

BAB II

TINJAUAN PUSTAKA

II.1. Aplikasi

Aplikasi adalah suatu sub kelas 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 atarmuka 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 ditenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah. Jenis-jenis *Software* Aplikasi :

- a. *Software* aplikasi hiburan, contohnya yaitu *winamp* untuk mendengarkan musik, *games* dan sebagainya untuk hiburan.
- b. *Software* aplikasi pendidikan yaitu *software* yang digunakan untuk mempelajari atau mereferensikan tentang pendidikan atau pengetahuan.
- c. *Software* aplikasi bisnis yaitu *software* aplikasi bisnis yang digunakan

- d. untuk aplikasi bisnis.
- e. *Software* aplikasi khusus
- f. *Software* aplikasi untuk produktivitas kerja. (Abdullah dan Erlina ; 2012 : 38-39)

II.2. Video

Video merupakan media penyampai pesan termasuk media *audio-visual* atau media pandang-dengar. Media *audio-visual* dapat dibagi menjadi dua jenis: pertama, dilengkapi fungsi peralatan suara dan gambar dalam satu unit, dinamakan media *audio-visual* murni dan kedua, media *audio-visual* tidak murni. Film bergerak, televisi, dan *video* termasuk jenis yang pertama, sedangkan *slide*, *opaque*, OHP dan peralatan *visual* lainnya yang diberi suara termasuk jenis yang kedua. (Purwanti ; 2015 : 42-47)

Adapun *screenshot* dari *video* dapat dilihat pada gambar II.1. berikut ini :



Gambar II.1. Screenshot Video

(Sumber : <https://www.youtube.com/> Full Video, Membuat Aplikasi Android dengan Java, Eclipse, Android SDK Tutorial)

II.3. 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 piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*. (Murtiwiwati dan Lauren; 2013 : 1-10)

Android mempunyai banyak sistem operasi dengan versi berbeda yang dapat dilihat pada gambar II.2.



Gambar II.2. Versi Android
(Sumber : <https://www.android.com> ; Diakses pada 29 Juni 2016)

II.4. GIF (*Graphic Interchange Format*)

GIF (*Graphic Interchange Format*) adalah format citra digital yang mendukung kompresi *lossless*, yaitu kompresi yang tidak menghilangkan informasi citra digital. Format ini tidak mengurangi kualitas citra digital dibandingkan dengan aslinya, namun kekurangan terbesarnya adalah terbatasnya kedalaman warna yang didukung yaitu hanya sampai 256 warna. (Hendri ; 2014 : 28)

Format *file* ini memiliki ukuran yang sangat kecil dan sederhana. Format *file* ini memiliki kombinasi warna yang terbatas dibandingkan dengan format grafis yang lain karena hanya mendukung 256 warna dan memiliki kedalaman *bit* sebesar 8 *bit* saja. Format GIF hanya mendukung mode warna *Grayscale*, *Bitmap* dan *Indexed Color*. Namun ada kelebihan dari format ini, format ini bisa diaplikasikan dalam animasi grafis dua dimensi sederhana dan juga dapat mengaplikasikan *transparency masking*. Untuk pemanfaatan animasi *banner* sederhana dalam *website* biasanya pengguna grafis memanfaatkan format *file* ini karena ukurannya sangat kecil sehingga tidak membebani *website*. Format GIF tidak cocok apabila diterapkan pada gambar atau objek foto dengan level warna yang rumit dikarenakan hanya mendukung image 256 warna saja. (Sukanto ; 2010: 31)

II.5. *Eclipse Juno*

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 *Microsoft Windows*, *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, tes perangkat lunak, pengembangan *website* dan lain sebagainya. (Murtiwiwati dan Lauren ; 2013 : 1-10)

II.6. *Android SDK (Software Development Kit)*

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman *JAVA*. Sebagai *platform* aplikasi netral, *Android* memberi kesempatan bagi semua orang untuk membuat aplikasi yang dibutuhkan, yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur *Android* yang paling penting adalah :

- a. Mesin *Virtual Dalvik* yang dioptimalkan untuk perangkat *mobile*.
- b. *Integrated browser* berdasarkan *engine open source WebKit*.
- c. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1.0* (Opsional akselerasi *hardware*).
- d. SQLite untuk penyimpanan data (*database*).
- e. Media yang mendukung *audio*, *video*, dan gambar.
- f. *Bluetooth*, EDGE, 3G dan WiFi.
- g. Kamera, GPS, dan kompas.
- h. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE *Eclipse*. (Lindung , Yunus Dwi ; 2012)

II.7. AVD (*Android Virtual Device*)

AVD (*Android Virtual Device*) merupakan *emulator* untuk menjalankan aplikasi android. Setiap AVD terdiri dari:

