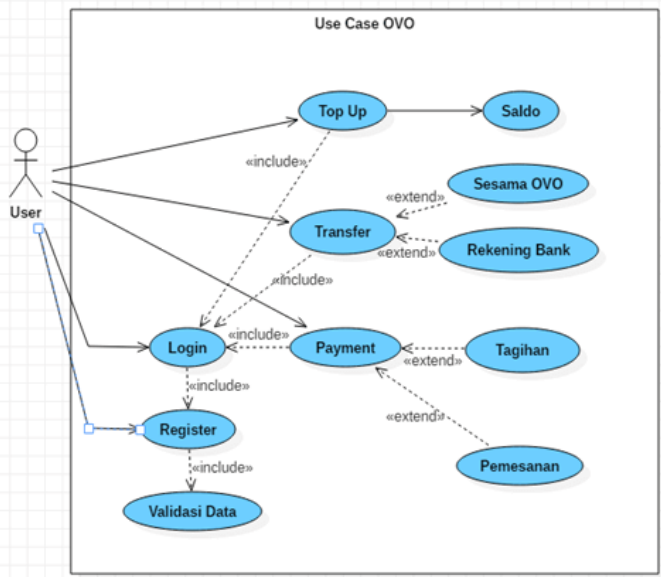
SIMBOL USE CASE

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
Aktor 	Orang, Proses atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / association 	Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor. association antara actor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila actor berinteraksi secara pasif dengan system anda

SIMBOL USE CASE

Simbol	Deskripsi
Ekstensi / extend <<extend>>>	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.
Generalisasi / Generalization ————>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Include <<include>>>	Relasi Use Case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

SIMBOL USE CASE

Simbol	Deskripsi
<p>boundary boxes</p>  <pre>graph LR User((User)) subgraph "Use Case OVO" TopUp((Top Up)) Transfer((Transfer)) Login((Login)) Register((Register)) ValidasiData((Validasi Data)) Saldo((Saldo)) SesamaOVO((Sesama OVO)) RekeningBank((Rekening Bank)) Payment((Payment)) Tagihan((Tagihan)) Pemesanan((Pemesanan)) end User --> TopUp User --> Transfer User --> Login User --> Register TopUp --> Saldo TopUp -.-> «include» Transfer Transfer -.-> «extend» SesamaOVO Transfer -.-> «extend» RekeningBank Login -.-> «include» Payment Register -.-> «include» Login ValidasiData -.-> «include» Register Payment -.-> «extend» Tagihan Payment -.-> «extend» Pemesanan</pre>	<ul style="list-style-type: none">• Digambarkan dengan kotak disekitar use case, untuk menggambarkan jangkauan system anda (scope of of your system).• Biasanya digunakan apabila memberikan beberapa alternative system yang dapat dijadikan pilihan• System boundary boxes dalam penggunaannya optional

MENEMUKAN AKTOR

- SIAPA yang akan menggunakan system?
- APAKAH system tersebut akan memberikan NILAI bagi actor?
- Tidak semua actor adalah MANUSIA.
- Oleh karena itu, gambarlah diagram use case yang berfokus pada nilai yang akan diberikan kepada aktor

CONTOH PENCARIAN AKTOR

SISTEM PENCATATAN PENJUALAN DI SUPERMARKET

Pertanyaan	Analisis
Siapa sajakah yang berinteraksi dengan system pencatatan penjualan di supermarket ?	<ul style="list-style-type: none">• Bagian yang akan mencatat penjualan barang• Bagian yang ingin tahu berapa besar keuntungan yang didapatkan• Bagian yang ingin tahu berapa banyak produk yang berkurang
Peran apa saja yang terlibat?	Kasir, Manajer, bagian Gudang.
Nilai apa sajakah yang akan diberikan system kepada actor?	<p>Nilai bagi kasir :</p> <ul style="list-style-type: none">• Ia akan mendapatkan struk belanja• Lama aktivitas kerja akan terekam kedalam system <p>Nilai bagi manajer</p> <ul style="list-style-type: none">• Ia perlu mengetahui laporan keuntungan dalam rentang waktu tertentu <p>Nilai bagi bagian Gudang</p> <ul style="list-style-type: none">• Ia perlu mengetahui produk apa saja yang berkurang.

