BAB 2
TINJAUAN PUSTAKA

2.1. Profil Perusahaan

PT. Pindad (Persero) merupakan perusahaan BUMN (Badan Usaha Milik Negara) yang bergerak dalam bidang Alutsista (Alat Utama Sistem Persenjataan) dan produk komersial, sebagai berikut [3]:

1. Produksi / Manufaktur melakukan produksi baik produk alutsista maupun nonalutsista, mengolah bahan mentah tertentu menjadi bahan prosuk maupun produk jadi serta melakukan proses assembling (perakitan) pada produk berikut:
   a. Produk senjata munisi
   b. Produk kendaraan khusus
   c. Produk pyroteknik, bahan pendorongan dan bahan peledak (militer dan komersial)
   d. Produk konversi energi
   e. Produk komponen, sarana dan prasarana dalam bidang transportasi
   f. Produk mesin industri dan peralatan industrial
   g. Produk mekanikal, elektrikal dan opto elektronik
2. Bidang jasa memberikan jasa untuk industri pertambangan, kosntruksi, mesin industri seperti :
   a. Perekayasaan system industrial
   b. Pemeliharaan produk/ peralatan industri
   c. Pengujian mutu dan kalibrasi
   d. Konstruksi
   e. Pemesinan
   f. Heat and surface treatment
   g. Drilling
   h. Blasting
   i. Jasa pemusnahan bahan peledak
   j. Jasa transportasi bahan peledak

k. Jasa pergudangan bahan peledak

3. Perdagangan melaksanakan pemasaran, penjualan dan distribusi produk dan jasa perusahaan termasuk produksi pihak lain, baik di dalam maupun di luar negeri seperti :
   a. Ammonium Nitrate
   b. Panfo
   c. Detonator Listrik
   d. Detonator Non Listrik
   e. Detonating COD
   f. Booster
   g. Geodetoseis
   h. Geopentoseis

2.1.1. Sejarah Perusahaan

1. Masa Kolonial Belanda dan Pendudukan Jepang

Pada tahun 1808, William Herman Daendels mendirikan bengkel pembuatan dan perbaikan munisi *Artilerie Constructie Winkel* (ACW) yang berlokasi di Surabaya dan Semarang. Pada tahun 1932 ACW dipindahkan ke Bandung. Di era pendudukan Jepang, ACW tidak mengalami perubahan, penambahan instalasi, maupun proses produksinya. Perubahan hanya terjadi pada nama yang berganti menjadi *Daichi Ichi Kozo*. Pada tanggal 9 Oktober 1945 Laskar Pemuda Pejuan berhasil merebut ACW dari tangan Jepang dan menamakannya Pabrik Senjata Kiaracondong. Pendudukan pemuda tidak berlangsung lama, karena sekutu kembali ke Indonesia dan mengambil alih kekuasaan. Pabrik pertama acw kemudian digabung dengan pabrik lainnya dan berubah menjadi *Leger Produktie Bedrijven* (LPB) [4].

2. Bagian Dari TNI AD

Pada tahun 1949 Belanda harus menyerahkan aset-asetnya secara bertahap pada pemerintahan Indonesia. LPB kemudian berganti namanya menjadi Pabrik Senjata dan Mesiu (PSM) yang pengelolaannya diserahkan

kepada Tentara Nasional Indonesia Angkatan Darat (TNI-AD). Delapan tahun berjalan, PSM pun diubah namanya menjadi Pabrik Alat Peralatan Angkatan Darat (Pabal AD) pada tanggal 1 Desember 1958. Pabal AD bukan sekedar memperoduksi senjata dan munisi saja namun juga peralatan milter yang lain. Sekitar tahun 1962, nama Pabal AD diubah menjadi Perindustrian TNI Angkatan Darat (Pindad). Tahapan pengembangan di era Pindad lebih berfokus pada tujuan pembinaan yang disesuaikan dengan prinsip-prinsip pengelolaan terpadu dan kemajuan teknologi mutakhir.

Dalam perkembangan selanjutnya, sebagai realisasi Keputusan Menteri Pertahanan dan Keamanan/Panglima Angkatan Bersenjata No. Kep/18/IV/1976 tertanggal 28 April 1976 tentang Pokok-pokok Organisasi dan Prosedur Tentara Nasional Indonesia Angkatan Darat nama Kopindad dikembalikan menjadi Pindad. Pindad berubah dari komando utama pembinaan menjadi badan pelaksana utama di lingkungan TNI-AD.

Seiring perubahan tersebut Pindad diharapkan dapat mengembangkan kemampuan teknologi dan produktivitasnya dalam memenuhi kebutuhan logistik TNI-AD sehingga mengurangi ketergantungan pada luar negeri. Selain itu diharapkan juga dapat mengembangkan sarana prasarana non-militer yang dapat menunjang pembangunan nasional di bidang pertanian, perkebunan, pertambangan, industri dan transportasi baik untuk instansi pemerintah, swasta maupun masyarakat luas.

3. Pindad Sebagai Perseroan

Pada tahun 1980-an pemerintah Indonesia semakin gencar menggalakan program alih teknologi, saat inilah muncul gagasan untuk mengbah status pindad menjadi peruahaan berbentuk perseroan terbatas. Berdasarkan keputusan Presiden RI No.47 Tahun 1981, Badan Pengkajian Penerapan Teknologi (BPPT) yang sudah berdiri sejak tahun 1978, harus lebih memperhatikan proses transformasi teknologi yang ditetapkan

pemerintah Indonesia itu, termasuk pengadaan mesin-mesin untuk kebutuhan Industri.

Perubahan status Pindad dilatarbelakangi oleh keterbatasan ruang gerak Pindad sebagai sebuah industri karena terikat peraturan-peraturan dan ketergantungan ekonomi pada anggaran Dephankam sehingga tidak dapat mengembangkan kegiatan produksinya. Selain itu, Pindad pun dinilai membebani Dephankam karena biaya penelitian dan pengembangan serta investasi yang cukup besar. Karena itu Dephankam menyarankan pemisahan antara *war making activities* dan *war support activities*. Kegiatan Pindad memproduksi prasarana dan perlengkapan militer adalah bagian*war support activities* sehingga harus dipisahkan dari Dephankam dan menjadi perseroan terbatas yang sahamnya dimiliki oleh pemerintah Indonesia [4].

2.1.2. Visi dan Misi

Berikut merupakan penjelasan visi dan misi dari PT. Pindad :

2.1.2.1. Visi Perusahaan

Menjadi produsen peralatan pertahanan dan keamana terkemuka di Asia pada tahun 2023, melalui upaya inovasi produk dan kemitraan strategis [5].

2.1.2.2. Misi Perusahaan

Melaksanakan usaha terpadu di bidang peralatan pertahanan dan keamanan serta peralatan industrial untuk mendukung pembangunan nasional dan secara khusus untuk mendukung pertahanan dan keamanan negara.

