BAB II
TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, yaitu dari komponen Crypto dan Graphia. Crypto berarti penyembunyian, sedangkan graphia berarti ilmu. Kriptografi adalah disiplin ilmu yang mencakup ilmu dan seni menjaga kerahasiaan informasi, menggunakan prosedur matematika yang berkaitan dengan beberapa aspek keamanan informasi, termasuk kerahasiaan data, validitas data, integritas data, dan autentikasi data, yang dijalankan oleh seorang kriptografer. Kriptografi berarti "tulisan rahasia." Definisi ini merujuk pada uraian Bruce Schneier dalam bukunya Applied Cryptography (1996), yang menyatakan bahwa kriptografi adalah seni dan ilmu untuk memastikan keamanan komunikasi. Sebagai perbandingan terhadap pengertian tersebut di atas, Rinaldi Munir mengutip definisi yang dikemukakan oleh Alfret J. Menezes, Paul C. van Oorschot, dan Scott A. Vanston dalam buku mereka, Handbook of Applied. Kriptografi (1996) merupakan disiplin ilmu yang mengkaji metode-metode matematika yang relevan dengan unsur-unsur keamanan informasi, termasuk keamanan, integritas data, dan autentikasi. Julius Caesar mengubah alfabet dengan mengganti setiap huruf, sehingga 'a' menjadi 'd', 'b' menjadi 'e', 'c' menjadi 'f', dan seterusnya. (Mukhtar, 2018). Empat tujuan mendasar kriptografi, yang juga merupakan aspek-aspek keamanan informasi, adalah sebagai berikut:

1. Kerahasiaan (*Confidentiality*)
2. Integritas Data (*DataIntegrity*)
3. Otentikasi (*Autentication*)
4. Ketiadaan Peyangkalan (*Nonrepudiation*)

Kriptografi, sebagai disiplin ilmu, memiliki banyak terminologi penting; Di antara terminologi yang sering digunakan dalam kriptografi adalah:

1. Plaintext: Plaintext mengacu pada pesan asli yang ditujukan untuk transmisi dan perlindungan. Pesan ini hanyalah informasi tersebut.

2. Ciphertext: Ciphertext mengacu pada komunikasi yang telah dikodekan.

3. Cipher: Cipher adalah algoritma matematika yang digunakan untuk mengkodekan plaintext menjadi ciphertext.

4. Enkripsi: Enkripsi adalah prosedur yang digunakan untuk mengubah plaintext menjadi ciphertext.

5. Dekripsi: Dekripsi adalah prosedur yang dilakukan untuk mengambil plaintext dari ciphertext.

6. Kriptanalisis: Disiplin menguraikan ciphertext secara tidak sah.

7. Kriptografi: Disiplin matematika yang menjadi dasar kriptografi dan kriptanalisis.

Algoritma kriptografi adalah fungsi matematika yang digunakan untuk tujuan enkripsi dan dekripsi. Algoritma kriptografi menunjukkan kekuatan yang lebih besar ketika durasi yang dibutuhkan untuk memecahkan kode diperpanjang. Algoritma lebih aman untuk digunakan. Untuk mengodekan dan mendekodekan data Kriptografi menggunakan algoritma (sandi) dan kunci. Algoritma kriptografi kontemporer bergantung pada keamanan enkripsi kunci daripada kerahasiaan algoritma itu sendiri. Plaintext yang identik, ketika dikodekan dengan berbagai kunci, akan menghasilkan ciphertext yang berbeda. Akibatnya, teknik kriptografi mungkin universal dan dapat diakses oleh siapa saja; Namun, tanpa kunci, informasi terenkripsi tetap kebal terhadap dekripsi. Sistem kriptografi terdiri dari algoritma kriptografi, kunci, dan semua plaintext, ciphertext, dan kunci potensial (Megantara & Rafrastara, 2019).

