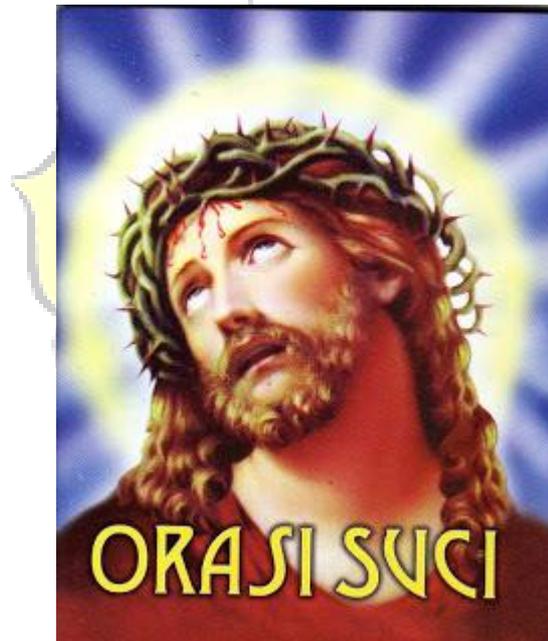


BAB II

LANDASAN TEORI

2.1 Orasi Suci

Orasi Suci adalah doa-doa dan kisah sengsara Yesus Kristus yang dibukukan agar orang katolik dapat merenungkan kisah sengsara Yesus Kristus. Salinan Surat Suci Orasi Yesus ini, aslinya ada di makam suci Yesus penebus manusia di yerusalem, di semayamkan dalam sebuah peti perak oleh Bapa Suci dan para Raja dan Ratu Iman Kristen.



gambar 2.1 orasi suci(buku orasi suci, 1988)

2.2 Novena Katolik

Dalam Gereja Katolik, novena adalah doa dan devosi sembilan kali atau sembilan hari berturut-turut. Di dalamnya, orang menyampaikan permohonan untuk rahmat khusus yang diminta. Dalam tradisi ada empat kategori permohonan novena: oleh mereka yang sedang berduka, sedang mempersiapkan hari-hari perayaan besar, sedang berharap untuk permintaan khusus, dan sedang memohon silih atas dosa-dosa yang telah dilakukan. Novena tentu saja bisa meliputi lebih dari satu intensi, dan bisa dilakukan entah di rumah, gereja, atau tempat lain yang kiranya cocok dan pantas untuk membangun suasana doa. Novena tidak memiliki basis acuan langsung dalam Kitab Suci. Tradisi berdoa ini malah lebih dipengaruhi adat kebiasaan keluarga dan orang-orang Yunani dan Romawi. Ketika mereka berduka atas kematian anggota keluarga, mereka berdoa sembilan kali, yang kemudian diikuti dengan perayaan dan syukur (Bdk Kis 1:14). Tradisi melanjutkan keyakinan bahwa Bunda Maria dan para murid berdoa sembilan kali menjelang datangnya Roh Kudus. Novena adalah praktik yang pada dasarnya dilakukan di lingkungan Gereja Katolik. Meski demikian, sebagian umat Ortodoks, Anglikan, dan Lutheran juga menekuni cara berdoa ini.



Gambar 2.2 Novena (www.doanovena.com)

2.3 Aplikasi Android

Menurut Teguh Arifianto (2011 : 1), android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Menurut Hermawan (2011 : 1), Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

Berdasarkan pendapat diatas, maka dapat ditarik kesimpulan bahwa android adalah sistem operasi berbasis linux yang sedang berkembang ditengah OS lainnya.

2.3.1 Karakteristik Android

Menurut sumber (sumber : <http://www.android.com.about/>) Android memiliki empat karakteristik sebagai berikut :

1. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

2. Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (third-party application). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang

diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

(sumber : <http://www.android.com/about/>)

2.3.2 Fitur-Fitur Android

Android memiliki beberapa fitur yang menarik bagi yang ingin mengembangkan aplikasi, diantaranya sebagai berikut:

- a. Kerangka aplikasi : itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- b. Dalvik mesin virtual : mesin virtual dioptimalkan untuk perangkat telepon seluler.
- c. Grafik : grafik 2D dan grafis 3D berdasarkan pustaka OpenGL.
- d. SQLite : untuk penyimpan data.
- e. Mendukung media : audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- f. GSM, Bluetooth, EDGE, 3G, 4G, dan Wifi (tergantung piranti keras).

- g. Kamera, Global Positioning System (GPS), kompas, NFC dan accelerometer (tergantung piranti keras).

