BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Instansi

Tinjauan intansi menjelaskan sejarah perusahaan, visi dan misi perusahaan, dan struktur organisasi perusahaan tempat penelitian tugas akhir. Berikut tinjauan intansi PT Len Industri.

2.1.1 Sejarah Perusahaan

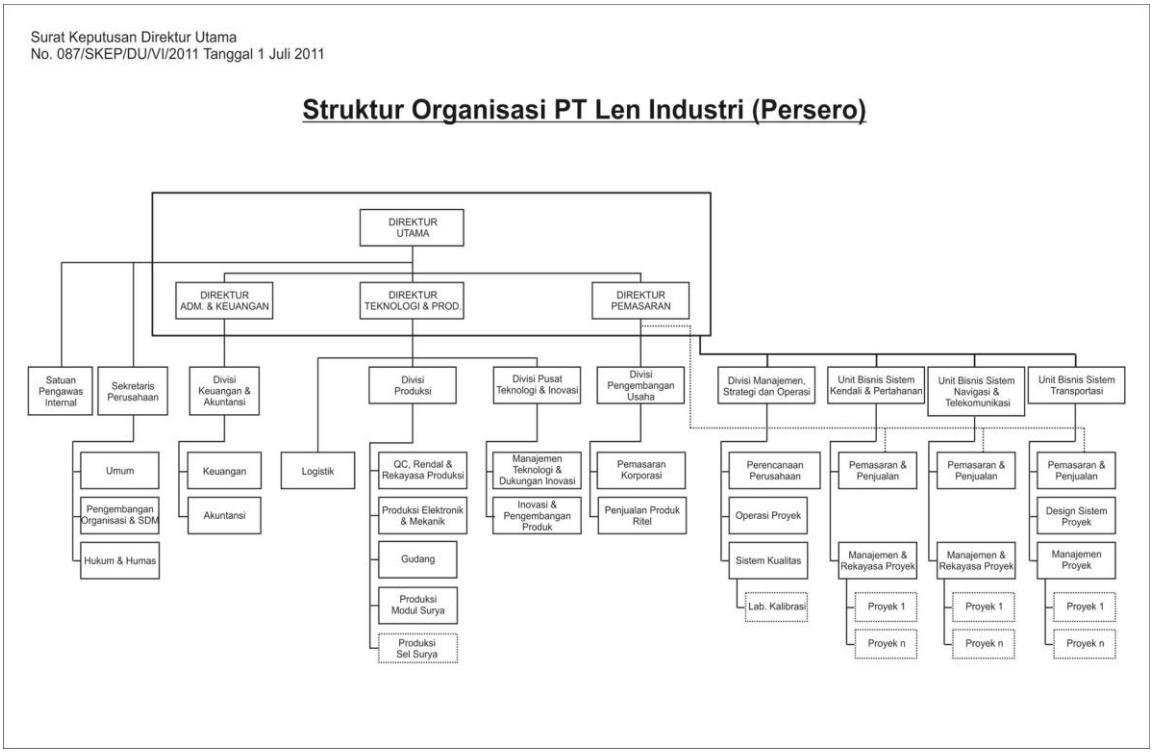
Didirikan sejak tahun 1965, LEN (Lembaga Elektronika Nasional) kemudian bertransformasi menjadi sebuah Badan Usaha Milik Negara (BUMN) pada tahun 1991. Sejak saat itu, Len bukan lagi merupakan kepanjangan dari Lembaga Elektronika Nasional (LEN), tetapi telah menjadi sebuah entitas bisnis profesional dengan nama PT Len Industri. Saat ini Len berada di bawah koordinasi Kementrian Negara BUMN. Selama ini, Len telah mengembangkan bisnis dan produk-produk dalam bidang elektronika untuk industri dan prasarana, serta telah menunjukkan pengalaman dalam berbagai bidang. Diantaranya yaitu, bidang Broadcasting selama lebih dari 30 tahun, dengan ratusan Pemancar TV dan Radio yang telah terpasang di berbagai wilayah di Indonesia. Jaringan infrastruktur telekomunikasi yang telah terentang baik di kota besar maupun daerah terpencil. Elektronika untuk pertahanan, baik darat, laut, maupun udara. Sistem Persinyalan Kereta Api di berbagai jalur kereta api di Pulau Jawa dan Sumatera. Sistem Elektronika Daya untuk kereta api listrik. Pembangkit Listrik Tenaga Surya.

2.1.2 Visi dan Misi

1. Visi PT Len Industri adalah menjadi perusahaan elektronika kelas dunia.
2. Misi untuk mewujudkan visi tersebut adalah dengan meningkatkan kesejahteraan stakeholder melalui inovasi produk elektronika industri dan prasarana.

2.1.3 Struktur Organisasi

Struktur organisi PT Len Industri digambarkan pada gambar 2.1. sebagai berikut: `



Gambar 2. 1 Struktur Organisasi PT Len Industri

2.2 Landasan Teori

Landasan teori merupakan kajian ilmu dari berbagai sumber seperti buku dan karya ilmiah yang sesuai dengan penelitian yang dilakukan.

2.2.1 Komunikasi Data

Komunikasi data merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer – komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajiakan oleh isyarat digital. Komunikasi data merupakan bagian penting masyarakat informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain. Berikut adalah komponen dari komunikasi data:

1. Pengirim, merupakan piranti yang mengirimkan data
2. Penerima, merupakan piranti yang menerima data
3. Data, merupakan informasi yang akan dipindahkan
4. Media pengiriman, merupakan media atau saluran yang digunakan untuk mengirimkan data
5. Protokol, adalah aturan-aturan yang digunakan untuk mengatur jalannya komunikasi data antara beberapa komputer yang terdapat dalam suatu jaringan. Supaya perangkat yang berlainan itu dapat berkomunikasi, mengirim ataupun menerima maka dibutuhkan suatu pengertian. Maka dengan protokol semua itu bisa terjadi [2].

2.2.2 Keamanan Jaringan

Perkembangan teknologi jaringan komputer menyebabkan terkaitnya satu komputer dengan komputer lainya. Hal ini membuka banyak peluang dalam pengembangan aplikasi komputer tetapi juga mebuat peluang adanya ancaman terhadap pengubahan dan pencurian data. Sebuah aplikasi yang melintasi jaringan publik seperti internet diasumsikan dapat diakses oleh siapapun termasuk orang-orang atau pihak-pihak yang memang berniat untuk mencuri atau mengubah data. Oleh karena itu, untuk melindungi data terhadap akses, pengubahan dan penghalangan yang tidak dilakukan oleh pihak yang berwenang, peranti keamanan data yang melintas di jaringan komputer harus disediakan [3].

2.2.3 Layanan Keamanan Jaringan

Layanan-layanan keamanan jaringan didefinisikan berdasarkan kebutuhan yang harus disediakan untuk memenuhi permintaan terhadap keamanan jaringan. Berikut adalah pembahasan jenis-jenis layanan keamanan jaringan.

1. Otentikasi

Otentikasi yaiu layanan keamanan jaringan yang memungkinkan penerima pesan dapat memastikan keaslian pengirim pesan. Misalnya Andi melakukan komunikasi data dengan Budi melalui jaringan ada dua persoalan yang muncul, yaitu bagaimana Andi bisa yakin kalau yang sedang berkomunikasi denganya adalah benar Budi dan bagaimana Andi bisa yakin bahwa data yang diterimanya memang berasal dari Budi. Layanan otentikasi (*authentication*) memastikan

keduanya. Layanan pertama disebut dengan otentikasi entitas (*entity authentication*) yaitu layanan yang memberikan kepastian terhadap identitas sebuah entitas yang terlibat dalam komunikasi data. Sedangkan layanan kedua disebut dengan otentikasi terhadap keaslian data (*data origin authentication*) yaitu layanan yang memastikan sumber dari sebuah data.

2. Kendali Akses

Kendali akses (*acces control*) adalah layanan keamanan jaringan yang menghalangi penggunaan terhadap tidak terotorisasi terhadap sumber daya. Pada aplikasi jaringan biasanya kebijakan kemampuan (baca, modifikasi, tulis dan eksekusi sebuah data/layanan sistem) ditentukan oleh jenis pengguna. Misalnya sebuah data rekam medik elektronik hanya dapat diakses oleh pasien dan paramedis yang terlibat.

3. Kerahasiaan Data

Kerahasiaan data (*confidentiality*) merupakan layanan keamanan jaringan yang memproteksi data tertranmisi terhadap pengungkapan oleh pihak yang tidak berwenang. Misalnya Andi mengirim data rahasia melaluli internet ke Budi, pada saat yang sama Rani mampu membaca data rahasia yang terkirim itu melalui *router* maka layanan kerahasiaan data memastikan bahwa data rahasia meskipun mampu dibaca oleh Rani.

4. Keutuhan Data

Keutuhan data (*integrity*) adalah layanan keamanan jaringan yang memastikan bahwa data yang diterima oleh penerima adalah benar-benar sama dengan data yang dikirim oleh pengirim. Sebagai contoh Andi ingin mengirim pesan *P* kepada Budi maka layanan keutuhan data memberikan pengetahuan kepada Budi apabila pesan *P* telah berubah. Ada dua jenis layanan keutuhan data yaitu keutuhan data dengan pemulihan dan tanpa pemulihan.

5. Non-Repudiation

Layanan *Non-Repudiation* adalah layanan keamanan jaringan yang menghindari penolakan atas penerimaan/pengiriman data yang telah terkirim. Misalnya Andi mengirim pesan *P* kepada Budi, maka Budi dengan layanan *non-*

repudiation dapat memberi bukti bahwa data terkirim oleh Andi dan sebaliknya Andi dapat membuktikan bahwa pesan telah terkirim ke Budi.

6. Ketersediaan

Layanan ketersediaan (*availability*) adalah layanan sistem yang membuat sumber daya sistem tetap dapat diakses dan digunakan ketika ada permintaan dari pihak yang berwenang. Serangan terhadap sistem seperti *denial of services* membuat sistem tidak dapat diakses oleh pihak yang berwenang.

2.2.4 Serangan Terhadap Keamanan Jaringan

Serangan keamanan jaringan yang dioperasionalkan pada jaringan publik rentan terhadap serangan oleh pihak yang tidak berwenang. Serangan terhadap sistem keamanan jaringan bertujuan untuk mengalahkan tujuan layanan keamanan jaringan . misalnya penyerang ingin mengungkap isi teks asli sehingga ia dapat mengungkap teks sandi lainya. Secara umum serangan pada sistem keamanan jaringan dapat dikategorikan menjadi 2 jenis, yaitu serangan pasif dan serangan aktif.