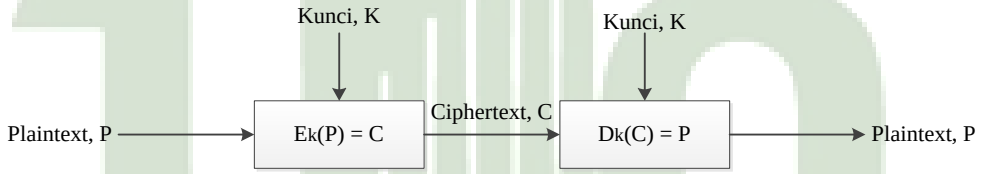
Kriptografi dulunya didefinisikan sebagai ilmu tentang penyembunyian pesan. Dalam istilah kontemporer, kriptografi adalah disiplin ilmu yang menggunakan metode matematika untuk menyediakan keamanan informasi, yang meliputi keamanan, integritas data, dan autentikasi entitas. Pemahaman kontemporer tentang kriptografi tidak hanya mencakup penyembunyian pesan tetapi juga serangkaian pendekatan komprehensif yang memastikan keamanan informasi (Karman & Nurhasan, 2019).

2.1.1 Jenis Kriptografi

Algoritma kriptografi dikategorikan menjadi dua jenis menurut kunci yang digunakan: Algoritma Simetris dan Algoritma Asimetris.

1. Algoritma Kriptografi Simetris

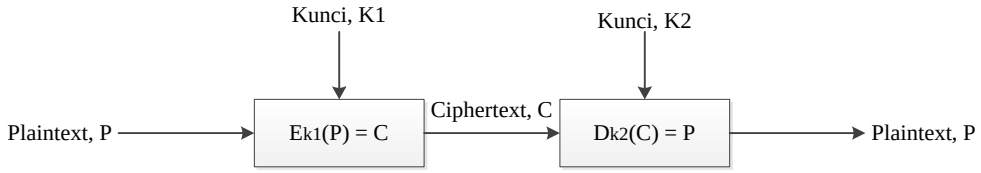
Sistem kriptografi kunci simetris, yang disebut sebagai "kunci simetris," beroperasi dengan asumsi bahwa pengirim dan penerima memiliki kunci yang sama sebelum pertukaran pesan. Keamanan teknik kriptografi simetris bergantung pada keamanan kunci. Sandi yang menggunakan kriptografi simetris sering kali dieksekusi dalam mode blok, di mana enkripsi/dekripsi terjadi pada blok data, atau dalam mode sandi aliran, di mana prosedur dilakukan pada satu bit atau byte data. Tujuan utama kriptografi simetris adalah untuk menjaga enkripsi data yang dikirimkan melalui saluran yang tidak aman dan untuk memastikan privasi data yang disimpan pada perangkat penyimpanan yang rentan. Gambar 2.1 mengilustrasikan skema proses kriptografi simetris (Basim & Painem, 2020).



Gambar 2.1. Algoritma Kriptografi Simetri
(Sumber :Basim & Painem, 2020)

2. Algoritma Asimetri

Dalam kriptografi asimetris, kunci enkripsi bersifat publik dan dapat diakses oleh siapa saja, sedangkan kunci dekripsi bersifat privat dan hanya diketahui oleh penerima pesan. Dalam kriptografi asimetris, setiap pengguna program memiliki sepasang kunci: kunci privat dan kunci publik. Pengirim mengenkripsi pesan dengan kunci publik penerima. Hanya penerima pesan yang dapat mendekodekannya, karena hanya mereka yang memiliki kunci rahasia unik. Sistem kriptografi diilustrasikan dalam Gambar 2.2. (Basim & Painem, 2020).



Gambar 2.2. Algoritma Kriptografi Asimetri
(Sumber :Basim & Painem, 2020)

2.1.2 Keamanan Data

Data dapat dipahami sebagai representasi realitas yang dicirikan oleh nilai, angka, urutan, karakter, atau simbol yang menyampaikan makna tertentu. Informasi digambarkan sebagai hasil pemrosesan data yang disajikan dengan cara yang lebih bermanfaat bagi penerima, yang berfungsi sebagai alat untuk pengambilan. Keamanan adalah kondisi terbebas dari bahaya.