CONTOH PENCARIAN AKTOR

SISTEM PENCATATAN PENJUALAN DI SUPERMARKET

Pertanyaan	Analisis
Apakah system pencatatan penjualan bergantung pada sesuatu?	<p>Printer</p> <ul style="list-style-type: none">• Untuk mencetak Struk <p>EDC</p> <ul style="list-style-type: none">• Untuk menarik sejumlah uang pada account seseorang



Kasir



Manajer



Bagian
Gudang



Printer



EDC

MENEMUKAN USE CASE

Untuk menemukan use case, lihat dari sudut pandang actor :

- Informasi apa sajakah yang akan didapatkan actor dari system?
- Apakah ada kejadian dari system yang perlu diberitahukan ke actor?

Kemudian dari sudut pandang system :

- Apakah ada informasi yang perlu disimpan atau diambil dari system?
- Apakah ada informasi yang harus dimasukkan oleh aktor?

STUDI KASUS

Sistem informasi manajemen perpustakaan merupakan sebuah sistem informasi untuk mengelola informasi yang diperlukan dalam suatu perpustakaan yang meliputi pendaftaran pustaka, anggota, dan proses peminjaman pustaka. Aturan perpustakaan yang harus diatasi pada sistem informasi manajemen perpustakaan yang akan dimodelkan adalah sebagai berikut:

1. Pustaka dapat memiliki lebih dari satu pengarang
2. Anggota dapat memiliki lebih dari satu nomor telepon
3. Seorang anggota dapat melakukan sebuah peminjaman dalam satu waktu dan boleh lebih dari satu pustaka
4. Seorang anggota dapat mengembalikan pustaka yang dipinjam tidak dalam waktu yang bersamaan walaupun pustaka-pustaka itu dipinjam pada waktu yang sama.
5. Pengunjung yang bukan anggota tidak diperbolehkan meminjam pustaka.
6. Proses pendaftaran pustaka, anggota, dan peminjaman dilakukan oleh petugas perpustakaan.
7. Anggota dan pengunjung dapat melakukan pencarian pustaka.

STUDI KASUS

Sistem informasi yang akan dibuat adalah aplikasi berbasis web. Manajemen perpustakaan meliputi fungsi-fungsi sebagai berikut:

