PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
REKAYASA PERANGKAT LUNAK LANJUT

Rekayasa Perangkat Lunak Lanjut

Object Oriented Design

Silabus Mata Kuliah

1. Pendahuluan
2. *Overview: Analisis Terstruktur*
3. Overview: Perancangan Terstruktur – Arsitektur, Interface, Data
4. Analisis Berorientasi Objek
5. Perancangan Berorientasi Objek
6. Pengenalan Web App. + Requirement Web App.
7. Konsep Web Engineering

Berorientasi Objek?

Attribute:

Baju, Jaket,
Tas Punggung,
Tangan, Kaki, Mata

Behavior:

Cara Jalan ke Depan
Cara Jalan Mundur
Cara Belok ke Kiri
Cara Memanjat

Berorientasi Objek?

Attribute (State):

Ban, Stir, Pedal Rem, Pedal Gas,
Warna, Tahun Produksi

Behavior:

Cara Menghidupkan Mesin
Cara Manjalankan Mobil
Cara Memundurkan Mobil

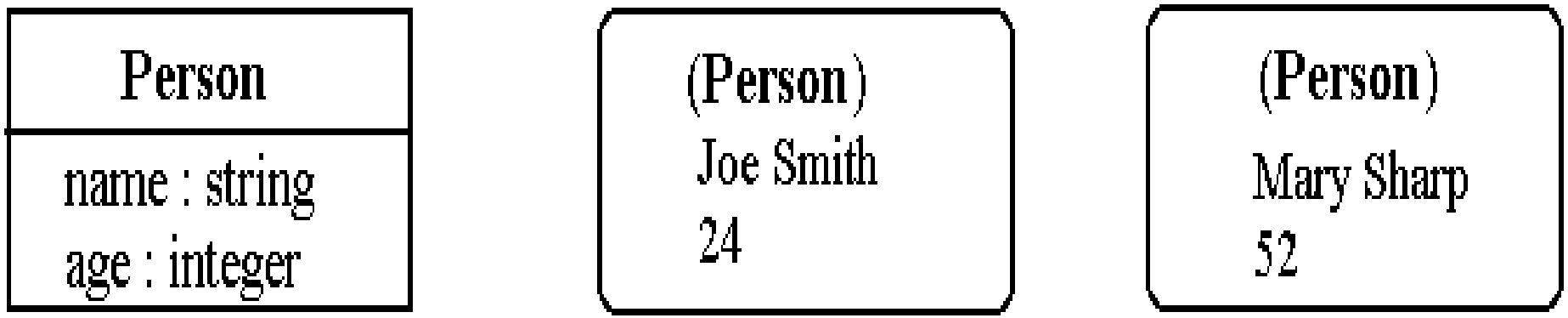
Attribute → Variable(Member)
Behavior → Method(Fungsi)

Perbedaan Class dan Object

- Class: konsep dan deskripsi dari sesuatu
  - Class mendeklarasikan method yang dapat digunakan (dipanggil) oleh object
- Object: instance dari class, bentuk (contoh) nyata dari class
  - Object memiliki sifat independen dan dapat digunakan untuk memanggil method
- Contoh Class dan Object:
  - Class: mobil
  - Object: mobilnya pak Joko, mobilku, mobil berwarna merah

Perbedaan Class dan Object

- Class seperti cetakan kue, dimana kue yg dihasilkan dari cetakan kue itu adalah object
- Warna kue bisa bermacam-macam meskipun berasal dari cetakan yang sama (object memiliki sifat independen)

