PENERAPAN SISTEM KEAMANAN DATA MENGGUNAKAN ALGORITMA VIGENERE CIPHER PADA KOMUNIKASI NIRKABEL ATMEGA328

TUGAS AKHIR

Diajukan Untuk Melengkapi dan Memenuhi Persyaratan Guna Menyelesaikan Program Strata Satu (S1) pada Program Studi Teknik Komputer

Oleh

DICKY WAHYUDI
2014050034

PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNIK UNIVERSITAS SERAMBI MEKKAH
BANDA ACEH
2024

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS SERAMBI MEKKAH
FAKULTAS TEKNIK
Jalan Tgk. Imum Lueng Bata, Batoh, Banda Aceh, Kode Pos 23245
Website: www.serambimekkah.ac.id, Surel: akademik@serambimekkah.ac.id

LEMBAR PENGESAHAN

TUGAS AKHIR

PENERAPAN SISTEM KEAMANAN DATA MENGGUNAKAN ALGORITMA VIGENERE CIPHER PADA KOMUIKASI NIRKABEL ATMEGA328

Oleh:

Nama : Dicky Wahyudi

NPM : 2014050034

Program Studi : Teknik Komputer

Menyetujui:

Dosen Pembimbing I	Desen Pembimbing II
(Yeni Yanti, S.T., M.T.)	**(Taufik Hidayat, S.T., M.T.)**
NIDN. 0111048601	**NIDN. 1311078701**

Mengetahui:

Ketua Program Studi
Teknik Komputer

(Munawir, S.T., M.T.)
NIDN. 1324078901

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET DAN TEKNOLOGI
UNIVERSITAS SERAMBI MEKKAH
FAKULTAS TEKNIK
Jalan Tgk. Imum Lueng Bata, Batoh, Banda Aceh, Kode Pos 23245
Website: www.serambimekkah.ac.id, Surel: akademik@serambimekkah.ac.id

HALAMAN PENGESAHAN

TUGAS AKHIR

PENERAPAN SISTEM KEAMANAN DATA MENGGUNAKAN ALGORITMA VIGENERE CIPHER PADA KOMUNIKASI NIRKABEL ATMEGA328

Oleh:

Nama : Dicky Wahyudi
NPM : 2014050034
Program Studi : Teknik Komputer

Telah Disidangkan pada Tanggal 17 Juli 2024 dan Dinyatakan Telah LULUS.

Menyetujui,

Pembimbing I	: Yeni Yanti, S.T., M.T	(.....)
Pembimbing II	: Taufik Hidayat, S.T., M.T	(.....)
Penguji I	: Dr. Ir. Erdiwansyah, S.T., M.T., IPM	(.....)
Penguji II	: Munawir, S.T., M.T	(.....)

Mengetahui,

Dekan Fakultas Teknik
Universitas Serambi Mekkah

(Dr. Ir. Elvitriana, M.Eng)
NIDN. 0129016601

Ketua Program Studi
Teknik Komputer

(Munawir, S.T., M.T)
NIDN. 1324078901

KATA PENGANTAR

Alhamdulillaahirabbil'alamin, Segala puji dan syukur senantiasa penulis panjatkan kehadirat Allah SWT yang telah memberikan nikmat, rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini. Shalawat serta salam tercurahkan kepada Rasulullah SAW, keluarga dan sahabatnya.

Selama menyelesaikan tugas akhir ini penulis banyak memperoleh bimbingan, motivasi, arahan, dan dukungan baik secara materil maupun spiritual, oleh karena itu penulis pada kesempatan ini ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Orang tua dan seluruh keluarga yang senantiasa memberikan cinta, dorongan, Doa yang telah dipanjatkan tiada henti-hentinya, bimbingan, motivasi, serta kasih sayang terhadap penulis tanpa mengharapkan balasan.
2. Bapak Dr. Teuku Abdurahman, SH., SpN selaku Rektor Universitas Serambi Mekkah Banda Aceh.
3. Dr. Ir. Elvitriana, M.Eng selaku Dekan Fakultas Teknik Universitas Serambi Mekkah.
4. Bapak Munawir, S.T., M.T selaku Ketua Prodi Teknik Komputer di Universitas Serambi Mekkah.
5. Ibu Yeni Yanti, S.T., M.T Selaku Pembimbing I yang telah menyediakan waktu dan membimbing Saya dalam proses penyelesain pembuatan tugas akhir ini.
6. Bapak Taufik Hidayat, S.T., M.T Selaku Pembimbing II yang telah menyediakan waktu dan membimbing Saya selama proses penyelesaian pembuatan tugas akhir ini.
7. Bapak dan Ibu Dosen Fakultas Teknik yang telah memberikan ilmunya kepada penulis, semoga Bapak dan Ibu Dosen selalu dalam rahmat dan lindungan Allah SWT. Sehingga ilmu yang telah diberikan dapat bermanfaat dikemudian hari.
8. Teman-teman mahasiswa Teknik khususnya teman-teman seperjuangan Jurusan Teknik Komputer 2020, yang selalu memberikan support kepada penulis dalam menyusun tugas akhir ini sampai selesai.

Semoga segala bantuan yang telah diberikan kepada penulis mendapat balasan yang setimpal dari Allah SWT. Penulis menyadari masih banyak kekurangan dalam penyusunan tugas akhir ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan tugas akhir ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga tugas akhir ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Banda Aceh / 17 Juli 2024

Penyusun

Dicky Wahyudi

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Internet of Things (IoT) memainkan peran penting dalam menjembatani sistem komunikasi, memfasilitasi pertukaran informasi, komunikasi interpersonal, bisnis, pendidikan, dan banyak aspek kehidupan sehari-hari. *Internet of Things (IoT)* telah mengalami transformasi yang signifikan selama beberapa dekade terakhir. Hal ini disebabkan oleh peningkatan penggunaan internet, komunikasi digital, dan pertukaran data yang semakin luas.

Penggunaan teknologi IoT tidak luput dari penyalahgunaan, seperti penyalahgunaan untuk memanfaatkan perangkat IoT untuk tindak kejahatan. Penyalahgunaan tersebut justru dapat membuat perangkat IoT beralih menjadi suatu hal yang dapat merusak. Akibat yang ditimbulkan dari penyalahgunaan tersebut tergantung pada bidang perangkat IoT-nya. Selain itu, permasalahan yang terjadi pada tingkat keamanan yang tidak memenuhi standar keamanan Karena perangkat IoT saling berkomunikasi dan bertukar data menggunakan jaringan internet, rentan terjadi serangan pada komunikasi jaringan adalah *Man-In-The-Middle (MITM).* Oleh karena itu dalam membangun IoT dibutuhkan sistem pengaman yang dapat melindungi dari ancaman-ancaman yang terjadi. Pengamanan sistem di IoT salah satunya mengankan keamanan data, yang juga meliputi berbagai elemen. Informasi pada sensor, sistem transmisi dan pengolah data tidak boleh di rusak, dicuri maupun dipalsukan.

Salah satu cara efektif dalam pengamanan sistem IoT adalah dengan implementasi kriptografi. Kriptografi merupakan perlindungan terhadap informasi dan komunikasi melalui penggunaan kode sehingga hanya dapat dibaca dan diproses oleh orang yang berwenang (Bhandari & Kirubanand, 2019). Pada kriptografi, akan dilakukan proses enkripsi untuk merubah informasi menjadi bentuk yang tidak bisa dibaca dengan fungsi tertentu. Menurut dokumen IEC (*International Electrotechnical Commision*) nomor 62591, enkripsi data merupakan unsur penting pada *Internet of Things* (Roman et al., 2017).

Selain aspek keamanan, keterbatasan kemampuan komputasi juga merupakan isu yang harus diperhatikan pada perangkat IoT. Perangkat IoT biasanya memiliki kemampuan pemrosesan yang terbatas (Skirelis & Navakauskas, 2017), terdapat peningkatan kemampuan tetapi masih tidak dapat bersaing dengan komputasi *desktop* (Babar & Sohail Khan, 2021). Terbatasnya energi dan kemampuan komputasi merupakan tantangan besar pada IoT (Wei et al., 2020). Dalam melakukan komputasi pada perangkat IoT, perlu memperhatikan kemampuan komputasi sehingga didapatkan eksekusi yang efektif dan efisien.

Agar dapat memberikan keamanan pada sistem IoT maka diperlukan kriptografi untuk mengacak informasi yang diproses sehingga tidak bisa dibaca oleh selain penerima yang berhak. Pemilihan algoritma enkripsi juga perlu disesuaikan dengan kemampuan komputasi yang terbatas pada perangkat IoT. Beberapa penelitian yang diperoleh dari sumber – sumber penelitian jurnal terkait seperti: Asang & sembiring (2017), dilakukan implementasi algoritma *Rivest Shamir Adleman* (RSA) pada perangkat Arduino. Algoritma dapat diimplemetasikan tetapi dengan konsumsi memori yang tidak sedikit. Penelitian yang dilakukan Fernando & Lukas (2017), membandingkan algoritma *Data Encryption Standard Lightweight* (DESL) dan *Data Encryption Standard* (DES) dengan enkripsi kunci RSA, dimana didapatkan hasil bahwa perbedaan waktu komputasi tidak terlalu signifikan. Penelitian yang mencoba mengimplementasikan kriptografi pada sistem IoT juga dilakukan oleh Endrayanto et al (2019), yang menguji penggunaan algoritma AES128 dengan hasil waktu eksekusi yang relatif lebih cepat dan penggunaan memori yang relatif kecil. Penelitian ini menggunakan modul IoT *particle photon*.

Pada tugas akhir ini, penulis ingin membuat suatu rancangan sistem yaitu "Penerapan Sistem Keamanan Data menggunakan Algoritma Vigenere Cipher pada Mikrokontroler Arduino Uno ATmega328". Penulis melakukan penelitian untuk lebih menyerhanakan komputasi algoritma penelitian terdahulu. sehingga penulis menggunakan algoritma *vigenere cipher* yang mana tingkat kompleksitas komputasi yang rendah.

Penulis berharap penelitian ini dapat membantu dan mempermudah keamanan data lebih serdehana, sehingga tercipta suatu perangkat yang lebih efesien dan keamanannya terjaga. Adapuan untuk perangkat yang digunakan yaitu mikrokontroler Arduino Uno. Untuk melihat keakuratan sistem algoritma *vigenere cipher* yang dibuat, dilakukan kalkulasi matematika algoritma vigenere cipher.

1.2 Rumusan Masalah

Dalam pelaksanaan penelitian tugas akhir ini terdapat beberapa permasalahan yang menjadi titik utama pembahasan, diantaranya adalah sebagai berikut:

1. Bagaimana cara agar dapat memastikan bahwa data teks pada perangkat mikrokontroler arduino uno *ATmega328* yang dienkripsi kemudian didekripsi tidak mengalami perubahan.
2. Bagaimana hasil pengamanan data teks pada perangkat mikrokontroler arduino uno *ATmega328* berdasarkan kalkulasi menggunakan algoritma *vigenere cipher*.

1.3 Batasan Masalah

1. Hanya melakukan pengujian enkripsi dan deskripsi data teks pada perangkat mikrokontroler arduino uno *ATmega328* yang dienkripsi kemudian didekripsi tidak mengalami perubahan.
2. Melakukan Analisa hasil enkripsi dan dekripsi data teks pada perangkat mikrokontroler arduino uno *ATmega328* berdasarkan kalkulasi menggunakan algoritma *vigenere cipher*.

1.4 Tujuan

1. Menghasilkan enkripsi *vigenere cipher* yang dapat memastikan data teks pada perangkat mikrokontroler arduino uno *ATmega328* yang dienkripsi kemudian didekripsi tidak mengalami perubahan.
2. Menghasilkan sistem keamanan data teks pada perangkat mikrokontroler arduino uno *ATmega328* berdasarkan kalkulasi menggunakan algoritma *vigenere cipher*.

BAB II
LANDASAN TEORI

2.1 Jaringan Komputer

Menurut Badrul dan Akmaludin (2019), menyimpulkan bahwa jaringan komputer adalah kumpulan beberapa komputer dan perangkat lain seperti *router*, *switch*, dan sebagainya yang saling terhubung satu sama lain melalui media perantara.

Menurut Lestari dan Permana (2019), jaringan komputer merupakan sebuah sistem yang terdiri dari sekelompok komputer otonom yang saling terkoneksi satu dengan yang lainnya menggunakan protokol komunikasi melalui media komunikasi untuk dapat saling berbagi informasi, jalur yang digunakan dalam komunikasi jaringan dapat berbentuk media kabel (*Wired Network*), maupun non kabel (*Wireless Network*). Jaringan komputer yang dibentuk menggunakan media kabel, dapat memanfaatkan berbagai jenis kabel, diantara kabel yang sering digunakan dalam membentuk sebuah jaringan adalah kabel coaxial, kabel UTP, kabel STP dan kabel *Fiber Optic*.

Berdasarkan teori di atas dapat disimpulkan bahwa jaringan komputer adalah sebuah kumpulan komputer, *printer* dan peralatan lainnya yang terhubung dalam satu kesatuan. Jaringan dan konektor yang digunakan adalah konektor tipe DIX. Panjang kabel transceiver maksimum 50 m, panjang kabel *Thick Ethernet* maksimum 500 m dengan maksimum 100 transceiver terhubungi. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan *hardware/software* yang terhubung dengan jaringan. Setiap komputer, printer atau *periferal* yang terhubung dengan jaringan disebut node.

