

BAB II

TINJAUAN PUSTAKA

I. Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu subkelas 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. Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, yaitu dari kata *crypto* dan *graphia* yang berarti penulisan rahasia. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut kriptologi (*cryptology*). Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah. Perancang algoritma kriptografi disebut kriptografer. (Emy Setyaningsih; 2015: 2).

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: “*cryptós*” artinya “*secret*” (rahasia), sedangkan “*gráphein*” artinya “*writing*” (tulisan). Jadi, kriptografi berarti “*secret writing*” (tulisan rahasia). Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi (Mariana; 2011: 2).

Kriptografi merupakan seni dan ilmu menyembunyikan informasi dari penerima yang tidak berhak. Kata kriptografi berasal dari kata Yunani *kryptos* (tersembunyi) dan *graphein* (menulis). Dalam teknologi informasi telah dan sedang dikembangkan cara-cara untuk menangkal berbagai serangan, seperti penyadap dan perubahan data yang sedang dikirimkan. Transformasi ini memberikan solusi pada dua macam masalah keamanan data, yaitu masalah privasi (*privacy*) dan keotentikan (*authentication*). Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya.

a. Enkripsi

Proses enkripsi adalah proses penyandian pesan terbuka (*plaintext*) menjadi pesan rahasia (*ciphertext*). *Ciphertext* inilah yang nantinya akan dikirimkan melalui saluran komunikasi terbuka. Pada saat *ciphertext* diterima oleh penerima pesan, maka pesan rahasia tersebut diubah lagi menjadi pesan terbuka melalui proses dekripsi sehingga pesan tadi dapat dibaca kembali oleh penerima pesan.

b. Dekripsi

Dekripsi merupakan proses kebalikan dari proses enkripsi, merubah *ciphertext* kembali ke dalam bentuk *plaintext*. Untuk menghilangkan penyandian yang diberikan pada saat proses enkripsi, membutuhkan penggunaan sejumlah informasi rahasia, yang disebut sebagai kunci (Fricles Ariwisanto Sianturi; 2013: 43).

III. File Text dan Gambar (Citra)

File text adalah arsip yang disimpan dalam suatu media corage berupa data yang terdiri dari karakter yang menyatukan kata-kata / symbol. File text terpakai sebagai penyimpanan yang memiliki sifat dan tidak memiliki organisasi data yang jelas melakukan proses yang kompleks untuk melakukan pengelompokan data.

Binary adalah yang berisi data-dat 8 byte, biasanya file ini disimpan untuk file sehingga tidak bias dibaca oleh manusia tanpa perangkat bantuan dan dapat dimengerti oleh mesin dipakai penyimpanan file yang bersifat terstruktur dilengkapi dengan index dan pointer sebagai acuan data. (Suprianto :2012;105)

Citra adalah sebuah representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal *video* seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra digital sebagai fungsi intensitas cahaya dua-dimensi $f(x,y)$ di mana x dan y menunjukkan koordinat spasial, dan nilai f pada suatu titik (x,y) sebanding dengan *brightness (gray level)* dari citra di titik tersebut.

Secara umum pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai *real* maupun kompleks yang direpresentasikan dengan deretan *bit* tertentu. Suatu citra dapat didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y koordinat spasial, dan intensitas f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Nilai pada suatu irisan dan antara baris dan kolom (pada posisi x,y) disebut dengan *picture elements, image elements, pels*, atau *pixel*. istilah *pixel* sering digunakan pada citra digital. Setiap *pixel* tidak hanya satu titik dalam sebuah citra merupakan sebuah bagian berupa kotak yang merupakan bagian terkecil (sel). Nilai dari sebuah *pixel* haruslah dapat menunjukkan nilai rata-rata yang sama untuk seluruh bagian dari sel tersebut.

1. *Pixel* Setiap *pixel* mewakili tidak hanya satu titik dalam sebuah citra melainkan sebuah bagian berupa kotak yang merupakan bagian terkecil (sel). Nilai dari sebuah *pixel* haruslah dapat menunjukkan nilai rata-rata yang sama untuk seluruh bagian dari sel tersebut.

2. Resolusi Citra Resolusi citra merupakan tingkat detail suatu citra. Semakin tinggi resolusi citra maka akan semakin tinggi pula tingkat detail dari citra tersebut(Imam Juhari ; 2014:82).