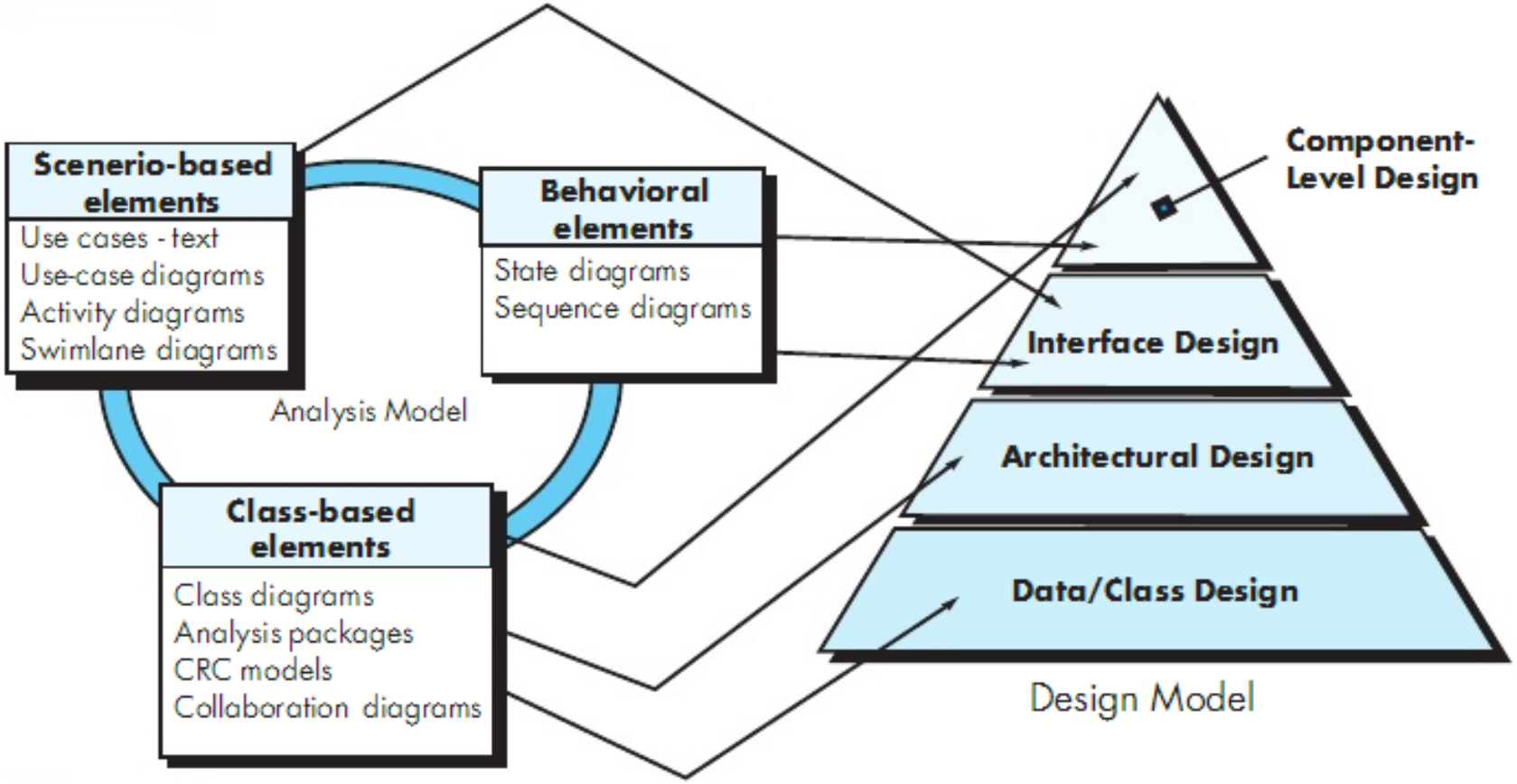


Class with Attributes

Objects with Values

Analysis to Design [1]

Transformasi model analisis OO



** SEPA 8th ed, Roger S. Pressman*

Software Design Engineering

Desain: mengumpulkan kebutuhan stakeholder, keperluan bisnis dan pertimbangan teknologi untuk memformulasikan suatu produk / system

Memodelkan aktivitas dan persiapan untuk tahap konstruksi (*coding* dan *testing*)

Goal : Memodelkan **SOLUSI** yang siap diimplementasikan (membuat program)

** SEPA 8th ed, Roger S. Pressman*

Proses Desain

- ✓ Proses iteratif untuk menerjemahkan kebutuhan menjadi "*blueprint*" untuk membangun perangkat lunak
- ✓ Karakteristik untuk mengevaluasi desain yang baik:
  - Desain harus mengimplementasikan seluruh kebutuhan baik yang eksplisit dan implisit
  - Desain harus mudah dibaca dan dipahami
  - Desain harus menyediakan gambaran lengkap suatu perangkat lunak

** SEPA 8th ed, Roger S. Pressman*

Application Development Governance

Software Development Life Cycle

Planning

- **Identifying Business Value**
  - Change Request
  - Software Request
- **Feasibility Analysis**
  - Technical Feasibility
    - Project Size Estimation
      - Use Case Points
  - Economic Feasibility
  - Organizational Feasibility

Software Proposal

Analysis

- **Business Process Identification**
  - Use Case Diagram
- **Business Process Modelling**
  - Business Process Model and Notation
- **Business Process Realization**
  - Sequence Diagram
- **Requirement Gathering**
  - Nonfunctional Requirements

Software Requirements Specification

Design

- **Program Design**
  - Class Diagram
  - Deployment Diagram
  - User Interface Design
- **Data Model**
  - Conceptual Data Diagram
  - Logical Data Model
  - Physical Data Model

Implementation

- **Software Construction**
  - Program Code
- **Software Testing**
  - Testing Plan
- **Documentation**
  - Software Documentation
  - User Documentation
- **Installation**
  - Technical Installation
  - Change Management

New Software

UML based Software Analysis and Design

(Wahono, 2009)

1. Systems **Analysis**

1.1 Identifikasi Proses Bisnis dengan **Use Case Diagram**

1.2 Pemodelan Proses Bisnis dengan **Activity Diagram** atau **BPMN**

1.3 Realisasi Proses Bisnis dengan **Sequence Diagram**

(**Boundary** - **Control** - **Entity**)

2. Systems **Design**

2.1 Pemodelan **Class Diagram**

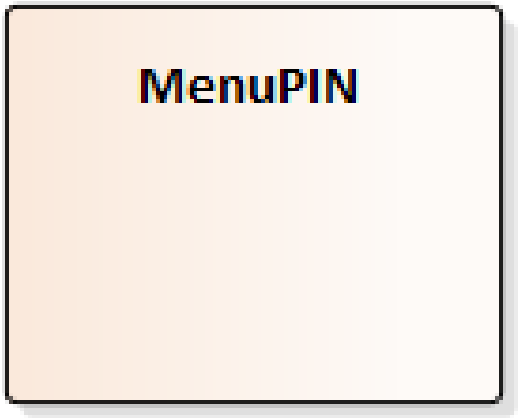
2.2 Pemodelan **User Interface Design**

2.3 Pemodelan **Data Model**

2.4 Pemodelan **Deployment Diagram**

Pemodelan Class Diagram

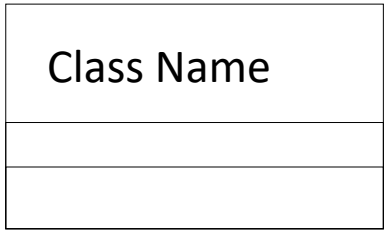
Class



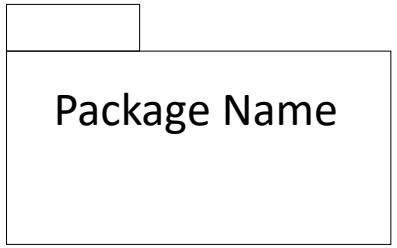

```
public class MenuPIN{

    ........

}
```

Class vs Package

- ✓ What is a class?
  - Penjelasan tentang satu set objek yang berbagi tanggung jawab yang sama, hubungan, operasi, atribut, dan semantik.



