



# Modul Praktikum **RPL**



## DFD LEVEL 2 DAN UML

### A. TUJUAN

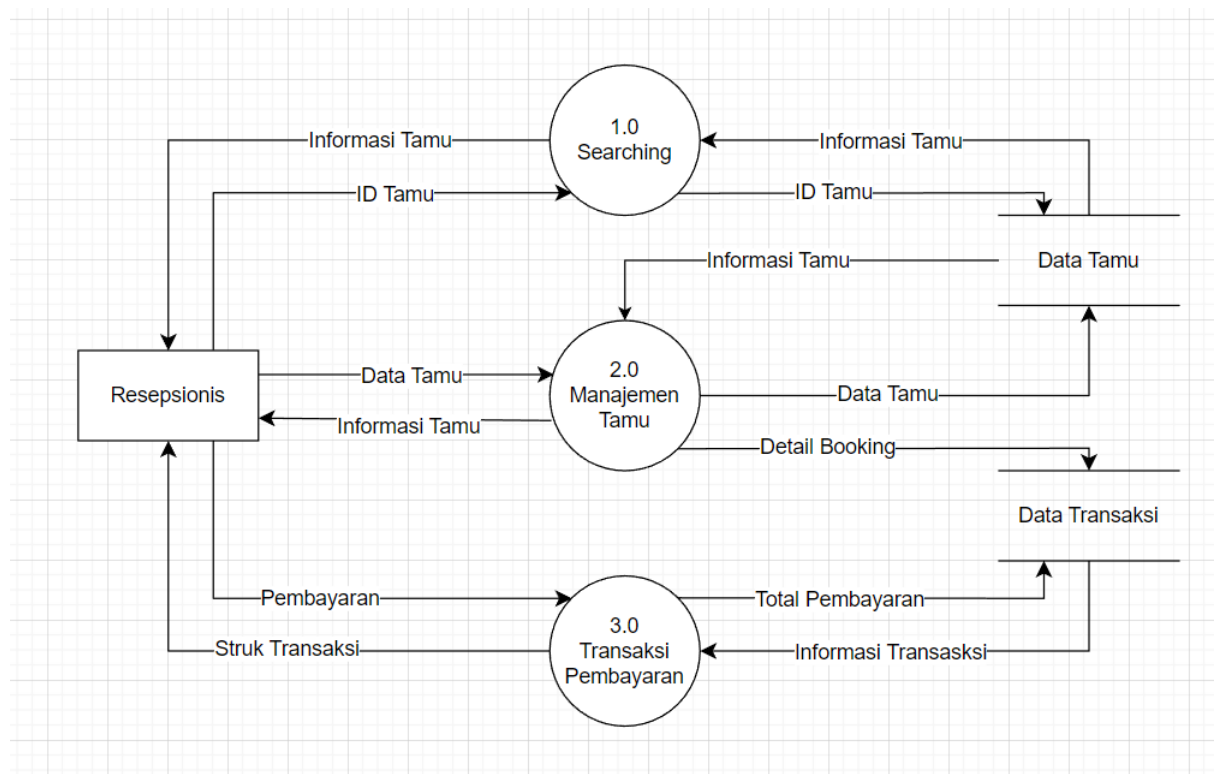
1. Mahasiswa dapat mengetahui tujuan dari levelisasi DFD
2. Mahasiswa dapat memahami tahapan levelisasi DFD
3. Mahasiswa dapat menggambar levelisasi DFD
4. Mahasiswa dapat mengetahui konsep dan bagian UML

