PERANCANGAN APLIKASI ENKRISI DAN DESKRIPSI TEKS MENGGUNAKAN ALGORITMA PLAYFAIR CIPHER BERBASI WEB

TUGAS AKHIR

Untuk Memenuhi Persyaratan Memperoleh Gelar Ahli Madya Pada Program Studi Manajemen Informatika Fakultas Sains dan Teknologi Universitas Labuhanbatu

OLEH :

MUHAMMAD PRISTIWANTO
16.051.00.031

PROGRAM STUDI MANAJEMEN INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS LABUHANBATU
RANTAUPRAPAT

2019

LEMBARAN PENGESAHAN/PERSETUJUAN SKRIPSI

JUDUL TUGAS AKHIR : PERANCANGAN APLIKASI ENKRIPSI DAN DESKRIPSI TEKS MENGGUNAKAN ALGORITMA PLAYFAIR CIPHER BERBASIS WEB

NAMA : MUHAMMAD PRISTIWANTO

NPM : 16.051.00.031

PRODI : MANAJEMEN INFORMATIKA

Disetujui Pada Tanggal : 1 Agustus 2019

Pembimbing I,	Pembimbing II
(Deci Irmayani, S.Kom,M.Kom)	(Sentosa Pohan, S.Kom.,M.Kom)
NIDN. 0127058602	NIDN. 0107128401

PERNYATAAN

Yang bertandatangan dibawah ini :

Nama	: MUHAMMAD PRISTIWANTO
NPM	: 16.051.00.031
Judul Tugas Akhir	: PERANCANGAN APLIKASI ENKRIPSI DAN DESKRIPSI TEKS MENGGUNAKAN ALGORITMA PLAYFAIR CIPHER BERBASIS WEB

Dengan ini penulis menyatakan bahwa Tugas Akhir ini disusun sebagai syarat untuk memperoleh gelar Ahli Madya pada Program Studi Manajemen Informatika Fakultas Sains dan Teknologi Universitas Labuhanbatu adalah hasil karya tulis penulis sendiri. Semua kutipan maupun rujukan dalam penulisan Tugas Akhir ini telah penulis cantumkan sumbernya dengan benar sesuai dengan ketentuan yang berlaku.

Jika di kemudian hari ternyata ditemukan seluruh atau sebagian Tugas Akhir ini bukan hasil karya penulis atau plagiat, Penulis bersedia menerima sanksi pencabutan gelar akademik yang disandang dan sanksi-sanksi lainnya sesuai dengan peraturan perundang-undangan yang berlaku.

Rantauprapat, 1 Agustus 2019
Yang Membuat Pernyataan,

Muhammad pristiwanto
16.051.00.031

Abstrak

Pada zaman sekarang ini informasi/pesan tidak hanya dikirimkan melalui kurir atau secara tradisional saja akan tetapi sudah disesuaikan dengan perkembangan teknologi. Salah satu fenomena yang terjadi karena melibatkan tenologi Internet dalam pengiriman pesan dan pertukaran data adalah adanya isu penyadapan, pemalsuan bahkan pencurian pesan. Kriptografi memiliki peran yang sangat penting di era digitalisasi yang mana bertujuan untuk mengamankan informasi. Informasi yang bersifat privasi dapat terhindar dari orang ketiga dan informasi yang akan disampaikan dapat dilindungi. Atas dasar pemikiran ini, perlu dibuat media atau aplikasi yang bisa digunakan untuk melakukan proses enkripsi dan deskripsi pesan sehingga pesan yang dikirimkan dapat diterima oleh penerima dalam keadaan terjamin legitimasinya. Penelitian ini menggunakan algoritma kriptografi Playfair Cipher. Dari hasil pengujian terhadap aplikasi yang dibangun, dalam melakukan proses enkripsi maupun deskripsi maka dapat terbukti menjamin kemananan maupun kerahasiaan pesan.

Kata kunci: *Kriptografi, Algoritma, Playfair Cipher..*

Abstract

In this day and age information/messages not only sent via courier or traditionally only but it's been adapted to technological developments. One of the phenomena that happen because it involves tenologi Internet in message delivery and data exchange is the issue of wiretapping, forgery even theft message. Cryptography has a very important role in the digitalization era which aims to secure the information. Information privacy interests can be spared from the third person and the information will be delivered can be protected. On the basis of this reasoning, it should be made of the media or application that can be used to perform the encryption process and a description of the message so that the message sent can be received by recipients in the State guaranteed legitimacy. This research uses the cryptographic algorithms the Playfair Cipher. From the results of testing against application built, in doing the encryption process or the description then it can be proven.

Keywords: *Cryptography, algorithms, the Playfair Cipher ..*

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah Subhanahu wa Ta'ala, atas segala nikmat iman, kehidupan, dan kesempatan mengenyam ilmu, sehingga penulis dapat menyelesaikan Tugas Akhir ini yang mana disusun sebagai salah satu persyaratan mencapai status Diploma III Manajemen Informatika di AMIK Labuhan Batu.

Pada kesempatan ini penulis mengucapkan terima kasih kepada orang tua kami Bapak Sunarto dan Ibu Nurhayati serta seluruh keluarga atas cinta dan doa terus menerus kepada kami agar kami dapat menyelesaikan Tugas Akhir ini. Ucapan terima kasih dan penghargaan sebesar-besarnya juga kami haturkan kepada:

1. Bapak Dr. H. Amarullah Nasution, SE.,MBA, Selaku Ketua Yayasan Universitas Labuhanbatu.
2. Ibu Deci Irmayani, S.Kom.,M.Kom, Selaku Direktur AMIK Labuhanbatu.
3. Bapak Ronal Watrianthos, S.Kom.,M.Kom, Selaku Pembantu Direktur I AMIK Labuhanbatu.
4. Ibu Marnis Nasution, S.Kom,M.Kom , Selaku Ketua Program Studi Manajemen Informatika
5. Ibu Deci Irmayani, S.Kom.,M.Kom, Selaku Dosen Pembimbing I yang telah membimbing penulis dari awal penulisan hingga akhir.
6. Bapak Sentosa Pohan, S.Kom.,M.Kom Selaku Dosen Pembimbing II yang telah membimbing penulis dari awal penulisan hingga akhir.

7. Bapak/Ibu Dosen serta staff AMIK Labuhanbatu yang telah memberikan ilmu pengetahuan kepada penulis.
8. Seluruh teman-teman seperjuangan di kelas MI-A atas ilmu, pengalaman, dan persahabatan yang diberikan kepada kami.

Tidak ada gading yang tak retak, tidak ada karya yang sempurna, karena kesempurnaan hanya milik Allah semata. Akhir kata, semoga Tugas Akhir ini menjadi amal bagi penulis, amin ya rabbala'lamin.

Rantauprapat, .. …. 2019
Penulis,

Muhammad Pristiwanto

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

DAFTAR RIWAYAT HIDUP

DATA DIRI

- Nama : MUHAMMAD PRISTIWANTO
- Tempat, Tanggal Lahir : Aek Buru, 16 Juli 1996
- Alamat : Dsn. Ranto Kapal, Desa Tanjung Mulia Kecamatan Kampung Rakyat Labuhanbatu Selatan
- No. Telp : 0822-9449-0992
- Jenis Kelamin : Laki - laki
- Agama : Islam
- Status : Belum Menikah
- Kewarganegaraan : Warga Negara Indonesia

RIWAYAT PENDIDIKAN

Formal

- 2004 - 2010 : SDN Negeri No. 112236 Kecamatan Kampung Rakyat
- 2010 - 2013 : SMPN 1 Kecamatan Kampung Rakyat
- 2013- 2016 : SMK Swasta PGRI 17 LOHSARI
- 2016 – 2019 : Universitas Labuhanbatu, Kabupaten Labuhanbatu.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Berkat perkembangan teknologi yang begitu pesat memungkinkan manusia dapat berkomunikasi dan saling bertukar informasi secara jarak jauh. Antar kota antar wilayah antar negara bahkan antar benua bukan merupakan suatu kendala lagi dalam melakukan komunikasi dan pertukaran informasi. Seiring dengan itu tuntutan akan sekuritas (keamanan) terhadap kerahasiaan informasi yang saling dipertukarkan tersebut semakin meningkat. Begitu banyak pengguna seperti departemen pertahanan, suatu perusahaan atau bahkan individu-individu tidak ingin informasi yang disampaikannya diketahui oleh orang lain atau kompetitornya atau negara lain. Oleh karena itu dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan informasi atau dikenal dengan istilah Kriptografi.

Kriptografi merupakan salah satu cara untuk mengamankan informasi, yaitu dengan menyandikan pesan asli (*plaintext*) ke dalam bentuk pesan rahasia (*ciphertext*). Proses pengamanan ini melibatkan algoritma dan kunci. Kunci enkripsi dapat dengan mudah mengembalikan *plaintext* dari *ciphertext* [1].

Dalam kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi. Enkiripsi adalah proses mengubah pesan asli (*plaintext*) menjadi pesan yang disandikan (*ciphertext*) berdasarkan metode yang telah ditentukan yang mana

proses enkripsi bekerja dengan kunci untuk mengkonversi *plaintext* ke dalam *ciphertext*. Dekripsi adalah proses mengembalikan pesan yang disandikan (ciphertext) menjadi pesan asli (plaintext) sehingga informasi tersebut terjaga kerahasiaannya pada saat sampai ke tujuan yang mana proses dekripsi bekerja dalam urutan terbalik[2].

Playfair Cipher merupakan suatu algoritma kriptografi klasik yang termasuk ke dalam polygram cipher, dimana *plaintext* diubah menjadi bentuk poligram dan proses enkripsi dekripsi dilakukan untuk poligram tersebut. Kunci kriptografinya adalah 25 buah huruf yang disusun di dalam bujursangkar 5x5 dengan menghilangkan huruf J dari abjad. Kemungkinan kuncinya adalah 25!. Pada umumnya, kunci yang digunakan adalah serangkaian kata yang mudah dimengerti. *Playfair cipher* memiliki mekanisme mengganti J dengan I.

Dari permasalahan yang terjadi, maka penulis membuat **PERANCANGAN APLIKASI ENKRIPSI KATA MENGGUNAKAN ALGORITMA *PLAYFAIR CIPHER* BERBASIS WEB,** yang diharapkan dapat menambah ilmu pengetahuan kepada pembaca dan menambah keamanan dari sebuah informasi yang bersifat rahasia.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut dapat dibuat suatu rumusan masalah, yaitu:

1. Bagaimana melakukan proses enkripsi dan deskripsi menggunakan algoritma *Playfair Cipher*?
2. Bagaimana menambahkan mekanisme menggati huruf J dengan tidak hanya huruf I ?
3. Bagaimana membangun aplikasi enkripsi kata dengan algoritma *Playfair Chiper* berbasis web?

1.3 Batasan Masalah

Adapun yang menjadi pembatasan masalah adalah sebagai berikut :

1. Algoritma kriptografi yang digunakan adalah *Playfair*.
2. Bahasa pemrograman yang digunakan adalah Web PHP.
3. Data yang dienkripsi dan dideskripsi berupa teks/kata.

1.4 Tujuan Penelitian

Tujuan dari penulisan tugas akhir ini adalah.

1. Melakukan prose enkripsi dan deskripsi menggunakan algoritma *Playfair Cipher*.
2. Menambahkan mekanisme mengganti huruf J dengan huruf selain huruf I.
3. Membangun aplikasi enkripsi kata menggunakan algoritma *Playfair Chiper* berbasis web php.

1.5 Sistematika Penulisan

Adapun sistematika penulisan dari tugas akhir ini adalah :

BAB I PENDAHULUAN

Bab ini berisikan penjelasan mengenai latar belakang penulisan tugas akhir ini yang menguraikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan masalah dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini mengemukakan dasar-dasar teori yang berhubungan dan berkenaan dengan topik yang di bahas dan akan dipakai sebagai dasar dalam menganalisa dan memecahkan masalah dan menjelaskan secara singkat tentang enkripsi.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang metode pengumpulan data dan metode perancangan aplikasi yang terdiri dari rancangan input, rancangan output, dan rancangan interface.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang program yang dirancang berupa hasil program yang telah di eksekusi meliputi menu utama, input, dan output, serta membahas prosedur kerja aplikasi tentang kelemahan dan kelebihan aplikasi.

BAB V KESIMPULAN DAN SARAN

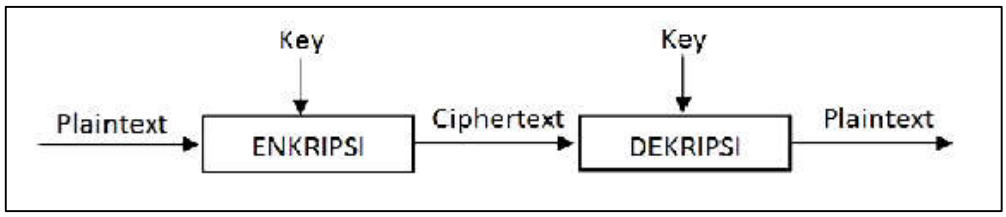
Bab ini mengemukakan kesimpulan dari pemecahan masalah dan memberikan saran terhadap perbaikan aplikasi yang digunakan saat ini.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Secara etimologis, kriptografi berasal dari bahasa Yunani, yaitu *crypto* dan *graphy*. *Crypto* artinya *secret* dan *graphy* artinya menulis. Maka, kriptografi berdasarkan dari bahasanya didefinisikan sebagai menulis secara rahasia. Kriptografi adalah sutau ilmu dan seni untuk mengamankan informasi yang berupa pesan yang terbaca (*plaintext*) menjadi pesan yang tidak bisa dibaca (*ciphertext*), sehingga hanya pengirim pesan dan penerima pesan yang dapat mengganti, menghapus dan membaca pesan tersebut. Ada dua proses pembentukkan kunci pada kriptografi, yaitu kunci simetris dan asimetris. Di mana kunci simetris memiliki kunci yang sama pada saat proses enkripsi dan dekripsi. Sedangkan, kunci asimetris memiliki kunci yang berbeda pada saat proses enkripsi dan dekripsi [1].