Frasa ini mengacu pada aktivitas kriminal dan berbagai jenis kecelakaan. Keamanan mencakup berbagai domain, termasuk keamanan nasional terhadap terorisme, keamanan siber terhadap peretas, keamanan perumahan terhadap pencurian dan intrusi, keamanan finansial terhadap kemerosotan ekonomi, dan berbagai skenario terkait lainnya. Host komputer yang terhubung ke jaringan memberikan risiko keamanan yang lebih besar bagi host yang tidak terhubung. Mengendalikan keamanan jaringan membantu mengurangi bahaya ini (Santoso & Fakhriza, 2018).

2.1.3 Aspek Keamanan Data

Kriptografi adalah tindakan pengamanan data yang diterapkan untuk mencegah akses tidak sah ke informasi rahasia dan penting. Sejumlah persyaratan berkaitan dengan masalah keamanan data, khususnya:

1. Kerahasiaan

Kerahasiaan adalah mekanisme yang digunakan untuk melindungi informasi dari akses tidak sah oleh pihak mana pun. Akibatnya, informasi hanya dapat diakses oleh individu yang berwenang.

2. Autentikasi

Autentikasi adalah layanan yang berkaitan dengan identifikasi entitas yang mencari akses ke sistem informasi (autentikasi entitas) dan verifikasi validitas data dari sistem informasi (autentikasi asal data). Dalam pertukaran informasi, penting bagi kedua belah pihak untuk memverifikasi keaslian identitas pengirim sebagaimana dinyatakan.

3. Integritas Data

Integritas data adalah layanan yang dirancang untuk menggagalkan perubahan informasi secara ilegal. Untuk menjaga integritas data, sistem informasi harus mampu mendeteksi modifikasi data. Manipulasi data mencakup penyisipan, penghapusan, atau pengaturan data. Persyaratan ini menjamin bahwa setiap pesan yang dikirim akan sampai ke penerima yang dituju tanpa perubahan, duplikasi, kerusakan, penataan ulang, atau penambahan konten. Tak terbantahkan

4. *Nonrepudiation*

Nonrepudiation memastikan bahwa pengirim atau penerima tidak dapat membantah telah mengirim atau menerima pesan atau informasi. Jika pesan dikirim, penerima dapat memverifikasi bahwa pesan tersebut benar-benar dikirim oleh pengirim yang ditunjuk. Sebaliknya, setelah menerima pesan, pengirim dapat memverifikasi bahwa pesan tersebut telah diakui oleh penerima yang dituju (Suhardi, 2016)

2.1.4 Perkembangan Kriptografi

Teknik kriptografi dapat dikategorikan menjadi dua jenis yang berbeda:

1. Kriptografi Tradisional

Ini adalah bentuk enkripsi yang digunakan pada zaman dahulu, sebelum penemuan komputer, atau selama pengembangan awal ketika komputer tidak serumit model kontemporer. Enkripsi klasik ini hanya mengacak huruf A-Z; tidak direkomendasikan untuk menjaga informasi penting karena kemudahan penggunaannya.

2. Kriptografi Kontemporer

Ini adalah teknik kriptografi yang berfungsi dalam mode bit, bukan mode karakter. Metode kriptografi dalam mode bit ini menandakan bahwa semua

data dan informasi (kunci, plaintext, dan ciphertext) direpresentasikan sebagai urutan string biner yang terdiri dari 0 dan 1. Metode enkripsi dan dekripsi beroperasi pada semua data dan informasi sebagai urutan bit. Urutan bit yang mewakili plaintext diubah menjadi ciphertext sebagai urutan bit, dan sebaliknya (Suhardi, 2016).