2.2.4.1 Serangan Pasif

Pada seragnan pasif, penyerang hanya mengumpulkan data yang melintas pada jaringan publik (jaringan yang bisa diakses oleh semua orang). Serangan pasif tidak melakukan modifikasi data yang melintas atau merusak sistem, penyerang hanya bisa membaca (*read only*). Berdasarkan data yang dikumpulkan tadi penyerang melakukan analisis untuk menggagalkan tujuan layanan keamanan jaringan. Karena tidak melakukan modifikasi dan mengganggu sistem, serangan pasif sulit di deteksi, tapi dapat dicegah dengan cara menggunakan sandi pada saat mengirim pesan misalnya. Berikut beberapa jenis serangan pasif:

1. *Snooping*, merupakan kegiatan yang bermaksud untuk mendapatkan data yang sedang dikirimkan melalui jaringan oleh pihak yang tidak berwenang. Contoh kegiatan *snooping* yaitu penyadapan email, *snooping* ini bisa dicegah dengan cara menyandikan pesan menggunakan mekanisme penyandian (*enchiperment*)

2. *Taffic Analysis,* merupakan kegiatan serangan dengan melakukan pemantauan terhadap lalu lintas jaringan untuk kemudian datanya dikumpulkan dan dianalisis untuk mendapatkan informasi dari data – data tersebut.

2.2.4.2 Serangan Aktif

Serangan aktif dapat mengakibatkan perubahan data yang sedang dikirimkan melaui jaringan dan mengganggu sistem. Serangan pasif memungkinkan penyerang untuk mengubah data pada lalu lintas data dijaringan. Jenis – jenis serangan aktif adalah sebagai berikut:

1. *Masquerade,* merupakan serangan aktif yang dilakukan oleh penyerang dengan cara mengambil alih (menirukan) perilaku pengirim atau penerima. Contohnya pada saat Andi ingi membuat kunci bersama dengan Budi, disaat yang bersamaan secara tidak sadar Rani mengambil alih peran Budi sehingga Andi mengirim pesan ke Rani bukan kepada Budi.
2. *Modification,* merupakan serangan dengan cara mengambil alih jalur komunikasi untuk mengubah atau menghapus pesan yang sedang dikirimkan untuk keuntungan penyerang. Misalnya pesan “kirim Rp 100.000, 00 ke nomer rekening Andi” dirubah oleh Rani menjadi “kirim Rp 100.000, 00 ke nomer rekening Rani”.
3. *Replay,* merupakan serangan yang terdiri dari pencatatan secara pasif data dan transmisi ulang untuk menimbullkan efek yang diinginkan oleh penyerang. Misalnya Rani pernah meminta Budi untuk mengirim Rp 100.000, 00 ke Rani, lalu Budi mengirim pesan “kirim Rp 100.000, 00 ke Rani” ke Bank, pada saat pesan dikirimkan Rani mencatat pesan “kirim Rp 100.000, 00 ke Rani” lalu mengirim ulang pesan tersebut ke Bank sehingga Rani mendapat uang dengan jumlah dua kali lebih besar.
4. *Denial of Service,* merupakan jenis serangan yang bertujuan menyebabkan sistem menjadi tidak dapat bekerja sama sekali sehingga tidak bisa memeberikan pelayanan yang seharusnya kepada pengguna. Contohnya adalah penyerang membuat permintaan kosong (*dummy*) sehingga *server* menjadi *overload.*

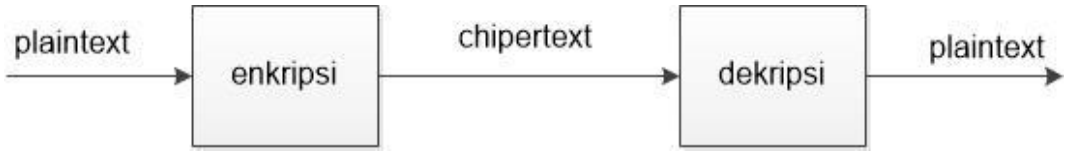
Biasanya sebuah sistem keamanan jaringan dikatakan aman apabila tahan terhadap serangan aktif. Untuk mengembangkan sistem keamanan yang baik, seorang perancang keamanan jaringan harus menganalisis kemungkinan serangan – serangan terhadap layanan keamanan jaringan.

2.2.5 Mekanisme Keamanan Jaringan

Untuk mendapatkan layanan keamanan jaringan seorang perancang keamanan dapat menggunakan mekanisme keamanan jaringan. Berikut adalah beberapa jenis mekanisme keamanan jaringan:

2.2.5.1 Enkripsi

Pesan yang belum disandikan disebut dengan *plaintext* proses yang digunkaan untuk mengubah *plaintext* menjadi *chipertext* disebut enkripsi (*encryption*) atau *enchiperment*. Sedangkan proses untuk mengubah *chipertext* kembali menjadi *plaintext* disebut dengan dekripsi (*decryption*). Gambar 2.2 berikut akan menjelaskan istilah-istilah yang dijelaskan diatas.



Gambar 2. 2 Proses Enkripsi dan Dekripsi

Kunci-kunci yang digunakan untuk proses enkripsi dan dekripsi tidak harus selalu identik, tergantung pada sistem yang digunakan. Secara umum operasi enkripsi dan dekripsi dapat diterangkan secara matematis sebagai berikut:

$EK\ (M) = C$ (Proses Enkripsi)

$DK\ (C) = M$ (Proses Dekripsi)

Pada saat proses enkripsi kita menyandikan pesan *M* dengan suatu kunci *K* lalu dihasilkan pesan *C.* sedangkan pada proses dekripsi, pesan *C* tersebut diuraikan dengan menggunakan kunci *K* sehingga dihasilkan pesan *M* yang sama seperti pesan sebelumnya. Dengan demikian keamanan suatu pesan bergantung pada kunci ataupun kunci-kunci yang digunakan, bukan pada algoritma yang digunakan. Sehingga algoritma-algortima yang digunakan tersebut dapat

dipublikasikan untuk dianalaisis dan produk-produk yang menggunakan algoritma tersebut dapat diproduksi masal.

2.2.5.2 Keutuhan Data

Mekanisme keutuhan data digunakan untuk memastikan keutuhan data pada unit data atau pada suatu aliran (*stream*) dan unit. Cara yang lumrah digunakan adalah dengan menambahkan nilai penguji (*check value*) pada data asli menggunakan *fungsi hash.* Jadi, sebelum data dikirimkan terlebih dahulu dihitung *fungsi hash* dari data tersebut lalu kemudian data dikirimkan bersama *fungsi hash.* Di pihak penerima kemudian data diuji dengan menghitung *fungsi hash* lalu kemudian dibandingkan dengan *fungsi hash* yang dikirim bersamaan dengan dengan data tadi. Apabila hasilnya sama maka penerima pesan bisa menyimpulkan bahwa data tidak berubah.

2.2.5.3 *Digital Signature* (Tanda Tangan Digital)

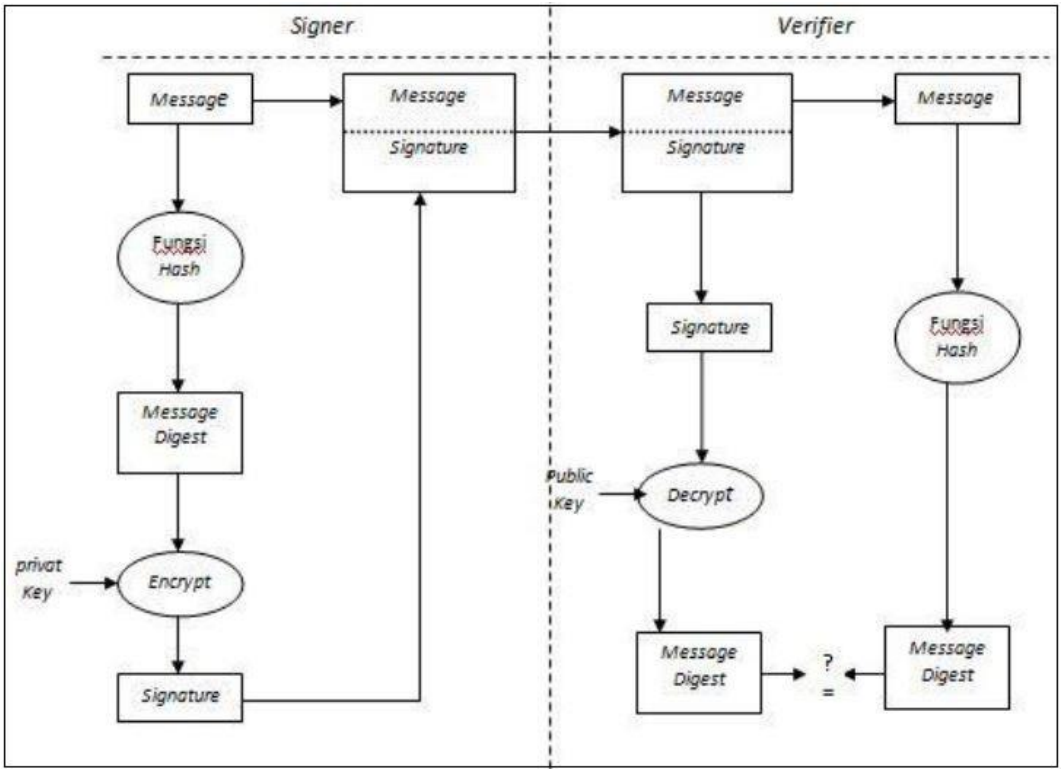
Digital signature merupakan mekanisme keamanan jaringan yang menyediakan cara bagi pengirim pesan untuk "menandatangani" secara elektronik sebuah data dan penerima data dapat memverifikasi "tanda tangan" itu secara elektronik. Layanan keamanan jaringan yang diberikan oleh *digital signature* yaitu *confidentiality* (kerahasiaan data), *integrity* (keutuhan data), *Non-repudiation* (anti penyangkalan).

