

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Penelitian lain juga dilakukan oleh Gratia Vintana dan Mardi Hardjianto pada tahun 2016 dengan judul “*SecurityChatting* Berbasis *Desktop* dengan Enkripsi Caesar *CipherKeyRandom*”. Dari penelitian tersebut telah dihasilkan aplikasi *chatting* untuk digunakan pada komputer yang terdapat pada satu area jaringan dengan kemampuan untuk melakukan enkripsi dan dekripsi pesan yang dikirim maupun diterima. Dari penelitian tersebut terdapat perbedaan dengan penelitian yang akan dilaksanakan, yaitu pada penelitian yang akan dilaksanakan aplikasi *chatting* yang akan dibuat akan digunakan secara *online* pada perangkat android.

Penelitian lain juga dilakukan oleh Ivan Luckiyana Firdaus, et al. pada tahun 2017 dengan judul “Aplikasi Kriptografi Komposisi One Time Pad Cipher Dan Affine Cipher” menghasilkan sebuah aplikasi yang dapat melakukan enkripsi dan dekripsi terhadap teks. Kelebihan dari aplikasi ini adalah memiliki sistem enkripsi yang aman karena menggunakan gabungan dari dua algoritma kriptografi. Kekurangan dari hasil penelitian tersebut adalah aplikasi yang dihasilkan hanya sebatas untuk melakukan enkripsi dan dekripsi teks saja. Perbedaan dengan penelitian yang dibuat adalah pada penelitian ini hasil akhirnya adalah sebuah aplikasi *realtimemessenger* pada *smartphone* android. Algoritma OTP pada penelitian ini dikombinasikan dengan algoritma ROT 13.

## II.2. Uraian Teoritis

### II.2.1. Kriptografi

Kriptografi (*Cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu kripito dan graphia. Kripito artinya menyembunyikan, sedangkan graphia artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi.

Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak lain yang tidak berhak untuk mengetahui isi pesan tersebut. Untuk menjaga pesan, maka pesan tersebut dapat diubah menjadi sebuah kode yang tidak dapat dimengerti pihak lain.

Enkripsi adalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang bisa dimengerti (*plaintext*) menjadi sebuah kode yang tidak bisa dimengerti (*chipertext*). Sedangkan proses kebalikannya untuk mengubah *chipertext* menjadi *plaintext* disebut dekripsi. Proses enkripsi dan deskripsi memerlukan suatu mekanisme dan kunci tertentu. Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang

mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. (Amin, M. Miftakul ; 2016 : 130-131)

### **II.2.2. Penjelasan Algoritma**

Algoritma adalah sistem kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistem tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada *software* komputer. *Software* terbangun atas susunan program) dan *syntax* (cara penulisan/pembuatan program). Untuk menyusun program atau *syntax*, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu *software*. Maka, algoritma berperan penting dalam penyusunan program atau *syntax* tersebut.

Pengertian algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*. Dalam dunia sehari-hari, mungkin tanpa kita sadari algoritma telah masuk dalam kehidupan kita. (Maulana, Gun Gun ; 2017 : 70)

### **II.2.3. Algoritma ROT 13**

Algoritma caesar merupakan metode enkripsi paling pertama, ditemukan dan digunakan oleh Julius Caesar dan tentaranya pada saat terjadi perang gaul di tahun 50 SM. Cara kerja dari algoritma ini, semua karakter alfabet digeser

sebanyak n-karakter. ROT13 (Rotate 13) adalah enkripsi substitution cipher yang umum digunakan di sistem operasi UNIX. Pada sistem enkripsi ROT13 sebuah huruf digantikan dengan huruf yang letaknya di atas 13 posisi darinya. (Frisai Anistasari Sinaga dan Mesran ; 2017 : 38-41)

