BAB II

TINJAUAN PUSTAKA

II.1. Steganografi

Steganography adalah seni dan ilmu menulis atau menyembunyikan pesan tersembunyi dengan suatu cara sehingga si pengirim dan si penerima, tidak ada seorang pun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Kata *steganography* berasal dari bahasa Yunani *Steganos* yang artinya " Tersembunyi atau Terselubung ", dan *Graphein,* "Menulis", di ambil dari nama objek atau karya oleh Trithemus (1462-1516) yang berjudul " *Steganographia* ". *Steganography* adalah seni dan ilmu menulis atau menyembunyikan pesan tersembunyi dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Dari contoh-contoh *steganography* konvensial tersebut dapat dilihat bahwa semua teknik *steganography* konvensional berusaha merahasiakan komunikasi dengan cara menyembunyikan pesan ataupun mengkamuflase pesan. Maka sesungguhnya prinsip dasar dalam *steganography* lebih dikonsentrasikan pada kerahasiaan komunikasinya bukan pada datanya.

Tujuan dari *steganography* adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya kebanyakan diselesaikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari pihak ketiga,sebagai contoh sebuah gambar yang terlihat tidak mengandung suatu hal yang bersifat rahasia dan berbahaya. Perubahan ini bergantung pada kunci (sama

pada *cryptography*) dan pesan untuk disembunyikan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan. Pada umumnya, pesan *steganography* muncul dengan rupa lain seperti gambar,artikel,daftar belanjaan, atau pesan-pesan lainnya. Pesan yang tertulis ini merupakan tulisan yang menyelubungi atau menutupi. Contohnya suatu pesan bisa disembunyikan dengan menggunakan tinta yang tidak terlihat diantara garis-garis yang kelihatan. Teknik *steganography* meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau *image)* di dalam *file-file* lain mengandung teks, *image* bahkan audio dan video tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari *file* semula. Metode lain termasuk tinta yang tidak Nampak, *microdots,* pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi *spectrum*. Pada metode *steganography* cara ini sangat berguna jika digunakan pada cara *steganography computer* karena banyak format *file* digital yang dapat dijadikan media untuk menyembunyikan pesan. Format yang bisa digunakan diantaranya :

1. Format *image* : bitmap (BMP),jpg,gif,png,tiff,pcx.
2. Format audio : wav,voc, mp3
3. Format lain : teks *file*,html,pdf,video
4. Kelebihan *steganography* daripada *cryptography* adalah pesan-pesannya tidak menarik perhatian dan kecurigaan pihak ketiga. Pesan-pesan dengan karakter kode yang acak dan sulit dipahami dalam *cryptography* yang tidak disembunyikan, walaupun tidak dapat dipecahkan, akan menimbulkan kecurigaan. Seringkali, *steganography* dan *cryptography* digunakan secara

bersamaan untuk menjamin keamanan pesan.(Nugraha dan Suarna ; 2013 : 59-70)

II.2. Citra

Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi.

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagai dari berkas cahaya tersebut.Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya.Sehingga bayangan objek yang disebut citra tersebut terakam.(Permadi dan Murinto ; 2015 : 1028-1038)

II.3. *Least Significant Bit* (LSB)

Least significant bit adalah bagian dari barisan data *biner* (basis dua) yang mempunyai nilai paling tidak berarti/paling kecil. Letaknya adalah paling kanan dari barisan *bit*. Sedangkan *most significant bit* adalah sebaliknya, yaitu angka yang paling berarti/paling besar dan letaknya disebelah paling kiri.

Contohnya adalah bilangan biner dari 255 adalah 11111111 (kadang-kadang diberi huruf b pada akhir bilangan menjadi 11111111b). Bilangan tersebut dapat berarti :

$$1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Gambar II.1. Bilangan biner
(Sumber : Tri Prasetyo Utomo ; 2012)

Dari barisan angka 1 di atas, angka 1 paling kanan bernilai 1, dan itu adalah yang paling kecil. Bagian tersebut disebut dengan *least significant bit* (*bit* yang paling tidak berarti), sedangkan bagian paling kiri bernilai 128 dan disebut dengan *most significant bit* (*bit* yang paling berarti).