Digital signature dihasilkan berdasarkan pesan yang ingin ditandatangani dan berubah-ubah sesuai dengan pesan yang akan ditandatangani. *Digital signature* dikirimkan beserta pesan kepada penerima. Mekanisme *digital signature* ini memungkinkan penerima untuk memverifikasi terlebih dahulu apakah informasi yang terkandung pada pesan utuh dan sesuai dengan yang dikirimkan oleh pengirim dan juga dapat menyakinkan bahwa orang yang mengirimkan pesan adalah orang yang berhak.

Dalam kehidupan nyata entitas yang terlibat dalam sebuah komunikasi melalui jarigan cenderung lebih banyak menggunakan *digital signature* daripada enkripsi biasa. Misalnya, seorang nasabah tidak peduli jika setiap orang mengetahui bahwa dia mentransfer uang sebesar seratus juta rupiah ke rekening

temannya, tapi nasabah ini pasti lebih ingin tahu apakah *teller* bank tersebut benar-benar asli, benar-benar ada dan sah.

Pada konsep yang telah diketahui sebelumnya, proses enkripsi biasanya menggunakan kunci publik untuk proses enkripsi dan menggunakan kunci privat untuk proses dekripsinya. Sebaliknya pada konsep *digital signature* sebuah informasi justru dibubuhi "tanda tangan" menggunakan kunci privat sumber, apabila informasi tersebut dapat di verifikasi (dekripsi) dengan kunci publik sumber yang telah dipublikasikan berarti informasi tersebut adalah benar-benar asli dari sumber. Konsep *digital signature* dapat dilihat pada gambar berikut:



Gambar 2. 3 Konsep *Digital Signature*

Cara membuat *digital signature* menurut gambar diatas adalah sebagai berikut:

1. Suatu pesan dibuat nilai *hash*-nya menggunakan fungsi *hash* menghasilkan *message digest.*
2. Kemudian nilai *hash* tadi di enkrip (di tanda tangani) menggunakan kunci privat pihak yang akan mengirim pesan (*signer*).
3. Pesan beserta *digital signature*-nya dikirimkan ke pihak penerima (*verifier*).

Cara memverifikasi *digital signature* menurut gambar diatas adalah sebagai berikut:

1. Pesan dibuat nilai *hash*-nya menggunakan fungsi yang sama dengan fungsi *hash* yang digunakan oleh *signer* (pembuat tanda-tangan) menghasilkan *message digest*.
2. *Digital signature* di dekripsi menggunakan kunci publik *signer* (pengirim pesan) menghasilkan *message digest.*
3. Kemudian *message digest* dari hasil langkah pertama dicocokan dengan *message digest* pada langkah kedua, apabila hasilnya cocok berarti *digital signature* berasal dari orang yang benar.

Pada umumnya tanda *digital signature* memiliki tiga algoritma:

1. Algoritma pembangkit kunci
2. Algoritma *sign* (pemberian tanda tangan)
3. Algoritma verifikasi tanda tangan

2.2.5.4 Authentication Exchange

Merupakan mekanisme yang memberikan cara agar pada saat dua orang yang sedang bertukar data dapat saling meng-otentikasi degan cara bertukar pesan untuk dapat saling membuktikan identitas.

2.2.5.5 Traffic Padding

Merupakan mekanisme keamanan jaringan yang menyediakan cara untuk pencegahan analisis lalu lintas data pada jaringan yaitu dengan menambah data palsu pada lalu lintas data.

2.2.5.6 Routing Control

Routing control menyediakan cara untuk memilih dan secara terus menerus mengubah alur (*route*) pada jaringan komputer antara pengirim dan penerima. Mekanisme ini dapat menghindarkan komunikasi pada jaringan dari penguping.

2.2.5.7 Notarisasi

Mekanisme notarisasi menyediakan cara untuk memilih pihak ketiga yang bisa dipercaya untuk pengendali proses komunikasi antara pengirim dan penerima.

2.2.5.8 Kendali Akses

Kendali akses memungkinkan pengguna untuk mendapat hak akses terhadap sebuah data, misalnya dengan membuat tabel relasi pengguna dan level kemampuan aksesnya.

2.2.6 Kriptografi

Layanan keamanan jaringan diwujudkan dengan menggunakan mekanisme keamanan jaringan. Mekanisme keamanan jaringan pada implementasinya menggunakan teknik-teknik penyandian yaitu kriptografi.

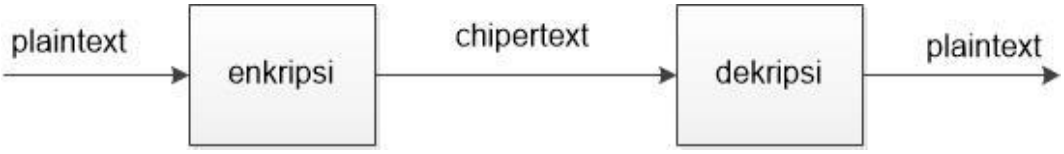
2.2.6.1 Pengertian Kriptografi

Kriptografi berasal dari bahasa yunani, menurut bahasa dibagi menjadi dua *kripto* dan *graphia*, *kripto* berarti *secret* (rahasia) dan *graphia* berarti *writting* (tulisan). Menurut terminologinya kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Keamanan pesan diperoleh dengan menyandikannya menjadi pesan yang tidak memiliki makna. Zaman sekarang ini kerahasiaan informasi menjadi sesuatu yang penting. Informasi yang rahasia perlu disembunyikan agar tidak diketahui oleh pihak yang tidak berhak. Kriptografi atau yang sering dikenal dengan sebutan ilmu pendian data, merupakan suatu bidang ilmu dan seni (*art and science*) yang bertujuan untuk menjaga kerahasiaan suatu pesan yang berupa data dari akses pihak-pihak yang tidak berhak untuk menghindarkan kerugian

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana cara menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang berdasarkan pada teknik matematika yang erat kaitannya dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah bukan hanya penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi.

Pesan asli sebelum dirahasiakan disebut plainteks (*plaintext* artinya teks asli yang dapat dimengerti), sedangkan pesan hasil penyandian disebut dengan *ciphertext* (artinya teks tersandi). Pesan yang telah tersandikan dapat dikembalikan lagi ke pesan aslinya hanya oleh orang yang berhak (orang yang

berhak adalah orang yang mengetahui metode penyandian atau memiliki kunci penyandian). Proses penyandian plainteks menjadi cipherteks disebut enkripsi dan proses mengembalikan cipherteks menjadi plainteks adalah dekripsi [4].



Gambar 2. 4 Konsep Enkripsi

2.2.6.2 Sejarah Kriptografi

Kriptografi mempunyai sejarah panjang dan sangat menarik. Kriptografi sudah digunakan 4000 tahun yang lalu yang diperkenalkan oleh orang-orang Mesir untuk mengirim pesan ke pasukan militer yang berada di lapangan dan supaya pesan tersebut tidak bisa dibaca oleh pihak musuh apabila pasukannya tertangkap maka pesan tersebut disandikan.

Pada zaman Romawi kuno dikisahkan pada suatu saat ketika Julius Caesar ingin mengirimkan satu pesan rahasia kepada seorang Jenderal di medan perang, pesan tersebut harus dikirimkan melalui seorang kurir. Tetapi karena pesan tersebut bersifat rahasia, supaya pesan tersebut terbuka ditengah jalan baik oleh musuh maupun oleh kurirnya sendiri maka Julius Caesar memikirkan cara bagaimana mengatasinya dengan cara mengacak pesan tersebut menjadi suatu pesan yang tidak dapat dipahami oleh siapapun kecuali hanya oleh Jenderalnya saja. Sebelumnya sang Jenderal tersebut telah diberi tahu bagaimana cara membaca pesan yang telah diacak tersebut karena telah mengetahui kuncinya.

Pada perang dunia kedua, negara Jerman menggunakan enigma atau juga disebut dengan mesin rotor yang digunakan *Hitler* untuk mengirim pesan ke tentaranya. Jerman sangat percaya pesan yang dikirim melalui enigma tidak terpecahkan kode-kode enkripsinya. Tapi anggapan itu keliru, setelah bertahun-tahun sekutu dapat memecahkan kode-kode tersebut setelah mempelajarinya. Setelah Jerman mengetahui kode-kode tersebut dapat terpecahkan, maka enigma yang digunakan pada perang dunia kedua, beberapa kali mengalami perubahan.

Enigama yang digunakan Jerman bisa mengenkripsikan satu pesan mempunyai 15 milyaran kemungkinan hasil dekripsinya. Selama bertahun-tahun kritografi menjadi bidang khusus yang dipelajari oleh pihak militer seperti agen keamanan nasional Amerika, Uni Soviet, Inggris, Perancis, Israel, dan negara-negara lainya yang telah membelanjakan milyaran dollar untuk mengamankan komunikasi mereka agar tidak dapat diketahui oleh negara lain tetapi disisi lain mereka juga mempelajari kode-kode rahasia negara lain, dengan adanya persaingan ini maka kriptografi terus berkembang sesuai dengan perkembangan zaman.

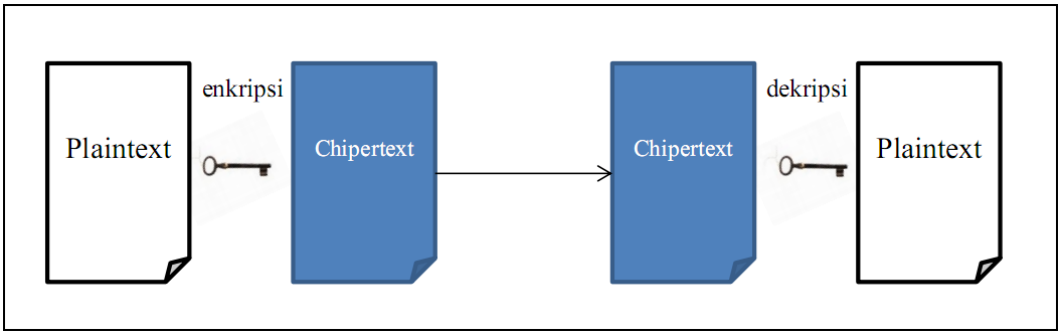
Namun pada 30 tahun terakhir ini, kriptografi tidak hanya dimonopoli oleh pihak militer saja, hal yang sama juga dilakukan oleh individu-individu yang menginginkan pesan dan komunikasi mereka tidak diketahui oleh pihak lain. Apalagi pada zaman sekarang ini persaingan yang begitu tinggi, mereka rela mengeluarkan milyaran dollar hanya untuk menjaga *privacy* mereka.