### B. DASAR TEORI

#### 1. DFD Level 2

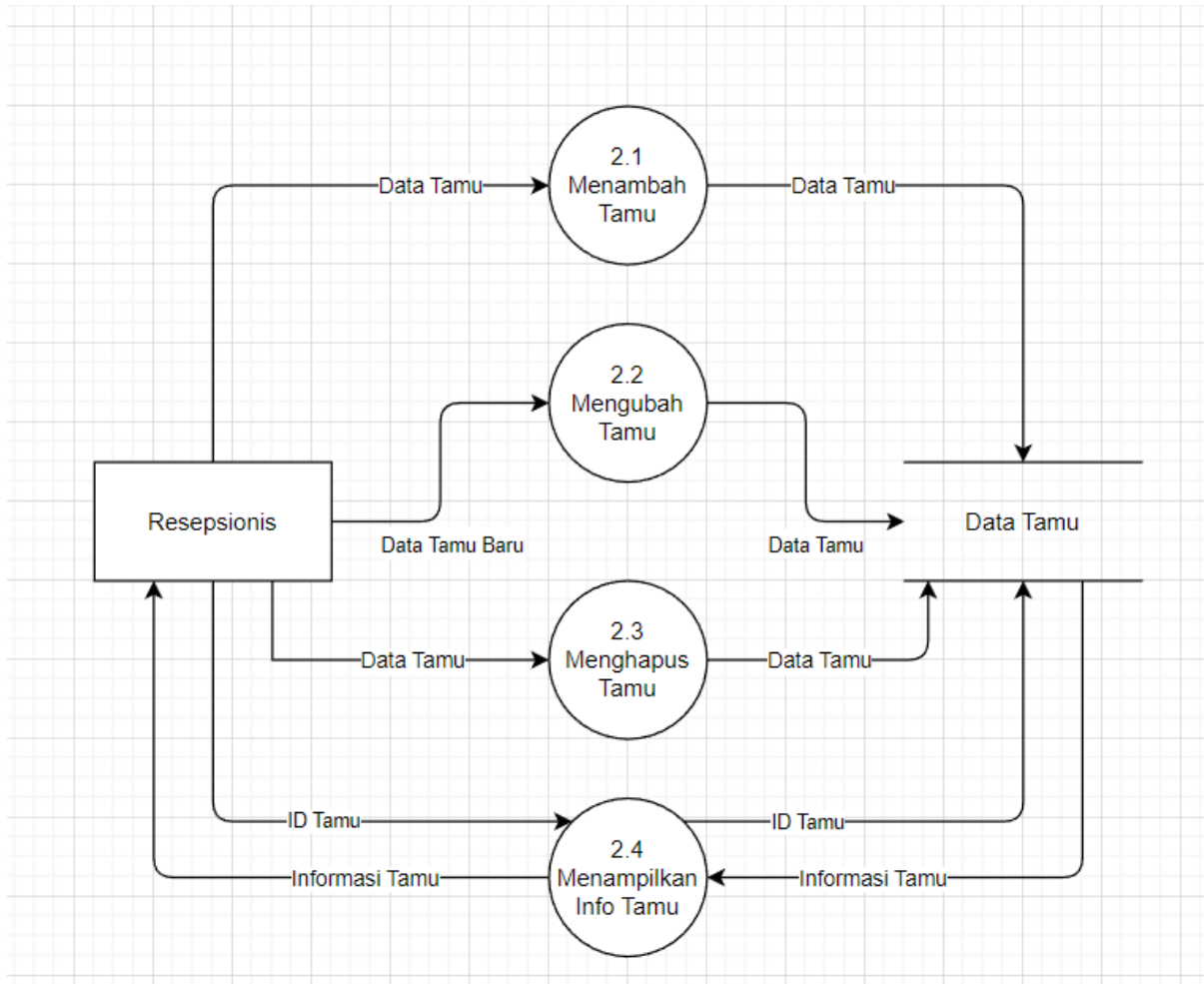
DFD level 2 merupakan diagram yang dibentuk dari dekomposisi proses yang terdapat pada DFD level 1. Tidak semua proses yang terdapat pada DFD level 1 harus di *down grade*(dekomposisi) ke dalam DFD level 2 melainkan sesuai dengan kebutuhan. Jika proses yang terdapat pada level 1 butuh perlu dijabarkan lagi, maka hal ini perlu dikerjakan kedalam DFD level 2.

Contoh:



Gambar 1 DFD level 1

Gambar diatas adalah contoh system sederhana dari resepsionis hotel, kita akan membuat DFD level 2 dari system 2.0 yaitu manajemen tamu karena proses tersebut masi perlu di dekomposisi agar rancangan system lebih jelas.



Gambar 2 DFD level 2 dari proses 2.0

Gambar diatas merupakan DFD level 2 dari proses 2.0 pada DFD level 1, pada DFD level 2 akan diperlihatkan lebih detail mengenai proses alur data dalam sebuah proses.