Gambar 2.1 Skema enkripsi dan deskripsi

Bentuk awal dari penulisan rahasia membutuhkan lebih sedikit dari implementasi penulisan sejak banyak orang tidak dapat membaca. lawan yang lebih terpelajar, membutuhkan kriptografi yang nyata. Tipe sandi klasik utama ialah *sandi transposisi*, di mana mengatur aturan huruf pada pesan (contoh 'hello

world' menajdi 'ehlol owrdl' pada skema pengubahan sederhana ini), dan sandi subtitusi, di mana secara sistematis mengganti huruf atau grup kata dengan kata lainnya dari grup kata (contoh 'fly at once' menjadi 'gmz bu podf' dengan mengganti setiap huruf dengan yang lain di alfabet Latin. Substitusi sandi pada awalnya disebut sandi Caesar, di mana setiap kata pada teks diganti degan huruf dari jumlah tetap pada posisi di alfabet.

Laporan *Suetonius* menyebutkan Julius Caesar mengunakannya untuk berkomunikasi dengan jendral-jendralnya. *Atbash* merupakan contoh dari sandi Ibrani pada mulanya. Penggunaan awal kriptografi yang diketahui merupakan teks sandi yang diukir pada batu di Mesir (1900 sebelum Masehi), tetapi teks sandi ini digunakan hanya sebagai hiburan untuk pengamat terpelajar daripada cara untuk menyimpan informasi.

Yunani kuno menyebutkan telah mengetahui sandi (contoh sandi transposisi scytale yang diklaim telah digunakan oleh militer Sparta. *Steganograf* (menyembunyikan kehadiran pesan sehingga pesan tersebut menjadi rahasia) juga pertama kali diperkenalkan pada masa kuno. Contoh awal seperti, dari *Herodotus*, menyembunyikan pesan - sebuah tato pada kepala budaknya - di bawah rambut yang kembali tumbuh. COntoh yang lebih modern dari steganografi termasuk penggunaan tinta tak tampak, mikrodot, dan tanda air digital untuk menyembunyikan informasi.

Di India, Kamasutra dari *Vātsyāyana* yang berumur 2000 tahun berbicara dengan dua jenis sandi yang berbeda yang disebut Kautiliyam dan Mulavediya. Di Kautiliyam, substitusi kata sandi berdasarkan relasi fonetik, seperti vokal menjadi konsonan. Di Mulavediya, alfabet sandi terdiri dari kata-kata yang berpasangan

dan bertimbal-balik. Teks sandi yang dihasilkan dengan *sandi klasik* (dan beberapa sandi modern) selalu mengungkapkan informasi statistik tentang teks awal, yang sering dapat digunakan untuk memecahkannya. Setelah ditemukannya *analisi frekuensi* oleh matematikawan Arab dan *polymath* Al-Kindi (juga dikenal sebagai *Alkindus*) pada abad ke-9, hampir semua jenis sandi menjadi lebih sulit dipecahkan oleh penyerang yang memiliki informasi tersebut. Seperti sandi klasik yang masih populer hingga saat ini, meskipun lebih banyak dalam bentuk puzzle. Al-Kindi menuliskan buku kriptografi yang berjudul *Risalah fi Istikhraj al-Mu'amma* (*Risalah untuk Mnejermahkan Pesan Kriptografi*), yang menjelaskan teknik analisis frekuensi kriptanalisis yang pertama kalinya.

Pada dasarnya semua sandi tetap rentan kepada kriptanalisis menggunakan teknik analisis frekuensi hingga pengembangan dari sandi polyalphabetic, yang dijelaskan oleh *Leon Battista Alberti* sekitar tahun 1467, meskipun terdapat beberapa indikasi bahwa hal ini telah terlebih dahulu diketahui oleh Al-Kindi. Penemuan Alberti menggunakan sandi yang berbeda (seperti subtitusi alfabet) untuk beberapa bagian pesan (mungkin untuk setiap teks surat berturut-turut hingga akhir). Dia juga menemukan apa yang mungkin menjadi alat sandi ototmatis untuk pertama kalinya, roda yang menerapkan pelaksanaan dari penemuannya. Pada sandi Vigenère polyalphabetic, enkripsi menggunakan *kata kunci*, yang mengatur substitusi surat berdasarkan surat mana dari kata kunci yang digunakan. Pada pertengahan abad ke-19 Charles Babbage menunjukkan bahwa sandi Vigenère sangat rentan terhadap *pemeriksaan Kasiski*, tetapi hal ini diterbitkan pertama sekali kira-kira sepuluh tahun kemudian oleh Friedrich Kasiski.

Walaupun analisis frekuensi dapat sangat kuat dan menjadi teknik umum melawan banyak sandi, enskripsi masih sangat efektif dalam penerapannya, sebagaimana banyak kriptanalisis masih khawatir akan penerapannya. Memecahkan pesan tanpa menggunakan analisis frekuensi pada dasarnya membutuhkan pengetahuan sandi dan mungkin kunci yang digunakan, sehingga membuat spionase, penyuapan, pencurian, dll. Hal ini secara tegas mengakui kerahasiaan algoritme sandi pada abad 19 sangat tidak peka dan tidak menerapkan praktik keamanan pesan; faktanya, hal ini lebih lanjut disadari bahwa setiap skema kriptografi yang memadai (termasuk sandi) harus tetap aman walaupun musuh benar-benar paham tentang algoritme sandi itu sendiri. Keamanan kunci yang digunakan harus dapat menjamin keamanan pemegang kunci agar tetap rahasia bahkan ketika diserang sekalipun. Prinsip fundamental ini pertama kali dijelaskan pada tahun 1883 oleh *Auguste Kerckhoffs* dan secara umum dikenal dengan *Prinsip Kerckhoff*; secara alternatif dan blak-blakan, hal ini dijelaskan kembali oleh *Claude Shannon*, penemu teori informasi dan fundamental dari teori kriptografi, seperti *pribahasa Shanon* - 'musuh mengetahui sistemnya'.

Alat-alat bantu yang berbeda telah banyak digunakan untuk membantu sandi. Salah satu alat paling tua yang dikenali merupakan scytale dari Yunani, tangkai yang digunakan oleh Spartan sebagai alat bantuk untuk memindahkan sandi. Pada zaman pertengahan, alat bantu lainnya ditemukan seperti *jerejak sandi*, yang juga dikenal sebagai jenis steganografi. Dengan penemuan polialfabetik, sandi menjadi lebih mutakhir dengan bantuan disk sandi milik Alberti, skema *tabula recta Johanner Trithemius*, dan silinder multi *Thomas Jefferson* (tidak banyak diketahui, dan ditemukan kembali oleh *Bazeries* sekitar tahun 1900. Banyak alat

mekanik enkripsi/dekripsi ditemukan pada awal abad ke-20, dan beberapa telah dipatenkan, di antaranya *mesin rotor* yang dikenal dengan nama mesin Enigma digunakan oleh pemerintah dan militer Jerman dari akhir tahun 1920-an dan selama Perang Dunia II. The ciphers implemented by better quality examples of these machine designs brought about a substantial increase in cryptanalytic difficulty after WWI.

Kriptografi kunci-simetris merujuk pada metode enskripsi di mana kedua pengirim dan penerima membagi kunci yang sama (atau, walaupun kuncinya tidak mirip, tetapi dapat berhubungan dengan cara komputasi sederhana). Hal ini menjadi satu-satunya jenis enkripsi yang ketahui publik hingga Juni 1976 Berkas International Data Encryption Algorithm InfoBox Diagram.svg|jmpl|Satu putaran (dari 8.5) chiper *International Data Encryption Algorithm*, digunakan pada beberapa versi *PGP (Pretty Good Privacy)* untuk enkripsi tingkat tinggi, seperti e-mail Chiper kunci simetris diimplementasikan baik itu sebagai chiper blok atau chiper stream. Sebuah block chiper enchiper masukan pada blok plainteks sebagai lawanan untuk karakter individual, bentuk masukan yang digunakan oleh chiper aliran.

Standar Enkripsi Data (SED) dan Standar Enkripsi Lanjutan (SEL) merupakan desain chiper blok yang telah ditunjuk sebagai *standar kriptografi* oleh pemerintah Amerika (walaupun penunjukan SED pada akhirnya ditarik setelah SEL diadopsi). Walaupun penarikannya sebagai standar resmi, SED (masih menjadi varian yang masih terbukti dan lebih aman) masih cukup terkenal; Hal ini digunakan oleh banyak penerapan dari enskripsi ATM hingga keamanan e-mail dan akses remote aman. Banyak chiper blok lainnya telah didesain dan dirilis,

dengan kualitas yang bervariasi. Banyak telah juga yang dihancurkan, seperti *FEAL*

Beberapa chiper, yang berbeda dengan tipe 'blok', membuat berkas panjang material kunci yang panjang, di mana dikombinasikan dengan bit-bit teks atau karakter-karakter, sedikit mirip dengan *one-time pad*. Pada chiper aliran, aliran keluarannya diciptakan berdasarkan keadaan internal yang tersembunyi yang berubah saat chiper bekerja. Keadaan internal mulanya diatur menggunakan bahan kunci rahasia. *RC4* sangat luas digunakan sebagai chiper aliran. Chiper blok dapat digunakan sebagai chiper aliran.

Fungsi hash Kriptografi merupakan algoritme kriptografi tipe ke-tiga. Fungsi ini mengambil segala panjang pesan sebagai input, dan panjang keluaran hash yang pendek dan tetap, yang dapat digunakan sebagai (sebagai contoh) tanda tangan digital. Untuk memiliki fungsi hash yang baik, penyerang tidak dapat mencari dua pesan yang dapat menghasilkan hash yang sama. MD4 merupakan fungsi hash pangjang yang sekarang telah dapat dipecahkan; MD5, varian yang lebih kuat dari MD4, sudah luas digunakan namun dapat dipecahkan saat beroperasi. Agensi keamanan nasional Amerika mengembangkan serial Algoritme Hash Aman seperti fungsi hash MD5: SHA-0 ialah algoritme cacat yang kemudian ditarik; SHA-1 digunakan secara luas dan lebih aman dari MD5, tetapi kriptanalisis telah menemukan serangan padanya; keluarga SHA-2 meningkatkan performa SHA-1, tetapi belum secara luas digunakan; dan kewenangan Amerika mengatakan hal ini cukup bijaksana dari sudut pandang keamanan untuk mengembangkan standar baru "toolkit algoritme hash NIST secara keseluruhan

untuk peningkatan kekuatan secara signifikan. Sehingga, pada tahun 2012, standar nasional Amerika memilih SHA-3 sebagai standar desain hash yang baru.

Message authentication code (MAC) hampir mirip dengan fungsi hash kriptografi, kecuali terdapat kunci rahasia yang dapat digunakan untuk membuktikan nilai hash melalui serangkaian resep kerumitan tambahan yang melindungi skema serangan algoritme penyingkat sederhana, dan dianggap cukup menguntungkan.

Kriptosistem kunci-simetris menggunakan kunci yang sama untuk enkripsi dan dekripsi sebuah pesan, walaupun pesan atau kelompok pesan dapat memiliki kunci yang berbeda dari yang lain. Kerugian yang paling signifikan dari chiper simetris ialah kebutuhan *manajerial kunci* untuk menggunakannya secara aman. Setiap sepasang pihak komunikasi yang berbeda harus, idealnya, membagi kunci yang berbeda, dan juga membagi textchiper yang berbeda juga. jumlah kunci yang dibutuhkan meningkat dua kali lipat dari jumlah anggota jaringan, yang sangat cepat membutuhkan skema manajemen kunci komplek untuk menjaga semuanya tetap konsisten dan rahasia. Kesulitan dari menciptakan kunci rahasia yang aman di antara dua pihak yang saling berkomunikasi, ialah, ketika belum adanya *jaringan aman* di antara keduanya, juga kehadiran *chicken-and-egg problem* yang dianggap menjadi tantangan praktikal untuk pengguna kriptografi di dunia nyata. (Berkas:Diffie and Hellman.jpg|jmpl|kiri|Whitfield Diffie dan Martin Hellman, penulis jurnal pertama kriptografi kunci-publik) Pada jurnal pionir tahun 1976, Whifiled Diffie dan Martin Hellman mengusulkan istilah dari kriptografi*kunci-publik* (juga, secara umum, disebut *kunci asimetris*) pada dua istilah yang berbeda namun secara matematis terdapat kunci yang berhubungan, yaitu kunci *publik* dan

kunci *privat*. Sistem kunci pablik dikonstruksikan sangat baik sehingga kalkulasi dari satu kunci ('kunci privat') secara komputasional tidak mirip dengan (kunci 'publik') walaupun secara kebutuhan mereka mirip. Malah, kedua kunci dihasilkan secara rahasia, sebagai pasangan yang tidak berhubungan. Sejarawan David Kahn menjelaskan kriptografi kunci public sebagai konsep baru paling revolusioner dalam bidang ini sejak substitusi polialfabetik yang ditemukan pada masa Renaissance. Dalam ekosistem kunci-publik, kunci publik dapat secara bebas terdistribusi, saat pasangannya kunci privat harus selalu terjaga rahasia. Pada sistem enkripsi kunci-publik, *kunci publik* digunakan untuk enkripsi, sedang *kunci privat* atau *rahasia* digunakan untuk dekripsi. Sementara Diffie dan Hellman tidak dapat menemukan sistem seperti itu, mereka menunjukkan bahwa kriptografi kunci-publik memang benar mungkin dengan menunjukkan protokol Diffie-Hellman key exchange, sebuah solusi yang sekarang digunakan secara luas dalam komunikasi aman, mengizinkan dua kelompok untuk secara rahasia membagi kunci enkripsi.

Jurnal Diffie dan Hellman menyebar luas pada dunia akademi dalam mencari sistem enkripsi kunci-publik praktis. Lalu pada tahun 1978 Ronald Rivest, Adi Shamir, dan Len Adleman, menemukan solusi yang kini dikenal sebagai algoritme RSA.