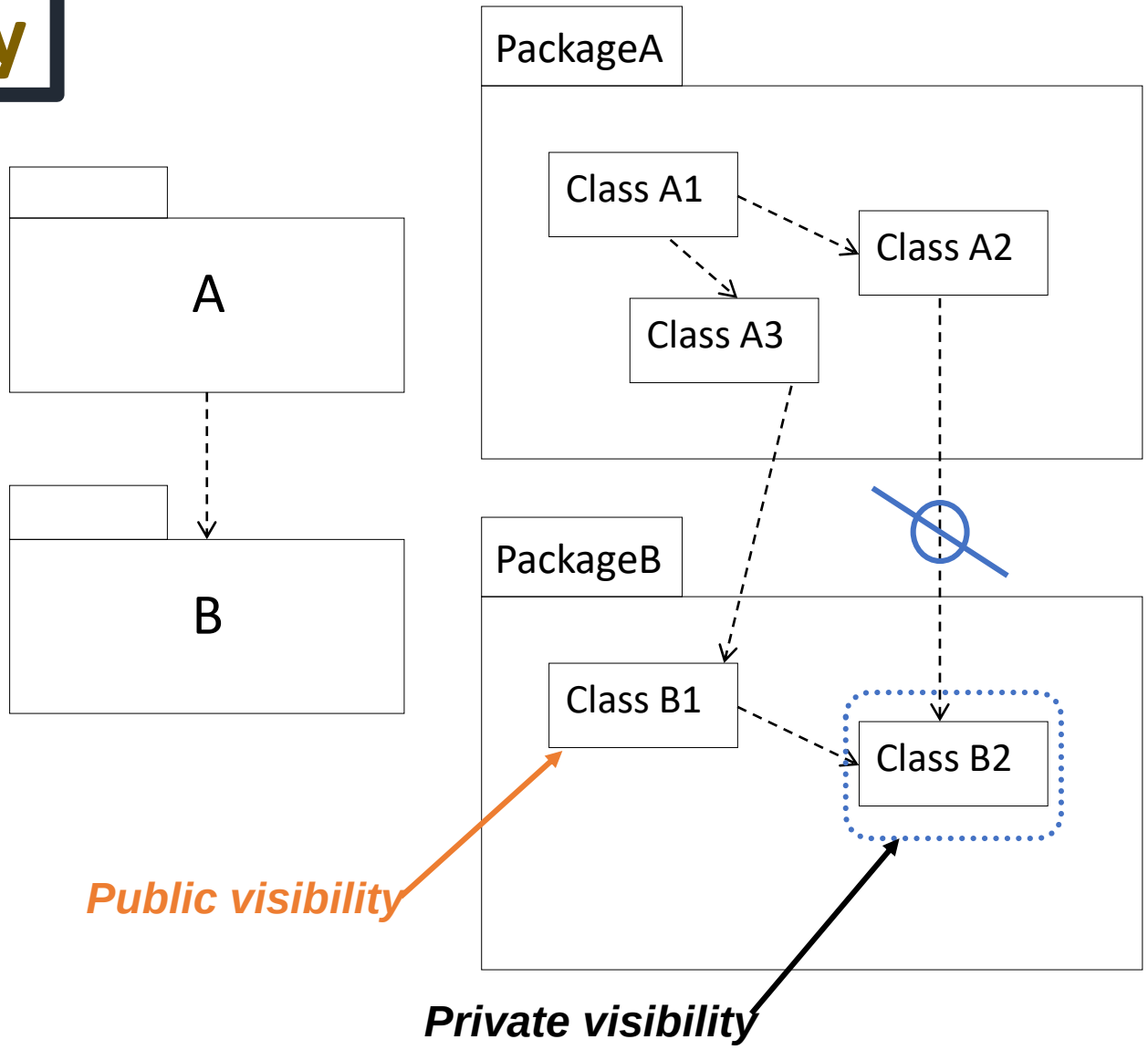
- ✓ What is a package?
  - Mekanisme umum untuk mengatur elemen dalam kelompok-kelompok
  - Sebuah model elemen yang dapat berisi elemen model lainnya



** SEPA 8th ed, Roger S. Pressman*

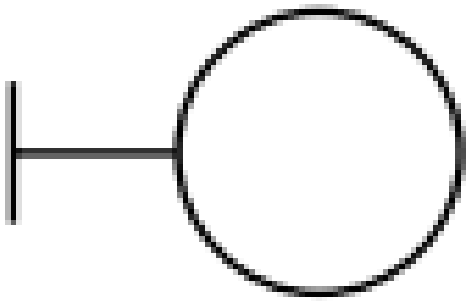
Package Visibility

Hanya public classes yang dapat diakses oleh class lain di luar packagenya sendiri



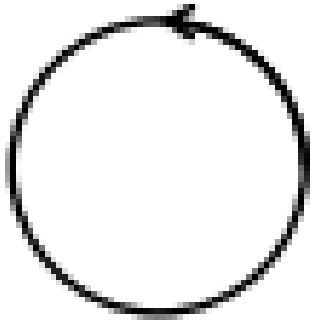
Boundary Class

- ✓ Digunakan untuk memodelkan interaksi antara sistem dan aktor
- ✓ Sering mewakili windows, forms, sensors, terminals
- ✓ Terkait dengan setidaknya satu aktor dan sebaliknya



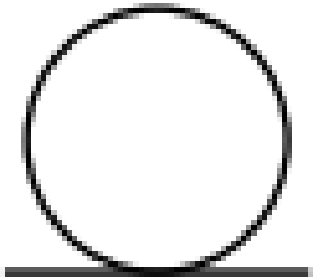
Control Class

- ✓ Merepresentasikan koordinasi, urutan, transaksi dan pengendalian benda-benda lain
- ✓ Sering merangkum use case yang terkait control
- ✓ Satu use case memiliki satu control class