2.2 Kombinasi Algoritma Beaufort Cipher dan Algoritma Hill Cipher

Sandi Beaufort adalah versi sandi Vigenère, dengan metode enkripsi dan dekripsi yang sangat mirip dengan sandi Vigenère. Sandi Beaufort dirancang oleh Laksamana Sir Francis Beaufort dari Angkatan Laut Kerajaan, yang juga mengembangkan skala Beaufort, alat meteorologi untuk mengukur kecepatan angin. Perbedaan yang lebih jelas antara kedua pendekatan tersebut terletak pada fungsi kunci; dalam sandi Vigenère, kunci berfungsi sebagai penjumlah pada teks biasa dan pengurang pada teks sandi. Dalam rumus sandi Beaufort, kunci dikurangi dari teks biasa atau teks sandi. Rumus enkripsi dan dekripsi sandi Beaufort adalah sebagai berikut:

$$Cc = (k-Pc) \bmod 26 \quad (2.1)$$

$$Pc = (k-Cc) \bmod 26 \quad (2.2)$$

Keterangan:

C: memodelkan ciphertext

P: memodelkan plaintext

K: memodelkan kunci.(Rachmadsyah et al., 2020)

Sandi Hill, sandi polialfabetik, dapat didekonstruksi sebagai sandi blok karena teks disegmentasi menjadi blok-blok dengan ukuran tertentu. Setiap karakter dalam blok akan memengaruhi karakter lain dalam proses enkripsi dan dekripsi, memastikan bahwa karakter identik tidak dipetakan satu sama lain. Metode kriptografi ini menggunakan matriks persegi sebagai kunci untuk proses enkripsi dan dekripsi. Sandi Hill dikembangkan oleh Lester S. Hill pada tahun 1929. Sandi Hill tidak mengganti huruf identik dalam teks biasa dengan huruf identik yang sesuai dalam teks sandi, karena menggunakan perkalian matriks untuk enkripsi dan dekode. Sandi Hill adalah teknik kriptografi klasik yang menimbulkan tantangan

signifikan bagi kriptanalis ketika hanya teks sandi yang tersedia. Meskipun demikian, strategi ini dapat dengan mudah diselesaikan asalkan kriptanalis memiliki berkas teks sandi dan segmen berkas teks biasa. Metode kriptoanalisis ini disebut sebagai serangan teks biasa yang tidak diketahui. (Megantara & Rafrastara, 2019) Proses enkripsi Hill cipher terjadi pada setiap blok plaintext. Ukuran blok setara dengan dimensi matriks kunci. Sebelum melakukan segmentasi teks menjadi serangkaian blok, plaintext diubah menjadi nilai numerik, dengan A diberi nilai 1, B diberi nilai 2, dan berlanjut secara berurutan hingga Y, yang diberi nilai 25. Z diberi nilai 0.

Dengan menggunakan rumus enkripsi dan dekripsi, maka penggunaanalgoritma Beaufort dan algoritma Hill Cipher dengan plaintext = KAMPUS dan kunci = BFJI dapat dicontohkan sebagai berikut :

Plaintext(P) : KAMPUS

Kunci (K) : BFJI

Pada penelitian ini akan digunakan himpunan 26 karakter sehingga didapatkan nilai masing-masing karakter adalah :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

1. Proses Enkripsi

C1 = (K1 - P1) mod 26
= (B - K mod 26
= (1 - 10) mod 26
= 17
= R

C2 = (K2 - P2) mod 26
= (F - A) mod 26
= (5 - 0) mod 26
= 5
= F

C3 = (K3 - P3) mod 26
= (J - M) mod 26
= (9 - 12) mod 26
= 23
= X

C4 = (K4 - P4) mod 26
= (I - P) mod 26
= (8 - 15) mod 26
= 19
= T

C5 = (K5 - P5) mod 26

C6 = (K6 – P6) mod 26

= (B - U) mod 26 = (F- S) mod 26

= (1 - 20) mod 26 = (5 - 18) mod 26

= 7 = 15

= H = P

Ciphertext yang dihasilkan menggunakan algoritma Beaufort adalah '**RFXTHP**'. Selanjutnya hasil enkripsi tersebut di enkripsi kembali menggunakan algoritma Hill Cipher sebagai berikut :

