



**PERANCANGAN APLIKASI KEAMANAN DATA MENGGUNAKAN
ALGORITMA KRIPTOGRAFI DES**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

NAMA : MUHAMMAD HARYANSYAH
NPM : 1414370306
PROGRAM STUDI : SISTEM KOMPUTER

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

ABSTRAK

MUHAMMAD HARYANSYAH
Percancangan Aplikasi Keamanan Data Menggunakan
Algoritma Kriptografi DES

Kriptografi adalah bidang ilmu untuk menjaga keamanan pesan (*message*). Kriptografi telah banyak diimplementasikan di banyak hal. Smart card, Anjungan Tunai Mandiri (*ATM*), *Pay TV*, *Mobile Phone*, dan Komputer adalah beberapa contoh produk teknologi yang menggunakan kriptografi untuk keamanannya. Cara kerjanya adalah dengan mengubah pesan asli yang dapat dimengerti/dibaca manusia (plainteks) ke bentuk lain yang tidak dapat dimengerti/dibaca oleh manusia (cipherteks). Proses transformasi plainteks menjadi chiperteks diistilahkan dengan enkripsi. Sedang proses pengembalian pesan chiperteks menjadi plainteks diistilahkan dengan dekripsi. Ada banyak algoritma kriptografi, dalam penelitian ini aplikasi kriptografi yang dikembangkan menggunakan algoritma simetri DES (*Data Encryption Standard*) dengan bahasa pemrograman *Java*. DES menggunakan sandi blok kunci simetrik dengan ukuran blok 64-bit dan ukuran kunci 56-bit.

Kata Kunci: Plainteks, Chiperteks, Kriptografi, Enkripsi, Dekripsi.

DAFTAR ISI

LEMBAR JUDUL	
LEMBAR PENGESAHAN	
ABSTRAK	
KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	v
DAFTAR TABEL	vi
DAFTAR LAMPIRAN	vii
DAFTAR ISTILAH	viii

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Metode Penelitian	3
1.6.1 Analisis	3
1.6.2 Perancangan	3
1.6.3 Pengkodean	4
1.6.4 Ujicoba program	4
1.6.5 Pembuatan kesimpulan	4
1.7 Sistematika Penulisan	4

BAB II LANDASAN TEORI

2.1 Kriptografi	6
2.2 Tujuan Kriptografi	6
2.3 Kriptografi DES	7
2.4 Algoritma	7
2.4.1 Kriteria algoritma	8
2.4.2 Tujuan dan fungsi algoritma	9
2.4.3 Klasifikasi algoritma	10
2.5 Keamanan data	11
2.5.1 Elemen keamanan data dan informasi	12
2.5.2 Perlunya pengamanan pada setiap data dan informasi	13
2.6 Bilangan Biner	14
2.7 Bilangan Hexadesimal	16
2.7.1 Konversi dari heksadesimal ke decimal	17
2.7.2 Konversi dari desimal ke heksadesimal	18
2.8 Bahasa Pemrograman Java	19
2.8.1 Sejarah java	21
2.8.3 Karakteristik java	23
2.9 Flowchart	26

BAB III METODE PENELITIAN

3.1	Analisa Permasalahan	29
3.2	Analisa Sistem	29
3.3	Algoritma DES (Data Encryption Standard)	29
3.4	Pembangkitan Kunci Internal	32
3.5	Flowchart Sistem	34
3.6	Perancangan Antar Muka	35

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1	Tampilan Program Kriptografi DES	36
4.1.1	Tampilan enkripsi program kriptografi des	36
4.2	Proses Enkripsi Dari Kriptografi DES	38

BAB V PENUTUP

5.1	Kesimpulan	56
5.2	Saran	56

DAFTAR PUSTAKA
BIOGRAFI PENULIS
LAMPIRAN

KATA PENGANTAR

Assalaamua'laikum Warahmatullaahi Wabarakaatuh.

Alhamdulillah rabbil 'aallamin, puji syukur penulis panjatkan kehadirat Allah SWT karena berkat Rahmat dan Karunia-Nya penulis dapat menyelesaikan skripsi ini. Shalawat beserta salam semoga senantiasa terlimpah curahkan kepada Baginda Nabi Muhammad SAW, kepada keluarganya, para sahabatnya, dan kepada umatnya hingga akhir zaman, Amiin.

Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan Pendidikan Strata Satu (S1) dan memperoleh gelar Sarjana Komputer pada Program Studi Sistem Komputer Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan.

Dalam penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis dengan kerendahan hati mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada yang terhormat:

1. Bapak Dr. H. Muhammad Isa Indrawan, SE., MM., selaku Rektor Universitas Pembangunan Panca Budi Medan.
2. Ibu Sri Shindi Indira, ST., M.Sc., selaku Dekan Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan.
3. Bapak Dr. Muhammad Iqbal, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer.
4. Bapak Andysah Putera Utama Siahaan, S.Kom., M.Kom., Ph.D., selaku Dosen Pembimbing I yang telah memberikan curahan pengetahuan serta bimbingan dalam penyelesaian skripsi ini.
5. Ibu Sri Wahyuni, S.Kom., M.Kom., selaku Dosen Pembimbing II yang juga telah memberikan curahan pengetahuan serta bimbingan dalam penyelesaian skripsi ini.
6. Seluruh Dosen Universitas Pembangunan Panca Budi yang telah mengajarkan banyak hal kepada penulis.
7. Kedua orang tua ayahanda Her Supriadi dan ibunda Sulastri dan keluarga penulis tercinta yang telah banyak memberikan dukungan kepada penulis secara moril maupun materil hingga skripsi ini dapat selesai.
8. Sahabat dan rekan seperjuangan tercinta yang tiada henti memberi dukungan dan motivasi kepada penulis.
9. Seluruh pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya.