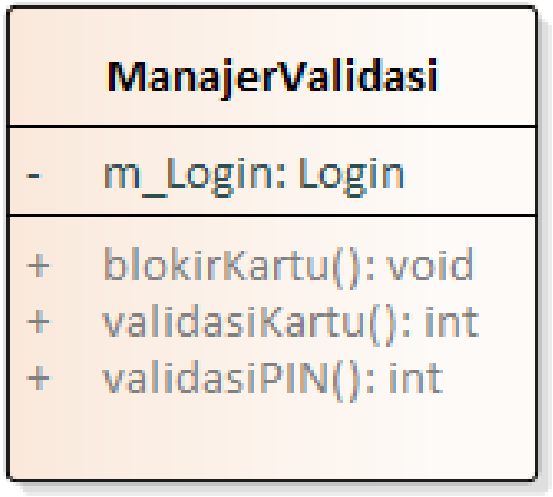


Entity Class

- ✓ Digunakan untuk memodelkan informasi
- ✓ Biasanya berumur panjang dan/ atau persisten
- ✓ Biasanya berasal langsung dari badan usaha
- ✓ Dapat dibagi oleh batas dan kontrol beberapa kelas

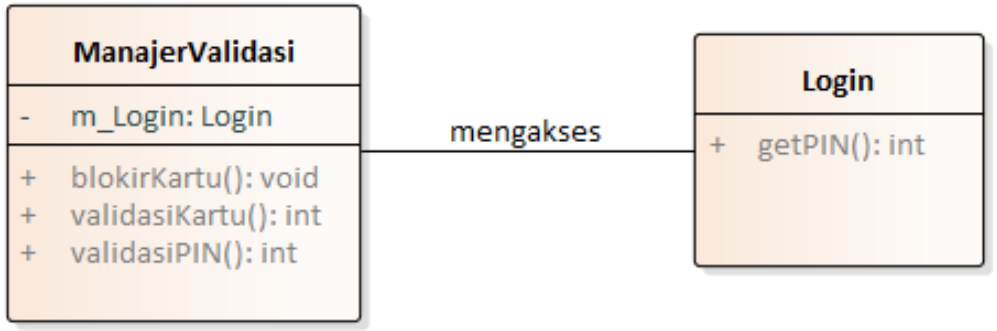


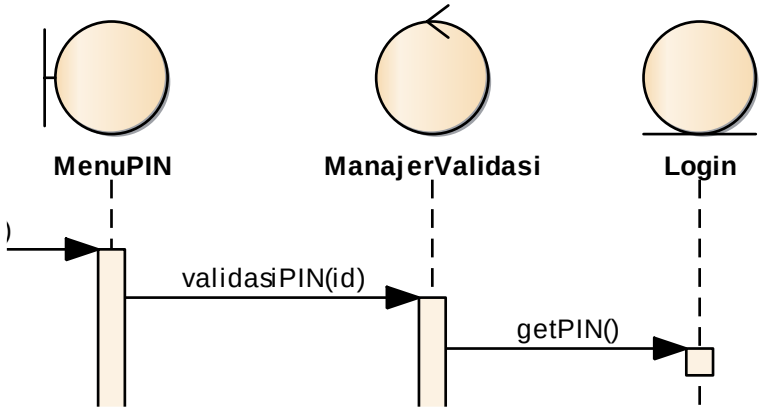
Class with Attribute and Method




```
public class ManajerValidasi {

        private Login m_Login;

        public int validasiKartu(){
            return 0;
        }

        public int validasiPIN(){
            return 0;
        }

        public void blokirKartu(){

        }
    }
```

Class Association

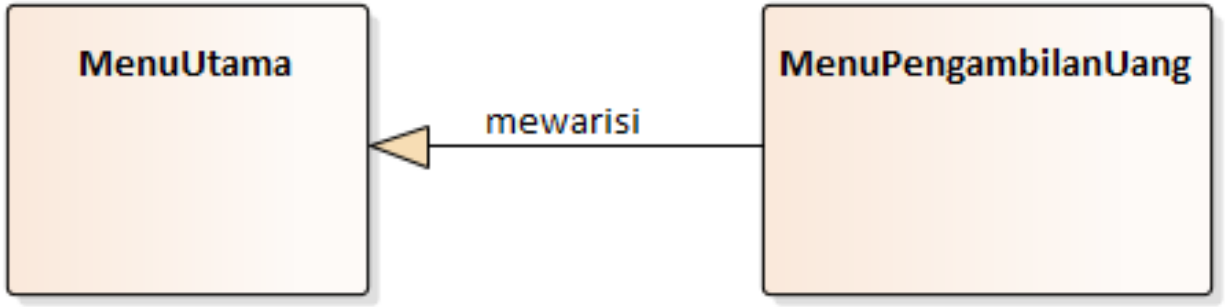





```
public class ManajerValidasi {

        private Login m_Login;

        public int validasiKartu(){
            return 0;
        }

        public int validasiPIN(){
            int pin = m_login.getPIN();
            return 0;
        }

        public void blokirKartu(){

        }
}
```

```
public class Login {


        public int getPIN(){
            return 0;
        }

}
```