Plaintext = RFXTHP = 17 5 23 19 7 15

Kunci = BFJI = 1 5 9 8

Proses enkripsi dilakukan dengan cara perhitungan matriks dengan plaintext dibagi menjadi blok yang masing-masing bloknya berukuran 2 karakter.Blok plaintext dienkripsi dengan kunci :

$C_{1,2}$ $= \begin{bmatrix}1 & 5\\ 9 & 8\end{bmatrix}\begin{bmatrix}17\\ 5\end{bmatrix}(mod\ 26) = \begin{bmatrix}16\\ 11\end{bmatrix}$ = QL

$C_{3,4}$ $= \begin{bmatrix}1 & 5\\ 9 & 8\end{bmatrix}\begin{bmatrix}23\\ 19\end{bmatrix}(mod\ 26) = \begin{bmatrix}14\\ 21\end{bmatrix}$ = OV

$C_{5,6}$ $= \begin{bmatrix}1 & 5\\ 9 & 8\end{bmatrix}\begin{bmatrix}7\\ 15\end{bmatrix}(mod\ 26) = \begin{bmatrix}4\\ 1\end{bmatrix}$ = EB

Hasil akhir dari proses enkripsi menggunakan kombinasi algoritma Beaufort dan Hill Cipher adalah "**QLOVEB**".

2. Proses Dekripsi

Tahap awal dari proses dekripsi adalah mencari nilai inverse dari kunci yang digunakan. Mencari invers dapat dilakukan dengan menggunakan metode operasi baris (row operation) atau metode determinan. Setelah melakukan perhitungan, didapat matriks K^{-1}yang merupakan invers dari matriks K, yaitu:

$$K^{-1} = \begin{bmatrix}27 & -54\\ -18 & 45\end{bmatrix} = \begin{bmatrix}4 & 17\\ 15 & 7\end{bmatrix}(mod\ 26)$$

Kunci K^{-1}yang digunakan untuk melakukan dekripsi ini telah memenuhi persamaan (1) karena:

$$K.K^{-1} = \begin{bmatrix}53 & 234\\ 26 & 105\end{bmatrix} = K^{-1} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}(mod\ 26) = I$$

Selanjutnya dilakukan proses dekripsi menggunakan algoritma Hill Cipher sebagai berikut :

$C_{1,2}$ $= \begin{bmatrix} 4 & 17 \\ 15 & 7 \end{bmatrix} \begin{bmatrix} 16 \\ 11 \end{bmatrix} (mod\ 26) = \begin{bmatrix} 17 \\ 5 \end{bmatrix}$ = RF

$C_{3,4}$ $= \begin{bmatrix} 4 & 17 \\ 15 & 7 \end{bmatrix} \begin{bmatrix} 14 \\ 21 \end{bmatrix} (mod\ 26) = \begin{bmatrix} 23 \\ 19 \end{bmatrix}$ = XT

$C_{5,6}$ $= \begin{bmatrix} 4 & 17 \\ 15 & 7 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} (mod\ 26) = \begin{bmatrix} 7 \\ 15 \end{bmatrix}$ = HP

Hasil dekripsi yang di dapat menggunakan algoritma Hill Cipher adalah "**RFXTHP**". Selanjutnya dilakukan kembali proses dekripsi menggunakan algoritma Beaufort sebagai berikut :

P1	= (K1 - C1) mod 26	P2	= (K2 - C2) mod 26
	= (B - R) mod 26		= (F - F) mod 26
	= (1 - 17) mod 26		= (5 - 5) mod 26
	= 10		= 0
	= K		= A
P3	= (K3 - C3) mod 26	P4	= (K4 - C4) mod 26
	= (J - X) mod 26		= (I - T) mod 26
	= (9 - 23) mod 26		= (8 - 19) mod 26
	= 12		= 15
	= M		= P
P5	= (K5 - C5) mod 26	P6	= (K6 -C6) mod 26
	= (B - H) mod 26		= (F- P) mod 26
	= (1 - 7) mod 26		= (5 - 15) mod 26
	= 20		= 18
	= U		= S

Dari proses dekripsi menggunakan algoritma Beaufort di dapatkan hasil akhir *plaintext*berupa '**KAMPUS**'.