Least significant bit sering kali digunakan untuk kepentingan penyisipan data ke dalam suatu media digital lain, salah satu yang memanfaatkan *Least significant bit* sebagai metode penyembunyian adalah steganografi *audio* dan gambar.

Metode yang digunakan untuk penyembunyian pesan rahasia pada aplikasi ini adalah dengan cara menyisipkan pesan ke dalam *bit* rendah (*least significant bit*) pada data *pixel* yang menyusun file gambar BMP 24 *bit* tersebut.

Pada file gambar BMP 24 *bit* setiap *pixel* pada gambar terdiri dari susunan tiga warna yaitu merah, hijau, biru (RGB) yang masing-masing disusun oleh bilangan 8 *bit* (1 *byte*) dari 0 sampai 255 atau dengan *format biner* 00000000 sampai 11111111. Sebagai contoh *file* gambar BMP 24 *bit* dengan warna merah murni dalam *format biner* akan terlihat sebagai berikut :

00000000 00000000 11111111

00000000 00000000 11111111

Sedangkan untuk warna hijau murni dalam *format biner* akan terlihat sebagai berikut :

00000000 11111111 00000000

00000000 11111111 00000000

Sedangkan untuk warna biru murni dalam *format biner* akan terlihat sebagai berikut :

11111111 00000000 00000000

11111111 00000000 00000000

Dari uraian di atas dapat dilihat bahwa informasi dari warna biru berada pada *bit* pertama sampai *bit* delapan, dan informasi warna hijau berada pada *bit* sembilan sampai dengan *bit* 16, sedangkan informasi warna merah berada pada *bit* 17 sampai dengan *bit* 24.

Metode penyisipan LSB (*least significant bit*) ini adalah menyisipi pesan dengan cara mengganti *bit* ke 8, 16 dan 24 pada representasi *biner file* gambar dengan representasi *biner* pesan rahasia yang akan disembunyikan. Dengan demikian pada setiap *pixel file* gambar BMP 24 *bit* dapat disisipkan 3 *bit* pesan, misalnya terdapat data *raster original file* gambar adalah sebagai berikut :

00100111 11101001 11001000

00100111 11001000 11101001

11001000 00100111 11101001

Sedangkan representasi *biner* huruf A adalah 01000001, dengan menyisipkannya ke dalam *pixel* di atas maka akan dihasilkan :

00100111 11101001 11001000

00100111 11001000 11101000

11001000 00100111 11101001

Terlihat pada *bit* kedelapan, enam belas dan 24 diganti dengan representasi biner huruf A, dan hanya tiga *bit* rendah yang berubah (cetak tebal), untuk penglihatan mata manusia sangatlah mustahil untuk dapat membedakan warna pada *file* gambar yang sudah diisi pesan rahasia jika dibandingkan dengan *file* gambar asli sebelum disisipi dengan pesan rahasia. (Tri Prasetyo Utomo ; 2012)

II.4. Metode LSB+1

Metode LSB+1 adalah Proses penyisipan pesan ke dalam citra digital dengan menggunakan metode LSB hampir sama saja dengan metode LSB, bedanya ada pada bit tempat penyisipan pesan. Jika pada metode LSB, pesan disisipkan pada bit LSB (bit ke-8), maka pada metode LSB+1, pesan disisipkan pada bit ke-7. Sebagai contoh, misalkan tiga piksel yang berdekatan (sembilan bytes) dengan kode RGB berikut :

00110101 11010110 11101010

11110100 00111001 11100001

01110001 10010001 11100001

Pesan yang akan disisipkan adalah karakter “R”, yang nilai binernya adalah “01010010”. Pesan akan disisipkan dengan menggunakan metode LSB+1, maka akan dihasilkan citra hasil dengan urutan bit sebagai berikut:

00110101 11010110 11101000

11110110 00111001 11100001

01110011 10010001 11100001

Pada contoh di atas, dapat dilihat bahwa sebagian bit LSB+1 (bit ke-7) yang ada pada citra asal (original) digantikan dengan bit dari pesan yang akan disisipkan. (Andrian Yudhi ; 2013)