Berikut tahap-tahap proses enkripsi :

a. ROT13 cipher

*Plaintext* = Frisai Anistasari Sinaga

$$P_1 = 'F' = 70 \quad \square C_1 = E(70) = (70 + 13) \bmod 256 = 83 = 'S'$$

$$P_2 = 'r' = 114 \quad \square C_2 = E(114) = (114 + 13) \bmod 256 = 127 = '\square'$$

$$P_3 = 'i' = 105 \quad \square C_3 = E(105) = (105 + 13) \bmod 256 = 118 = 'v'$$

$$P_4 = 's' = 115 \quad \square C_4 = E(115) = (115 + 13) \bmod 256 = 128 = 'ç'$$

$$P_5 = 'a' = 97 \quad \square C_5 = E(97) = (97 + 13) \bmod 256 = 110 = 'n'$$

$$P_6 = 'i' = 105 \quad \square C_6 = E(105) = (105 + 13) \bmod 256 = 118 = 'v'$$

$$P_7 = ' ' = 32 \quad \square C_7 = E(32) = (32 + 13) \bmod 256 = 45 = '-'$$

$$P_8 = 'A' = 65 \quad \square C_8 = E(65) = (65 + 13) \bmod 256 =$$

$$78 = \text{'N'}$$

$$P_9 = \text{'n'} = 110 \quad \square C_9 = E(110) = (110 + 13) \bmod 256 = 123 = \text{'\{'} ,$$

$$P_{10} = \text{'i'} = 105 \quad \square C_{10} = E(105) = (105 + 13) \bmod$$

$$256 = 118 = \text{'v'}$$

$$P_{11} = \text{'t'} = 116 \quad \square C_{11} = E(116) = (116 + 13) \bmod 256 = 129 = \text{'\ddot{u}'} ,$$

$$P_{12} = \text{'a'} = 97 \quad \square C_{12} = E(97) = (97 + 13) \bmod 256$$

$$= 110 = \text{'n'}$$

$$P_{13} = \text{'s'} = 115 \quad \square C_{13} = E(115) = (115 + 13) \bmod$$

$$256 = 128 = \text{'\c{c}'}$$

$$P_{14} = \text{'a'} = 97 \quad \square C_{14} = E(97) = (97 + 13) \bmod 256$$

$$= 110 = \text{'n'}$$

$$P_{15} = \text{'r'} = 114 \quad \square C_{15} = E(114) = (114 + 13) \bmod$$

$$256 = 127 = \text{'\square'}$$

$$P_{16} = \text{'i'} = 105 \quad \square C_{16} = E(105) = (105 + 13) \bmod$$

$$256 = 118 = \text{'v'}$$

$$P_{17} = \text{' ' } = 32 \quad \square C_{17} = E(32) = (32 + 13) \bmod 256$$

$$= 45 = \text{'-'}$$

$$P_{18} = \text{'S'} = 83 \quad \square C_{18} = E(83) = (83 + 13) \bmod 256$$

$$= 96 = \text{'\text{'}}$$

$$P_{19} = \text{'i'} = 105 \quad \square C_{19} = E(105) = (105 + 13) \bmod$$

$$256 = 118 = \text{'v'}$$

$$P_{20} = \text{'n'} = 110 \quad \square C_{20} = E(110) = (110 + 13) \bmod$$

$$256 = 123 = \text{'\{'} ,$$

$$P_{21} = 'a' = 97 \quad \square C_{21} = E(97) = (97 + 13) \bmod 256 \\ = 110 = 'n'$$

$$P_{22} = 'g' = 103 \quad \square C_{22} = E(103) = 103 + 13 \bmod 256 \\ = 116 = 't'$$

$$P_{23} = 'a' = 97 \quad \square C_{23} = E(97) = (97 + 13) \bmod 256 \\ = 110 = 'n'$$

dari proses enkripsi perhitungan ROT13 di atas, maka hasil *ciphertext*

sementara yaitu, **S□vçnv- N{vünçn□v-'v{ntn**

Selanjutnya *ciphertext* dari ROT13 tersebut

dienkripsikan kembali ke dalam algoritma *caesar cipher*.