Algoritme Diffie-Hellman dan RSA, sebagai tambahan dalam menciptakan contoh algoritme kunci-publik kualitas tinggi pertama yang dikenal publik, telah sangat luas digunakan. Yang lain termasuk Kriptosistem Cramer-Shoup, Enkripsi ElGamal, dan varian Teknik kurva eliptis. Lalu, dokumen yang dipublikasikan pada tahun 1997 oleh *Government Communication Headquarters (GCHQ)*,

organisasi rahasia Inggris, mengungkapkan bahwa kriptografer di GCHQ telah mengantisipasi beberapa pengembangan akademik. Dilaporkan, sekitar tahun 1970, James H. Ellis telah memahami prinsip kriptografi kunci asimetris. Pada tahun 1973, Clifford Cocks menemukan sulisi yang esensialnya menyerupai algoritme RSA. Dan pada tahun 1974, Malcom J. Williamson diklaim telah mengembangkan algoritme pertukaran Diffie-Hellman.

Kriptografi kunci-publik dapat juga digunakan untuk mengimplementasikan skema tanda tangan digital. Tanda tangan digital berhubungan dengan tanda tangan pada umumnya; mereka memiliki karateristik yang sama dimana mudah bagi pengguna untuk membuatnya, tetapi sangat sulit bagi orang lain untuk memalsukannya. Tanda tangan digital dapat juga secara permanen mengikat pada konten pesan yang sedang ditanda tangani; mereka lalu tidak dapat 'dipindahkan' dari satu dokumen ke dokumen yang lain, dan setiap usaha akan dapat terdeteksi. Pada skema tanda tangan digital, terdapat dua algoritme: satu untuk *menandatangani*, di mana kunci rahasia digunakan untuk memproses pesan (atau hash dari pesan, atau keduanya), dan satu untuk *verivikasi,* di mana kunci publik yang sesuai digunakan dengan pesan untuk memeriksa validitas tanda tangan. RSA dan DSA merupakan dua skema tanda tangan digital yang paling terkenal. Tanda tangan digital merupakan pusat dari operasi infrastruktur kunci publik dan banyak skema keamanan jaringan lainnya (seperti Transport Layer Security, VPN, dll).

Algoritme kunci publik paling sering didasari pada teori masalah kompleksitas komputasional, sering dengan teori bilangan. Sebagai contoh, kekuatan RSA berhubungan dengan masalah faktorisasi integer, sedangkan Diffie-Hellman dan

DSA berhubungan dengan masalah logaritma diskrit. Baru-baru saja, *kriptografi kurva eliptis* telah ditemukan, sistem di mana keamanan yang didasari pada masalah teoretis bilangan yang melibatkan kurva eliptis. Dikarenakan kesulitan masalah pokok, kebanyakan algoritme kunci-publik melibatkan operasi seperti eksponensial dan perkalian aritmetika modular, di mana teknik ini secara komputasional lebih *mahal* ketimbang teknik yang digunakan pada banyak chiper blok, khususnya dengan ukuran kunci yang dibutuhkan. Hasilnya, kriptosistem kunci-publik seringnya merupakan *kriptosistem hybrid*, yang merupakan algoritme enkripsi kunci-simetris berkualitas tinggi digunakan untuk pesan itu sendiri, sedang kunci simetris yang relevan dikirimkan dengan pesan, tetapi dienkripsikan menggunakan algoritme kunci publik. Hampir sama, skema tanda tangan hybrid sering digunakan, di mana fungsi hash kriptogfari dihitung secara komputer, dan hanya hash hasil yang ditanda tangani secara digital.

Banyak karya teoritikal kriptografi berkaitan dengan kriptografi sederhana-algoritme dengan sifat kriptografi dasar-dan hubungannya pada masalah kriptografi lainnya. Alat kriptografi yang lebih sulit lalu diciptakan dari kriptografi sederhana ini. Kesederhanaan ini menyediakan sifat yang penting, yang digunakan untuk mengembangkan alat yang lebih kompleks yang disebut *kriptosistem* atau *protokol kriptografi*, yang menjamin sifat keamanan level satu atau lebih tinggi. Bagaimanapun, perbedaan antara kriptografi *sederhana*dan kriptosistem, cukup tipis; sebagai contoh, algoritme RSA kadang disebut kriptosistem, dan kadang sederhana. Contoh tipikal kriptografi sederhana termasuk fungsi pseudorandom, fungsi satu-arah, dll.

Satu atau lebih kriptografi sederhana sering digunakan untuk mengembangkan algoritme yang lebih kompleks, disebut sistem kriptografi, atau *kriptosistem*. Kriptosistem (seperti enkripsi ElGamal didesan untuk menyediakan fungsi tertentu (seperti enkripsi kunci publik) sembari menjamin sifat keamanan tertentu keamanan (seperti serangan teks-terpilih) seperti pada model oracle acak. Kriptosistem menggunakan sifat kriptografi sederhana utama untuk mendukung sifat keamanan sistem. Tentu saja, karena perbedaan antara kriptosistem dan kriptografi tidak jelas, kriptosistem yang canggih dapat diperoleh dari kombinasi beberapa kriptosistem sederhana. Pada banyak kasus, struktur kriptosistem melibatkan komunikasi maju mundur di antara dua atau lebih kelompok dalam ruangan. (seperti di antara pengirim dari pesan aman dan penerimanya) atau melewati waktu (seperti data yang dilindungi dengan kriptografi). Kriptosistem yang seperti itu disebut *protokol kriptografi*.

Beberapa kriptosistem yang terkenal termasuk *enkripsi RSA*, tanda tangan Schnorr, enkripsi El-Gamal, PGP, dll. Kriptosistem yang lebih rumit melibatkan sistem uang elektronik, sistem *tanda-tangan enkripsi*, dll. Beberapa kriptosistem *teoritikal* termasuk *sistempembuktian interaktif*, seperti *pembuktian pengetahuan-nol*), sistem untuk *pembagian rahasia*, Hingga saat ini, banyak sifat keamanan kriptosistem didemonstrasikan menggunakan teknik empirial atau menggunakan alasan ad hoc. Saat ini, terdapat upaya untuk mengembangkan teknik formal untuk menyelesaikan keamanan kriptosistem; Hal ini secara umum disebut *keamanan terbukti*. Ide umum dari keamanan terbukti ialah untuk memberikan argumen tentang kesulitan komputasional yang dibutuhkan untuk membahayakan aspek keamanan kriptosistem (dari setiap musuh).

Ilmu yang melihat seberapa baik implementasi dan integrasi kriptografi dalam penerapannya pada perangkat lunak disebut bidang tenik kriptografi dan teknik keamanan.

Kriftografi tidak hanya memberikan kerahasiaan dalam telekomunikasi, namun juga memberikan komponen-komponen berikut ini:

1. ***Authentication***. Penerimaan pesan dapat memastikan keaslian pengirimnya. Penyearang tidak dapat berpura-pura sebagai orang lain.
2. ***Integrity***. Penerima harus dapat memeriksa apakah pesan telah dimodifikasi di tengah jalan atau tidak. Seorang penyusup seharusnya tidak dapat memasukkan tambahan ke dalam pesan, mengurangi atau mengubah pesan selama data berada dijalanan.
3. ***Non Repudiation***. Pengirim seharusnya tidakdapat mengelak bahwa dialah pengirim pesan yang sesungguhnya. Tanpa kriptografi, seseorang dapat mengelak bahwa dialah pengirim e-mail yang sesungguhnya.
4. ***Authority***. informasi yang berada pada system jaringan pada seharusnya hanya dapat dimodifikasi oleh pihak yang berwenang.modifikasi yang tidak diinginkan, dapat berupa penulisan tambahan pesan, pengbahan isi, pengubahan status, pengapusan, pembuatan pesan baruh, pemalsuan,atau menyalin pesan untuk digunakan kemudian oleh penyerang. Terdapat persyaratan penting bagi interaksi di dunia nyata. Seseorang yang mempunyai identitas diri, baik berupa KTM, SIM atau passport diharapkan bahwa identitas diri itu memang sah dan benar isinya. Inilah yang diberikan oleh otentikasi,integritas dan non repudiation[2].

Proses kriptografi terdiri dari enkripsi dan dekripsi. Proses enkripsi adalah proses konversi *plaintext* ke dalam bentuk *ciphertext*, dan sebaliknya untuk proses dekripsi yang mana proses konversi *ciphertext* ke dalam bentuk *plaintext*. Parameter yang perlu dilakukan adalah pembentukkan kunci yang mana kunci tersebut digunakan untuk tranformasi proses enkripsi dan dekripsi. Salah satu algoritma kriptografi yang memiliki kunci simetris adalah *playfair cipher* [1].

2.2 Algoritma

Algoritma merupakan fondasi yang harus dikuasai oleh setiap mahasiswa yang ingin menyelesaikan suatu masalah secara terstruktur, efektif, dan efisien.

Definisi Algoritma:

1. Teknik penyusunan langkah – langkah penyelesaian masalah dalam bentuk kalimat dengan jumlah kata terbatas tetapi tersusun secara logis dan sistematis.
2. Suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan mengguanakan langkah – langkah tertentu dan terbatas jumlahnya.
3. Susunan langkah yang pasti, yang bila diikuti akan mentransformasi data input menjadi output yang berupa informasi[3].

2.3 Algoritma Kriptografi

Algoritma kriptografi berdasarkan jenis kunci yang digunakan dapat dibedakan menjadi dua jenis yaitu :

1. Algoritma simetris Dimana kunci yang digunakan untuk proses enkripsi dan dekripsi adalah kunci yang sama.

2. Algoritma asimetris Dimana kunci yang digunakan untuk proses enkripsi dan dekripsi menggunakan kunci yang berbeda.

Sedangkan berdasarkan besar data yang diolah dalam satu kali proses, maka algoritma kriptografi dapat dibedakan menjadi dua jenis yaitu :

1. Algoritma *blok cipher*

   Informasi/data yang hendak dikirim dalam bentuk blok-blok besar (misal 64- bit) dimana blok-blok ini dioperasikan dengan fungsi enkripsi yang sama dan akan menghasilkan informasi rahasia dalam blok-blok yang berukuran sama.

2. Algoritma *stream cipher*

   Informasi/data yang hendak dikirim dioperasikan dalam bentuk blok-blok yang lebih kecil (byte atau bit), biasanya satu karakter persatuan persatuan waktu proses, menggunakan tranformasi enkripsi yang berubah setiap waktu[4].

2.4 Playfair Cipher

Playfair cipher atau sandi playfair ditemukan oleh Charles Wheatstone pada tahun 1854 yang mana dulu populer disebut Lord Playfair. Proses pembentukkan kunci pada metode ini hampir mirip dengan metode kriptografi *Vigenere Cipher*, tetapi pada *playfair cipher* memiliki teknik pemetaan yang lebih sulit jika dibandingkan dengan *Vigenere Cipher*.

Menurut Aftab, Chaoudhary, Vatsa ciphertext hasil enkripsi relatif mudah dipecahkan ketika kriptanalisis mengetahui ciphertext dan tabel cipher-nya, walaupun kriptanalis hanya mengetahui ciphertext tanpa mengetahui tabel cipher

kriptanalis dapat menebak bigram berdasarkan huruf yang bermakna dari sebuah kata.

Menurut Harris dan Attia, Kumar, Nidhal dan Wasfi, Shakti dan Gupta, tabel bawaan yang ada pada playfair cipher tidak dapat mengenkripsi plaintext yang berisi huruf kecil (a-z), angka (0-9) dan simbol-simbol. Kelemahan yang lain pada playfair adalah terjadinya ambigu pada hasil dekripsi karena pada persiapan enkripsi playfaircipher memiliki mekanisme mengganti J dengan I. Perlunya modifikasi tabel playfair cipher yang dapat digunakan untuk melakukan enkripsi huruf kapital, huruf kecil, angka dan simbol.

Menurut Stallings, playfair cipher menggunakan papan kunci yang berbentuk bujursangkar dalam melakukan penyandian. Papan kunci ini berukuran 5x5, dimana setiap bagian dalam papan kunci mewakili huruf-huruf dalam alfabet (abjad) dengan menghilangkan huruf J dari abjad.

Playfair merupakan digraphs cipher, artinya setiap proses enkripsi dilakukan pada setiap dua huruf. Sandi Playfair hanya dapat digunakan untuk proses enkripsi dan dekripsi data yang berupa teks alfabet, karakter yang tidak berupa teks alfabet dapat dihindari dengan menuliskannya dalam bentuk teks alfabet. Matriks kunci akan diisi sesuai dengan urutan kemunculan huruf pada kunci. Huruf yang digunakan tidak boleh digunakan lagi, sedangkan huruf yang tidak digunakan kunci akan disusun setelahnya sesuai dengan urutan alphabet.

Algoritma playfair merupakan bagian dari algoritma kriptografi klasik yang menggunakan teknik subtitusi. Subtitusi adalah penggantian setiap karakter plainteks dengan karakter lain. Berdasarkan jenis kuncinya algoritma

playfair merupakan algoritma simetri. Kunci yang digunakan untuk enkripsi sama dengan dekripsinya.

Playfair Cipher mengenkripsi pasangan huruf (digram atau digraf), bukan huruf tunggal seperti pada cipher klasik/tradisional lainnya. Tujuannya untuk membuat analisis frekuensi menjadi sulit sebab frekuensi kemunculan huruf di dalam ciphertextakan menjadi datar.

Menurut Stallings Sebelum melakukan enkripsi, pesan yang akan dienkripsi (plaintext) diatur terlebih dahulu sebagai berikut :

1. Semua spasi dan karakter yang bukan alfabet harus dihilangkan dari plaintext (jika ada).
2. Jika ada huruf J pada plaintext, maka ganti huruf tersebut dengan huruf I.
3. Pesan yang akan dienkripsi ditulis dalam pasangan huruf (bigram).
4. Jika ada huruf yang sama dalam pasangan huruf, maka sisipkan huruf X atau Z di tengahnya. Huruf yang disisipkan sebaiknya huruf X karena sangat kecil kemungkinan terdapat huruf X yang sama dalam bigram, tidak seperti huruf Z
5. Jika jumlah huruf pada plaintext adalah ganjil maka pilih sebuah huruf tambahan yang dipilih oleh orang yang mengenkripsi dan tambahkan di akhir plaintext. Huruf tambahan dapat dipilih sembarang misalnya huruf Z atau X.

Algoritma enkripsi untuk setiap bigram adalah sebagai berikut:

1. Jika ada dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kanannya
2. Jika ada dua huruf terdapat pada kolom yang sama maka tiap huruf diganti dengan huruf di bawahnya.

