

# MODUL 4

## Unified Software Development Process (USDP)

---

### Daftar Isi

4.1 Pengantar USDP.....	2
4.2 Fase USDP.....	2
4.2.1 Fase, Workflow dan Iterasi .....	3
4.2.2 Perbedaan USDP dan Siklus Hidup Waterfall .....	3
4.2.3 Prinsip Pokok USDP .....	4
4.3 Aktivitas Utama .....	5
4.3.1 Requirements Capture and Modelling .....	5
4.3.2 Requirement Analysis.....	7
4.3.3 System Design.....	7
4.3.4 Class Design .....	7
4.3.5 User Interface Design .....	7
4.3.6 Data Management Design .....	7
4.3.7 Construction.....	7
4.3.8 Testing .....	8
4.3.9 Implementation .....	8
Referensi.....	9

## 4.1 Pengantar USDP

Suatu proses pengembangan harus mampu melakukan spesifikasi terhadap apa yang dilakukan, pada saat kapan hal tersebut dikerjakan, bagaimana cara mengerjakannya dan siapa yang mengerjakan sehingga tujuannya dapat tercapai. Teknik manajemen proyek (*Project Management technique*) digunakan untuk mengatur, dan pengontrol proyek yang dikerjakan. Pada pengembangan perangkat lunak hal tersebut dilakukan dengan menggunakan salah satu tools Rational Unified Proses yang dikeluarkan oleh IBM berdasarkan pada *Unified Software Development Process* (USDP) (Jacobson, et al., 1999). USDP dikembangkan oleh team yang membangun UML. USDP terdiri atas :

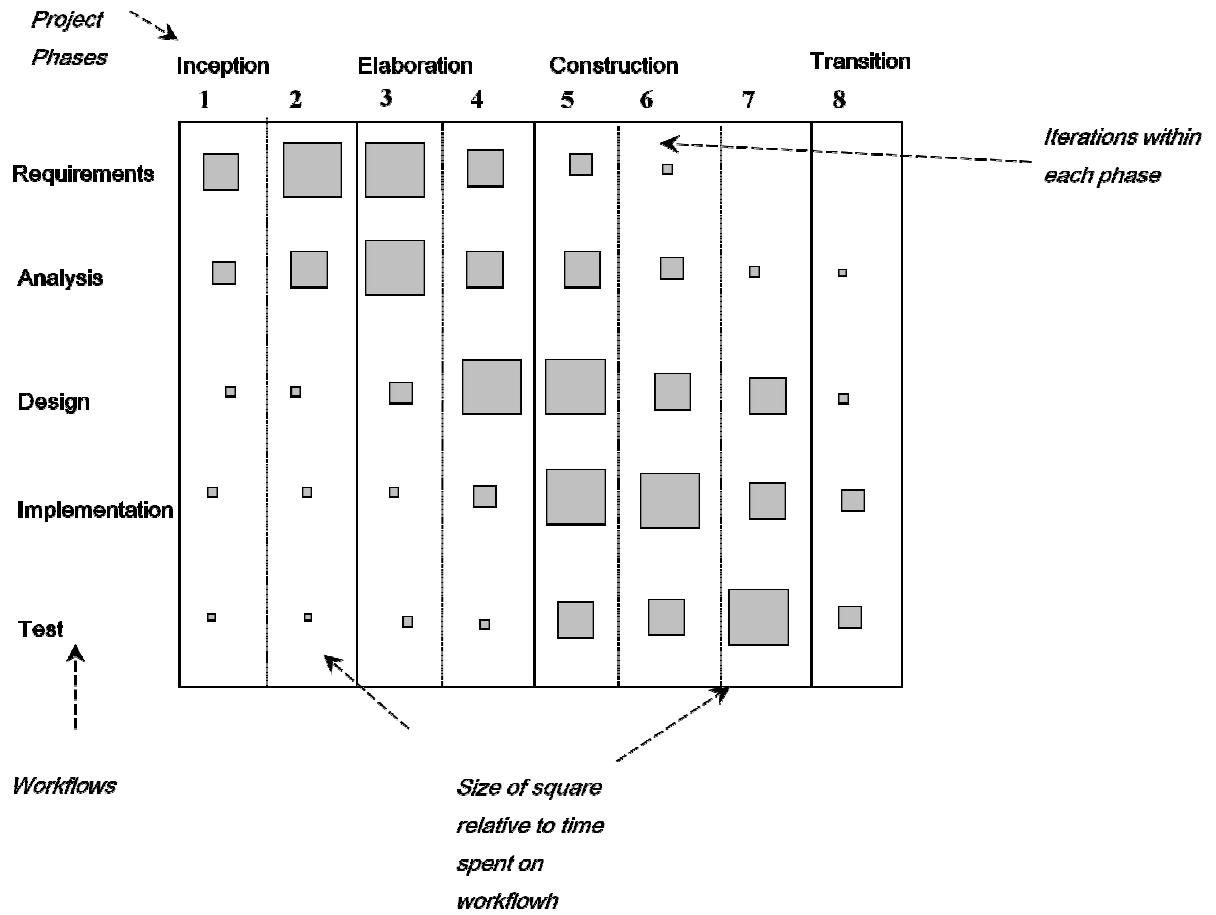
- Pengembangan secara iteratif dan incremental (*Iterative and Incremental Development*)
- Pengembangan berdasarkan komponen (*Component Based Development*)
- Pengembangan berdasarkan kebutuhan (*Requirement Driven Development*)
- Kemampuan untuk melakukan konfigurasi (*Configurability*)
- Arsitektur terpusat (*Architecture centrism*)
- Teknik pemodelan visual (*Visual Modelling technique*)