Dengan sepuh hati penulis memohon maaf atas segala kekurangan dan kelemahan dalam penyusunan skripsi ini baik sistematika penulisan maupun penggunaan bahasa yang tidak sesuai dengan aturan yang telah ditetapkan. Semua itu disebabkan ketidaksengajaan dan kesilapan penulis dalam mengerjakan skripsi ini. Penulis menyadari skripsi ini masih jauh dari kesempurnaan, baik dalam penyajian skripsi, penyajian materi, pembahasan masalah, maupun penyusunan kata-kata. Untuk itu penulis mengharapkan kritik dan saran dari berbagai sumber

yang bersifat membangun dalam penyempurnaan skripsi ini, agar lebih bermanfaat bagi penulis dan bagi kita semua. Semoga skripsi ini dapat bermanfaat bagi semua pihak yang membacanya. Akhir kata penulis ucapkan banyak terima kasih.

Wassalaamua'laikum Warahmatullaahi Wabarakaatuh.

Medan, 27 Agustus 2019
Penulis,

(MUHAMMAD HARYANSYAH)
1 4 1 4 3 7 0 3 0 6

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada dasarnya dalam membangun sebuah keamanan komputer diperlukan suatu sistem pengamanan data atau *file* yang kita miliki. Misalnya seseorang yang biasa menyimpan data-data penting kedalam suatu *file* dengan karakter yang tidak terkode (*plaintext*), sangatlah rawan apabila tidak berhati-hati. Apabila jika data tersebut disimpan dalam suatu komputer yang digunakan secara bebas, bagi siapa saja yang ingin menggunakannya karena seperti saat ini yang kita ketahui aktifitas pencurian data baik itu terhadap komputer yang terhubung pada suatu jaringan maupun tidak, sudah menjadi hal yang sering terdengar dan tidak asing lagi bagi kalangan intelektual khususnya dan masyarakat luas pada umumnya. Hal-hal yang berkaitan dengan pengamanan data-data penting tersebut haruslah benar-benar diperhatikan agar data yang akan disampaikan atau masih tersimpan dalam komputer kita tetap aman dari orang-orang yang tidak bertanggung jawab.

Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, penulis menggunakan algoritma kriptografi *DES* untuk proses enkripsi dan deskripsi data.

Pada penelitian ini penulis menarik judul “**Perancangan Aplikasi Keamanan Data Menggunakan Algoritma Kriptografi DES**”

1.2 Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang ada sebelumnya,

Penulis dapat merumuskan masalah yaitu :

1. Bagaimana membangun aplikasi keamanan data menggunakan algoritma kriptografi *DES*?
2. Bagaimana cara menyandikan data dengan menggunakan algoritma kriptografi *DES*?

1.3 Batasan Masalah

Agar dalam perancangan ini dapat mencapai sasaran dan tujuan yang diharapkan, maka permasalahan yang ada dibatasi sebagai berikut:

1. Membahas enkripsi dan dekripsi teks pada data menggunakan algoritma *DES* saat mengirim dan menerima data.
2. Pada aplikasi keamanan tipe data yang diinputkan berupa *file* yang berekstensi *.txt.
3. Pada aplikasi Kriptografi *DES* menggunakan *base 64*.

1.4 Tujuan Penelitian

Berikut ini beberapa tujuan penelitian yang akan dibahas dalam penelitian ini yaitu sebagai berikut :

1. Merancang dan membangun aplikasi yang mampu melakukan enkripsi dan dekripsi dengan menerapkan algoritma kriptografi *DES* sehingga dirasakan aman.

1.5 Manfaat Penelitian

Berikut ini beberapa manfaat penelitian yang akan dibahas dalam penelitian ini yaitu sebagai berikut :

1. Mengamankan data yang telah disandikan sehingga tidak mudah untuk diakses oranglain.
2. Menerapkan ilmu pengetahuan yang telah dipelajari, sehingga dapat mengolah data yang diinginkan untuk mengamankan data tersebut.

1.6 Metode Penelitian

Metode penelitian yang digunakan penulis dalam pembuatan skripsi ini adalah mengambil data dengan metode :

1.6.1 Analisis

Menganalisa sistem merupakan langkah dalam membuat sistem baru. Dalam analisis sistem digunakan metode-metode yang telah dijelaskan pada rancangan penelitian.

1.6.2 Perancangan

Perancangan sistem merupakan tahap pengembangan setelah analisis sistem dilakukan. Beberapa proses dalam perancangan aplikasi tersebut, seperti

perancangan *input* dan *output flowchart* keamanan data menggunakan algoritma kriptografi *DES*.

1.6.3 Pengkodean

Membuat aplikasi yang telah dirancang dengan mengaplikasikan kedalam bentuk bahasa pemrograman *java*, menggunakan *software netbeans IDE*.

1.6.4 Ujicoba program

Setelah program selesai, maka akan dilakukan uji coba program untuk mengetahui apakah program tersebut telah bekerja dengan benar dan sesuai dengan sistem yang dibuat.

