

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Hasil penelitian terdahulu merupakan referensi bagi peneliti untuk melakukan penelitian ini. Penelitian terdahulu, yakni :

1. Mirza Eka Pramudika, Anjik Sukmaaji, dan Achmad Holil Noor Ali. (2014), dalam jurnalnya yang berjudul “Aplikasi Social Network For School Berbasis Mobile. (Studi kasus sekolah islam Al-Azhar Jawa Timur)”. Pokok permasalahan yang diangkat ialah ingin mengetahui tentang pertukaran informasi antara guru dan orangtua sekitar fenomena dan peristiwa yang meliputi dari social network, seperti facebook dan twitter bersifat terbuka dan memperbolehkan semua orang untuk berpartisipasi. Tetapi berdasarkan penelitian yang dilakukan, melalui social network, seorang user akan lebih mudah berekspresi mengenai hal yang berhubungan dengan privasi dan *personal information*. Sistem ini dalam menjalankan aplikasinya seorang user mengisi email dan passwordnya masing-masing berbeda.
2. Bangun Wijayanto, S.T.,M.Cs (2013), dalam jurnalnya yang berjudul “Penggunaan Jejaring Sosial Scringo Pada UKMDROID (CASHREGISTER BERBASIS ANDROID).” Memiliki tujuan untuk menghitung, merekam transaksi penjualan dan tersambung pada brankas

uang. Cash register biasanya juga dapat mencetak nota dari konsumen (wijayanto, 2012). Pada usaha mikro di pedesaan penggunaan cash register sangat jarang di jumpai, umumnya setiap transaksi akan dicatat secara manual bahkan tidak jarang harga suatu barang tidak tercatat melainkan diingat oleh penjual. Minimnya penggunaan teknologi informasi dalam proses perekaman transaksi membuat usaha mikro sulit untuk berkembang. UKMDroid adalah aplikasi cash register berbasis android yang dikembangkan oleh Bangun Wijayanto pada tahun 2011. Aplikasi tersebut dapat mengatasi masalah penggunaan cashregister pada usaha mikro dengan biaya yang murah.

3. Berdasarkan hasil dari penelitian yang terdahulu, maka dibuatlah kesimpulan untuk merancang sistem suatu proyek aplikasi, yang nantinya akan dapat di implementasikan, penelitian yang akan dibangun yaitu dengan judul : Rancang Bangun Aplikasi Social Network Untuk Kampus Berbasis Android, aplikasi ini bersifat private, dan menggunakan sistem SSO (Single Sign On). Dengan mekanisme SSO, pengguna hanya membutuhkan satu email dan satu password. Dalam penggunaanya, pengguna perlu sekali login untuk melihat semua status yang dibuat pada website melalui aplikasi ini. Sehingga pengguna tidak perlu mengingat beberapa akun, hanya satu akun untuk sekali login dan tidak diperlukan untuk login berulang kali, Aplikasi ini berisi login, pemberitahuan, dan Obrolan.

II.2. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. (Fricles Ariwisanto Sianturi ; 2013 : 43).

II.3. Single Sign On (SSO)

Single Sign On (SSO) adalah sebuah mekanisme dimana masing-masing *user* hanya memiliki satu akun yang berfungsi sebagai identitas user satu-satunya. Satu akun tersebut dapat digunakan untuk meminta izin dari sistem agar *user* dapat mengakses beberapa layanan aplikasi dengan *email* dan *password* yang sama. Sistem SSO menghindari login ganda dengan cara mengidentifikasi subjek secara ketat dan memperkenankan informasi otentifikasi untuk digunakan dalam sistem yang terpercaya. Sistem SSO dapat meningkatkan kegunaan jaringan secara keseluruhan dan pada saat yang sama dapat memusatkan perolehan dari parameter sistem yang relevan (Hero Wintoro, Dalijan, & Candra Kasih Ratri Marlina. Vol. 5, No.2, 2013).