#### b. *Caesarcipher*

*Plaintext* = S□vçnv-N{vünçn□v-'v{ntn (*ciphertext*  
ROT13)

$$P_1 = 'S' = 83 \quad \square C_1 = E(83) = (83 + 3) \bmod 256 = \\ 86 = 'V'$$

$$P_2 = '□' = 127 \quad \square C_2 = E(127) = (127 + 3) \bmod 256 \\ = 130 = 'é'$$

$$P_3 = 'v' = 118 \quad \square C_3 = E(118) = (118 + 3) \bmod 256 = \\ 121 = 'y'$$

$$P_4 = 'ç' = 128 \quad \square C_4 = E(128) = (128 + 3) \bmod 256 \\ = 131 = 'â'$$

$$P_5 = 'n' = 110 \quad \square C_5 = E(110) = (110 + 3) \bmod 256 =$$

$$113 = \text{'q'}$$

$$P_6 = \text{'v'} = 118 \quad \square C_6 = E(118) = (118 + 3) \bmod 256 \\ = 121 = \text{'y'}$$

$$P_7 = \text{'-'} = 45 \quad \square C_7 = E(45) = (45 + 3) \bmod 256 = \\ 48 = \text{'0'}$$

$$P_8 = \text{'N'} = 78 \quad \square C_8 = E(78) = (78 + 3) \bmod 256 = \\ 81 = \text{'Q'}$$

$$P_9 = \text{'{'}} = 123 \quad \square C_9 = E(97) = (123 + 3) \bmod 256 = \\ 126 = \text{'~'}$$

$$P_{10} = \text{'v'} = 118 \quad \square C_{10} = E(118) = (118 + 3) \bmod 256 \\ = 121 = \text{'y'}$$

$$P_{11} = \text{'ü'} = 129 \quad \square C_{11} = E(129) = (129 + 3) \bmod 256 \\ = 132 = \text{'ä'}$$

$$P_{12} = \text{'n'} = 110 \quad \square C_{12} = E(110) = (110 + 3) \bmod 256 \\ = 113 = \text{'q'}$$

$$P_{13} = \text{'ü'} = 128 \quad \square C_{13} = E(128) = (128 + 3) \bmod 256 \\ = 131 = \text{'â'}$$

$$P_{14} = \text{'n'} = 110 \quad \square C_{14} = E(110) = (110 + 3) \bmod 256 \\ = 113 = \text{'q'}$$

$$P_{15} = \text{'□'} = 127 \quad \square C_{15} = E(127) = (127 + 3) \bmod 256 \\ = 130 = \text{'é'}$$

$$P_{16} = \text{'v'} = 118 \quad \square C_{16} = E(118) = (118 + 3) \bmod 256 \\ = 121 = \text{'y'}$$

$$P_{17} = \text{'-'} = 45 \quad \square C_{17} = E(45) = (45 + 3) \bmod 256 = 48 = \text{'0'}$$

$$P_{18} = \text{' ' } = 96 \quad \square C_{18} = E(96) = (96 + 3) \bmod 256 = 99 = \text{'c'}$$

$$P_{19} = \text{'v'} = 118 \quad \square C_{19} = E(118) = (118 + 3) \bmod 256 = 121 = \text{'y'}$$

$$P_{20} = \text{'\{'} = 123 \quad \square C_{20} = E(123) = (123 + 3) \bmod 256 = 126 = \text{'\~'}$$

$$P_{21} = \text{'n'} = 110 \quad \square C_{21} = E(110) = (110 + 3) \bmod 256 = 113 = \text{'q'}$$

$$P_{22} = \text{'t'} = 116 \quad \square C_{22} = E(116) = (116 + 3) \bmod 256 = 119 = \text{'w'}$$

$$P_{23} = \text{'n'} = 110 \quad \square C_{23} = E(110) = (110 + 3) \bmod 256 = 113 = \text{'q'}$$

Proses enkripsi dari perhitungan *caesar cipher* di atas, menghasilkan *ciphertext* yaitu, **Véyâqy0Q~yâqâqéy0cy~qwq** (*caesar cipher*) Karakter-karakter yang dihasilkan berdasarkan perhitungan ini akan digunakan sebagai hasil akhir enkripsi pada aplikasi penyandian teks.