Tujuan dari jaringan komputer adalah untuk melakukan komunikasi data, sharing data maupun pemakaian *resource* bersama seperti *printer* dan media penyimpanan sekunder. Komunikasi data sendiri memiliki tujuan yang lebih khusus, yaitu:

1. Memungkinkan pengiriman data dalam jumlah besar efisien, ekonomis, dan tanpa kesalahan dari suatu tempat ke tempat yang lain.
2. Memungkinkan penggunaan sistem komputer dan peralatan pendukung dari jarak jauh (*remote*).
3. Memungkinkan penggunaan komputer secara terpusat maupun secara tersebar sehingga mendukung manajemen dalam hal kontrol, baik desentralisasi ataupun sentralisasi.
4. Mempermudah kemungkinan pengelolaan dan pengaturan data yang ada dalam berbagai macam sistem komputer.
5. Mengurangi waktu untuk pengelolaan data.
6. Mendapatkan data langsung dari sumbernya.
7. Mempercepat penyebarluasan informasi.
8. Komunikasi data berkaitan dengan pertukaran data diantara dua perangkat yang terhubung secara langsung yang memungkinkan adanya pertukaran data antar kedua pihak.

2.2 Perangkat IOT

Internet berkembang jauh lebih pesat dibandingkan dengan teknologi lain. Bermula dari hanya beberapa komputer yang terhubung satu dengan yang lain hingga saat ini internet dapat diakses oleh hampir semua orang di dunia dan telah menjadi bagian penting dalam kehidupan manusia. Internet dapat menghubungkan kita, peralatan, perangkat lunak, mesin, dan hal-hal di sekitar kita. Rancangan jaringan ini disebut IoT.

Gambar 2.1 Ilustrasi IoT

IoT didefinisikan objek-objek cerdas yang memiliki kemampuan untuk mengatur objek lain yang ada di dalam satu jaringan dengan otomatis, berbagi informasi, data, dan sumber daya dengan objek lain bereaksi dan bertindak dalam situasi dan perubahan wajah di lingkungan (Madakam, 2015).

Ide awal *Internet of Things* pertama kali dimunculkan oleh Kevin Ashton pada tahun 1999, dimana benda-benda di sekitar kita dapat berkomunikasi antara satu sama lain melalui sebuah jaringan seperti internet.

Berawal dari *Auto-ID Center*, teknologi yang berbasis pada Radio Frequency Identification (RFID). RFID merupakan identifikasi kode produk elektronik yang bersifat unik ini kemudian berkembang menjadi teknologi bahwa pada setiap benda dapat memiliki alamat internet (Tashia, 2015).

Dengan adanya IoT dapat memudahkan manusia dalam bekerja di berbagai bidang. Dalam hal keamanan manusia dapat dengan mudah memeriksa kondisi tempat tinggalnya melalui *Closed-circuit Television* (CCTV) yang terkoneksi dengan smartphone. Dalam hal industri seorang peternak dapat memeriksa kondisi kelembapan, suhu dan berapa banyak pakan ternaknya melalui komputernya. Dalam hal transportasi, polisi dapat melihat kondisi jalan yang sedang mengalami kemacetan dan kondisi jalan yang lenggang.

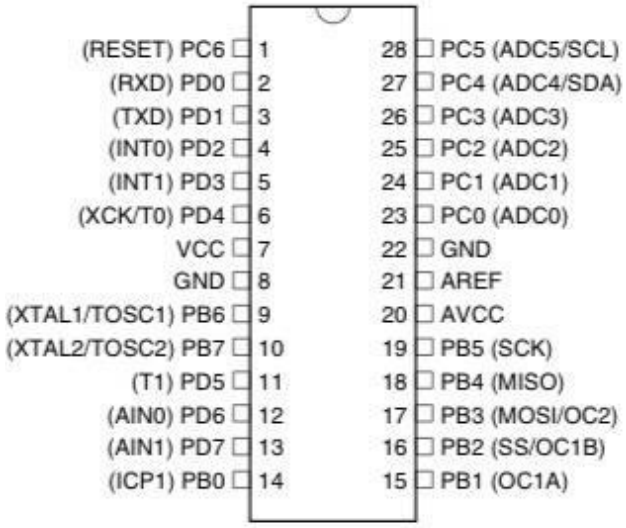
2.3 Mikrokontroler ATmega328

Dharma et al., (2019), *ATmega328* adalah mikrokontroler keluaran dari atmel yang mempunyai arsitektur *Reduce Instruction Set Computer (RISC)* dimana setiap proses eksekusi data lebih cepat dari pada arsitektur *Completed Instruction Set Computer (CISC).* Mikrokontroller ini memiliki beberapa fitur antara lain.

a. 130 macam instruksi yang hampir dieksekusi dalam satu siklus clock.
b. 32 x 8-bit *register* serba guna.
c. Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz.
d. 32 KB Flash memory dan pada arduino memiliki *bootloader* yang menggunakan 2 KB dari flash memori sebagai *bootloader*.

e. Memiliki *EEPROM (Electrically Erasable Programmable Read Only Memory)* sebesar 1KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
f. Memiliki *SRAM (Static Random Access Memory)* sebesar 2KB.
g. Memiliki pin I/O digital sebanyak 14 pin 6 diantaranya *PWM (Pulse Width Modulation) output*.
h. Master / Slave SPI Serial *interface*. ATmega328 yang terdapat dalam mikrokontroler arduino uno memiliki bentuk fisik seperti yang terdapat pada gambar.

Gambar 2.2 Bentuk fisik atmega328



Gambar 2.3 Konfigurasi pin atmega328

2.4 Arduino uno

Dharma, I. P. L., Tansa, S., & Nasibu, I. Z. (2019), Arduino merupakan sebuah board minimum system mikrokontroller yang bersifat *open source*. Bentuk fisik arduino terdapat pada Gambar 2. Didalam rangkaian board arduino terdapat

mikrokontroller AVR seri 328 yang merupakan produk dari Atmel. Arduino memiliki kelebihan tersendiri dibanding board mikrokontroller yang lain selain bersifat *open source,* arduino juga mempunyai bahasa programnya sendiri berupa bahasa C. selain itu dalam *board* arduino sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan ketika diprogram mikrokontroller didalam arduino.

Gambar 2.4 Arduino uno

2.5 Kabel Jumper

Kabel jumper adalah kabel elektrik untuk menghubungkan antar komponen di *breadboard* tanpa memerlukan solder. Kabel jumper umumnya memiliki *connector* atau pin di masing-masing ujungnya. *Connector* untuk menusuk disebut male connector, dan connector untuk ditusuk disebut *female connector*. Kabel jumper dibagi menjadi 3 yaitu: *Male to Male, Male to Female* dan *Female to Female* (Fathulrohman et al., 2019).

Gambar 2.5 Kabel jumper

2.6 Komunikasi Data

Noris et al., (2020), Komunikasi data adalah pertukaran informasi antara dua pihak yang bertindak sebagai pengirim dan penerima pada waktu yang bersamaan. Informasi yang dikirimkan berupa data digital, dapat berupa teks, audio maupun video. Komunikasi data dibedakan menjadi dua jenis, yaitu komunikasi darat dan komunikasi satelit. Di darat, komunikasi data didukung oleh kabel dan media nirkabel, sedangkan satelit menggunakan satelit sebagai jalur akses. Komponen yang ada dalam sitem komunikasi data, diantaranya:

1) Sumber (*source*)

   Alat / komponen yang menghasilkan data atau informasi yang akan dikirimkan dapat menjadi perangkat input pada komputer. Alat ini bisa mengubah informasi audio, video atau teks menjadi unit data untuk selanjutnya diproses pada sistem komputer.

2) Pengirim (*transmitter*)

   Suatu alat untuk mengolah data atau informasi sumber sehingga dapat ditransmisikan melalui suatu sistem / media transmisi. Bentuk fisik dapat berupa bentuk komputer pribadi yang dapat mengolah segala bentuk informasi, atau secara khusus dapat berupa telepon yang digunakan untuk berkomunikasi dengan informasi berupa suara (*voice*).

3) Sistem Transmisi

   Dalam bentuk saluran transmisi tunggal atau jaringan kompleks yang menghubungkan sistem sumber ke sistem target, dapat berupa media kabel maupun media nirkabel.

   Jalur transmisi tunggal mengacu pada koneksi antara perangkat pengirim dan penerima dari jenis media di segmen jaringan, dan jalur kompleks berarti perangkat pengirim dan penerima dihubungkan oleh satu sistem.

4) Penerima (*receiver*)

   Suatu alat yang digunakan untuk menerima sinyal dari suatu sistem transmisi dan mengolahnya menjadi informasi yang dapat ditangkap / diolah oleh target.

5) Tujuan (*destination*)

Tangkap informasi yang dihasilkan oleh penerima, dan kemudian ubah informasi yang diterima oleh perangkat target menjadi jenis informasi yang sama dengan informasi yang dikirim, dan gunakan perangkat lunak aplikasi untuk menjembatani informasi ini ke pengguna jaringan komputer.

2.7 Keamanan Jaringan

Ri2M (2010), Keamanan jaringan dapat digambarkan secara umum yaitu apabila komputer yang terhubung dengan jaringan yang lebih banyak mempunyai ancaman keamanan dari pada komputer yang tidak terhubung ke mana-mana. Namun dengan adanya pengendalian maka resiko yang tidak diinginkan dapat dikurangi. Adanya keamanan jaringan maka para pemakai berharap bahwa pesan yang dikirim dapat sampai dengan baik ke tempat yang dituju tanpa mengalami adanya kecacatan yang diterima oleh si penerima, misalnya saja adanya perubahan pesan. Di dalam keamanan jaringan terdapat pula resiko jaringan komputer yang merupakan segala bentuk ancaman bauk fisik maupun *logic* yang langsung atau tidak langsung mengganggu kegiatan yang sedang berlangsung dalam jaringan. Resiko dalam jaringan komputer disebabkan oleh beberapa faktor yaitu:

1. Kelemahan manusia.
2. Kelemahan perangkat keras komputer.
3. Kelemahan sistem operasi jaringan.
4. Kelemahan sistem jaringan komnikasi.

Keamanan jaringan juga mempunyai tujuan yang dapat membuat kemanan jaringan lebih ditingkatkan lagi, yaitu:

1. *Confidentiality*: Adanya data - data yang paling penting yang biasanya tidak boleh di akses oleh seseorang, maka dilakukan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Biasanya *confidentiality* ini berhubungan dengan informasi yang diberikan ke pihak lain.
2. *Integrity*: Bahwa pesan yang disampaikan tetap orisinil yang tidak diragukan keaslianya, tidak dimodifikasi selama dalam perjalanan dari sumber ke penerimanya.

3. *Avaibility*: Dimana *user* yang mempunyai hak akses diberi akses tepat pada waktunya, biasanya ini berhubungan dengan ketersediaan informasi atau data ketika dibutuhkan. Apabila sistem informasi ini diserang maka akan menghambat bahkan menyebabkan tidak dapat mengakses informasi tersebut.

Tujuan keamanan jaringan dapat dicapai dengan suatu metode keamanan jaringan yang dapat melindungi sistem baik dari dalam maupun dari luar jaringan, namun bukan hanya melindungi tetapi harus dapat bertindak apabila terjadi serangan yang ada di dalam jaringan. Namun dibutuhkan juga suatu pemahaman tentang menentukan kebijakan keamanan (*security policy*) dalam keamanan jaringan. Jika ingin menentukan apa saja yang harus di lindungi maka harus mempunyai perencanaan keamanan yang matang dan baik.

2.7.1 Mengenali Ancaman Terhadap Network Security

Langkah awal dalam mengembangkan rencana *network security* yang efektif adalah dengan mengenali ancaman yang mungkin datang.

a. Akses tidak sah, oleh orang yang tidak mempunyai wewenang.
b. Penolakan terhadap *service*, segala masalah mengenai *security* yang menyebabkan sistem mengganggu pekerjaan-pekerjaan yang produktif.
c. Kesalahan informasi, segala masalah yang dapat menyebabkan diberikanya informasi yang penting atau sesitif kepada orang yang salah, yang seharusnya tidak boleh mendapatkan informasi tersebut.

2.8 Kriptografi

Zamara, S. (2019), Kriptografi pada awalnya di jabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu bersandarkan pada Teknik matematikan untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasientitas. Jadi pengertian kriptografi adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan Teknik yang menyediakan keamanan informasi. Berikut ini adalah rangkuman beberapa mekanisme yang berkembang pada kriptografi:

1. Fungsi hash. Fungsi hash adalah fungsi yang melakukan pemetaan pesan dengan panjang sembarang dengan sebuah teks khusus yang disebut *message* digest dengan panjang tetap. Fungsi hash umumnya dipakai sebagai nilai uji (*check value*) pada mekanisme keutuhan data.
2. Penyandian dengan kunci simetrik (*symmetric key enchipherment*). Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai sama. Kunci pada penyandian simetrik diasumsikan bersifat rahasia hanya pihak yang melakukan enkripsi dan dekripsi yang mengetahui nilainya. Oleh karena itu penyandian dengan kunci simetrik disebut juga penyandian dengan kunci rahasia *secret key encipherment*.
3. Penyandian dengan kunci asimetrik (*asymmetric key enchipherment*). Penyandian dengan kunci asimetrik atau sering juga disebut dengan penyandian kunci publik (*public key*) adalah penyandian dengan kunci enkripsi dan deskripsi berbeda nilai. Kunci enkripsi yang juga disebut dengan kunci publik (*public key*) bersifat terbuka. Sedangkan, kunci dekripsi yang juga disebut kunci privat (*private key*) bersifat rahasia.