II.4. Sign On (SO)

Sign On (SO) adalah sebuah mekanisme dimana masing-masing user memiliki username dan password yang berbeda, sehingga tidak diketahui orang lain. Maka *SO (Sign On)* merupakan hak untuk masuk, memakai, dan memanfaatkan kawasan atau zona-zona tertentu (Schlanger dan Ostrom : 1992).

II.5. Android

Menurut Arifianto Teguh, android adalah sebuah *platform* pertama yang betul-betul terbuka dalam pengembangannya dan komprehensif untuk perangkat *mobile*, semua perangkat lunak yang ada difungsikan menjalankan sebuah *device mobile* tanpa memikirkan kendala kepemilikan yang menghambat inovasi.

Dalam definisi lain, Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware*, dan aplikasi inti yang dirilis oleh Google. (Ahmad; 2015 : 191)

II.5.1. Sejarah Android

Pada awal mulanya Android Inc, merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Pao Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis IT & Communication; Andy Rubin, Rich Miner, Nick Searr dan Chris White. Menurut Rubin, Android Inc didirikan untuk mewujudkan mobile device yang lebih peka terhadap lokasi preferensi pemilik. Dengan kata lain, Android Inc, ingin mewujudkan mobile device yang lebih mengerti pemiliknya. Konsep yang

dimiliki Android Inc , ternyata mengunggah miniat Google untuk memilikinya. Pada bulan Agustus 2005, akhirnya Android Inc diakuisisi oleh Google Inc. Seluruh sahamnya dibeli oleh Google.

Banyak yang memperkirakan nilai pembelian Android Inc oleh Google adalah sebesar USD 50 juta. Saat itu banyak berspekulasi, bahwa akuisisi ini adalah langkah awal yang dilakukan Google untuk masuk ke pasar mobile phone. Andy Rubin, Rich Miner, Nick Searr dan Chris White tetap di Android Inc yang dibeli Google, sehingga mereka semua menjadi bagian dari raksasa Google dan sejarah android. Saat itulah mereka mulai menggunakan platform linux untuk membuat sistem operasi bagi mobile phone. Saat ini terdapat dua jenis distributor sistem operasi di dunia. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service* (GMS), dan yang kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung dari Google atau dikenal dengan *Open Handset Distribution* (OHD). (Akhmad Dharma Kasman ; 2015 : 2)

II.5.2. Perkembangan Versi OS Android

Keunikan dari nama sistem operasi (OS) Android adalah dengan menggunakan nama makanan hidangan penutup (Dessert). Selain itu juga nama-nama OS Android memiliki huruf awal berurutan sesuai abjad; Cupcake, Donut, Eclair, Froyo, Gingerbread, Honey Comb, Ice Cream, Jelly Bean. Namun juru bicara Google, Randall Sarafa enggan member tahu alasannya. Sarafa hanya menyatakan bahwa pemberian nama-nama itu merupakan hasil keputusan internal dan Google memilih tampil sedikit ajaib dalam hal ini.

1. Android Beta

Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 software Development Kit (SDK) dirilis oleh Google. (Akhmad Dharma Kasman ; 2015 : 4).

2. Android 1.0 Astro

Pertama kali dirilis pada 23 September 2008. Sebenarnya Android versi pertama ini akan dinamai “Astro” tapi karena alasan hak cipta dan trademark nama “Astro” tidak jadi disematkan pada versi pertama dari OS Android ini. HTC Dream adalah ponsel pertama yang menggunakan OS ini. (Akhmad Dharma Kasman ; 2015 : 4)

3. Android 1.1 Bender

Pertama kali dirilis pada 9 Februari 2009. Versi Android kedua ini juga mengalami masalah penamaan yang sama dengan versi pertamanya. Pada awalnya Android ini akan diberi nama “Bender” akan tetapi karena alasan melanggar trademark, nama “Bender” tidak jadi disematkan pada versi Android ini. Awalnya versi OS Android ini dirilis untuk perangkat T-Mobile G1 saja. Versi ini merupakan update untuk memperbaiki beberapa bugs, mengganti API dan menambah beberapa fitur. (Akhmad Dharma Kasman ; 2015 : 4)

