BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan informasi merupakan aspek yang sangat penting dalam komunikasi digital. Seiring dengan pesatnya perkembangan teknologi dan meningkatnya ketergantungan masyarakat pada layanan berbasis internet, ancaman terhadap data pribadi dan informasi sensitif juga semakin besar. Serangan siber seperti penyadapan, peretasan, dan pencurian data sering kali terjadi, mengakibatkan kebocoran informasi yang dapat merugikan individu maupun organisasi. Oleh karena itu, diperlukan metode enkripsi yang kuat untuk melindungi data selama proses transmisi.

Dalam era digital yang semakin berkembang, keamanan informasi menjadi aspek yang sangat krusial, terutama dalam komunikasi berbasis jaringan yang rawan terhadap ancaman seperti penyadapan dan peretasan. Algoritma *Advanced Encryption Standard* (AES) sebagai metode enkripsi simetris memiliki efisiensi tinggi dalam mengamankan data. Di sisi lain, algoritma *Diffie-Hellman Key Exchange* (DHE) menawarkan pertukaran kunci secara aman di jaringan publik, namun perlu diuji efektivitas dan keamanannya saat diintegrasikan dalam sistem *end-to-end encryption*. Permasalahan utama yang ingin dipecahkan dalam penelitian ini adalah bagaimana mengimplementasikan dan mengombinasikan algoritma DHE dan AES secara efektif dalam sistem enkripsi pesan *end-to-end* agar dapat menjamin keamanan dan efisiensi proses komunikasi digital, khususnya dalam pengamanan pertukaran kunci dan perlindungan terhadap isi pesan dari akses pihak yang tidak berwenang.

Dalam implementasi enkripsi E2EE, algoritma kriptografi berperan penting dalam menjaga keamanan data. *Advanced Encryption Standard* (AES) merupakan salah satu algoritma enkripsi simetris yang telah menjadi standar global karena keamanannya yang tinggi dan efisiensinya dalam pemrosesan data. AES menggunakan kunci yang sama untuk proses enkripsi dan dekripsi, yang membuatnya sangat cepat dan efisien dalam mengamankan pesan. Namun, karena AES menggunakan sistem kunci simetris, muncul tantangan dalam mendistribusikan kunci secara aman antara pengirim dan penerima.

Untuk mengatasi permasalahan pertukaran kunci pada algoritma enkripsi simetris, digunakan *Diffie-Hellman Key Exchange* (DHE). DHE adalah salah satu algoritma pertukaran kunci yang memungkinkan dua pihak untuk berbagi kunci enkripsi secara aman melalui jaringan yang tidak aman, tanpa harus mengirimkan kunci tersebut secara langsung. Dengan menggunakan prinsip matematika eksponensial modular, DHE memungkinkan pembentukan kunci rahasia bersama yang hanya dapat dihitung oleh kedua pihak yang berkomunikasi. Hal ini membuat metode ini sangat efektif dalam melindungi proses pertukaran kunci dari serangan pihak ketiga.

Kombinasi antara *Diffie-Hellman Key Exchange* (DHE) dan *Advanced Encryption Standard* (AES) dalam enkripsi pesan *end-to-end* menjadi solusi yang efektif dalam meningkatkan keamanan komunikasi digital. DHE digunakan untuk mendistribusikan kunci secara aman, sedangkan AES digunakan untuk mengenkripsi pesan dengan efisiensi tinggi. Implementasi gabungan kedua algoritma ini dapat meningkatkan keamanan data dan mencegah akses tidak sah selama proses komunikasi berlangsung.

Penelitian ini bertujuan untuk mengimplementasikan algoritma *Diffie-Hellman Key Exchange* (DHE) dan AES dalam enkripsi pesan *end-to-end* serta menganalisis kinerja dan keamanannya. Melalui penelitian ini, diharapkan dapat diperoleh pemahaman yang lebih dalam mengenai efektivitas kombinasi kedua algoritma dalam menjaga kerahasiaan komunikasi digital. Selain itu, hasil penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem keamanan komunikasi yang lebih baik, terutama di era digital yang semakin rentan terhadap ancaman keamanan siber.