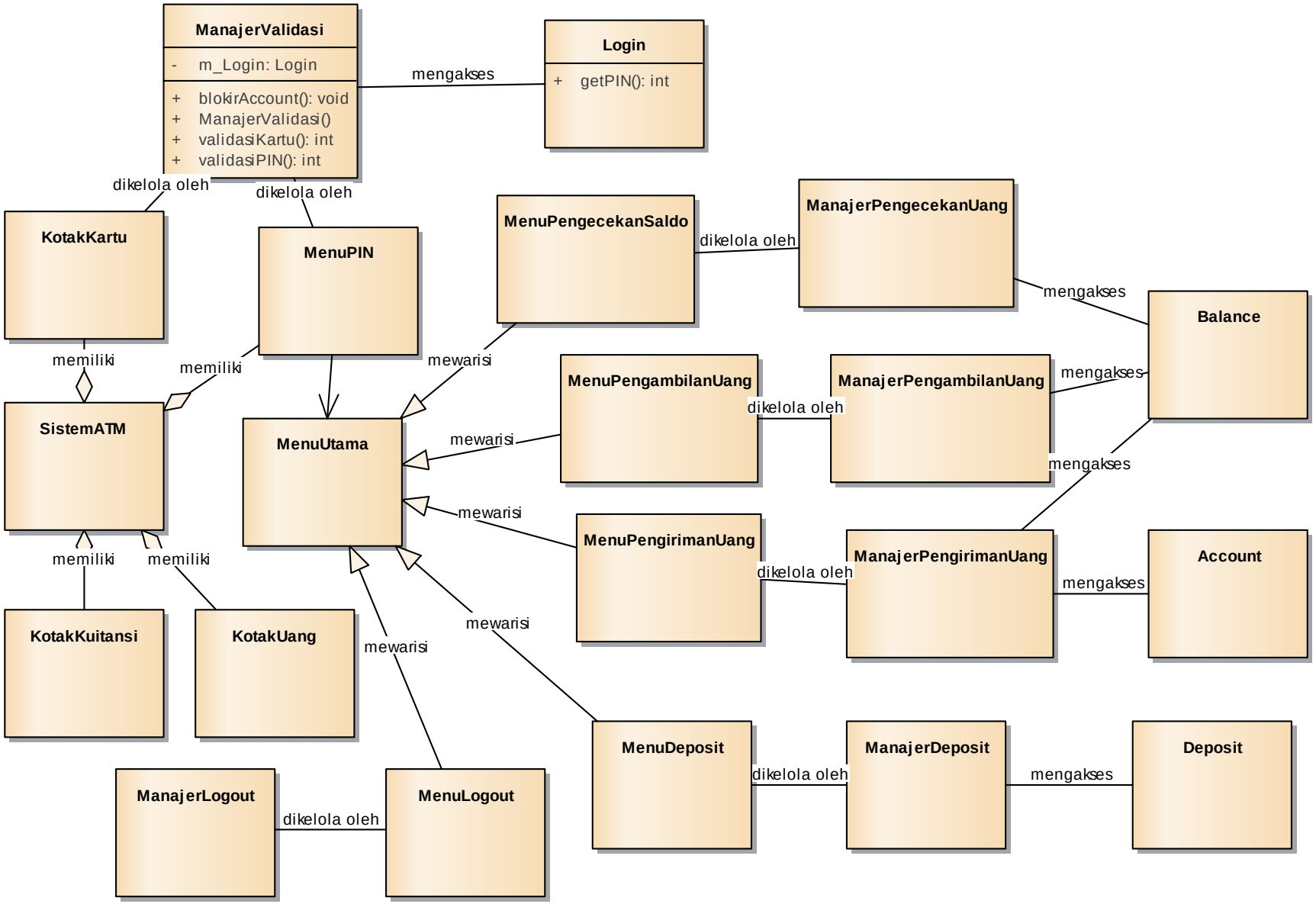
Class Inheritance




```
public class MenuUtama {

    ......

}
```

```
public class MenuPengambilanUang
        extends MenuUtama {

    .....

}
```

Class Diagram: Sistem ATM



UML based Software Analysis and Design

(Wahono, 2009)

1. Systems **Analysis**

1.1 Identifikasi Proses Bisnis dengan **Use Case Diagram**

1.2 Pemodelan Proses Bisnis dengan **Activity Diagram** atau **BPMN**

1.3 Realisasi Proses Bisnis dengan **Sequence Diagram**

(**Boundary** - **Control** - **Entity**)