- a. Sebuah profil perangkat keras yang dapat mengatur pilihan untuk menentukan fitur *hardware emulator*. Misalnya, menentukan apakah menggunakan perangkat kamera, apakah menggunakan *keyboard QWERTY* fisik atau tidak, berapa banyak memori internal, dan lain-lain.
- b. Sebuah pemetaan versi android, maksudnya kita menentukan versi dari *platform* android akan berjalan pada *emulator*.
- c. Pilihan lainnya, misalnya menentukan *skin* yang kita ingin gunakan pada *emulator*, yang memungkinkan untuk menentukan dimensi layar, tampilan, dan sebagainya. Kita juga dapat menentukan *SD Card virtual* untuk

digunakan di *emulator*. (Lindung , Yunus Dwi ; 2012)

II.8. XML

XML merupakan dasar terbentuknya *web service* yang digunakan untuk mendeskripsikan data. Pada level paling *detail web service* secara keseluruhan dibentuk diatas XML. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan partner luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lain. (Deviana ; 2011 : 61)

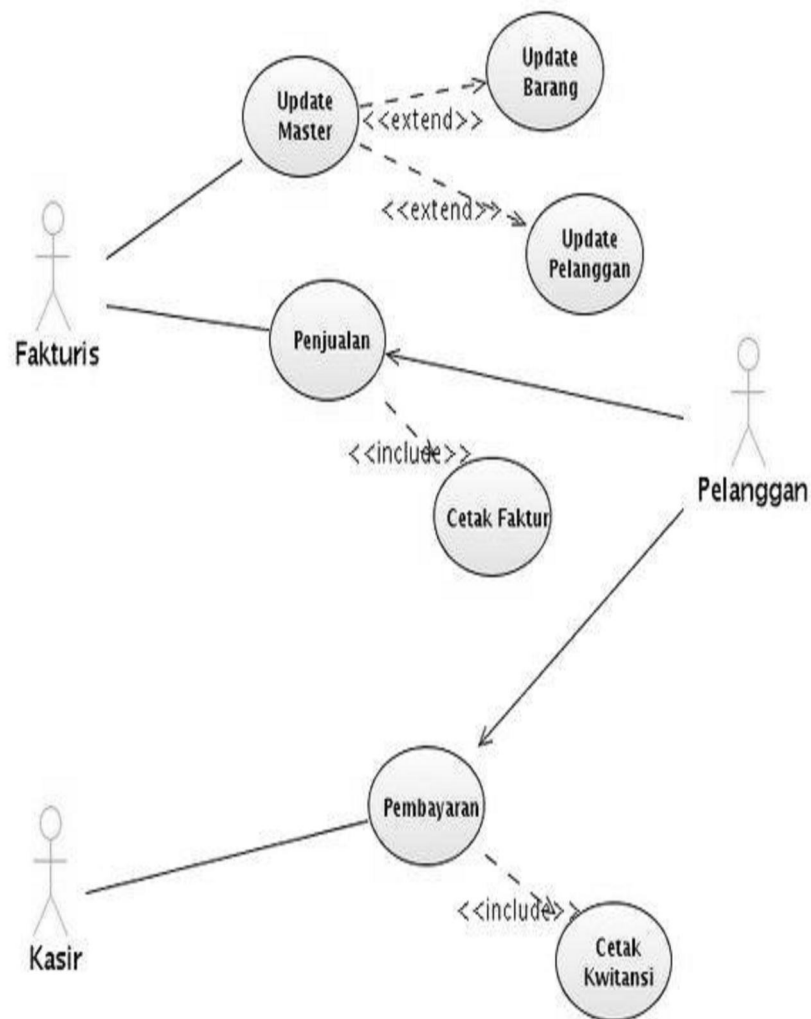
II.9. Pengertian UML

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 pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Windu dan Grace ; 2013 : 81). Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis 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

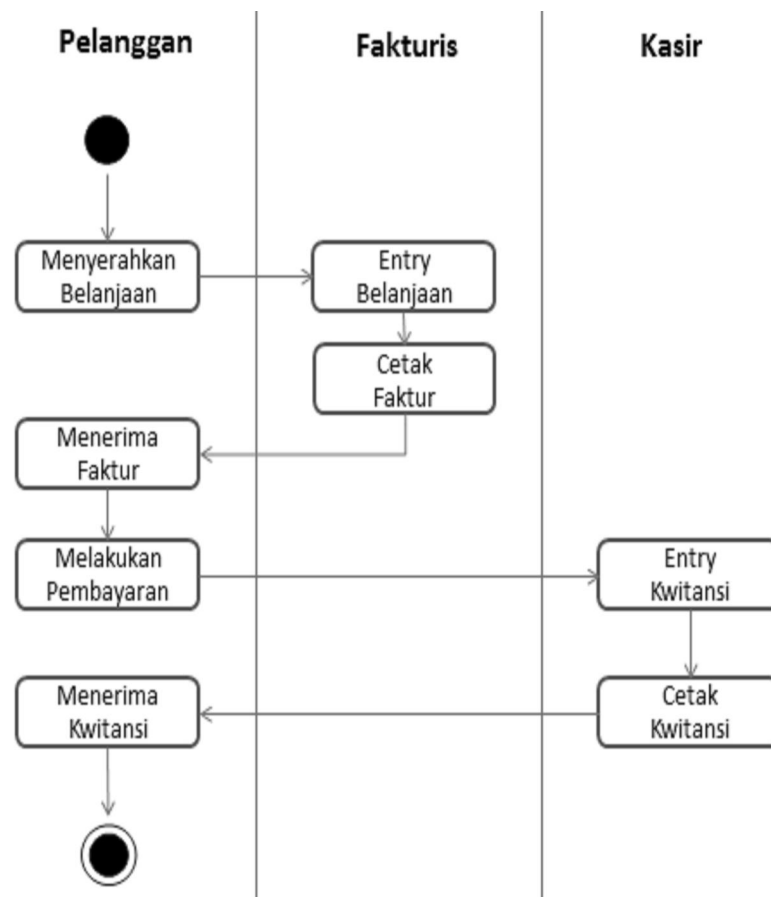
informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Windu dan Grace ; 2013 : 81). Contoh pembuatan *use case diagram* dapat dilihat pada gambar II.3. berikut :