#### 1. Proses dekripsi

Dekripsi merupakan kebalikan dari enkripsi, karna proses dekripsi ini dimulai dengan menggunakan algoritma *caesar cipher* dan selanjutnya didekripsikan kembali ke dalam algoritma ROT13 *cipher*. Teks yang akan didekripsi merupakan teks yang dipilh dari *file word* dan *file* yang dipilih berikan teks yang tersandi/*ciphertext* dari nama lengkap penulis yaitu, **Véyâqy0Q~yâqâqéy0cy~qwq**

Berikut tahap-tahap proses dekripsi :

##### a. *Caesar cipher*

*Ciphertext* = Véyâqy0Q~yâqäqéy0cy~qwq (*ciphertext caesar*)

$$P_1 = 'V' = 86 \quad C_1 = E(86) = (86 - 3) \bmod 256 = 83 = 'S'$$

$$P_2 = 'é' = 130 \quad C_2 = E(130) = (130 - 3) \bmod 256 = 127 = 'ü'$$

$$P_3 = 'y' = 121 \quad C_3 = E(121) = (121 - 3) \bmod 256 =$$

$$118 = 'v'$$

$$P_4 = 'â' = 131 \quad C_4 = E(131) = (131 - 3) \bmod 256 = 128 = 'ç'$$

$$P_5 = 'q' = 113 \quad C_5 = E(113) = (113 - 3) \bmod 256 = 110 = 'n'$$

$$P_6 = 'y' = 121 \quad C_6 = E(121) = (121 - 3) \bmod 256 = 118 = 'v'$$

$$P_7 = '0' = 48 \quad C_7 = E(48) = (48 - 3) \bmod 256 = 45 = '-'$$

$$P_8 = 'Q' = 81 \rightarrow C_8 = E(81) = (81 - 3) \bmod 256 = 78 \\ = 'N'$$

$$P_9 = '~' = 126 \quad C_9 = E(126) = (126 - 3) \bmod 256 = 123 = '{'$$

$$P_{10} = 'y' = 121 \quad C_{10} = E(121) = (121 - 3) \bmod 256 =$$

$$118 = 'v'$$

$$P_{11} = 'ä' = 132 \quad C_{11} = E(132) = (132 - 3) \bmod 256 = 129 = 'ü'$$

$$P_{12} = 'q' = 113 \quad C_{12} = E(113) = (113 - 3) \bmod 256 = 110 = 'n'$$

$$P_{13} = 'â' = 131 \quad C_{13} = E(131) = (131 - 3) \bmod 256 = 128 = 'ç'$$

$$P_{14} = 'q' = 113 \quad C_{14} = E(113) = (113 - 3) \bmod 256 = 110 = 'n'$$

$$P_{15} = 'é' = 130 \quad C_{15} = E(130) = (130 - 3) \bmod 256 = 127 = 'ü'$$

$$P_{16} = 'y' = 121 \quad C_{16} = E(121) = (121 - 3) \bmod 256 =$$

$$118 = 'v'$$

$$P_{17} = '0' = 48 \quad C_{17} = E(48) = (48 - 3) \bmod 256 = 45 = '-'$$

$$P_{18} = 'c' = 99 \quad C_{18} = E(99) = (99 - 3) \bmod 256 = 96 = '‘'$$

$$P_{19} = 'y' = 121 \quad C_{19} = E(121) = (121 - 3) \bmod 256 = 118 = 'v'$$

$$P_{20} = \text{'~'} = 126 \quad C_{20} = E(126) = (126 - 3) \bmod 256 = 123 = \text{'{'}$$

$$P_{21} = \text{'q'} = 113 \quad C_{21} = E(113) = (113 - 3) \bmod 256 = 110 = \text{'n'}$$

$$P_{22} = \text{'w'} = 119 \quad C_{22} = E(119) = (119 - 3) \bmod 256 = 116 = \text{'t'}$$

$$P_{23} = \text{'q'} = 113 \quad C_{23} = E(113) = (113 - 3) \bmod 256 = 110 = \text{'n'}$$