## 2. DFD Level 3

DFD level 3 merupakan diagram yang dibentuk dari dekomposisi proses yang terdapat pada DFD level 2. Tidak semua proses yang terdapat pada DFD level 1 harus di *down grade*(dekomposisi) ke dalam DFD level 3 melainkan sesuai dengan kebutuhan. Jika proses yang terdapat pada level 2 butuh perlu dijabarkan lagi, maka hal ini perlu dikerjakan kedalam DFD level 3.

### Catatan :

Diagram konteks adalah proses umum (ruang lingkup) sistem, sedangkan DFD Level 1 sampai selanjutnya adalah dekomposisi dari proses-proses pada diagram konteks.



### 3. UML

UML adalah *Unified Modelling Language* (UML) bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan dan untuk membangun perangkat lunak, UML merupakan metodologi dalam mengembangkan system berorientasi objek dan UML adalah alat untuk mendukung pengembangan system (Hendini, 2016). Dengan kata lain UML “Bahasa” yang menjadi standar dalam industry untuk memvisualkan, merancang dan mendokumentasikan system piranti lunak.

Tujuan dari penggunaan UML:

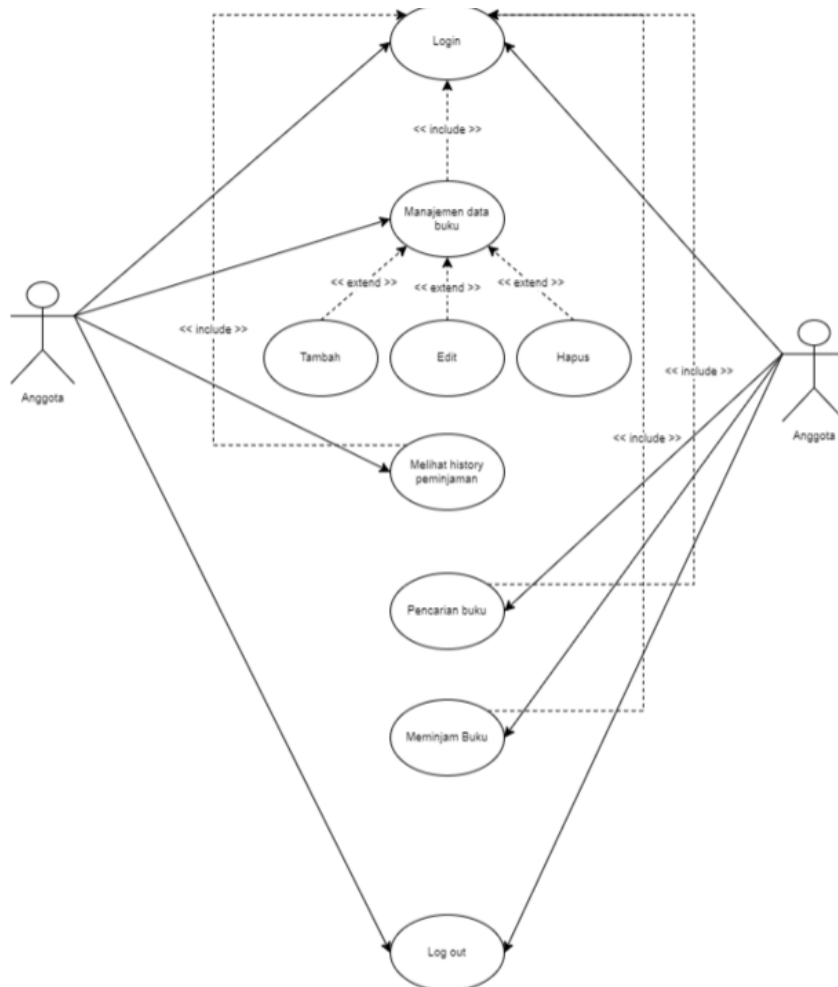
1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi object.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.
3. Berguna sebagai *blueprint*, jelas ini nantinya menjelaskan informasi yang lebih detail dalam perancangan berupa *coding* suatu program