1.6.5 Pembuatan kesimpulan

Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari makalah skripsi ini dan kesimpulan tentang program yang telah dibuat.

1.7 Sistematika Penulisan

Didalam penyusunan skripsi ini pembahasan skripsi dibagi dalam lima bab yang dapat membantu dan memudahkan pembaca, memahami, dan mengerti akan penulisan skripsi ini. Adapun sistematikanya adalah :

1. BAB I : Pendahuluan

Bab ini merupakan bagian yang berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan.

2. BAB II : Landasan Teori

Bab ini berisi uraian teoritis mengenai pengertian-pengertian dan mengenai pembahasan pemrograman yang digunakan.

3. BAB III : Analisa Dan Perancangan Sistem

Bab ini membahas tentang perancangan umum maupun uraian lebih lanjut mengenai perancangan sistem dalam pembuatan perangkat lunak. Uraian perancangan sistem ini meliputi perancangan data mengenai data input dan output sistem, perancangan antar muka dalam desain dan implementasi yang di gunakan.

4. BAB IV : Implementasi Sistem

Bab ini berisi uraian pengertian dan implementasi sistem, spesifikasi aplikasi. Pengujian program dan tampilan halaman aplikasi perangkat lunak yang dirancang.

5. BAB V : Penutup

Pada bab ini akan dijelaskan mengenai kesimpulan yang bisa ditarik penyelesaian masalah dan pembuatan program juga beberapa saran yang bisa diberikan untuk dapat memperbaiki kekurangan pada program.

BAB II

LANDASAN TEORI

2.1 Kriptografi

(Munir, 2008) Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*) selain itu ada pengertian tentang kriptografi yaitu kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata “seni” di dalam definisi di atas maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata “*graphy*” di dalam “*cryptography*” itu sendiri sudah menyiratkan sebuah seni.

2.2 Tujuan Kriptografi

Menurut (Munir, 2008) Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi, yaitu:

1. *Confidentiality* (kerahasiaan), yaitu memberikan kerahasiaan pesan dan menyimpan data dengan menyembunyikan data dengan menyembunyikan informasi lewat teknik-teknik enkripsi.
2. *Message integrity* (integritas data), yaitu memberikan jaminan bahwa dari setiap bagian tidak mengalami perubahan dari saat data dibuat / dikirim sampai dengan saat data tersebut di buka.

3. *Non-repudiation* (nirpenyangkalan), yang memberikan cara untuk membuktikan bahwa suatu dokumen datang dari setiap seseorang apabila ia mencoba menyangkal memiliki dokumen tersebut.
4. *Authentication* (autentikasi), yang memberikan dua layanan. Yang pertama mengidentifikasi keaslian dari suatu pesan dan memberikan jaminan keotentikannya. Kedua, untuk menguji identitas seseorang apabila ia akan memasuki sebuah sistem.

2.3 Kriptografi DES

(Primartha Rifkie, 2011) Algoritma *DES* (*Data Encryption Standard*) merupakan algoritma enkripsi yang paling banyak digunakan didunia yang diadopsi oleh *NIST* (*National Institute of Standards and Technology*) sebagai standar pengolah informasi Federal AS. Data *plaintext* dienkripsi dalam blok blok 64 bit menjadi 64 bit data ciphertext menggunakan kunci 56 bit kunci internal (internal key). *DES* mentransformasikan input 64 bit dalam beberapa tahap enkripsi ke dalam output 64 bit. Dengan demikian, *DES* termasuk blok cipher. Dengan tahapan dan kunci yang sama, *DES* digunakan untuk membalik enkripsi. Kunci internal pada algoritma *DES* dibangkitkan dari kunci eksternal (*external key*) 64 bit.

2.4 Algoritma

(Fathoni, 2015) Apa yang dimaksud dengan algoritma (*algorithm*)? Dalam ilmu komputer dan matematika, pengertian algoritma adalah suatu urutan dari beberapa langkah logis dan sistematis yang digunakan untuk menyelesaikan masalah tertentu.

Pendapat lain mengatakan definisi algoritma adalah proses atau serangkaian aturan yang harus diikuti dalam perhitungan atau operasi pemecahan masalah lainnya, terutama oleh komputer. Dengan kata lain, semua susunan logis yang diurutkan berdasarkan sistematika tertentu dan digunakan untuk memecahkan suatu masalah dapat disebut dengan algoritma.

Algoritma digunakan untuk melakukan penghitungan, penalaran otomatis, serta mengolah data pada komputer dengan menggunakan *software*. Dalam algoritma terdapat rangkaian terbatas dari beberapa intruksi untuk menghitung suatu fungsi yang jika dieksekusi dan diproses akan menghasilkan *output*, lalu berhenti pada kondisi akhir yang sudah ditentukan.

Berikut ini adalah bentuk dasar algoritma:

1. Algoritma Sekuensial (*Sequence Algorithm*)
2. Algoritma Perulangan (*Looping Algorithm*)
3. Algoritma Percabangan atau Bersyarat (*Conditional Algorithm*)
4. Baca juga: Pengertian Komputer

2.4.1 Kriteria algoritma

Algoritma memiliki lima ciri utama yang saling berhubungan satu dengan lainnya. Menurut (Donald E. Knuth), adapun kriteria algoritma adalah sebagai berikut:

Ada *Input*, yaitu permasalahan yang dihadapi dan akan dicarikan solusinya.