2.2.6.3 Jenis-Jenis Kriptografi Berdasarkan Kunci

Berdasarkan jenis kuncinya kriptografi dibagi dua yaitu kriptografi kunci simetri dan kriptografi kunci asimetri. Berikut penjelasan dari kriptografi kunci simetri dan kunci asimetri.

2.2.6.3.1 Kriptografi Kunci Simetri

Dalam sistem kriptografi klasik selalu mengasumsikan pihak yang mengenkripsi dan pihak yang mendekripsi memiliki kunci rahasia yang sama misalnya *K*. Sistem kriptografi seperti ini dinamakan kriptografi dengan kunci simetri, dimana kunci rahasia *K* harus dibangkitkan secara rahasia dan didistribusikan ke pengenkripsi dan pendekripsi melalui saluran yang di asumsikan aman. Pada praktiknya mendapatkan saluran aman adalah hal yang sulit apalagi sekarang setiap aplikasi membutuhkan jaringan terbuka seperti internet yang dikategorikan sebagai jaringan tidak aman.



Gambar 2. 5 Penggunaan Kriptografi Kunci Simetris

Kriptografi kunci simetri merupakan jenis kriptografi yang paling umum dipergunakan, kunci yang digunakan untuk enkripsi sama dengan kunci untuk dekripsi. Jadi pembuat pesan dan penerimanya harus memiliki kunci yang sama dan harus dirahasiakan. Kelebihan kriptografi kunci simetri adalah lebih cepat apabila dibandingkan dengan seistem kriptografi lain. Kelemahan dari sistem ini bukan hanya masalah pengiriman *chipertext*-nya melainkan pengirim harus mencari cara yang tepat untuk memberitahukan kunci kepada penerima. Selain itu masalah efisiensi kunci dimana apabila terdapan pengguna sebanyak *n*, maka dibutuhkan kunci sebanyak *n(n-1)/2* sehingga untuk jumlah pengguna yang banyak sistem kriptografi kunci publik kurang efisien. Dua kategori algoritma yang termasuk pada sistem kriptografi kunci simetri ini adalah algoritma *block chiper* dan *stream chiper*.

1. Block Chiper

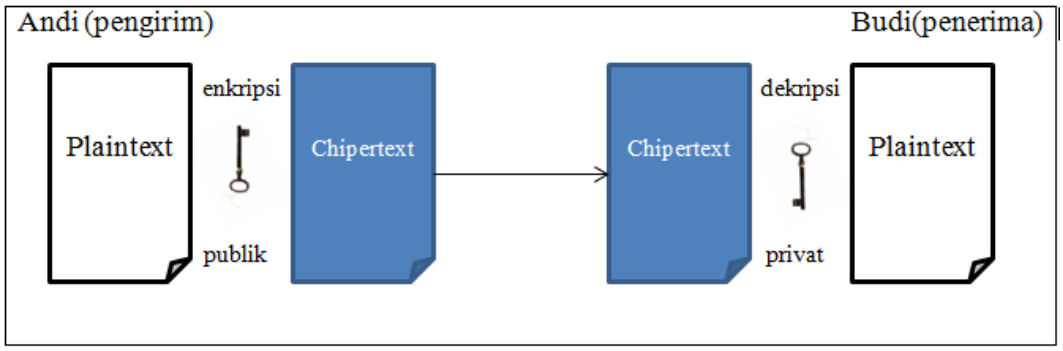
Algoritma *block chiper* merupakan algoritma yang inputan dan keluarannya berupa satu blok dan satu bloknya terdiri dari banyak bit, misalnya 1 blok terdiri dari 64 bit atau 128 bit. Contoh algoritma *block chiper* adalah DES (*Data Encryption Standard*), AES (*Advanced Encryption Standard*), dan IDEA (*International Data Encryption Algoritm*).

2. Stream Chiper

Sandi *stream* yang beroperasi pada data stream sehingga operasi penyandian dilakukan per satu bit atau per satu *byte* pada satu waktu. *Stream chiper* berasal dari hasil XOR antara setiap bit *plaintext* dengan setiap bit kuncinya. Contohnya adalah algoritma OTP (*One Time Pad*), A5, dan RC4.

2.2.6.3.2 Kriptografi Kunci Asimetri (Kunci Publik)

Kebutuhan akan salauran yang aman untuk mendapatkan kunci rahasia *K* menjadi kelemahan utama sistem kriptografi simetri. Sistem kriptografi asimetri (kunci publik) mengatasi asumsi ini, dimana tidak dibutuhkan saluran aman untuk distribusi kunci. Karena sistem kriptografi kunci publik memiliki kunci untuk enkripsi *Ke* yang berbeda dengan kunci untuk dekripsi yaitu *Kd*. Kunci untuk enkripsi *Ke* disebut juga sebagai kunci publik dan berifat tidak rahasia sehingga dapat didistribusikan melalui saluran yang tidak aman sekalipun. Sedangkan kunci untuk dekripsi *Kd* disebut juga dengan kunci *private* dan bersifat rahasia dan harus dijaga kerahasiaannya oleh pemegang kunci.



Gambar 2. 6 Penggunaan Kriptografi Kunci Asimetris

Proses enkripsi dan dekripsi sistem kriptografi kunci publik di ilustrsikan oleh gambar. Pada gambar Andi dan Budi sama-sama menggunakan sistem kriptografi kunci publik. Andi berperan sebagai pengenkripsi sedangkan Budi berperan sebagai orang yang mendekripsi pesan dari Andi. Langkah-langkah sistem kriptografi kunci publik adalah sebagai berikut:

1. Budi membangkitkan sepasang kunci yaitu kunci privat dan kunci publik dengan memanggil fungsi PembangkitKunci:

$$(K_{publik,}\, K_{privat}) \leftarrow \text{PembangkitKunci}() \qquad (2.1)$$

Budi mempublikasikan kunci publik, namun tetap menjaga kerahasiaan kunci privat-nya.

2. Andi mengenkripsi sebuah teks asli (*M*) dengan kunci publik Budi (K_{publik}) menghasilkan sebuah *chiper text* dengan memanggil fungsi Enkripsi:

$$ChiperText \leftarrow Enkripsi\ (K_{publik}, M) \qquad (2.2)$$

Andi mengirimkan *chipertext* kepada Budi melalui saluran yang tidak aman.

3. Kemudian Budi mendekripsi teks sandi (*chipertext*) menggunakan kunci privatnya sendiri (K_{privat}) untuk mendapatkan hasil berupa teks asli (*plaintext*) dengan memanggil fungsi Dekripsi:

$$M \leftarrow Dekripsi(K_{privat}, ChiperText) \quad (2.3)$$

Budi mendapatkan M yang sama dengan M yang sebelum di enkripsi oleh Andi jika Andi menggunakan K_{publik} Budi.

Sistem kriptografi kunci publik (asimetri) yang paling banyak digunakan diantaranya adalah RSA (Rivest Shamir Adleman), DH (Deffie Hellman), Quantum, ECC (*Elliptic Curve Cryptography*), DSA (*Digital Signature Algorithm*). Berikut penjelasan dari masing-masing algoritma:

1. RSA (Rivest, Shamir, Adleman)

RSA dirumuskan oleh tiga orang bernama Ron Rivest, Adi Shamir, dan Leonard Adleman yang kemudian disingkat menjadi RSA pada tahun 1977. Meskipun pada tahun 1997 badan sandi Inggris mempublikasikan bahwa Clifford Cock telah lebih dahulu merumuskan sistem kriptografi yang sama dengan RSA tiga tahun lebih dulu daripada Rivest, Shamir, dan Adleman. Tidak pernah ada bukti bahwa algoritma RSA aman atau tidak, hanya karena sulitnya memfaktorkan bilangan yang sangat besar maka RSA dianggap aman. RSA merupakan algoritma asimetri yang berarti memiliki dua kunci, yaitu kunci publik dan kunci privat. Terdapat 3 algoritma pada sistem kriptografi RSA, yaitu algoritma pemabngkitan kunci, algoritma enkripsi, dan algoritma dekripsi.

2. DH (Deffie Hellman)

Algoritma deffie-Hellman dianggap merupakan algoritma kunci publik yang pertama kali ditemukan yaitu sekitar tahun 1976, meskipun NSA mengaku telah menemukan algoritma asimetrik jauh-jauh hari sebelumnya. Keamanan algoritma Deffie-Hellman diperoleh dari sulitnya menghitung logaritma diskrit pada bilangan yang sangat besar. Algoritma ini biasanya digunakan pada proses pertukaran kunci pada sistem kriptografi simetri.

3. Quantum

Kriptografi Quantum ditemukan oleh Bennett dan Brassard pada tahun 1984. Kriptografi jenis ini tidak mengandalkan matematika untuk keamanannya melainkan bersdasarkan pada sifat fisika cahaya. Dalam fisika dikenal istilah foton yang merupakan partikel terkecil dari cahaya, foton ini tidak bisa dibagi lagi menjadi bagian yang lebih kecil. Sama halnya dengan Deffie-Hellman algoritma Quantum ini digunakan untuk pertukaran kunci pada sistem kriptografi simetri. Jadi tidak bisa digunakan untuk enkripsi secara langsung pada data.

4. ECC (*Elliptic Curve Algorithm*)

Sistem kriptografi RSA dan ElGamal merupakan sistem kriptografi yang paling sering digunakan namun memiliki kelemahan yaitu membutuhkan ukuran kunci yang sangat besar (minimal 1024 bit). Sistem kriptografi berbasis kurva *elliptic* memberikan alternatif untuk mewujudkan sistem kriptografi asimetrik dengan panjang kunci yang lebih kecil. Kurva eliptik merupakan sehimpunan solusi yang memenuhi persamaan kubik yang melibatkan 2 variabel.

5. DSA (*Digital Signature Algorithm*)

Pada bulan Agustus tahun 1991, *National Institute of Standards and Technology* (NIST) yang merupakan lembaga standarisasi Amerika Serikat mengusulkan *Digital Signature Algorithm* (DSA). Dan baru pada tahun 1994 DSA dipublikasikan menjadi standar tanda tangan digital (DSS), jadi DSA merupakan algoritma sedangkan DSS (*Digital Signature Standard*) merupakan standar. DSA berbasis pada skema *digital signature* ElGamal dan Schnorr dan menggunakan fungsi hash SHA-1. DSA tidak bisa digunakan untuk enkripsi atau distribusi kunci tapi hanya untuk pembuatan tanda tangan digital [5].