2.8.1 Pesan, Plaintext, Ciphertext

Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks (*plaintext*) atau teks-jelas (cleartext). Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan didalam media perekaman (kertas, *storage*, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi juga dapat berbentuk citra (*image*), suara/bunyi (*audio*), dan video, atau berkas biner lainnya. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*). Cipherteks harus dapat ditranformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca.

2.8.2 Vigenere Cipher

Vigenere cipher adalah salah satu algoritma kriptografi klasik yang di perkenalkan pada abad 16 pada tahun 1986. Algoritma kriptografi ini dipublikasikan oleh seorang diplomat dan juga kriptologis yang brasal dari prancis, yaitu *blaise de vigenere*, namun sebenarnya algoritma ini telah digambarkan sebelumnya pada buku La Cifra Del Sig Giovan Batista Belaso, pada tahun 1553. Cara kerja dari *vigenere cipher* ini mirip dengan caesar cipher, yaitumengenkripsi plaintext pada pesan dengan cara mengeser huruf pada pesan tersebut sejauh pada kunci nilai alphabet. *Vigenere cipher* adalah salah satu algoritma kriptografi klasik yang menggunakan metode subsitusi abjad majemuk. Subsitusi abjad tunggal yang semua huruf disuatu pesan di enkripsi menggunakan kunci yang sama, berikut proses model matematika dari enkripsi pada algoritma *vigenere cipher* ini:

$$Ci = Ek (Mi) + Ki) \bmod 26$$

Dan model matematika untuk deskripsinya adalah:

$$Mi = Dk (Ci) = (Ci – Ki) \bmod 26$$

Dengan c memodelkan ciphertext, M memodelkan plaintext, dan K memodelkan kunci, huruf pada kunci akan di konversi menjadi sebuah nilai, misalnya A=0, B=1, sampai dengan Z=25. Setelah itu prosesnya sama seperti pada caesar cipher dimana setiap huruf pada plaintext akan digeser sejauh nilai kunci yang posisinya bersesuaian. Adapun langkah-langkah enkripsi *vigenere cipher* adalah:

1. Tentukan plaintext yang akan dienkripsi besertakunci.
2. Jika panjang kunci tidak sama dengan panjang plaintext maka kunci yang ada diulang secarapriodik sehingga jumlah karakter kuncinya sama dengan jumlah plaintext nya.
3. Selanjutnya ubah plaintext ke bentuk nilaidesimal kemudian ditambahkan dengan kunci. Jika penambahan lebih besar dari jumlah mod, maka diambil nilai sisa hasil bagi nya.
4. Setelah dijumlahkan dengan kunci maka langkah berikutnya adalah mengubah kembali ke bentuk karakter.

Contoh perhitungannya, plaintext yang digunakan “KONSTATINOPEL” dengan kunci “ROMA”.

- Proses Enkripsi

  Diketahui: Pi = KONSTATINOPEL

  Ki = ROMA

Tabel 2.1 Konversi Nilai plainteks Algoritma Vigenere Cipher

K	O	N	S	T	A	T	I	N	O	P	E	L
10	14	13	18	19	0	19	8	13	14	15	4	11
R	O	M	A	R	O	M	A	R	O	M	A	R
17	14	12	0	17	14	12	0	17	14	12	0	17

C1 = K + R mod 26

= 10 + 17 mod 26

= 1 = B

C2 = O + O mod 26

= 14 + 14 mod 26

= 2 = C

C3 = N + M mod 26

= 13 + 12 mod 26

= 25 = Z

C4 = S + A mod 26

= 18 + 0 mod 26

= 18 = S

C5 = T + R mod 26

= 19 + 17 mod 26

= 10 = K

C6 = A + O mod 26

= 0 + 14 mod 26

= 14 = O

C7 = T + M mod 26

= 19 + 12 mod 26

= 5 = F

C8 = I + A mod 26

= 8 + 0 mod 26

= 8 = I

C9 = N + R mod 26

= 13 + 17 mod 26

= 4 = E

C10 = O + O mod 26

= 14 + 14 mod 26

= 2 = C

C11 = P + M mod 26

= 15 + 12 mod 26

= 1 = B

C12 = E + A mod 26

= 4 + 0 mod 26

= 4 = E

C13 = L + R mod 26

= 11 + 17 mod 26

= 2 = C

Setelah proses enkripsi maka didapatlah hasil menjadi: "BCZSKOFIECBEC".

- Proses Dekripsi

Diketahui: Ci = BCZSKOFIECBEC

Ki = ROMA

Tabel 2.2 Konversi nilai Cipherteks Algoritma Vigenere Cipher

B	C	Z	S	K	O	F	I	E	C	B	E	C
1	2	25	18	10	14	5	8	4	2	1	4	2
R	O	M	A	R	O	M	A	R	O	M	A	R
17	14	12	0	17	14	12	0	17	14	12	0	17

P1 = B – R mod 26
= 1 – 17 mod 26
= 10 = K
P2 = C – O mod 26
= 2 – 14 mod 26
= 14 = O
P3 = Z – M mod 26
= 25 – 12 mod 26
= 13 = N
P4 = S – A mod 26
= 18 – 0 mod 26
= 18 = S
P5 = K – R mod 26
= 10 – 17 mod 26
= 19 = T
P6 = O – O mod 26
= 14 – 14 mod 26
= 0 = A
P7 = F – M mod 26
= 5 – 12 mod 26
= 19 = T
P8 = I – A mod 26
= 8 – 0 mod 26
= 8 = I
P9 = E – R mod 26
= 4 – 17 mod 26
= 13 = N
P10 = C – O mod 26
= 2 – 14 mod 26
= 14 = O

P11 = B – M mod 26
= 1 – 12 mod 26
= 15 = P
P12 = E – A mod 26
= 4 – 0 mod 26
= 4 = E
P13 = C – R mod 26
= 2 – 17 mod 26
= 11 = L

Setelah proses dekripsi maka didapatlah hasil menjadi: "KONSTATINOPEL".

2.8.3 Caesar Cipher

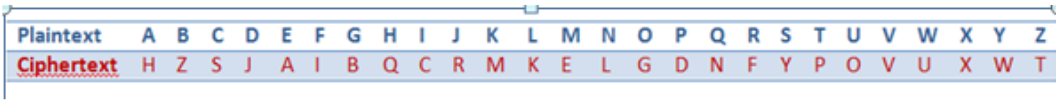
Soeb Aripin dan Muhammad Syahrizal (2022). *Caesar cipher* merupakan salah satu algoritma *cipher* tertua dan paling diketahui dalam perkembangan ilmu kriptografi. *Caesar cipher* merupakan salah satu jenis *cipher substitusi* yang membentuk *cipher* dengan cara melakukan penukaran karakter pada *plainteks* menjadi tepat satu karakter pada *cipherteks*. Teknik seperti ini disebut juga sebagai *cipher* abjad tunggal. Adapun langkah-langkah yang dilakukan untuk membentuk *cipherteks* dengan *Caesar cipher* adalah:

a) Menentukan besarnya pergeseran karakter yang digunakan dalam membentuk *cipherteks* ke *plainteks*.
b) Menukarkan karakter pada *plainteks* menjadi *cipherteks* dengan berdasarkan pada pergeseran yang telah ditentukan sebelumnya. Misalnya diketahui bahwa pergeseran = 3, maka huruf A akan digantikan oleh huruf D, huruf B menjadi huruf E, dan seterusnya.

2.8.4 Subtitution Cipher

Membaca secara *vertikal*. yang terdapat pada sebuah teks menjadi karakter yang lain. Karakter yang diganti dapat berupa angka maupun huruf. Jika pada *Caesar Cipher*, kunci dari cipher teksnya merupakan menggeser huruf pada

susunan alfabet, pada *cipher mono alphabetic* hal tersebut tidak dilakukan. Dengan menyusun huruf cipher secara acak. Di sini, huruf *cipher* pengganti huruf asli disusun secara acak. Kuncinya merupakan susunan hurufnya. Menjadi contoh:



Plaintext	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
Ciphertext	H	Z	S	J	A	I	B	Q	C	R	M	K	E	L	G	D	N	F	Y	P	O	V	U	X	W	T

Gambar 2.6 Subtitution Cipher

Plaintext: Abjad Tunggal
Chippertext: HZRHJPOLBBHK
Key: HZSJAIBQCRMKELGDNFYPOVUXWT

Contoh di atas, abjad pengganti atau huruf ciphernya benar-benar disusun secara acak tanpa ada pola tertentu sehingga kunci dari kriptografi jenis ini merupakan pada susunan seluruh huruf penggantinya. Sehingga bila kuncinya hilang, maka akan sangat sulit memecahkan cipherteksnya karena banyak sekali kemungkinannya serta tidak adanya pola tertentu (Septian Widiyanto et al., 2021).

2.8.5 Affine Cipher

Aripin Rambe (2022), *Affine chipper* adalah perluasan dari *Caesar cipher* yang mengalikan *plainteks* dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran. *Enkripsi plainteks* P menghasilkan cipherteks C dapat dinyatakan dengan fungsi kongruen sebagai berikut:

$$E(P) = (ax + b) \bmod m$$

Dimana:
m = Ukuran alphabet.
a = Bilangan bulat yang harus relatif prima dengan m (bila tidak relatif prima maka dekripsi tidak bias).
b = Jumlah pergeseran (Caesar cipher adalah khusus dari affine cipher dengan m = 1)
x = Plainteks yang dikonversi menjadi bilangan bulat dari 0 sampai m - 1 sesuai dengan urutan dalam alphabet.

E (P) = Cipherteks yang dikonversi menjadi bilangan bulat dari 0 sampai m - 1 sesuai dengan urutan dalam alphabet.

Sedangkan fungsi dekripsinya dapat dituliskan dengan menggunakan persamaan sebagai berikut:

D (x) = a – 1 (x - b) mod m

Dimana a - 1 adalah invers perkalian a modulus m yang dapat memenuhi persamaan berikut:

1 = a - 1 mod m

Invers perkalian a hanya ada jika a dan m adalah *coprime*. Jika tidak maka proses algoritma akan terhenti. Fungsi *dekripsi* merupakan kebalikan dari fungsi *enkripsi* yang dapat dituliskan sebagai berikut:

D (E (P)) = a - 1(E (P) - b) mod m
= a - 1 (((ax + b) mod m) - b) mod m
= a - 1(ax + b - b) mod m
= a - 1 ax mod m
D (E (x)) = x mod m

Contoh konkrit dari kegiatan satu mengenkripsi dan satu menderipsikan dimana alfabet akan menjadi huruf A sampai huruf Z dan akan memiliki nilai sesuai dengan Tabel Enkripsi dan dekripsi dibawah.

Tabel 2.3 Konversi nilai alphabet Algoritma Affine Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

2.8.6 Hill Cipher

Haris, C. A., & Ariyus, D. (2020). *Hill cipher* termasuk jenis kriptografi klasik yang diciptakan oleh Lester S. Hill pada tahun 1929. Algoritma Hill cipher ini mengganti setiap abjad *plainteksnya* ke dalam bentuk angka *numerik* yang berkorespondensi dengan 0 sampai 25. Dalam penerapannya algoritma ini yang menggunakan persamaan aritmatika modulo terhadap *matriks*, dengan perkalian dan teknik *invers* terhadap matriks. Kuncinya menggunakan matriks n x n dengan n merupakan ukuran blok. *Plainteks* terlebih dahulu dirubah kedalam bentuk angka seperti blok pada Tabel Enkripsi dan dekripsi dibawah.

Tabel 2.4 Konversi nilai alphabet Algoritma Hill Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

Perhitungan untuk proses *enkripsi algoritma hill cipher* ini menggunakan persamaan.

Ciphertext = Matriks Kunci * Matriks Plaintext Modulo 26

Perhitungan untuk proses *dekripsi algoritma hill cipher* ini menggunakan persamaan sebelumnya.