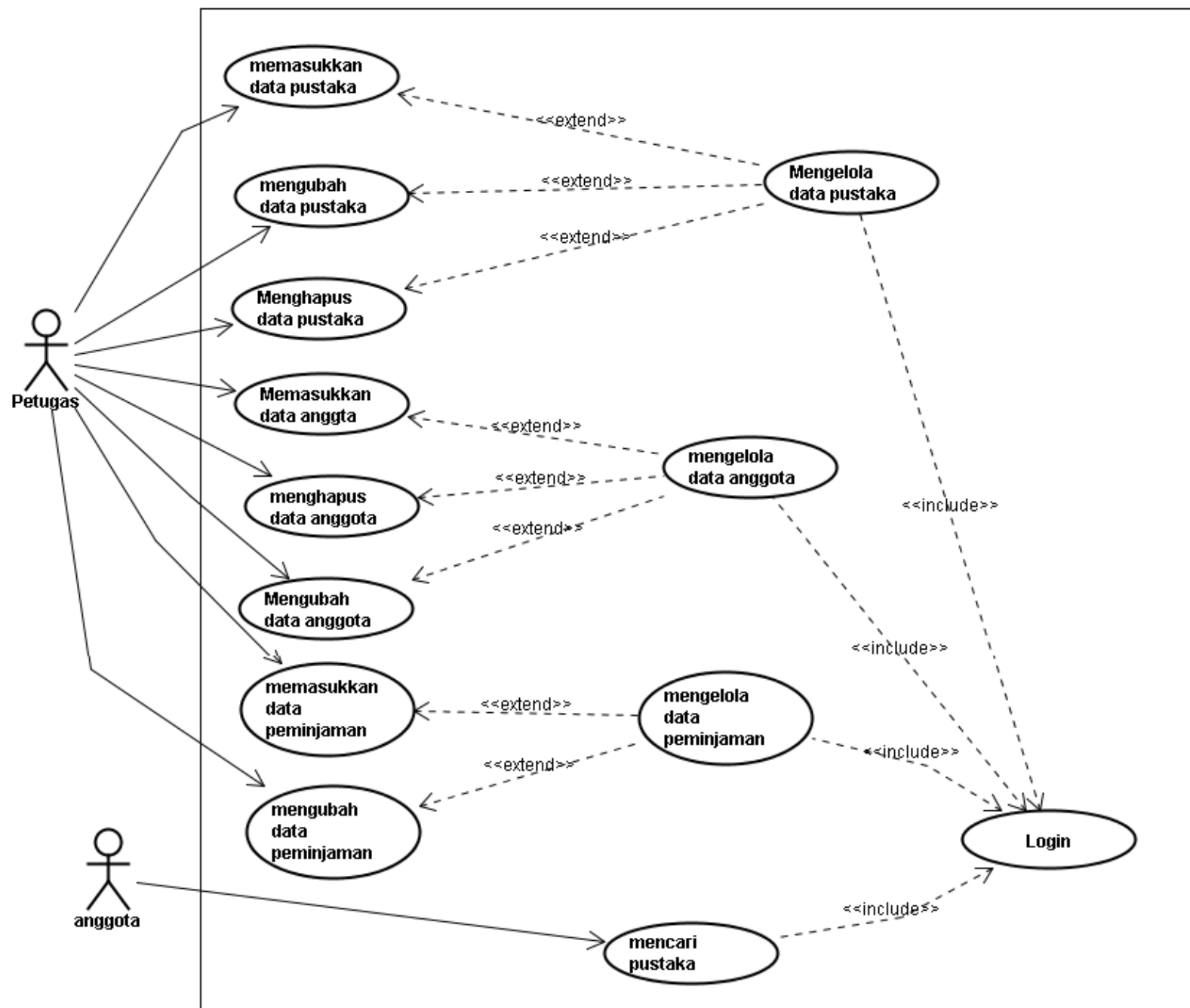
1. Mengelola data pustaka, meliputi:
 - a. Memasukkan data pustaka
 - b. Mengubah data pustaka
 - c. Menghapus data pustaka
2. Mengelola data anggota, meliputi:
 - a. Memasukkan data anggota
 - b. Mengubah data anggota
 - c. Menghapus data anggota
3. Mengelola data peminjaman, meliputi:
 - a. Memasukkan data peminjaman
 - b. Mengubah data peminjaman (mekanisme pengembalian pustaka)
4. Mencari pustaka

Jawab:

Pemecahan studi kasus tahap pertama yaitu melakukan pencarian aktor. mulailah bertanya dengan SIAPA, PERAN dan NILAI apa yang akan didapatkan.

No	Aktor	Deskripsi
1.	Petugas perpustakaan	orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data pustaka, anggota, dan proses pemiinjaman pustaka
2.	Anggota/pengunjung perpustakaan	anggota adalah orang yang diperbolehkan meminjam pustaka sesuai dengan hak aksesnya, sedangkan pengunjung hanya memiliki hak akses melihat pustaka dan membaca di perpustakaan tanpa memiliki hak untuk meminjam pustaka.

USE CASE SISTEM INFORMASI MANAJEMEN PERPUSTAKAAN



Tahap selanjutnya adalah menemukan use case. Mulailah bertanya dengan INFORMASI apa yang akan diberikan oleh sistem kepada aktor.