Berikut ini adalah beberapa ekstensi gambar yang sering digunakan :

- a. JPG adalah jenis data yang dikembangkan oleh *Joint Photographic Experts Assemble (JPEG)* yang dijadikan standar untuk para fotografer profesional. **menggunakan kombinasi DCT dan pengkodean Huffman untuk mengkompresikan suatu file citra. JPEG adalah suatu algoritma kompresi yang bersifat lossy, (yang berarti kualitas citranya agak kurang bagus).**
- b. **GIF (Graphics Interchange Format) adalah sebuah format yang sering digunakan dalam dunia web maupun dalam dunia citra digital. Format ini sering digunakan karena ukurannya yang relatif kecil dan juga banyaknya software editor gambar yang telah mendukung citra ini. GIF berukuran kecil karena membatasi jumlah warnanya sebanyak 256 warna sehingga dapat menghemat ukuran berkas.**
- c. **BMP file format, atau kadang-kadang disebut dengan DIB (Device Independent bitmap) file format adalah format gambar yang digunakan untuk menyimpan gambar digital berupa bitmap. Biasanya terdapat di Microsoft dan OS/2. BMP merupakan file gambar yang tidak terkompresi. Dan karena itu ukuran file BMP biasanya besar. File berukuran besar tidak tepat untuk transfer di internet karena akan memenuhi bandwidth dan menjadi lambat.**

IV. Java

Bahasa pemrograman Java dikembangkan oleh Sun Microsystems yang dimulai oleh James Gosling dan dirilis pada tahun 1995. Saat ini Sun Microsystems telah diakuisisi oleh Oracle Corporation.

Java bersifat *Write Once, Run Anywhere* (program yang di tulis satu kali dan dapat berjalan pada banyak platform).

Berikut ini fitur-fitur Java:

- a. **Berorientasi Objek:** Dalam Java, semua adalah objek.
- b. **Bersifat Platform Independent:** Java di-compile dalam bit kode platform idependen dan bukan pada mesin platform spesifik seperti pada C dan C++.
- c. **Sederhana:** Java di desain untuk dapat dengan mudah di pelajari.
- d. **Aman:** Dengan fitur keamanan Java, anda dapat membuat sistem yang bebas virus dan *powerful*.
- e. **Bersifat Architectural-neutral:** Compiler Java membuat format file objek yang architectural-neutral, yang membuat kode yang decompile dapat di eksekusi pada berbagai prosesor yang memiliki sistem runtime Java.
- f. **Portabel:** Java bersifat portable karena adanya fitur platform independent dan architectural-neutral.
- g. **Kuat dan powerful:** Java mengeliminasi error dengan menjalankan pengecekan pada waktu compile dan runtime.
- h. **Multithreaded:** Dengan fitur Multithreaded Java, anda dapat membuat program yang dapat mengerjakan banyak tugas sekaligus.
- i. **Terinterpretasi:** Kode bit Java ditranslasi secara langsung pada instruksi mesin dan tidak di simpan.
- j. **Performa Tinggi:** Java memiliki performa yang tinggi karena menggunakan compiler langsung.
- k. **Terdistribusi:** Java di desain untuk lingkungan distribusi internet.

1. **Dinamis:** Java lebih dimanis dari C dan C++ karena Java di desain untuk beradaptasi dengan lingkungan pengembangan. (PT Elex Media Komputindo, Jakarta: 2014)

II.5. Base64

Algoritma Base64 merupakan algoritma yang menggunakan salah satu konsep algoritma enkripsi modern yaitu algoritma Block Cipher yang berupa operasi pada mode bit namun algoritma Base64 ini lebih mudah dalam pengimplementasiannya dari algoritma-algoritma yang lainnya. Base64 adalah metoda yang untuk melakukan encoding (penyandian) terhadap data binary menjadi format 6-bit character. Pada algoritma ini, rangkaian bit-bit plaintext dibagi menjadi blok-blok bit dengan panjang yang sama, biasanya 64 bit yang direpresentasikan dengan karakter ASCII. Base64 menggunakan karakter A – Z, a – z dan 0 – 9 untuk 62 nilai pertama, sedangkan 2 nilai terakhir digunakan symbol (+ dan /). Standar yang penulis gunakan adalah MIME (Multipurpose Internet Mail Extensions)/RFC 1521. RFC ini menegaskan sebuah standar untuk implementasi Base64 terhadap data binary dan melampirkan sebuah karakter padding “=” jika terdapat kekurangan pada byte. Dalam streaming base64, spesifikasi mengharuskan setiap baris menjadi paling banyak 76 basis-64 karakter. (Hayatun Nufus :2011:4)

