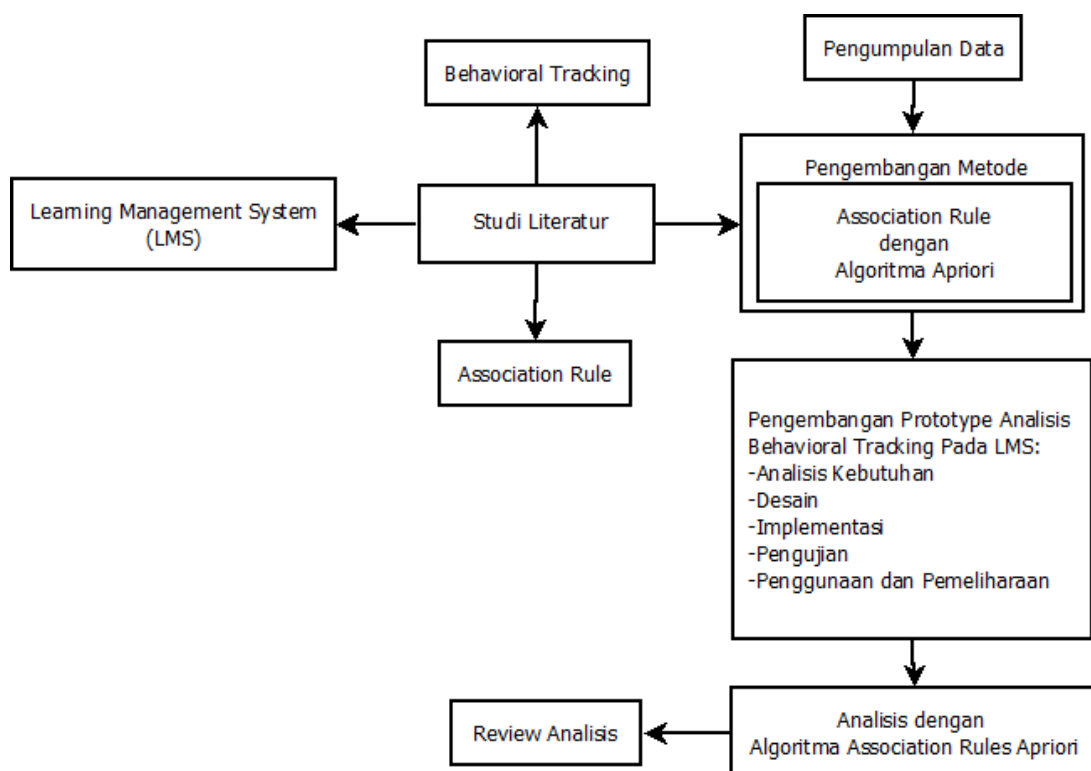


## BAB III METODOLOGI

Pada bab ini akan dijelaskan metodologi penelitian yang dilakukan data penelitian, dan desain penelitian yang digunakan.

### 1.1 Desain Penelitian

Desain penelitian yang digunakan dalam penelitian ini dapat dilihat pada Gambar 3.1 berikut ini:



Gambar 3.1. Desain Penelitian

Keseluruhan proses penelitian dilakukan secara sistematis, dan saling sambung menyambung satu sama lain. Berdasarkan gambar diatas terdapat tahapan diantaranya:

#### 1. Studi Literatur

Pada tahap ini dilakukan studi terhadap materi, dan keilmuan yang berkaitan, dan menunjang penelitian ini. Teori yang dicari diantaranya *Behavioral Tracking*, *Learning Management System (LMS)*, dan *Association Rule* yang tertera pada BAB II.

## 2. Pengumpulan Data

Pada tahap ini, dicari data berupa *log file* LMS Moodle yang cocok, dan sesuai dengan apa yang akan peneliti buat, dan analisis.

## 3. Pengembangan Metode

Tahap ini melakukan pencarian status pembelajaran siswa dengan modle MOCLog, dan juga menggunakan teknik *association rule* dengan algoritma *apriori* dari *log file* yang telah didapat untuk mengetahui aktivitas dominan penentu status pembelajaran siswa.

## 4. Pengembangan Perangkat Lunak

Pengembangan perangkat lunak pada penelitian ini menggunakan metode *waterfall* dimana tahapannya ialah analisis, desain, implementasi, pengujian, dan pengoperasian & pemeliharaan

## 5. Analisis

Setelah data hasil perangkat lunak didapat, data dianalisis sebagai bahan penarikan kesimpulan penelitian.