3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua.

4. Huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari huruf yang digunakan

a. Algoritma dekripsi merupakan kebalikan dari algoritma enkripsi untuk setiap bigram adalah sebagai berikut:

1. Jika ada dua huruf terdapat pada baris kunci yang sama maka tiap huruf diganti dengan huruf di kirinya

2. Jika ada dua huruf terdapat pada kolom yang sama maka tiap huruf diganti dengan huruf di atasnya.

3. Jika dua huruf tidak pada baris yang sama atau kolom yang sama, maka huruf pertama diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua.

4. Huruf kedua diganti dengan huruf pada titik sudut keempat dari persegi panjang yang dibentuk dari huruf yang digunakan.

Adapun tahapan enkripsi dalam pembentukkan kunci pada *playfair cipher* di penelitian ini adalah sebagai berikut:

1. Susun huruf ke dalam bentuk matriks n×n dengan menghilang huruf yang sama atau berulang dari abjad kunci, dan tambahkan huruf yang belum ada.
2. Koreksi apabila terdapat dua huruf yang sama pada baris kunci, maka tiap huruf diganti dengan huruf di kanannya.

3. Apabila terdapat dua haruf pada kolom kunci yang sama, maka huruf tersebut harus diganti dengan huruf di bawahnya.
4. Apabila pada baris atau kolom tidak terdapat dua huruf tidak yang sama, maka huruf pertama harus diganti dengan huruf pada perpotongan baris huruf pertama dengan kolom huruf kedua. Selanjutnya, huruf kedua diganti dengan huruf pada titik sudut keempat dari matriks persegi tersebut yang dibentuk dari 3 huruf.

Selanjutnya untuk tahapan dekripsi untuk mengkonversi *ciphertext* ke dalam *plaintext* pada *playfair cipher* adalah sebagai berikut:

1. Apabila terdapat huruf J, maka diganti dengan huruf I.
2. Tulis pesan dalam pasangan huruf.
3. Tidak boleh terdapat huruf yang sama, tetapi jika terdapat huruf yang sama, maka sisipkan huruf Z di tengahnya.
4. Apabila terdapat jumlah huruf ganjil, maka tambahkan huruf Z di akhir dari matriks yang telah dibentuk[1].

Tabel 2.1 Playfair Cipher matrixs 5x5

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

2.5 Pengertian Web (Website)

Menurut Rohi Abdulloh Website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet[5].

2.6 Pengertian HTML

HTML singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur website. Beberapa tugas utama HTML dalam membangun website diantaranya sebagai berikut[5]:

- Menentukan layout website.
- Memformat text dasar seperti pengaturan paragraf, dan format font.
- Membuat list.
- Membuat tabel.
- Menyisipkan gambar, video, dan audio.
- Membuat link.
- Membuat formulir.

2.7 Pengertian PHP

PHP sendiri sebenarnya merupakan singkatan dari (Hypertext Preprocessor), yang merupakan sebuah dokumen bahasa scriptingtingkat tinggi yang dipasang pada dokumen HTML. Sebagian besar sintaks dalam PHP mirip dengan bahasa C, Java dan Perl, namun pada PHP ada beberapa funsi yang lebih spesifik. Sedangkan tujuan utama dari penggunaan bahasa ini adalah untuk memungkinkan perancangan web yang dinamis dan dapat bekerja secara otomatis.

Untuk membuat halaman web, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bias saja membuat website hanya mengguanakan HTML saja. Web yang dihasilkan dengan HTML dan CSS ini dikenal dengan website statis, dimana konten didalam web bersifat tetap. Sebagai perbandingan, website dinamis yang bias dibuat menggunakan PHP adalah situs web yang bias menyesuaikan tampilan konten tergantung situasi. Website dinamis juga bias menyimpan data ke dalam database, membuat halam yang berubah-ubah sesuai input dari user, memproses form, dll. Untuk pembuatan web, kode PHP biasanya disisipkan ke dalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai scripting language atau bahasa pemrograman script.

Sebagi contoh penggunaan PHP, misalkan kita ingin membuat list dari 1 sampai 5 dengan menggunakan HTML murni kita bisa membuat secara manual, kode seperti berikut.

```
<!DOCTYPE html>
<html>
<head>
    <title>Contoh list dengan PHP</title>
</head>
<body>
    <h2>Daftar Nomer</h2>
    <ul>
        <?php
        for ($i=1; $i <=50 ; $i++) {
            echo "<li>$i</li>";
        }
        ?>
    </ul>
</body>
</html>
```

Gambar 2.2 Contoh list dengan HTML

Kode html diatas adalah contoh murni html untuk membuat list sampai 5 karna kalau saya tulis semua terlalu panjang. Sekarang kita coba terapkan mengguanakan PHP dengan perulangan 1 sampai dengan 50 dengan perintah yang lebih singkat seperti dibawah ini.

```
<!DOCTYPE html>
<html>
<head>
    <title>Contoh list dengan HTML</title>
</head>
<body>
    <h2>Daftar Nomer</h2>
    <ul>
        <li>1</li>
        <li>2</li>
        <li>3</li>
        <li>4</li>
        <li>5</li>
    </ul>
</body>
</html>
```

Gambar 2.3 Contoh list dengan PHP

Dengan menggunakan kode baris yang bahkan lebih sedikit, kita dapat membuat list tersebut menjadi 1000kali, bahkan 100.000 kali dengan hanya mengubah sebuah variable $i. PHP tidak hanya dapat melakukan pengulangan tersebut,masih banyak hal lain yang bisa kita lakukan dengan PHP, seperti menginput data ke database, menghasilkan gambar, mengonversi halaman text menjadi PDF, managemen cookie dan session[6].

2.8 Pengertian CSS

Menurut Rohi Abdulloh CSS singkatan dari *cascading style sheets*, yaitu skrip yang digunakan untuk mengatur desain website. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan website, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur website yang dibuat dengan HTML terlihat lebih rapi dan indah[5].

2.9 Pengertian XAMPP

Gambar 2.4 Logo aplikasi XAMPP

Menurut Heriyanto, Xampp adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah server. Kegunaan Xampp ini untuk membuat jaringan local sendiri dalam artian kita dapat membuat website secara offline untuk masa coba-coba di komputer sendiri. Jadi fungsi dari Xampp server itu sendiri merupakan server website kita untuk cara memakainya. Disebut server karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk mengakseskan web, untuk itu komputer kita harus menjadi server.

Dapat disimpulkan xampp adalah aplikasi *tools* untuk menyediakan paket lunak yang berisi konfigurasi Web Server, Apache, PHP, MySQL untuk membantukita dalam proses pembuatan aplikasi web yang menyatu menjadi satu sehingga memudahkan kita dalam membuat program web[7].

2.10 UML*(Unified Modeling Language)*

Berikut adalah definisi dari pengertian UML (*Unified Modeling Language*) adalah: Adapun pengertian UML menurut Yasin "*Unifield Modeling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem".

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu; use-case, class, object, state, sequence, collaboration, activity, component, dan deployment diagram.

- Use Case Diagram, menggambarkan sekelompok use cases dan aktor yang disertai dengan hubungan diantaranya. Diagram use cases ini menjelaskan dan menerangkan kebutuhan / requirement yang diinginkan/ dikehendaki user/pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.
- Class Diagram, yang memperlihatkan struktur statis dari kelas actual didalam sistem.
- Object Diagram, yang merupakan varian dari kelas diagram yang memperlihatkan lebih detail banyaknya obyek yang mengintantiasi (instances) kelas.
- State Diagram, yang memperliatkan semua keadaan (state) yang dapat dimiliki oleh kelas dan event yang dapat merubah keadaan tersebut.
- Sequence Diagram, yang memperlihatkan kolaborasi dinamik antara objek-objek dengan suatu urutan pesan (a sequence of message) antar objek tersebut.
- Collaboration Diagram, yang memperlihatkan kolaborasi dinamik antar objek tanpa memperhatikan aspek waktu.
- Activity Diagram, yang memperlihatkan aliran urutan aktifitas.

- Component Diagram, yang memperlihatkan struktur fisik dari source code dalam terminology code components. Komponen berisi informasi tentang logical class dapat berupa komponen source code, komponen biner atau komponen yang dapat dieksekusi.
- Deployment Diagram, yang memperlihatkan arsitektur fisik dari hardware dan software pada sistem.

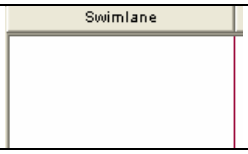
DAFTAR SIMBOL

UML (*Unified Modelling Language*)

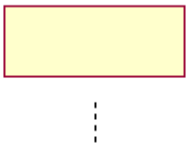
1. Diagram Use Case Proses

No.	Simbol	Keterangan
1.		Aktor Menunjukan user yang akan menggunakan sistem baru
2.		Use Case Menunjukan proses yang terjadi pada sistem baru
3.		Unidirectional Association Mnunjukan hubungan antara actor dengan dan use case atau antar use case

2. DIAGRAM ACTIVITY

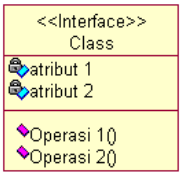
No.	Simbol	Keterangan
1		Kondisi Awal Menunjukan awal dari suatu diagram aktivitas
2		Kondisi Akhir Menunjukan akhir dari suatu diagram aktivitas
3		Kondisi transisi Menunjukan kondisi transisi antar aktivitas
4		Swimlane Menunjukan aktor dari diagram aktivitas yang dibuat
5		Aktivitas Menunjukan aktivitas-aktivitas yang terdapat pada diagram
6		Pengecekan kondisi Menunjukan pengecekan terhadap suatu kondisi

3. DIAGRAM SEQUENCE

No.	Simbol	Keterangan
1		Objek Menunjukan objek yang yang terdapat di diagram sequence

2		Pesan ke Objek sendiri Menunjukan pesan yang diproses pada objek itu sendiri
3		Pesan Objek Menunjukan pesan yang disampaikan ke objek lain dalam

4. DIAGRAM CLASS

No.	Simbol	Keterangan
1		Class Menunjukan class-class yang dibagung berdasarkan proses-proses sebelumnya (diagram sequence)
2		Unidirectional Association Mnunjukan hubungan antara class pada diagram class

2.11 Sublime text3

Gambar 2.5 Logo aplikasi Sublime Text

Sublime Text itu adalah salah satu *text editor* yang biasa digunakan oleh para programmer, khususnya *Web Developer*. Bisa diibaratkan sebagai senjata-nya *Web Developer*. Setiap *web developer* wajib untuk menggunakan Sublime Text sebagai ‘senjata harian’-nya[8].

2.12 Aplikasi

Gambar 2.6 Macam-macam aplikasi

Pengertian aplikasi

Kata Aplikasi diadopsi dari Bahasa Inggris "*Application*" yang berarti enggunaan, penerapan. Pengertian aplikasi secara lengkap adalah suatu penerapan perangkat lunak (*software*) yang dikembangkan untuk melakukan tugas tertentu. Dalam pengembangannya, aplikasi dibedakan menjadi aplikasi desktop, aplikasi web dan aplikasi mobile. Aplikasi yang hanya dapat dijalankan di perangkat komputer/PC disebut aplikasi desktop. Aplikasi yang dijalankan jika ada koneksi internet. Sedangkan aplikasi mobile adalah aplikasi yang dapat dijalankan di perangkat mobile. Suatu aplikasi dapat berjalan di berbagai perangkat dioperasikan oleh OS (*Operating System*) yang terdapat dalam perangkat tersebut[9].

BAB III

METODOLOGI PENELITIAN

3.1. Metode Pengumpulan Data

Pada bab ini penulis menguraikan tentang metode penelitian yang digunakan dalam perancangan sistem, yaitu metode pengumpulan data. Metode pengumpulan data adalah salah satu cara yang harus di lakukan untuk mengumpulkan data yang di perlukan untuk merancang sistem. Dan data yang di gunakan dalam penelitian ini merupakan data sekunder, penulis mengumpulkan data dari artikel-artikel, informasi dari internet, jurnal dan buku-buku dari perpustakaan.

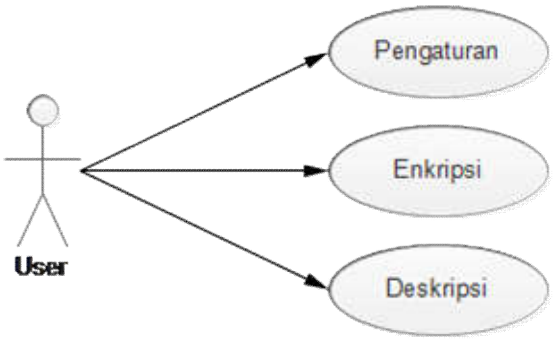
3.2. Metode Perancangan Sistem

Metode perancangan sistem adalah suatu perencanaan, penggambaran, dan pembuatan sistem dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem menentukan bagaimana sistem direncanakan, didesain, dan dibangun sedemikian rupa. Tahap perancangan sistem merupakan tahap lanjutan dalam pengembangan sistem informasi yang dilakukan setelah melakukan analisis sistem yang bertujuan untuk memberikan gambaran kepada *user* tentang bagaimana sistem baru yang diusulkan akan berkerja.

Dalam pembuatan sistem alat bantu yang digunakan dalam membuat perancangan dan desain yaitu dengan menggunakan *Unified Modeling Language* (UML). *Unified Modelling Language* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.