Berikut adalah jenis-jenis UML

Jenis UML	Diagram	Fungsi/Tujuan
<i>Structure Diagram</i>	<i>Class Diagram</i>	digunakan untuk menggambarkan paket-paket atau kelas-kelas yang ada pada sistem dan akan digunakan pada suatu sistem
	<i>Object Diagram</i>	diagram yang menunjukkan gambaran dari struktur model yang ada di dalam sistem untuk jangka waktu tertentu. Diagram ini akan bergantung pada class diagram yang dibuat untuk suatu sistem
<i>Behaviour Diagram</i>	<i>Use case diagram</i>	adalah urutan interaksi yang memiliki keterkaitan antara sistem dan aktor. Use case diagram dijalankan dengan cara menggambarkan tipe interaksi yang terjadi diantara user yang terlibat di dalam sistem
	<i>Activity Diagram</i>	diagram yang menunjukkan model dari metode yang ada di dalam sistem.



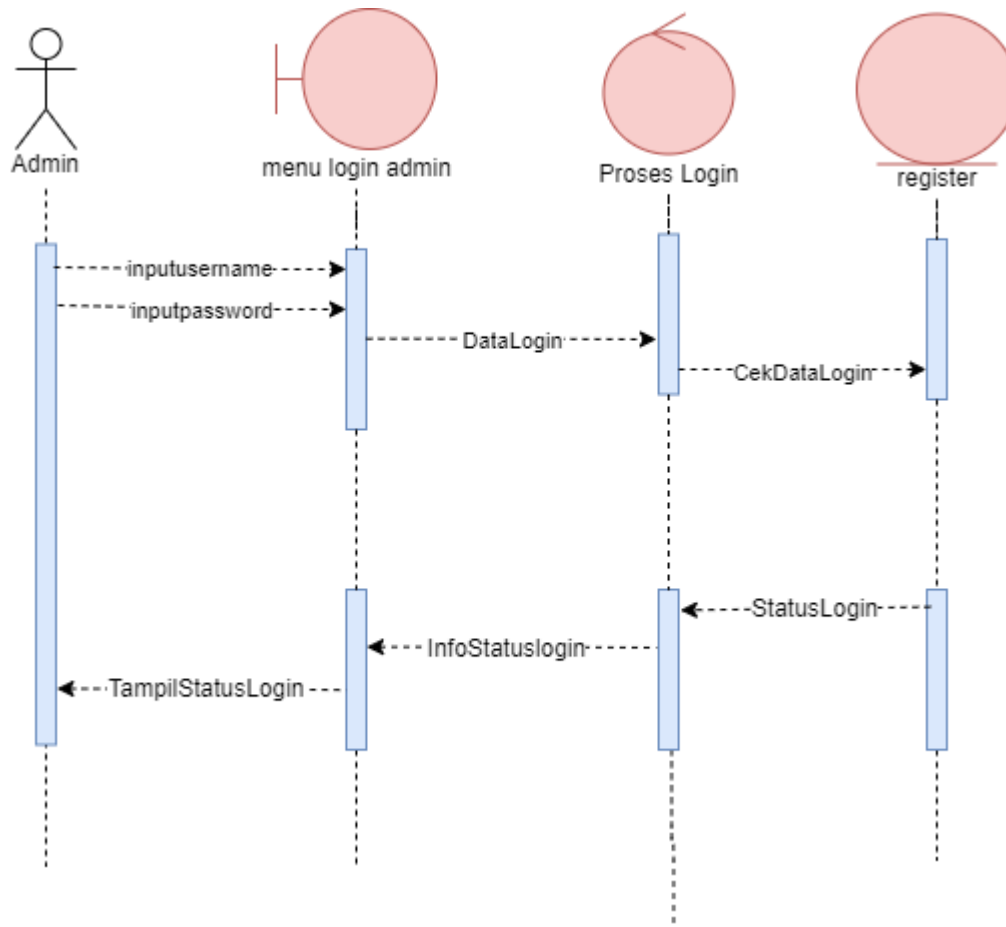
		Diagram dapat menggambarkan perancangan sistem, proses awal sistem berjalan, keputusan yang dapat terjadi dalam sistem, dan akhir dari sistem
<i>Interaction Diagram</i>	<i>Sequence Diagram</i>	adalah diagram UML yang menggambarkan hubungan di objek yang di dasarkan pada urutan waktu