Algoritma memiliki nol atau lebih *input* (masukan).

1. Ada Proses, yaitu rencana atau langkah-langkah yang harus dilakukan untuk mencapai tujuan akhir.
2. Ada *Output*, yaitu solusi atau tampilan akhir yang didapatkan dari suatu algoritma yang memiliki minimal satu *output*.
3. Ada intruksi-intruksi yang jelas dan tidak ambigu, yaitu instruksi yang jelas dalam algoritma sehingga tidak terjadi kesalahan dalam menghasilkan *output*.
4. Ada tujuan akhir yang dicapai, yaitu akhir dari program dimana program akan berhenti ketika tujuan akhir telah tercapai.

2.4.2 Tujuan dan fungsi algoritma

Pada dasarnya tujuan dan fungsi utama dari algoritma adalah untuk memecahkan suatu masalah. Lebih jelasnya, adapun tujuan dan fungsi algoritma adalah sebagai berikut:

1. Untuk membantu menyederhanakan suatu program yang rumit dan besar.
2. Untuk memudahkan dalam membuat sebuah program untuk masalah tertentu.
3. Algoritma dapat digunakan berkali-kali untuk menyelesaikan suatu permasalahan.
4. Untuk meminimalisir penulisan program secara berulang-ulang.
5. Untuk memudahkan membuat program yang lebih rapi dan terstruktur sehingga lebih mudah dipahami dan dikembangkan.

6. Memudahkan proses modifikasi pada program karena bisa dilakukan hanya pada satu modul tanpa harus mengubah modul lainnya.
7. Ketika terjadi kesalahan, algoritma dapat membantu menemukannya karena alur kerja yang jelas.

2.4.3 Klasifikasi algoritma

Algoritma dapat diklasifikasikan berdasarkan implementasinya. Mengacu pada pengertian algoritma di atas, adapun klasifikasi algoritma adalah sebagai berikut:

1. Rekursi dan Iterasi; Algoritma rekursi adalah algoritma yang memanggil dirinya sendiri secara berulang-ulang. Sedangkan algoritma iterasi adalah algoritma yang memakai konstruksi berulang dimana terkadang terdapat data tambahan pada struktur yang dibuat.
2. *Logical*; Algoritma *logical* adalah algoritma yang dapat memposisikan diri seperti logika deduksi yang terkontrol.
3. Serial, Parallel, atau Terdistribusi; Algoritma serial adalah algoritma yang menjalankan satu instruksi saja. Algoritma parallel adalah algoritma yang dapat mengerjakan suatu perintah dalam waktu yang sama. Sedangkan algoritma terdistribusi adalah algoritma yang memakai banyak mesin yang terkoneksi dengan jaringan.
4. Deterministik atau Non-deterministik; Algoritma deterministik adalah algoritma yang dapat memecahkan suatu masalah dengan keputusan yang

tepat. Sedangkan algoritma Non-deterministik adalah algoritma yang memecahkan suatu masalah dengan metode penerkaan.

5. Tepat atau Perkiraan; Suatu algoritma mungkin saja memiliki solusi yang tepat, atau setidaknya mempunyai perkiraan yang mendekati solusi yang benar. Dalam merumuskannya dapat dilakukan dengan strategi deterministik ataupun secara acak.
6. Algoritma Quantum; Algoritma quantum adalah algoritma yang menggunakan model realistik dari komputasi quantum.

2.5 Keamanan data

(Paryati, 2008) Aman sering diartikan dengan istilah "*free from danger*" yang artinya bebas dari ancaman bahaya.

Keamanan biasanya digambarkan sebagai kebebasan dari bahaya atau sebagai kondisi keselamatan. Keamanan komputer, secara rinci adalah perlindungan data di dalam suatu sistem melawan terhadap otorisasi tidak sah, modifikasi, atau perusakan dan perlindungan sistem komputer terhadap penggunaan tidak sah atau modifikasi.

Keamanan informasi adalah cabang studi dari teknologi informasi yang mengkhususkan diri untuk mempelajari metode dan teknik untuk melindungi informasi dan sistem informasi dari akses, penggunaan, penyebaran, perusakan, perubahan, dan penghancuran tanpa otorisasi yang sah.

Ada empat aspek utama dalam keamanan data dan informasi.

1. *Privacy/Confidentiality* yaitu usaha menjaga data informasi dari orang yang tidak berhak mengakses (memastikan bahwa data atau informasi pribadi kita tetap pribadi).
2. *Integrity* yaitu usaha untuk menjaga data atau informasi tidak diubah oleh yang tidak berhak.
3. *Authentication* yaitu usaha atau metoda untuk mengetahui keaslian dari informasi, misalnya apakah informasi yang dikirim dibuka oleh orang yang benar (asli) atau layanan dari server yang diberikan benar berasal dari server yang dimaksud.
4. *Availability* berhubungan dengan ketersediaan sistem dan data (informasi) ketika dibutuhkan.

Keempat aspek ini menjadi dasar untuk melakukan pengamanan terhadap data dan informasi.

Keamanan komputer adalah sebuah proses, yang harus dijalankan untuk mengamankan sistem dan dalam penerapannya harus dilakukan dengan menyeluruh. Bagian-bagian keamanan yang ada pada *Data Center* mengacu pada empat aspek dasar keamanan yang disebutkan sebelumnya.