II.5. Metode LSB+2

Pada metode LSB+2, pesan disisipkan pada bit ke-6. Sebagai contoh, misalkan tiga piksel yang berdekatan dengan kode RGB seperti pada metode LSB+1 akan disisipkan adalah karakter "R" dengan menggunakan metode LSB+2, maka akan dihasilkan citra hasil dengan urutan bit sebagai berikut :

00110001 11010110 11101010

11110100 00111001 11100001

01110101 10010001 11100001

Pada contoh di atas, dapat dilihat bahwa sebagian bit LSB+2 (bit ke-6) yang ada pada citra asal (original) digantikan dengan bit dari pesan yang akan disisipkan. (Andrian Yudhi ; 2013)

II.6. Penyisipan Pesan

Untuk melakukan penyisipan pesan baik itu pada pesan teks, gambar, suara dan *video* dibutuhkan masukan berupa *file digital* yang akan disisipkan pesan, pesan yang akan disisipkan (*message*), dan kunci (*key*). Beberapa contoh media penyisipan pesan rahasia yang digunakan dalam teknik *Steganography* antara lain adalah:

a. Teks

Dalam algoritma *Steganography* yang menggunakan teks sebagai media penyisipannya biasanya digunakan teknik NLP (*Natural Language Processing*) sehingga teks yang telah disisipi pesan rahasia tidak akan mencurigakan untuk orang yang melihatnya.

b. Gambar

Format gambar paling sering digunakan, karena *format* ini merupakan salah satu *format file* yang sering dipertukarkan dalam dunia *internet*. Alasan lainnya adalah banyaknya tersedia algoritma *Steganography* untuk media penampung yang berupa citra.

c. Suara

Format suara sering dipilih karena biasanya berkas dengan format ini berukuran relatif besar. Sehingga dapat menampung pesan rahasia dalam jumlah yang besar pula.

d. Video

Format *video* memang merupakan format dengan ukuran *file* yang relatif sangat besar namun jarang digunakan karena ukurannya yang terlalu besar sehingga mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

Steganografi berfungsi untuk menyembunyikan keberadaan pesan dan dapat dianggap sebagai pelengkap dari kriptografi yang bertujuan untuk menyembunyikan isi pesan. Berbeda dengan kriptografi dalam Steganografi pesan disembunyikan sedemikian rupa sehingga pihak lain tidak dapat mengetahui

adanya pesan rahasia. Pesan rahasia tidak diubah menjadi karakter ‘aneh’ seperti halnya kriptografi. Pesan tersebut hanya disembunyikan ke dalam suatu media berupa gambar, teks, musik, atau media *digital* lainnya dan terlihat seperti pesan biasa.

Untuk memudahkan dalam proses penyembunyian pesan teks ke dalam image, maka dirancang suatu aplikasi Steganografi untuk penyisipan pesan. Aplikasi dirancang dengan tiga proses yaitu mengambil *image*, menambahkan pesan ke dalam *image* (*encode image*) dan menampilkan pesan rahasia (*extract*) dalam *image*. (Siti Rohayah, et al ; 2015 : 975-981)

II.7. Pengertian Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, *game*, pelayanan masyarakat, periklanan dan hampir semua proses kegiatan.

Yang dimaksud perangkat lunak aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. (Siti Rohayah, et al ; 2015 : 975-981)

II.8. Pengertian Perancangan

Perancangan merupakan desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan, tahap ini menyangkut mengkon-figurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analasis sistem. (Mardi Iwan Gunawan Saragih ; 2014 : 99-103)

II.9. Android

Menurut Arifianto Teguh, android adalah sebuah *platform* pertama yang betul-betul terbuka dalam pengembangannya dan komperehensif untuk perangkat *mobile*, semua perangkat lunak yang ada difungsikan menjalankan sebuah *device mobile* tanpa memikirkan kendala kepemilikan yang menghambat inovasi pada teknologi *mobile*. Dalam definisi lain, Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware*, dan aplikasi inti yang dirilis oleh Google. Sedangkan Android *SDK (Software Development Kit)* menyediakan *tools* dan API yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java.

