# Penerapan Docker Kontainer dan Kriptografi Pengamanan Data Pesan Teks Dengan Metode *Diffie Hellman.*

**Implementing Docker Container and Text Message Data Security Cryptography *Using Diffie Hellman Method***

**Febrianto1, Arita Witanti2**

1Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta, Jl. Wates Km. 10 Yogyakarta 55753, Indonesia

Email: 115111016@student.mercubuana-yogya.ac.id, 2arita@mercubuana-yogya.ac.id

# ABSTRAK

Keamanan data menjadi hal yang sangat penting pada saat ini karena banyaknya data berisikan informasi penting dan terbatas untuk diketahui orang lain. Kriptografi adalah salah satu langkah tepat dalam pengaman data informasi dan mengisolasikan semua data dalam satu wadah kontainer. Penelitian ini mencoba untuk mengimplementasikan algoritma yang sudah dikembangkan sebelumnya dan menyatukan data-data yang dibutuhkan ke dalam satu wadah dengan teknologi kontainer. Pertama dalam melakukan penelitian ini adalah mengidentifikasi masalah dan akan dilanjutkan dengan mempelajari tentang kriptografi dan teknologi kontainer. Ide utama dari penelitian ini adalah membangun sebuah sistem yang sangat aman dengan mengisolasi setiap data maupun wadah yang akan digunakan dalam pengembangan. Hasil penelitian ini menunjukan proses algoritma kriptografi dalam mengamankan setiap pesan teks dengan memproses beberapa karakter, kalimat dan paragraf di dalam wadah *Docker* yang sudah diisolasi.Dari penelitian ini dapat diambil kesimpulan bahwa kriptografi ini mengamankan data dari pesan teks dengan beberapa kali pengujian dan diketahui bahwa algoritma kriptografi ini mempunyai hasil enkripsi yang baik. Algoritma ini juga dijalankan dalam wadah yang sudah diisolasi namun spesifikasi server mempengaruhi kecepatan proses wadah yang dijalankan.

**Kata kunci**: *Diffie Hellman; Docker; Keamanan Data; Kriptografi; Wadah.*

# ABSTRACT

Data security is very important since much data contains important and limited information for other people to know. Cryptography is one of the right steps in securing information data and isolating all data in one container. This study tried to implement an algorithm developed previously and united the required data into one container with container technology. The first step in conducting this study was identifying the problem and continuing learning about cryptography and container technology. The main idea of ​​this study was to build a very secure system by isolating every data or container used in development. The study results showed the process of cryptographic algorithms securing each text message by processing several characters, sentences and paragraphs in an isolated Docker container. This study concluded that this cryptography secured data from text messages through several tests, and it was known that this cryptographic algorithm had good encryption results. This algorithm was also run in an isolated container, but the server specifications affected the speed of the run container process.

**Keywords**: *Container; Cryptography; Data Security; Diffie Hellman; Docker.*

## 1. PENDAHULUAN

**Penerapan Docker Kontainer Dan Kriptografi Pengamanan Data Pesan Teks Dengan Metode *Diffie Hellman.***

Perkembangan teknologi informasi semakin memudahkan penggunanya dalam berkomunikasi melalui bermacam-macam media. Tidak sedikit orang memanfaatkan kemajuan teknologi informasi untuk berkomunikasi dengan keluarga, teman, bahkan hingga dosen. Informasi dan data dapat dengan mudah diproses melalui jaringan komputer. Keutuhan suatu sistem merupakan sesuatu yang sangat penting. Hal ini tentu saja menimbulkan resiko jika ada pelaku kejahatan dunia maya yang mencoba untuk mencari celah keamanan untuk mendeteksi dan memanipulasi pesan (Azizah, 2020).

Rumusan masalah dalam Penerapan Docker Kontainer Dan Kriptografi Pengamanan Data Pesan Teks Dengan Metode Diffie Hellman dapat didefinisikan dalam penelitian ini diantaranya sebagai berikut: Bagaimana mendesainkan sistem kriptografi pengamanan data pesan dengan algoritma *Diffie Hellman*?,Bagaimana mengimplementasikan system kriptografi pengamanan data pesan dengan algoritma *Diffie Hellman* dan teknologi *Docker Container*?, Bagaimana akurasi kriptografi data dengan algoritma *Diffie Hellman*?