2.1.3. Badan Hukum Instansi

Berdasarkan Keputusan Presiden RI no. 59 tahun 1983, sejak 29 April 1983 PINDAD (Perindustrian TNI-AD) beralih status menjadi Badan Usaha Milik Negara Industri Strategis (BUMNIS) dengan nama PT. Pindad (Persero). Pada tahun 2002 PT. Pindad (Persero) di bawah pembinaan Kementrian Negara BUMN.

2.1.4. Logo Perusahaan



Gambar 2.1 Logo Pindad

Makna Logo PT. Pindad (Persero) :

1. Pengertian

   Logo PT. Pindad (Persero) berupa senjata cakra dangan bintang bersudut lima dan bertuliskan Pindad.

2. Arti dan Makna Logo

   a. Cakra, adalah senjata pamungkas kresna

      Keampuhannya memiliki kemampuan untuk menghancurkan dan ataun sebaliknya menambarkan (menetralisis) bahaya yang datang mengancamnya, sehingga dengan demikian memiliki potensi untuk mendukung perang ataupun menciptakan kedamaian.

   b. Bintang bersudut lima

      Melambangkan bahwa gerak dan laju PT. Pindad (Persero) berlandaskan Pancasila, falsafah / dasar / ideologi bangsa dan Negara Indonesia di dalam ikut serta mewujudkan teriptanya masyarakat adil dan makmur.

   c. Pisau Frais, melambangkan industri dengan:

      1. 4 (empat) buah lubang Spi

         Melambangkan kemampuan teknologi untuk mengelola, meniru, merubah, dan menciptakan suatu bahan atau produk.

      2. 8 (delapan) buah pisau (cakra)

         Melambangkan kemampuan untuk memproduksi sarana militer / hankam dan sarana sipil / komersil dalam rangka ikut serta mendukung terciptanya ketahanan nasional

bangsa Indonesia yang bertumpu pada 8 (delapan) gatra (aspek).

d. Bintang dan ekor

Melambangkan pengendalian gerak dan laju PT. Pindad (Persero) secara berdaya dan berhasil guna, 4 (empat) helai sirip ekor, melambanhkan keserasian gerak atara unsur – unsur manusia, modal, metoda, dan pemasaran.

e. Warna

1. Senjata Cakra : Biru Laut
2. Bintang : Kuning Emas
3. Tulisan “Pindad” : Kuning Emas

2.1.5. Struktur Organisasi

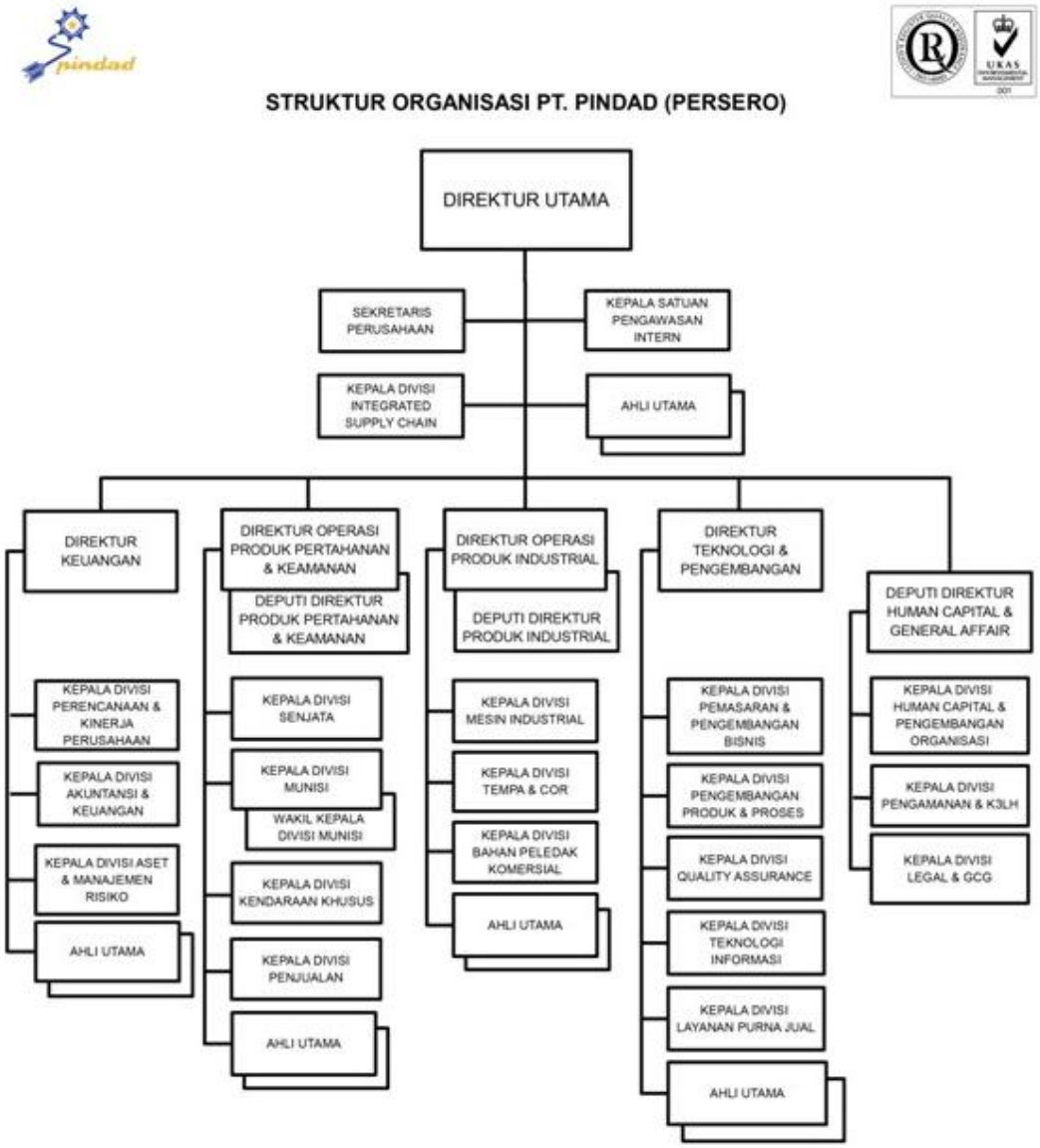
Struktur Organisasi perusahaan merupakan bagian yang sangat penting dalam penyelenggaraan kegiatan, karena struktur organisasi erusahaan terdapat pendelegasian wewenang terhadap setiap lini/tingkatan dalam perusahaan.

PT. Pindad (Persero) adalah perusahaan industri manufaktur dan satu-satunya produsen dalam negeri untuk peralatan militer yang memiliki fasilitas memadai serta sumber daya manusia yang berpengalaman yang berbasis teknologi mekanik. Struktur organisasi PT. Pindad (Persero) merupakan wujud sinergi dari kredibilitas para staf dengan fungsi manajer (Direktur Utama) yang memiliki peranan penting dalam menentukan penggunaan kebijakan / keputusan yang dibantu dengan beberapa staf dalam mewujudkan objek (tujuan) perusahaan.

Struktur organisasi PT.Pindad (persero) merupakan suatu sistem, dimana satu departemen dengan departemen lain memiliki keterkaitan dan saling mempengaruhi dalam proses pencapaian objek perusahaan melalui kerjasama [6].

Jajaran Direksi PT. Pindad (Persero) meliputi Direktur Utama yang membawahi 4 direktur lainnya, diantaranya :

1. **Direktur Keuangan, yang bertanggung jawab terhadap :**
   a. Kepala divisi dan kinerja perusahaan.
   b. Kepala divisi akuntansi dan keuangan.
   c. Kepala divisi aset dan manajemen resiko.
2. **Direktur Operasi Produk Pertahanan dan Keamanan, yang bertanggung jawab terhadap:**
   1. Kepala divisi senjata.
   2. Kepala divisi munisi.
   3. Kepala divisi kendaraan khusus.
   4. Kepala divisi penjualan.
3. **Direktur Operasi Produk Industrial, yang bertanggung jawab terhadap:**
   1. Kepala divisi mesin industrial.
   2. Kepala divisi tempa dan cor.
   3. Kepala divisi bahan peledak komersial.
4. **Direktur Teknologi dan Pengembangan, yang bertanggung jawab terhadap:**
   1. Kepala divisi pemasaran dan pengembangan bisnis.
   2. Kepala divisi pengembangan produk dan proses.
   3. Kepala divisi quality assurance.
   4. Kepala divisi teknologi informasi.
   5. Kepala divisi layanan purna jual.



Gambar 2.2 Struktur Organisasi PT. Pindad (Persero).

2.2. Landasan Teori

Landasan teori merupakan penjelasan konsep dasar dari teori-teori yang berkaitan dalam pembangunan aplikasi sistem keamanan algoritma enkripsi citra digital menggunakan kombinasi dua *chaos* map dan penerapan teknik selektif.

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 [7].

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 [7].

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. Otentifikasi

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. Kriptografi

Perkembangan teknologi telekomunikasi dan penyimpanan data dengan menggunakan komputer memungkinkan pengiriman data jarak jauh yang relatif cepat dan murah. Di lain pihak pengiriman data jarak jauh melalui gelombang radio maupun media lain yang digunakan masyarakat luas (*public)* sangat memungkinkan pihak lain dapat menyadap dan mengubah data yang dikirimkan [8].

Salah satu cara yang dapat ditempuh untuk mengatasi masalah ini adalah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak ketiga. Transformasi ini memberikan solusi pada dua masalah keamanan data, yaitu masalah privasi (*privacy*) dan keautentikan (*authentication*). Privasi mengandung arti bahwa data yang dikirimkan hanya dapat dimengerti

informasinya oleh penerima yang sah. Sedangkan keautentikan mencegah pihak ketiga untuk mengirim data atau mengubah data yang dikirimkan.

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 kaitanya 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 (teks asli yang dapat dimengerti), sedangkan pesan hasil penyandian disebut dengan *cipherteks* (pesan tersandi). Pesan yang telah tersandikan dapat dikembalikan lagi ke pesan aslinya hanya oleh orang yang berhak (orang yang mengetahui metode penyandian atau memiliki kunci penyandian). Proses penyandian plainteks menjadi cipherteks disebut enkripsi dan proses mengembalikai cipherteks menjadi plainteks disebut dekripsi.



Gambar 2.3 Konsep Enkripsi

2.2.4.1. 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.

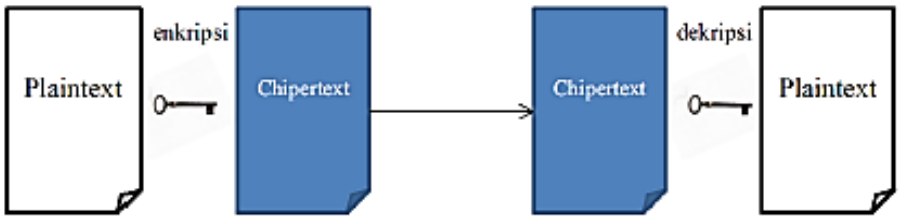
Namun pada 30 tahun terakhir ini, kriptografi tidak hanya dimonopoli oleh pihak militer saja, hal yang sama juga dilakukan oleh individu-individu yang menginginkan pesan dan komunikasi mereka tidak diketahui oleh pihak lain. Apalagi pada zaman sekarang ini persaingan yang begitu tinggi, mereka rela mengeluarkan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

2.2.4.2. 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

A. Kriptografi Knnci 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 2.4 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. Blok Cipher

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*

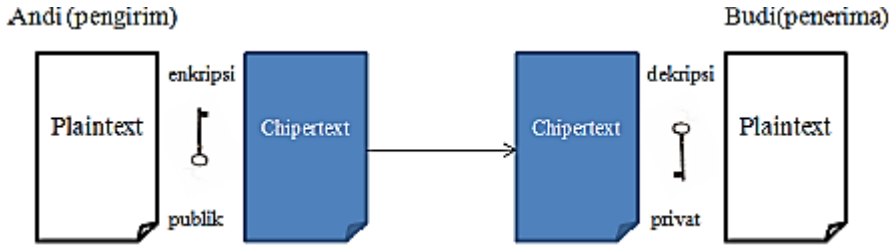
adalah DES (*Data Encryption Standard*), AES (*Advanced Encryption Standard*), dan IDEA (*International Data Encryption Algoritm*).

2. Stream Cipher

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.

B. 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 2.5 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:

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

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

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

$$CipherText \longleftarrow Enkripsi\ (K_{publik}, M) \qquad (2.2)$$

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

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

$$M \longleftarrow Dekripsi\ (K_{privat}, CipherText) \qquad (2.3)$$

Budi mendapatkan *M* yang sama dengan *M* yang sebelum di enkripsi oleh Andi jika Andi menggunakan (K_{privat}) 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 [9].

2.2.5. Citra Digital

Citra secara harafiah, adalah gambar pada bidang dua dimensi (dwimatra). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sedangkan citra digital adalah citra yang dapat diolah oleh komputer. Dalam konteks lebih luas, pengolahan citra digital mengacu pada pemrosesa setiap data dua dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun komplek yang direpresentasikan dengan deret bit tertentu.

Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut *pixel*, yaitu elemen terkecil dari sebuah citra. Sebuah citra digital dapat ditulis dalam bentuk 2.1 berikut [10] :

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & .. & f(0,M-1) \\ f(1,0) & .. & .. & f(1,M-1) \\ .. & .. & .. & .. \\ f(N-1,0) & f(N-1,1) & .. & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

Suatu citra f(x,y) dalam fungsi matematis dapat dituliskan sebagai berikut:

$$0 \leq x \leq M\text{-}1$$

$0 \leq y \leq N\text{-}1$

$$0 \leq f(x,y) \leq G\text{-}1$$

dimana : M = jumlah pixel baris (*row*) pada array citra

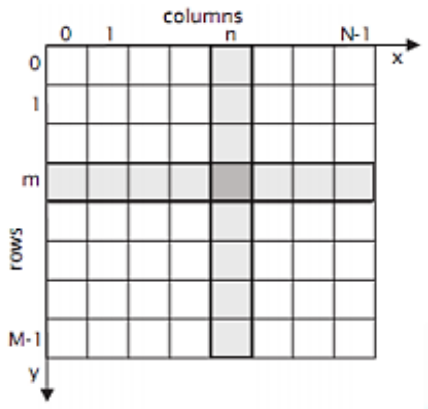
N = jumlah pixel kolom (*column*) pada array citra

G = nilai skala keabuan (*grayscale*)

Besarnya nilai M, N, dan G pada umumnya merupakan perpangkatan dari dua

$$M = 2^m ; N = 2^n ; G = 2^k$$

dimana nilai n,m dan k adalah bilangan bulat positif. Interval (0,G) disebut skala keabuan (*grayscale*). Besar G tergantung pada proses digitalisasi. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan).



Gambar 2.6 Representasi citra digital dalam 2 dimensi

Terdapat banyak format file dari citra digital, format file citra ini terdiri dari beberapa jenis dan setiap jenisnya memiliki karakteristik masing-masing. Beberapa contoh format umum, pada citra digital yaitu *Bitmap* (BMP), *Joint Photographic Group Experts* (JPEG), *Graphics Interchange Format* (GIF), dan *Portable Network Graphics* (PNG).

1. *Bitmap (BMP)*

Bitmap merupakan format baku citra pada sistem operasi windows dan IBM OS/2. Citra berformat BMP merupakan citra yang tidak terkompresi, sehingga pada umumnya citra berformat BMP mempunyai ukuran yang relatif lebih besar dibandingkan dengan forman citra lainnya. Intensitas pixel dari citra berformat BMP dipetakan ke sejumlah bit tertentu. Panjang setiap pixel pada bitmap yaitu 4 bit, 8 bit, sampai 24 bit yang merepresentasikan nilai intensitas pixel. Dengan demikian ada sebanyak 28 = 256 derajat keabuan, mulai dari 0 sampai 255.

2. *Joint Photographic Group Experts (JPEG)*

Joint Photographic Group Experts (JPEG) merupakan standar kompresi file yang dikembangkan oleh Group Joint Photographic Experts menggunakan kombinasi DCT dan pengkodean Huffman untuk mengkompresikan citra. Citra JPEG merupakan citra terkompresi yang bersifat lossy, artinya citra tidak bisa dikembalikan ke bentuk aslinya. Citra ini memiliki ukuran yang relatif lebih kecil dibandingkan dengan citra berformat BMP karena telah terkompresi.

3. Graphics Interchange Format (GIF)

Graphics Interchange Format (GIF) merupakan format citra terkompresi. Ukuran dari citra berformat GIF merupakan yang terkecil dari semua format citra digital. Kombinasi warna yg tersedia sebanyak 256 warna. Citra jenis ini banyak dugunakan untuk keperluan website, untuk membuat sebuah icon, logo dan keperluan lainnya yang mengharuskan menggunakan citra dengan ukuran yang kecil. Citra jenis ini juga dapat dibuat animasi.

4. Portable Network Graphics (PNG)

Portable Network Graphics (PNG) adalah salah satu format penyimpanan citra yang menggunakan metode kompresi yang tidak menghilangkan bagian dari citra tersebut (*lossless compression*). Citra berformat PNG merupakan salah satu format yang baik untuk digunakan pengolahan citra, karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah [11].

Format PNG ini diperkenalkan untuk menggantikan format penyimpanan citra GIF. Selain itu, citra dengan format PNG mempunyai faktor kompresi yang lebih baik dibandingkan dengan GIF (5%-25% lebih baik dibanding format GIF).

Satu fasilitas dari GIF yang tidak terdapat pada PNG format adalah dukungan terhadap penyimpanan multi-citra untuk keperluan animasi. Untuk keperluan pengolahan citra, meskipun format PNG bisa dijadikan alternatif selama proses pengolahan citra - karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah (sehingga penyimpanan berulang ulang dari citra tidak akan menurunkan kualitas citra) namun format JPEG masih menjadi pilihan yang lebih baik.

PNG (Format berkas grafik yang didukung oleh beberapa web browser. PNG mendukung transparansi gambar seperti GIF, berkas PNG bebas paten dan merupakan gambar bitmap yang terkompresi. PNG diciptakan untuk menggantikan keberadaan GIF karena masalah lisensi. Format PNG lebih baik daripada GIF. Masalahnya ada pada kurangnya dukungan yang dimampukan oleh web browser. Format ini dibuat sebagai alternatif lain dari

format GIF. Format ini digunakan untuk menyimpan berkas dengan kedalaman 24 bit serta memiliki kemampuan untuk menghasilkan background transparan dengan pinggiran yang halus.

Format PNG menggunakan metode kompresi lossless untuk menampilkan gambar 24-bit atau warna-warna solid pada media sharing (online). Format ini mendukung transparansi di dalam alpha channel. Format PNG sangat baik digunakan pada dokumen daring (online), dan mempunyai dukungan warna yang lebih baik saat dicetak dari pada format GIF.

Pada dasarnya, format PNG bukan merupakan format baru karena telah dikembangkan pada tahun 1995 untuk mengganti format GIF dan format TIFF. Format ini tidak digunakan lagi secara luas oleh browser dan perangkat lunak aplikasi pengolah gambar, sehingga dukungan terhadap format tidak begitu besar hingga tahun 2003, di mana format PNG semakin dikenal dan dipergunakan untuk aplikasi manipulasi gambar.

2.2.5.1. Jenis Citra

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga jenis citra tersebut yaitu citra berwarna, citra berskala keabuan, dan citra biner.

1. Citra Berwarna

Citra berwarna, atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan delapan bit (nilainya berkisara antara 0 sampai 256). Dengan demikian, kemungkinan warna yang dapat disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna. Tabel 2.1 menunjukan contoh warna dan nilai RGB [12].

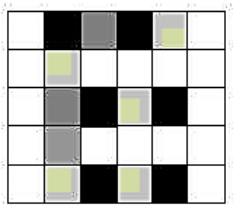
Tabel 2.1 Warna dan nilai penyusun warna

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0

Putih	255	255	255
Kuning	0	255	255

2. Citra Berskala Keabuan (*Grayscale*)

Citra *grayscale* disebut juga dengan citra 8-bit karena memiliki 2^8 (256) kemungkinan nilai pada masing-masing pikselnya. Nilai tersebut dimulai dari nol untuk warna hitam dan 255 untuk warna putih. Citra ini disebut juga citra hitam putih atau citra *monoakromatik.* Skala keabuan 4 bit jumlah kemungkinan s^4 = 16 warna. Kemungkinan warna 0 (min) sampai 15 (max). Skala keabuan 8 bit jumlah kemungkinan 2^8 = 256 warna kemungkinan warna 0 (*min*) sampai 255 (*max*).

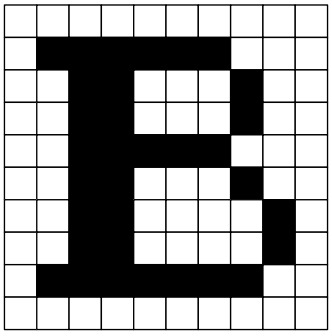


= 15 0 6 0 13 15
= 15 12 15 15 15 15
= 15 5 0 12 0 15
= 15 8 15 15 15 15
= 15 10 0 13 0 15

Gambar 2.7 Skala Keabuan 4 bit

3. Citra Biner

Citra biner (*binary image*) adalah citra yang hanya mempunyai dua nilai derajat keabuan: hitam dan putih. *Pixel – pixel* objek bernilai 1 dan *pixel-pixel* latar belakang bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Jadi, pada citra biner, latar belakang berwarna putih sedangkan objek berwarna hitam. [13]



0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	1	1	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0
0	0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0

Gambar 2.8 Citra Biner dan Representasi Citra Biner

2.2.6. Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++, namun dirancang agar lebih mudah dipakai dan platform independen, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer. Bahasa ini juga dirancang untuk pemrograman di internet sehingga dirancang agar aman dan portable. [14]

2.2.6.1. Sejarah Perkembangan Java

Bahasa pemograman 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 program lainnya dari Sun Microsistems. Salah satu hasil proyek ini adalah mascot Duke yang dibuat oleh Joe Palrang.

Pertemuan proyek berlangsung di sebuah gedung perkantoran Sand Hill Road di Menlo Park. Sekitar musim panas 1992 proyek ditutup dengan menghasilkan sebuah program Java Oak pertama yang ditunjukan 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 33 mereka. Pada awal tahun 1990-an, internet masih merupakan rintisan yang dipakai hanya di kalangan akademis dan militer.

Mereka menjadikan perambah (browser) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner,

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

Pada sekitar bulan maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan pemberitaan pertama kali pada surat kabar San Jose Mercury News pada tanggal 23 Maret 1995.

2.2.6.2. Versi Awal

Versi awal Java ditahun 1996 sudah merupakan versi release sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

1. Java.lang: Peruntukan kelas elemen-elemen dasar.
2. Java.io: Peruntukan kelas input dan output, termasuk pengguna berkas.
3. Java.util: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas penanggalan.
4. Java.net: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan computer lain menggunakan jaringan TCP/IP.
5. Java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI).
6. Java.applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

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 [15].

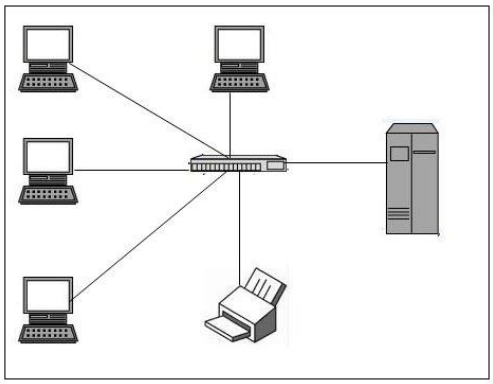
Berikut pembagian jaringan komputer berdasarkan skala/area :

1. LAN (*Local Area Network*)
2. MAN (*Metropolitan Area Network*)
3. WAN (*Wide Area Network*)
4. 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 2.9 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:

A. Kabel

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

1. Kabel *Coastal*

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.

B. 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

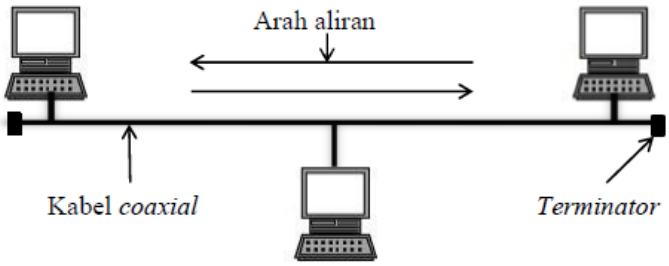
Topologi jaringan adalah suatu aturan/*rules* bagaimana menghubungkan komputer (*node*) satu sama lain secara fisik dan pola hubungan antara komponen-komponen yang berkomunikasi melalui media/peralatan jaringan, seperti: *server*, *workstation, hub/switch,* dan pengabelanya (media transimisi data). Ketika 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:

A. Topologi Bus

Topologi *bus* sering disebut *daisy chain* atau *ethernet bus topologies*. Sebutan terakhir diberikan karena perangkat jaringan yang digunakan pada

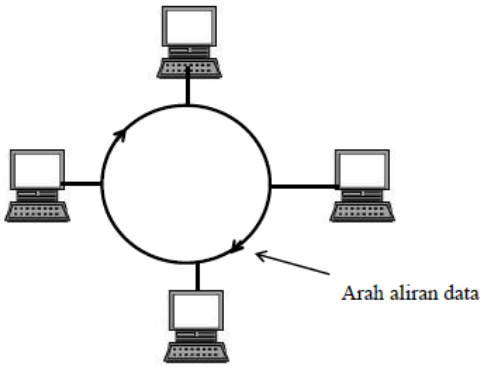
topologi ini adalah NIC (*Network Interface Card*) bernama *ethernet.* Topologi *bus* dapat dikenali dari penggunaan sebuah kabel *backbone* (kabel utama) yang menghubungkan semua peralatan jaringan. Apabila kabel utama terputus akan menyebabkan jaringan mati total, karena kabel *backbone* (kabel utama) merupakan satu-satunya jalan bagi lalu lintas data. Artinya, apabila salah satu sambungan terputus maka seluruh sambungan akan ikut terputus juga.



Gambar 2.10 Topologi Bus

B. 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.

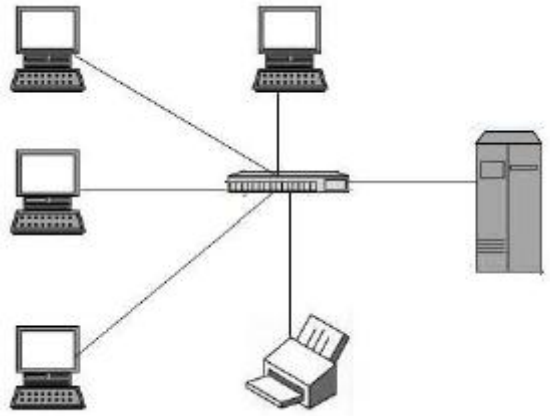


Gambar 2.11 Topologi Ring

C. 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

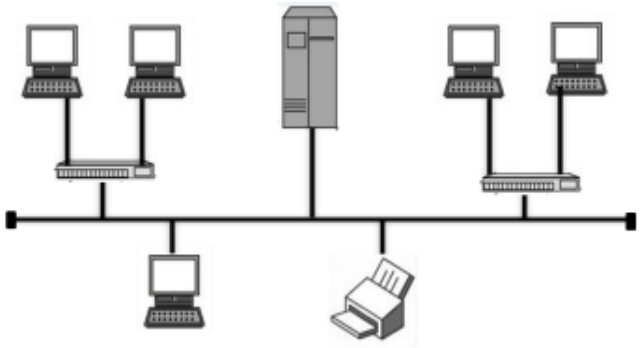
banyak digunakan untuk jaringan rumahan, perkantoran, sekolah dan lain-lain.



Gambar 2.12 Topologi Star

D. 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 2.13 Topologi Tree

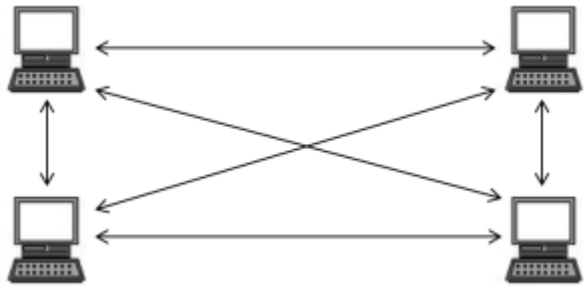
2.2.7.5. Model Hubungan Pada LAN

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.

A. *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 2.14 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

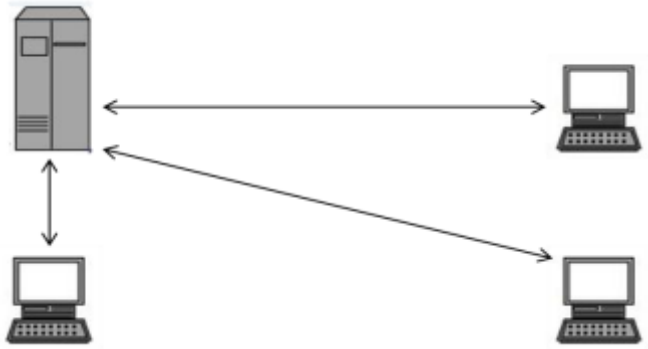
Kekurangan model *peer to peer*:

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

B. 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 2.15 Model Hubunga *Client-Server*

Kelebihan model *client-server*:

1. Sumber daya keamanan data terpusan 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. Chaos Map

Teori chaos ini merupakan cabang dari matematika yang mempelajari bagaimana membangkitkan bilangan secara acak. Teori chaos ini sangant sensitive terhadap nilai awal (*nitial condition*). Hal ini sangat berguna dan dapat diterapkan di dalam dunia kriptografi sebagai pembangkit kunci acak yang nantinya akan diolah sebagai sarana dalam melakukan proses enkripsi. Semakin acak bilangan yang dihasilkan, semakin baik pula tingkat keamanan dari suatu *chiperteks*.

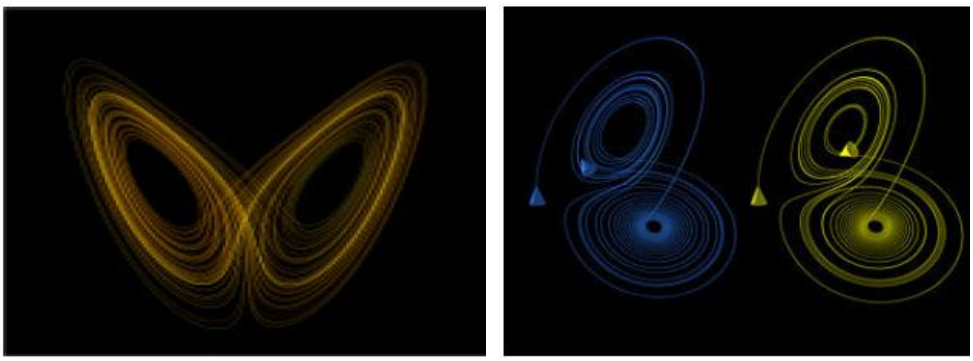
2.2.8.1. Sejarah Teori Chaos

Gagasan teori chaos atau teori acak telah dipikirkan sejak lama, akan tetapi yang tercatat sebagai penemu teori chaos ini adalah Henri Poncare yang pada tahun 1880 menemukan bahwa terdapat orbit yangbersigat nonperiodik, dalam arti tidak memiliki proses kemunculan secara tetap atau formulatif. Selain itu Jacques Hadamard juga mempublikasikan teori serupa pada tahun 1898. Selain kedua orang tersebut, masih banyak teori yang bermunculan pada zaman tersebut yang hampir semuanya dikembangkan oleh matematikawan.

Teori chaos sendiri mulai diformulasikan sejak pertengahan abad ke -20, dengan perintisnya adalah Edward Lorenz yang menemukan permasalahan teori chaos saat melakukan peramalan cuaca pada tahun 1961. Lorenz menemukan bahwa prediksi yang ia buat dengan mesin ternyata jauh berbeda dengan kenyataan yang ada. Hal inilah yang menjadi dasar berkembangnya teori chaos. Setelah penemuan dari Lorenz ini, banyak ahli yang mempelajari teori chaos dan menerapkannya pada berbagai bidang, mulai dari dunia meteorologi, biologi, matematika, hingga dunia kriptografi[16].

2.2.8.2. Definisi Teori Chaos

Dalam dunia matematika, teori chaos menggambarkan kebiasaan dari suatu system dinamis, yang keadaannya selalu berubah seiring dengan berubahnya waktu, dan sangat sensitif terhadap kondisi awal dirinya sendiri. Teori chaos ini juga sering disebut dengan sebutan butterfly effect. Pada Gambar 2.16 akan menggambarkan teori chaos sekaligus menjawab megapa teori chaos disebut memiliki butterfly effect.



Gambar 2.16 Butterfly Effect

Dikarenakan oleh sensitivitas yang dimiliki teori chaos terhadap keadaan awal dirinya, teori chaos memiliki sifat untuk muncul secara chaos (kacau). Bahkan perubahan keadaan awal sekecil (10^-100) saja akan membangkitkan bilangan yang benarbenar berbeda. Hal ini terjadi walaupun system yang digunakan bersifat deterministik, dalam arti perubahan kondisi dari kondisi yang ada sekarang bersifat statik atau tetap. Salah satu contoh nyata dari teori chaos ini dapat kita lihat pada kehidupan sehari-hari, terutama pada sistem alamiah seperti cuaca. Seperti yang telah disebutkan dalam bab sebelumnya, bahwa Lorentz telah meneliti perubahan cuaca, dan mencoba memodelkannya dengan mesin yang dia rancang sendiri. Akan tetapi, perubahan cuaca sebenarnya ternyata sangat berbeda dengan prediksi mesin yang dirancang oleh Lorentz.