Aplikasi *Android* ditulis dalam bahasa pemrograman *java*, yaitu kode *java* yang terkompilasi bersama-sama dengan data dan *file-file* sumber yang dibutuhkan oleh aplikasi yang digabungkan oleh *app tools* menjadi paket aplikasi Android, sebuah file yang ditandai dengan akhiran *.apk*. file inilah yang didistribusikan sebagai aplikasi dan diinstal pada *handset Android*. File ini diunduh oleh pengguna ke perangkat *mobile* mereka. Semua kode dijadikan satu

file *.apk,* dan kemudian kita sebut sebagai sebuah aplikasi.(Ahmad : 2015 : 190-200)

II.10. 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 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, test perangkat lunak, pengembangan *web* dan lain sebagainya.(Murtiwiyati dan Lauren ; 2013 : 1-10)

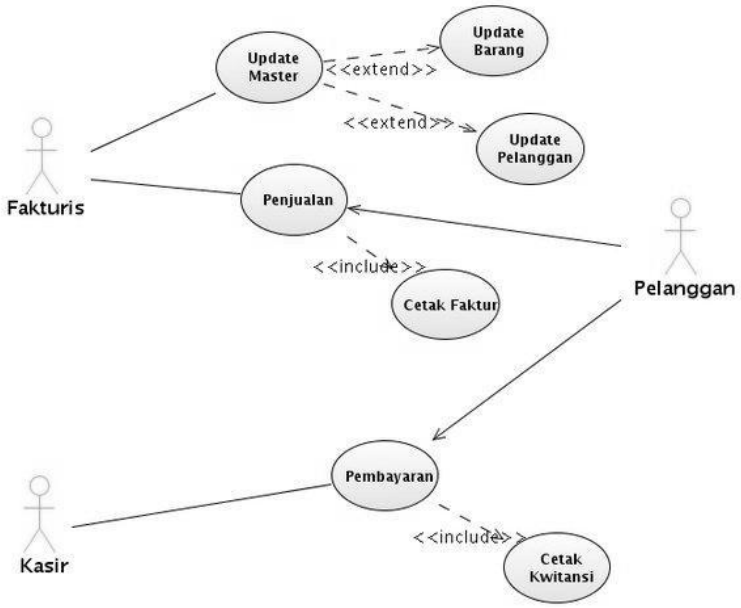
II.11. Pengertian UML

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 pemodalan 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.11.1. *Use Case Diagram*

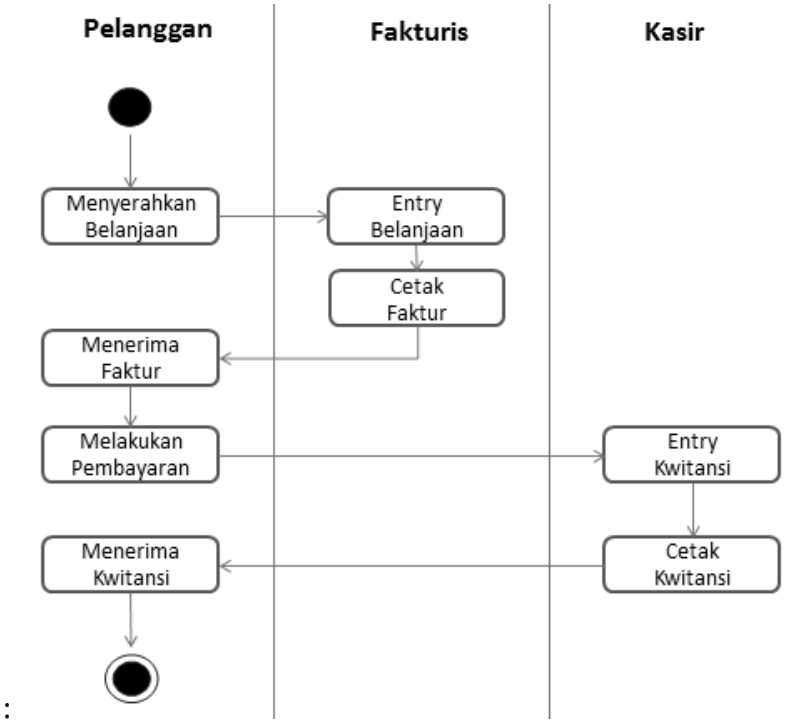
Use case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada didalam 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.2.