## 2. TINJAUAN PUSTAKA

Penelitan mengenai *Performance Study On Diffie Hellman Key Exchange Algorihm*. Dalam penelitian ini menyimpulkan bahwa mengamankan lalu lintas jaringan selalu menjadi persyaratan wajib untuk jaringan apa pun aplikasi yang menggunakan saluran komunikasi yang tidak aman. Alasannya untuk memberikan perlindungan untuk data yang dikirim melalui jaringan terhadap pengungkapan dan modifikasi yang tidak sah pesan antara pihak yang berkomunikasi. Metode penelitian ini adalah algoritma *Diffie-Hellman*. Hasil penelitian ini adalah untuk mengaktifkan dua pihak untuk bertukar kunci sesi dengan aman yang kemudian dapat digunakan untuk enkripsi pesan simetris berikutnya. Namun, *Diffie-Hellman* sendiri tidak mengotentikasi komunikasi tersebut. Kemudian menjelaskan protokol pertukaran kunci yang diautentikasi dan protokol pertukaran kunci sekali jalan varian dari protokol *Diffie-Hellman* (Mishra & Jayaprakash, 2017).

Algoritma Diffie Hellman ini menyediakan cara untuk menghasilkan kunci bersama antara dua orang sedemikian rupa sehingga kunci tersebut tidak dapat dilihat dengan mengamati komunikasi. Pertukaran kunci Diffie Hellman secara umum dijelaskan dengan asumsi bahwa Alice dan Bob ingin menggunakan sandi simetris sehingga perlu menukar kunci pribadi. Mula-mula Alice dan Bob sepakat pada dua bilangan g atau base dan p atau bilangan primitif dengan 𝑔 < . Bob dan Alice menghitung bilangan bulat rahasia mereka dengan :

|  |  |  |
| --- | --- | --- |
| 𝛼⏟ = 𝑔 𝑎 𝑚𝑜𝑑 𝑝  Perhitungan Alice | dan  𝛽⏟ = 𝑔 𝑏 𝑚𝑜𝑑 𝑝  Perhitungan Bob | (1) |

Alice yang sudah memilih kunci private dan menghitung 𝛼 = 𝑔𝑎 𝑚𝑜𝑑 𝑝. Alice kemudian mengirimkan 𝛼 ke Bob. Sementara itu Bob mengambil kunci private 0 < 𝑏 dan menghitung 𝛽 = 𝑔𝑏 𝑚𝑜𝑑 𝑝. Bob kemudian mengirimkan 𝛽 ke Alice. Bob dan Alice sudah menemukan kuncinya masing-masing lalu kita akan beralih pada langkah terakhir Alice menghitung 𝛼′ = 𝛽𝑎 𝑚𝑜𝑑 𝑝 dan Bob menghitung 𝛽′ = 𝛼𝑏 𝑚𝑜𝑑 𝑝.

𝛼⏟′ 𝛽 𝑜𝑑 𝑝 dan (2)

Perhitungan 𝛽⏟′ 𝛼 𝑜𝑑 𝑝

Alice Perhitungan Bob

Keduanya mendapatkan nilai k yang sama kemudian dapat digunakan sebagai kunci rahasia.

𝛼′ 𝑎 𝑏(3)

(𝑚𝑜𝑑 𝑝)

Nilai umum ini adalah kunci pertukaran mereka. Algoritma Diffie Hellman dapat kita lihat pada Tabel 2.1.