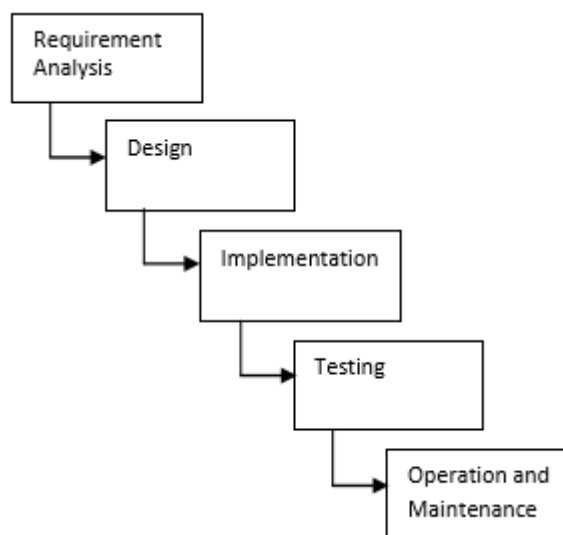
### 1.2 Metode Penelitian

#### 3.2.1 Metode Pengumpulan Data

Data yang digunakan pada eksperimen penelitian ini diambil dari data *backup course elearning* Ilmu Komputer UPI. Data ini berisi informasi *course*, kuis, tugas, log pengaksesan *course*, beserta nilai kuis, dan tugas. Namun data *backup course* yang diambil tidak mengandung data-data mahasiswa seperti nama, *username*, *password*, dan data lain yang menyangkut privasi pengguna.

### 3.2.2 Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak pada penelitian ini menggunakan metode *waterfall* dimana tahapannya ialah analisis, desain, implementasi, pengujian, dan pengoperasian & pemeliharaan. Berikut penjelasan tahapan-tahapan pengembangan perangkat lunak dengan metode *waterfall*:



Gambar 3.2. Model Waterfall (A & Adenowo, 2013)

Pendekatan pengembangan perangkat lunak yang paling sederhana dapat tergambarkan oleh model *waterfall* yang mudah dimengerti. Model ini merupakan model statis, dan melakukan pendekatan pada perangkat lunak secara sekuensial, menyelesaikan satu aktivitas sebelum menyelesaikan aktivitas lain. Model ini memiliki tujuan berbeda pada setiap fase pengembangan sebagaimana digambarkan pada gambar 3.2. Setelah setiap fase selesai dikembangkan, proses pengembangan melanjutkan ke fase selanjutnya, dan tidak bisa kembali lagi ke fase sebelumnya.

#### 1. *Requirement Analysis*

Pada tahap ini dilakukan pengumpulan kebutuhan perangkat lunak secara intensif, dan spesifik. Untuk memahami sifat program yang akan dibangun, pembuat perangkat lunak, fungsi yang diperlukan, perilaku, kinerja, dan antarmuka perangkat lunak. Semua hasil analisis ini didokumentasikan untuk ditinjau bersama pelanggan.

## 2. *Design*

Tahap ini difokuskan pada pendefinisian atribut-atribut program, seperti: struktur data, arsitektur perangkat lunak, representasi antarmuka, dan algoritma program secara mendetail. Proses ini juga menterjemahkan kebutuhan perangkat lunak ke dalam sebuah representasi perangkat lunak yang dapat dinilai kualitasnya sebelum proses pengkodean dimulai. Seperti halnya pada tahap analisis, semua pada tahap ini juga didokumentasikan, dan akan dijadikan bagian dari konfigurasi perangkat lunak.

## 3. *Implementation*

Pada tahap ini dilakukan penerjemahan hasil desain ke dalam kode program yang bisa dibaca mesin komputer. Jika desain dilakukan secara rinci, pembuatan kode dapat dilakukan secara mekanis.

## 4. *Testing*

Setelah kode program dihasilkan, tahap berikutnya adalah pengujian. Pengujian ini difokuskan pada internal perangkat lunak untuk memastikan bahwa setiap kode program diuji, dan pengujian pada fungsionalitas perangkat lunak untuk memastikan tidak terjadi *error*, serta memastikan keluaran yang dihasilkan perangkat lunak sesuai dengan target keluaran yang diharapkan.

## 5. *Operation, and maintenance*

Setelah perangkat lunak selesai dibangun maka fase selanjutnya adalah penggunaan, dan pemeliharaan. Pada tahap ini perangkat lunak dioperasikan,

dan diaplikasikan sesuai tujuan awal. Jika pada masa pemakaian terdapat gangguan atau perubahan, maka pengembang harus memperbaikinya, dan tidak mungkin melakukan seluruh fase pengembangan dari awal.