2. Systems **Design**

2.1 Pemodelan **Class Diagram**

2.2 Pemodelan **User Interface Design**

2.3 Pemodelan **Data Model**

2.4 Pemodelan **Deployment Diagram**

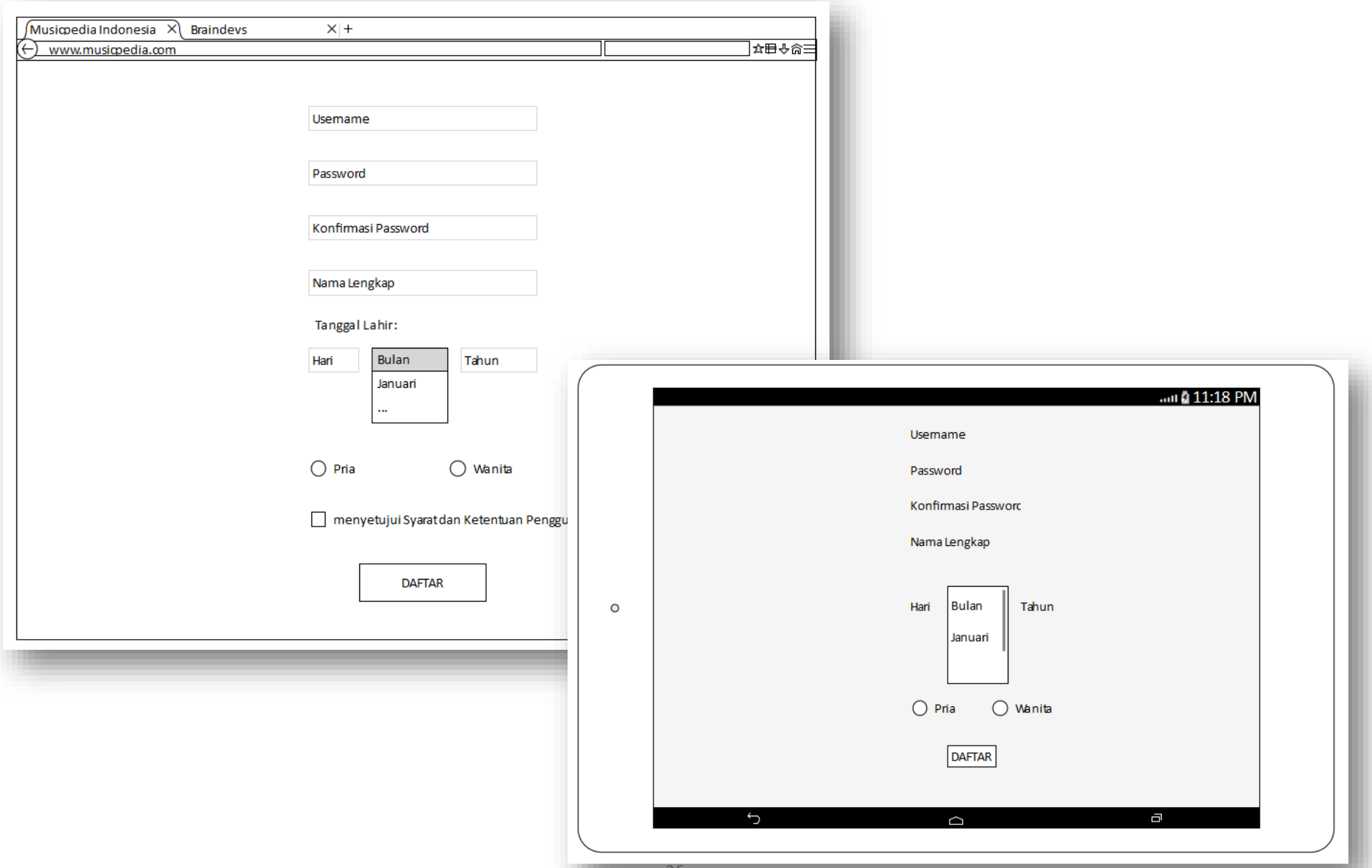
Pemodelan User Interface Design

Philosophy and Principles (Interface Design)

- Interface design is an art
- Balance between
  - Making the interface useful and
  - Presenting too much information

- Principles for User Interface Design:
  1. Layout: Consistent use of screen area
  2. Content awareness: Users know where they are
  3. Aesthetics: White space vs. functionality
  4. User experience: Ease of use vs. learning curve
  5. Consistency: User can predict what will happen for each action
  6. Minimal user effort: Simple to use, three click rule

User Interface Design Melakukan Registrasi (versi Web dan versi Android)



UML based Software Analysis and Design

(Wahono, 2009)

1. Systems **Analysis**

1.1 Identifikasi Proses Bisnis dengan **Use Case Diagram**

1.2 Pemodelan Proses Bisnis dengan **Activity Diagram** atau **BPMN**

1.3 Realisasi Proses Bisnis dengan **Sequence Diagram**

(**Boundary** - **Control** - **Entity**)