2.2.6.4 Protokol Kriptografi

Protokol merupakan aturan yang berisi rangkaian langkah-langkah, yang melibatkan dua atau lebih orang, yang dibuat untuk menyelesaikan suatu kegiatan. Sedangkan protokol kriptografi adalah protokol yang menggunakan kriptografi. Orang yang berpartisipasi dalam protokol kriptografi memerlukan protokol tersebut misalnya untuk:

1. Berbagai komponen rahasia untuk menghitung sebuah nilai

2. Membangkitkan rangkaian bilangan acak
3. Menyakinkan identitas orang lain (otentikasi)

Protokol kriptografi dibangun dengan melibatkan beberapa algoritma kriptografi. Sebagian besar protokol kriptografi dirancang untuk dipakai oleh kelompok yang terdiri dari 2 orang pemakai, tetapi ada juga beberapa protokol yang dirancang untuk dipakai oleh kelompok yang terdiri dari lebih dari dua orang pemakai [6].

2.2.6.4.1 Protokol Komunikasi dengan Sistem Kriptografi Kunci Publik

Berikut atrannya:

1. Budi dan Andi menyepakati algoritma kriptografi kunci publik yang akan digunakan.
2. Budi mengirimi Andi kunci publiknya (kunci publik Budi).
3. Andi mengenkripsi pesannya dengan kunci publik Budi kemudian mengirimkannya kepada Budi.
4. Budi mendekripsi pesan dari Andi menggunakan kunci privat miliknya (kunci privat Budi).

Pada umumnya, pengguna di jaringan menyepakati algoritma kriptografi kunci publik yang digunakan. Setiap pengguna jaringan mempunyai kunci publik dan kunci privat, yang dalam hal ini kunci publik dipublikasikan melalui basisdata yang dapat diakses bersama. Dengan demikian, protokol kriptografi kunci publik menjadi lebih sederhana sebagai berikut:

1 Andi mengambil kunci publik Budi dari basisdata kunci-publik
2 Andi mengenkripsi pesannya dengan kunci publik Budi kemudian mengirimkanya kepada Budi
3 Budi mendekripsi pesan dari Andi dengan kunci privat miliknya (kunci privat Budi).

2.2.6.4.2 Protokol Digital Signature

Berikut aturan menandatangani dokumen dengan sistem kriptografi kunci publik dan fungsi *hash* satu arah:

1. Andi meringkas dokumenya menjadi *message digest* dengan fungsi *hash* satu arah.

2. Andi membuat *digital signature* pada *message digest* dengan kunci privatnya.
3. Andi mengirim dokumen yang telah di enkripsi (*digital signature*) kepada Budi.
4. Budi meringkas dokumen dari Andi menjadi *message digest* dengan fungsi *hash* yang sama dengan yang digunakan Andi. Kemudian Budi mendekripsi (verifikasi *digital signature*) dokumen dengan kunci publik Andi. Jika hasil dekripsinya sama dengan *message digest* yang dihasilkan, maka tanda tangan digital tersebut benar dari Andi.

2.2.6.4.3 Protokol Untuk Digital Signature dengan Enkripsi

Protkol ini dapat dianalogikan seperti pengiriman surat yang menggunakan amplop tertutup. Tanda tangan pada surat memberikan bukti kepemilikan, hal ini sama dengan fungsi tanda tangan pada dokumen elektronis. Sedangkan amplop memberikan perlindungan keamanan (*privacy*), hal ini sama dengan fungsi enkripsi pada dokumen. berikut aturannya:

1. Andi menandatangani dokumen atau pesan (*M*) dengan menggunakan kunci privat (*A*).

$$S_A(M) \tag{2.4}$$

2. Andi mengenkripsi dokumen yang sudah ditandangani dengan kunci publik Budi (*B*) dan mengirimkannya kepada Budi

$$E_B(S_A(M)) \tag{2.5}$$

3. Budi mendekripsi chiperteks yang diterima dengan kunci privatnya

$$D_B\left(E_B\left(S_A(M)\right)\right) = S_A(M) \tag{2.6}$$

4. Budi melakukan verifikasi dengan mendekripsi hasil pada langkah 3 dengan menggunakan kunci publik Andi dan sekaligus mendapatkan kembali dokumen yang belum dienkripsi.

$$V_A\left(S_A(M)\right) = M \tag{2.7}$$

2.2.6.5 Manajemen Kunci

Aspek penting lain yang perlu diperhatikan dalam sistem kriptografi adalah manajemen kunci. Tujuan dari manajemen kunci adalah menyediakan

prosedur yang aman untuk menangani kunci kriptografi baik yang simetris maupun asimetris. Walaupun algoritma enkripsi sangat sulit secara perhitungan untuk dipecahkan, namun sistem akan mudah diketahui apabila kunci tidak dilindungi kerahasiaanya.

2.2.6.5.1 *Secret Key*

Untuk manajemen kunci yang digunakan oleh *single user* dalam melindungi data, pendekatan yang paling mudah adalah mencegah penyimpanan kunci di dalam sistem. Para pemakai bertanggung jawab untuk mengelola kuncinya dan menggunakannya pada saat proses enkripsi maupun dekripsi.

Pada sistem konvensional, pemakai mendaftarkan kunci privat pada sistem untuk membangun saluran yang aman. Kunci tersebut harus dilindungi, cara yang paling mudah yaitu dengan menyimpanya dalam suatu *file* yang telah dienkripsi dibawah kontrol *master key* sistem. Tidak seperti pada *password,* kunci tidak dapat dilindungi dengan sistem *one-way-function* karena akan menjadi sangat sulit untuk mendapatkan kembali kunci-kunci tersebut.

2.2.6.5.2 *Public Key* (Kunci Publik)

Pada sistem kunci publik, seorang pemakai tidak perlu mendaftarkan kunci privat pada sistem untuk membangun saluran yang aman. Hal ini dikarenakan sifat kunci privat yang tidak boleh diketahui oleh orang lain selain pemakai yang bersangkutan. Kunci privat ini harus dijaga kerahasaianya dari pihak yang tidak berhak untuk mencegah terjadinya pemalsuan.

Kunci publik dapat didistribusikan baik ke luar dari sistem maupun di dalam sistem yaitu pada *public key directory* yang dikelola oleh *system directory manager.* Kunci publik tidak perlu dijaga keamanannya, karena kerahasiaan kunci publik tidak diperlukan untuk *secure communication*

2.2.6.6 Sistem Kriptografi Berbasis Kurva Eliptik

Sistem kriptografi berbasis kurva eliptik memberikan alternatif untuk mewujudkan sistem kriptografi asimetrik dengan ukuran kunci yang lebih kecil daripada sistem kriptografi RSA dan ElGamal. Seperti sudah diketahui bahwa sistem kriptografi RSA dan ElGamal merupakan sistem kriptografi yang populer dan banyak digunakan, namun kedua sistem kriptografi tersebut memiliki

kelemahan yaitu membutuhkan ukuran kunci yang sangat besar (minimal 1024 bit). Berdasarkan literatur sistem kriptografi ElGamal dengan kurva eliptik mencapai tingkat keamanan yang sama dengan sistem kriptografi RSA dan ElGamal dengan panjang kunci yang lebih pendek. Misalnya sistem kriptografi ElGamal berbasis kurva eliptic dengan panjang kunci 160 bit mencapai tingkat keamanan yang sama dengan sistem ElGamal dengan panjang Kunci 1024 bit. Berikut hasil penelitian internasional tentang perbandingan sistem kriptografi *symetric*, RSA, dan sistem kriptografi berbasis kurva eliptik.

Tabel 2. 1 Perbandingan Jumlah bit Algoritma RSA dan ECDSA dengan Tingkat Keamanan Sama

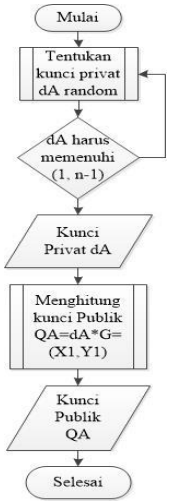
Symetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521
Table1: NIST Recommended Key Sizes		

2.2.6.6.1 *Digital Signature* Menggunakan ECDSA

ECDSA (*Elliptic Curve Digital Signature Algorithm*) merupakan salah satu skema *digital signature* yang berbasis kurva eliptik. ECDSA memberikan layanan *authentication, data integrity, dan non-repudiation.* Pembuat tanda tangan menggunakan kunci privat untuk membuat *digital signature,* sedangkan kunci publik yang merupakan pasangan dari kunci privat digunakan untuk memverifikasi *digital signature.* Satu orang user memiliki satu pasang kunci yaitu kunci privat dan kunci publik, kunci publlik boleh disebarkan kepada pihak lain sedangkan kunci privat harus dijaga kerahasianya. Pada skema ECDSA pihak yang akan membuat *digital signature* memiliki parameter domain kurva eliptik berupa $D=\{q,FR,a,b,G,n,h\}$ dan pasangan kunci rahasia dA dan kunci publik QA.