Tabel 2.1 Diffie Hellman Key Exchange

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | | | | |
| Pihak yang telah dipilih membangkitkan nilai p dan integer g | | | | |
| Alice |  | Bob |  | |
| Bilangan rahasia a | bulat | Bilangan rahasia b | bulat | |
| Menghitung 𝛼 =  𝑔𝑎 𝑚𝑜𝑑 𝑝 | | Menghitung  𝑔𝑏 𝑚𝑜𝑑 𝑝 | | 𝛽 = |
| Pertukaran Kunci Publik  Alice mengirim 𝛼 ke Bob  → 𝛼  𝛽 ← Bob mengirimkan 𝛽 ke Alice | | | | |
| Perhitungan Kunci Rahasia | | | | |

Tabel 2.2 Perhitungan Diffie-Hellman

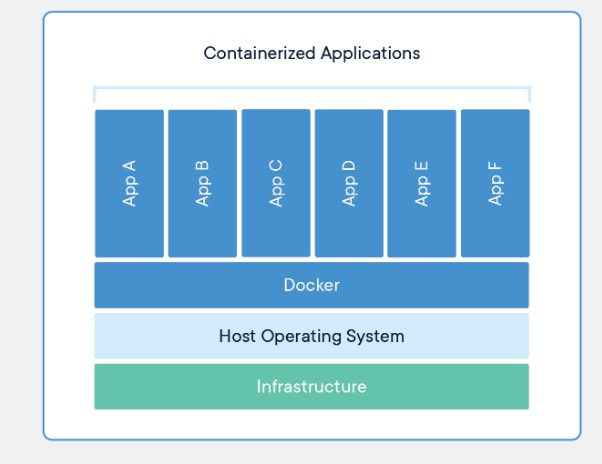
|  |  |  |
| --- | --- | --- |
| Alice | Bob |  |
| Menghitung nilai a  Menghitung 𝛼′ =  𝛽𝑎 𝑚𝑜𝑑 𝑝 | Bilangan rahasia b Menghitung  𝛼𝑏 𝑚𝑜𝑑 𝑝 | bulat  𝛽′ = |
| Nilai kunci rahasia 𝛼′ 𝑎  𝑔𝑎𝑏 (𝑚𝑜𝑑 𝑝) | | |

Protokol Diffie – Hellman tidak menyediakan otentikasi pihak yang berkomunikasi dan akibatnya rentan terhadap serangan orang-di-tengah. Kerentanan ini dapat dihindari dengan penggunaan otentikasi mekanisme seperti tanda tangan digital dan sertifikat kunci public

## Teknologi Kontainer

Dalam mengembangkan aplikasi saat ini membutuhkan lebih dari sekadar menulis kode. Berbagai bahasa, kerangka kerja, arsitektur, dan antarmuka terputus-putus antar alat untuk setiap tahap siklus hidup menciptakan kompleksitas

yang sangat besar



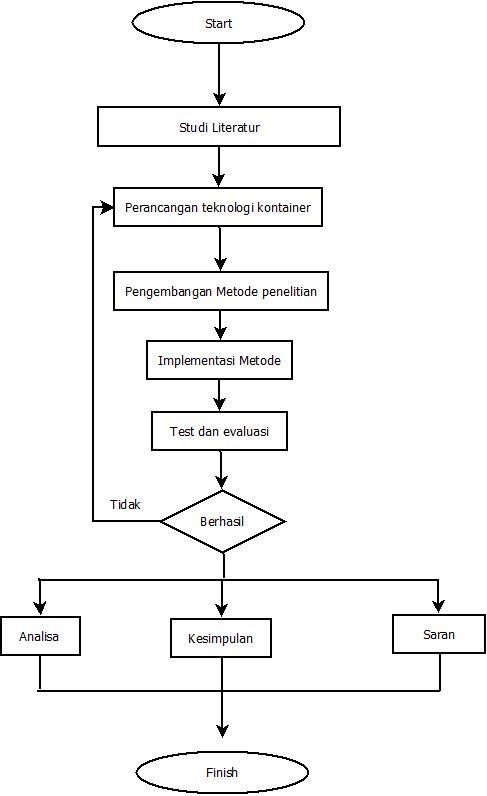
Gambar 2.1 Arsitektur Docker (Docker.inc, 2020)

## 3. METODOLOGI PENELITIAN

Bahan yang digunakan pada penelitian ini adalah menggunakan Docker sebagai metode wadah aplikasi algoritma *Diffie Hellman* sebagai metode enkripsi pesan biasa. Setiap aplikasi *web server*, *database* dan bahasa pemograman yang kita gunakan nanti akan dimasukkan kedalam satu wadah begitu juga dengan program kode algoritma enkripsi pesan teks akan berjalan didalam wadah container yang telah dibuat. Dalam program enkripsi menggunakan metode yang telah dikembangkan sebelumnya yang telah diproses itu dinamakan enkripsi dan akan didkembalikan seperti semulanya yang akan dinamakan deksripsi.