2.3.3 Pengembangan Aplikasi Android

Ada 4 hal mendasar yang harus kita pahami dalam membangun aplikasi berbasis Android:

- a. *Activity*, adalah tampilan grafis yang kita lihat ketika menjalankan sebuah aplikasi. Aplikasi dapat memiliki lebih dari satu *Activity*.
- b. *Intent*, adalah serangkaian nilai yang menunjukkan apa yang harus dilakukan ketika terjadi perpindahan layar.
- c. *Service*, adalah layanan yang bekerja di belakang layar (*background*).
- d. *Content provider*, memungkinkan sebuah aplikasi untuk dapat menyimpan dan menerima data dari basis data. (Nazrudin Safaat, 2012)

2.4 Java

Java merupakan bahasa pemrograman yang berbasis objek, maksudnya adalah semua aspek yang terdapat pada *java* adalah objek sehingga sangat memudahkan untuk mendesain , membuat dan mengembangkan program java dengan cepat.

Java adalah bahasa yang dimana para pengembang mengekspresikannya dengan *Source Code* (program text), *Java Syntax* sebagian mengikuti pola dari bahasa C/C++. *Java* adalah *platform* untuk mengeksekusi program. *Java platform* terdiri dari *virtual machine* dan lingkungan dalam eksekusiya. *Virtual machine* adalah

processor berdasarkan kumpulan instruksi *software* itu sendiri. Lingkungan eksekusilah yang terdiri dari *library* untuk menjalankan program dan berinteraksi dibalik operasi sistem tersebut. (*Jeff Friesen* : 2010)

Kelebihan

a. Multiplatform.

Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja*. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan.

b. OOP (Object Oriented Programming)

Pemrogram Berorientasi Objek yang artinya semua aspek yang terdapat di Java adalah Objek. Java merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis Java secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan Java sebagai salah satu bahasa pemrograman termudah.

c. Perpustakaan Kelas Yang Lengkap,

Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d. Bergaya C++,

memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java.

e. Pengumpulan sampah otomatis

memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

Kekurangan

a. Tulis sekali, jalankan di mana saja

Masih ada beberapa hal yang tidak kompatibel antara *platform* satu dengan *platform* lain. Untuk J2SE, misalnya *SWT-AWT bridge* yang sampai sekarang tidak berfungsi pada Mac OS X.

b. Mudah didekompilasi.

Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*reverse-engineer*.

c. Penggunaan memori yang banyak.

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). (Maleh Rupo, 2013)

2.5 Perangkat Lunak Yang Digunakan Dalam Pembuatan Aplikasi

2.5.1 IDE Eclipse

Eclipse adalah merupakan sebuah *integrated Development Environmet* (IDE), yaitu perangkat lunak serba ada yang fungsinya

untuk menulis (*Writing*), merubah (*editing*), *compiling*, dan menjalankan program komputer. Eclipse bukan merupakan *Java Development Environment* semata, Eclipse sebenarnya hanya wadah, tempat untuk banyak jenis *add-ons* bahasa pemrograman, mulai dari java, C++, atau bias juga COBOL development environment. Setiap *add-ons* dikenal dengan nama *Plug-in*, dan Eclipse memiliki lebih dari 80 *Plug-in* yang sangat berguna.(Barry Burd, 2004)

Android Development Tools (ADT) merupakan salah satu *Plug-in* Eclipse, *Plug-in* diperlukan untuk membangun sebuah aplikasi android, gunanya adalah penghubungan antara IDE Eclipse dengan android SDK. Peralatan yang ada dalam paket SDK android juga mengikutsertakan emulator di dalamnya sehingga kita tidak memerlukan perangkat mobile dengan OS Android untuk membuat sebuah aplikasi Android. (Nazruddin Safaat H, 2012)

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Dikembangkan oleh Sun Microsystems dan diterbitkan tahun 1995. Java tidak boleh disalahpahami sebagai JavaScript. JavaScript adalah bahasa scripting yang digunakan oleh web browser.

2.5.2 SQLite

SQLite adalah *embedded relational* yang bersifat *open source*. Dirilis pada tahun 2000, didesain untuk menyediakan cara yang mudah bagi sebuah aplikasi untuk mengatur data tanpa perlu ada biaya tambahan yang biasanya diperlukan untuk membuat suatu *relation database* tersendiri. SQLite sudah banyak dikenal karena mobilitasnya yang tinggi, mudah digunakan, kompak, dan handal.