2.5.1 Elemen keamanan data dan informasi :

Berikut ini adalah elemen-elemen keamanan data dan informasi :

1. Keamanan informasi harus sejalan dengan misi organisasi.
2. Keamanan informasi merupakan bagian integral dari praktik manajemen yang baik.

3. Keamanan informasi harus efektif dalam hal harga.
4. Tanggung jawab dan kewenangan keamanan informasi harus dijelaskan secara eksplisit.
5. Pemilik sistem memiliki tanggung jawab keamanan diluar organisasinya.
6. Keamanan informasi memerlukan pendekatan yang komprehensif dan terintegrasi.
7. Keamanan informasi harus dievaluasi ulang secara periodic.
8. Keamanan informasi dibatasi oleh faktor social.

Pihak yang memiliki peranan dan tanggung jawab dalam penerapan usaha pengamanan sistem informasi

- a. Jajaran Manajemen Senior
- b. Manajer Fungsional
- c. Manajer keamanan informasi komputer
- d. Staf ahli teknologi
- e. Organisasi pendukung
- f. Pengguna atau user

2.5.2 Perlunya pengamanan pada setiap data dan informasi

Banyak pertanyaan yang mungkin timbul di pikiran kita. Mengapa kita membutuhkan kemandan, atau seberapa aman, atau apa yang hendak kita lindungi, seberapa pentingkah data kita sehingga perlu memusingkan diri dengan masalah keamanan. Pertama akan dijelaskan mengapa kita membutuhkan keamanan. Dalam dunia global dengan komunikasi data yang selalu berkembang dengan pesat dari

waktu ke waktu, koneksi internet yang semakin murah, masalah keamanan seringkali luput dari perhatian pemakai komputer dan mulai menjadi isu yang sangat serius. Keamanan data saat ini telah menjadi kebutuhan dasar karena perkomputeran secara global telah menjadi tidak aman. Sementara data anda berpindah dari satu titik ke titik lainnya di Internet, mungkin data tersebut melewati titik-titik lain dalam perjalanannya, yang memberikan kesempatan kepada orang lain untuk mengganggunya. Bahkan mungkin beberapa pengguna dari sistem anda, mengubah data yang dimiliki menjadi sesuatu yang tidak anda inginkan. Akses yang tidak terotorisasi ke dalam sistem anda mungkin bisa diperoleh oleh penyusup, yang disebut '*cracker*', yang kemudian menggunakan kemampuannya untuk mencuri data, atau pun melakukan hal - hal lain yang merupakan mimpi buruk bagi anda.

Berbagai ancaman dan jenis-jenisnya ancaman yang dapat merusak data dan informasi

1. Sistem *Avability Secrecy Integrity*.
2. *Hardware* Dicuri atau dirusak.
3. *Software* Program dihapus *Software* di *copy* Program dimodifikasi.
4. Data *File* dihapus atau dirusak Dicuri, disadap *File* dimodifikasi.
5. *Line* komunikasi Kabel diputus Informasi Informasi dimodifikasi.

2.6 Bilangan Biner

(Feoh & Kom, n.d.)Bilangan Biner umum digunakan pada dunia komputasi. Komputer menggunakan Bilangan Biner agar bisa saling berkomunikasi antar

komponen (*hardware*) maupun antar sesama komputer. Karena komputer hanya menggunakan bahasa mesin, yaitu apabila komputer mendapatkan sinyal listrik atau tegangan listrik, berarti bernilai 1. Apabila komputer tidak mendapatkan sinyal listrik atau tegangan listrik, berarti bernilai 0.

Bilangan Biner dapat dikonversikan ke jenis sistem bilangan lain seperti bilangan Desimal dan Oktal. Manusia sering menggunakan bilangan Desimal dalam kehidupannya sehari-hari. Bilangan Biner dan jenis sistem bilangan lainnya saling menyusun satu sama lain. Misalnya bilangan biner 00000010 merupakan angka 2 dalam bilangan Desimal. Begitupun sebaliknya, apabila angka 2 Desimal, maka berarti angka 00000010 dalam Bilangan Biner.

Bilangan Biner digunakan juga untuk menyusun suatu data ataupun file yang terdapat di dalam komputer. Misalnya terdapat suatu *file* yang berukuran 1mb. Apabila 1 *Byte* = 8 bit, berarti file tersebut tersusun atas beratus-ratus bit menjadi sebuah *file* tersebut.

Bilangan Biner juga digunakan untuk berkomunikasi antar sesama komputer dalam suatu jaringan. Karena komputer hanya mengerti Bilangan Biner, maka komputer menstransmisikan sinyal-sinyal listrik ke perangkat jaringan untuk bisa berkomunikasi satu sama lain. Bilangan Biner sangat penting dalam menyusun suatu jaringan komputer. Untuk menyusun suatu *ip address*, Bilangan Biner sangatlah diperlukan.

Jika bilangan biner dikonversi menjadi desimal maka hasilnya akan menjadi sebagai berikut

$$- \quad 0000 = 0$$

- 0001 = 1
- 0010 = 2
- 0011 = 3
- 0100 = 4
- 0101 = 5
- 0110 = 6
- 0111 = 7
- 1000 = 8
- 1001 = 9
- 1010 = 10
- 1011 = 11
- 1100 = 12
- 1101 = 13
- 1110 = 14
- 1111 = 15

dan seterusnya.