Plaintext = Matriks Kunci - 1 * Matriks Ciphertext Modulo 26

Adjoin K = $\begin{matrix} d & -b \\ -c & a \end{matrix}$

Det K = (a x d) – (b x c)

|K| = Det K x n Mod 26 = 1

Dimana:

K: matriks kunci

n: nilai yang memenuhi agar hasil = 1

2.8.7 Transpotition Columnar Cipher

Reswan, Y., & Yuliansyah, B. T. (2018) *Algoritma transposisi columnar* merupakan *algoritma* klasik yang penggunaanya cukup sederhana. Pada tahap *enkripsi transposition cipher* tidak mengganti huruf *plaintext* untuk menghasilkan *ciphertext* layaknya *subtitution cipher*. Hasil enkripsi karakter *plaintext* dengan posisi yang berbeda. *Plaintext* akan ditulis dalam *matrik* dengan panjang kolom sesuai dengan panjang karakter kunci yang digunakan. Penulisan *plaintext* ditulis dari baris per baris dimulai dengan baris pertama. *Ciphertext transposisi columnar cipher* dihasilkan dari penyusunan ulang *plaintext*. Kolom yang disusun pertama adalah kolom yang berhubungan dengan karakter sesuai urutan abjad. Contoh *enkripsi* menggunakan pesan "MEET ME AT NEXT MID NIGHT" dan kunci "FANCY". Penyusunan dimulai dari kolom yang berhubungan dengan karakter urutan pertama pada abjad yaitu "A", kemudian "C", "F", "N" dan "Y". Hasil dari enkripsi tersebut adalah "EATITNIHMEXNETMGMEDT". Model matematis proses *enkripsi transposisi columnar cipher* menggunakan persamaan:

$Ct\ of\ P = Y$o ……. Yl

*Xpo*1

……

*Xpo*2

……

*Xpl*2

Xpom

……

Xplm

Keterangan:

$Ct\ of\ P$ = *Columnar Transposition* dari pesan

Yo = Karakter pertama dari kunci

Yl = Karakter terakhir dari kunci

$X\ po$ = Karakter pertama dari pesan yang berelasi dengan 0

$X\ pl$1 = Karakter pertama dari pesan yang berelasi dengan $Y\ l$

$X\ pom$ = Karakter terakhir dari pesan yang berelasi dengan $Y0$

$X\ plm$ = Karakter terakhir dari pesan yang berelasi dengan Y e

Jika pada persamaan (1) $Ct\ of\ P$ didefinisikan sebagai $CtPi$ dengan i adalah kolom pada persamaan. Maka *cipher* text dari proses *enkripsi* tersebut dapat dimodelkan sebagai:

Cp = {Ct P1 + Ct P2 + Ct P3 + …. + CtPm}

Keterangan:

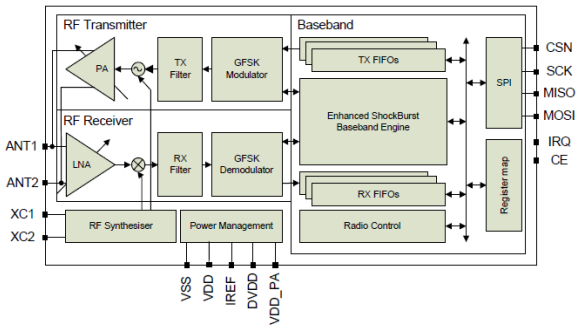
Cp: Hasil *enkripsi (Cipher teks)*

m: Kolom terakhir dari persamaan

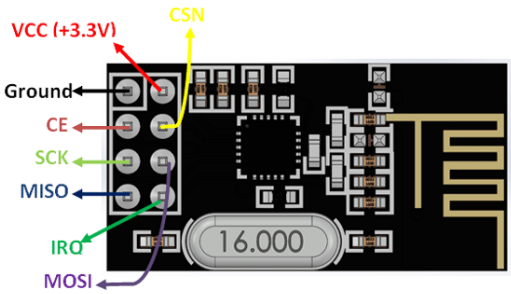
2.9 NRF24

NRF24L01+ merupakan transceiver 2.4GHz chip tunggal dengan protokol baseband tertanam. nRFL01+ sangat sesuai untuk digunakan pada perangkat nirkabel dengan daya sangat kecil. nRF24L01+ dirancang agar bekerja pada frekuensi ISM di pita 2.4-2.4835 GHz. nRF24L01+ dapat digunakan bantuan mikrokontroler.

nRF24L01+ dapat diatur dan dioperasikan melalui Serial Peripheral Interface (SPI). SPI dapat digunakan untuk mengakses peta register yang berisi semua register pengaturan pada nRF24L01+ dan dapat diakses pada semua mode operasi pada chip. nRF24L01+ menggunakan modulasi Gausian Frequency Shift Keying (GFSK) sebagai front end radio dimana parameter seperti kanal frekuensi, daya pancar dan date rate-nya dapat diatur oleh user. Data rate yang dapat digunakan diantaranya 250 kbps, 1 Mbps, dan 2 Mbps.



Gambar 2.7 Blok diagram nrf24l01



Gambar 2.8 Nrf24l01

Pada gambar ditampilkan block diagram chip nRF24L01. nRF24l01 terdiri dari dua bagian besar yaitu bagian radio frequency (RF) dan bagian baseband. Blok RF terdiri dari pemancar dan penerima dengan modulasi GFSK. Pemancar GFSK terdiri dari modulator yang memodulasi sinyal *baseband* kemudian difilter kemudian dimultiplier dengan RF synthesiser sebelum dikuatkan dengan *power amplifier*. Sinyal yang telah di kuatkan kemudian dipancarkan melalui antena. Penerima GFSK akan menerima sinyal dari antena kemudian dilakukan diumpankan ke *Low Noise Amplifier* (LNA). Sinyal dari LNA dimixer dengan sinyal dari RF synthesiser sebelum difilter pada RX filter dan didemodulasi kemudian diteruskan ke bagian baseband.

Bagian baseband terdiri dari buffer TX FIFO dan RX FIFO, Enhanced ShockBurst Baseband Engine, Radio Control, SPI dan Register map. Buffer TX FIFO dan RX FIFO dengan sistem *First in First Out* (FIFO) dimana data dari TX akan diberikan ke modulator untuk di pancarkan dan data yang di terima demodulator akan disimpan di buffer RX FIFO. *Enhanced ShockBurst* merupakan lapisan link data berbasis data paket. Fitur Enhanced ShockBurst melakukan penggabungan paket, timing, acknowledgement dan *re-transmission* packet otomatis. Fitur ini juga mengimplemantasikan komunikasi dengan performa tinggi dengan daya sangat rendah dengan host mikrokontroler (Nordic, 2007). Radio *Control* berfungsi untuk mengatur proses radio *transceiver*. *Register Map* berfungsi untuk konfigurasi dan pengaturan chip radio melalui akses SPI. Modul nRF24L01 ditunjukkan pada gambar (Hutabarat. et al. 2021).

2.10 LCD

LCD (*Liquid Crystal Display*) adalah komponen yang menggunakan kristal cair yang mempunyai fungsi sebagai tampilan suatu data, baik karakter, huruf atau grafik.

Gambar 2.9 LCD

LCD lebih hemat energi dibandingkan dengan model CRT. Konsumsi daya listrik yang rendah ini membuat *batery* akan lebih tahan lama. Dengan LCD tampilan gambar akan kelihatan halus (Effendi, H., & Puspitaningrum, R., 2021).

2.11 Sofware Arduino

Untuk memprogram *processor* yang ada pada Arduino Nano, diperlukan sebuah *software* bernama Arduino IDE. IDE merupakan singkatan dari *Integrated Developtment Environment*, atau secara bahasa mudahnya lingkungan terintegrasi yang digunakan untuk pengembangan. Arduino IDE merupakan aplikasi pemrograman untuk perangkat Arduino agar Arduino dapat melakukan fungsi-fungsi yang dapat kita inginkan. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Aplikasi Arduino IDE juga memiliki kumpulan contoh program yang berada pada *library* sehingga pemula dapat dengan mudah untuk melakukan pemrogramanan (allgo, 2017). Program dalam *Sketch Arduino* dapat dibagi dalam tiga bagian utama, yaitu:

1. Structure

   Struktur *software* pada *sketch* terdiri dari 2 fungsi utama, yaitu:

   a) Setup()

      Fungsi ini dipanggil pertama kali ketika menjalankan *sketch* dan digunakan sebagai tempat untuk inisialisasi *variable*, pin *mode*, penggunaan

library dan lain-lain. Fungsi ini dijalankan sekali ketika papan arduino dinyalakan atau di reset.

b) Loop()

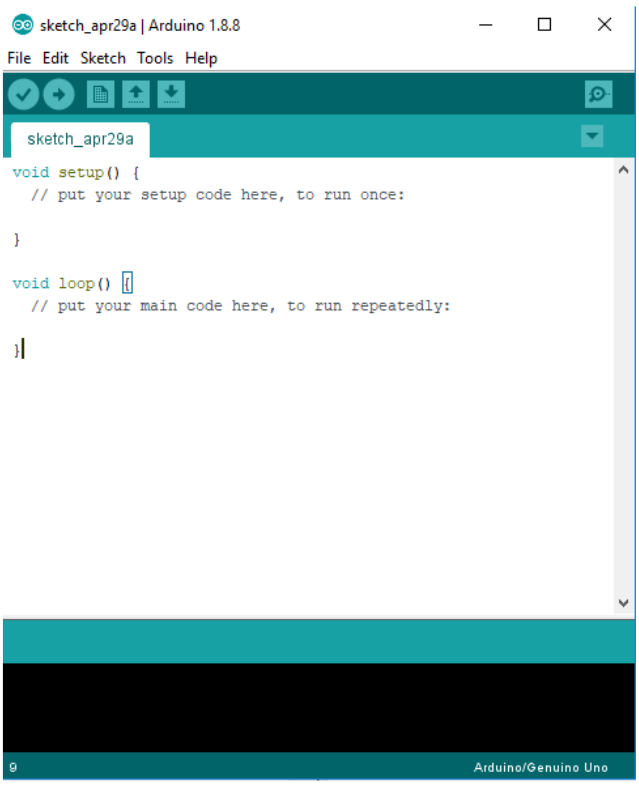
Setelah membuat fungsi *setup*() sebagai tempat inisialisasi variabel dan menetapkan nilai maka selanjutnya fungsi *loop*() akan melakukan perulangan berturut-turut, memungkinkan program untuk mengubah dan menanggapi. Loop() ini digunakan untuk mengontrol papan arduino.

2. Values

Values pada *sketch* berisi variabel atau konstanta sesuai dengan tipe data yang didukung oleh arduino.

3. Function

Segmentasi kode program ke bentuk *function* memungkinkan *programmer* untuk membuat potongan-potongan kode yang melakukan tugas yang terdefinisi dan kemudian kembali ke asal kode dari mana *function* itu "dipanggil". Penggunaan function adalah ketika ada kebutuhan untuk melakukan tindakan yang sama beberapa kali dalam sebuah program.



Gambar 2.10 *Software* Arduino IDE

2.12 Bahasa Pemograman Arduino Berbasis Bahasa C

Seperti yang telah dijelaskan diatas program Arduino sendiri menggunakan bahasa C. walaupun banyak sekali terdapat bahasa pemrograman tingkat tinggi (*high level language*) seperti *pascal*, *basic*, cobol, dan lainnya. Walaupun demikian, sebagian besar dari paraprogramer profesional masih tetap memilih bahasa C sebagai bahasa yang lebih unggul (Fathulrohman et al., 2019).

BAB III
METODELOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian akan dilaksanakan pada bulan Januari hingga Juli 2024. Jadwal penelitian sebagaimana yang dimuat pada tabel.

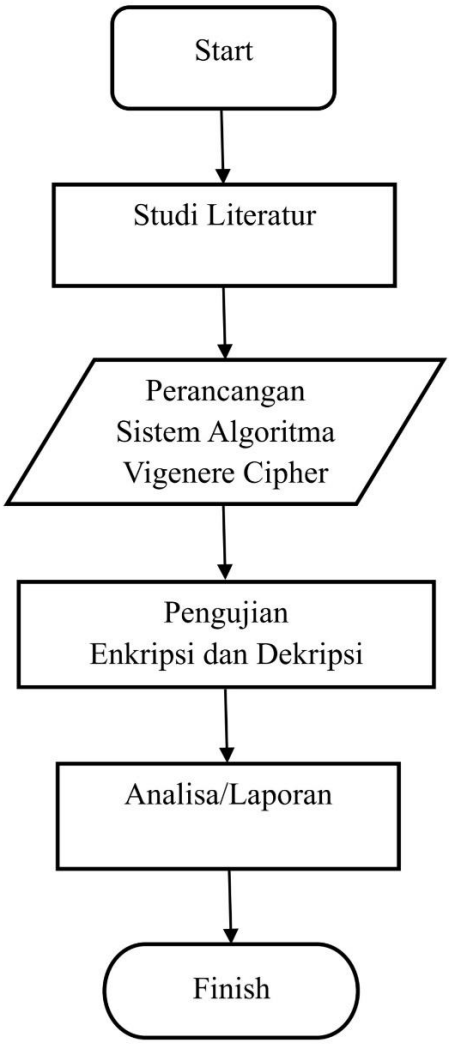
Tabel 3.1 Jadwal penelitian

NO.	KEGIATAN	JANUARI 2024				MARET 2024				MEI 2024				JULI 2024
		I	II	III	IV	I	II	III	IV	I	II	III	IV	III
1.	Studi Literatur													
2.	Perancangan													
3.	Seminar Proposal													
4.	Implementasi													
5.	Pengujian													
6.	Hasil													
7.	Sidang Akhir													

3.2 Metodelogi Penelitian

Menyelesaikan penelitian ini digunakan sebuah metode yang digunakan untuk mendukung dalam proses pembuatan. Dimana metode yang digunakan yaitu metode mengamankan data teks pada perangkat mikrokontroler arduino uno *ATmega328*. Untuk melindungi kerahasiaan data teks tersebut, kami mengadopsi algoritma kriptografi vigenere cipher. Penelitian ini bertujuan untuk menganalisis efektivitas penggunaan algoritma *vigenere cipher* pada platform mikrokontroler arduino uno *ATmega328.*

Berikut ini ialah alir penelitian yang akan di lakukan dalam penelitian mengamankan data teks pada perangkat mikrokontroler arduino uno *ATmega328*.



Gambar 3.1 Alir Penelitian

Studi Literatur:

- Melakukan kajian pustaka untuk memahami dasar teori yang relevan dengan penelitian ini, termasuk konsep keamanan data, algoritma kriptografi, dan khususnya Vigenere Cipher.
- Mengkaji penelitian-penelitian terdahulu yang telah menggunakan ATmega328 untuk aplikasi keamanan data.
- Mengidentifikasi kekuatan dan kelemahan dari berbagai metode enkripsi yang dapat diterapkan pada perangkat mikrokontroler.