No	Use case	Deskripsi
1.	Memasukkan data pustaka	merupakan proses memasukkan data pustaka ke dalam basis data
2.	Memasukkan data anggota	merupakan proses memasukkan data anggota ke dalam basis data
3.	Memasukkan data peminjaman	merupakan proses memasukkan data peminjaman ketika ada anggota yang meminjam pustaka
4.	Mencari pustaka	mencari pustaka berdasarkan judul, nama pengarang, jenis, dan kode pustaka dimana akan menampilkan data pustaka yang dicari

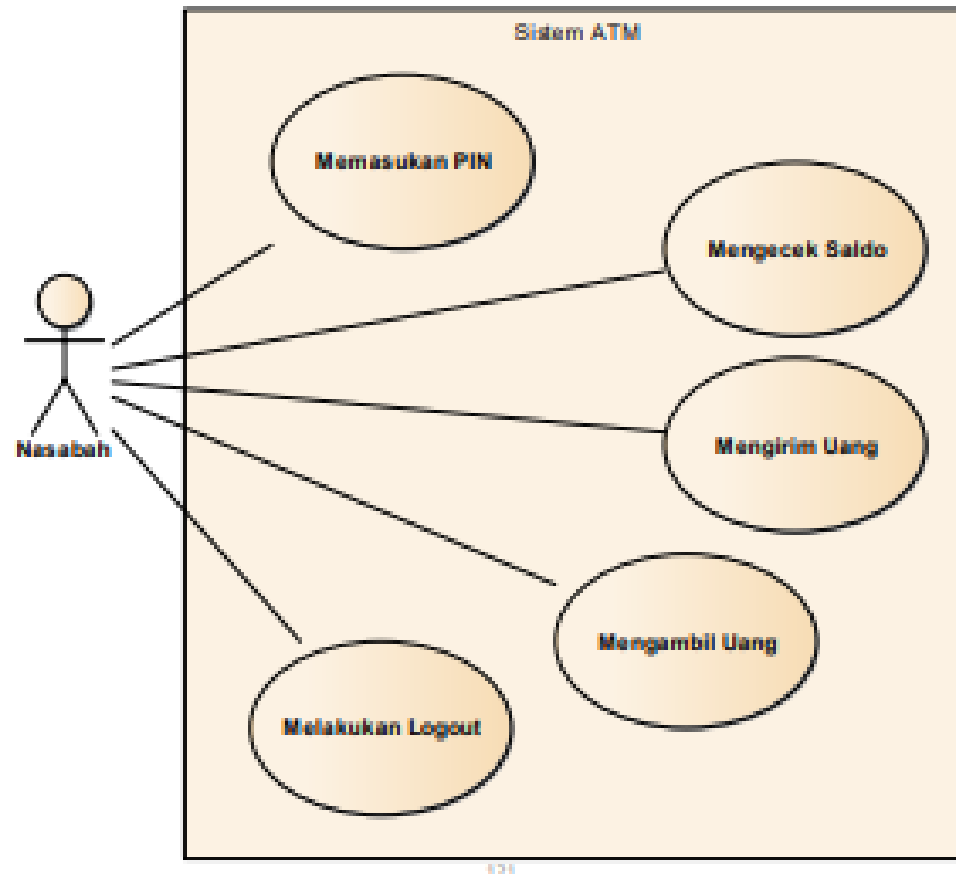
Tahap ketiga adalah membuat skenario per-use case.