2.7 Bilangan Hexadesimal

(Feoh & Kom, n.d.) *Hexadesimal* atau sistem bilangan basis 16 adalah sebuah sistem bilangan yang menggunakan 16 simbol. Berbeda dengan sistem bilangan desimal, simbol yang digunakan dari sistem ini adalah angka 0 sampai 9, ditambah dengan 6 simbol lainnya dengan menggunakan huruf A hingga F. Sistem bilangan ini digunakan untuk menampilkan nilai alamat memori dalam

pemrograman komputer. Nilai desimal yang setara dengan setiap simbol tersebut diperlihatkan pada tabel berikut:

Tabel 2.1 Bilangan Hexadesimal

$0_{\text{hex}} = 0_{\text{dec}} = 0_{\text{oct}}$	0	0	0	0
$1_{\text{hex}} = 1_{\text{dec}} = 1_{\text{oct}}$	0	0	0	1
$2_{\text{hex}} = 2_{\text{dec}} = 2_{\text{oct}}$	0	0	1	0
$3_{\text{hex}} = 3_{\text{dec}} = 3_{\text{oct}}$	0	0	1	1
$4_{\text{hex}} = 4_{\text{dec}} = 4_{\text{oct}}$	0	1	0	0
$5_{\text{hex}} = 5_{\text{dec}} = 5_{\text{oct}}$	0	1	0	1
$6_{\text{hex}} = 6_{\text{dec}} = 6_{\text{oct}}$	0	1	1	0
$7_{\text{hex}} = 7_{\text{dec}} = 7_{\text{oct}}$	0	1	1	1
$8_{\text{hex}} = 8_{\text{dec}} = 10_{\text{oct}}$	1	0	0	0
$9_{\text{hex}} = 9_{\text{dec}} = 11_{\text{oct}}$	1	0	0	1
$A_{\text{hex}} = 10_{\text{dec}} = 12_{\text{oct}}$	1	0	1	0
$B_{\text{hex}} = 11_{\text{dec}} = 13_{\text{oct}}$	1	0	1	1
$C_{\text{hex}} = 12_{\text{dec}} = 14_{\text{oct}}$	1	1	0	0
$D_{\text{hex}} = 13_{\text{dec}} = 15_{\text{oct}}$	1	1	0	1
$E_{\text{hex}} = 14_{\text{dec}} = 16_{\text{oct}}$	1	1	1	0
$F_{\text{hex}} = 15_{\text{dec}} = 17_{\text{oct}}$	1	1	1	1

2.7.1 Konversi dari heksadesimal ke desimal

Untuk mengkonversinya ke dalam bilangan desimal, dapat menggunakan formula berikut:

Dari bilangan heksadesimal H yang merupakan untai digit $h_n, h_{n-1}, \dots, h_2, h_1, h_0$, jika dikonversikan menjadi bilangan desimal D, maka:

$$D = \sum_{k=0}^n h_k \times 16^k$$

1. Sebagai contoh, bilangan heksa 10E yang akan dikonversi ke dalam bilangan decimal, digit-digit 10E dapat dipisahkan dan mengganti bilangan A sampai F (jika terdapat) menjadi bilangan desimal padanannya. Pada contoh ini, 10E diubah menjadi barisan: 1,0,14 (E = 14 dalam basis 16).
2. Mengalikan dari tiap digit terhadap nilai tempatnya.

$$\begin{aligned} & 1 \times 16^2 + 0 \times 16^1 + 14 \times 16^0 \\ & = 256 + 0 + 14 \\ & = 270 \end{aligned}$$

Dengan demikian, bilangan 10E heksadesimal sama dengan bilangan desimal 270.

2.7.2 Konversi dari desimal ke heksadesimal

Sedangkan untuk mengkonversi sistem desimal ke heksadesimal caranya sebagai berikut (kita gunakan contoh sebelumnya, yaitu angka desimal 270):

- 270 dibagi 16 hasil: 16 sisa 14 (= E)
- 16 dibagi 16 hasil: 1 sisa 0 (= 0)
- 1 dibagi 16 hasil: 0 sisa 1 (= 1)

Dari perhitungan di atas, nilai sisa yang diperoleh (jika ditulis dari bawah ke atas) akan menghasilkan : 10E yang merupakan hasil konversi dari bilangan desimal ke heksadesimal itu.

2.8 Bahasa Pemrograman Java

(Heru, 2019) *Java* merupakan pemrograman yang sangat populer karena rentang aplikasi yang bisa dibuat menggunakan bahasa ini sangatlah luas, mulai dari komputer hingga *smartphone*. Bahasa pemrograman *java* dikembangkan pertama kali oleh *sun microsystem* yang dimulai oleh James Gosling dan diakuisisi oleh *Oracle Corporation*. Apabila sudah terbiasa dengan bahasa C dan C++, anda bisa mempelajari *java* dengan cepat.

Java bersifat *write once, run anywhere* (program yang ditulis satu kali dan dapat berjalan pada banyak platform). Dengan demikian tidak mengherankan apabila aplikasi yang dibuat menggunakan *java* bisa ditemukan di lingkungan komputer dan *smartphone* tanpa perbedaan yang berarti.