Gambar. II.3. Use Case Diagram
(Sumber : Windu dan Grace ; 2013 : 83)

II.9.2. Activity Diagram

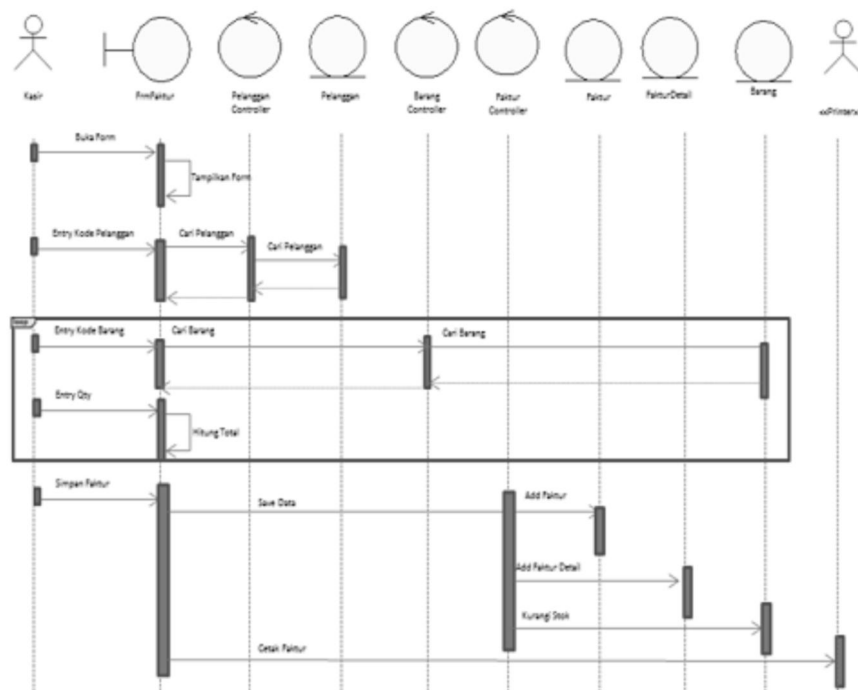
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Windu dan Grace ; 2013 : 81). Contoh pembuatan *activity diagram* dapat dilihat pada gambar II.4. berikut :



Gambar. II.4. 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.5. :

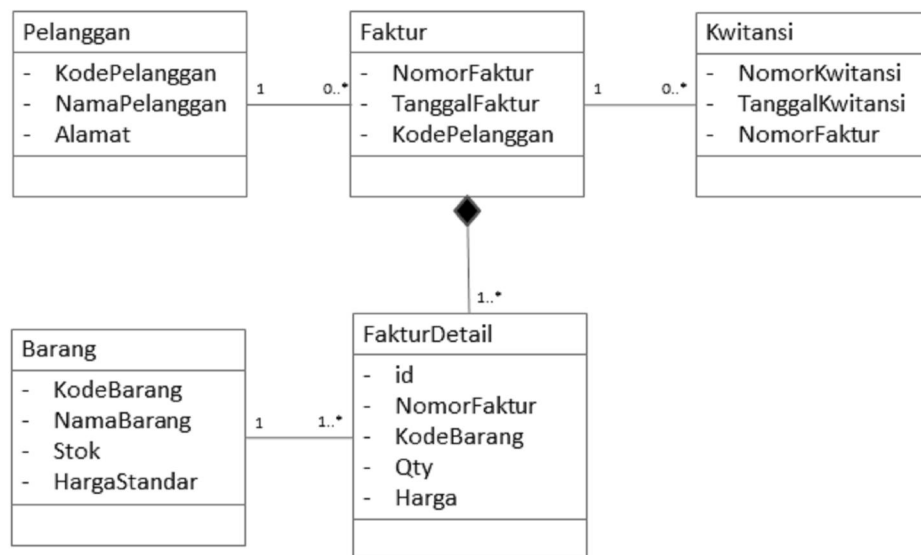


Gambar. II.5. Sequence Diagram
(Sumber : Windu dan Grace ; 2013 : 84)

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.6. berikut :



Gambar. II.6. Class Diagram
(Sumber : Windu dan Grace ; 2013 : 83)