2.3 Dokumen

Istilah 'dokumen' umumnya didefinisikan oleh para profesional dalam dua cara:

1. Sumber tertulis informasi sejarah, berbeda dengan kesaksian lisan,

artefak, relik yang dilukis, dan peninggalan arkeologi.

2. Dirancang untuk korespondensi resmi dan dokumen negara, termasuk perjanjian, undang-undang, hibah, konsesi, dan hal-hal serupa.

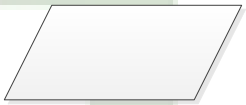
Selain itu, dokumen (dokumentasi) mencakup setiap proses pembuktian yang berasal dari berbagai sumber, termasuk bentuk tertulis, lisan, fotografi, atau arkeologi. Berdasarkan definisi yang diberikan, dokumen adalah setiap bentuk catatan tertulis, gambar, atau rekaman yang relevan dengan persyaratan manajemen, yang mencakup format fisik dan digital (Mohamad Ali Murtadho, 2016).

2.4 *Flowchart*

Flowchart atau Bagan alir adalah representasi grafis dari langkah-langkah dan urutan prosedur dalam suatu program. Bagan alir adalah diagram yang menggambarkan perkembangan logis dari suatu program atau prosedur sistem. Bagan alir sebagian besar berfungsi sebagai alat untuk komunikasi dan dokumentasi. Ada lima kategori bagan alir, termasuk:

1. Bagan Alir Sistem, adalah diagram yang menggambarkan alur kerja sistem secara menyeluruh.
2. Bagan Alir Dokumen, juga dikenal sebagai bagan alir formulir, menggambarkan perkembangan laporan dan formulir, termasuk salinannya.
3. Bagan Alir Skema, menggambarkan prosedur dalam sistem dengan simbol bagan alir standar dan ilustrasi komputer dan peralatan lain yang digunakan oleh sistem.
4. Bagan Alir Program, adalah diagram yang menggambarkan langkah-langkah berurutan dari proses program secara terperinci.
5. Bagan Alir Proses adalah diagram yang umum digunakan dalam teknik industri untuk menggambarkan langkah-langkah dalam suatu teknik (Verawati & Liksha, 2018)

Tabel 2.1. Simbol-simbol Flowchart

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROCESS	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINE PROCESS	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchat yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchat yang berada pada halaman berbeda

(Sumber : Verawati & Liksha, 2018)

2.5 Pemrograman Java

Java adalah teknologi dalam bidang perangkat lunak komputer, yang berfungsi sebagai bahasa pemrograman dan platform. Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang terbiasa dengan C/C++. Java adalah bahasa pemrograman berorientasi objek yang mewakili paradigma pemrograman progresif. Java, sebagai bahasa pemrograman, dirancang untuk keandalan dan keamanan. Java dirancang untuk kompatibilitas di semua platform. Selain itu, Java dirancang untuk mengembangkan aplikasi dengan kinerja optimal, yang dicontohkan oleh aplikasi basis data Oracle 8i/9i, yang pada dasarnya dibangun menggunakan bahasa pemrograman Java.

Java adalah bahasa komputer yang banyak digunakan, yang dibuat oleh Sun Microsystems. Java sebagian besar digunakan untuk mengembangkan aplikasi asli untuk Android. Bahasa pemrograman ini multiplatform, yang menunjukkan penerapannya di beberapa sistem, termasuk desktop, Android, dan sistem operasi Linux. Java bersifat netral arsitektur karena kompilernya menghasilkan kode yang kompatibel dengan semua arsitektur perangkat keras, yang dikenal sebagai Java Bytecode (Mubarok et al., 2021).