Gambar. II.2. *Use Case Diagram*
(Sumber : Windu dan Grace ; 2013 : 83)

II.11.2. *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.3.

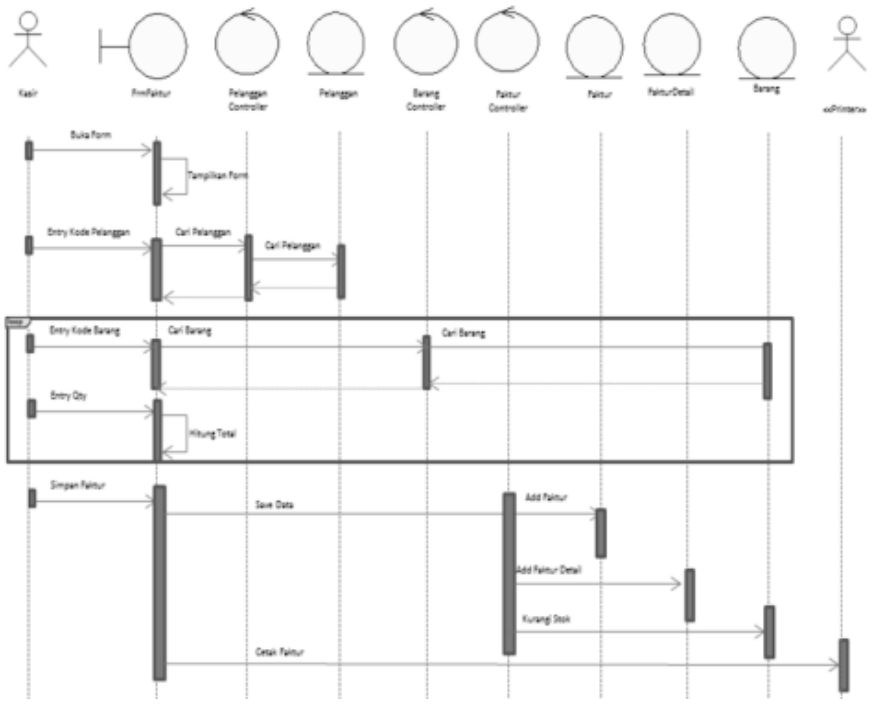


:

Gambar. II.3. *Activity Diagram*
(Sumber : Windu dan Grace ; 2013 : 83)

II.11.3. *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.4.

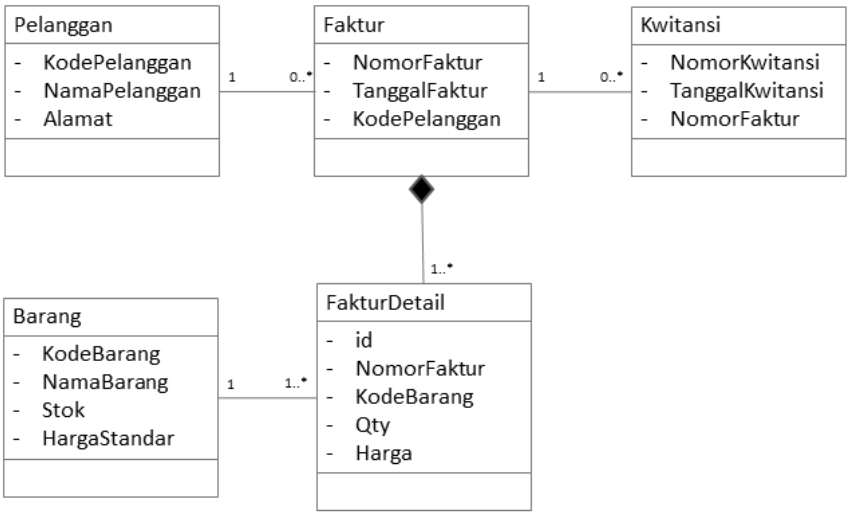


Gambar. II.4. *Sequence Diagram*
(Sumber : Windu dan Grace ; 2013 : 84)

II.11.4. *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 menunjukan 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.5. berikut :



Gambar. II.5. *Class Diagram*
(Sumber : Windu dan Grace ; 2013 : 83)