Memasukkan data pustaka

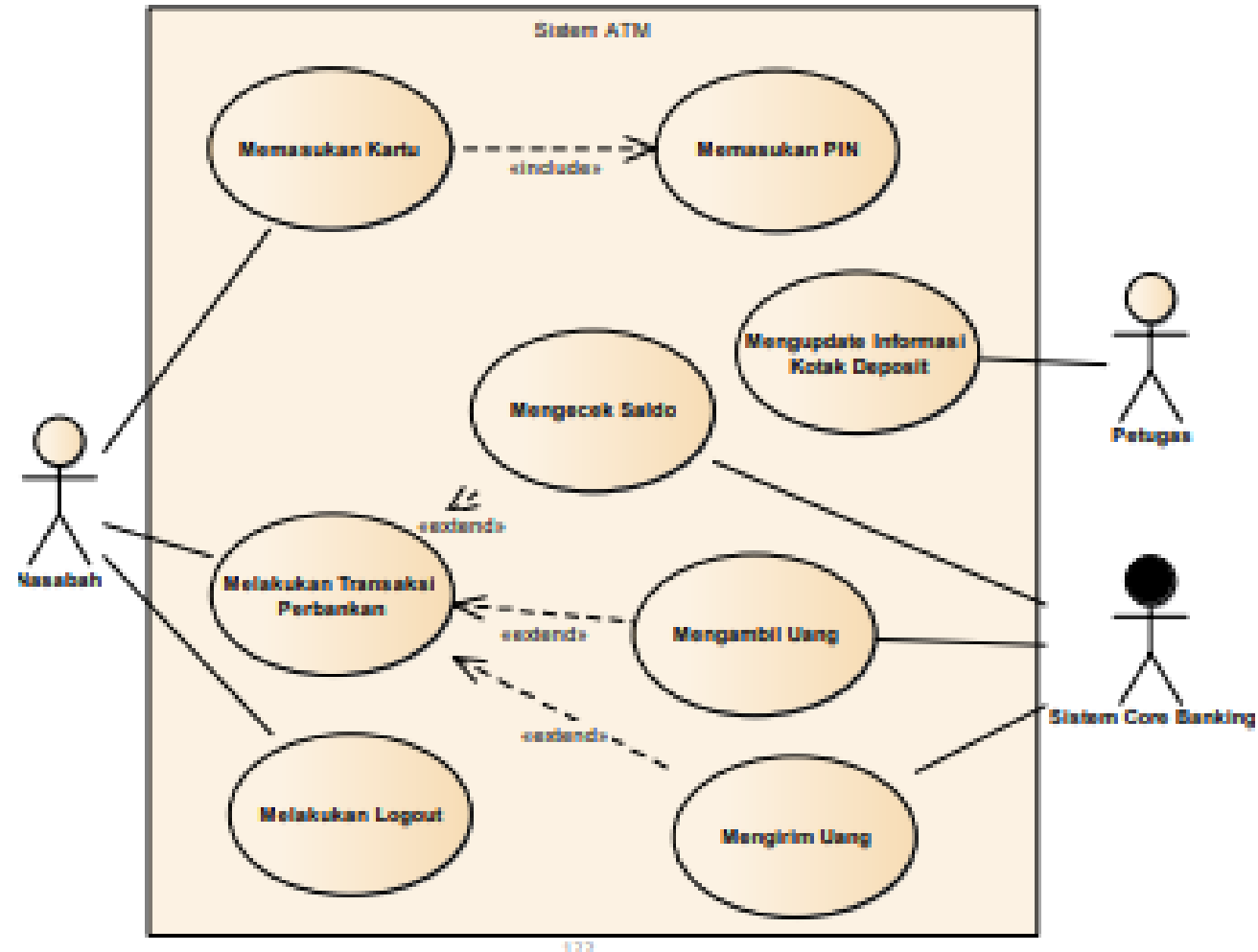
Aksi Aktor	Reaksi Sistem
1. Memasukkan data pustaka seperti judul buku, penerbit, tahun terbit, pengarang, jumlah halaman, kondisi buku di menu memasukkan pustaka	
2. Menekan tombol “Simpan”	
	3. Mengecek valid tidaknya data masukan
	4. Jika data pustaka yang dimasukkan valid, maka data pustaka akan disimpan di database dan akan menampilkan pesan “sukses disimpan”
Alur alternatif No 4 a. Jika data pustaka yang dimasukkan tidak valid, maka akan menampilkan pesan “tidak sukses disimpan”	

Aksi Aktor	Reaksi Sistem
1. Nasabah memasukkan kartu ATM ke dalam Mesin ATM	
	2. Melakukan pengecekan apakah kartu yang dimasukan valid atau tidak
	3. Apabila valid, sistem akan menampilkan pesan “Masukan PIN”, Apabila tidak valid maka sistem akan mengeluarkan kartu tersebut.
4. Nasabah memasukkan PIN ATM lalu menekan tombol “Enter”	
	5. Melakukan pengecekan apakah PIN ATM yang dimasukan valid atau tidak
	6. Apabila valid, sistem akan berlanjut ke menu transaksi. Apabila tidak valid maka sistem akan menampilkan pesan “PIN Salah”