Hasil *plaintext* sementara menggunakan *caesarcipher* dari perhitungan diatas yaitu, **S vçnv- N{vünçn v-'v{ntn**

*Plaintext* sementara yang dihasilkan dari *caesarcipher* didekripsikan kembali dengan algoritma ROT13 *cipher*, sehingga memperoleh teks asli.

a. ROT13 *cipher*

*Ciphertext* = S vçnv-N{vünçn v-'v{ntn (*plaintext caesar cipher*)

$$P_1 = \text{'S'} = 83 \quad C_1 = E(83) = (83 - 13) \bmod 256 = 70 = \text{'F'}$$

$$P_2 = \text{'ç'} = 127 \quad C_2 = E(127) = (127 - 13) \bmod 256 = 114 = \text{'r'}$$

$$P_3 = \text{'v'} = 118 \quad C_3 = E(118) = (118 - 13) \bmod 256 = 105 = \text{'i'}$$

$$P_4 = \text{'ç'} = 128 \quad C_4 = E(128) = (128 - 13) \bmod 256 = 115 = \text{'s'}$$

$$P_5 = \text{'n'} = 110 \quad C_5 = E(110) = (110 - 13) \bmod 256 = 97 = \text{'a'}$$

$$P_6 = 'v' = 118 \quad \square C_6 = E(118) = (118 - 13) \bmod 256 \\ = 105 = 'i'$$

$$P_7 = '-' = 45 \quad \square C_7 = E(45) = (45 - 13) \bmod 256 \\ = 32 = ''$$

$$P_8 = 'N' = 78 \quad \square C_8 = E(78) = (78 - 13) \bmod 256 = \\ 65 = 'A'$$

$$P_9 = '{' = 123 \quad \square C_9 = E(123) = (123 - 13) \bmod 256 \\ = 110 = 'n'$$

$$P_{10} = 'v' = 118 \quad \square C_{10} = E(118) = (118 - 13) \bmod \\ 256 = 105 = 'i'$$

$$P_{11} = 'ü' = 129 \quad \square C_{11} = E(129) = (129 - 13) \bmod \\ 256 = 116 = 't'$$

$$P_{12} = 'n' = 110 \quad \square C_{12} = E(110) = (110 - 13) \bmod \\ 256 = 97 = 'a'$$

$$P_{13} = 'ç' = 128 \quad \square C_{13} = E(128) = (128 - 13) \bmod \\ 256 = 115 = 's'$$

$$P_{14} = 'n' = 110 \quad \square C_{14} = E(110) = (110 - 13) \bmod \\ 256 = 97 = 'a'$$

$$P_{15} = '□' = 127 \quad \square C_{15} = E(127) = (127 - 13) \bmod \\ 256 = 114 = 'r'$$

$$P_{16} = 'v' = 118 \quad \square C_{16} = E(118) = (118 - 13) \bmod \\ 256 = 105 = 'i'$$

$$P_{17} = '-' = 45 \quad \square C_{17} = E(45) = (45 - 13) \bmod 256$$

$$= 32 = \text{' '}$$

$$P_{18} = \text{' , ' } = 96 \quad \square C_{18} = E(96) = (96 - 13) \bmod 256 =$$

$$83 = \text{'S'}$$

$$P_{19} = \text{'v' } = 118 \quad \square C_{19} = E(118) = (118 - 13) \bmod 256 = 8 = \text{'i'}$$

$$P_{20} = \text{'{ ' } = 123 \quad \square C_{20} = E(123) = (123 - 13) \bmod 256$$