USDP tidak mengikuti siklus hidup pengembangan perangkat lunak tradisional (*waterfall model*) melainkan mengadopsi pendekatan *iterative* dengan 4 fase utama. Setiap fase menggambarkan penekanan pada kegiatan yang penting pada pengembangan sistem. Fase-fase tersebut digambarkan dalam bentuk aliran kerja (*workflows*) dimana aliran tersebut merupakan serangkaian aktifitas.

## 4.2 Fase USDP

Fase-fase pada USDP meliputi :

1. Inception
2. Elaboration
3. Construction
4. Transition



Gambar 4. 1 Fase dan Workflow USDP

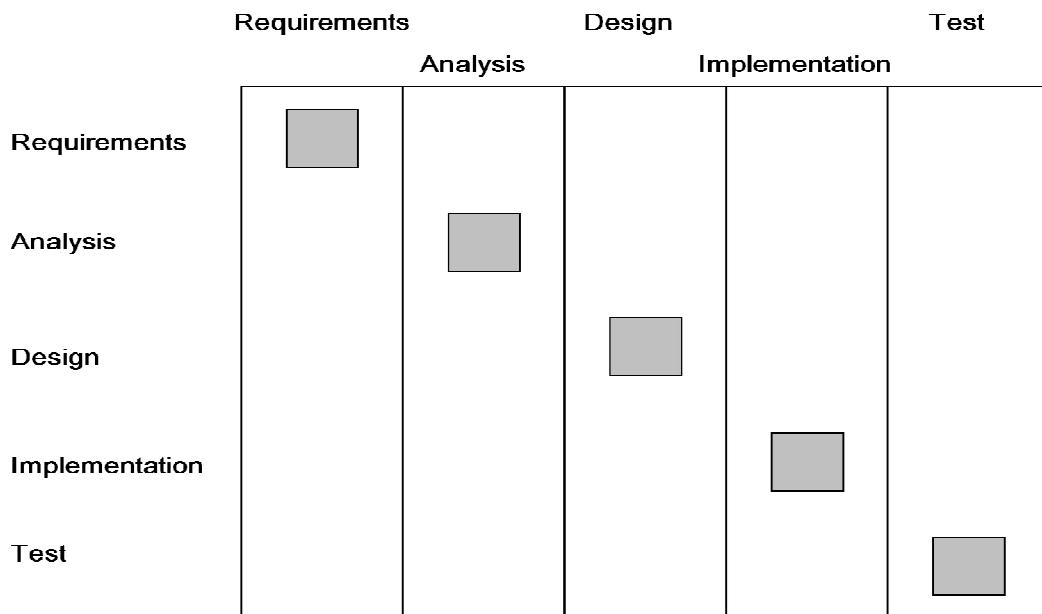
#### 4.2.1 Fase, Workflow dan Iterasi

Satu fase pengembangan terdiri atas beberapa aliran kerja (*workflow*). Usaha yang dilakukan seperti lama waktu penggerjaan untuk setiap aliran kerja bervariasi dari fase ke fase. Dalam satu fase bisa lebih dari satu iterasi.