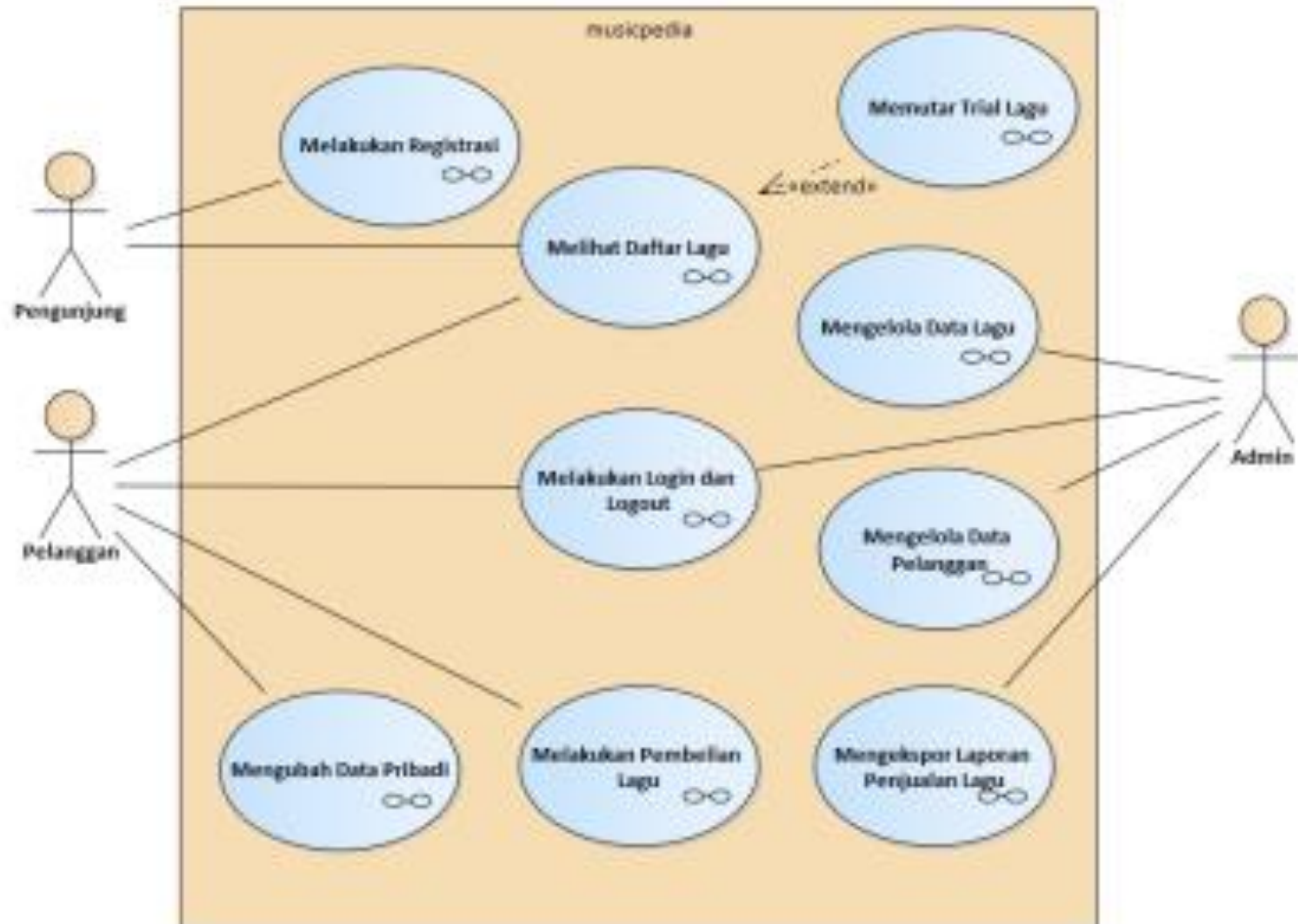
Use Case Diagram Sistem ATM (versi Sederhana)