II.6. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) yang merupakan metodologi kolaborasi antara metoda booch, OMT (*Object Modeling Technique*), serta OOSE (*Oriented Software Engineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP). (Adi Nugroho;2009;4)

UML (*Unified Modeling Language*) adalah sebuah ”bahasa” yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantic*. Notasi UML merupakan sekumpulan bentuk khusus untuk

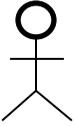
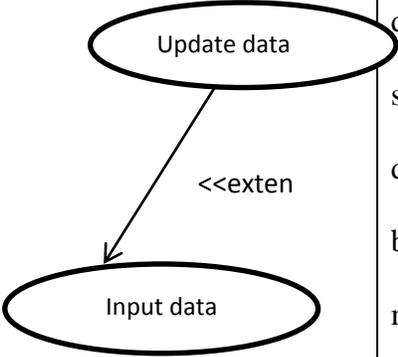
menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk – bentuk tersebut dapat dikombinasikan. *Unified Modeling Language* biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi – fungsi sistem secara umum, di buat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang di laksanakan secara umum, di buat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Yuni Sugiarti; 2013 :36)

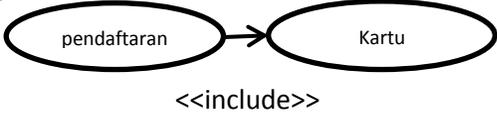
II.6.1. Use Case Diagram

Use case diagrams merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use case*, *actor* dan relasi. Berikut adalah sismbol – simbol yang ada pada diagram *use case*. (Yuni Sugiarti; 2013: 42)

Tabel II.1 Simbol – simbol pada Use Case Diagram

Simbol	Deskripsi
<p>Use case</p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit atau <i>actor</i>; biasanya ditanyakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p>Aktor</p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i>.</p>
<p>Asosiasi/ <i>association</i></p> 	<p>Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.</p>
<p>Extend</p> 	<p>Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjuk pada use</p>



<<extend>>	case yang dituju. Contoh :
<p data-bbox="185 273 289 302">Include</p> <p data-bbox="185 382 357 411"><<include>></p> 	<p data-bbox="568 273 1242 814">Relasi use case tambahan kesebuah use case dimana use case yang yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, <i>include</i> berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, contoh :</p> 

Sumber: (Yuni Sugiarti; 2013)

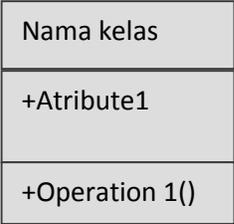
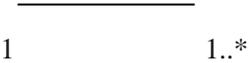
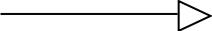
II.6.2. Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefenisian kelas – kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang di sebut atribut dan metode atau operasi.

1. Atribut merupakan variabel- variabel yang di miliki oleh suatu kelas.
2. Atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
3. Operasi atau metode adalah fungsi – fungsi yang di miliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis – jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan – batasan yang terdapat dalam hubungan – hubungan objek tersebut.(Yuni Sugiarti; 2013: 57)

Tabel II.2 Simbol – simbol Class Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka / interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu di gunakan oleh kelas yang lain, asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus).
Kebergantungan / defedency 	Relasi antar kelas dengan makna kebergantungan antar kelas

Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)
-----------------------------------------------------------------------------------------------	--------------------------------------------------------------------