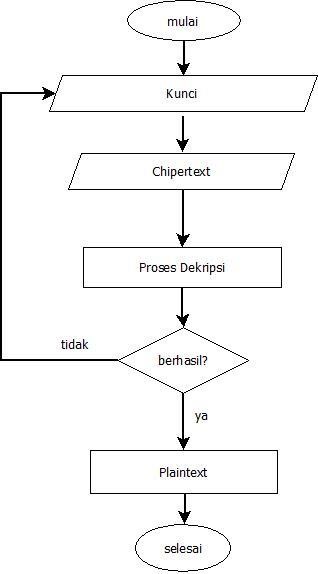
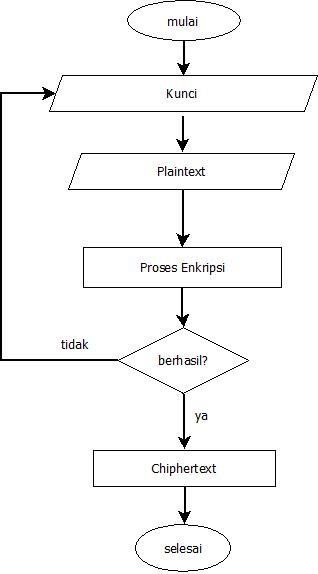
Alat penelitian diantaranya perangkat keras yaitu Laptop Asus, Processor Intel I3 @2.6GHz, 6.00 GB RAM, SSD 128 dan perangkat lunak yaitu sistem operasi Linux Server, Dia Diagram, Docker Container, PHP versi 7.4, Javascript.

## Jalan Penelitian



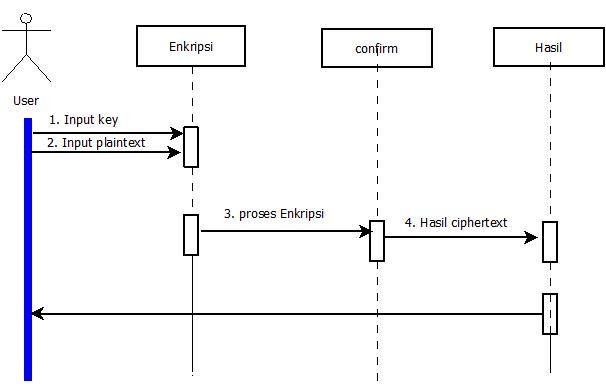
Gambar 3.1 Tahapan Penelitian

## Desain Sistem



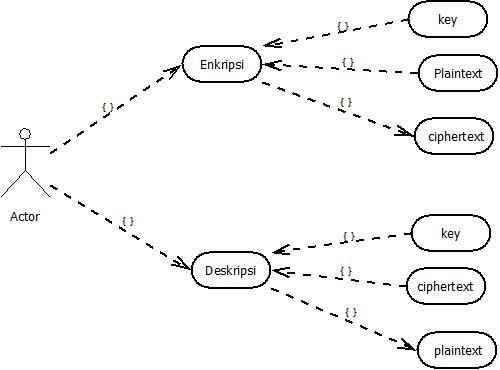
Gambar 3.3 Flowchart Enkripsi dan Deskripsi.