4. Android 1.5 Cupcake

Pertama kali dirilis pada 30 April 2009. Nah, mulai versi Android ini penamaan menggunakan nama makanan pencuci mulut (dessert) mulai digunakan, karena ini versi yang ketiga maka penamaan dimulai dari huruf “C” dan jadilah

“Cupcake” menjadi nama resmi dari versi OS android ketiga ini. OS ini berbasiskan pada kernel Linux 2.6.27 dan menambah beberapa update serta UI baru dari versi Android sebelumnya. Mulai terdapat “widget” yang dapat dibesar kecilkan. Kemudian ditambah kemampuan untuk meng-upload video dan gambar ke Youtube dan Picasa. (Akhmad Dharma Kasiman ; 2015 : 5)

5. Android 1.6 Donut

Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih *userfriendly*. Pada versi ini juga sudah mendukung teknologi CDMA/EVDO, 802.1x, VPNs. Kemudian support layar dengan resolusi WVGA. (Akhmad Dharma Kasman ; 2015 : 5)

6. Android 2.0/2.1 Eclair

Dirilis pertama kali 9 Desember 2009. Terjadi penambahan fitur untuk mengoptimalkan software, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk camera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Beberapa versi updatenya antara Android v2.0 kemudian v2.0.2 dan terakhir v2.1. (Akhmad Dharma Kasman ; 2015 : 5)

7. Android 2.2 Froyo (Froze Yoghurt)

Dirilis pertama kali pada 20 Mei 2010 pada smartphone Google Nexus One. Pada versi ini sudah support terhadap Adobe Flash Player 10.1. Peningkatan pada kecepatan membuka dan menutup aplikasi serta penggunaan SD-Card sebagai tempat penyimpanan aplikasi. Ketika Android Froyo hadir mulai muncul banyak diskusi membahas mengenai persaingan antara Android dengan iOS yang semakin

ketat dimasa yang akan datang. Beberapa update yang dirilis antara lain Android v.2.2.1. hingga v.2.2.3. (Akhmad Dharma Kasman ; 2015 : 5)

8. Android 2.3 Gingerbread

Petama kali diperkenalkan 6 Desember 2010. Terjadi banyak peningkatan pada versi Android yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalkan kemampuan aplikasi dan game. Serta mulai digunakan Near Field Communication (NFC). Perbaikan terhadap dukungan layar resolusi WXGA dan di atasnya. Beberapa versi update yang dirilis antara lain v.2.3.3 hingga v.2.3.7. Sampai saat ini Android Gingerbread merupakan versi Android yang memiliki penggunaan terbanyak dibandingkan versi Android lainnya, yaitu mencapai 65% dari seluruh versi Android yang dirilis. (Akhmad Dharma Kasman ; 2015 : 6)

9. Android 3.0/3.1 Honeycomb

Pertama kali diperkenalkan pada 22 Februari 2011 dan Motorola Xoom adalah yang pertama kali menggunakannya. Android ini merupakan OS yang didesain khusus untuk mengoptimalkan penggunaan pada tablet PC. (Akhmad Dharma Kasman ; 2015 : 6)

10. Android 4.0 ICS (Ice Cream Sandwich)

Pertama kali dirilis pada 19 Oktober 2011. Smartphone yang pertama kali menggunakan Android ini adalah Samsung Nexus. Secara teori semua perangkat seluler yang menggunakan versi Android sebelumnya, Gingerbread, dapat di-update ke Android Ice Cream Sandwich. (Akhmad Dharma Kasman ; 2015 : 6)

11. Android versi 4.1 (Jelly Bean)

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali pada produk tablet Asus, yakni Google Nexus 7. (Akhmad Dharma Kasman ; 2015 : 6)

12. Android versi 4.2 (Jelly Bean)

Fitur photo sphere untuk panorama, daydream sebagai screensaver, power control, lock screen widget, menjalankan banyak user (dalam tablet saja), widget terbaru. Android 4.2 pertama kali dikenalkan melalui LG Google Nexus 4. (Akhmad Dharma Kasman ; 2015 : 7)