2. Systems **Design**

2.1 Pemodelan **Class Diagram**

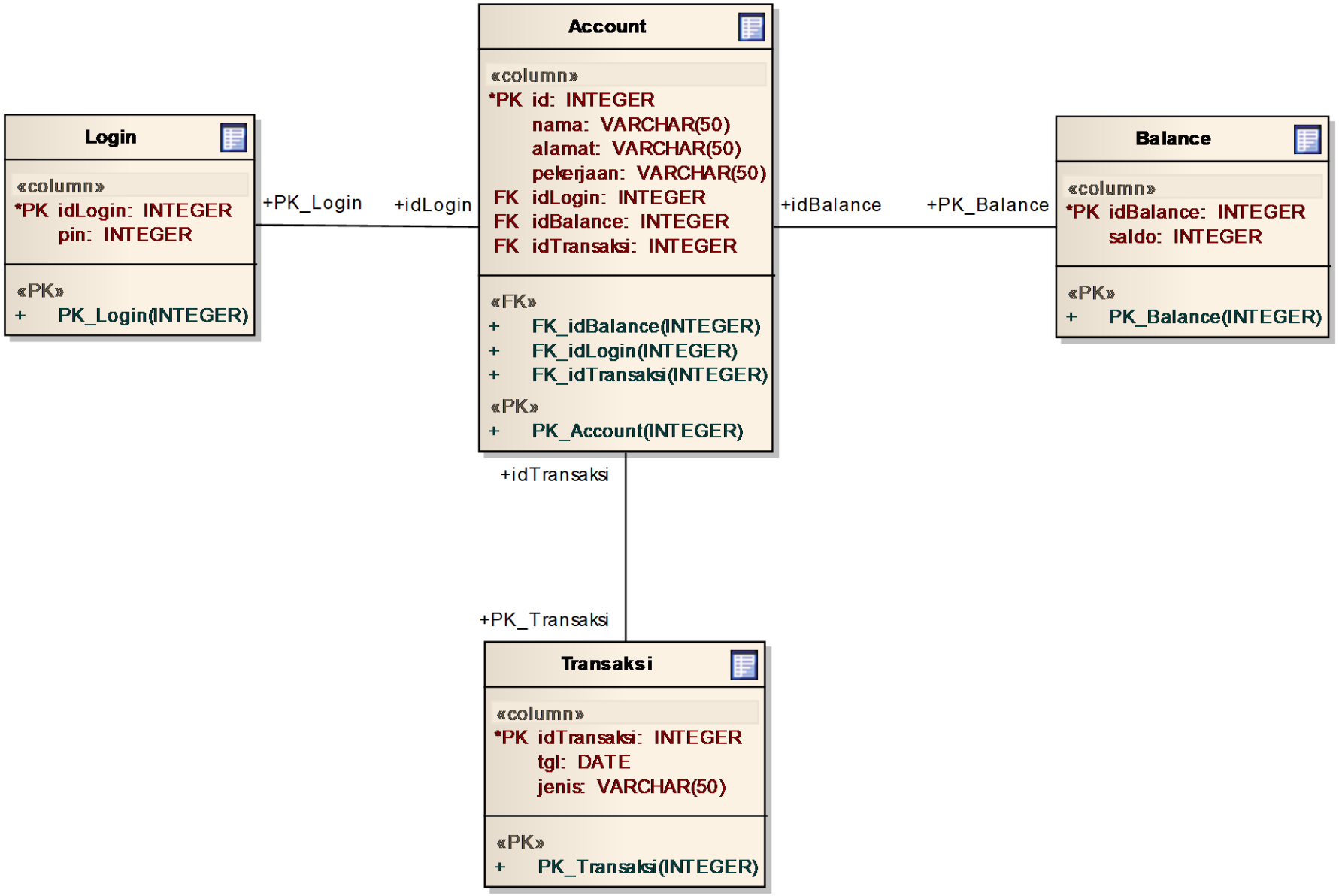
2.2 Pemodelan **User Interface Design**

2.3 Pemodelan **Data Model**

2.4 Pemodelan **Deployment Diagram**

Pemodelan Data Model

Data Model Sistem ATM



UML based Software Analysis and Design

(Wahono, 2009)

1. Systems **Analysis**

1.1 Identifikasi Proses Bisnis dengan **Use Case Diagram**

1.2 Pemodelan Proses Bisnis dengan **Activity Diagram** atau **BPMN**

1.3 Realisasi Proses Bisnis dengan **Sequence Diagram**

(**Boundary** - **Control** - **Entity**)

2. Systems **Design**

2.1 Pemodelan **Class Diagram**

2.2 Pemodelan **User Interface Design**

2.3 Pemodelan **Data Model**

2.4 Pemodelan **Deployment Diagram**

Pemodelan Deployment Diagram

Deployment Diagram

- Servers
  - Mainframes, Minis, Micros
- Clients
  - Input/Output HW used by users
  - Terminals, PCs, special purpose HW
- Network
  - HW and SW to connect clients to servers
- Nodes
  - Any piece of hardware in the model
  - A computational resource
  - Labeled by its name
  - Stereotype to label the type of node
- Artifacts
  - Piece of the information system, such as software or a database table
- Node with Deployed Artifact
  - Shows artifact placed on a physical node
  - Good for showing distribution data or software
- Communication paths
  - Links between nodes of the network