Berdasarkan latar belakang tersebut, penulis akan membuat skripsi dengan judul "**Implementasi Algoritma *Diffie-Hellman Key Exchange* (DHE) Dan AES Dalam Enkripsi Pesan *End-To-End***".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang mendasari penulis melakukan penelitian ini, penulis merumuskan beberapa rumusan masalah antara lain sebagai berikut:

1. Bagaimana cara merancang mekanisme pertukaran kunci yang aman antara pengirim dan penerima pesan dengan menggunakan algoritma *Diffie-Hellman Key Exchange* (DHE)?
2. Bagaimana menerapkan algoritma AES untuk melakukan enkripsi dan dekripsi pesan setelah kunci berhasil dibentuk melalui DHE?
3. Bagaimana mengintegrasikan algoritma DHE dan AES agar dapat membentuk sistem enkripsi pesan end-to-end yang efektif dan efisien?
4. Bagaimana tingkat efisiensi kombinasi algoritma DHE dan AES dalam keamanan komunikasi digital dari ancaman.

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak melebar dan memudahkan dalam proses penelitian maupun proses perancangan, maka diperlukan batasan masalah sebagai berikut :

1. Penelitian ini hanya membahas implementasi algoritma *Diffie-Hellman Key Exchange* (DHE) sebagai metode pertukaran kunci dan *Advanced Encryption Standard* (AES) sebagai metode enkripsi pesan.
2. Pengujian dilakukan dalam skenario komunikasi berbasis teks, tanpa mempertimbangkan jenis data lain seperti gambar, audio, atau video.
3. Penerapan algoritma ini menggunakan AES dengan 128 bit dan DHE dengan 2048 bit.
4. Sistem yang dikembangkan hanya berfokus pada keamanan enkripsi dan pertukaran kunci, tanpa membahas aspek lain seperti autentikasi pengguna atau manajemen sesi.
5. Pemodelan data menggunakan UML.

1.4 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan oleh penulis antara lain sebagai berikut :

1. Merancang mekanisme pertukaran kunci aman menggunakan algoritma *Diffie-Hellman Key Exchange* (DHE).
2. Menerapkan algoritma AES untuk melakukan proses enkripsi dan dekripsi pesan.
3. Mengintegrasikan algoritma DHE dan AES dalam satu sistem sehingga menghasilkan enkripsi pesan end-to-end.

4. Menguji dan menganalisis tingkat keamanan yang dihasilkan oleh kombinasi algoritma DHE dan AES dalam menjaga kerahasiaan pesan selama proses komunikasi berlangsung.

1.5 Manfaat Penelitian

1. Menambah wawasan dan literatur mengenai implementasi algoritma kriptografi dalam enkripsi pesan end-to-end.
2. Membantu dalam pengembangan sistem komunikasi yang lebih terlindungi dari ancaman peretasan atau penyadapan data.

1.6 Sistematika Penulisan

Untuk mempermudah dalam penyusunan dan memahami skripsi maka penulis menyajikan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini memuat tentang materi-materi pendukung dalam penyusunan skripsi, mulai dari teori-teori yang digunakan, konsep-konsep yang akan diterapkan dalam menyelesaikan permasalahan yang penulis teliti dalam penelitian ini.

BAB III : METODE PENELITIAN

Pada bab ini memuat mengenai metode yang penulis gunakan dalam

menyelesaikan rumusan masalah, tahap-tahap mengenai teknik pengolahan data, perancangan aplikasi, dan pembuatan aplikasi.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini memuat hasil-hasil yang didapat dari penelitian serta melakukan pembahasan atas hasil yang diperoleh. Kesulitan yang ditemukan saat perancangan dan pembuatan aplikasi.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini memuat kesimpulan dan saran penulis atas penelitian yang dilakukan