Sedangkan pihak penerima yang akan melakukan verifikasi terhadap tanda tangan, memiliki salinan dokumen D yang otentik dan kunci *QA.*

1. Prosedur Pembangkit Kunci
   a. Memilih sebuah bilangan bulat random *dA,* yang nilainya (1, n-1)
   b. Menghitung *QA*=*dA* x *G*=(x1,y1) (2.1)
   c. Kunci rahasia = *dA,* dan kunci publik = *QA*



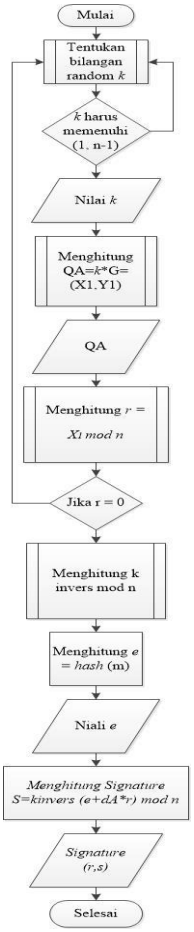
Gambar 2. 7 Flowcart Pembuatan Kunci ECDSA

2. Prosedur

   Pembuatan *Digital Signature* (*signing*)
   a. Memilih sebuah bilangan bulat random *k,* yang nilainya diantara (1, n-1)
   b. Menghitung $QA = k\ X\ G = (X_1, Y_1)$ (2.2)
   c. Menghitung $r = X_1 mod\ n$ (2.3)

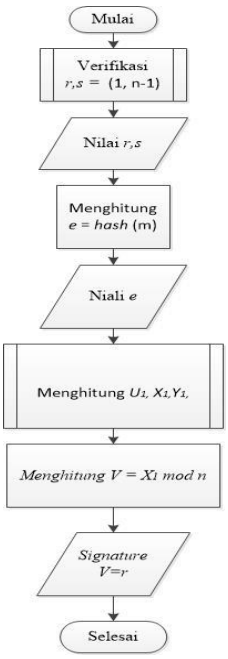
      jika *r=0,* maka kembali ke langkah 1.
   d. Menghitung $k^{-1} mod\ n$ (2.4)
   e. Menghitung *e=hash* (m) (2.5)
   f. Menghitung $s = k^{-1}$ {e+dA x r} mod n (2.6)

Hasilnya berupa tanda tangan untuk pesan m adalah (r,s)



Gambar 2. 8 Flowcart Pembuatan Signature

3. Prosedur Verifikasi *Digital Signature*
   a. Verifikasi bahwa r dan s adalah bilangan bulat antara (1, n-1)
   b. Menghitung *e = hash (m)*
   c. Menghitung $w = s^{-1}\ mod\ n$ (2.7)
   d. Menghitung $u_1 = ew\ mod\ n\ dan\ u_2 = rw\ mod\ n$ (2.8)
   e. Menghitung $u_1\ X\ G +\ u_2\ X\ QA =\ X_1, Y_1$ (2.9)
   f. Menghitung $V =\ X_1 mod\ n$ (2.10)
   g. Menerima tanda tangan jika dan hanya jika V =r



Gambar 2. 9 Flowcart Verifikasi Signature ECDSA

Keterangan:

d_A= kunci rahasia

Q_A=kunci publik

p : Orde dari finite field (bilangan prima)

a,b: koefisien persamaan kurva eliptik $y^2 = x^3 + ax + b$

G: basic point, yaitu elemen pembangunan grup eliptik $E_p(a,b) atas\ F_p$

r : Orde dari titik titik dalam kurva eliptik (bilangan prima).

S: Seed yang digunakan untuk membangkitkan bilangan random pada saat penentuan parameter parameter kurva.

c : output dari fungsi hash SHA -1 pada saat penentuan parameter kurva eliptik secara acak

a : koefisien a

b : koefisien b untuk persamaan kurva

n: order basic point, yaitu bilangan bulat positip terkecil n.G=O

h: kofaktor, h= #E/n, dengan #E adalah banyaknya titik dalam grup eliptik

Gx : Koordinat x dari titik basis dalam kurva

Gy : Koordinat x dari titik basis dalam kurva

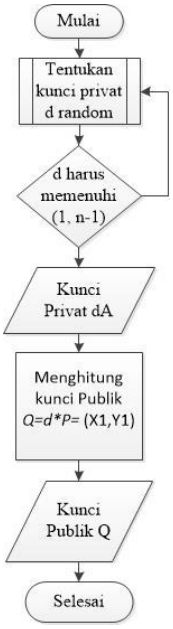
2.2.6.6.2 Enkripsi Menggunakan ElGamal Berbasis Kurva Eliptik

Berdasarkan literatur sistem kriptografi ElGamal dengan kurva eliptik mencapai tingkat keamanan yang sama dengan sistem kriptografi RSA dan ElGamal dengan panjang kunci yang lebih pendek. Berdasarkan grup siklik yang dibentuk oleh pembangkit α untuk kurva eliptik pada GF(p) atau GF (2^m) persoalan logaritma diskrit dapat juga diterapkan pada kurva eliptik. Permasalahan logaritma diskrit pada kurva eliptik dapat diformulasikan sebagai berikut: temukan *k* sehingga *k* memenuhi $k\alpha=\beta$ dengan $\alpha,\beta \in E$ dan $k \in F_p$.

Terdapat tiga algoritma yang digunakan pada proses enkripsi menggunakan sistem kriptografi ElGamal berbasis kurva eliptik. Yaitu algoritma pembangkit kunci, algoritma enkripsi, dan algoritma dekripsi.

1. Algoritma pembangkit kunci.

Algoritma pembangkit kunci mengembalikan kunci publik dan kunci privat. Kunci publik terddiri dari sebuah kurva eliptik yang dipilih $E_p(a,b)$, generator grup kurva eliptik titik α dan titik $\beta = d\alpha$. Sedangkan kunci privat ditetapkan sebagai integer *d.*



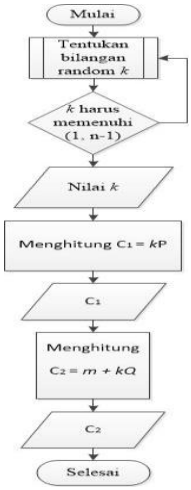
Gambar 2. 10 Flowcart Pembuatan Kunci ElGamal Berbasis Kurva Eliptik

2. Algoritma Enkripsi

Teks asli (P) sebagai masukan algortima enkripsi sistem kriptografi ElGamal berbasis kurva eliptik adalah sebuah titik pada kurva eliptik dan kunci publik $K_{publik} = (E(a, b), \alpha, \beta)$. Pengenkripsi memilih secara acak integer r dan kemudian menghitung teks sandi sebagai berikut:

$$\boldsymbol{C_1 = rX\alpha} \quad (2.11)$$

$$\boldsymbol{C_1 = P + rX\beta} \quad (2.12)$$



Gambar 2. 11 Flowcart Proses Enkripsi ElGamal Berbasis Kurva Eliptik

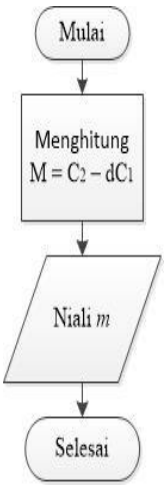
3. Algoritma Dekripsi

Setelah mendapatkan C_1 dan C_2 dapat memulihkan teks asli P dengan menggunakan kunci privat $K_{privat} = d$ dan menghitung:

$$P = C_2 - (dXC_1) \quad (2.13)$$

Bukti bahwa P yang didapatkan adalah sama dengan P oleh pengenkripsi adalah:

$$P = C_2 - (d\ X\ C_1) = P + r\ X\ \beta - (d\ X\ r\ X\ \alpha) = P + r.d\ X\ \alpha = P \quad (2.14)$$



Gambar 2. 12 Flowcart Proses Dekripsi Pesan

2.2.7 Jaringan Komputer

Menurut definisi, yang dimaksud jaringan komputer adalah skumpulan komputer (dan perangkat lain seperti *printer*, *hub*, dan sebagainya) yang saling terhubung satu sama lain melalui media perantara. Media perantara ini bisa berupa kabel ataupun tanpa kabel (nirkabel). Untuk memudahkan memahami jaringan komputer, jaringan komputer dibagi berdasarkan beberapa klasifikasi, diantaranya berdasarkan skala/area [7].

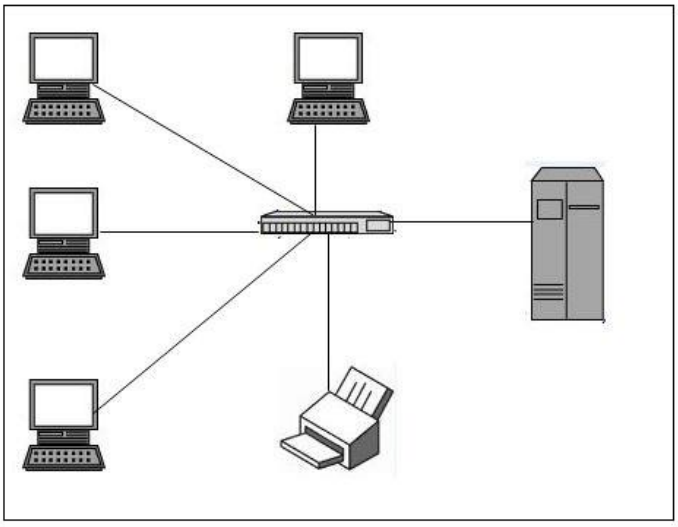
Berikut pembagian jaringan komputer berdasarkan skala/area :

a. LAN (*Local Area Network*)
b. MAN (*Metropolitan Area Network*)
c. WAN (*Wide Area Network*)
d. Internet

2.2.7.1 Jaringan LAN

Local Area Network adalah jaringan lokal yang dibuat pada area tertutup. Misalkan dalam suatu gedung atau dalam suatu ruangan. Kadangkala jaringan lokal disebut juga jaringan privat. LAN biasanya digunakan untuk jaringan kecil yang menggunakan resource bersama-sama, seperti penggunaan printer bersama, dan penggunaan media penyimpaan bersama.

2.2.7.2 Perangkat Jaringan



Gambar 2. 13 Jaringan LAN (*Local Area Network*)

Perangkat jaringan adalah sejumlah perangkat yang digunakan dalam jaringan sebagai pemecah jaringan (contoh: hub, bridge, switch), mengatur perutingan jaringan (contoh: router), penguat jaringan (contoh: repeater), pengkonfersi data jaringan (contoh: modem), interface end user dengan jaringan (NIC & wireless adapter).

1. *Hub*

Hub merupakan perangkat yang dapat menggandakan *frame* data yang berasal dari salah satu komputer ke semua *port* yang terdapat pada *hub* tersebut. Sehingga semua komputer yang terhubung dengan *port hub* akan menerima data juga.

2. *Switch*

Switch bekerja pada lapisan Data Link seperti halnya Bridge. Cara kerja switch sebetulnya mirip dengan Bridge, tetapi switch memiliki sejumlah port sehingga switch sering disebut juga multiport bridge.

3. *Reapeter*

Repeater merupakan contoh dari active hub. Repeater merupakan perangkat yang dapat menerima sinyal, kemudian memperkuat dan mengirim

kembali sinyal tersebut ke tempat lain. Sehingga sinyal dapat menjangkau area yang lebih jauh.