Perancangan Algoritma Vigenere Cipher:

- Mengembangkan algoritma Vigenere Cipher yang akan diimplementasikan pada ATmega328.
- Menentukan struktur kunci enkripsi dan dekripsi serta mekanisme untuk memasukkan dan menyimpan kunci tersebut.

Implementasi Algoritma Vigenere Cipher:

- Menulis kode program untuk implementasi algoritma Vigenere Cipher menggunakan bahasa pemrograman yang sesuai, seperti C atau Arduino.
- Memprogram perangkat ATmega328 dengan algoritma yang telah dikembangkan.
- Mengintegrasikan input dan output teks pada perangkat untuk proses enkripsi dan dekripsi.

Pengujian Enkripsi dan Dekripsi:

- Melakukan uji coba untuk memastikan bahwa proses enkripsi dan dekripsi berjalan sesuai dengan yang diharapkan.
- Menguji algoritma dengan panjang dan kompleksitas teks serta kunci untuk mengevaluasi kinerja dan keamanannya.

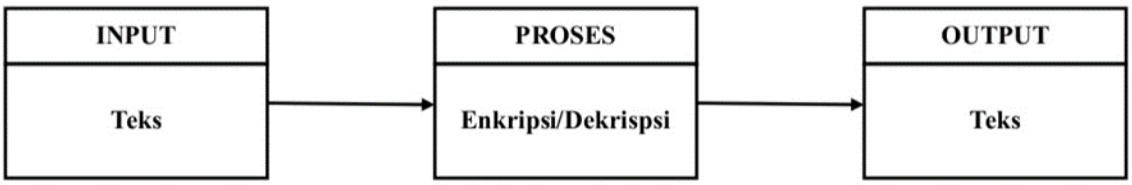
Analisis dan Pelaporan:

- Menganalisis hasil pengujian untuk mengidentifikasi kekuatan dan kelemahan dari implementasi algoritma Vigenere Cipher pada ATmega328.
- Membuat laporan yang merinci setiap tahap penelitian, mulai dari studi literatur, perancangan algoritma, implementasi, hingga pengujian.
- Memberikan rekomendasi untuk penelitian lanjutan atau perbaikan algoritma berdasarkan temuan selama penelitian.

Dengan mengikuti alur penelitian ini, diharapkan penelitian dapat dilakukan secara sistematis dan menghasilkan solusi yang efektif untuk mengamankan data teks pada perangkat ATmega328.

3.3 Prosedur Penelitian

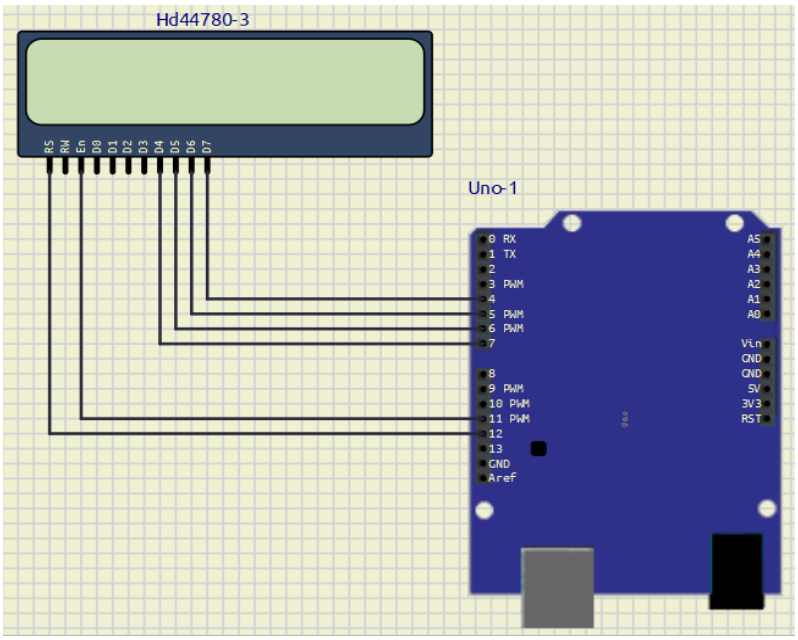
Untuk mengimplementasikan algorima *vigenere cipher* pada perangkat mikrokontroler arduino uno *ATmega328* diperlukan perancangan sistem, agar diketahui prinsip kerja secara keseluruhan dari rangkaian yang dibuat. Sehingga keseluruhan blok yang dibuat dapat membentuk suatu sistem yang dapat difungsikan atau sistem yang bekerja sesuai dengan perancangan. Keseluruhan rangkaian sistem yang dibuat dapat dilihat pada gambar dibawah.



Gambar 3.2 Blok Rangkaian

3.4 Rangkaian LCD 16x2

Lcd yang digunakan tipe 16x2. Lcd 16x2 terhubung pada kaki, pin D7_pin = 4; D6_pin = 5; D5_pin = 6; D4_pin = 7; EN_pin = 11; RS_pin = 12 pada mikrokontroller arduino uno ATmega328. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



Gambar 3.3 Rangkaian mikrokontroler Arduino Uno ATmega328 yang dihubungkan menggunakan kabel jumper dengan LCD

3.5 Kalkulasi Matematika Algoritma Vigenere Cipher

Sebelum melakukan pembuatan sistem keamanan algoritma vigenere cipher, maka untuk dapat memastikan bahwa penerapan ini dapat dilakukan dengan akurat bahwa data teks yang dienkripsi kemudian didekripsi tidak mengalami perubahan diperlukannya kalkulasi matematika berikut.

Plainteks yang digunakan "MUHAMMAD_ALFATIH" dengan key yang digunakan "ROMA".

1. Kalkulasi Matematika Enkripsi Plainteks menjadi Cipherteks Algoritma Vigenere Cipher.

Diketahui: Pi = MUHAMMAD_ALFATIH (128 bit)
Ki = ROMA (32 bit)

Tabel 3.2 Konversi nilai plainteks Algoritma Vigenere Cipher

M	U	H	A	M	M	A	D	_	A	L	F	A	T	I	H
22	30	17	10	22	22	10	13	75	10	21	15	10	29	18	17
R	O	M	A	R	O	M	A	R	O	M	A	R	O	M	A
27	24	22	10	27	24	22	10	27	24	22	10	27	24	22	10

C1 = M + R mod 94
= 22 + 27 mod 94
= 49 = n

C2 = U + O mod 94
= 30 + 24 mod 94
= 54 = s

C3 = H + M mod 94
= 17 + 22 mod 94
= 39 = d

C4 = A + A mod 94
= 10 + 10 mod 94
= 20 = K

C5 = M + R mod 94
= 22 + 27 mod 94

= 49 = n

C6 = M + O mod 94

= 22 + 24 mod 94

= 46 = k

C7 = A + M mod 94

= 10 + 22 mod 94

= 32 = W

C8 = D + A mod 94

= 13 + 10 mod 94

= 23 = N

C9 = _ + R mod 94

= 75 + 27 mod 94

= 8 = 8

C10 = A + O mod 94

= 10 + 24 mod 94

= 34 = Y

C11 = L + M mod 94

= 21 + 22 mod 94

= 43 = h

C12 = F + A mod 94

= 15 + 10 mod 94

= 25 = P

C13 = A + R mod 94

= 10 + 27 mod 94

= 37 = b

C14 = T + O mod 94

= 29 + 24 mod 94

= 53 = r

C15 = I + M mod 94

= 18 + 22 mod 94

= 40 = e

C16 = H + A mod 94
= 17 + 10 mod 94
= 27 = R

Dari kalkulasi matematika enkripsi plainteks menjadi cipherteks algoritma vigenere cipher maka didapatlah: “nsdKnkWN8YhPbreR” yang diharapkan sesuai dengan hasil sistem keamanan yang akan dibuat.

2. Kalkulasi Matematika Dekripsi Cipherteks menjadi Plainteks Algoritma Vigenere Cipher, yang sebelumnya didapat dari kalkulasi enkripsi plainteks menjadi cipherteks diatas dengan menggunakan key yang sama agar didapatkan hasil yang sesuai.

Diketahui: Ci = nsdKnkWN8YhPbreR (128 bit)
Ki = ROMA (32 bit)

Tabel 3.3 Konversi Nilai Cipherteks Algoritma Vigenere Cipher

N	s	D	K	N	K	W	N	8	Y	h	P	b	r	E	R
49	54	39	20	49	46	32	23	8	34	43	25	37	53	40	27
R	O	M	A	R	O	M	A	R	O	M	A	R	O	M	A
27	24	22	10	27	24	22	10	27	24	22	10	27	24	22	10

P1 = n – R mod 94
= 49 – 27 mod 94
= 22 = M

P2 = s – O mod 94
= 54 – 24 mod 94
= 30 = U

P3 = d – M mod 94
= 39 – 22 mod 94
= 17 = H

P4 = K – A mod 94
= 20 – 10 mod 94

= 10 = A

P5 = n – R mod 94

= 49 – 27 mod 94

= 22 = M

P6 = k – O mod 94

= 46 – 24 mod 94

= 22 = M

P7 = W – M mod 94

= 32 – 22 mod 94

= 10 = A

P8 = N – A mod 94

= 23 – 10 mod 94

= 13 = D

P9 = 8 – R mod 94

= 8 – 27 mod 94

= 75 = _

P10 = Y – O mod 94

= 34 – 24 mod 94

= 10 = A

P11 = h – M mod 94

= 43 – 22 mod 94

= 21 = L

P12 = P – A mod 94

= 25 – 10 mod 94

= 15 = F

P13 = b – R mod 94

= 37 – 27 mod 94

= 10 = A

P14 = r – O mod 94

= 53 – 24 mod 94

= 29 = T

P15 = e – M mod 94
= 40 – 22 mod 94
= 18 = I
P16 = R – A mod 94
= 27 – 10 mod 94
= 17 = H

Dari kalkulasi matematika dekripsi cipherteks menjadi plainteks algoritma vigenere cipher maka didapatlah: “MUHAMMAD_ALFATIH” yang diharapkan sesuai dengan hasil sistem keamanan yang akan dibuat.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

Dari permasalahan dalam penerapan sistem keamanan data menggunakan algoritma vigenere cipher pada mikrokontroler arduino uno atmega328. Hasil dari penelitian ini berupa implementasi sistem keamanan pada atmega328 dengan menggunakan metode algoritma vigenere cipher.

Pada penelitian ini menggunakan Arduino IDE sebagai tempat perancangan *source code*. Kemudian disimulasikan dengan menggunakan SimulIDE. Langkah awal adalah Analisa dari inti penelitian yaitu untuk menerapkan sistem keamanan.

4.2 Pembahasan

Kerentanan dalam sistem keamanan data terjadi ketika ada celah yang dapat dimanfaatkan oleh pihak yang tidak sah untuk mengakses, mengubah, atau merusak data yang dikirimkan melalui jaringan komunikasi. Ini bisa terjadi karena berbagai alasan, termasuk penggunaan algoritma enkripsi yang tidak efesien, protokol komunikasi yang rentan terhadap serangan, atau bahkan kesalahan manusia seperti kurangnya kesadaran tentang praktik keamanan. Salah satu kerentanan yang umum adalah enkripsi yang tidak efesien. Jika algoritma enkripsi yang digunakan tidak cukup kuat, penyerang dapat memecahkannya dan mendapatkan akses ke informasi sensitif. Serangan man-in-the-middle (MITM) juga merupakan ancaman yang serius. Dalam serangan ini, penyerang menyisipkan diri di antara pengirim dan penerima data, sehingga mereka dapat memata-matai atau bahkan mengubah data yang dikirimkan.

Selain itu, serangan denial-of-service (DoS) juga merupakan kerentanan yang signifikan. Penyerang dapat membanjiri sumber daya sistem komunikasi dengan permintaan palsu atau tidak sah, membuat layanan menjadi tidak tersedia bagi pengguna yang sah. Kerentanan pada protokol komunikasi, perangkat lunak klien, server, atau perangkat jaringan juga dapat dimanfaatkan oleh penyerang untuk merusak keamanan sistem komunikasi data. Pengelolaan akses yang lemah juga bisa menyebabkan kerentanan. Ketika sistem tidak membatasi akses secara

tepat, pengguna mungkin memiliki hak akses yang tidak seharusnya, atau sebaliknya, pengguna yang tidak berhak bisa mendapatkan akses yang tidak sah. Untuk mengatasi kerentanan dalam sistem keamanan komunikasi data, penting untuk menggunakan teknologi enkripsi yang efesien, menerapkan protokol keamanan yang tepat, memperbarui perangkat lunak secara teratur, meningkatkan kesadaran pengguna tentang praktik keamanan, dan secara aktif memantau ancaman keamanan yang muncul.

4.2.1 Penbuatan Program Algoritma Vigenere Cipher

Pada penelitian ini mengatasi kerentanan sistem keamanan data diperlukan metode dan cara pengamanan data. Program sistem keamanan data menggunakan algoritma vigenere cipher yang kemudian di compile dalam format .hex dapat dilihat dibawah ini.