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program *java* tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram *java* banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas *java*, yang disebut dengan *Java Application Programming Interface (API)*. Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). *Java API* telah menyediakan fungsionalitas yang memadai untuk menciptakan applet dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam *java*, yaitu mempelajari bahasa *java* dan bagaimana mempergunakan kelas pada *java API*. Kelas merupakan satu-satunya

cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada *java* program *java* untuk mengkompilasi file kode sumber *java* menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi **.java*. Kompilator *javac* menghasilkan file *bytecode* kelas dengan ekstensi **.class*. Interpreter merupakan modul utama sistem *java* yang digunakan aplikasi *java* dan menjalankan program *bytecode java*.

Java juga merupakan bahasa pemrograman yang multi platform dan *multi device*. Apabila pengguna menuliskan sebuah program dengan menggunakan *java*, maka pengguna dapat menjalankannya hampir di semua komputer dan perangkat lain yang mendukung *java*, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis *java* ini dikompulasikan ke dalam *p-code* dan bisa dijalankan dengan *java virtual machine*. Fungsionalitas dari *java* ini dapat berjalan dengan platform sistem operasi yang berbeda karena sifatnya yang umum dan non-spesifik.

Java tidak memilii ketergantungan dengan platform, artinya *Java* dapat dijalankan pada sembarang komputer dan pada sembarang sistem operasi yang sering disebut dengan portabilitas. Yang menariknya portabilitas *Java* tidak terbatas pada program sumber (*source code*), melainkan juga pada tingkat *bytecode*. Dengan demikian suatu program yang sudah anda kompilasi dengan *java* di *OS Windows* dapat juga berjalan di *Macintosh* secara langsung tanpa kompilasi ulang.

Java termasuk bahasa pemrograman yang sederhana. *Java* juga dirancang agar mudah dipelajari dan digunakan secara efektif. *Java* tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan

pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan *Java* secara otomatis seperti dealokasi memori. Bagi pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa *Java* namun harus waspada karena mungkin *Java* mengambil arah (semantiks) yang berbeda dibanding C++.

Java merupakan bahasa berorientasi objek (OOP) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain. *Java* tidak mengijinkan pewarisan jamak namun menyelesaikan kebutuhan pewarisan jamak dengan fasilitas antarmuka yang lebih elegan. Karena fungsionalitasnya yang memungkinkan aplikasi *Java* mampu berjalan di beberapa platform sistem operasi yang berbeda, *Java* dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun".

2.8.1 Sejarah java

Java adalah bahasa pemrograman yang berorientasi objek (OOP). *Java* diciptakan oleh James Gosling, developer dari *Sun Microsystems* pada tahun 1991.

Perkembangan *Java* tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*. Sebagai bahasa pemrograman yang berorientasi objek, *Java* menggunakan kelas untuk membentuk suatu objek. Bahasa pemrograman *Java* pertama lahir dari *The Green Project*, yang berjalan selama 18 bulan dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan *Oak*.

Proyek ini dimotori oleh *Patrick Naughton*, *Mike Sheridan*, *James Gosling* dan *Bill Joy*, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh *Joe Palrang*. Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road di Menlo Park*. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program *Java Oak* pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai “*7 (*Star Seven*). Setelah era *Star Seven* selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek *The Green Project*. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 *Hamilton Avenue, Palo Alto*.

Perusahaan baru ini bertambah maju, jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, internet masih merupakan rintisan yang hanya dipakai di kalangan akademisi dan militer. Mereka menjadikan perambah (*browser*) *Mosaic* sebagai landasan awal untuk membuat perambah *Java* pertama yang

dinamai *Web Runner*, terinspirasi dari film 1980-an, *Blade Runner*. Pada perkembangan rilis pertama, *Web Runner* berganti nama menjadi *Hot Java*.

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber *Java* versi 1.02 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995. Sayangnya terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel *Sheraton Palace*. Tiga dari pimpinan utama proyek, *Eric Schmidt* dan *George Paolini* dari *Sun Microsystems* bersama *Marc Andreessen*, membentuk *Netscape*.

Nama *Oak*, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja “Bapak *Java*”, *James Gosling*. Nama *Oak* ini tidak dipakai untuk versi rilis *Java* karena sebuah perangkat lunak sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi “*Java*”. Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan *Gosling*. Slogan *Java* adalah “Tulis sekali, jalankan di manapun”. Sekarang ini *Java* menjadi sebuah bahasa pemrograman yang populer dan dimanfaatkan secara luas untuk pengembangan perangkat lunak. Kebanyakan perangkat lunak yang menggunakan *Java* adalah ponsel *feature* dan ponsel pintar atau *smartphone*.

2.8.3 Karakteristik java

Terdapat beberapa karakteristik dari *Java*, diantaranya yaitu:

1. Sederhana (Simple)

Bahasa pemrograman *Java* menggunakan sintaks mirip dengan *C++* namun sintaks pada *Java* telah banyak diperbaiki terutama menghilangkan

penggunaan pointer yang rumit dan *multipleinheritance*. *Java* juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Terdistribusi (*Distributed*)

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada *Java*.

4. *Interpreted*

Program *Java* dijalankan menggunakan interpreter yaitu *Java Virtual Machine (JVM)*. Hal ini menyebabkan *source code Java* yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada *platform* yang berbeda-beda.

5. *Robust*

Java mempunyai reliabilitas yang tinggi. *Compiler* pada *Java* mempunyai kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. *Java* mempunyai *runtime-Exception handling* untuk membantu mengatasi *error* pada pemrograman.