2.6 Variabel

Variabel adalah segmen memori bernama yang digunakan untuk penyimpanan informasi dalam program Java. Setiap segmen memori yang ditetapkan dalam program Java dibatasi untuk menyimpan data dengan tipe tertentu. Variabel yang ditetapkan untuk menyimpan integer tidak dapat menampung nilai pecahan, seperti 0,45. Variabel yang ditetapkan untuk merujuk objek Hat dapat secara eksklusif merujuk objek Hat. Sifat tipe data yang tidak dapat diubah untuk setiap variabel memungkinkan kompiler untuk memastikan kesesuaian setiap variabel yang dibuat dalam program Java. Jika suatu metode dalam program dirancang untuk memproses integer, kompiler dapat memverifikasi apakah akan menggunakan metode tersebut untuk data dengan tipe data integer.

Dalam program Java, nilai data eksplisit disebut sebagai literal. Setiap literal dapat memiliki tipe data yang berbeda. Misalnya, 25 adalah literal integer yang diklasifikasikan sebagai tipe int. Anda harus menentukan nama dan tipenya dalam pernyataan deklarasi (Siahaan & Sianipar, 2020)

2.7 Android

Android adalah sistem operasi turunan Linux yang digunakan untuk perangkat seluler, termasuk telepon pintar dan komputer tablet (PDA). Android adalah sistem operasi berbasis Linux untuk perangkat seluler yang mencakup sistem operasi, middleware, dan aplikasi. Android menawarkan platform terbuka bagi pengembang untuk membangun aplikasi yang memfasilitasi aktivitas di berbagai domain, yang memungkinkan aksesibilitas bagi setiap pengguna di perangkat mereka. Android adalah sistem operasi turunan Linux yang dikembangkan untuk perangkat seluler layar sentuh, termasuk telepon pintar dan komputer tablet. Android awalnya dibuat oleh Android, Inc. dengan dukungan finansial dari Google, yang kemudian dihentikan pada tahun 2005 (Khaliq, 2021).

Sistem operasi Android versi 1.0 resmi dirilis pada 23 September 2008. Sekitar satu bulan kemudian, pada 22 Oktober 2008, telepon pintar perdana yang beroperasi pada Android 1.0, HTC Dream, diperkenalkan ke pasaran. Pada 9 Februari, Android versi 1.1 dirilis untuk memperbaiki kesalahan dari versi sebelumnya dan menyertakan fungsionalitas baru. Setelah versi 1.1, iterasi Android berikutnya mengadopsi nomenklatur permen dalam urutan abjad, dimulai dengan 1.5 Cupcake, dirilis pada 30 April 2009. Versi Android berikutnya, khususnya Donut, Éclair, Froyo, dan Gingerbread, secara eksklusif dirancang untuk telepon pintar. Meskipun demikian, Apple memperkenalkan iPad pada tahun 2010, sehingga meningkatkan minat populer terhadap komputer tablet. Banyak pengembang Android mencoba membuat tablet Android untuk menyaingi iPad, termasuk Samsung Galaxy Tab, yang menggunakan versi modifikasi dari Gingerbread. Google dan OHA memulai pengembangan versi baru Android yang dioptimalkan untuk tablet.

Android Honeycomb dirilis pada 22 Februari 2011, diikuti oleh peluncuran

tablet Honeycomb pertama, Motorola Xoom, pada 24 Februari 2011. Pada 19 Oktober 2011, Android merilis Ice Cream Sandwich, versi yang dioptimalkan untuk telepon pintar dan tablet. Pembaruan Android yang akan datang, Jelly Bean, berupaya untuk menyempurnakan fungsionalitas Ice Cream Sandwich yang ada dengan memperbaiki masalah dan menyertakan kemampuan baru. Pada 3 September 2013, iterasi Android berikutnya, Android 4.4 Kit Kat, diumumkan. Android telah memperoleh otorisasi dari Nestlé dan Hershey, pemilik merek dagang Kit Kat. Sebelum pengungkapan ini, beberapa orang menduga bahwa iterasi Android berikutnya akan diberi nama 5.0 dan diberi nama Key Lime Pie. Tabel berikutnya mencantumkan semua sistem operasi Android yang telah dirilis hingga saat ini. Pada saat tulisan ini dibuat, sistem operasi Android terbaru adalah Android 6.0 Marshmallow Yusfrizal, 2019).