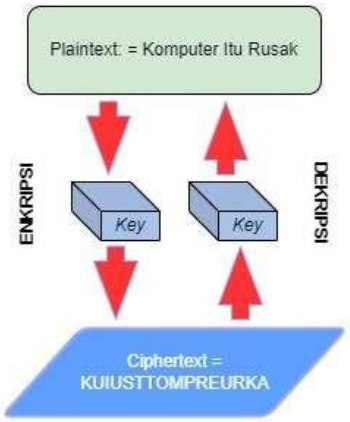
BAB II

TINJAUAN PUSTAKA

2.1.1 Kriptografi

2.1.1 Pengertian Kriptografi

Menurut (Cristy & Riandari, 2021) kriptografi berasal dari bahasa Yunani yaitu crypto yang memiliki arti rahasia dan graphia yang berarti tulisan. Kriptografi juga didefinisikan oleh (Sitepu et al., 2022) bahwa kriptografi adalah ilmu yang mempelajari seni mengamankan pesan atau data dan informasi sehingga pesan dan data tersebut aman untuk dikirimkan ke tujuannya. Selain itu, terdapat definisi lain menurut Munir dalam (Ziliwu et al., 2022) yaitu cabang ilmu yang mempelajari teknik komputasi yang berkaitan dengan masalah keamanan informasi seperti kerahasiaan, integritas data, dan otentikasi. Berdasarkan beberapa definisi tersebut, maka dapat disimpulkan bahwa kriptografi adalah cabang ilmu yang mempelajari beberapa teknik komputasi yang berkaitan dengan aspek keamanan informasi guna menjaga kerahasiaan pesan. Menurut (Azhari et al., 2022) terdiris atas 2 proses yakni enkripsi dan dekripsi. Berikut ini merupakan contoh ilustrasi kriptografi.



Gambar 2.1 Ilustrasi Kriptografi
(Sumber: Nur Wachid Hidayatulloh, 2023)

2.1.2 Tujuan Kriptografi

Ilmu kriptografi tercipta tidak hanya sekedar dalam menjaga keamanan sebuah pesan yang akan dikirimkan. Adanya ilmu kriptografi pada keamanan jaringan juga memiliki tujuan bagi para pengguna. Adapun dasar-dasar tujuan adanya ilmu kriptografi ini yaitu (Ziliwu et al., 2022)

1. Kerahasiaan: Ilmu kriptografi bertujuan agar isi pesan tidak diketahui orang lain selama proses pengiriman.
2. Integritas Data: pesan yang diterima utuh dan dan tidak mengalami modifikasi apapun selama proses pengiriman.
3. Otentikasi: layanan kriptografi mengenai identifikasi terlebih dahulu antara pengirim dan penerima pesan.
4. Anti Penyangkalan: menghindari pihak yang menyampaikan suatu penyangkalan dimana pengirim pesan menyangkal telah mengirim pesan dan sebaliknya penerima pesan tidak mengakui telah menerima pesan tersebut.

2.1.3 Elemen Sistem Kriptografi

Elemen-elemen pada ilmu kriptografi diantaranya (Condro dikutip dari Hidayatulloh, N. W., dkk. 2023):

1. *Plain text*: pesan atau sumber yang pertama dibuat oleh *user* (pengguna); pesan yang dapat dibaca oleh semua orang.
2. *Cipher text*: pesan *plain text* yang telah diubah bentuknya menjadi lebih aman sehingga tidak dapat dibaca.
3. *Cryptographic algorithm*: langkah-langkah yang digunakan berdasarkan operasi matematika untuk mengubah *plain text* menjadi *cipher text*.

4. *Key*: kunci yang digunakan didasarkan pada *cryptographic algorithm* untuk melakukan proses enkripsi dan dekripsi terhadap pesan yang dikirim. Hal ini berarti hanya pengguna yang memiliki kunci yang dapat melakukan dekripsi pesan dalam bentuk *ciphertext*.

2.1.4 Enkripsi dan Dekripi

Enkripsi merupakan Proses yang dilakukan untuk mengamankan pesan (*plain text*) hingga menjadi pesan tersembunyi (Kusumo dikutip dari Hidayatulloh, N. W., dkk. 2023). Selain itu enkripsi juga didefinisikan sebagai keamanan data yang dikirimkan agar terjaga kerahasiaannya (Abdul et al., 2019). Pada sebuah keamanan jaringan, enkripsi sering disebut sebagai *ciphertext* atau kode. *Ciphertext* memiliki definisi yakni sebuah pesan yang tidak dapat dibaca dengan mudah (Kusumo dikutip dari Hidayatulloh, N. W., dkk. 2023).

Dekripsi kebalikan dari proses enkripsi yaitu proses mengubah data yang telah dibuat tidak dapat dibaca melalui enkripsi kembali ke bentuk yang tidak dienkripsi. Data yang dienkripsi atau dikodekan akan dikembalikan ke bentuk aslinya (Novianti dan Hidayat, 2023).