6. *Secure*

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, *Java* memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

7. *Architecture Neutral*

Program *Java* merupakan platform *independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform berbeda dengan *Java Virtual Machine*.

8. *Portable*

Source code maupun program *Java* dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

9. *Performance*

Performance pada *Java* sering dikatakan kurang tinggi. Namun performance *Java* dapat ditingkatkan menggunakan kompilasi *Java* lain seperti buatan *Inprise*, *Microsoft* ataupun *Symantec* yang menggunakan *Just In Time Compilers (JIT)*.

10. *Multithreaded*

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

11. *Dynamic*

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu class dengan menambahkan properties

ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan class tersebut.

2.9 Flowchart

(Murdock & Murdock, 2018) *Flowchart* adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *System flowchart* adalah urutan proses dalam sistem dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data. Program *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program



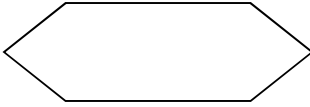

2.9.1 Pedoman-pedoman dalam membuat flowchart

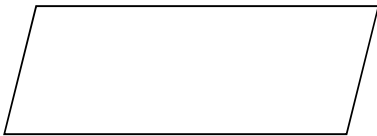

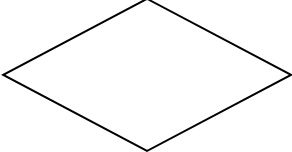
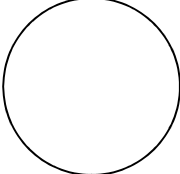
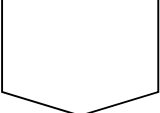
Jika seorang analis dan programmer akan membuat *flowchart*, ada beberapa petunjuk yang harus diperhatikan, seperti :

1. *Flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.

4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan Melakukan penggandaan diri.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada *flowchart* yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.
7. Gunakan simbol-simbol *flowchart* yang standar.

Tabel 2.2 Simbol-Simbol Flowchart

No	Simbol	Fungsi
1		<i>Terminator</i> ; untuk memulai dan mengakhiri suatu program.
2		Garis Alir (<i>Flow Line</i>) ; Arah aliran program
3		<i>Preparation</i> ; proses inisialisasi / pemberian harga awal
4		<i>Process</i> ; untuk membuat proses perhitungan / proses pengolahan data

5		<p><i>Input / Output Data</i> ; proses <i>input / output</i> data, parameter, informasi</p>
6		<p><i>Fredefined Process</i> (Sub Program) ; permulaan sub program / proses menjalankan sub program.</p>
7		<p><i>Decision</i> ; perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.</p>
8		<p><i>On Page Connector</i> ; penghubung bagian-bagian flowchart yang berada pada satu halaman.</p>
9		<p><i>Off Page Connector</i> ; penghubung bagian-bagian flowchart yang pada halaman berbeda</p>

Sumber (Murdock & Murdock, 2018)

DAFTAR PUSTAKA

- Des, S. (n.d.). *Standar enkripsi data*.
- Fathoni, M. dan S. (2015). Pengantar Algoritma dan Pemrograman. *Pemrograman*.
- Feoh, B. G., & Kom, S. (n.d.). *Sistem bilangan dan konversi bilangan*. 1–16.
- Heru, rohman. (2019). pengertian java. In *belajar java*.
- Munir, R. (2008). Pengantar Ilmu Kriptografi. *Penerbit Andi*.
<https://doi.org/10.1017/CBO9781107415324.004>
- Murdock, D. H., & Murdock, D. H. (2018). Flowcharts. *Auditor Essentials*, 235–239. <https://doi.org/10.1201/9781315178141-51>
- Paryati. (2008). Keamanan Sistem Informasi. *Seminar Nasional Informatika 2008 (SemnasIF 2008) UPN "Veteran" Yogyakarta, 24 Mei 2008*.
- Primartha Rifkie. (2011). Penerapan Enkripsi dan Dekripsi File Menggunakan Data Encryption Standard (DES). *ISSN: 2355-4614 / Universitas Sriwijaya*, 3(2), 371–387.
- Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. *Jurnal Teknik dan Informatika*, 5(2), 13-19.
- Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). *Jurnal Media Informatika Budidarma*, 2(2).
- Rahim, R., Supiyandi, S., Siahaan, A. P. U., Listyorini, T., Utomo, A. P., Triyanto, W. A., & Khairunnisa, K. (2018, June). TOPSIS Method Application for Decision Support System in Internal Control for Selecting Best Employees. In *Journal of Physics: Conference Series* (Vol. 1028, No. 1, p. 012052). IOP Publishing.

- Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. *Int. J. Sci. Res. Sci. Technol*, 3(6), 470-473.
- Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 100-109.
- Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.
- Khairul, k., ilhamiarsyah, u., wijaya, r. F., & utomo, r. B. (2018, september). Implementasi augmented reality sebagai media promosi penjualan rumah. In *seminar nasional royal (senar)* (vol. 1, no. 1, pp. 429-434).
- Siahaan, A. P. U., Aryza, S., Nasution, M. D. T. P., Napitupulu, D., Wijaya, R. F., & Arisandi, D. (2018). Effect of matrix size in affecting noise reduction level of filtering.
- Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." *Int. J. Eng. Trends Technol* 42.5 (2016): 218-222.
- Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas