

BAB II

TINJAUAN PUSTAKA

II.1. Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Contoh utama aplikasi adalah pengolah kata, lembar kerja, memanipulasi foto, merancang rumah dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja dan beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah. Jenis-jenis *Software* Aplikasi :

1. *Software* aplikasi hiburan, contohnya yaitu winamp untuk mendengarkan musik, games dan sebagainya untuk hiburan.
2. *Software* aplikasi pendidikan yaitu *software* digunakan untuk mempelajari atau mereferensikan tentang pendidikan atau pengetahuan.
3. *Software* aplikasi bisnis yaitu *software* yang digunakan untuk aplikasi bisnis.

4. *Software* aplikasi khusus

5. *Software* aplikasi untuk produktivitas kerja. (Dahlan Abdullah dan Cut Ita ; 2012 : 141)

II.2.Ujian Nasional

Menurut Dinas Pendidikan(2011), pada tahun 2011 peserta didik dinyatakan lulus Ujian Sekolah (US) tingkat SMA apabila peserta didik telah memenuhi kriteria kelulusan yang ditetapkan oleh satuan pendidikan berdasarkan perolehan Nilai Sekolah. (Ryandis Rahmatullah Firdausy, et al. ; 2012)

II.3. SKL(Standar Kompetensi Lulusan)

Menurut Menteri Pendidikan Nasional (2010), Standar Kompetensi Lulusan (SKL) adalah standar yang diberikan pemerintah untuk meningkatkan kualitas peserta didik. SKL merupakan kisi-kisi dari soal yang akan dikeluarkan pada saat UN. SKL digunakan guru sebagai pedoman dalam pembuatan soal latihan. SKL digunakan sebagai pedoman penilaian dalam menentukan kelulusan peserta didik. Selain itu, SKL bertujuan agar peserta didik dapat mengimplementasikan ilmu yang diajarkan dalam kehidupan sehari-hari. (Ryandis Rahmatullah Firdausy, et al. ; 2012)

II.4. Pemrograman Java

Java merupakan pemrograman berorientasi objek yang cukup tangguh dan sudah terbukti kelayakannya. Orientasi objek dari Java memungkinkan anda untuk membuat program dengan

memusatkan rancangan pada data (objek) serta antarmuka. Melalui proses pengembangan yang sangat baik dari *SunMicrosystem* selaku pengembang resmi, Java dari masa ke masa senantiasa *up-to-date* dalam hal teknologi.

Kelebihan-kelebihan yang dimiliki oleh java, seperti kesederhanaan, orientasi objek, keandalan, keamanan, kinerja yang tinggi, *multithread*, semakin menjadikan Java sebagai pilihan para programmer dalam mengembangkan aplikasi. Terlebih lagi dengan dukungan prinsip kuat Java, yakni *writeln anywhere*, dimana anda dapat menjalankan program Java pada berbagai mesin dengan tanpa melakukan kompilasi ulang. Hal ini kiranya tidak banyak ditemukan pada bahasa pemrograman lain. (Prasetyo, Didik dwi ; 2004 : v)

II.5. *SQLite*

SQLite adalah suatu *library* yang menerapkan mesin *databaseself-contained*, *server-less*, *zero-configuration*, dan *transactional*. *Self-contained* berarti *SQLite* membutuhkan sedikit sekali dukungan dari *library* eksternal atau dari sistem operasi. *Serverless* berarti *SQLite* dalam mengakses database baik itu *read* atau *write* dapat secara langsung dari *filedatabase* tanpa melalui proses *server* dan tidak mendukung pengaksesan secara *remote* (artinya *databaseSQLite* bisa dikendalikan dari jarak jauh dengan adanya jaringan komputer (“*ComputerNetwork*”), baik melalui jaringan lokal (*intranet*) atau *internet*), dimana kebanyakan mesin *SQLdatabase* diterapkan sebagai proses *server* yang terpisah. *Zero-configuration* menunjukkan *SQLite* tidak membutuhkan instalasi sebelum penggunaannya. *Transactional SQLite* merupakan suatu transaksional database, dimana dalam melakukan perubahan proses *query* menerapkan *Atomic*, *Consistent*, *Isolated*, and *Durable*(ACID).(Setiyadi dan Harihayati ; 2015 : 221-225)

II.6. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *OpenHandsetAlliance*, konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *OpenHandsetAlliance* menyatakan mendukung pengembangan *opensource* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *openplatform* perangkat seluler. (Nazruddin Safaat H. ; 2012 : 1)

II.7.Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform_independent*). Berikut ini adalah sifat dari *Eclipse*:

a. *Multi platform* :

Target sistem operasi *Eclipse* adalah *MicrosoftWindows*, Linux, Solaris, AIX, HP-UX dan Mac OS X.