4. *Bridges*

Bridge merupakan peralatan yang dapat menghubungkan beberapa segmen dalam sebuah jaringan. Berbeda dengan hub, bridge dapat mengenalai MAC Address tujuan.

5. *Router*

Router adalah peralatan jaringan yang dapat menghubungkan satu jaringan dengan jaringan yang lain. Sepintas lalu, router mirip dengan bridge, namun router lebih "cerdas" dibandingkan dengan bridge. Router bekerja dengan routing table yang disimpan di memorinya untuk membuat keputusan kemana dan bagaimana paket dikirimkan. Router dapat memutuskan route terbaik yang akan ditempuh oleh paket data. Router akan memutuskan media fisik jaringan yang "disukai" dan "tidak disukai". Protokol routing dapat mengantisipasi berbagai kondisi yang tidak dimiliki oleh peralatan bridge. Router bekerja pada layer network.

6. *NIC*

NIC merupakan peralatan jaringan yang langsung berhubungan dengan komputer dan didesain agar komputer-komputer dalam jaringan dapat saling berkomunikasi. NIC juga menyediakan akses ke media fisik jaringan. NIC bekerja pada layer fisik OSI.

7. *Modem*

Modem berasal dari singkatan MOdulator DEModulator. Modulator merupakan bagian yang mengubah sinyal informasi kedalam sinyal pembawa (Carrier) dan siap untuk dikirimkan, sedangkan Demodulator adalah bagian yang memisahkan sinyal informasi (yang berisi data atau pesan) dari sinyal pembawa (carrier) yang diterima sehingga informasi tersebut dapat diterima dengan baik. Modem merupakan penggabungan kedua-duanya, artinya modem adalah alat komunikasi dua arah.

8. *Wireless Adapter*

Wireless Adapter pada prinsipnya mirip dengan NIC. Hanya saja wireless adapater merupakan interface end user ke jaringan wireless. Pada laptop-laptop jenis baru, wireless adapter sudah terintegrasi di dalamnya, dan biasa disebut PCMCIA. Untuk laptop dan PC yang belum tersedia, dapat digunakan wireless external.

2.2.7.3 Media Transmisi

Media transmisi merupakan media yang digunakan untuk mentransmisikan data dari satu komputer ke komputer lain atau dari server ke client di dalam jaringan. Media transmisi ini ada yang menggunakan kabel dan ada yang tidak (nirkabel). Beberapa media dalam jaringan komputer adalah sebagai berikut:

2.2.7.3.1 Kabel

Dalam jaringan komputer, kabel merupakan media penghubung utama. Berikut macam-macam kabel yang sering digunakan pada jaringan LAN.

1. Kabel *Coaxial*

Kabel *coaxial* adalah kabel yang memiliki satu *copper conductor* pada bagian tengahnya. Sebuah lapisan plastik menutupi diantar konduktor dan lapisan pengaman serat besi. Lapisan serat besi tersebut membantu menutupi konduktor dari gangguan seperti dari arus listrik, lalu lintas kendaraan atau mesin dan komputer.

Kelemahan kabel *coaxial* ini yaitu, sulit dalam konfigurasi dan tidak tahan terhadap sinyal-sinyal tertentu seperti sinyal radio. Kelebihannya yaitu dapat memiliki ukuran yang lebih panjang dalam satu jaringan dibandingkan dengan kabel UTP. Ada dua jenis kabel *coaxial* yaitu *thick-coaxial* dan *thin-coaxial.*

2. Kabel UTP (*Unshielded Twisted Pair*)

Kabel UTP adalah jenis kabel yang paling populer dan umumnya merupakan pilihan terbaik untuk jaringan sederhana. Terdapat empat macam kabel didalamnya dan konektor yang digunakan untuk kabel UTP ini adalah konektor RJ-45.

3. Kabel STP (*Shielded Twisted Pair*)

Kabel STP biasa digunakan pada topologi *Token Ring*, kabel ini sangat sensistif terhadap sinyal radio dan arus listrik. Biasanya kabel jenis ini digunakan pada jaringan yang kurang pengaruh listriknya.

4. Kabel *Fibre Optic*

Kabel jenis ini mempunyai kemampuan mentransmisikan sinyal pada jarak yang relatif lebih jauh dibandingkan dengan kabel *coaxial* atau kabel UTP. Dalam hal kecepatan pun kabel ini sangat baik bila dibandingkan dengan dua kabel diatas.

2.2.7.3.2 Jaringan Tanpa Kabel (Wireless)

Wireless network adalah jaringan yang tidak menggunakan kabel melainkan menggunakan gelombang radio sebagai media penghantarnya. Frequensi radio yang digunakan yang digunakan untuk jaringan yaitu 2.4 GHz dan 5.8 GHz.

2.2.7.4 Topologi Jaringan

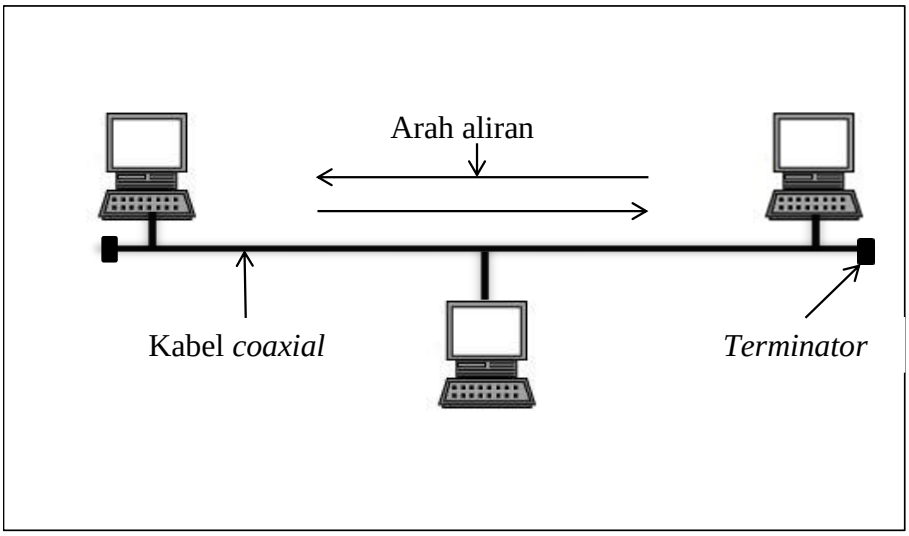
Topologi jaringan adalah suatu aturan/*rules* bagaimana menghubungkan komputer (*node*) satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan, seperti: *server, workstation, hub/switch*, dan pengabelanya (media transimisi data). Ketika kita memutuskan untuk memilih suatu topologi maka kita perlu mengikuti beberapa spesifikasi tertentu.

Ada dua macam jenis topologi, yaitu *physical topology* (topologi fisik) dan *logical topology* (topologi logika). Topologi fisik adalah bentuk jaringan seperti bagaimana memilih perangkat jaringan dan melakukan instalasi perangkat jaringan. Sedangkan topologi logika adalah bagaimana data mengalir di dalam topologi fisik. Ada beberapa macam topologi jaringan berikut penjelasannya:

2.2.7.4.1 Topologi Bus

Topologi *bus* sering disebut *daisy chain* atau *ethernet bus topologies*. Sebutan terakhir diberikan karena perangkat jaringan yang digunakan pada topologi ini adalah NIC (*Network Interface Card*) bernama *ethernet.* Topologi *bus* dapat dikenali dari penggunaan sebuah kabel *backbone* (kabel utama) yang

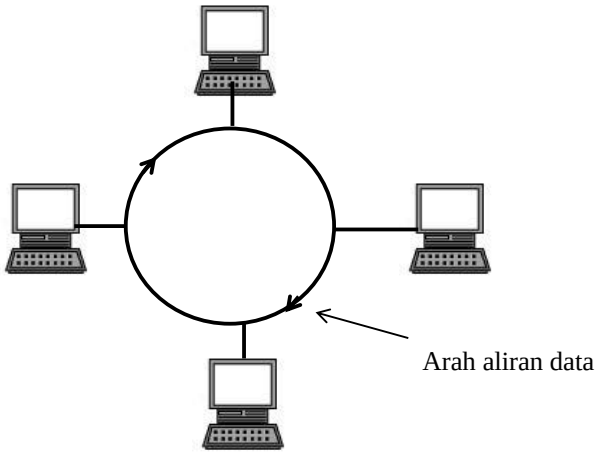
mengghubungkan semua peralatan jaringan. Apabila kabel utama terputus akan menyebabkan jaringan mati total, karena kabel *backbone* (kabel utama) merupakan satu-satunya jalan bagi lalu lintas data. Artinya, apabila salah satu sambungan terputus maka seluruh sambungan akan ikut terputus juga.



Gambar 2. 14 Topologi *Bus*

2.2.7.4.2 Topologi Ring

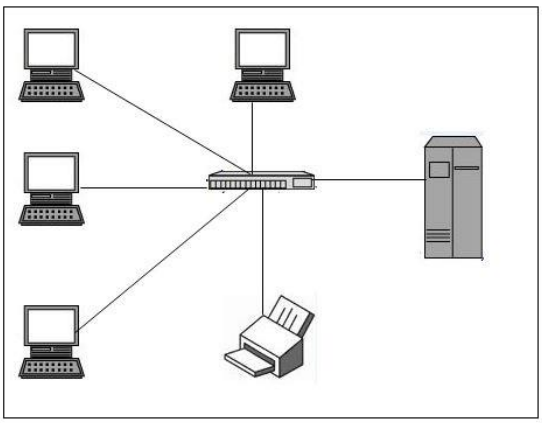
Topologi *ring* menggunakan kabel *backbone* (kabel utama) yang berbentuk cincin. Setiap komputer terhubung dengan kabel *backbone,* pada ujung kabel akan dihubungkan kembali dengan komputer pertama.