Contoh lain dari chaos ini adalah pertumbuhan suatu populasi pada lingkungan hidup, gerakan pada neuron, pergerakan satelit dalam sistem tata surya, hingga pergerakan dari kerak bumi.

2.2.9. Arnold Cat Map

Arnold cat map (ACM) merupakan fungsi *chaos* dwimatra dan bersifat *reversible.* Fungsi *chaos* ini ditemukan oleh Vladimir Arnold pada tahun 1960, dan kata "*cat*" muncul karena dia menggunakan citra seekor kucing dalam eksperimennya [17].

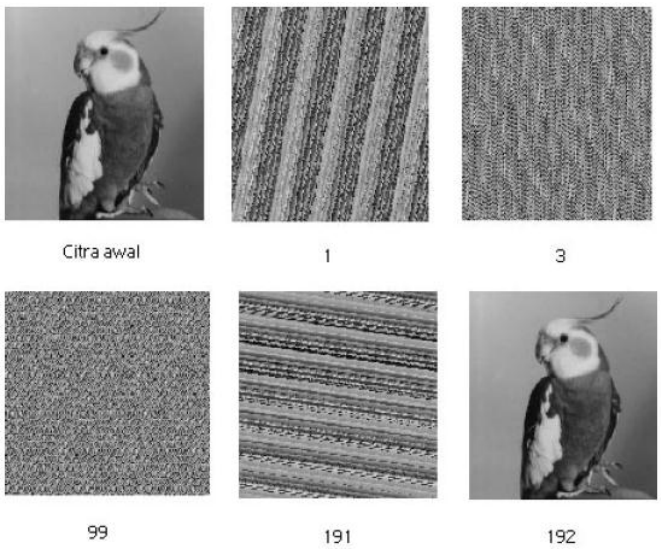
ACM mentransformasikan koordinat (x,y) di dalam citra yang berukuran N x N ke koordinat baru (x' ,y'). Persamaan iterasinya adalah :

$$\begin{bmatrix} X \text{ i-1} \\ Y \text{ i-1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} X \text{ i} \\ Y \text{ i} \end{bmatrix} \text{ mod N} \qquad (1)$$

yang dalam hal ini (x_i , y_i) adalah posisi pixel di dalam citra, (x_{i-1} , y_{i-1}) posisi pixel yang baru setelah iterasi ke-I; b dan c adalah integer positif sembarang. Determinan matriks $\begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}$ harus sama dengan 1 agar hasil transformasinya bersigat *area-preserving*, yaitu tetap berada di dalam area citra yang sama. ACM termasuk pemetaan yang bersifat satu ke satu karena setiap posisi pixel selalu ditransformasikan ke posisi lain secara unik. ACM diiterasikan sebanyak m kali dan setiap iterasi menghasilkan citra yang acak. Nilai b, c, dan jumlah iterasi m dapat dianggap sebagai kunci rahasia.

Proses yang terjadi di dalam setiap *iterasi ACM* adalah pergeseran (*shear*) dalam arah *y*, kemudian dalam arah *x*, dan semua hasilnya (yang mungkin berada di luar area gambar) dimodulokan dengan *N* agar tetap berada di dalam area gambar (*area preserving*).

Setelah *ACM* diiterasi sebanyak *m* kali, maka terdapat *T* sedemikian sehingga (*xT*, *yT*) = (*x*, *y*), yang dalam hal ini nilai *T* bergantung pada *b*, *c*, dan ukuran *N* (We-bin, 2009). Ini berarti sesudah *ACM* diiterasi sebanyak *T* kali, maka hasil iterasinya kembali ke citra semula, sehingga dikatakan *ACM* besifat *reversible* dan periodenya adalah *T*. Menurut Struss (2009), penelitian Freeman J. Dyson dan Harold Falk menemukan bahwa $T < 3N$. Pada Gambar 2.17 memperlihatkan iterasi *ACM* terhadap citra ‘burung’. Pada iterasi ketiga hasilnya sudah terlihat seperti citra acak, semakin banyak iterasinya citra hasil semakin acak (dalam hal ini *ACM* telah berada dalam fase *chaos*). Jika proses iterasi diteruskan maka hasilnya kembali menuju citra semula. Pada contoh ini citra ‘burung’ kembali ke bentuk semula pada iterasi ke-192 sehingga dikatakan periodenya adalah $T = 192$.



Gambar 2.17 Iterasi ACM pada Citra Burung dengan Periode T = 192.

Seperti umumnya fungsi *chaos* yang bersifat deterministik, citra yang sudah teracak oleh ACM dapat direkonstruksi menjadi citra semula dengan menggunakan kunci yang sama (b, c , dan m). Persamaan iterasinya adalah

$$\begin{bmatrix} X \text{ i-1} \\ Y \text{ i-1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}^{-1} \begin{bmatrix} X \text{ i} \\ Y \text{ i} \end{bmatrix} \bmod N \quad (2)$$

setelah iterasi terakhir citra hasil sama seperti semula. Proses dekripsipun selesai.

2.2.10. Logistic Map (Persamaan Logistic)

Persamaan logistik merupakan contoh pemetaan polinomial derajat dua, dan seringkali digunakan sebagai contoh bagaimana rumitnya sifat chaos (kacau) yang dapat muncul dari suatu persamaan yang sangat sederhana. Persamaan ini dipopulerkan oleh seorang ahli biologi yang bernama Robert May pada tahun 1976, melanjutkan persamaan logistik yang dikembangkan oleh Pierre Francois Verhulst.

Secara matematis, persamaan logistik dapat dinyatakan dengan persamaan:

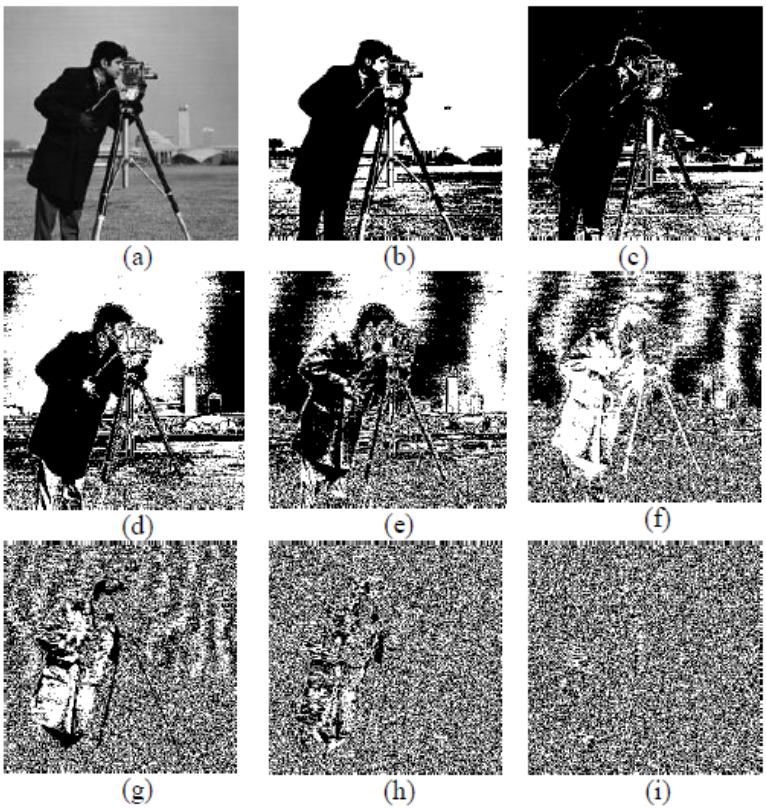
$$X_{i+1} = r\, x_i\, (1 - x_i) \quad (3)$$