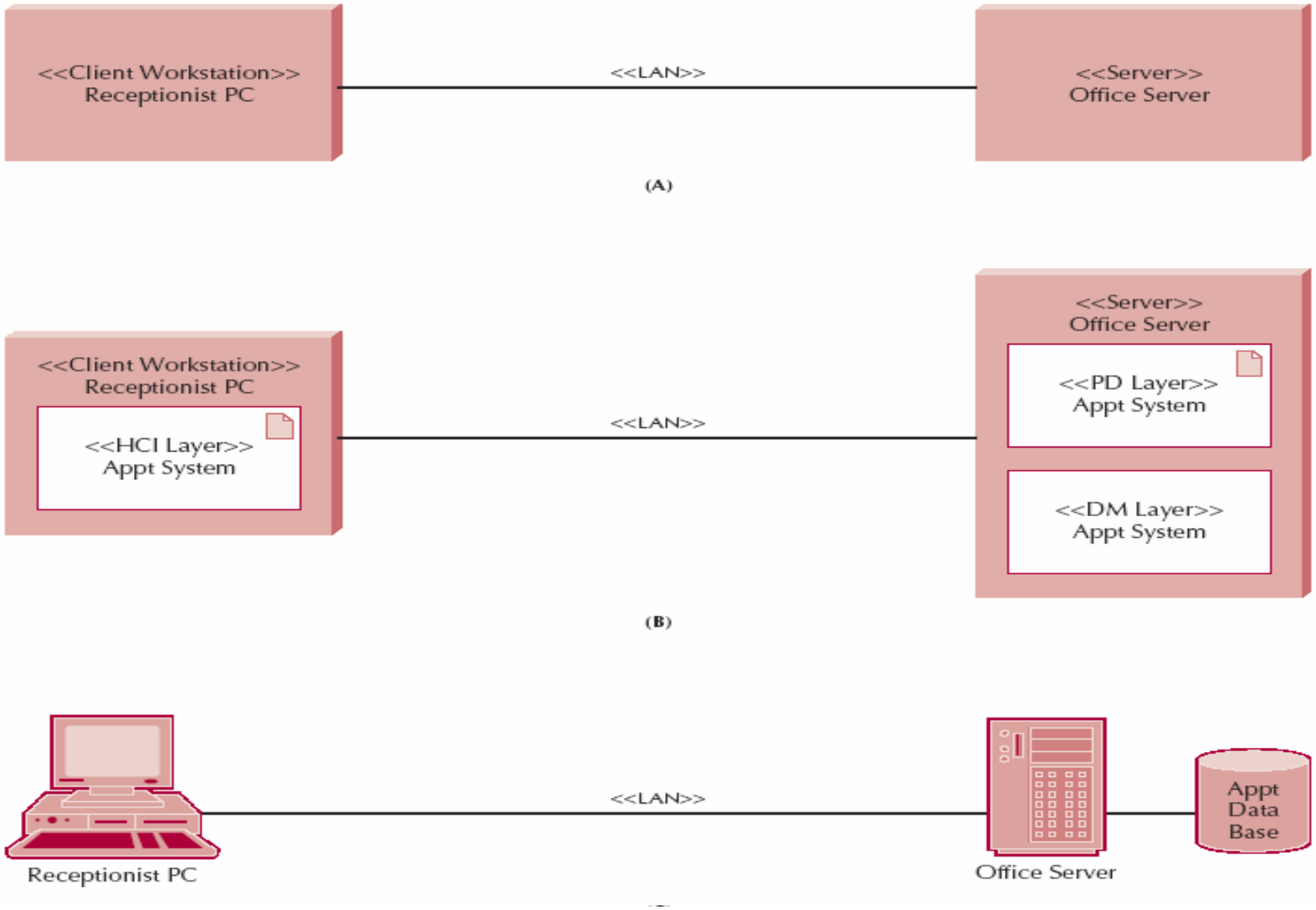
Node

Artifact

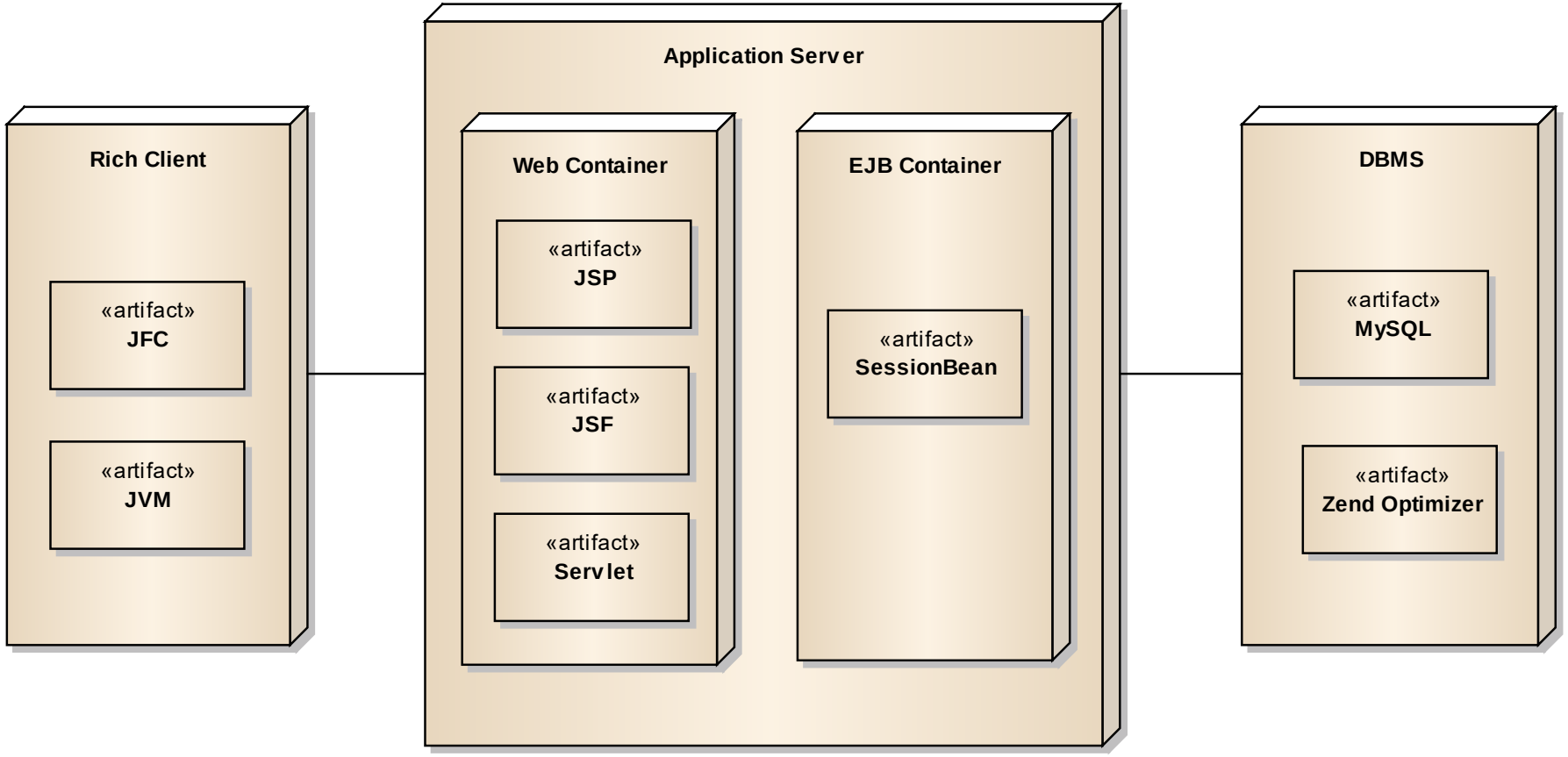
Node with Deployment Artifact

Communication Path

Diagram Examples



Deployment Diagram (3 Tier)



References

1. Slide Materi Romi Satrio Wahono - Ilmu Komputer.com dan Brainmatics
2. Alan Dennis et al, Systems Analysis and Design with UML 5th Edition, *John Wiley and Sons*, 2016
3. Joseph S. Valacich and Joey F. George, Modern Systems Analysis and Design 8th Edition, *Pearson Education*, 2017
4. Scott Tilley and Harry J. Rosenblatt, Systems Analysis and Design 11th Edition, *Cengage Learning*, 2017
5. Kenneth E. Kendall and Julie E Kendall, Systems Analysis and Design 8th Edition, *Prentice Hall*, 2010
6. John W. Satzinger, Robert B. Jackson, Stephen D. Burd, Systems Analysis and Design in a Changing World 6th Edition, *Course Technology*, 2012
7. Hassan Gomaa, Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, *Cambridge University Press*, 2011
8. Howard Podeswa, UML for the IT Business Analyst 2nd Edition, *Course Technology*, 2009
9. Jeffrey A. Hoffer et al, Modern Systems Analysis and Design 6th Edition, *Prentice Hall*, 2010
10. Albert Endres and Dieter Rombach, A Handbook of Software and Systems Engineering, *Pearson Education*, 2003

THANKS

ANY QUESTIONS?