SQLite adalah sebuah embeded database, sehingga SQLite tidak menjalankan prosesnya secara tersendiri di tempat lain, melainkan berada pada aplikasi itu sendiri. Kode dari SQLite menjadi satu atau tertanam dalam aplikasi tersebut. Keuntungannya adalah kita tidak perlu direpotkan dengan pengaturan jaringan, pengaturan administrative, hak akses, dan tambahan biaya lain (Grant Allen dan Mike, 2010)

2.5.3 Adobe Photoshop CS3

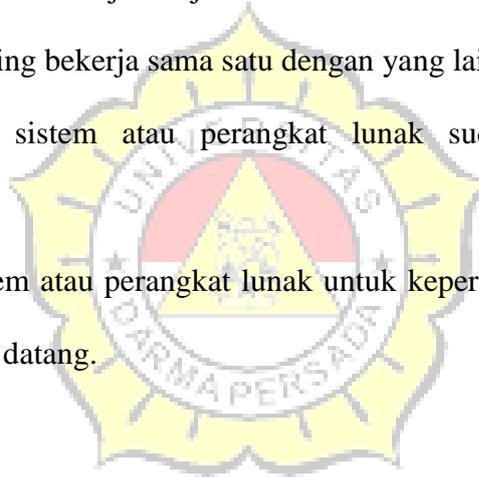
Adobe Photoshop adalah sebuah program untuk mengelola gambar. Program ini didesain untuk membantu mengelola gambar baik itu berupa gambar digital, foto, dan lainnya. Apapun sumber gambar tersebut, hasil foto dari kamera digital, gambar yang di scan menggunakan scanner, ataupun gambar yang dibuat langsung di Photoshop, semuanya tersusun dari persegi kecil berwarna yang disebut juga dengan pixels. Photoshop pada dasarnya merubah dan mengatur warna yang terdapat pada persegi tersebut, bias secara satu persatu ataupun sekaligus dalam

satu waktu untuk membuat hasil gambar sesuai dengan apa yang user inginkan. (Peter Bauer, 2007)

2.6 Unified Modeling Language

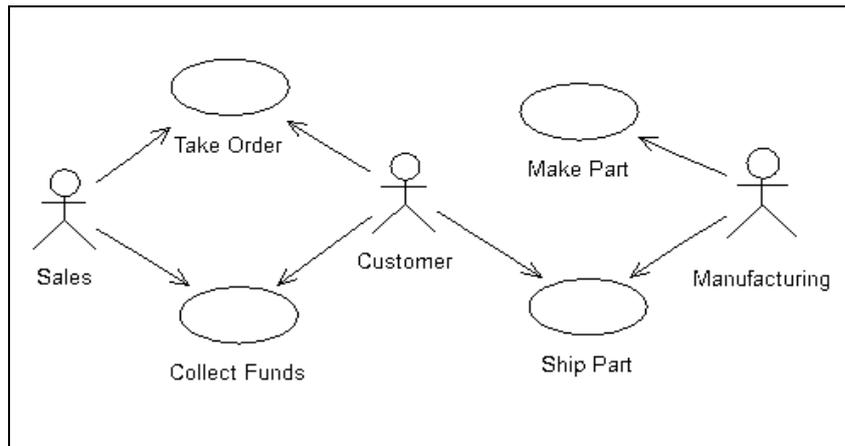
Unified Modeling Language (UML) (Adi Nugroho, 2005) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek. Dengan menggunakan UML, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaahan bagaimana objek-objek dalam sistem saling mengirimkan pesan (*message*) dan saling bekerja sama satu dengan yang lain.
3. Menguji apakah sistem atau perangkat lunak sudah berfungsi seperti seharusnya.
4. Dokumentasi sistem atau perangkat lunak untuk keperluan-keperluan tertentu dimasa yang akan datang.



2.6.1 Use Case Diagram

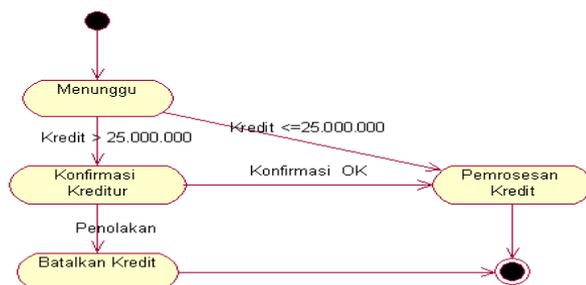
Use case (Adi Nugroho, 2005) adalah peringkat tertinggi dari fungsionalitas yang dimiliki sebuah sistem. *Diagram use case* memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case-use case* dalam sistem. Hal ini berfungsi agar para pengguna dapat melihat fungsionalitas yang disediakan oleh sistem dan menyetujui atau menolak lingkup sistem sebelum proyek berjalan lebih jauh dan akan menghabiskan waktu dan biaya. Berikut contoh sederhana *use case diagram*. (Adi Nugroho, 2005)