2.8 Versi Android

Android adalah sistem operasi seluler yang didasarkan pada versi modifikasi *linux*. Hampir semua *smartphone* memiliki sistem operasi *android*. Dalam pengembangannya andorid telah mengalami cukup banyak pembaruan sejak awal dirilis yang akan ditunjukan pada tabel 2.2.(Pramadana et al., 2019)

Tabel 2.2. Versi-versi Android

Versi	Nama	Tanggal Rilis
1.5	*Cupcake*	30 April 2009
1.6	*Donut*	15 September 2009
2.0-2.1	*Éclair*	26 Oktober 2009
2.2	*Froyo*	20 Mei 2010
2.3-2.3.2	*Gingerbread*	6 Desember 2010
2.3.3-2.3.7	*Gingerbread*	9 Februari 2011
3.1	*Honeycomb*	10 Mei 2011
3.2	*Honeycomb*	15 Juli 2011
4.0.3-4.0.4	*Ice Cream Sandwich*	16 Desember 2011
4.1.x	*Jelly Bean*	9 Juli 2012
4.2.x	*Jelly Bean*	13 November 2012
4.3.x	*Jelly Bean*	24 Juli 2013
4.4.x	*Kitkat*	31 Oktober 2013
5.0	*Lollipop*	15 Oktober 2014
6.0	*Marshmallow*	5 Oktober 2015

7.1	*Nougat*	4 Oktober 2016
7.4	*Nougat*	5 Desember 2016
8.0	*Oreo*	21 Agustus 2017
9.0	*Pie*	6 Agustus 2018
10.0	*Android Q*	7 Agustus 2019

(Sumber : Pramadana et al., 2019)

2.9 Android Studio

Android Studio merupakan lingkungan pengembangan perangkat lunak terpadu Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan Intellij IDEA. Selain merupakan editor kode Intellij dan alat pengembang yang berdaya guna, Android Studio juga menawarkan banyak fitur untuk meningkatkan produktivitas saat membuat aplikasi Android.Android studio sendiri dikembangkan berdasarkan Intellij IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android Studio memiliki fitur :

1. Projek berbasis pada Gradle Build
2. Refactory dan pembenahan bug yang cepat
3. Tools baru yang bernama “Lint” diklaim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
4. Mendukung Proguard And App-signing untuk keamanan.
5. Memiliki GUI aplikasi android lebih mudah
6. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan. (Khaliq, 2021)

2.10 Android SDK (*Software Development Kit*)

Android *SDK* mencakup perangkat *tools* pengembangan yang komprehensif. Android *SDK* terdiri dari *debugger*, *libraries*, *handsetemulator*, dokumentasi, contoh kode program dan tutorial. Saat ini *Android* sudah mendukung arsitektur x86 pada *Linux* (distribusi *Linux* apapun untuk *desktop* modern), Mac OS X 10.4.8 atau lebih, *Windows* XP atau Vista.Persyaratan mencakup *JDK*, Apache Ant dan Python 2.2 atau lebih.*IDE* yang didukung secara resmi adalah *Eclipse* 3.2 atau lebih dengan menggunakan plugin Android *DevelopmentTools* (*ADT*), dengan ini

pengembang dapat menggunakan *IDE* untuk mengedit dokumen *Java* dan *XML* serta menggunakan peralatan *commandline* untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android (misalnya *reboot*, menginstal paket perangkat lunak).

Android SDK mencakup perangkat tools pengembangan yang komprehensif. android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan ide untuk mengedit dokumen java dan XML serta menggunakan peralatan command line untuk menciptakan, membangun, melakukan debug aplikasi android dan pengendalian perangkat android. (Taruna et al., 2021)