13. Android 4.4 KitKat

Android KitKat dirilis pada tanggal 31 Oktober 2013. KitKat merupakan merk coklat yang dikeluarkan oleh Nestle. Android KitKat juga disebutkan lebih bersahabat untuk smartphone atau tablet bertipe low-end karena diklaim mampu berjalan lancar dengan memory RAM 512 MB. Hal ini dapat tercapai karena peningkatan memory manajemen dan optimasi di kernel, system, framework dan aplikasi. (Akhmad Dharma Kasman ; 2015 : 7)

14. Android 5.0 Lollipop

Android 5.0 Lollipop dirilis pada tanggal 15 Oktober 2013. Lollipop merupakan sebuah permen manis dalam stick yang biasanya terbentuk lingkaran atau bulat. Pada perilisan versi terbarunya, OS Android versi 5.0 Lollipop dibekalkan dengan sektor tampilan yang lebih berwarna dan responsive. Selain itu, pihak Android juga memberikan jaminan pada pengguna Android jika navigasi dari OS 5.0 Lollipop lebih mudah dan tidak akan menyulitkan pengguna. (Akhmad Dharma Kasman ; 2015 : 7)

II.5.3. ADT (Android Development Tools)

Android Development Tools adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi *project* android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui eclipse. Dengan ADT juga kita dapat melakukan pembuatan package android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang. (Nazruddin Safaat H ; 2012 : 6)

II.5.4. Android SDK(SoftwareDevelopment Kit)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android.

Android berjalan di dalam *Dalvik Virtual Machine* (DVM). Android SDK sebagai alat bantu dan API untuk memulai mengembangkan Aplikasi pada *Platform* Android menggunakan bahasa pemrograman Java. Untuk versi *operating system* terbaru tersedia pada SDK dan dapat diupdate, *Source* Android dapat diunduh langsung pada situs resmi pengembang SDK Android. (Nazruddin Safaat H ; 2015 : 169)

II.5.5. Android Virtual Device (AVD)

Android Virtual Device adalah sebuah emulator Android, berfungsi sebagai emulator untuk mencoba aplikasi apakah berjalan dengan baik atau tidak sebelum dijalankan di Smartphone dan Tablet Android. Android Virtual Devices (AVD) adalah konfigurasi dari emulator sehingga kita dapat menjalankan perangkat Android sesuai model yang dipilih, misal Android 1.5, 2.2 atau 2.3 dan versi Android Selanjutnya. (Mohamad Nur Fadzri Lamusu, et al ; 2013 : 6)

II.5.6. Eclipse IDE

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in seperti penambahan plugin ADT untuk pengembangan Android dan aplikasi Android.

Jenis-jenis komponen pada yang digunakan pada aplikasi Android yaitu :

a) *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity*, misalnya click tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah *activity* dinyatakan dengan *method Activity.SetContentView()*. *ContentView* adalah objek yang berada pada root hirarki.

b) *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya *media player* sedang memutar lagu dari *list* yang ada, Aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan user untuk memilih lagu misalnya, atau menulis sms sambil *player* sedang berjalan. Untuk menjaga musik tetap di jalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada thread utama dari proses aplikasi.

c) *Broadcast Receiver*

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti notifikasi zona waktu

berubah, baterai *low*, gambar telah selesai diambil oleh camera, atau perubahan referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi broadcast misalnya memberikan informasi pada aplikasi lain bahwa ada data telah diunduh ke perangkat dan siap untuk digunakan.