$$= 110 = \text{'n'}$$

$$P_{21} = \text{'n' } = 110 \quad \square C_{21} = E(110) = (110 - 13) \bmod 256$$

$$= 97 = \text{'a'}$$

$$P_{22} = \text{'t' } = 116 \quad \square C_{22} = E(116) = (116 - 13) \bmod 256$$

$$= 103 = \text{'g'}$$

$$P_{23} = \text{'n' } = 110 \quad \square C_{23} = E(110) = (110 - 13) \bmod 256$$

$$= 97 = \text{'a'}$$

Hasil *plaintext* menggunakan kombinasi dekripsi dengan algoritma ROT13 dan *caesarcipher* dari perhitungan di atas yaitu Frisai Anitasari Sinaga.

#### II.2.4. Algoritma One Time Pad(OTP)

Algoritma *OneTimePad* (OTP) adalah *streamcipher* yang melakukan enkripsi dan dekripsi satukarakter setiap kali. Algoritma ini merupakan perbaikandari Vernam *cipher* untuk menghasilkan keamanan yang sempurna. *Cipher* ini termasuk ke dalam kelompok algoritma kriptografi simetri.

*OneTimePad* (*pad* =

kertas bloknot) berisi barisan karakter-karakter kunci yang dibangkitkan secara acak. Aslinya, satu buah *OneTimePad* adalah sebuah pita (tape) yang berisi

barisan karakter-karakter kunci. Satu pad hanya digunakan sekali (*onetime*) saja untuk mengenkripsi pesan, setelah itu *pad* yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain. Rumus dari enkripsi *OneTimePad* yaitu :

$$C_i = (P_i + K_i) \bmod 26$$

dan rumus dekripsi dari *OneTimePad* yaitu :

$$P_i = (C_i - K_i) \bmod 26$$

Keterangan rumus :

$C_i$  = Cipherteks (*Ciphertext*),

$P_i$  = Plainteks (*Plaintext*),

$K_i$  = kunci (*Key*). (Khoiruddin dan Khairina ; 2017 : 60)

Berikut adalah contoh proses enkripsi.

Plaintext : ONETIMEPAD

Kunci : TBFGRFARM

Misalkan  $A = 0, B = 1, \dots, Z = 25$ .

Chipertext : HOJKOREGHP

Yang diperoleh dengan cara sebagai berikut.

$$(O + T) \bmod 26 = H$$

$$(N + B) \bmod 26 = O$$

$$(E + F) \bmod 26 = J, \text{ dan seterusnya}$$

## II.2.5. Aplikasi *Chat Messenger* (*Chatting*)

Aplikasi adalah sebuah media penunjang dalam sebuah objek yang memiliki beberapa instruksi yang disusun sedemikian rupa sehingga dapat menghasilkan *input* dan *output*. Pengertian *Chat Messenger* atau *Chatting* adalah mengobrol jika diterjemahkan langsung dari bahasa Inggris. Dalam dunia komputer dan *internet*, pengertian *Chat Messenger* adalah suatu fasilitas dalam *internet* untuk berkomunikasi sesama pengguna *internet* yang sedang *on-line*. Komunikasi dapat berupa teks. (Sutikno, et al. ; 2018 : 2)

*Chatting* adalah percakapan yang biasanya dilakukan antara dua orang atau lebih secara langsung atau *realtime* dengan memanfaatkan fasilitas jaringan *internet*. Percakapan yang dilakukan bisa berupa teks, Ada banyak sekali aplikasi yang dapat digunakan untuk melakukan aktifitas *chatting*, diantaranya yang populer adalah IRC, Yahoo *Messenger*, AIM, Jabber dan masih banyak yang lainnya. (Dedisyah Putra ; 2017 : 434)