2.2 *Diffie-Hellman Key Exchange* (DHE)

Diffie Hellmen merupakan algoritma pertukaran kunci yang memungkinkan dua pihak yang berkomunikasi melalui jaringan publik untuk dapat membangun kunci bersama yang bersifat rahasia. Kunci bersama tersebut dapat digunakan untuk melakukan enkripsi dan dekripsi dokumen dengan kriptografi simetris. Mekanisme pertukaran kunci dengan Diffie hellman dapat dianalogikan ketika dua pihak yang berkomunikasi katakanlah alice dan bob masing-masing memiliki dokumen yang ingin dibagikan secara rahasia. Untuk melakukan hal tersebut mereka harus

menyepakati sebuah informasi awal (public information) dan selanjutnya informasi tersebut dikombinasikan dengan informasi previledge masingmasing serta dikirimkan melewati jaringan yang tidak aman. Alice dan bob masing-masing menerima informasi yang telah dikirimkan tadi dan selanjutnya dikombinasikan lagi dengan informasi rahasia masing-masing. Hasil akhirnya berupa informasi bersama yang bersifat rahasia (common secret key). Common secret key berfungsi untuk melakukan enkripsi dan dekripsi dokumen sebelum dikirimkan melalui jaringan yang tidak aman (Muhammad Rizka, 2021).

Alice dan Bob berkomunikasi dengan Langkah sebagai berikut. Pertama Alice dan Bob setuju pada 2 buah bilangan prima besar g dan p, sehingga $g < p$. Nilai g dan p tidak dirahasiakan, pihak-pihak tersebut dapat membicarakannya melalui saluran yang tidak aman. Alice menghasilkan bilangan bulat acak xyang besar dan mengirimkan hasil perhitungan berikut ke Bob :

$$A = g^a \ mod \ p. \quad \text{.....} (2.1)$$

Bob membangkitkan bilangan bulat acak yang besar dan mengirim hasil perhitungan ke Alice sebagai berikut:

$$B = g^b mod \ p. \quad \text{.....} (2.2)$$

Alice menghitung

$$s = B^a \ mod \ p. \quad \text{.....} (2.3)$$

Bob menghitung

$$s = A^b \ mod \ p. \quad \text{.....} (2.4)$$

Antara Alice dan Bob mempunyai nilai yang sama karena dibawah modulo p.

Keterangan:

p = Bilangan prima

g = Generator

a = Bilangan acak rahasia dari Alice

b = Bilangan acak rahasia dari Bob

A = Kunci publik milik Alice

B = Kunci Publik milik Bob

s = kunci bersama

Berikut ini pseudocode algoritma DHE untuk mempermudah dalam memahami penggunaan algoritma DHE.

- Parameter publik (disepakati bersama)

  P = bilangan prima besar

  g = bilangan basis (generator)

- Pihak A (Alice)

  pilih a = bilangan acak (kunci privat A)

  hitung A = (g^a mod P) # kunci publik A

- Pihak B (Bob)

  pilih b = bilangan acak (kunci privat B)

  hitung B = (g^b mod P) # kunci publik B

- Pertukaran kunci publik

  Alice mengirim A ke Bob

  Bob mengirim B ke Alice

- Perhitungan shared secret

  Alice menghitung: sA = (B^a mod P)

  Bob menghitung: sB = (A^b mod P)

- Hasil

sA = sB # shared secret yang sama

2.3 *Advance Encryption Standard* (AES)

Pada artikel yang ditulis oleh (Murdowo dikutip dari Hidayatulloh, N. W., dkk. 2023) dan berjudul "Mengenal Proses Perhitungan Enkripsi menggunakan Algoritma Kriptografi *Advance Encryption Standard* (AES) Rijdnael" menjelaskan sejarah dari AES sebagai berikut. "Pada tahun 1997, *National Institute of Standard and Technology* (NIST) mengeluarkan *Advance Encryption Standard* (AES) untuk menggantikan *Data Encryption Standard* (DES). AES dikembangkan dengan tujuan memastikan tata kelola di berbagai bidang. Algoritma AES dirancang untuk menggunakan minimum blok *input* enkripsi 128-bit dan mendukung 3 ukuran kunci yaitu 128- bit, 192-bit, dan 256-bit. Di Agustus 1998, NIST mengumumkan bahwa 15 proposal AES telah diterima dan dievaluasi setelah melalui proses seleksi algoritma yang masuk. Di tahun 1999, NIST mengumumkan bahwa hanya 5 algoritma yang diterima. Algoritma tersebut yaitu RC6, MARS, Snake, Rijdnael dan Twofish. 5 algoritma ini akan menjalankan berbagai pengujian. Di bulan Oktober 2000, Rijndael diumumkan sebagai algoritma pilihan untuk standar AES yang baru."