Broadcast Receiver tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

d) *Content Provider*

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite. *Content Provider* menyediakan cara untuk mengakses data dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (Map), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka di sinilah fungsi *content provider*. (Nazruddin Safaat H ; 2015 : 9-10)

II.5.7. Java

Bahasa pemrograman java dikembangkan oleh Sun Microsystem yang dimulai oleh James Gosling dan dirilis pada tahun 1995. Saat ini Sun Microsystem telah diakui oleh Oracle Corporation. Java bersifat Write Once, Run Anywhere (program yang ditulis satu kali dan dapat berjalan pada banyak platform. (Jubilee Enterprise ; 2015 : 1)

II.5.8. Sejarah Java

Java sejak awal sudah didesain sebagai bahasa pemrograman yang bisa berjalan di berbagai komputer termasuk telepon genggam. Bahasa ini dikembangkan pertama kali oleh James Gosling saat masih bergabung di Sun Microsystems, saat ini Sun sudah diakuisisi oleh Oracle sehingga menjadi bagian dari Oracle. Bahasa Java awal dirilis tahun 1995. Sintaksnya banyak meniru sintaks yang terdapat pada C dan C++, hanya saja model objek dibuat lebih sederhana serta dukungan rutin-rutin level bawah yang minimal.

Versi awal Java pada tahun 1996 sudah merupakan versi rilis dan bukan versi beta, sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya. Paket-paket yang ada antara lain :

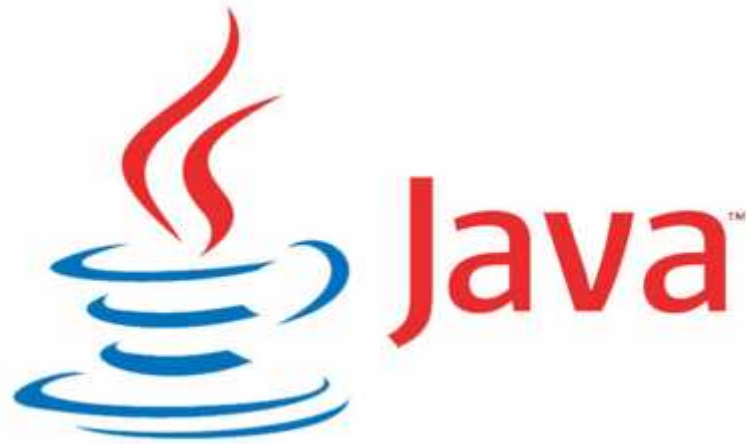
- a. java.lang : Peruntukan kelas elemen-elemen dasar.
- b. Java.io : Peruntukan kelas input dan output, termasuk penggunaan berkas.
- c. Java.util : Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanggalan.
- d. Java.net : Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan computer lain menggunakan jaringan TCP/IP.
- e. Java.awt : Kelas dasar aplikasi antarmuka dengan penggunaan (GUI).
- f. Java.applet : Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Bahasa pemrograman java dibuat dengan memperhatikan beberapa prinsip berikut:

1. Harus sederhana, berorientasi objek dan familier.
2. Kuat dan aman.
3. Tidak tergantung platform dan portable.
4. Bisa dieksekusi dengan performa tinggi.
5. Bisa diinterpretasikan, threaded dan dinamis.

Aplikasi-aplikasi berbasis java nantinya akan dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai *Java Virtual Machine (JVM)*. Ditinjau dari segi penggunaannya, Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose).

Versi asli dan implementasi *reference* ke *compiler*, *Virtual Machine*, dan pustaka class dari java, awalnya dirilis oleh Sun di bawah lisensi *proprietary*. Tapi sejak bulan Mei 2007, agar sesuai dengan proses *open source Java*, maka Sun melisensikan ulang teknologi Java dengan teknologi *GNU General Public License*. Ada banyak pihak lain yang turut mengembangkan implementasi alternatif dari teknologi Sun Microsystem ini, termasuk *GNU Compiler for Java (compiler untuk bytecode)*, *GNU Classpath* (pustaka standar), dari *Iced Tea Web (Plugin browser untuk applet)*.