## Sequence Diagram



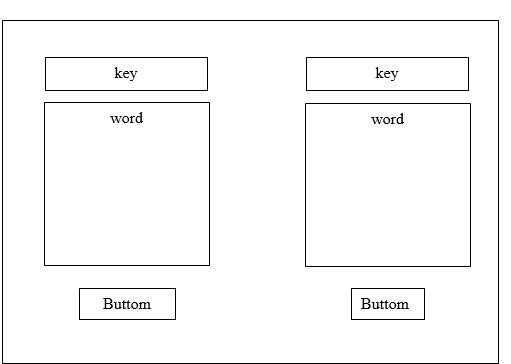
Gambar 3.6 Sequence diagram enkripsi

# Unified Modeling Language



Gambar 3.4 Use Case diagram enkripsi dan deskripsi

## Rancangan Antar Muka



Gambar 3.7 Rancangan antar muka enkripsi deskripsi

# Docker

Tahap implementasi teknologi docker adalah sebagai berikut : Langkah pertama : Membuat directori

$ cd /home/localhost/skripsi

Langkah Kedua: Membuat Config Nginx

$ mkdir etc/nginx

Selanjutnya buatlah default.template.conf

$ vim

etc/nginx/default.template.conf

Lalu tambahkan kode berikut kedalam default.template.conf

|  |
| --- |
| # Nginx configuration  server { listen 80 default\_server; listen [::]:80 default\_server;  server\_name localhost};    index index.php index.html;  error\_log  /var/log/nginx/error.log; access\_log  /var/log/nginx/access.log; root /var/www/html; } |

Listen disini berfungsi sebagai mendefinisikan port tempat server dan mendengarkan permintaan masuk. Error\_log dan access\_log berfungsi sebagai mengarahkan kemana webserver akan menulis file log. Root berfingsi sebagai menetapkan jalur folder root web itu sendiri.

Langkah ketiga : Konfigurasi PHP

Sebelum kita menentukan infrastruktur docker-compose.yml kita dapat mengkonfigurasikan PHP untuk sebagai pembatas dari permintaan webserver nginx.

Buatlah direktori PHP dengan :

$ mkdir etc/php

Selanjutnya menambahkan file php.ini

$ vim etc/php/php.ini

Lalu tambahkan kode berikut ke php ini kita

;PHPStorm [Xdebug] xdebug.remote\_enable=1 xdebug.idekey=PHPSTORM xdebug.profiler\_enable=0 xdebug.max\_nesting\_level=700 xdebug.remote\_host=192.168.100.14 xdebug.remote\_port=9000

Pada xdebug.remote\_host masukkan ip kalian, ini berfungsi sebagai meremote module xdebug.

Langkah keempat : Membuat dockercompose.yml

Membangun project aplikasi kita dengan docker compose dengan menyedehanakan proses pengaturan pembuatan infrastruktuk kita. Untuk menyiapkannya kita hanya perlu melakukan perintah berikut :

$ vim docker-compose.yml

Dalam file docker-compose kita akan medefinisikan tiga layanan yaitu webserver, database, bahasa pemograman dan managemen database. Tambahkan kode berikut ke file docker-compose.yml

|  |
| --- |
| version: '3' services: web: image: nginx:alpine volumes:  -  "./etc/nginx/default.conf:/etc/nginx/conf.  d/default.conf"   * "./web:/var/www/html"   -  "./etc/nginx/default.template.conf:/etc/ngi nx/conf.d/default.template" ports:   * "8000:80" - "3000:443" environment: |
| -  NGINX\_HOST=${NGINX\_HOST}  command: /bin/sh -c "envsubst  '$$NGINX\_HOST' <  /etc/nginx/conf.d/default.template >  /etc/nginx/conf.d/default.conf && nginx  -g 'daemon off;'" restart: always depends\_on:   * php:   image: nanoninja/php-  fpm:${PHP\_VERSION}  restart: always volumes:  -  "./etc/php/php.ini:/usr/local/etc/php/conf.  d/php.ini"   * "./web:/var/www/html" |

Layanan yang didefinisikan disini meliputi : web, didefinisikan berupa aplikasi webserver dan menggunkan image docker nginx:alpine. php, mendefinisikan layanan berupa Bahasa pemograman dan

menjalankan image custom nanoninja/phpfpm. Setiap nama kontainer mendefinisikan setiap layanan yang digunakan. Docker akan memberikan nama kontainer berdasarkan docker-compose yang sudah kita buat. Untuk memfalisitasi komunikasi antar container, layanan akan dihubungkan dengan jaringan yang bisa dibuatkan secara manual maupun secara otomatis dari docker itu sendri.