A. Kelebihan AES

Berdasarkan artikel yang ditulis oleh (Asriyanik dikutip dari Hidayatulloh, N. W., dkk. 2023) menyatakan bahwa terdapat beberapa kelebihan dari AES yang ditemukan, diantaranya:

1. Panjang kunci minimal pada AES adalah 128 bit. Sehingga dengan teknologi yang sekarang, AES tahan terhadap serangan exhaustive key lookup. Dengan panjang kunci 128 bit adalah $2128 \approx 3{,}4 \times 1038$

kemungkinan larangan.

2. Kekuatan AES terletak pada sifat karakteristik bidang GF(28), di mana untuk setiap bilangan prima selalu ada satu bidang unik hingga sehingga semua representasi GF(28) adalah isomorfik dan pemilihan polynomial biner derajat 8.

B. Kelemahan AES

Berdasarkan artikel yang ditulis oleh (Asriyanik dikutip dari Hidayatulloh, N. W., dkk. 2023) menyatakan bahwa terdapat beberapa kelemahan dari AES yang ditemukan, diantaranya:

1. Kesulitan dalam manajemen kunci muncul dengan jenis kunci simetris. Ini karena diperlukan kunci yang berbeda untuk setiap pengiriman dan penerimaan data dengan pengguna yang berbeda.
2. AES merupakan salah satu algoritma kriptografi dengan tipe kunci simetris dalam proses pengiriman dan penerimaan data. Hal ini menyebabkan kunci simetris mudah bocor meski dalam jangka waktu yang lama.

Berikut adalah pseudocode algoritma AES 128-bit:

A. Pseudocode enkripsi algoritma AES 128 bit

Input:

plaintext (16 byte)

key (16 byte)

Langkah-langkah:

1. Expand key menjadi roundKey[0..10] # KeySchedule
2. Inisialisasi state = plaintext
3. Tambahkan round key awal: state = state XOR roundKey[0]

4. Ulangi untuk round = 1 sampai 9:
    a. SubBytes(state) (substitusi tiap byte dengan S-Box)
    b. ShiftRows(state) (geser baris ke kiri (0,1,2,3 kali))
    c. MixColumns(state) (campur kolom dengan matriks tetap)
    d. AddRoundKey (state, roundKey[round])
5. Round terakhir (round = 10):
    a. SubBytes(state)
    b. ShiftRows(state)
    c. AddRoundKey(state, roundKey[10])
6. Ciphertext = state

B. Pseudocode dekripsi algoritma AES 128 bit

Input:

ciphertext (16 byte)

key (16 byte)

1. Expand key menjadi roundKey[0..10]
2. Inisialisasi state = ciphertext
3. Tambahkan round key terakhir:

    state = state XOR roundKey[10]

4. Ulangi untuk round = 9 turun ke 1:
    a. InvShiftRows(state)
    b. InvSubBytes(state)
    c. AddRoundKey(state, roundKey[round])
    d. InvMixColumns(state)
5. Round terakhir:

a. InvShiftRows(state)

b. InvSubBytes(state)

c. AddRoundKey(state, roundKey[0])

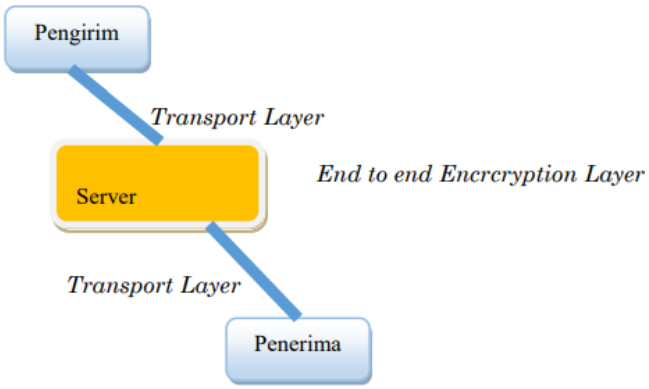
Plaintext = state

2.4 Pesan *End-To-End*

Enkripsi end-to-end adalah proses pertukaran data yang aman dari pengirim ke penerima, dengan mengenkripsi seluruh konten selama transmisi sehingga tidak dapat diakses atau dimodifikasi oleh pihak ketiga. Beberapa algoritma kriptografi digunakan untuk tujuan ini. Komponen-komponen seperti identitas pengguna, protokol pertukaran kunci, dan implementasi yang aman bekerja bersama-sama untuk memberikan keamanan yang terbaik kepada pengguna akhir (Blaise, Awodele, & Yewande, 2021).