Gambar II.1. Logo Pemrograman Java

(Sumber :Tim EMS ; 2015 : 4)

Java sedari awal didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karenanya aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya “Write Once, Run Anywhere” atau diterjemahkan sebagai “Tulis sekali, jalankan di mana pun”. Artinya program java hanya perlu ditulissekali saja, dan bisa dijalankan sebagai platform system operasi seperti Linux, dan Windows tanpa harus menulis program ulang lagi. Sekarang, java telah menjadi bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Yang dianggap paling sukar di Java adalah menginstal infrasrtuktur pemrograman. Karena ada banyak hal yang perlu diinstal terlebih dahulu. Namun untungnya saat ini banyak tool yang langsung memudahkan bahasa pemrograman java, misalnya Eclipse dan Netbeans. Keduanya selain memudahkan pemrograman, juga menyediakan tool perancangan GUI yang lengkap. Tool yang

memudahkan pemrograman tersebut bernama IDE (Integrated Development Environment). Ini memudahkan pemrograman karena memudahkan Anda menulis program java. (Tim EMS ; 2015 : 2-5)

II.5.9. JRE

Java adalah bahasa pemrograman yang dirancang tidak tergantung pada platform alias platform independent, sehingga program yang ditulis di bahasa pemrograman java idealnya harus bisa dijalankan di semua sitem operasi. Karena itu ada JRE atau runtime environment java yang tersedia untuk banyak sistem operasi, sehingga JRE ini memberikan ruang hidup bagi program java. Kalau program java ibarat ikan, JRE ini ibarat aquariumnya. Ada aquarium untuk Windows, Linux, Mac os dan sebagainya. Dengan demikian, programmer tidak perlu bingung menjalankan program java, karena semua JRE sudah tersedia dan tinggal dieksekusi di sistem operasi yang bersangkutan. Jadi walaupun komputer Anda memakai Windows, Linux, Mac os, maka ini bisa menjalankan satu program yang sama. Bahkan teorinya, program yang dikembangkan di Windows bisa dijalankan pula di sistem operasi lain, seperti Linux. Ini karena adanya Java Virtual Machine untuk tiap versi sistem operasi yang di dukung oleh Java. Virtual Machine adalah program yang memproses kode Anda sehingga Anda perlu menginstal program ini di tiap platform Anda, sebelum Anda bisa menjalankan kode Java. Jadi kalau mau menjalankan java di Windows, ya harus menginstal JRE untuk Windows, Kalau mau menjalankan program java di Linux, harus menginstal JRE untuk Linux. (Tim EMS ; 2015 : 5)

II.5.10. JDK (Java Development Kit)

JDK atau Java Development Kit berguna sebagai platform atau run time yang diperlukan untuk membuat program dengan bahasa Java. Termasuk di dalam JDK ini adalah library-library yang diperlukan untuk membuat program java. (Tim EMS ; 2015 : 13)

II.5.11. UML (Unified Modeling Language)

Unified Modeling Language adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. Pengertian lain dari UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OOP (*Object-Oriented Programming*). UML juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*.


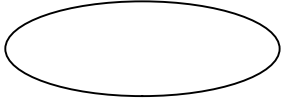
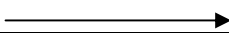
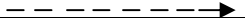
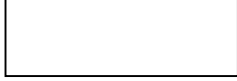
Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh Booch untuk membuat sebuah project pendekatan metode yang uniform atau seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan draft metoda UML version 0,8 dan diselesaikan serta di realese pada bulan oktober 1995.

1. Diagram Use Case

Diagram Use Case menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. Menurut Whitten, (2004) usecase diagram adalah diagram yang menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.

Adapun simbol-simbol yang digunakan pada saat pembuatan *use case diagram* diperlihatkan pada Table II.1 berikut :

Tabel II.1. Simbol UseCase Diagram

Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use case</i> : Peringkat tertinggi dari fungsionalitas yang dimiliki sistem.
	<i>Association</i> : adalah relasi antara <i>actor</i> dan <i>use case</i> .
	<i>Generalisasi</i> : untuk memperlihatkan struktur pewaris yang terjadi.
	<i>Process</i> : untuk menjalankan suatu sistem pada seorang <i>actor</i> / <i>user</i> .

