

# USE CASE DIAGRAM

---

# UML

- UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi object dan design berorientasi object (OOA&D) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an
- UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML ini akan mencakup lebih luas daripada OOA&D
- UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat

# UML

- UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya
- Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *object*. UML dibuat oleh *Grady Booch*, *James Rumbaugh*, dan *Ivar Jacobson* di bawah bendera *Rational Software Corp*
- UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan

# System Analysis and Design with UML

## 1. System Analysis

1. Business Process Identification
  - **Use Case Diagram**
2. Business Process Modeling
  - **Activity Diagram** or Business Process Modeling Notation (BPMN)
3. Business Process Realization
  - **Sequence Diagram** (Buat untuk setiap use case dengan menggunakan pola **Boundary-Control-Entity**)

## 2. System Design

1. Program Design
  1. **Class Diagram** (Gabungkan **Boundary-Control-Entity** Class dan susun story dari sistem yang dibangun)
  2. **Package Diagram** (Gabungan class yang sesuai, boleh menggunakan pola B-C-E)
  3. **Deployment Diagram** (arsitektur software dari sistem yang dibangun)
2. **User Interface Design** (Buat UI design dari **Boundary Class**)
3. **Entity-Relationship Model** (Buat ER diagram dari **Entity Class**)

# UML DIAGRAMS

- Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Adapun jenis diagram dalam UML antara lain:

1. *Class diagram*

2. *Use-case diagram*

3. *Behavior diagrams:*

- 3.1. *Interaction diagram:*

- 3.1.1. *Sequence diagram*

- 3.1.2. *Collaboration diagram*

- 3.2. *State chart diagram*

- 3.3. *Activity diagram*

4. *Implementation diagram:*

- 4.1. *Component diagram*

- 4.2. *Deployment diagram*

# USE CASE DIAGRAM

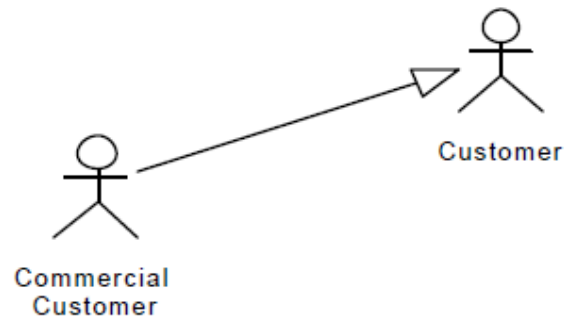
- Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem.
- *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*.
- *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

# ACTOR

- Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor* dimana *actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
- Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem, *actor* hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*.
- *Actor* digambarkan dengan *stick man*
- *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*

# ACTOR

- Ada beberapa kemungkinan yang menyebabkan *actor* tersebut terkait dengan sistem antara lain:
  - ✓ Yang berkepentingan terhadap sistem dimana adanya arus informasi baik yang diterimanya maupun yang dia inputkan ke sistem.
  - ✓ Orang ataupun pihak yang akan mengelola sistem tersebut.
  - ✓ *External resource* yang digunakan oleh sistem.
  - ✓ Sistem lain yang berinteraksi dengan sistem yang akan dibuat.



**Gambar 1.1 Actor**



# USE CASE

- *Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
- *Use case diagram* adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.

# USE CASE

- **Cara menentukan Use Case dalam suatu sistem:**
  - Pola perilaku perangkat lunak aplikasi.
  - Gambaran tugas dari sebuah *actor*.
  - Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*.
  - Apa yang dikerjakan oleh suatu perangkat lunak (bukan bagaimana cara mengerjakannya).



UseCase

**Gambar 1.2 UseCase**

# RELASI USE CASE

- Ada beberapa relasi yang terdapat pada *use case diagram*:
  1. *Association*, menghubungkan link antar element.
  2. *Generalization*, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
  3. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
  4. *Aggregation*, bentuk assosiation dimana sebuah elemen berisi elemen lainnya.

# RELASI USE CASE

- Tipe relasi/ *stereotype* yang mungkin terjadi pada *use case diagram*:
  1. **<<include>>** , yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
  2. **<<extends>>**, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.
  3. **<<communicates>>**, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah *communicates association* . Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use case*.