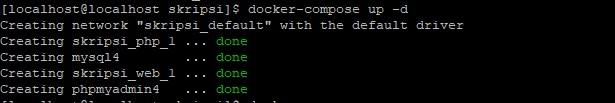
Langkah kelima : Menjalankan

Kontainer

Dengan semua layanan yang telah dibuat di docker-compose.yml, kita hanya perlu

$ docker-compose up -d

mengeluarkan satu perintah yaitu : Hasilnya



## 4. HASIL DAN PEMBAHASAN Hasil Implementasi Algoritma

Pengujian digunakan untuk melihat proses enkripsi data pesan dengan beberapa karakter dan huruf. Pesan kita nyatakan sebagai :

Demokrasi liberal (atau demokrasi konstitusional) adalah sistem politik yang menganut kebebasan individu.

Langkah pertama yang dilakukan server.

Membangkit nilai dari Diffie hellman

12

Langkah kedua memasukkan pesan

Demokrasi liberal (atau demokrasi konstitusional) adalah sistem politik yang menganut kebebasan individu.

Langkah proses ketiga mengekripsi pesan dengan algoritma enkripsi dengan mengambil kunci dan mengembalikan nilai terenkripsi dan menghasilkan

Dalam fungsi ini memetakan setiap karakter menjadi karakter lain berdasarkan panjang kunci yang diberikan. Langkah proses keempat mendeskripsikan Langkah ketiga kedalam plaintext dan masukkan kunci yang sebelumnya kita buat

12

Langkah kelima ini saya namakan proses deskripsi dari chiphertext kedalam bentuk plaintext. Masukkan Kembali hasil ciphertext sebelumnya

aAIll0OD2V9ML7mreUnDgueu8/TbPayseF 9mGikhXr5sXMkym5T1Gj8QshyCAMplGc R1nafKI4/sHpneW5Hm3wfufjO8ls0mYEH8 im/oysXUi7uLhtXWk2qbTCMdqdHVshH3 Msvo7zhO77LrHeJ9G4w=

Dalam fungsi ini memetakan setiap karakter menjadi karakter lain berdasarkan panjang kunci yang diberikan.

Langkah proses keempat mendeskripsikan Langkah ketiga kedalam plaintext dan masukkan kunci yang sebelumnya kita buat

12

Langkah kelima ini saya namakan proses deskripsi dari chiphertext kedalam bentuk plaintext. Masukkan Kembali hasil ciphertext sebelumnya

aAIll0OD2V9ML7mreUnDgueu8/TbP

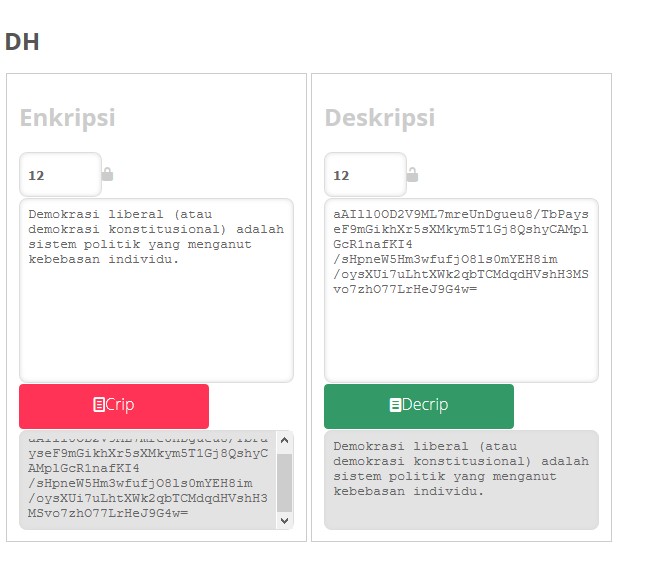
ayseF9mGikhXr5sXMkym5T1Gj8QshyCA MplGcR1nafKI4/sHpneW5Hm3wfufjO8ls0 mYEH8im/oysXUi7uLhtXWk2qbTCMdqdH VshH3Msvo7zhO77LrHeJ9G4w=

Langkah keenam ini dekripsi

Demokrasi liberal (atau demokrasi konstitusional) adalah sistem politik yang menganut kebebasan individu.