(Sumber :Verdi Yasin ; 2012 : 270)

2. Diagram Class

Diagram class memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram class bersifat statis; menggambarkan hubungan apa yang terjadi jika mereka berhubungan. Diagram class mempunyai 3 macam *relationships* (hubungan), sebagai berikut :

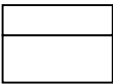
Association suatu hubungan antara bagian dari dua kelas. Terjadi association antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Didalam diagram, sebuah association adalah penghubung yang menghubungkan dua kelas.

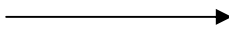
Aggregation suatu association dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. Aggregation memiliki titik pusat yang mencakup keseluruhan bagian. Sebagai contoh : Order detail merupakan kumpulan dari order.

Generalization suatu hubungan turunan dengan mengansumsikan suatu kelas merupakan suatu superclass (kelas super) dari kelas lain. Generalization memiliki tingkatan yang berpusat pada superclass. Contoh : payment adalah superclass dari cash, check, dan kredit.

Adapun simbol-simbol yang digunakan pada saat pembuatan *use case diagram* diperlihatkan pada Table II.2 berikut :

Tabel II.2. Simbol Class Diagram

Simbol	Keterangan
	Class : Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
1..*	<i>One-to-many</i> : Peringkat tertinggi dari fungsionalitas yang dimiliki sistem.

	<i>Association</i> : adalah relasi antara <i>actor</i> Dan <i>use case</i> .
---	--

3. Diagram sequence

Diagram class dan diagram object merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti diagram interaction. Diagram sequence merupakan salah satu diagram interaction yang menjelaskan bagaimana suatu operasi itu dilakukan ; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalanya operasi diurutkan dari kiri ke kanan bersasarkan waktu terjadinya dalam pesan yang terurut.

Adapun simbol-simbol yang digunakan pada saat pembuatan *use case diagram* diperlihatkan pada Table II.3 berikut :

Tabel II.3 Simbol Sequence Diagram

	<ul style="list-style-type: none"> • orang, animals, atau sistem lain yang berinteraksi dengan sistem, manusia dan perangkat lunak juga system. • Memiliki nama dan beberapa atribut. • Tidak memiliki nama kelas atau diagram. • Tidak memiliki nama objek.
	<ul style="list-style-type: none"> • Merupakan objek. • Tidak memiliki nama kelas atau diagram. • Memiliki nama dan beberapa atribut. • Tidak memiliki nama objek.
	<ul style="list-style-type: none"> • Menandakan kehidupan objek selama urutan. • diakhiri tanda X pada titik di mana kelas tidak lagi berinteraksi.
	<ul style="list-style-type: none"> • Tidak memiliki nama. • Adalah pesan panjang yang sempat panjang digambarkan di atas sebuah garis hidup. • Menandakan bahwa suatu objek menerima atau mengirim pesan.
	<ul style="list-style-type: none"> • objek mengirim satu pesan ke objek lainnya
	<ul style="list-style-type: none"> • menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
	<ul style="list-style-type: none"> • menyatakan bahwa suatu objek menggunakan masukan ke objek lainnya arah panah mengarah pada objek yang dikirim
	<ul style="list-style-type: none"> • objeknya mengembalikan suatu hasil ke objek tertentu, arah panah mengarah pada objek yang menerima hasil
	<ul style="list-style-type: none"> • menyatakan suatu objek mengembalikan hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.

4. Diagram Activity






Diagram Activity menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang

mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Activity diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi ditrigger oleh selesainya state sebelumnya (internal processing) oleh karena itu activity diagram tidak menggambarkan behavior internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana actor menggunakan sistem untuk melakukan aktivitas. (Nazruddin Safaat H ; 2015 : 177-181)

Adapun simbol-simbol yang digunakan pada saat pembuatan *activity diagram* diperlihatkan pada Table II.4 berikut :

Tabel II.4 Simbol Activity Diagram

Simbol	Keterangan
	Titik awal
	Titik akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Line Connector; untuk menghubungkan satu simbol dengan simbol lainnya

(Sumber :Verdi Yasin ; 2012 : 271)