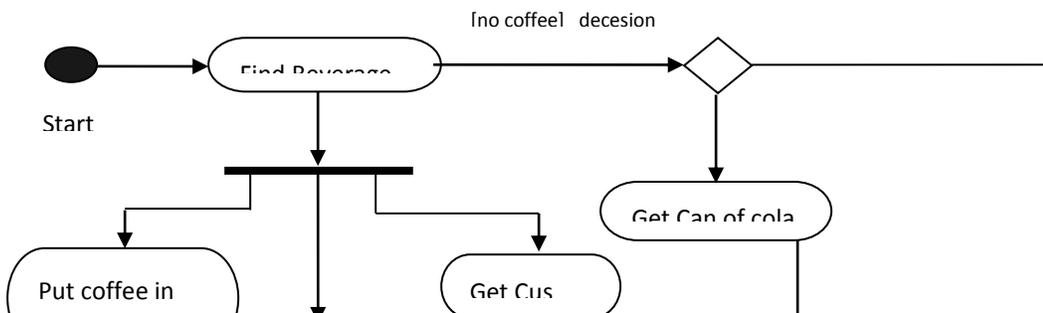
Sumber : (Yuni Sugiarti ; 2013)

II.6.3. Activity Diagram

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis.

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan behaviour 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 aktor menggunakan sistem untuk melakukan aktivitas.(Yuni Sugiarti; 2013: 75)



Gambar II.2. Activity Diagram

Sumber : (Yuni Sugiarti ; 2013)

II.6.4. Sequence Diagram

Diagram sequence menggambarkan kelakuan/ pelaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek – objek yang terlibat dalam sebuah use case beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

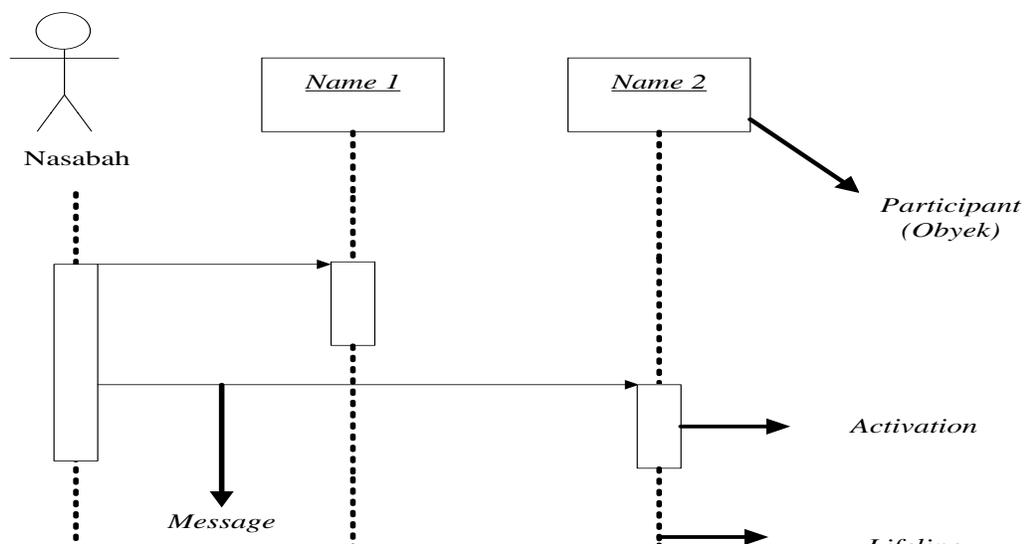
Diagram sequence memiliki ciri yang berbeda dengan diagram interaksi pada diagram kolaborasi sebagai berikut :

1. Pada diagram sequence terdapat garis hidup objek. Garis hidup objek adalah garis vertical yang mencerminkan eksistensi sebuah objek sepanjang periode waktu. Sebagian besar objek – objek yang tercakup dalam diagram interaksi akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek – objek itu diletakkan dibagian atas diagram

dengan garis hidup tergambar dari atas hingga bagian bawah diagram. Suatu objek lain dapat saja diciptakan, dalam hal ini garis hidup dimulai saat pesan *destroy*,

2. jika kasus ini terjadi, maka garis hidupnya juga berakhir.
3. Terdapat focus kendali (*Focus Of Control*), berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi suatu objek secara langsung atau sepanjang sub ordinat. Puncak dari empat persegi panjang adalah permulaan aksi, bagian dasar adalah akhir dari suatu aksi. Pada diagram ini mungkin juga memperhatikan penyaringan (*nesting*) dan *focus* kendali yang disebabkan oleh proses rekursif dengan menumpuk *focus* kendali yang lain pada induknya. (Yuni Sugiarti; 2013: 70)

Berikut simbol – simbol yang ada pada sequence diagram.



Gambar II.2 Simbol Sequence

Sumber : (Yuni Sugiarti ; 2013)