Use Case Diagram Sistem ATM (Versi Kompleks Include dan Extends)



Use Case Diagram Spotify



LATIHAN DI KELAS

MEMBUAT USE CASE DIAGRAM

- ✓ Koperasi UHAMKA adalah sebuah koperasi yang mengelola simpan pinjam bagi para anggotanya, berikut ini adalah kegiatan yang dilakukan oleh bagian Kredit dalam menangani pemberian pinjaman bagi para anggotanya.
- ✓ Setiap kali bagian kredit akan memberikan pinjaman kepada Anggota maka Anggota diharuskan mengisi Formulir Permohonan Pinjaman yang berisi *Nomor FPP, Tanggal Permohonan, Nomor Anggota, Nama Anggota, Jumlah Permohonan dan Keperluan*. Yang kemudian oleh Bagian Kredit dicatat dan disimpan kedalam Arsip FPP. Berdasarkan Arsip FPP tersebut Bagian Kredit membuat Bukti Peminjaman yang diberikan kepada Anggota yang berisi No. BP, tgl BP, Nomor Anggota, Nama Anggota, Jumlah Realisasi, Lama Angsuran, Jumlah Angsuran dan Bunga.
- ✓ Setiap Bulan Anggota diharuskan membayar Angsuran sejumlah Angsuran yang disepakati pada saat Peminjaman yang kemudian oleh bagian Kredit dicatat dan direkam kedalam Arsip Angsuran. Berdasarkan Arsip Angsuran tersebut bagian Kredit membuat Bukti Angsuran yang diberikan kepada Anggota yang berisi No. BA, Tanggal BA, No. BP, Jumlah Angsur dan Bunga
- ✓ Pada akhir bulan Bagian Kredit selalu membuat Laporan Peminjaman dan Laporan Angsuran yang diberikan Kepada Ketua Koperasi.

Pertanyaan :

Buatlah Use case diagram dan skenario usecase dari data diatas !

LATIHAN DI KELAS

MEMBUAT USE CASE DIAGRAM

PT. Nusantara adalah sebuah perusahaan yang bergerak dibidang penjualan Tunai barang-barang elektronik. Semua transaksi di perusahaan masih dilakukan secara manual. Berikut ini adalah kegiatan kegiatan yang dilakukan oleh **bagian Penjualan** dalam melaksanakan transaksi penjualan Barang di dalam perusahaan.

1. Pemesanan barang

Setiap kali Bagian penjualan akan menjual barang ia selalu menerima **surat pesanan** dari **pelanggan**. Berdasarkan Surat pesanan tersebut bagian penjualan kemudian mencatat dan merekamnya kedalam Arsip Surat Pesanan. Berdasarkan Arsip surat pesanan tersebut, bagian penjualan membuatkan **Faktur** dan **Surat Jalan** yang dikirimkan kepada **Pelanggan** sebagai bukti bahwa barang yang dipesan sudah terealisasi dan rangkapnya disimpan sebagai Arsip Faktur dan Arsip Surat Jalan.

2. Pembuatan Kwitansi

Apabila **Faktur** dan Surat Jalan sudah sampai ditempat pelanggan, maka pelanggan megirimkan Pembayaran yang kemudian oleh **bagian penjualan** dibuatkan **Kwitansi** yang dibuat berdasarkan Arsip Faktur yang kemudian diserahkan kepada **pelanggan** sebagai bukti pembayaran dan rangkapnya disimpan kedalam Arsip Kwitans

3. Pembuatan Laporan

Setiap akhir bulan **Bagian Penjualan** selalu membuat **Laporan Penjualan** berdasarkan Arsip Faktur dan **Laporan Pesanan** berdasarkan Arsip Pesanan dan **Laporan Pengiriman** berdasarkan Arsip Surat Jalan yang ditujukan kepada **Kepala Bagian Penjualan**

Pertanyaan :

Buatlah Use case diagram beserta skenario usecase dari data diatas !



SELESAI
TERIMA KASIH |