1. Program Enkripsi Plainteks menjadi Cipherteks Algoritma Vigenere Cipher

a. Adapun diawal program terdapat *liquidcrystal*, *liquidcrystal* ini merupakan library agar penggunaan lcd pada simulide dapat digunakan. Program dapat dilihat dibawah ini:

```
include <LiquidCrystal.h>

int D7_pin = 4;
int D6_pin = 5;
int D5_pin = 6;
int D4_pin = 7;
int EN_pin = 11;
int RS_pin = 12;

LiquidCrystal lcd(RS_pin, EN_pin, D4_pin, D5_pin, D6_pin, D7_pin);
```

b. Kemudian juga ada arai char ref dan juga integer enkrip, kedua arai ini berfungsi untuk menyimpan arai berupa angka-angka, huruf-huruf serta simbol-simbol yang kita butuhkan. Program dapat dilihat dibawah ini:

```
char Ref[94] =
                {'0','1','2','3','4','5','6','7','8','9','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','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','`','~','!','
                @','#','$','%','^','&','*','(',')','-
                ','_','+','=','[',']','{','}','|',';',':','"','<','>','/','?',',','.',',','\'',''\\'};

int Encrypt[94] =
                    {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
                    23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,4
                    2,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,
                    62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,8
                    1,82,83,84,85,86,87,88,89,90,91,92,93};
```

c. Selanjutnya ada string msg yang berupa “MUHAMMAD_ALFATIH” adalah plainteks yang kita miliki. Program dapat dilihat dibawah ini:

```
int button = 13;
String msg = "MUHAMMAD_ALFATIH";
int a = 3;
```

d. Kemudian void encryption, disini berisi program yang akan mengubah huruf-huruf pada plainteks. Program dapat dilihat dibawah ini:

```
void encryption(String str) {
  int msgLength = 0;
  int RefLength = 0;
  String dataEncrypt[20];

  RefLength = sizeof(Ref);
  msgLength = str.length();

  //Encrypt
  for(int i=0;i<msgLength;i++) {
    for(int j=0;j<RefLength;j++) {
      if(msg[i]==Ref[j]) {
        dataEncrypt[i] = String(Ref[(j+a) % RefLength]);
      }
    }
    lcd.print(dataEncrypt[i]);
    delay(30);
  }
}
```

e. Kemudian void vigenere yang berfungsi dalam perubahan pada huruf-huruf plainteks menggunakan teknik sandi vigenere. Key yang digunakan berupa string “ROMA”. Program dapat dilihat dibawah ini:

```
void vignere(String str) {
  String vigstring = "ROMA";
  int viglength = vigstring.length();
  int vigint[viglength];
  int strLength = str.length();
  int vigkey;
  String output[strLength];

  for(int i=0;i<viglength;i++) {
    for(int j=0;j<sizeof(Ref);j++) {
      if(vigstring[i]==Ref[j]) {
        vigint[i]=j;
      }
    }
  }
  for(int i=0;i<str.length();i++) {
    vigkey = vigint[i%viglength];
    output[i] = encrypt_char(str[i], vigkey);
    lcd.print(output[i]);
    delay(30);
  }
}
```

f. Selanjutnya ada char encrypt berfungsi untuk mengubah huruf-huruf pada plainteks dengan memanggil data encrypt. Program dapat dilihat dibawah ini:

```
char encrypt_char(char input, int key) {
  int intEncrypt;
  int RefLength = 0;
  char dataEncrypt;

  RefLength = sizeof(Ref);

  //Encrypt
  for(int i=0;i<RefLength;i++) {
    if(input==Ref[i]) {
      intEncrypt=(i+key)%94;
      dataEncrypt=Ref[intEncrypt];
    }
  }
  return dataEncrypt;
}
```

g. Terakhir void loop yang berfungsi melakukan loop dengan memanggil vigenere msg. program dapat dilihat dibawah ini:

```
void loop() {
  vignere(msg);
  delay(3000);
  lcd.clear();
  delay(3000);
}
```

Program Keseluruhan Enkripsi Plainteks menjadi Cipherteks Algoritma Vigenere Cipher

```
#include <LiquidCrystal.h>
int D7_pin = 4;
int D6_pin = 5;
int D5_pin = 6;
int D4_pin = 7;
int EN_pin = 11;
int RS_pin = 12;
LiquidCrystal lcd(RS_pin, EN_pin, D4_pin, D5_pin, D6_pin, D7_pin);
char Ref[94] =
                {'0','1','2','3','4','5','6','7','8','9','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','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','`','~','!','@','#','$','%'
                ,'^','&','*','(',')','-','_','+','=','[',']','{','}','|',';',':','"','<','>','/','?',',','.','\'','\\'};
int Encrypt[94] =
                  {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,2
                  4,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,
                  45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,6
                  5,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,
                  86,87,88,89,90,91,92,93};
int button = 13;
String msg = "MUHAMMAD_ALFATIH";
int a = 3;
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
}
void encryption(String str) {
  int msgLength = 0;
  int RefLength = 0;
```

```
  String dataEncrypt[20];
  RefLength = sizeof(Ref);
  msgLength = str.length();
  //Encrypt
  for(int i=0;i<msgLength;i++) {
    for(int j=0;j<RefLength;j++) {
      if(msg[i]==Ref[j]) {
        dataEncrypt[i] = String(Ref[(j+a) % RefLength]);
      }
    }
    lcd.print(dataEncrypt[i]);
    delay(30);
  }
}
void vignere(String str) {
  String vigstring = "ROMA";
  int viglength = vigstring.length();
  int vigint[viglength];
  int strLength = str.length();
  int vigkey;
  String output[strLength];
  for(int i=0;i<viglength;i++) {
    for(int j=0;j<sizeof(Ref);j++) {
      if(vigstring[i]==Ref[j]) {
        vigint[i]=j;
      }
    }
  }
  for(int i=0;i<str.length();i++) {
    vigkey = vigint[i%viglength];
    output[i] = encrypt_char(str[i], vigkey);
```

```
    lcd.print(output[i]);
    delay(30);
  }
}
char encrypt_char(char input, int key) {
  int intEncrypt;
  int RefLength = 0;
  char dataEncrypt;
  RefLength = sizeof(Ref);
  //Encrypt
  for(int i=0;i<RefLength;i++) {
    if(input==Ref[i]) {
      intEncrypt=(i+key)%94;
      dataEncrypt=Ref[intEncrypt];
    }
  }
  return dataEncrypt;
}
void loop() {
  vignere(msg);
  delay(3000);
  lcd.clear();
  delay(3000);
}
```

2. Program Dekripsi Cipherteks menjadi Plainteks Algoritma Vigenere Cipher

a. Adapun diawal program terdapat *liquidcrystal*, *liquidcrystal* ini merupakan library agar penggunaan lcd pada simulide dapat digunakan. Program dapat dilihat dibawah ini:

```
#include <LiquidCrystal.h>

int D7_pin = 4;
int D6_pin = 5;
int D5_pin = 6;
int D4_pin = 7;
int EN_pin = 11;
int RS_pin = 12;

LiquidCrystal lcd(RS_pin, EN_pin, D4_pin, D5_pin, D6_pin, D7_pin);
```

b. Kemudian juga ada arai char ref dan juga integer decrypt, kedua arai ini berfungsi untuk menyimpan arai berupa angka-angka, huruf-huruf serta simbol-simbol yang kita butuhkan. Program dapat dilihat dibawah ini:

```
char Ref[94] =
                {'0','1','2','3','4','5','6','7','8','9','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','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','`','~','!','
                @','#','$','%','^','&','*','(',')','-
                ','_','+','=','[',']','{','}','|',';',':','"','<','>','/','?',',','.','\'','\\'};

int Decrypt[94] =
                {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
                24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,4
                4,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,
                65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,8
                5,86,87,88,89,90,91,92,93};
```

c. Selanjutnya ada string msg yang berupa “nsdKnkWN8YhPbreR” adalah cipherteks yang kita miliki. Program dapat dilihat dibawah ini:

```
int button = 13;
String msg = "nsdKnkWN8YhPbreR";
int a = 3;
```

d. Kemudian void decryption, disini berisi program yang akan mengubah huruf-huruf pada cipherteks. Program dapat dilihat dibawah ini:

```
void decryption(String str) {
  int msgLength = 0;
  int RefLength = sizeof(Ref);
  String dataDecrypt;

  msgLength = str.length();

  //Decrypt
  for(int i=0;i<msgLength;i++) {
    for(int j=0;j<RefLength;j++) {
      if(str[i]==Ref[j]) {
        int index = j - a;
        if (index < 0) {
          index += RefLength;
        }
        dataDecrypt += Ref[index];
      }
    }
    lcd.print(dataDecrypt);
    delay(80);
  }
}
```

e. Kemudian void vigenere yang berfungsi dalam perubahan pada huruf-huruf plainteks menggunakan teknik sandi vigenere. Key yang digunakan berupa string “ROMA”. Program dapat dilihat dibawah ini:

```
void vignere(String str) {
  String vigstring = "ROMA";
  int viglength = vigstring.length();
  int vigint[viglength];
  int strLength = str.length();
  int vigkey;
  String output[strLength];

  for(int i=0;i<viglength;i++) {
    for(int j=0;j<sizeof(Ref);j++) {
      if(vigstring[i]==Ref[j]) {
        vigint[i]=j;
      }
    }
  }
  for(int i=0;i<str.length();i++) {
    vigkey = vigint[i%viglength];
    output[i] = decrypt(str[i], vigkey);
    lcd.print(output[i]);
    delay(30);
  }
}
```

f. Selanjutnya ada char decrypt berfungsi untuk mengubah huruf-huruf pada cipherteks dengan memanggil data decrypt. Program dapat dilihat dibawah ini:

```
char decrypt(char input, int key) {
  int intDecrypt;
  int RefLength = sizeof(Ref);
  char dataDecrypt;

  //Decrypt
  for(int i=0;i<RefLength;i++) {
    if(input==Ref[i]) {
      intDecrypt = (i - key) % RefLength;
      if (intDecrypt < 0) {
        intDecrypt += RefLength;
      }
      dataDecrypt = Ref[intDecrypt];
    }
  }
  return dataDecrypt;
}
```

g. Terakhir void loop yang berfungsi melakukan loop dengan memanggil vigenere msg. program dapat dilihat dibawah ini:

```
void loop() {
  vignere(msg);
  delay(3000);
  lcd.clear();
  delay(3000);
}
```

Program Keseluruhan Dekripsi Cipherteks menjadi Plainteks Algoritma Vigenere Cipher

```
#include <LiquidCrystal.h>
int D7_pin = 4;
int D6_pin = 5;
int D5_pin = 6;
int D4_pin = 7;
int EN_pin = 11;
int RS_pin = 12;
LiquidCrystal lcd(RS_pin, EN_pin, D4_pin, D5_pin, D6_pin, D7_pin);
char Ref[94] =
                {'0','1','2','3','4','5','6','7','8','9','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','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','`','~','!','@','#','$','%'
                ,'^','&','*','(',')','-','_','+','=','[',']','{','}','|',';',':','"','<','>','/','?',',','.','\'','\\'};
int Decrypt[94] =
                  {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,2
                  4,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,
                  45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,6
                  5,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,
                  86,87,88,89,90,91,92,93};
int button = 13;
String msg = "nsdKnkWN8YhPbreR";
int a = 3;
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
}
void decryption(String str) {
  int msgLength = 0;
  int RefLength = sizeof(Ref);
```

```
  String dataDecrypt;
  msgLength = str.length();
  //Decrypt
  for(int i=0;i<msgLength;i++) {
    for(int j=0;j<RefLength;j++) {
      if(str[i]==Ref[j]) {
        int index = j - a;
        if (index < 0) {
          index += RefLength;
        }
        dataDecrypt += Ref[index];
      }
    }
    lcd.print(dataDecrypt);
    delay(80);
  }
}
void vignere(String str) {
  String vigstring = "ROMA";
  int viglength = vigstring.length();
  int vigint[viglength];
  int strLength = str.length();
  int vigkey;
  String output[strLength];
  for(int i=0;i<viglength;i++) {
    for(int j=0;j<sizeof(Ref);j++) {
      if(vigstring[i]==Ref[j]) {
        vigint[i]=j;
      }
    }
  }
```

```
  for(int i=0;i<str.length();i++) {
    vigkey = vigint[i%viglength];
    output[i] = decrypt(str[i], vigkey);
    lcd.print(output[i]);
    delay(30);
  }
}
char decrypt(char input, int key) {
  int intDecrypt;
  int RefLength = sizeof(Ref);
  char dataDecrypt;
  //Decrypt
  for(int i=0;i<RefLength;i++) {
    if(input==Ref[i]) {
      intDecrypt = (i - key) % RefLength;
      if (intDecrypt < 0) {
        intDecrypt += RefLength;
      }
      dataDecrypt = Ref[intDecrypt];
    }
  }
  return dataDecrypt;
}
void loop() {
  vignere(msg);
  delay(3000);
  lcd.clear();
  delay(3000);
}
```

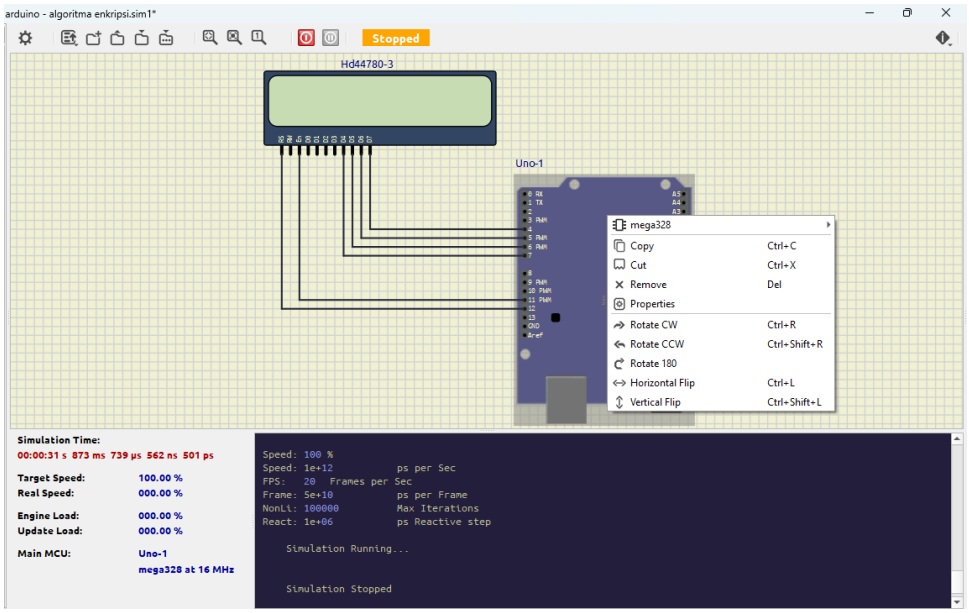
4.2.2 Pengujian Sistem

Algoritma vigenere cipher yang sebelunnya telah disimpan dalam format .hex menggunakan sofware Arduino IDE kemudian akan dilakukan pengujian pada software Simulide.