3.2.1. *Use Case Diagram*

Use Case diagram adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh *actor. Use case* diagram dalam Perancangan Aplikasi Enkripsi Kata Menggunakan Algoritma *Playfair Cipher* yaitu:

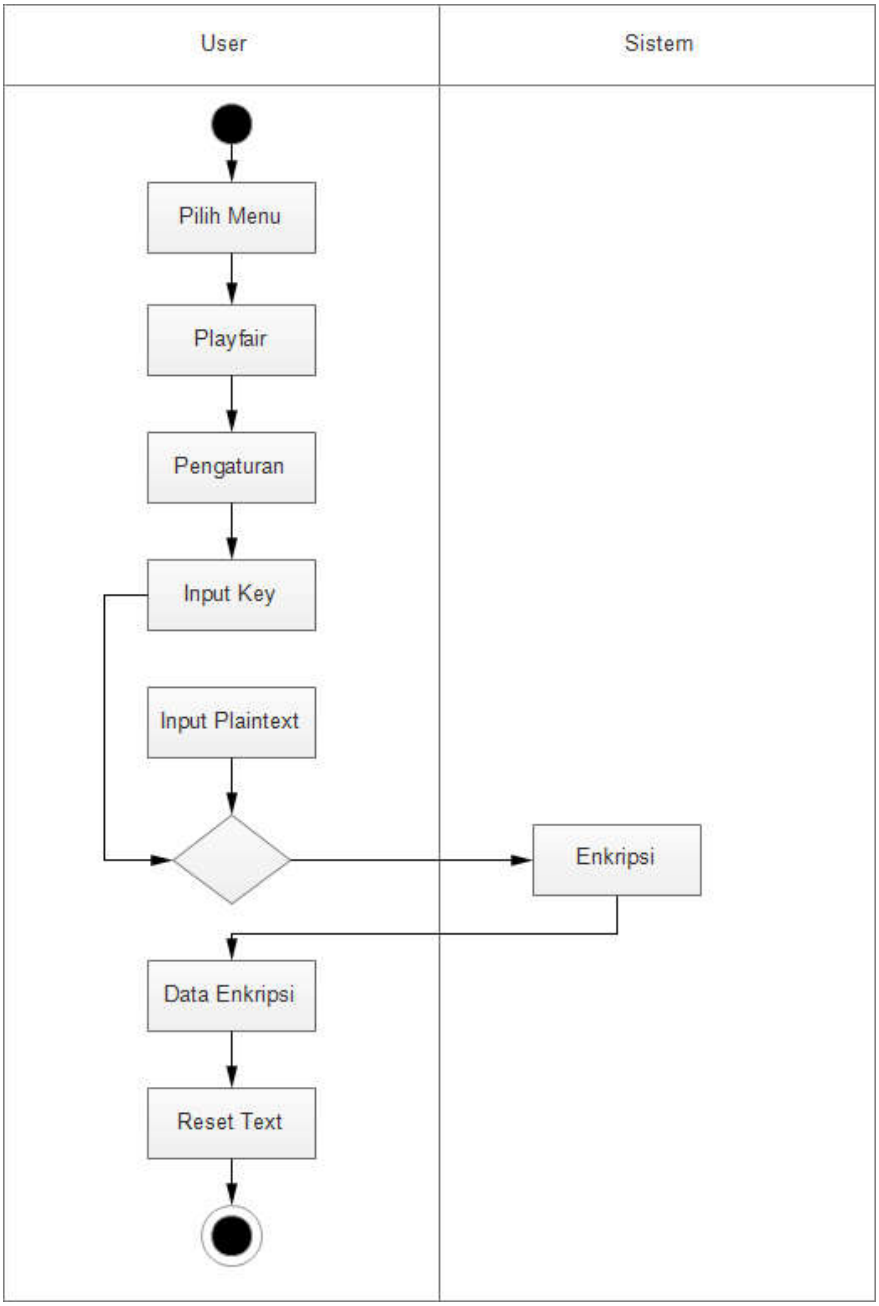


Gambar 3.1 *Use Case* Diagram

Pada gambar 3.2.1 *Use Case* Diagram menjelaskan aktivitas yang dapat dilakukan oleh user di sistem tersebut.

3.2.2. *Activity Diagram* Enkripsi

Gambar perancangan UML *Activity Diagram* enkripsi.



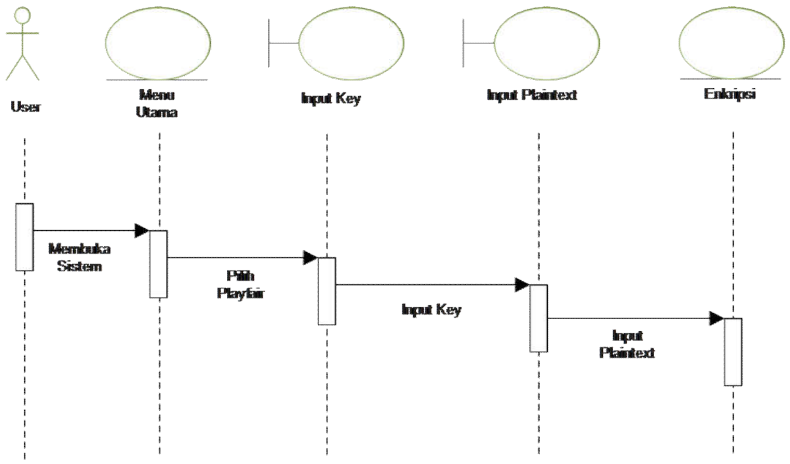
Gambar 3.2 *Activity Diagram* Enkripsi

Pada Gambar 3.2.2 *Activity Diagram* Enkripsi menjelaskan bahwa :

1. *User* menjalankan sistem dan memilih menu *Playfair*.
2. Sebelum melakukan enkripsi, *user* mengubah pengaturan sesuai dengan keinginan dan memilih metode ENKRIPSI/DESKRIPSI.
3. *User* menginputkan *key*(kunci).
4. Kemudian user menginputkan *Plaintext*(teks) yang akan di enkripsi.
5. *User* akan mendapatkan hasil enkripsi dari sistem secara *realtime*(langsung).
6. Setelah mendapatkan hasil enkripsi, *user* dapat mereset/menghapus *plaintext* yang di inputkan dan hasil enkripsi.

3.2.3. Squence Diagram

Gambar Perancangan UML *Squence Diagram* enkripsi.

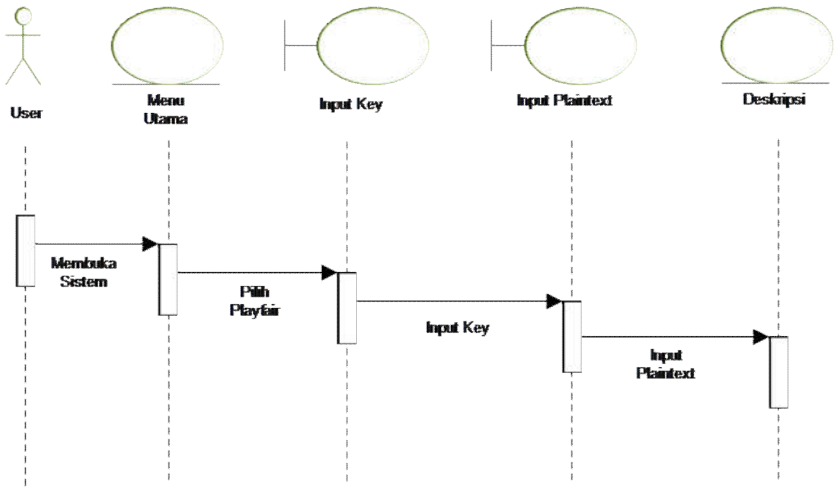


Gambar 3.3 *Squence Diagram* enkripsi

Pada gambar 3.2.3. *Squence Diagram* enkripsi menjelaskan bahwa :

1. *User* membuka sistem.
2. Setelah terbuka *user* memilih menu *Playfair*, dan mengatur/mengubah pengaturan sesuai dengan kebutuhan.
3. *User* menginputkan *KEY* dan memilih ENKRIPSI.
4. Kemudian *user* menginputkan *plaintext* untuk mendapatkan kalimat atau kata yang dienkripsi.

Gambar perancangan UML *Squence Diagram* deskripsi.



Gambar 3.4 *Squence Diagram* deskripsi

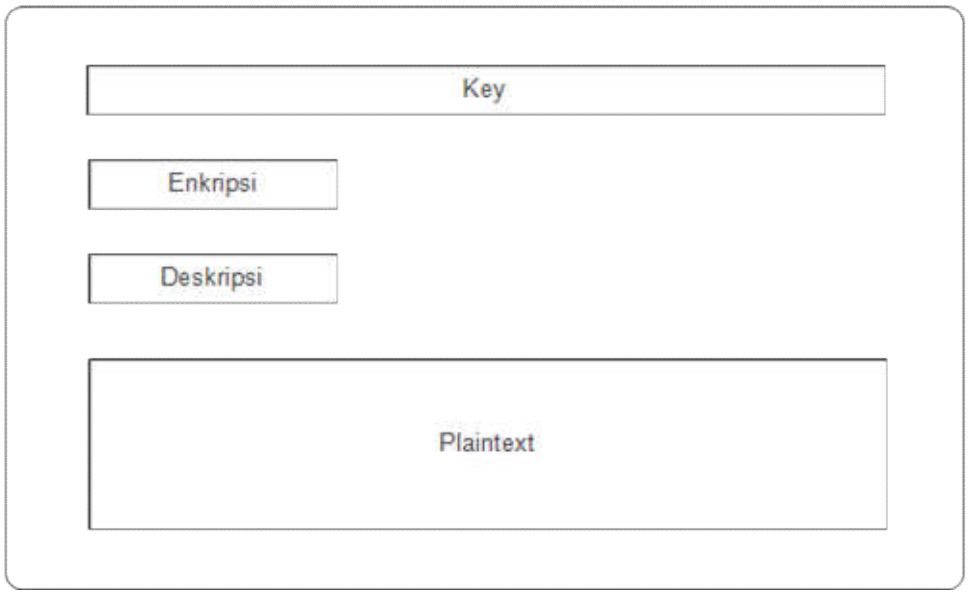
Pada gambar 3.2.4. *Squence Diagram* deskripsi menjelaskan bahwa :

1. *User* membuka sistem.
2. Setelah terbuka *user* memilih menu *Playfair*, dan mengatur/mengubah pengaturan sesuai dengan kebutuhan.

3. *User* menginputkan *KEY* dan memilih DESKRIPSI.
4. Kemudian *user* menginputkan *ciphertext* untuk mendapatkan kalimat atau kata yang dideskripsi.

3.2.4. Rancangan *input*(masukan)

Rancangan *input*(masukan) adalah rancangan sebuah form pengolahan pengaturan, *key* dan *plaintext* yang dimasukkan ke sistem kemudian di proses oleh sistem sehingga menghasilkan *output*(keluaran). Rancangan *input*(masukan) enkripsi/deskripsi adalah sebagai berikut :



Gambar 3.5 Rancangan *Input*(masukan) enkripsi/deskripsi

1. Nama Masukan : Form Enkripsi/Deskripsi
2. Fungsi : Untuk menginput Key enkripsi/deskripsi, plaintext
3. Distribusi : *User*

4. Keterangan : untuk enkripsi dan deskripsi

3.2.5. Rancangan Proses

Alat bantu yang digunakan dalam perancangan dan desain aplikasi enkripsi, adalah dengan menggunakan UML(*Unified Modelling Language*) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.

3.2.6. Rancangan *Output*(keluaran)

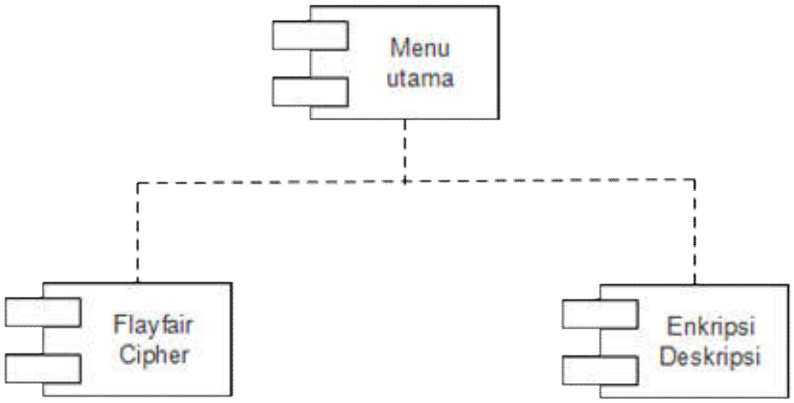
Rancangan *Output*(keluaran) dalam sistem adalah keluaran dari hasil proses yang dilakukan sistem. *Key* dan *Plaintext* yang diinputkan akan menghasilkan data keluaran berupa teks yang sudah menjadi *ciphertext,* begitu juga sebaliknya *ciphertext* dan *key* yang di *input*kan akan kembali menjadi *plaintext* pada form ini. Rancangan *output*(keluaran) enkripsi/deskripsi adalah sebagai berikut :

Gambar 3.6 Rancangan *output*(keluaran) enkripsi/deskripsi

1. Nama keluaran : Form Enkripsi/Deskripsi
2. Fungsi : Media hasil enkripsi/deskripsi
3. Distribusi : *User*
4. Keterangan : *Output* enkripsi dan deskripsi

3.2.7. Rancangan *Interface*

Rancangan *Interface* aplikasi enkripsi



Gambar 3.7 Rancangan *Componen Diagram* aplikasi enkripsi

Pada gambar 3.7 Rancangan *Componen Diagram* aplikasi enkripsi menjelaskan rancangan pada sistem terdapat *form playfair cipher* dan *form* enkripsi/deskripsi.

3.3. Proses enkripsi/deskripsi secara manual

Membuat dahulu kunci algoritma Playfair

Cipher :

Kunci : **SPONGEBOB**,

1. Susun dahulu hurufnya, huruf yang sudah Disebutkan tidak dituliskan lagi **SPONGEB** kemdian jika terdapat huruf J makan diganti dengan huruf I.