## **II.2.6. Bahasa Pemrograman Java**

Java dikembangkan oleh Sun Microsystems pada Agustus 1991. Java disebut juga merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu C dan C++. Pemrograman Java bersifat tidak bergantung pada platform, yang artinya, Java dapat dijalankan pada sembarang komputer dan bahkan pada sembarang sistem operasi. Sebagaimana halnya C++, salah satu bahasa yang mengilhami Java, Java juga merupakan bahasa pemrograman berorientasi objek. Sebagai bahasa pemrograman berorientasi objek, Java menggunakan kelas untuk membentuk suatu objek. Karakteristik Java antara lain adalah berorientasi objek

(*object-oriented*), terdistribusi (*distributed*), sederhana (*simple*), aman (*secure*), interpreted, robust, multithreaded, dan dinamis. (Annisa Rahmawati, et al. ; 2015 : 336)

## II.2.7. Android

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 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. (Ahmad : 2015 : 190-200)

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.2.7.1. Android Studio

Android *studio* adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *opensource* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada *event* Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan *Eclipse* sebagai IDE resmi untuk mengembangkan aplikasi Android.

Android *studio* sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan *Eclipse* disertai dengan ADT plugin (*Android Development Tools*).

Android *studio* memiliki fitur :

- a. Projek berbasis pada *GradleBuild*
- b. *Refactory* dan pembenahan *bug* yang cepat
- c. *Tools* baru yang bernama “*Lint*” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung *ProguardAndApp-signing* untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google *CloudPlatform* untuk setiap aplikasi yang dikembangkan. (Juansyah, Andi ; 2015)

### **II.2.8. Pengertian UML**

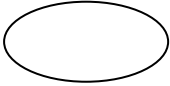


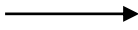
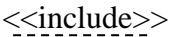
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa

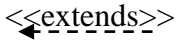
pemodalan umum dalam industri perangkat lunak dan pengembangan sistem (Ade Hendini ; 2016 : 108). Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

### II.2.8.1. 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. (Ade Hendini ; 2016 : 108)

**Tabel II.1. Use Case Diagram**

Gambar	Keterangan
	<i>UseCase</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktif, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>UseCase</i> , tetapi tidak memiliki kontrol terhadap <i>usecase</i>
	Asosiasi antara aktor dan <i>usecase</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan di dalam <i>usecase</i> lain ( <i>required</i> ) atau pemanggilan <i>usecase</i> oleh <i>usecase</i> lain, contohnya adalah



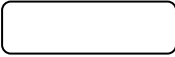
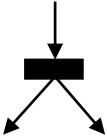

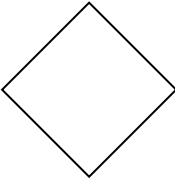
	pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>usecase</i> lain jika kondisi atau syarat

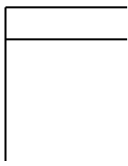
(Sumber : Ade Hendini ; 2016)

### II.2.8.2. Activity Diagram

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. (Ade Hendini ; 2016 : 109) Simbol-simbol yang digunakan dalam *activityDiagram* yaitu:

**Tabel II.2. Activity Diagram**

Gambar	Keterangan
	<i>StartPoint</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>EndPoint</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>DecisionPoints</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>





	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa
---	---

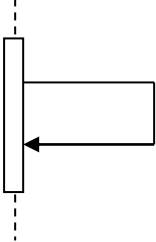


(Sumber : Ade Hendini ; 2016)

### II.2.8.3. *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan obyek pada *use case* dengan mendeskripsikan waktu hidup obyek dan pesan yang dikirimkan dan diterima antar obyek (Ade Hendini ; 2016 : 110). Simbol-simbol yang digunakan dalam *SequenceDiagram* yaitu:

**Tabel II.3. *Sequence Diagram***

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>BoundaryClass</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak
	<i>Controlclass</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i> , simbol mengirim pesan antar <i>class</i>

	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri</p>
	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i></p>

(Sumber : Ade Hendini ; 2016)

#### II.2.8.4. 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 (Ade Hendini ; 2016 : 110).

**Tabel II.4. Multiplicity Class Diagram**

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

(Sumber : Ade Hendini ; 2016)