4.2.3 Pengujian Enkripsi Algoritma Vigenere Cipher

Pada pengujian pertama yaitu pengujian enkripsi data teks dengan key yang telah ditentukan sehingga akan mendapat menghasilkan cipherteks yang sesuai perhitungan matematika sebelumnya.

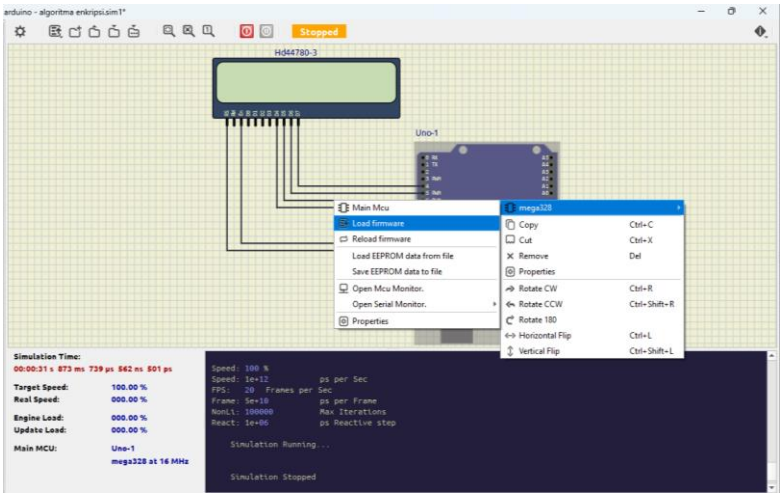
1. Untuk proses pengujiannya kita terlebih dahulu melakukan load firmware dari file code yang telah dicompile dengan mengklik kanan board Arduino lalu klik pada mega328 seperti gambar dibawah.



Gambar 4.1 Tampilan pop-up awal proses load firmware enkripsi plainteks menjadi cipherteks mikrokontroler Arduino Uno ATmega328 pada Simulide

Tahap ini merupakan langkah awal dalam proses melakukan load firmware enkripsi plainteks menjadi cipherteks mikrokontroler Arduino Uno ATmega328 dengan langkah tersebut akan menampilkan pop-up selanjutnya agar dapat melanjutkan proses enkripsi.

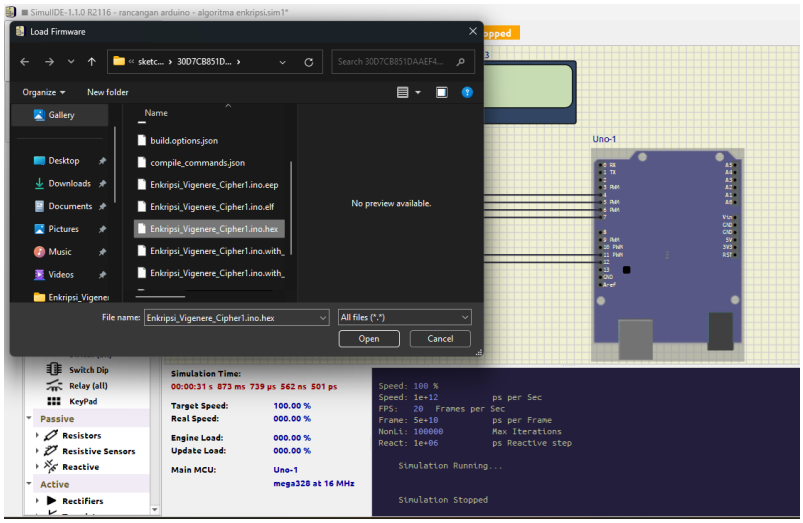
2. Kemudian akan muncul pop-up, klik load firmware.



Gambar 4.2 Tampilan pop-up proses load firmware enkripsi plainteks menjadi cipherteks mikrokontroler Arduino Uno ATmega328 pada Simulide

Setelah dapat menampilkan pop-up kedua dalam proses enkripsi plainteks menjadi cipherteks mikrokontroler Arduino Uno ATmega328 dan meng-klik load firmware selanjutnya kita akan dibawa memasuki ke library komputer untuk dapat mencari source code file .hex yang telah disimpan sebelumnya.

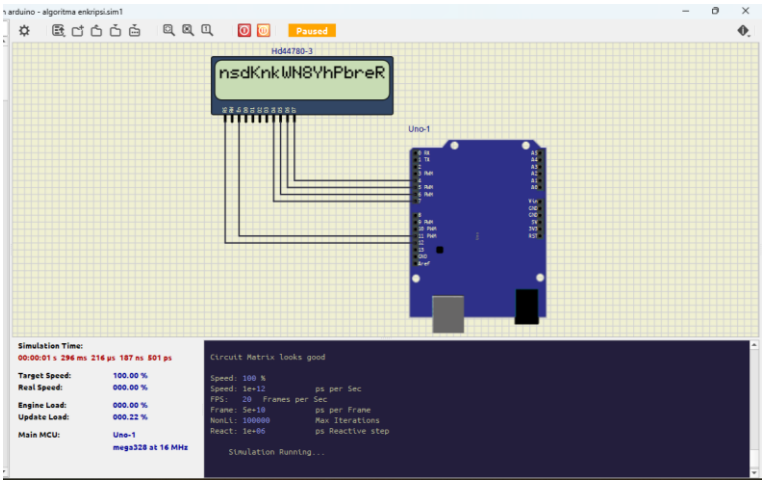
3. Setelahnya pilih source code yang telah di compile dan tersimpan dalam format .hex dan klik open.



Gambar 4.3 Tampilan library yang terdapat source code.hex enkripsi plainteks menjadi cipherteks mikrokontroler ATmega328

Jika telah menemukan source code .hex yang diperlukan dalam proses enkripsi plainteks menjadi cipherteks mikrokontroler Arduino Uno ATmega328 dan memilihnya lalu meng-klik open, maka program sistem keamanan algoritma vigenere cipher akan di load hingga selesai.

4. Setelah program selesai di loaded, kita dapat menjalankan program dengan menklik icon run.



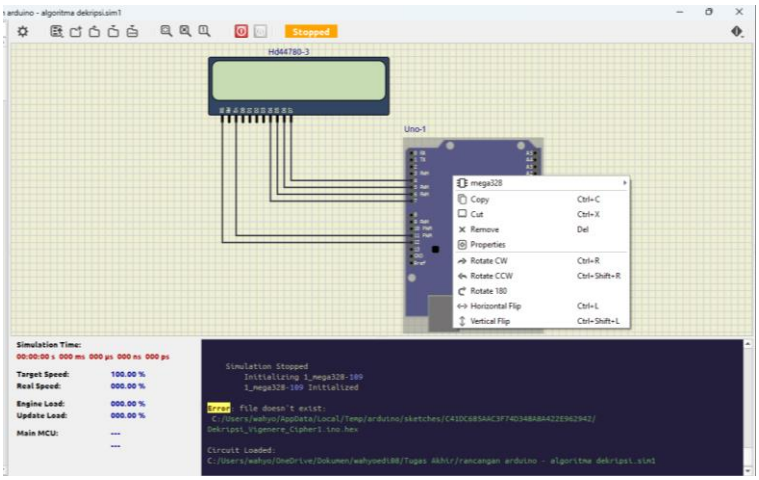
Gambar 4.4 Tampilan hasil enkripsi plainteks menjadi cipherteks pada LCD yang di proses Mikrokontroler ATmega328

Proses komputasi mikrokontroler Arduino Uno ATmega328 dijalankan begitu icon run diklik. Maka proses enkripsi sistem keamanan algoritma vigenere cipher dari plainteks dengan menggunakan key yang telah ditentukan, diperoleh hasil menjadi cipherteks seperti yang ditampilkan pada layar lcd gambar diatas: “nsdKnkWN8YhPbreR”.

4.2.4 Pengujian Dekripsi Algoritma Vigenere Cipher

Pengujian kedua yaitu mendekripsi cipherteks kembali menjadi plainteks dengan menggunakan key yang telah ditentukan, sama seperti sebelumnya dekripsi juga belewati beberapa proses berikut ini.

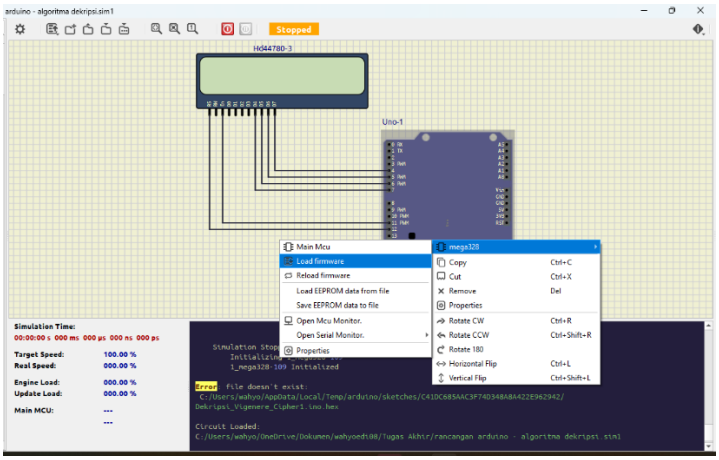
1. Untuk proses pengujiannya kita terlebih dahulu melakukan load firmware dari file code yang telah dicompile dengan mengklik kanan board Arduino lalu klik pada mega328 seperti gambar dibawah.



Gambar 4.5 Tampilan pop-up awal proses load firmware dekripsi cipherteks menjadi plainteks mikrokontroler Arduino Uno ATmega328 pada Simulide

Tahap ini merupakan langkah awal dalam proses melakukan load firmware dekripsi cipherteks menjadi plainteks mikrokontroler Arduino Uno ATmega328 dengan langkah tersebut akan menampilkan pop-up selanjutnya agar dapat melanjutkan proses dekripsi.

2. Kemudian akan muncul pop-up, klik load firmware.

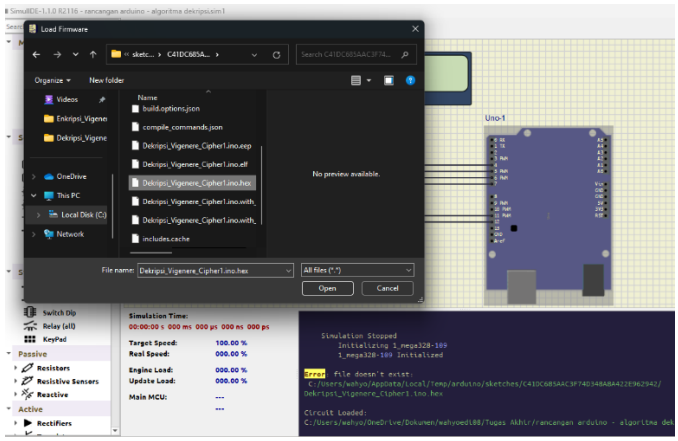


Gambar 4.6 Tampilan pop-up proses load firmware dekripsi cipherteks menjadi plainteks mikrokontroler Arduino Uno ATmega328 pada Simulide

Setelah dapat menampilkan pop-up kedua dalam proses dekripsi cipherteks menjadi plainteks mikrokontroler Arduino Uno ATmega328 dan meng-klik load firmware selanjutnya kita akan dibawa memasuki ke library

komputer untuk dapat mencari source code file .hex yang telah disimpan sebelumnya.

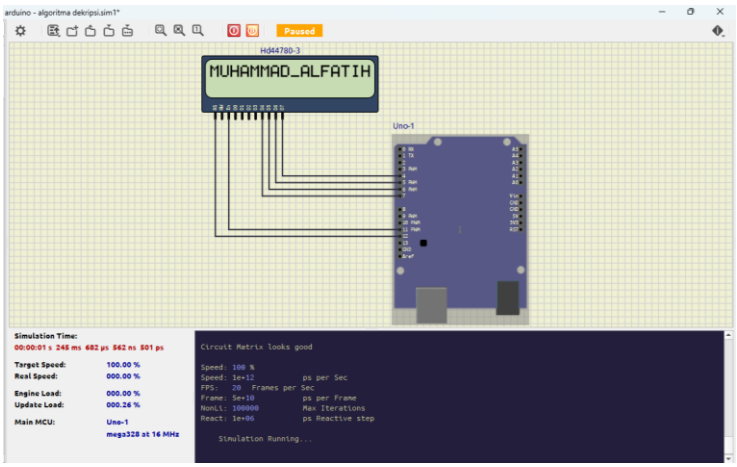
3. Setelahnya pilih source code yang telah di compile dan tersimpan dalam format .hex dan klik open.