2. Selanjutnya tambakan dengan huruf abjad sisanya yang tidak terdapat pada kunci tadi **ACDFHIKLMQRTUVWXYZ**

   Menjadi : **SPONGEBACDFHIKLMQRTUVWXYZ**

3. Masukkan ke dalam bujur sangkar

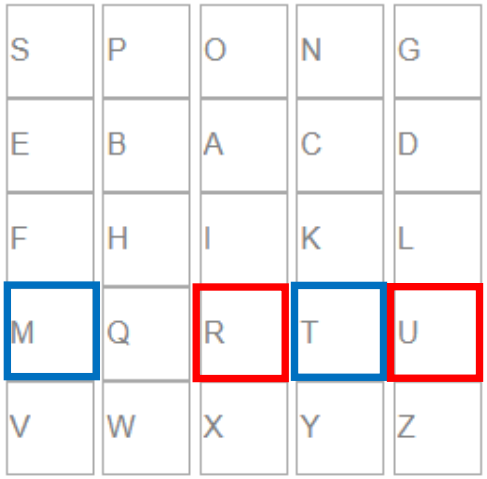
S	P	O	N	G
E	B	A	C	D
F	H	I	K	L
M	Q	R	T	U
V	W	X	Y	Z

Plaintext : **RUMAH NANAS**

- Hilangkan semua karakter yang bukan huruf abjad
- Tidak terdapat **j,** maka langsung lakukan pasangan huruf
- Jika ada pasangan huruf yang sama maka sisipkan huruf **Z**

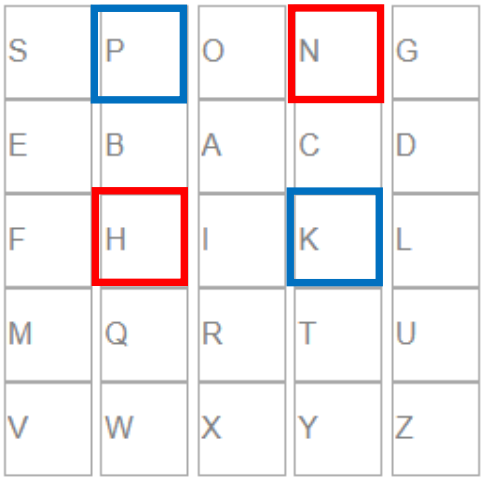
Jika telah di pasangkan **RU MA HN AN AS**

Enkripsi **RU** menjadi **TM**

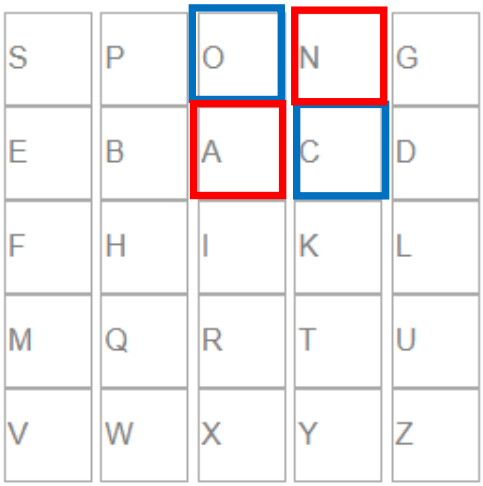


Enkripsi **MA** menjadi **RE**

Enkripsi **HN** menjadi **KP**



Enkripsi **AN** menjadi **CO**



Enkripsi **AS** menjadi **EO**

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

Setelah proses enkripsinya selesai, hasilnya adalah :

P : **RU MA HN AN AS**

C : **TM RE KP CO EO**

Key : **SPONGEBOB**

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. IMPLEMENTASI

Implementasi adalah penerapan cara kerja sistem berdasarkan hasil analisa dan juga perancangan yang telah dibuat sebelumnya ke dalam suatu bahasa pemrograman tertentu.

Tahap implementasi merupakan tahap penciptaan perangkat lunak, tahap kelanjutan dari kegiatan perancangan sistem. Tahap ini merupakan tahap dimana sistem siap untuk dioperasikan, yang terdiri dari penjelasan mengenai lingkungan implementasi, dan implementasi program.

Lingkup implementasi yang direkomendasikan meliputi lingkungan perangkat lunak (*software*) dan perangkat keras (*hardware*).

4.2. Spesifikasi Perangkat Keras (*Hardware*)

Berikut adalah spesifikasi perangkat yang digunakan saat merancang **Aplikasi Enkripsi Kata Menggunakan Algoritma *Playfair Cipher*** ini, yaitu :

1. Processor AMD A4-9125 RADEON R3, 4 CORES 2C+2G 2.30GHz
2. Memory 4GB
3. Hardisk 500GB
4. Monitor 14”

4.3. Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk mengimplementasikan **Aplikasi Enkripsi Kata Menggunakan Algoritma *Playfair Cipher*** ini adalah sebagai berikut :

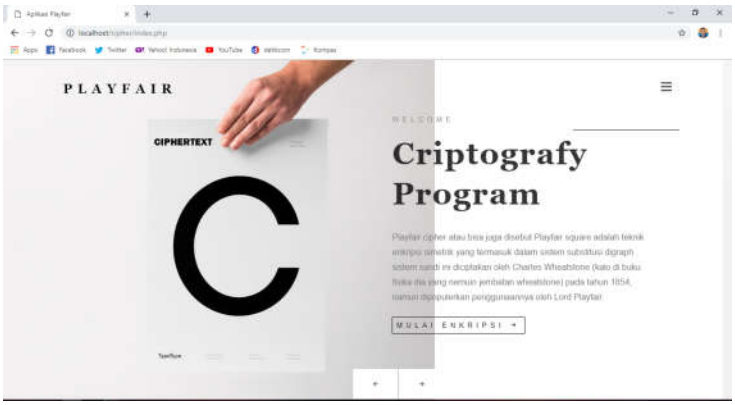
1. Sistem Operasi Windows 10
2. Google Chrome
3. XAMPP
4. Sublime Text3

4.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap halaman aplikasi yang dibuat dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka yang dibuat.

4.4.1. Halaman Utama

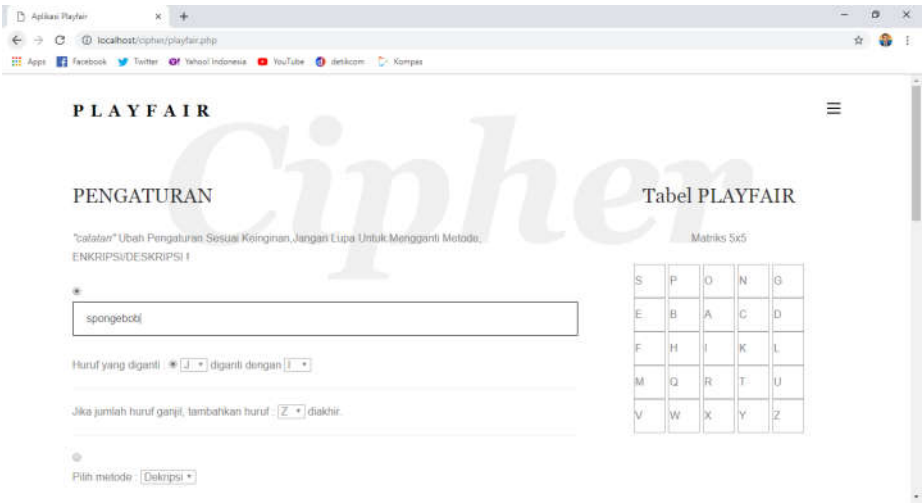
Halaman utama adalah halaman yang akan tampil pada awal aplikasi dibuka. Didalam halaman utama akan menampilkan sedikit tentang sejarah algoritma *Playfair* cipher.



Gambar 4.1. Halaman Utama

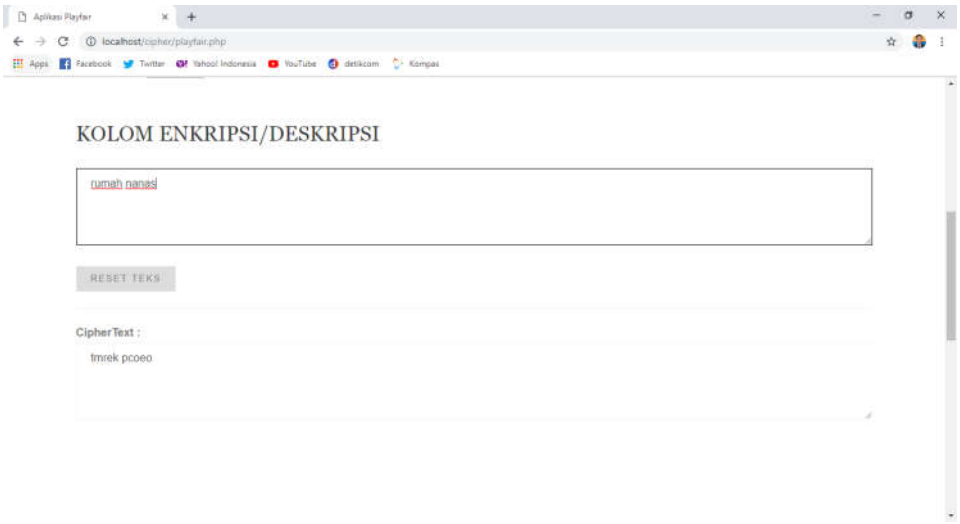
4.4.2. Halaman Enkripsi/Deskripsi

Halaman ini adalah tempat di mana kita menginputkan *key*(kata kunci), mengubah pengaturan sesuai dengan yang di butuhkan, memilih metode enkripsi/deskripsi, menginputkan *plaintext* kalimat/kata. Dihalaman ini juga terdapat tabel *Playfair* matrix 5x5 yang interaktif akan berubah sesuai dengan *key*(kata kunci) yang diinputkan.



Gambar 4.2. Halaman Enkripsi/Deskripsi

Dan setelah menginputkan key(kata kunci) “SPONGEBOB” dihalaman ini juga kita dapat menginputkan *plaintext* yang secara automatis akan langsung menghasilkan *chipertext* seperti gambar di bawah ini.



Gambar 4.3. Halaman Ekripsi/Deskripsi

Pada gambar 4.3. Halaman Enkripsi/deskripsi berisikan plaintext "RUMAH NANAS" yang secara automatis menghasilkan ciphertext "TMREK PCOEO" dan pada gambar tersebut terdapat sebuah tombol "RESET TEKS" yang berfungsi untuk mereset/menghapus *plaintext* dan *ciphertext.*

4.5. Pengujian Perangkat Lunak (Software)

Pada tahap ini akan dilakukan pengujian sistem yang bertujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Pengujian perangkat lunak ini menggunakan pengujian black box. Pengujian black box berfokus pada persyaratan fungsional perangkat lunak tanpa menguji desain dan program.

4.5.1. Pengujian Fungsional

Pengujian alpha dilakukan dengan menggunakan metode black box. Untuk pengujian alpha ini yaitu pada pengujian sebagai pengguna.

Tabel 4.1. Skenario Pengujian

Uji Fitur	Detail Pengujian	Jenis Pengujian
Enkripsi	Mengenkripsi kalimat/kata	*Black box*
Deskripsi	Mendeskripsi kalimat/kata	*Black box*

4.5.2. Kasus dan Hasil Pengujian

Berikut ini adalah hasil dari pengujian fungsional dari aplikasi:

Tabel 4.2. Pengujian Enkripsi dan Deskripsi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengenkripsi kalimat Contoh : *Plaintext* : RUMAH NANAS	Menghasilkan Ciphertext “TMREK PCOEO”	Sistem memproses *Plaintext* dan Menghasilkan *Ciphertext*	Diterima
Mendeskripsi kalimat Contoh : *Ciphertext* : TMREK PCOEO	Mengembalikan Plaintext “RUMAH NANAS	Sistem memproses *Ciphertext* dan mengembalikan *Plaintext*	Diterima

4.6. Kesimpulan Pengujian

Pengujian alpha dilakukan dengan menggunakan metode black box. merupakan pengujian sistem yang bertujuan untuk menemukan kesalahan kesalahan atau kekurangan pada perangkat lunak yang diuji. Dalam pengujian disini masih dalam tahapan pengujian yang sebatas pengujian secara fungsionalitas saja. Perihal yang tidak diinginkan dapat terjadi tanpa pengujian secara spesifik terutama pada bagian interface dimana pemograman kemampuan dinamis elemen antarmuka berbaur menggunakan *Hyper Text Markup Language* (HTML) serta penyajian dokumen dengan *Cassading Style Sheet* (CSS). Sehingga dalam menjalankan sistem sebagai aplikasi berbasis web tentunya berpengaruh pada web browser untuk menjalankan sistem sebagai aplikasi berbasis web.

BAB V

PENUTUP

5.1. Kesimpulan

Hasil perancangan aplikasi kriptografi dengan metode *Playfair* Cipher berbasis web dapat disimpulkan bahwa proses enkripsi dan deskripsi kata atau kalimat dapat dilakukan secara komputasi sehingga tidak perlu menjabarkan proses enkripsi dan dekripsi secara manual yang dapat membuang-buang waktu, dengan kata lain aplikasi ini untuk mengefisiensikan waktu untuk proses enkrpsi dan dekripsi. Selain itu, dipilihnya metode *Playfair* cipher karena metode ini dikenal mudah dipahami dan mudah diimplementasikan.

5.2. Saran

Dari penjabaran tentang perancangan program enkripsi, dekripsi sampai tahapan implementasi perlu dilakukan pengembangan agar menjadi aplikasi yang lebih baik lagi seperti :

1. Membuat aplikasi Enkripsi/Deskripsi berbasis android
2. Menambahkan metode cipher lainnya sehingga lebih variatif dan beragam.

DAFTAR PUSTAKA

[1] F. Azmi and R. Anugrahwaty, "Analisis Matriks 5x7 Pada Kriptografi Playfair Cipher," *J. Penelit. Tek. Inform.*, vol. 1, no. 2, pp. 27–30, 2017.

[2] R. Watrianthos, "Perbandingan teknik kriptografi metode sapphire II dan rc4," *J. Ilm. AMIK Labuhan Batu*, vol. 3, no. 2, pp. 17–40, 2015.

[3] P. D. Dr.suarga, M.Sc., M.Math., *ALGORITMA dan PEMROGRAMAN*. 2018.

[4] M. Mawardina, U. I. N. Sunan, G. Djati, and M. Mawardina, "Aplikasi Kriptografi dengan Metode Vigenere Cipher Berbasis Web," 2016.

[5] A. Josi, "DESA (STUDI KASUS DESA SUGIHAN KECAMATAN RAMBANG) STMIK-MUSIRAWAS LUBUKLINGGAU," *Jurnnal Teknol. Inf.*, vol. 9, no. 1, pp. 50–57, 2017.

[6] D. Setiawan, *BUKU SAKTI PEMROGRAMAN WEB*. 2017.

[7] jubilee enterprise, *otodidak desain dan pemrograman website*. 2017.

[8] G. Pratama, "apaitu Sublime Text?," 2019. [Online]. Available: https://belajarkoding.net/senjata-koding-sublime-text-3/.