#### 4.2.2 Perbedaan USDP dan Siklus Hidup Waterfall

Beberapa perbedaan antara USDP dan model Waterfall antara lain:

- Pada *waterfall life cycle* fase kegiatan dan aliran kerja terkait bersamaan sedangkan pada USDP antara fase dan aliran kerja terpisah



**Gambar 4. 2 Fase pada siklus hidup Waterfall**

- Pada fase *requirement* hanya melakukan aktivitas *requirement* saja sedangkan pada USDP ada serangkaian aktivitas yang dilakukan. (Lihat Tabel 4.1)
- Semua aktivitas *requirement* pada *waterfall* harus lengkap sebelum masuk ke fase analisis. Sedangkan USDP tidak harus. *Requirements* bisa dilakukan pada setiap fase. Hanya bobot aktivitasnya yang berbeda.
- Pada siklus hidup *iterative*, beberapa *requirement* dapat muncul pada saat analisa.

#### 4.2.3 Prinsip Pokok USDP

Pada prinsipnya USDP bukanlah suatu metodologi yang baru. USDP merupakan bentuk lain dari proses pengembangan sistem. Beberapa Karakteristik yang muncul dari USDP merupakan karakteristik umum dari metodologi yang sudah ada antara lain :

- *Iterative*
- *Incremental*
- *Requirements Driven*
- *Component-based*
- *Architectural*

## 4.3 Aktivitas Utama

Proses pengembangan sistem USDP melibatkan aktivitas utama, yaitu

- *Requirements capture and modeling*
- *Requirements Analysis*
- *System Design*
- *Class Design*
- *Data management design*
- *Construction*
- *Testing*
- *implementation*

masing-masing aktivitas saling berhubungan dan tergantung satu sama lain. Pada waterfall setiap kegiatan dijalankan secara berurutan (sequence). Berbeda dengan proses pengembangan iterative yang beberapa aktivitas bisa saling mendahului.

### 4.3.1 Requirements Capture and Modelling

*Requirement Capture and Modelling* merupakan suatu proses untuk melakukan identifikasi kebutuhan perangkat lunak. Requirement di dokumentasikan dengan use cases. Model requirement dan fungsionalitasnya dapat di lihat sebagai berikut.

Tabel 4. 1 Tabel Aktivitas Proses Pengembangan Sistem

Activity	Techniques	Key Deliverables
Requirements Capture and Modelling	<ul style="list-style-type: none"><li>• Requirements Elicitation</li><li>• Use Case Modelling</li><li>• Architectural Modelling</li><li>• Prototyping</li></ul>	<ul style="list-style-type: none"><li>• Use Case Model</li><li>• Requirements List</li><li>• Initial Architecture</li><li>• Prototypes</li><li>• Glossary</li></ul>
Requirements Analysis	<ul style="list-style-type: none"><li>• Communication Diagrams</li><li>• Class and Object Modelling</li><li>• Analysis Modelling</li></ul>	<ul style="list-style-type: none"><li>• Analysis Models</li></ul>
System Design	<ul style="list-style-type: none"><li>• Deployment Modelling</li></ul>	<ul style="list-style-type: none"><li>• Overview Design and</li></ul>