b. *Multi language* :

Eclipse dikembangkan dengan bahasa pemrograman Java. Akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

c. *Multi role* :

Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan *web* dan lain sebagainya. (Murtiwiayati dan Lauren ; 2013 : 1-10)

II.8. Aplikasi *Mobile*

Aplikasi *Mobile* adalah perangkat lunak yang berjalan pada perangkat *mobile* seperti *smartphone* atau tablet PC. Aplikasi *Mobile* juga dikenal sebagai aplikasi yang dapat diunduh dan memiliki fungsi tertentu sehingga menambah fungsionalitas dari perangkat *mobile* itu sendiri. Untuk mendapatkan *mobileapplication* yang diinginkan, *user* dapat mengunduhnya melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. *Google Play* dan *iTunes* merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna Android dan iOS untuk mengunduh aplikasi yang diinginkan. (Muhammad Irsan ; 2015)

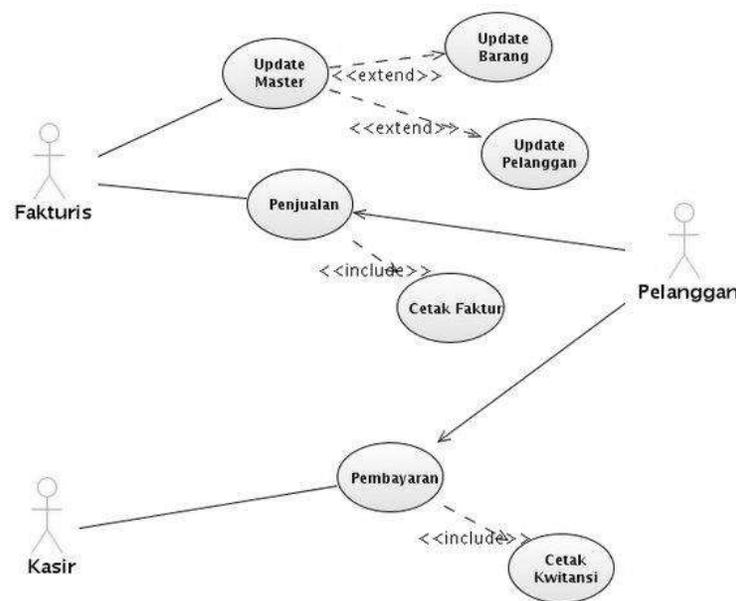
II.9. Pengertian UML (*UnifiedModelingLanguage*)

UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodalannya umum dalam industri perangkat lunak dan pengembangan

sistem (Windu dan Grace ; 2013 : 81). Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut.

II.9.1. Use Case Diagram

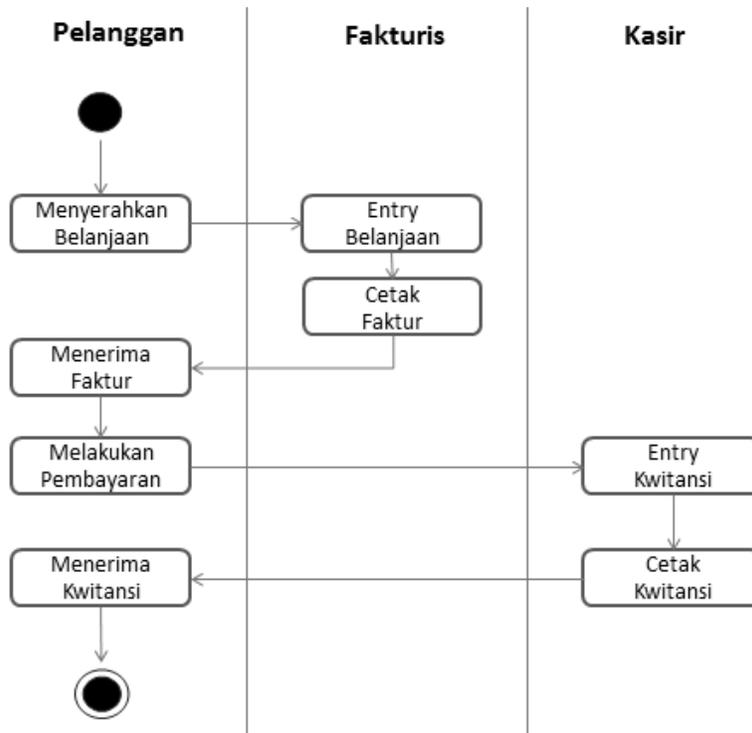
Use case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Windu dan Grace ; 2013 : 81). Contoh pembuatan *use case* diagram dapat dilihat pada gambar II.1.