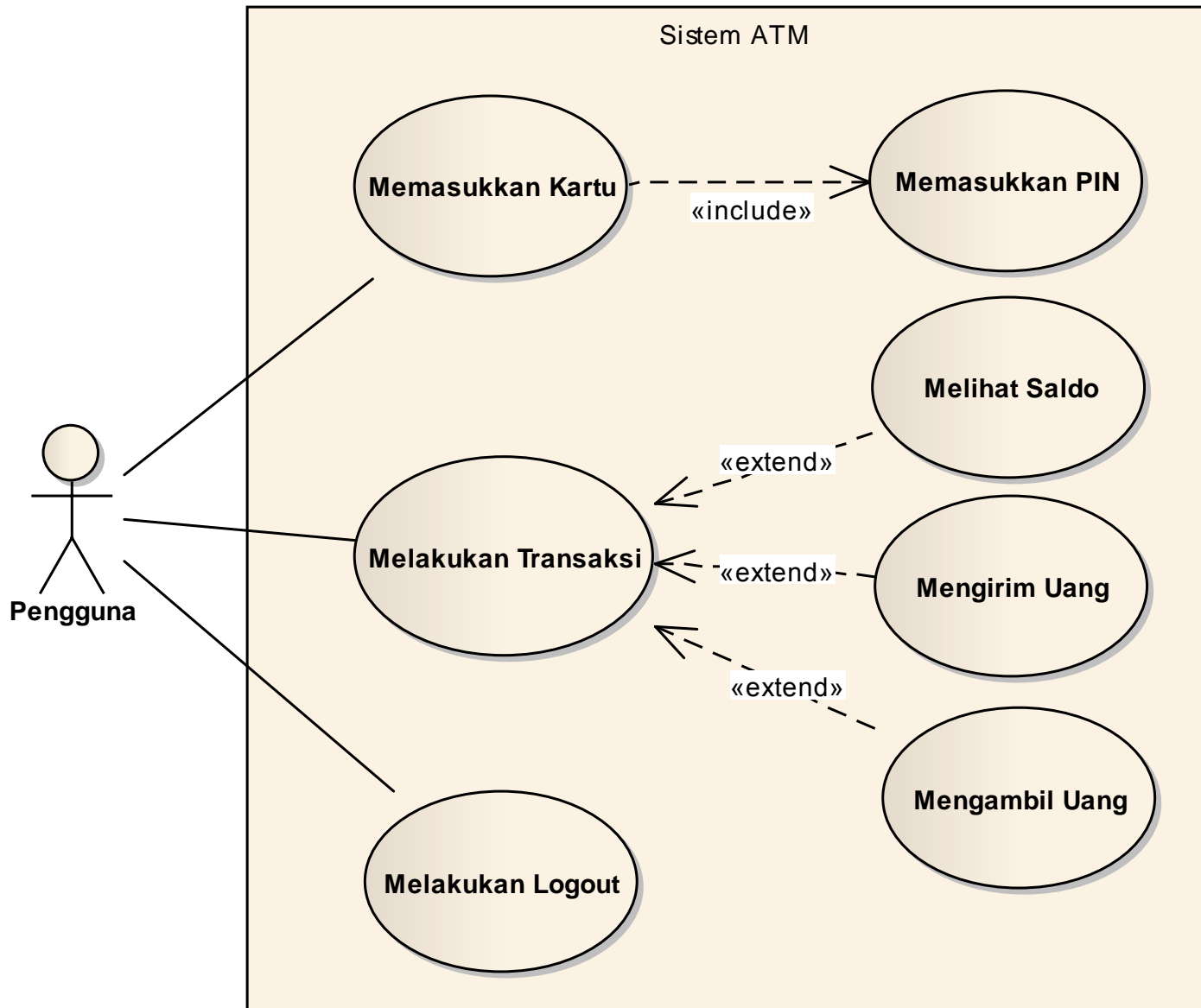
# Contoh Kasus

- Berikut ini adalah contoh dari sebuah studi kasus yang menangani Aplikasi pada sebuah ATM dengan skenario sbb:
  - ✓ Sebuah bank mengoperasikan ATM dan mengelola banyak tabungan, setiap nasabah memiliki setidaknya satu rekening tabungan pada satu bank tertentu.
  - ✓ Setiap tabungan dapat diakses melalui kartu debit. Proses utama sistem ATM berkomunikasi dengan pusat komputer dan didesain untuk menangani beberapa transaksi.
  - ✓ Setiap transaksi menunjuk sebuah tabungan tertentu. Suatu transaksi akan menghasilkan satu dari dua hal berikut: transaksi diterima atau mengeluarkan pesan penolakan transaksi".
  - ✓ Untuk melakukan sebuah transaksi akan melalui dua tahap: pengecekan tabungan dan pemrosesan transaksi.