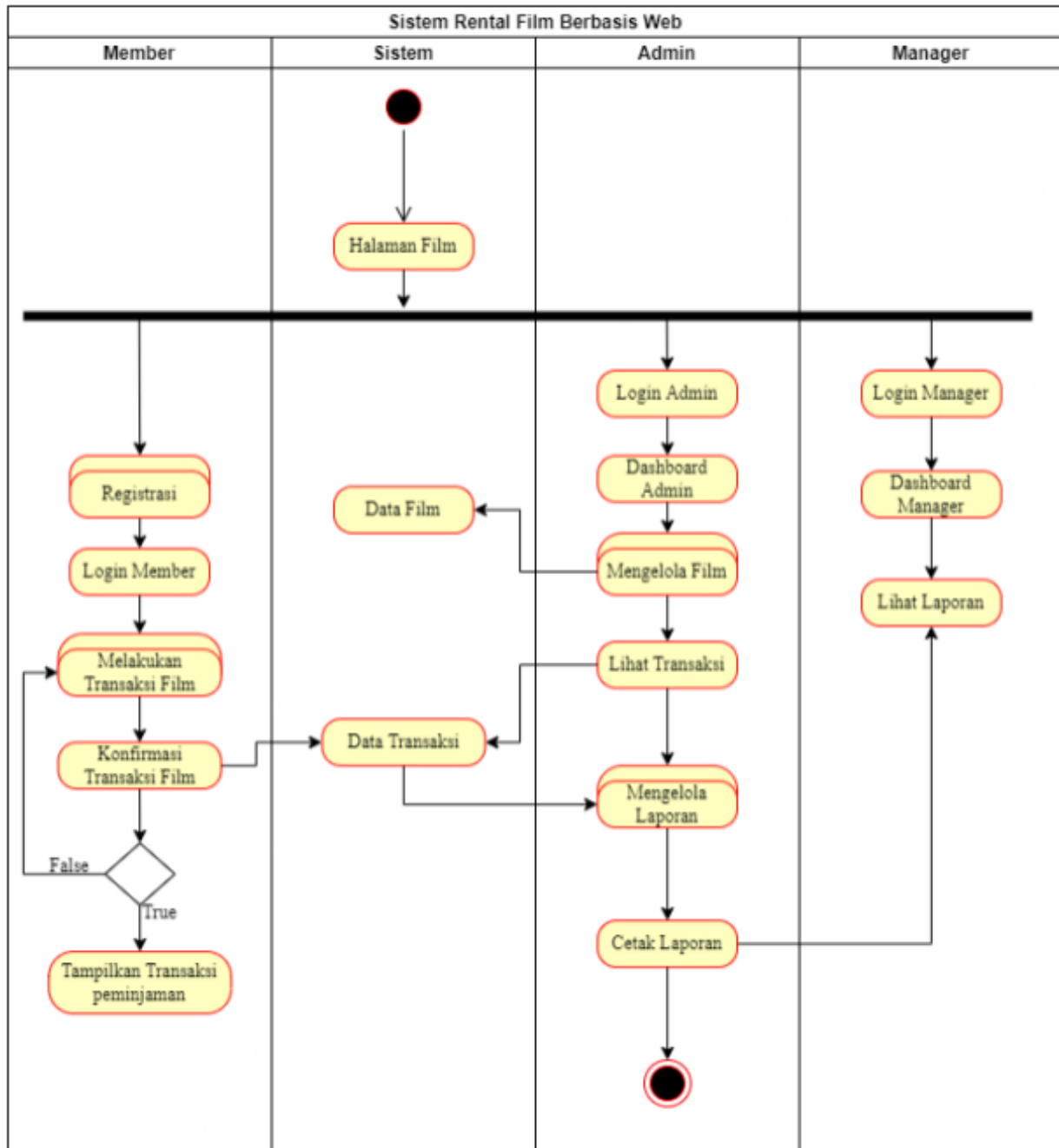
Gambar 3 contoh Use case Diagram



Gambar 4 contoh class diagram



Gambar 5 contoh sequence diagram



Gambar 6 contoh *activity diagram*