Gambar. II.1. Use Case Diagram
(Sumber : Windu dan Grace ; 2013 : 83)

II.9.2. Activity Diagram

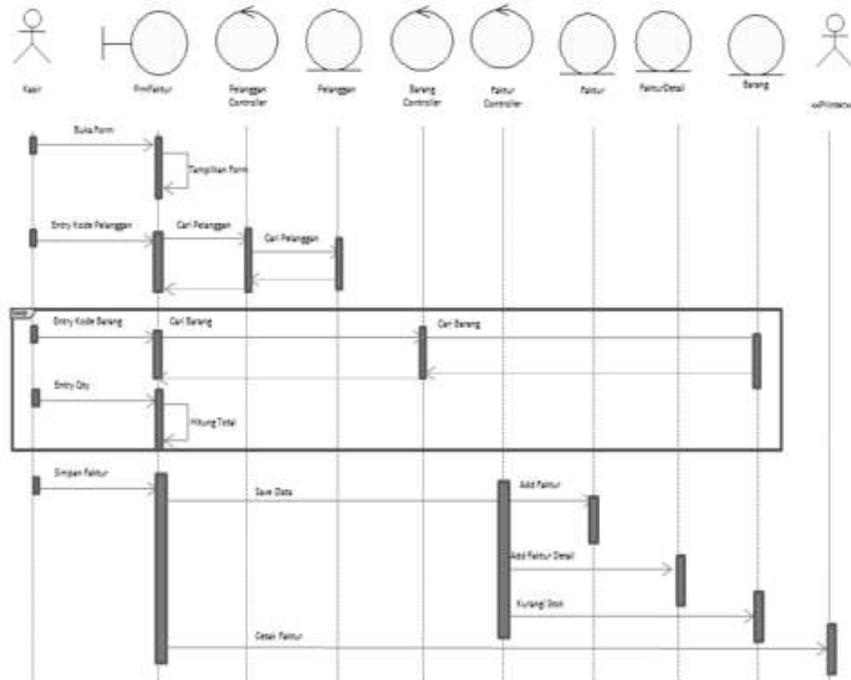
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Windu dan Grace ; 2013 : 81). Contoh pembuatan *activity diagram* dapat dilihat pada gambar II.2.



Gambar. II.2. Activity Diagram
(Sumber : Windu dan Grace ; 2013 : 83)

II.9.3. Sequence Diagram

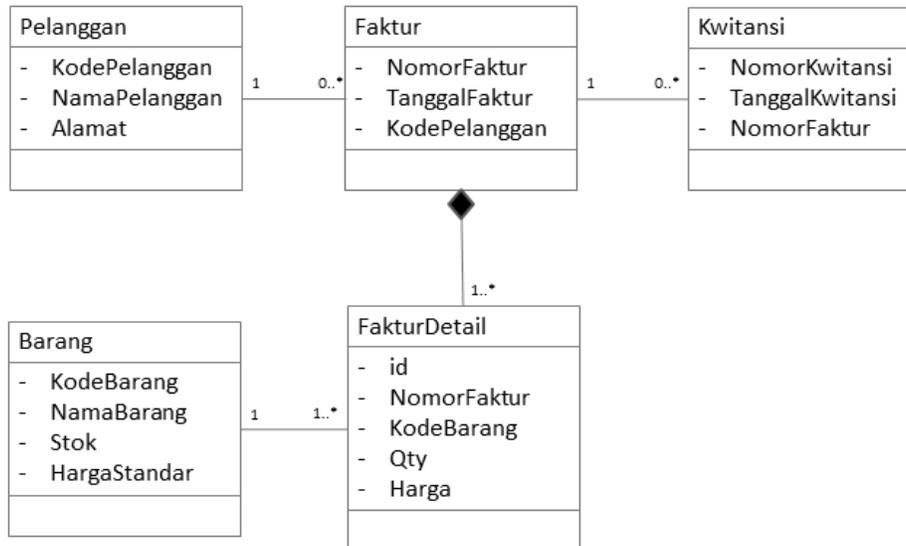
Sequence diagram menggambarkan kelakuan obyek pada *use case* dengan mendeskripsikan waktu hidup obyek dan pesan yang dikirimkan dan diterima antar obyek (Windu dan Grace ; 2013 : 81). Contoh pembuatan *sequence diagram* dapat dilihat pada gambar II.3. :



Gambar. II.3. Sequence Diagram
 (Sumber : Windu dan Grace ; 2013 : 84)

II.9.4. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan obyek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), *Relasi*, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), dan *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar Kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau kardinaliti (Windu dan Grace ; 2013 : 81). Contoh pembuatan *class diagram* dapat dilihat pada gambar II.4. berikut :



Gambar. II.4. Class Diagram
 (Sumber : Windu dan Grace ; 2013 : 83)