[9] I. Syahfitri, "Pengertian Aplikasi Beserta Fungsi dan Contoh Aplikasi," *BoxGrinder*, 2018. [Online]. Available: https://boxgrinder.org/blog/aplikasi-pengertian-sejarah-jenis-serta-fungsinya/.

[10] R. W. Simbolon, "CIPHER DAN STEGANOGRAFI DENGAN TEKNIK LEAST SIGNIFICANT BIT (LSB) PROTECTING THE STUDENT ACADEMIC TRANSCRIPT USING PLAYFAIR CIPHER," *J. Teknol. Inf. dan Komun.*, vol. 5, no. 1, pp. 59–70, 2016.

LISTING PROGRAM

```
<!DOCTYPE HTML>
<html>
	<head>
	<meta charset="utf-8">
	<meta http-equiv="X-UA-Compatible" content="IE=edge">
	<title>Aplikasi Playfair</title>
	<meta name="viewport" content="width=device-width, initial-scale=1">
	<meta name="description" content="" />
	<meta name="keywords" content="" />
	<meta name="author" content="" />

	<!-- Modernizr JS -->
	<script src="js/modernizr-2.6.2.min.js"></script>
	<!-- FOR IE9 below -->
	<!--[if lt IE 9]>
	<script src="js/respond.min.js"></script>
	<![endif]-->
	</head>
<script type="text/javascript">
function resetFields(pForm,type){
  if (type==0){
    document.getElementById('playfairCipherText').value='';
    document.getElementById('playfairCipherDiv').innerHTML='';
  }else{
    var text=document.getElementById('playfairCipherText').value;
    pForm.reset();
    repopulateReplace2(0);
    document.getElementById('playfairCipherRandom').disabled=true;
    document.getElementById('playfairCipherKey').disabled=false;
    document.getElementById('playfairCipherText').value=text;
    document.getElementById('playfairCipherDiv').innerHTML='';
  }
  playfairCipher();
}
</script>
<script type="text/javascript">
var inChars = new
Array('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');
function repopulateReplace2(id){
  if (id==1){
    var r1=document.getElementById('playfairCipherReplace1').value;
    var r2=document.getElementById('playfairCipherReplace2').value;
    if (r2==r1 && r1=='I'){
      r2='A';
    }else if(r2==r1){
```

```
        r2='I';
      }
    }else{
      r2='I'
    }
    document.getElementById('playfairCipherReplace2').options.length=0;
    var j=0;
    for (var i=0; i<inChars.length; i++){
      if (r1!=inChars[i].toUpperCase()){
        if (i==9){
          defSel=true;
        }else{
          defSel=false;
        }
        if (r2==inChars[i].toUpperCase()){
          sel=true;
        }else{
          sel=false;
        }
        document.getElementById('playfairCipherReplace2').options[j]=new
Option(inChars[i].toUpperCase(), inChars[i].toUpperCase(), defSel, sel);
        j++;
      }
    }
}
function populateSquare(action){
    //Action = 0 => empty
    //Action = 1 => random
    //Action = 2 => standard with key
    //Action = 3 => get random or standard from document
    if (action==3){
      if(document.getElementById('playfairCipherManualCheck').checked){
        action=1;
      }else{
        action=2;
      }
    }
    if (document.getElementById('playfairCipherReplaceCheck').checked){
      var omit='';
      var
subs=[document.getElementById('playfairCipherReplace1').value,document.getElement
ById('playfairCipherReplace2').value];
    }else{
      var omit=document.getElementById('playfairCipherOmit').value;
      var subs=['',''];
    }
    var charList=new Array();
```

```
  for(var i=0; i<inChars.length; i++){
    if (inChars[i]!=subs[0].toLowerCase() && inChars[i]!=omit.toLowerCase()){
      charList.push(inChars[i]);
    }
  }
  if (action==0){
    for(i=0; i<25; i++){
      document.getElementById(i).value='';
    }
  }
  if (action==1){
    for(var i=0; i<25; i++){
      var pos=Math.floor((Math.random()*(25-i))+1)-1;
      if (pos>=charList.length){
        pos=(charList.length-1);
      }
      if (pos<0){
        pos=0;
      }
      document.getElementById(i).value=charList[pos].toUpperCase();
      charList.splice(pos,1);
    }
  }
  if (action==2){
    var key=document.getElementById('playfairCipherKey').value;
    var elID=0;
    for(var i=0; i<key.length; i++){
      if (key.charAt(i).toUpperCase()==subs[0].toUpperCase()){
        var keyChar=subs[1].toUpperCase();
      }else{
        var keyChar=key.charAt(i);
      }
      if (charList.indexOf(keyChar.toLowerCase())>-1){
        document.getElementById(elID).value=keyChar.toUpperCase();
        charList.splice(charList.indexOf(keyChar.toLowerCase()),1);
        elID++;
      }
    }
    var keyLength=25-charList.length;
    for(var i=0; i<charList.length; i++){
      document.getElementById(i+keyLength).value=charList[i].toUpperCase();
    }
  }
  playfairCipher();
}
function setEnabled(id){
  if (id==1){
```

```
      if (document.getElementById('playfairCipherManualCheck').checked){
        document.getElementById('playfairCipherRandom').disabled=false;
        document.getElementById('playfairCipherKey').disabled=true;
        populateSquare(0);
      }else{
        document.getElementById('playfairCipherRandom').disabled=true;
        document.getElementById('playfairCipherKey').disabled=false;
        populateSquare(2);
      }
    }else if(id==2){
      if (document.getElementById('playfairCipherReplaceCheck').checked){
        document.getElementById('playfairCipherReplace1').disabled=false;
        document.getElementById('playfairCipherReplace2').disabled=false;
        document.getElementById('playfairCipherOmit').disabled=true;
        populateSquare(3);
      }else{
        document.getElementById('playfairCipherReplace1').disabled=true;
        document.getElementById('playfairCipherReplace2').disabled=true;
        document.getElementById('playfairCipherOmit').disabled=false;
        populateSquare(3);
      }
    }else if(id==3){
      if (document.getElementById('playfairCipherBreakCheck').checked){
        document.getElementById('playfairCipherDoubleChar').disabled=false;
        document.getElementById('playfairCipherBreakMethod').disabled=false;
      }else{
        document.getElementById('playfairCipherDoubleChar').disabled=true;
        document.getElementById('playfairCipherBreakMethod').disabled=true;
      }
    }
    playfairCipher();
}
function checkValues(num){
    if (document.getElementById('playfairCipherReplaceCheck').checked){
      var omit='';
      var
subs=[document.getElementById('playfairCipherReplace1').value,document.getElement
ById('playfairCipherReplace2').value];
    }else{
      var omit=document.getElementById('playfairCipherOmit').value;
      var subs=['',''];
    }
    var charList=new Array();
    for(var i=0; i<inChars.length; i++){
      if (inChars[i]!=subs[0].toLowerCase() && inChars[i]!=omit.toLowerCase()){
        charList.push(inChars[i]);
      }
```

```
    }
    var newVal=document.getElementById(num).value;
    if (newVal.toUpperCase()==subs[0]){
      document.getElementById(num).value=subs[1];
      newVal=subs[1];
    }
    var newValIndex=charList.indexOf(newVal.toLowerCase());
    if (newValIndex>-1){
      var charCounts=new Array(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
      for(i=0; i<25; i++){
        if (i!=parseInt(num)){
          var
oldValIndex=charList.indexOf(document.getElementById(i).value.toLowerCase());
          charCounts[oldValIndex]=1;
        }
      }
      if (charCounts[newValIndex]==0){
        document.getElementById(num).value=newVal.toUpperCase();
      }else{
        document.getElementById(num).value='';
      }
    }else{
      document.getElementById(num).value='';
    }
    playfairCipher();
}
function playfairCipher(){
    var chars=document.getElementById('playfairCipherText').value;
    if (chars.length==0){
      document.getElementById('playfairCipherDiv').innerHTML='';
      return;
    }
    var keySquare=new Array();
    var emptyFound=0;
    for(var i=0; i<25; i++){
      keySquare.push(document.getElementById(i).value);
      if (document.getElementById(i).value==''){
        emptyFound=1;
      }
    }
    if (emptyFound==1){
      newDiv='<b>Hasil :</b><br>Key square is incomplete';
      document.getElementById('playfairCipherDiv').innerHTML=newDiv;
      return;
    }
    var dropDownMethod = document.getElementById("playfairCipherMethod");
    var method = dropDownMethod.options[dropDownMethod.selectedIndex].value;
```

```
  var newDiv='<b>CipherText :</b><br>';
  newDiv+='<div class="form-group"><div class="form-line"><textarea class="form-
control no-resize" id="playfairCipherResult" rows="4">';
  if (document.getElementById('playfairCipherReplaceCheck').checked){
    var omit='';
    var
subs=[document.getElementById('playfairCipherReplace1').value,document.getElement
ById('playfairCipherReplace2').value];
  }else{
    var omit=document.getElementById('playfairCipherOmit').value;
    var subs=['',''];
  }
  if (document.getElementById('playfairCipherBreakCheck').checked){
    var
doublesMethod=document.getElementById('playfairCipherBreakMethod').value;
    var replace=document.getElementById('playfairCipherDoubleChar').value;
  }else{
    var doublesMethod='breakNone';
    var replace=document.getElementById('playfairCipherDoubleChar').value;
  }
  var charList=new Array();
  for(var i=0; i<inChars.length; i++){
    if (inChars[i]!=subs[0].toLowerCase() && inChars[i]!=omit.toLowerCase()){
      charList.push(inChars[i]);
    }
  }
  var charCount=new Array('',0);
  var tempChars='';
  if (doublesMethod!='breakNone'){
    var c1='';
    var c2='';
    for(var i=0; i<chars.length; i++){
      if (c1==''){
        if (omit=='' && chars.charAt(i).toUpperCase()==subs[0].toUpperCase()){
          if (chars.charAt(i)==chars.charAt(i).toUpperCase()){
            c1=subs[1].toUpperCase();
          }else{
            c1=subs[1].toLowerCase();
          }
          charCount[0]=c1;
          charCount[1]++;
        }else if (charList.indexOf(chars.charAt(i).toLowerCase())>-1){
          c1=chars.charAt(i);
          charCount[0]=c1;
          charCount[1]++;
        }
      }else{
```

```
            if (omit=='' && chars.charAt(i).toUpperCase()==subs[0].toUpperCase()){
               if (chars.charAt(i)==chars.charAt(i).toUpperCase()){
                  c2=subs[1].toUpperCase();
               }else{
                  c2=subs[1].toLowerCase();
               }
               charCount[0]=c2;
               charCount[1]++;
            }else if (charList.indexOf(chars.charAt(i).toLowerCase())>-1){
               c2=chars.charAt(i);
               charCount[0]=c2;
               charCount[1]++;
            }
         }
         if (c1!='' && c2!=''){
            if (c1.toLowerCase()==c2.toLowerCase()){
               if (c1==c1.toUpperCase()){
                  tempChars+=replace.toUpperCase();
                  charCount[0]=replace.toUpperCase();
               }else{
                  tempChars+=replace.toLowerCase();
                  charCount[0]=replace.toLowerCase();
               }
               charCount[1]++;
            }
            if (doublesMethod=='breakAll'){
               c1=c2;
               c2='';
            }else{
               c1='';
               c2='';
            }
         }
         if (omit=='' && chars.charAt(i).toUpperCase()==subs[0].toUpperCase()){
            if (chars.charAt(i)==chars.charAt(i).toUpperCase()){
               tempChars+=subs[1].toUpperCase();
            }else{
               tempChars+=subs[1].toLowerCase();
            }
         }else{
            tempChars+=chars.charAt(i);
         }
      }
      chars=tempChars;
   }else{
      for(var i=0; i<chars.length; i++){
         if (omit=='' && chars.charAt(i).toUpperCase()==subs[0].toUpperCase()){
```

```
            if (chars.charAt(i)==chars.charAt(i).toUpperCase()){
               charCount[0]=subs[1].toUpperCase();
               tempChars+=subs[1].toUpperCase();
            }else{
               charCount[0]=subs[1].toLowerCase();
               tempChars+=subs[1].toLowerCase();
            }
            charCount[1]++;
         }else if (charList.indexOf(chars.charAt(i).toLowerCase())>-1){
            charCount[0]=chars.charAt(i);
            charCount[1]++;
            tempChars+=chars.charAt(i);
         }else{
            tempChars+=chars.charAt(i);
         }
      }
      chars=tempChars;
   }
   if (charCount[1]%2!=0){
      if (charCount[0]==charCount[0].toUpperCase()){
         chars+=replace.toUpperCase();
      }else{
         chars+=replace.toLowerCase();
      }
   }
   //newDiv+=tempChars+'\n';
   //newDiv+=chars+'\n';
   //newDiv+=charCount[0]+' - '+charCount[1]+'\n\n';
   var i=0;
   var firstChar='';
   var secondChar='';
   var newChars='';
   while(i<chars.length){
      if (firstChar==''){
         if (charList.indexOf(chars.charAt(i).toLowerCase())>-1){
            firstChar=chars.charAt(i);
         }
      }else{
         if (charList.indexOf(chars.charAt(i).toLowerCase())>-1){
            secondChar=chars.charAt(i);
         }
      }
      if (firstChar!='' && secondChar!=''){
         var keyLocFirst=keySquare.indexOf(firstChar.toUpperCase());
         var rowNmbFirst=Math.floor(keyLocFirst/5);
         var columnNmbFirst=(keyLocFirst%5);
         var keyLocSecond=keySquare.indexOf(secondChar.toUpperCase());
```

```
        var rowNmbSecond=Math.floor(keyLocSecond/5);
        var columnNmbSecond=(keyLocSecond%5);
        if (method=='Encrypt'){
           var shift=1;
        }else{
           var shift=4;
        }
        if (firstChar==secondChar && doublesMethod=='breakNone'){
           if(firstChar==firstChar.toUpperCase()){
newChars+=keySquare[((rowNmbFirst+shift)%5)*5+((columnNmbFirst+shift)%5)].toUpp
erCase();
           }else{
newChars+=keySquare[((rowNmbFirst+shift)%5)*5+((columnNmbFirst+shift)%5)].toLow
erCase();
           }
           if(secondChar==secondChar.toUpperCase()){

newChars+=keySquare[((rowNmbSecond+shift)%5)*5+((columnNmbSecond+shift)%5)].t
oUpperCase();
           }else{
newChars+=keySquare[((rowNmbSecond+shift)%5)*5+((columnNmbSecond+shift)%5)].t
oLowerCase();
           }
        }else if (rowNmbFirst==rowNmbSecond){ // Same row
           if(firstChar==firstChar.toUpperCase()){
newChars+=keySquare[rowNmbFirst*5+((columnNmbFirst+shift)%5)].toUpperCase();
           }else{
newChars+=keySquare[rowNmbFirst*5+((columnNmbFirst+shift)%5)].toLowerCase();
           }
           if(secondChar==secondChar.toUpperCase()){
newChars+=keySquare[rowNmbSecond*5+((columnNmbSecond+shift)%5)].toUpperCas
e();
           }else{
newChars+=keySquare[rowNmbSecond*5+((columnNmbSecond+shift)%5)].toLowerCas
e();
           }
        }else if(columnNmbFirst==columnNmbSecond){ //Same column
           if(firstChar==firstChar.toUpperCase()){
newChars+=keySquare[((rowNmbFirst+shift)%5)*5+columnNmbFirst].toUpperCase();
           }else{