Gambar 2. 15 Topologi *Ring*

2.2.7.4.3 Topologi Star

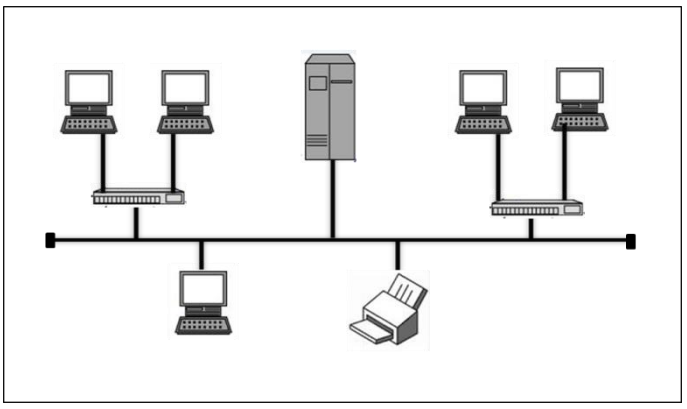
Topologi *star* menggunakan *hub* sebagai sebuah sentral yang digunakan untuk menghubungkan semua *node.* Setiap *node* menggunakan sebuah kabel UTP atau STP yang dihubungkan dari *ethernet card* ke *hub.* Topologi ini banyak digunakan untuk jaringan rumahan, perkantoran, sekolah dan lain-lain.



Gambar 2. 16 Topologi *Star*

2.2.7.4.4 Topologi Tree

Topologi *tree* disebut juga sebagai topologi *star-bus*, topologi ini merupakan penggabungan antara topologi *star* dan *bus*. Topologi *tree* digunakan untuk menghubungkan beberapa LAN dengan LAN lain. Media penghubung antar LAN yaitu menggunakan *hub,* masing-masing *hub* merupakan akar dari masing-masing pohon. Topologi ini dapat mengatasi permasalahan pada topologi *bus* yaitu masalah *broadcast traffic*, dan permasalahan pada topologi *star* yaitu keterbatasan kapasitas *port hub*.



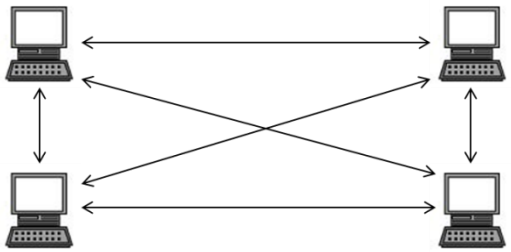
Gambar 2. 17 Topologi *Tree*

2.2.7.5 Model Hubungan Pada LAN

Model hubungan pada jaringan *local area network* merupakan cara menghubungkan antara satu komputer dengan komputer lainnya sehingga bisa saling mengakses sumber daya yang terdapat dalam komputer tersebut. Berikut merupakan model hubungan pada jaringan komputer.

2.2.7.5.1 *Peer to Peer*

Peer to peer merupakan jaringan komputer dimana setiap komputer bisa menjadi *server* sekaligus *client.* Setiap komputer dapat menerima dan memberikan akses ke komputer lain. Didalam model *peer to peer* ini seluruh komputer sama untuk memakai sumber daya yang tersedia dalam jaringan, biasanya jaringan ini di desain untuk jaringan berskala kecil dan menengah.



Gambar 2. 18 Model Jaringan *Peer to Peer*

Kelebihan model *peer to peer.*

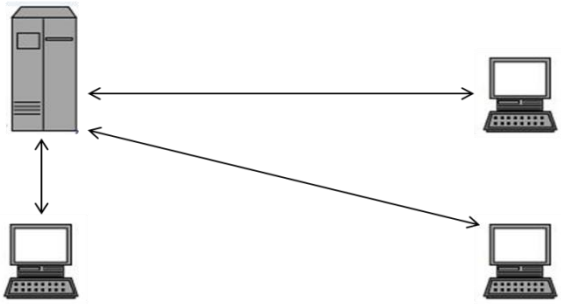
1. Tidak memerlukan biaya yang mahal, karena tidak membutuhkan *file server* yang terpusat.
2. Mudah dalam hal konfigurasi.

Kekurangan model *peer to peer.*

1. Tidak terpusat, terutama untuk penyimpanan data dan aplikasi.
2. Tidak menyediakan fasilitas untuk keamanan data.

2.2.7.5.2 *Client Server*

Client server merupakan jaringan komputer yang salah satu komputernya berfungsi sebagai *server* atau induk bagi komputer lain. Model *client-server* memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi pada satu atau dua *dedicated server*. Sebuah *file server* menjadi jantung dari keseluruhan sistem, memungkinkan sumber daya dan menyediakan keamanan. Model *client-server* menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak penggunaan secara bersama-sama sumber daya pada *file server.*



Gambar 2. 19 Model Hubungan *Client-Server*

Kelebihan model *client-server.*

1. Sumber daya dan keamanan data terpusat dikontrol melalui *server.*
2. Fleksibel dalam artian mudah dalam hal konfigurasi, dan apabila terjadi penambahan komputer baru tidak akan mengganggu sistem yang sedang berjalan.

Kekurangan model *client-server.*

1. Memerlukan biaya yang mahal, karena membutuhkan investasi untuk *dedicated file server*.
2. Memerlukan seorang administrator untuk mengatur agar sistem tetap berjalan efisien.
3. Seluruh *client* bergantung pada *server,* jadi apabila *server* mengalami *down* maka sistem tidak akan berjalan.

2.2.8 Permodelan Sistem

Dalam suatu proses pembuatan aplikasi, analisa dan rancangan merupakan sebuah terminologi yang sangat tua. Pada saat masalah ditlusuri dan spesifikasi dinegosiasikan, dapat dikatakan bahawa kita berada pada tahap perancangan. Merancang adalah menemukan sebuah cara yang dapat menyelesaikan masalah, salah satu tool/model untuk merancang pembangunan sebuah aplikasi yang berbasis *object oriented* adalah UML. Alasan mengapa harus menggunakan UML adalah *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek, alasan yang kedua adalah *dynamic modeling,* yang biasa digunakan untuk permodelan sistem yang dinamis dan *real time* [8].

2.2.8.1 UML

UML (*Unified Modelling Language*) adalah suatu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa permodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

Dengan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, yang diaman piranti tersebut dapat berjalan pada perangkat keras, sistem operasi, jaringan, dan ditulis dalam bahasa pemrograman apapun. Tetapi karena UML mengguanakan *class* dan *operation* pada konsep dasarnya, maka ia lebih cocok untuk penulisan sebuah aplikasi yang ditulis dalam bahasa pemrograman yang berorientasi objek.

UML mendefinisikan notasi dan *syntax,* notasi dalam UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram sebuah aplikasi dimana setiap bentuk memiliki makna tertentu. *Syntax* pada UML mendefinisikan bagaimana bentuk tersebut dapat dikombinasikan. Notasi pada UML diturunkan dari 3 notasi yang telah ada sebelumnya yaitu: Grady Booch *Object Oriented Design,* Jim Rumbaugh OMT (*Object Modeling Technique*) dan Ivan Jacobson OOSE (*Object Oriented Software Engineering*).

2.2.8.2 Jenis-Jenis UML

Menurut Whitten, Bentley, dan Dittman UML terdiri atas sembilan diagram yang dikelompokkan dalam lima kategori berdasarkan sudut pandangnya, yaitu:

2.2.7.2.1 Use Case Diagram

Use case diagram menggambarkan interaksi antara sistem dengan luar sistem dan sistem dengan pengguna. Dengan kata lain, use case diagram menggambarkan siapa yang akan menggunakan sistem dan dalam cara apa pengguna ingin berinteraksi dengan sistem.

2.2.7.2.2 *Static Structure Diagram*

UML menawarkan dua diagram untuk memodelkan struktur statis dari sistem informasi, yaitu :

1. *Class Diagram*

Class diagram merupakan sebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut) dari suatu sistem, selain itu *class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment,* pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok, yaitu sebagai berikut:

1. Nama
2. Atribut
3. Metode

Atribut dan metode dapat memiliki salah satu sifat berikut:

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected,* yaitu hanya dapat dipanggil oleh *class* yang bersangkutan dan anak yang mewarisinya.
3. *Public,* dapat dipanggil siapa saja.

2. *Object Diagram*

Class diagram melukiskan struktur sistem dalam bentuk objek. Disini digambarkan objek class yang membangun sistem berserta hubungan antar class.

2.2.7.2.3 *Interaction Diagram*

Interaction diagram memodelkan interaksi, terdiri atas sekumpulan objek, hubungan, dan pesan yang dikirimkan antar objek tersebut. Diagram ini memodelkan aspek dinamis dari sistem. UML memiliki dua diagram untuk tujuan ini, yaitu :

1. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display) berupa pesan yang digambarkan terhdap waktu. *Sequence diagram terdiri* dari dimensi vertikal yaitu waktu dan dimensi horizontal (objek-objek yang terkait). *Sequence* diagram biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *response* dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktifitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang akan dihasilkan.

2. *Collaboration Diagram*

Collaboration diagram serupa dengan *sequence diagram*, tetapi yang difokuskan disini bukanlah urutan (*sequence*) melainkan interaksi antar objek dalam format jaringan.

2.2.7.2.4 *State Diagram*

State diagram juga memodelkan aspek dinamis dari sistem. UML memiliki diagam untuk memodelkan perilaku kompleks dari objek dan diagram untuk memodelkan perilaku dari use case atau metode. Diagram tersebut adalah :

1. *Activity Diagram*

Activity diagram menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang

mungkin terjadi, dan bagaimana mereka berkahir. *Activity diagram* juga dapat menggambarkan proses pararel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, dimana sebagian besar *state* adalah *action* dan sebagian besar trasnisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* dari sebuah sistem secara eksak, tetapi lebih menggambarkan proses-proses aktivitas dari level atas secara umum.

2. *Statechart Diagram*

Statechart diagram digunakan untuk memodelkan aspek dinamis dari suatu objek. Pada statechart diagram diilustrasikan daur hidup objek, berbagai keadaan objek, dan peristiwa yang menyebabkan transisi dari keadaan yang satu ke keadaan yang lain.

2.2.7.2.5 *Implementation Diagram*

1. *Component Diagram*

Component diagram digunakan untuk menggambarkan organisasi dari sistem dan ketergantungan dari komponen perangkat lunak dalam sistem. *Component diagram* dapat juga digunakan untuk menunjukkan bagaimana kode program dibagi menjadi modul-modul (atau komponen).

2. *Deployment Diagram*

Deployment diagram mendeskripsikan arsitektur fisik dalam 'node' untuk perangkat keras dan perangkat Iunak dalam sistem. Disini digambarkan konfigurasi dari komponen perangkat lunak, processor, dan peralatan lain yang membangun arsitektur sistem secara run-time.