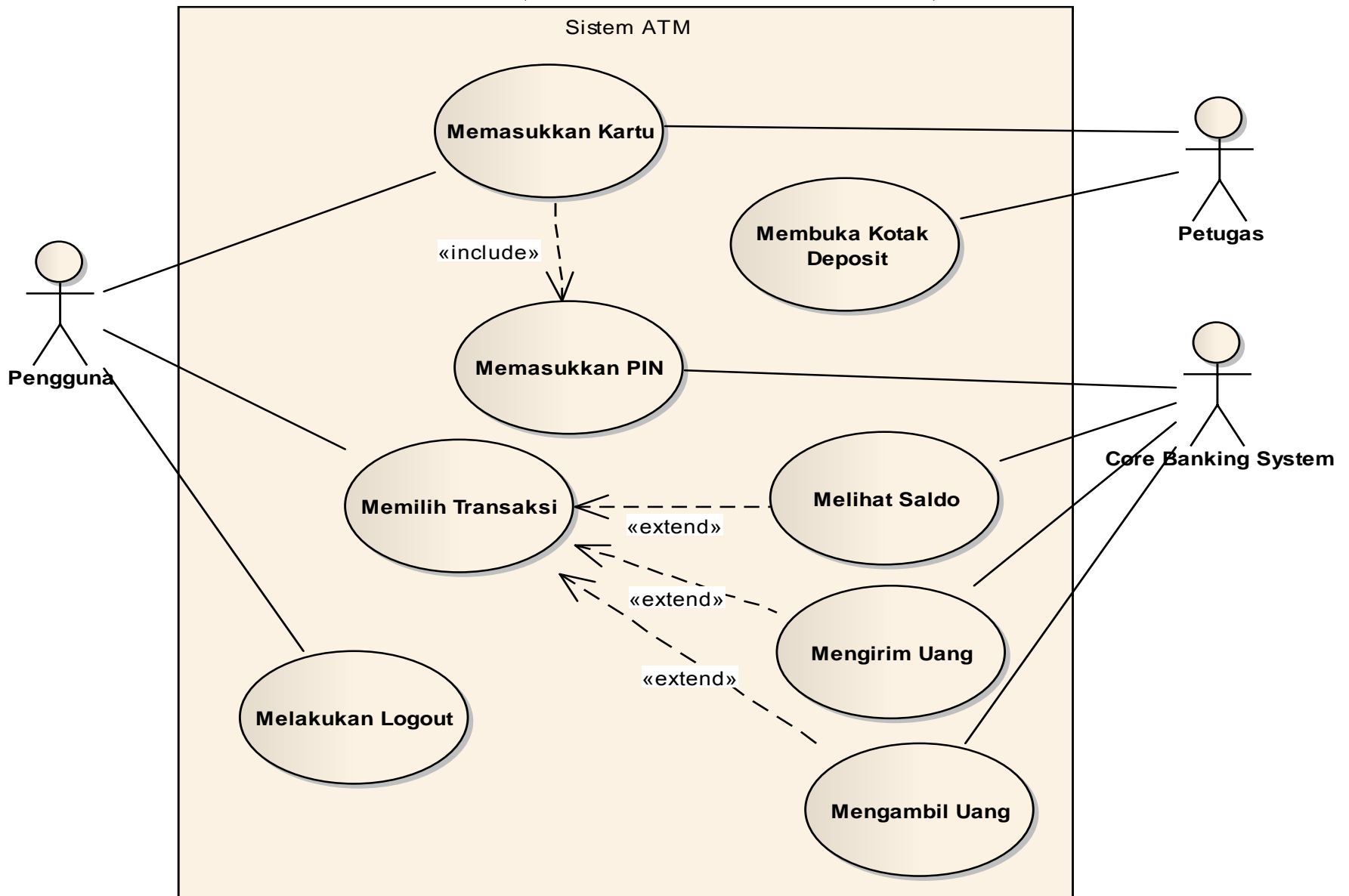
# Contoh Kasus

- ✓ Proses pengecekan tabungan akan menetapkan persetujuan untuk proses transaksi. Jika persetujuan ditolak, ATM akan mengeluarkan pesan penolakan, namun jika diterima, transaksi akan diproses dengan menggunakan nomor rekening tabungan dan ATM membaca dari kartu debit.
- ✓ Pengecekan tabungan dilakukan bersamaan pada saat ATM memvalidasi kartu debit dari bank yang bersangkutan. Jika kartu valid, password akan dicek dengan nasabah.

# Use Case ATM



# Use Case ATM (Multi Actors)





# Skenario Use Case

- Untuk memudahkan kita dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya maka kita dapat melakukan pemilahan terhadap skenario tersebut, antara lain:

Nama use case : Memasukkan Kartu (Memasukkan PIN)

Actor : User, bank

Type : Primary

Tujuan : verifikasi user

ACTOR	SISTEM
1. User Memasukkan kartu debit	
	2. ATM meminta PIN dari user
3. User memasukkan PIN & menekan OK	
	4. ATM Memverifikasi dengan Bank bahwa kartu & PIN adalah legal & rekening yg benar
	5. ATM meminta jenis transaksi

# Skenario Use Case

Nama use case : Melihat Saldo

Actor : User, bank

Type : Primary

Tujuan : Mengecek jumlah saldo tabungan

Deskripsi : User datang ke ATM dgn kartu debit untuk melakukan pengecekan saldo. User memasukkan kartu ke ATM. ATM meminta user untuk memasukkan PIN. User memasukkan PIN dan sistem memproses pengecekan saldo dan menampilkan jumlah saldo di layar.

ACTOR	SISTEM
1. User memilih transaksi Lihat Saldo	
	2. Menampilkan jumlah saldo

# Skenario Use Case

Nama use case : Mengirim Uang

Actor : User, bank

Type : Primary

Tujuan : Mengirimkan Uang ke Nasabah lain Melalui ATM

Deskripsi : User datang ke ATM dgn kartu debit untuk melakukan pengecekan saldo. User memasukkan kartu ke ATM. ATM meminta user untuk memasukkan PIN. User memasukkan PIN, memasukkan norek tujuan, nominal uang yang dikirim dan sistem akan memproses pengiriman uang dan mengeluarkan nota. ATM mengirim transaction record ke Bank untuk mengupdate saldo tabungan.

# Skenario Use Case

ACTOR	SISTEM
1. User memilih transaksi Transfer	
	2. ATM meminta norek tujuan
3. User memasukkan norek tujuan	
	4. ATM meminta jumlah nominal yg akan dikirimkan
5. User memasukkan jumlah nominal uang yang akan dikirimkan	
	6. ATM mengecek jumlah uang yang akan ditransfer dengan saldo minimal yang diperbolehkan pada bank tersebut.
	7. Update saldo
	8. ATM mengirimkan uang
	9. ATM mencetak nota dan mengeluarkan kartu

# Skenario Use Case

Nama use case : Mengambil Uang

Actor : User, bank

Type : Primary

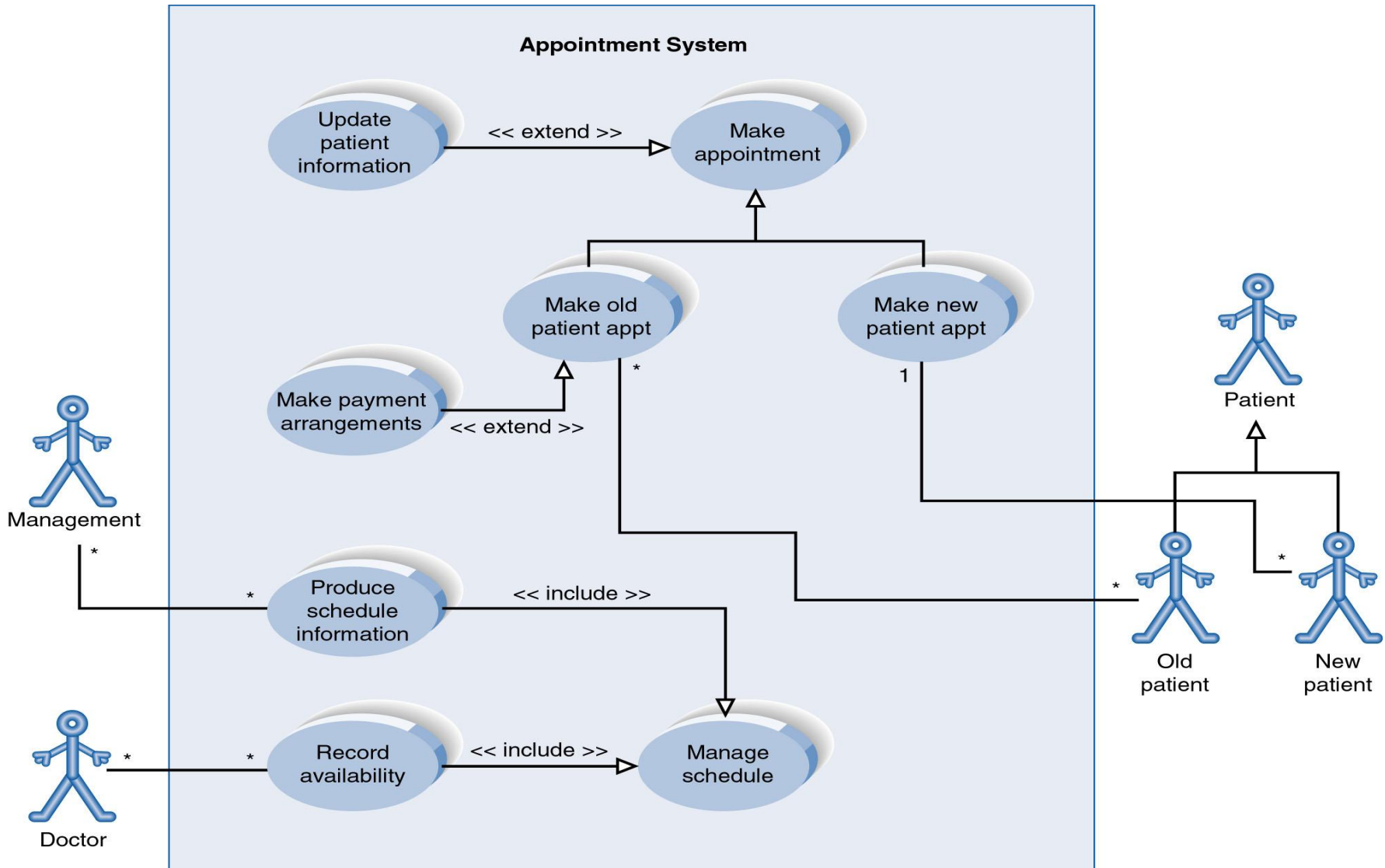
Tujuan : Penarikan uang secara cash

Deskripsi : User datang ke ATM dengan kartu debit untuk melakukan penarikan tunai. User memasukkan kartu ke ATM. ATM meminta user untuk memasukkan PIN. User memasukkan PIN dan sistem memproses penarikan tunai. ATM mengeluarkan uang dan mengeluarkan nota. ATM mengirim transaction record ke bank untuk meng-update saldo tabungan.

# Skenario Use Case

ACTOR	SISTEM
1. User memilih transaksi Penarikan	
	2. ATM meminta jumlah uang yang akan ditarik
3. User memasukkan jumlah uang yang akan ditarik	
	4. ATM mengecek jumlah uang yang akan ditarik dengan saldo minimal yang diperbolehkan pada bank tersebut.
	5. Update saldo
	6. ATM mengeluarkan uang
	7. ATM mencetak nota dan mengeluarkan kartu

# Use Case dengan Relasi & Generalisasi



# Kasus Sistem Penjualan

- **Business Rules**

1. Item adalah barang yang dijual di supermarket dan harus terdaftar di dalam sistem.
2. Kasir menjual item kepada pembeli. Terdapat 2 jenis kasir, yaitu kasir biasa dan kasir express. Kasir express hanya melayani penjualan max 5 item.
3. Sistem menangani penjualan item, pemasokan barang, penukaran item.
4. Pada penukaran item, item yang ditukarkan diusahakan merupakan item yang sama, namun jika supplier tidak menyediakan lagi maka dapat ditukarkan dengan item yang lain seharga item yang kadaluarsa atau sesuai dengan perjanjian.



# TUGAS PRAKTIKUM

1. BUAT RANCANGAN USE CASE DIAGRAM UNTUK KASUS MASING-MASING MAHASISWA
2. KUMPULKAN DALAM BENTUK FILE PDF ATAU DOCX