Hasil akhir pengujian enkripsi deskripsi adalah sebagai berikut :

aAIll0OD2V9ML7mreUnDgueu8/TbPayseF 9mGikhXr5sXMkym5T1Gj8QshyCAMplGc R1nafKI4/sHpneW5Hm3wfufjO8ls0mYEH8 im/oysXUi7uLhtXWk2qbTCMdqdHVshH3 Msvo7zhO77LrHeJ9G4w=



Gambar 4.5 Hasil enkripsi dan deskripsi

# Pengujian Algoritma

Tabel 4.1 Hasil Pengujian



## 4. KESIMPULAN

Pada penelitian ini berhasil dilakukan berdasarkan pada enkripsi hasil dari implemtasi dengan memproses pesan dengan beberapa karakter dan paragraf. Enkrisi disini mengolah data pesan utuh menjadi pesan yang tidak bisa dibaca. Proses enkripsi didasarkan pada proses analisis, sedangkan proses deskripsi didasarkan pada proses sintesis. Berdasarkan analisis hasil uji coba diketahui bahwa algoritma kriptografi ini mempunya nilai korelasi plaintext dan chiphertext. Algoritma ini juga mempunyai nilai kualitas enkripsi yang baik. Dari perhitungan waktu eksekusi dan analisis diketahui bahwa kecepatan teknologi wadah dalam menjalankan wadah bergantung pada spesifikasi server. Semakin tinggi spesifikasi server maka semakin proses wadah yang dijalankan. Algoritma kriptografi ini dalam prosesnya hanya melibatkan proses pertukaran kunci antara satu dengan yang lainnya, sehingga pesan hanya bisa dikirimkan melalui orang yang mempunya kunci publik tersebut.

## 5. UCAPAN TERIMA KASIH

Terselesaikannya penyusunan dan

penulisan skripsi ini tidak terlepas dari bantuan, bimbingan, dan motivasi dari berbagai pihak. Dengan segala kerendahan hati, penulis ingin menyampaikan terima kasih kepada: Orang tua serta keluarga penulis yang telah banyak memberikan dukungan, baik moral maupun material, doa-doa, semangat dan motivasi sehingga penulis bisa menyelesaikan studi S1 ini.

Terima kasih juga kepada Arita Witanti, S.T. M.Eng., selaku dosen pembimbing skripsi yang telah membimbing penulis dalam menyusun skripsi ini. Seluruh dosen dan staf Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta yang telah mengajar dan membantu penulis selama menuntut ilmu di Universitas Mercu Buana Yogyakarta. Semua teman-teman di Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta.

## DAFTAR PUSTAKA

Adinta, F., & Neforawati, i. (2017). *Rancang Bangun Aplikasi Chatting Berbasis Web Menggunakan Docker*, 1-68.

Alam, H., Widya, H., Syafrawali, & Pasaribu, P. K. (2019).

SEMNASTEK UISU. *Rancang*

*Bangun Aplikasi Penyandian Data*

*Text Menggunakan Algoritma Diffie\_Hellman dan Algoritma RC4*, 200-206.

Docker.inc. (2020). *Docker*. Diambil kembali dari Docker.com:

https://www.docker.com/

Kharbanda, K., & Kaur, K. (2019). *Performance Study of Applications Using Dockers Container*.

Mishra, M. R., & Jayaprakash. (2017). International Journal of Pure and

Applied Mathematics. *A Study On Diffie-Hellman Key Exchange Protokols*, 179-189.

Santosa, F. C., Data, M., & Nurwarsito, H.

(2018). *Perancangan Klaster Sistem Penguji Kode Program Menggunakan Docker*, 5659-5666.

8 *Jurnal Multimedia & Artificial Intelligence, Volume X, Nomor X, Februari 2019*