 newChars+=keySquare[((rowNmbFirst+shift)%5)*5+columnNmbFirst].toLowerCase();
           }
           if(secondChar==secondChar.toUpperCase()){
newChars+=keySquare[((rowNmbSecond+shift)%5)*5+columnNmbSecond].toUpperCas
e();
```

```
            }else{
newChars+=keySquare[((rowNmbSecond+shift)%5)*5+columnNmbSecond].toLowerCas
e();
            }
          }else{ //Rectangle
            if(firstChar==firstChar.toUpperCase()){
              newChars+=keySquare[rowNmbFirst*5+columnNmbSecond].toUpperCase();
            }else{
              newChars+=keySquare[rowNmbFirst*5+columnNmbSecond].toLowerCase();
            }
            if(secondChar==secondChar.toUpperCase()){
              newChars+=keySquare[rowNmbSecond*5+columnNmbFirst].toUpperCase();
            }else{
              newChars+=keySquare[rowNmbSecond*5+columnNmbFirst].toLowerCase();
            }
          }
          firstChar='';
          secondChar='';
        }
        i++;
    }
    newCharsLoc=0;
    for(var i=0; i<chars.length; i++){
      var currentChar=chars.charAt(i);
      if (currentChar.toUpperCase()==subs[0].toUpperCase()){
        currentChar=subs[1];
      }
      var keyLoc=keySquare.indexOf(currentChar.toUpperCase());
      if (keyLoc>-1){
        newDiv+=newChars.charAt(newCharsLoc);
        newCharsLoc++;
      }else{
        newDiv+=chars.charAt(i);
      }
    }
    while (newCharsLoc<newChars.length){
      newDiv+=newChars.charAt(newCharsLoc);
      newCharsLoc++;
    }
    newDiv+='</textarea>';
    document.getElementById('playfairCipherDiv').innerHTML=newDiv;
}
</script>
                    <header>
                    </header>
                    <?php
                                    function Cipher($ch, $key)
```

```
                                        {
                                        if (!ctype_alpha($ch))
                                        return $ch;
                                        $offset = ord(ctype_upper($ch) ? 'A' : 'a');
                                        return chr(fmod(((ord($ch) + $key) - $offset),
26) + $offset);
                                        }
                ?>
                <div id="colorlib-work">
                        <div class="container">
                                <div class="row text-center">
                                        <h2 class="bold"><i>Cipher</i></h2>
                                </div>
                                <div class="row">
                                        <div class="col-md-8">
                <div class="text-inner text-inner-right">
                <h2>PENGATURAN</h2>
                <p><i>"catatan"</i> Ubah Pengaturan Sesuai Keinginan,Jangan Lupa
Untuk Mengganti Metode, ENKRIPSI/DESKRIPSI !</p>
                <table width="100%">
                <tr><td>
                <div class="form-group">
                <div class="form-line" >
                                        <input type="radio" name="manualKey"
id="playfairCipherKeyCheck" checked="true" onClick="setEnabled(1)">
        <input class="form-control no-resize" placeholder="Masukan kata kunci..."
type="text" id="playfairCipherKey" onKeyDown="populateSquare(2)"
onKeyUp="populateSquare(2)">
        </div>
        </div></td></tr>
        <tr><td>
        Huruf yang diganti : <input type="radio" name="replaceOmit"
id="playfairCipherReplaceCheck" checked="true" onClick="setEnabled(2)">
         <select  id="playfairCipherReplace1"
onChange="repopulateReplace2(1);populateSquare(3);">
        <option value="A">A</option>
        <option value="B">B</option>
        <option value="C">C</option>
        <option value="D">D</option>
        <option value="E">E</option>
        <option value="F">F</option>
        <option value="G">G</option>
        <option value="H">H</option>
        <option value="I">I</option>
        <option selected value="J">J</option>
        <option value="K">K</option>
        <option value="L">L</option>
```

```
<option value="M">M</option>
<option value="N">N</option>
<option value="O">O</option>
<option value="P">P</option>
<option value="Q">Q</option>
<option value="R">R</option>
<option value="S">S</option>
<option value="T">T</option>
<option value="U">U</option>
<option value="V">V</option>
<option value="W">W</option>
<option value="X">X</option>
<option value="Y">Y</option>
<option value="Z">Z</option>
</select> diganti dengan <select id="playfairCipherReplace2"
onChange="populateSquare(3)">
<option value="A">A</option>
<option value="B">B</option>
<option value="C">C</option>
<option value="D">D</option>
<option value="E">E</option>
<option value="F">F</option>
<option value="G">G</option>
<option value="H">H</option>
<option selected value="I">I</option>
<option value="K">K</option>
<option value="L">L</option>
<option value="M">M</option>
<option value="N">N</option>
<option value="O">O</option>
<option value="P">P</option>
<option value="Q">Q</option>
<option value="R">R</option>
<option value="S">S</option>
<option value="T">T</option>
<option value="U">U</option>
<option value="V">V</option>
<option value="W">W</option>
<option value="X">X</option>
<option value="Y">Y</option>
<option value="Z">Z</option>
</select><hr></td></tr>
<tr><td>

Jika jumlah huruf ganjil, tambahkan huruf : <select
id="playfairCipherDoubleChar" onChange="playfairCipher()">
```

```
<option value="A">A</option>
<option value="B">B</option>
<option value="C">C</option>
<option value="D">D</option>
<option value="E">E</option>
<option value="F">F</option>
<option value="G">G</option>
<option value="H">H</option>
<option value="I">I</option>
<option value="J">J</option>
<option value="K">K</option>
<option value="L">L</option>
<option value="M">M</option>
<option value="N">N</option>
<option value="O">O</option>
<option value="P">P</option>
<option value="Q">Q</option>
<option value="R">R</option>
<option value="S">S</option>
<option value="T">T</option>
<option value="U">U</option>
<option value="V">V</option>
<option value="W">W</option>
<option value="X">X</option>
<option value="Y">Y</option>
<option selected value="Z">Z</option>
</select> diakhir.<hr></td></tr>
<tr><td>
<input type="radio" name="doubles" id="playfairCipherBreakCheck"
onClick="setEnabled(3)">
</td></tr>
<tr><td>Pilih metode : <select id="playfairCipherMethod"
onChange="playfairCipher()">
<option selected value="Decrypt">Dekripsi</option>
<option value="Encrypt">Enkripsi</option>
</select></td></tr>
</table>
                                        </div>
                                        </div>
                                        <div class="col-md-4">
                                        <div class="text-inner text-inner-left">
                                        <h2 align="center">Tabel PLAYFAIR</h2>
                                        <p align="center">Matriks 5x5</p>
                                        <table align="center">
                                <tr><td>
                                <input type="text" size="1" id="0" value="A"
onChange="checkValues(0)" onKeyDown="checkValues(0)" onKeyUp="checkValues(0)">
```

```
                                    <input type="text" size="1" id="1" value="B"
onChange="checkValues(1)" onKeyDown="checkValues(1)" onKeyUp="checkValues(1)">
                                    <input type="text" size="1" id="2" value="C"
onChange="checkValues(2)" onKeyDown="checkValues(2)" onKeyUp="checkValues(2)">
                                    <input type="text" size="1" id="3" value="D"
onChange="checkValues(3)" onKeyDown="checkValues(3)" onKeyUp="checkValues(3)">
                                    <input type="text" size="1" id="4" value="E"
onChange="checkValues(4)" onKeyDown="checkValues(4)" onKeyUp="checkValues(4)">
                                    </td></tr>
                                    <tr><td>
                                    <input type="text" size="1" id="5" value="F"
onChange="checkValues(5)" onKeyDown="checkValues(5)" onKeyUp="checkValues(5)">
                                    <input type="text" size="1" id="6" value="G"
onChange="checkValues(6)" onKeyDown="checkValues(6)" onKeyUp="checkValues(6)">
                                    <input type="text" size="1" id="7" value="H"
onChange="checkValues(7)" onKeyDown="checkValues(7)" onKeyUp="checkValues(7)">
                                    <input type="text" size="1" id="8" value="I"
onChange="checkValues(8)" onKeyDown="checkValues(8)" onKeyUp="checkValues(8)">
                                    <input type="text" size="1" id="9" value="K"
onChange="checkValues(9)" onKeyDown="checkValues(9)" onKeyUp="checkValues(9)">
                                    </td></tr>
                                    <tr><td>
                                    <input type="text" size="1" id="10" value="L"
onChange="checkValues(10)" onKeyDown="checkValues(10)"
onKeyUp="checkValues(10)">
                                    <input type="text" size="1" id="11" value="M"
onChange="checkValues(11)" onKeyDown="checkValues(11)"
onKeyUp="checkValues(11)">
                                    <input type="text" size="1" id="12" value="N"
onChange="checkValues(12)" onKeyDown="checkValues(12)"
onKeyUp="checkValues(12)">
                                    <input type="text" size="1" id="13" value="O"
onChange="checkValues(13)" onKeyDown="checkValues(13)"
onKeyUp="checkValues(13)">
                                    <input type="text" size="1" id="14" value="P"
onChange="checkValues(14)" onKeyDown="checkValues(14)"
onKeyUp="checkValues(14)">
                                    </td></tr>
                                    <tr><td>
                                    <input type="text" size="1" id="15" value="Q"
onChange="checkValues(15)" onKeyDown="checkValues(15)"
onKeyUp="checkValues(15)">
                                    <input type="text" size="1" id="16" value="R"
onChange="checkValues(16)" onKeyDown="checkValues(16)"
onKeyUp="checkValues(16)">
```

```
                                    <input type="text" size="1" id="17" value="S"
onChange="checkValues(17)" onKeyDown="checkValues(17)"
onKeyUp="checkValues(17)">
                                    <input type="text" size="1" id="18" value="T"
onChange="checkValues(18)" onKeyDown="checkValues(18)"
onKeyUp="checkValues(18)">
                                    <input type="text" size="1" id="19" value="U"
onChange="checkValues(19)" onKeyDown="checkValues(19)"
onKeyUp="checkValues(19)">
                                    </td></tr>
                                    <tr><td>
                                    <input type="text" size="1" id="20" value="V"
onChange="checkValues(20)" onKeyDown="checkValues(20)"
onKeyUp="checkValues(20)">
                                    <input type="text" size="1" id="21" value="W"
onChange="checkValues(21)" onKeyDown="checkValues(21)"
onKeyUp="checkValues(21)">
                                    <input type="text" size="1" id="22" value="X"
onChange="checkValues(22)" onKeyDown="checkValues(22)"
onKeyUp="checkValues(22)">
                                    <input type="text" size="1" id="23" value="Y"
onChange="checkValues(23)" onKeyDown="checkValues(23)"
onKeyUp="checkValues(23)">
                                    <input type="text" size="1" id="24" value="Z"
onChange="checkValues(24)" onKeyDown="checkValues(24)"
onKeyUp="checkValues(24)">
                                    </td></tr>
                                    </table>
                                                </div>
                                            </div>
</div>
<div class="row">
  <br><br>
                                            <div class="col-md-12">
        <div class="text-inner text-inner-right">
        <h2>KOLOM ENKRIPSI/DESKRIPSI</h2>
        <p></p>
<table width="100%">
<tr><td><div class="form-group">
                        <div class="form-line">
<textarea  rows="4" class="form-control no-resize" placeholder="Masukan PlainText..."
id="playfairCipherText" onKeyDown="playfairCipher()"
onKeyUp="playfairCipher()"></textarea>
                                                                    </div>
                                                                    </div>
                                                                </td></tr>
                                                                <tr><td><input
```

```
class="btn bg-teal waves-effect" name="Reset fields" value="RESET TEKS" type="button"
onclick="resetFields(this.form,0);"><hr></td></tr>
        <tr><td><div id="playfairCipherDiv"></div></td></tr>
        </table>
                                                        </div>
                                                </div>
                                        </div>
                                </div>
                        </div>
                </div>
                <div id="colorlib-subscribe">
                        <div class="overlay"></div>
                        <div class="container">
                                <div class="row">
                                        <div class="col-md-8 col-md-offset-2 text-center
colorlib-heading animate-box">
                                                <h2>Terima Kasih</h2>
                                                <p>Sudah menggunakan aplikasi ini</p>
                                        </div>
                                </div>
                        </div>
                </div>
                <footer>
                        <div id="footer">
                                <div class="container">
                                        <div class="row">
                                                <div class="col-md-4 col-pb-sm text-
center">
                                                        <div class="row">
                                                                <div class="col-md-10">

        <h2>addres</h2>
                                                                        <p>Tanjung
Mulia, <br>Kec.Kp.Rakyat LABUSEL 21463</p>              </div>
                                                        </div>
                                                </div>
                                                <div class="col-md-4 col-pb-sm text-
center">
                                                        <h2>email saya</h2>
                                                        <p><a
href="">m.prist88@gmail.com</a></p>
                                                </div>
                                                <div class="col-md-4 col-pb-sm text-
center">
                </footer>
        </body>
</html>
```