	<ul style="list-style-type: none"> <li>• Component Modelling</li> <li>• Package Modelling</li> <li>• Architectural Modelling</li> <li>• Design Patterns</li> </ul>	Implementation Architecture
Class Design	<ul style="list-style-type: none"> <li>• Class and Object Modelling</li> <li>• Interaction Modelling</li> <li>• State Modelling</li> <li>• Design Patterns</li> </ul>	<ul style="list-style-type: none"> <li>• Design Models</li> </ul>
User Interface Design	<ul style="list-style-type: none"> <li>• Class and Object Modelling</li> <li>• Interaction Modelling</li> <li>• State Modelling</li> <li>• Package Modelling</li> <li>• Prototyping</li> <li>• Design Patterns</li> </ul>	<ul style="list-style-type: none"> <li>• Design Models with Interface Specification</li> </ul>
Data Management Design	<ul style="list-style-type: none"> <li>• Class and Object Modelling</li> <li>• Interaction Modelling</li> <li>• State Modelling</li> <li>• Package Modelling</li> <li>• Design Patterns</li> </ul>	<ul style="list-style-type: none"> <li>• Design Models with Database Specification</li> </ul>
Construction	<ul style="list-style-type: none"> <li>• Programming</li> <li>• Component Re-use</li> <li>• Database DDL</li> <li>• Programming Idioms</li> <li>• Manual Writing</li> </ul>	<ul style="list-style-type: none"> <li>• Constructed System</li> <li>• Documentation</li> </ul>
Testing	<ul style="list-style-type: none"> <li>• Programming</li> <li>• Test Planning and Design</li> <li>• Testing</li> </ul>	<ul style="list-style-type: none"> <li>• Test Plans</li> <li>• Test Cases</li> <li>• Tested System</li> </ul>
Implementation	<ul style="list-style-type: none"> <li>• Planning</li> <li>• Training</li> <li>• Data Conversion</li> </ul>	<ul style="list-style-type: none"> <li>• Installed System</li> </ul>

### **4.3.2 Requirement Analysis**

Pada dasarnya, setiap use case menggambarkan satu buah user requirement utama. Setiap use case di analisa secara terpisah untuk mengidentifikasi objek yang diperlukan. Use case juga dianalisa untuk menentukan bagaimana objek tersebut berinteraksi dan bagaimana tanggung jawab setiap objek pada use case tersebut. Diagram komunikasi (Communication Diagram) digunakan untuk memodelkan interaksi dari setiap objek. Model pada setiap use case di integrasikan untuk mendapatkan analisa *class diagram*.

### **4.3.3 System Design**

Pada bagian ini dilakukan indentifikasi dan dokumentasi standar pengembangan (rancangan interface standar, standar untuk coding).

### **4.3.4 Class Design**

Setiap model analisa dari use case akan diuraikan secara terpisah. Diagram iterasi digunakan untuk memperlihatkan komunikasi antara objek serta state diagram digunakan untuk menunjukkan perilaku suatu objek yang kompleks. Dari kedua model ini akan di dapat rancangan class diagram. Setiap class yang dihasilkan memiliki atribut dan operai secara spesifik.

### **4.3.5 User Inteface Design**

User interface design diperoleh dari hasil realisasi functionalitas. User interface design menampilkan bentuk interaksi dengan user seperti menentukan posisi dan warna dari tombol buton di layar.

### **4.3.6 Data Management Design**

Bagian ini fokus pada mekanisme dan implementasi dari sistem manajemen database yang digunakan. Teknik database seperti normalisasi, entity relational diagram sangat bermanfaat jika menggunakan model data relasional. Antara manajemen data dan rancangan class diagram masing-masing bediri secara terpisah.

### **4.3.7 Construction**

Bagian construction adalah bagian untuk membangun sebuah aplikasi dengan menggunakan teknologi tertentu. Setiap bagian dari sistem bisa saja menggunakan bahasa pemrograman yang berbeda. Misalkan interface menggunakan bahasa java dan databasenya menggunakan teknologi oracle.

#### **4.3.8 Testing**

Sebelum sistem diberikan ke client maka harus di tes lebih dahulu. Script testing diperoleh dari deskripsi use case yang sudah disepakati oleh client. Testing merupakan elemen yang penting dan harus dilakukan.

#### **4.3.9 Implementation**

Akhir dari tahapan implementasi adalah dilakukan installasi dari berbagai komputer client yang akan digunakan. Termasuk didalamnya manajemen transisi dari sistem lama ke sistem yang baru. Bagian ini melibatkan manajemen resiko serta pelatihan untuk staff.

## Referensi

1. Simon Bennet, Steve McRobb and Ray Farmer, *Object Oriented Systems Analysis and Design Using UML*, Edisi 3. ; McGraw Hill, 2006. (SB)