Gambar 4.7 Tampilan library yang terdapat source code.hex dekripsi cipherteks menjadi plainteks mikrokontroler ATmega328

Jika telah menemukan source code .hex yang diperlukan dalam proses dekripsi cipherteks menjadi plainteks mikrokontroler Arduino Uno ATmega328 dan memilihnya lalu meng-klik open, maka program sistem keamanan algoritma vigenere cipher akan di load hingga selesai.

4. Setelah program selesai di loaded, kita dapat menjalankan program dengan menklik icon run.



Gambar 4.8 Tampilan hasil dekripsi cipherteks menjadi plainteks pada LCD yang di proses Mikrokontroler ATmega328

Proses komputasi mikrokontroler Arduino Uno ATmega328 dijalankan begitu icon run diklik. Maka proses dekripsi sistem keamanan algoritma vigenere cipher dari cipherteks dengan menggunakan key yang telah ditentukan, diperoleh hasil menjadi plainteks semula seperti yang ditampilkan pada layar lcd gambar diatas: “MUHAMMAD_ALFATIH”.

Dari program sistem keamanan algoritma vigenere cipher yang dibuat pada arduino ide menggunakan pemograman bahasa c yaitu enkripsi plainteks menjadi cipherteks maupun dekripsi cipherteks menjadi plainteks semula, menggunakan key yang sama pada mikorokontoler arduino uno ATmega328. Dan disimulasikan pada simulide didapatlah hasil yang sesuai atau akurat berdasarkan kalkulasi matematika algoritma vigenere cipher sehingga nantinya dapat mengamankan data komunikasi yang komputasinya sederhana.

BAB V
PENUTUP

5.1 Kesimpulan

1. Pengujian sistem keamanan algoritma vigenere cipher yang dibuat pada arduino ide menggunakan pemograman bahasa c yaitu enkripsi plainteks menjadi cipherteks yang disimulasikan simulide didapatlah hasil yang sesuai atau akurat berdasarkan kalkulasi matematika algoritma vigenere cipher.
2. Pengujian sistem keamanan algoritma vigenere cipher yang dibuat pada arduino ide menggunakan pemograman bahasa c yaitu dekripsi cipherteks menjadi plainteks yang disimulasikan simulide didapatlah hasil yang sesuai atau akurat berdasarkan kalkulasi matematika algoritma vigenere cipher.
3. Sistem keamanan algoritma vigenere cipher dapat diterapkan pada mikrokontroler arduino uno ATmega328.

5.2 Saran

Jika mikrokontroler yang digunakan untuk komunikasi data cukup mumpuni yaitu memiliki spesifikasi kemampuan komputasi yang tinggi maka dapat menggunakan algoritma yang lebih rumit sehingga akan lebih kuat dalam mengamankan komunikasi data.

DAFTAR PUSTAKA

Allgo. (2017, Oktober 26). *apa-itu-arduino-ide-dan-arduino-sketch*. Retrieved Januari 09 2022, from allgoblog.com: http://allgoblog.com/apa-itu-arduino-ide-dan-arduino-sketch/

Aripin, S., & Syahrizal, M. (2022). Analisis Modifikasi Algoritma Kriptografi Klasik Menggunakan Algoritma Blum-Micali Generator. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, *6*(1), 136-147.

Asang, M. S., & Sembiring, I. (2017). Keamanan Data Pada Perangkat Internet Of Things Menggunakan Metode Public-Key Cryptography. *AITI*, *14*(1), 80-87.

Babar, M., & Sohail Khan, M. (2021). ScalEdge: A framework for scalable edge computing in Internet of things–based smart systems. *International Journal of Distributed Sensor Networks*, *17*(7), 15501477211035332.

Badrul, M., & Akmaludin. (2019). Pengertian jaringan komputer, Manajemen Jaringan. PT Elex Media Komputindo. Jakarta.

Bhandari, R., & Kirubanand, V. B. (2019). Enhanced encryption technique for secure iot data transmission. *International Journal of Electrical and Computer Engineering*, *9*(5), 3732.

Dharma, I. P. L., Tansa, S., & Nasibu, I. Z. (2019). Perancangan Alat Pengendali Pintu Air Sawah Otomatis dengan SIM800l Berbasis Mikrokontroler Arduino Uno. *Jurnal Teknik*, *17*(1), 40-56.

Effendi, H., & Puspitaningrum, R. (2021). Rancang Bangun Sistem Monitoring Pemakaian Air Pam Dan Mutu Air Pada Komplek Perumahan Dengan Jaringan Nirkabel Lora Berbasis Arduino Uno. *Sinusoida*, *23*(1), 50-60.

Endrayanto, R. K., Muttaqin, A., & Setyawan, R. A. (2019). Advanced Encryption Standard (AES) pada Modul Internet of Things (IoT). *TELKA-Jurnal Telekomunikasi, Elektronika, Komputasi dan Kontrol*, *5*(2), 103-113.

Fathulrohman, Y. N. I., & Saepulloh, A. (2019). Alat Monitoring suhu dan kelembaban menggunakan arduino uno. *Jurnal Manajemen Dan Teknik Informatika (JUMANTAKA)*, *2*(1).

Hanisadewa, T., Viananta, T. Y., & Primawan, A. B. (2019). Unjuk Kerja Jaringan Sensor Nirkabel Dengan Menggunakan Topologi Star. In *Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO)* (Vol. 1, pp. 131-138).

Haris, C. A., & Ariyus, D. (2020). Kombinasi dan Modifikasi Vigenere Cipher dan Hill Cipher Menggunakan Metode Hybrid Kode Pos, Trigonometri, dan Konversi Suhu Sebagai Pengamanan Pesan. *Inform. Mulawarman J. Ilm. Ilmu Komput*, *15*(2), 90.

Hutabarat, N. F. R., Sirait, R., & Napitu, D. H. S. (2021). Analisa Unjuk Kerja nRF24l01 Pada Komunikasi Multi Hop Dengan Database Lokal. *Teknologi Rekayasa Jaringan Telekomunikasi*, *1*(2), 97-106.

Jimmy, M., & Christianto, Y. Y. (2019). pembuatan pengendali manual nirkabel untuk automated guided vehicle (agv) menggunakan modul radio zig-bee di pt. astra otoparts divisi winteq. *Technologic*, *2*(1).

Kurniawan, M. F., & Budiarto, S. P. (2019). Peningkatan Performa Jaringan Internet Rt Rw Net di Dusun Warengan Desa Bubuk Menggunakan Metode Hierarchical Token Bucket. *Jikom: Jurnal Informatika dan Komputer*, *9*(1), 72-90.

Lestari & Permana (2019). Pengertian jaringan komputer. Edisi revisi Bandung : Informatika.

Madakam, S. (2015, Desember 11). Internet of Things: Smart Things. *International Journal of Future Computer and Communication, 4*(4), 250-253. doi:10.7763/IJFCC.2015.V4.395

Noris,Shandi and Ardhianyah,Maulana and Octaviano, Alvino (2020) Komunikasi Data. Unpam Press, Tangerang Selatan. ISBN 978-623-7833-60-4

Ramady, G. D. (2020, October). Perancangan Model Simulasi Sistem Pengendali Suhu Ruang Kelas Berbasis Internet of Things. In *SENASTER" Seminar Nasional Riset Teknologi Terapan"* (Vol. 1, No. 1).

Reswan, Y., & Yuliansyah, B. T. (2018). Implementasi Kompilasi Algoritma Kriptografi Transposisi Columnar Dan Rsa Untuk Pengamanan Pesan Rahasia. *Jurnal Informatika Upgris*, *4*(2).

Ri2M (2010). Network Security Essentials: Applications and Standards (6th ed.). Pearson.

Roman, R., Najera, P., & Lopez, J. (2011). Securing the internet of things. *Computer*, *44*(9), 51-58.

Skirelis, J., & Navakauskas, D. (2017, November). Edge computing in IoT: Preliminary results on modeling and performance analysis. In *2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (pp. 1-4). IEEE.

Sutarti, S., Siswanto, S., & Subandi, A. (2018). Implementasi Dan Analisis QoS (Quality of Service) Pada VoIP (Voice Over Internet Protocol) Berbasis Linux. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, *5*(2).

Tashia. (2015, Desember 11). *dari-internet-of-thing-menuju-smart-city-dan-smart-people*. Retrieved Januari 07, 2022, from aptika.kominfo.go.id: https://aptika.kominfo.go.id/2015/12/dari-internet-of-thing-menuju-smart-city-dan-smart-people/

Widiyanto, S., Adnan, G., Fatkuroji, M., Handoyo, D. W., & Hasan, M. A. (2021). Pengamanan Pesan Text dengan menggunakan Kriptografi Klasik Metode Shift Chipper dan Metode Substitution Chipper. *RJOCS (Riau Journal of Computer Science)*, *7*(1).

Wei, H., Luo, H., Sun, Y., & Obaidat, M. S. (2019). Cache-aware computation offloading in IoT systems. *IEEE Systems Journal*, *14*(1), 61-72.

Yulianti, T., Samsugi, S., Nugroho, P. A., & Anggono, H. (2021). Rancang Bangun Pengusir Hama Babi Menggunakan Arduino dengan Sensor Gerak. *Jtst*, *2*(1), 21-27.

Zamara, S. (2019). Penerapan Algoritma Vegenere Cipher Dan Vernam Cipher Dalam Penggamanan File Text. *JURIKOM (Jurnal Riset Komputer)*, *6*(3), 326-332.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
UNIVERSITAS SERAMBI MEKKAH
FAKULTAS TEKNIK
Jalan Tgk. ImumLueng Bata, Batoh, Banda Aceh, Kode Pos 23245
Website :www.serambimekkah.ac.id,Surel : akademik@serambimekkah.ac.id

SURAT KEPUTUSAN
DEKAN FAKULTAS TEKNIK UNIVERSITAS SERAMBI MEKKAH
Nomor: 004/FT-USM/SK/I/2024

Tentang
PENUNJUKAN DOSEN PEMBIMBING TUGAS AKHIR
FAKULTAS TEKNIK UNIVERSITAS SERAMBI MEKKAH

DEKAN FAKULTAS TEKNIK UNIVERSITAS SERAMBI MEKKAH

Menimbang :
1. Bahwa untuk kelancaran penulisan Tugas Akhir mahasiswa Fakultas Teknik Tahun Akademik 2021/2022 perlu adanya program bimbingan yang kontinyu dan intensif kepada mahasiswa sebagai peserta didik.
2. Bahwa untuk keperluan tersebut, perlu ditunjuk dosen pe,bimbing Tugas Akhir dengan suatu surat keputusan

Mengingat :
1. Undang-undang No.20 Tahun 2003 tentang Sistem Pendidikan Nasional;
2. Undang-undang Nomor 14 tahun 2005 tentang Guru dan Dosen;
3. Peraturan Pemerintah RI No.17 Tahun 2010 tentang Pengelolaan dan Penyelenggaraan Pendidikan;
4. Peraturan Pemerintah RI No.66 Tahun 2010 tentang Perubahan Atas Peraturan Pemerintah No.17 tentang Pengelolaan dan Penyelenggraan Pendidikan
5. Permenristek-Dikti No 44 Tahun 2015 tentang Standart Nasional Pendidikan Tinggi;
6. Pedoman Akademik Universitas Serambi Mekkah Tahun 2015

MEMUTUSKAN

Menetapkan

Pertama : Menetapkan sdr/I : **YENI YANTI, ST.,MT** Sebagai Pembimbing I
TAUFIK HIDAYAT, ST.,MT Sebagai Pembimbing II

Untuk membimbing Tugas Akhir mahasiswa

Nama	:	**DICKY WAHYUDI**
NPM	:	**2014050034**
Program Studi	:	**TEKNIK KOMPUTER**

Kedua : Judul Skripsi : **PENERAPAN SISTEM KEAMANAN DATA MENGGUNAKAN ALGORITMA VIGENERE CIPHER PADA KOMUNIKASI NIRKABEL ATMEGA328**

Ketiga : Dengan ketentuan :
1. Bimbingan dilaksanakan dengan kontinyu dan bertanggung jawab serta harus diselesaikan selambat-lambatnya 1 (satu) tahun sejak keputusan ini dikeluarkan;
2. Apabila ketentuan poin 1 terlewati disebabkan oleh kelalaian mahasiswa, maka dikenakan sanksi administratif;
3. Apabila ketentuan poin 1 terlewati disebabkan oleh kelalaian pembimbing, maka akan diganti dosen pembimbing yang baru;

Keempat : Surat Keputusan ini diberikan kepada masing-masing yang bersangkutan untuk diketahui dan dilaksanakan sebagaimana mestinya.

Kelima : Surat Keputusan ini berlaku sejak tanggal ditetapkan dan apabila dikemudian hari ternyata terdapat kekeliruan dalam penetapannya atau memerlukan penyesuaian maka akan diadakan perbaikan dan perubahan sebagaimana mestinya

Banda Aceh, 02 Januari 2024
Wakil Dekan I,

Maulinda, S.SI., M.SI
NIDN.0112028001

Tembusan :
1.Ketua Prodi
2.Mahasiswa bersangkutan
3.Arsip