**BAB II**

**LANDASAN TEORI**

### 2.1 Unified Modeling Language (UML)

Menurut Nugroho (2010:6) “UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami”.

Menurut Widodo (2011:6), “UML adalah bahasa pemodelan standar yang memiliki sintak dan semantik”.

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OOP (*Object Oriented Programming*).

#### Model Unified Modeling Language (UML)

Menurut Widodo (2011:10), “Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram kelas (*Class Diagram*)

Bersifat statis, Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

2. Diagram paket (*Package Diagram*)

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

3. Diagram use-case (*Usecase Diagram*)

Bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan sequence (*Sequence Diagram*)

Bersifat dinamis. Diagram urutan adalah iterasiksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

5. Diagram komunikasi (*Communication Diagram*)

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.

6. Diagram statechart (*Statechart Diagram*).

Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas.

7. Diagram aktivitas (*Activity Diagram*).

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. Diagram komponen (*Component Diagram*).

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

9. Diagram deployment (*Deployment Diagram*).

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram, dan sebagainya.

#### Jenis-jenis diagram UML (Unified Modeling Language)

Berikut ini adalah definisi mengenai 5 diagram UML:

1. *Use Case Diagram*

*Use case diagram* secara grafis menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain use case diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (*user*) mengharapkan interaksi dengan sistem itu. *Use case* secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

2. *Class Diagram*

Menggambarkan struktur objek sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.

3. *Sequence Diagram*

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi.

4. *State Chart Diagram*

Digunakan untuk memodelkan behaviour objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan *event-event*(kejadian) yang menyebabkan objek beralih dari satu state ke state yang lain.

5. *Activity Diagram*

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan action yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.