Dalam konteks aplikasi pesan instan, keamanan dan privasi menjadi sangat penting (Ali & Alsaad, 2020). Jaringan sosial online dan aplikasi pesan instan seperti WhatsApp, Telegram, dan Facebook Messenger telah mengadopsi enkripsi *end-to-end* untuk melindungi komunikasi pengguna (S., 2022). Namun, dalam penggunaan sehari-hari, masih terdapat beberapa masalah keamanan dan privasi yang perlu diatasi guna melindungi informasi pribadi pengguna dan data yang dibagikan melalui aplikasi pesan instan ini (Ali & Alsaad, 2020). Dalam mengimplementasikan enkripsi end-to-end dalam aplikasi pesan instan, penting untuk mempertimbangkan potensi dan batasan yang ada. Meskipun enkripsi *end-to-end* memberikan perlindungan yang kuat terhadap kerahasiaan pesan dan privasi pengguna, aspek-aspek lain seperti keamanan perangkat, pengelolaan kunci enkripsi, dan verifikasi identitas penerima pesan juga penting untuk

diperhatikan. Selain itu, pengguna aplikasi pesan instan juga harus memperhatikan kebijakan privasi dan pernyataan keamanan dari penyedia layanan, serta terus memperbarui aplikasi mereka untuk mengatasi kerentanan keamanan yang mungkin muncul (Blaise, Awodele, & Yewande, 2021). Dengan memahami potensi dan batasan enkripsi *end-to-end*, serta melibatkan pengguna, pengembang, dan penyedia layanan, dapat dikembangkan teknologi keamanan dan privasi yang lebih kuat dan efektif di masa depan.



Gambar 2.2 Konsep Dasar Enkripsi *End-To-End*
(Sumber: Gellysa Urva, 2017)

2.5 *Flowchart*

Flowchart atau bagan alir merupakan diagram yang digunakan peracangan sebuah sistem atau alur kerja terhadap sistem yang telah dibuat agar dapat dengan mudah untuk dipahami (Angraina Fitri & Putri, 2022).

Flowchart adalah suatu peran penting yang digunakan untuk menggambarkan proses data pada sebuah sistem yang akan dilakukan suatu program melalui perancangan dalam bentuk diagram dengan modal simbol-simbol *flowchart* (Kurniawan & Mumtahana, 2021).

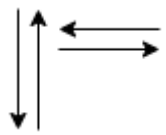
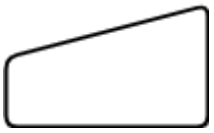
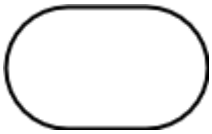
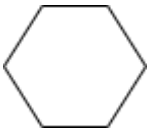
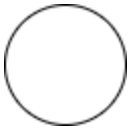
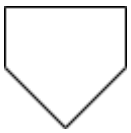
Flowchart merupakan diagram yang secara berurutan mempresentasikan algoritma dan logika pemrograman dari sistem yang dibuat seperti pengulangan dan

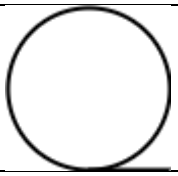
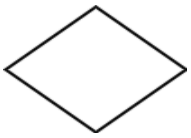
kondisi. *Flowchart* mempermudah untuk memahami bagaimana sebuah sistem bekerja, baik untuk *programmer*, maupun *stakeholder* (Nuroh, 2022).

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek (Yasa dan Rahayu, 2022:20).

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. *Flowchart* menolong analisis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian (Sutanti, dkk, 2020:2).

Tabel 2.1 Table Bentuk dan Simbol *Flowchart*

	Garis Penghubung atar simbol		Simbol manual input
	Simbol terminator		Simbol untuk mempersiapkan penyimpanan
	Simbol Koneksi keluar – masuk pada halaman yang sama		Simbol untuk pelaksanaan suatu bagian
	Simbol Koneksi Keluar – masuk pada halaman berbeda		Simbol display
	Simbol Proses		Simbol yang menyatakan input berasal atau disimpan

			ke disk
	Simbol Manual Operasi		Simbol magnetik tape unit
	Simbol Desicion/Pilihan		Simbol punch card
	Simbol Input-Output (masukan- keluaran)		Simbol dokumen