Gambar 2.3 Contoh *use case* (Adi Nugroho, 2005)

2.6.2 Activity Diagram

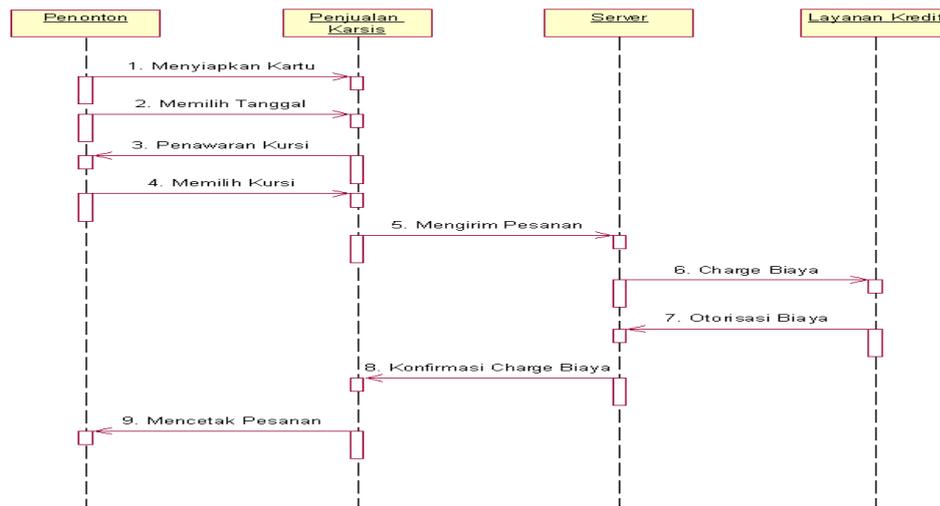
Activity diagram adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*. *Activity diagram* mungkin berdiri sendiri untuk memvisualisasikan, menspesifikasi, mengkonstruksi, serta mendokumentasi sifat dinamis dari sekumpulan objek. Namun, *activity diagram* tidak hanya penting untuk pemodelan aspek dinamis, tetapi juga penting untuk mengkonstruksi sistem-sistem yang dapat dieksekusi. Berikut ini merupakan contoh dari *activity diagram*



Gambar 2.4 Contoh *activity diagram* (Adi Nugroho, 2005)

2.6.3 Sequence Diagram

Sequence diagram adalah *interaction diagram* yang memperlihatkan *event-event* yang berurutan sepanjang berjalannya waktu. Masing-masing *sequence diagram* menggambarkan aliran-aliran pada suatu *use case*. Berikut ini merupakan contoh dari *sequence diagram*. (Adi Nugroho, 2005)



Gambar 2.5 Contoh *sequence diagram* (Adi Nugroho, 2005)

2.7 XML

XML (*Extensible Markup Language*) adalah merupakan singkatan dari *Extensible Markup Language*. Dalam kaitannya bahasa pemrograman, kata *extensible* berarti pengembang dapat mengembangkan kemampuannya. Informasi dalam XML disimpan dalam tag. Markup berarti sebuah bahasa yang dibangun dengan menggunakan tag untuk mengelilinginya atau markup text.

Dokumen XML berisi informasi dan markup yang dibagi menjadi beberapa bagian yang penting yaitu :

1. *Root node* yaitu node yang melingkupi keseluruhan dokumen. Dalam satu dokumen XML hanya ada satu *root node*. Node-node yang lainnya berada di dalam *root node*.
2. *Element node* yaitu bagian dari dokumen XML yang ditandai dengan tag pembuka dan tag penutup, atau bisa juga sebuah tag tunggal elemen kosong seperti `<anggota nama="budi"/>`. *Root node* biasa juga disebut *root element*.
3. *Attribute note* termasuk nama dan nilai atribut ditulis pada tag awal sebuah elemen atau pada tag tunggal.
4. *Text node*, adalah text yang merupakan isi dari sebuah elemen, ditulis diantara tag pembuka dan tag penutup
5. *Comment node* adalah baris yang tidak dieksekusi oleh parser
6. *Processing Instruction node*, adalah perintah pengolahan dalam dokumen XML. Node ini ditandai awali dengan karakter `<?` Dan diakhiri dengan `?>`. Tapi perlu diingat bahwa header standard XML `<?xml version="1.0" encoding="iso-8859-1"?>` bukanlah *processing instruction node*. *Header standard* bukanlah bagian dari hirarki pohon dokumen XML.
7. *NameSpace Node*, node ini mewakili deklarasi *namespace*. (Andi Sunyoto, 2010)