nilai-nlai *xi* adalah bilangan riil di dalam selang (0, 1), sedangkan r adalah parameter fungsi yang menyatakan laju pertumbuhan yang nilainya di dalam selang (0, 4). *Logistic Map* akan bersifat *chaos* bilamana $0 \leq r \leq 4$. Untuk memulai iterasi *Logistic Map* diperlukan nilai awal *x*0. Perubahan sedikit saja pada nilai awal ini (misalnya sebesar 10^{-10}) akan menghasilkan nilai-nilai *chaos* yang berbeda secara signifikan setelah *Logistc Map* diiterasi sejumlah kali. Di dalam sistem kriptografi simetri, nilai awal *chaos*, *x*0, dan parameter μ berperan sebagai kunci rahasia. Nilai-nilai acak yang dihasilkan dari persamaan (3) tidak pernah berulang kembali sehingga *Logistic Map* dikatakan tidak mempunyai periode [17].

2.2.11. Enkripsi Selektif

Pengubahan bit MSB menjadi dasar enkripsi selektif citra digital. Dengan hanya mengenkripsi bit MSB maka proses enkripsi menjadi lebih efisien, sebab tidak semua data citra dienkripsi dengan stream cipher.

Setiap *pixel* direpresentasikan dalam sejumlah *byte*, susunan bit pada setiap *byte* adalah $b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$. Bit – bit paling kiri adalah bit paling berarti (*most significant bits* atau MSB), sedangkan bit – bit paling kanan adalah *least significant bits* atau LSB. Jika setiap bit ke-1 dari setiap *pixel* pada citra *grayscale* diekstrasi dan diplot ke dalam setiap *bitplane image* maka kita memperoleh delapan buah citra biner. Gambar 2.18 (a) sampai 2.6 (i) memperlihatkan *bitplane image* dari citra *cameraman*. Gambar 2.18 (b) hingga 2.6 (f) yang diambil dati bit-bit MSB masih dapat memperlihatkan wujud objek di dalam citra tetapi dari Gambar 2.18 (g) hingga 2.6 (i) yang diambil dari bit-bit LSB sudah terlihat seperti citra acak [18].



Gambar 2.18 Delapan *Bitplane* pada citra *cameraman.*

Karena bit – bit MSB menentukan rupa obyek di dalam citra, maka dengan mengubah bit-bit MSB tersebut dihasilkan citra yang sudah tidak dapat dikenali lagi. Jadi, kita cukup memilih hanya bit-bit MSB saja yang dienkripsi sebab dengan hanya engenkripsi bit-bit tersebut maka keseluruhan citra menjadi tidak dapat dikenali lagi. Inilah yang mendasari teknik enkripsi selektif berbasis bit-bit MSB.

Xiang meneliti bahwa untuk memperoleh keseimbangan antara tingkat keamanan dan pertimbangan performansi komputasi, maka enkripsi empat bit MSB (yaitu b_7, b_6, b_5, b_4) merupakan pemilihan yang optimal. Dengan mengenkripsi hanya 4-bit MSB berarti kita hanya perlu mengenkripsi 50% saja dari keseluruhan citra untuk memperoleh citra terenkripsi namun tingkat keamananya tetap terjamin.

4-bit MSB dari setiap pixel dienkripsi seperti *stream cipher* dengan menggunakan operasi XOR sederhana.

$$c_i = p_i \oplus k_i \qquad (4)$$

untuk dekripsi digunakan persamaan kebalikan,

$$p_i = c_i \oplus k_i \qquad (5)$$

pada persamaan (3) dan (4) di atas p_1 adalah 4-bit MSB dari suatu pixel pada *plain image*, c_1 adalah 4-bit MSB suatu pixel pada *cipher image*, dan k_1 adalah *keystream* 4-bit yang dibangkitkan dari *Logistic Map*.

2.2.12. *Object Oriented Programming* (OOP)

Object Oriented Programming atau dikenal dengan Pemograman Berorientasi Objek (PBO) adalah sebuah teknik pemograman yang dalam proses pengembangannya menggunakan terminology objek, di mana setiap objek memiliki atribut beserta dengan fungsi yang dapat saling berinteraksi satu dengan yang lain seperti halnya objek. Ciri inilah yang menjadi ide dasar bagaimana mengembangkan sebuah perangkat lunak yang kompleks dengan menggunakan model objek tersebut. Dengan demikian, proses pengembangan perangkat lunak akan lebih mudah karena hanya akan menyesuaikan model pemograman dengan objek yang akan kita buat.

Pada saat ini metode berorientasi objek banyak digunakan karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari suatu tahap pengembangan ke tahap berikutnya, misalnya pada metode perkembangan terstruktur, jenis aplikasi yang dikembangkan saat ini beragam (aplikasi bisnis, real-time, utility, dana sebagainya) dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Pemograman berorientasi objek memiliki beberapa pilar diantaranya:

1. Class: kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh ‘class of cat’ adalah suatu uni yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari kucing.

2. Objek: membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program computer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program computer berorientasi objek.
3. Abstraksi: kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokuskan pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem tanpa mengungkapkan bagaimana kelebihan ini diterapkan.
4. Encapsulation (Pengkapsulan): merupakan sebuah teknik yang digunakan untuk melakukan penyembunyian informasi yang tidak diperlukan oleh user. Data yang disembunyikan berupa atribut dan fungsi yang saling berhubungan membentuk serangkaian struktur data.
5. Inheritance (Turunan): merupakan salah satu pilar dari OOP yang merepresentasikan pemodelan turunan dari dunia nyata, di mana setiap tipe data baru dapat merupakan turunan dari tipe data induknya atau superclass sehingga kelas turunan akan mewarisi sifat-sifat induknya.
6. Polymorphism: model atau form yang dapat digunakan pada objek yang beragam. Objek-objek yang berbeda yang berasal dari induk yang sama dapat memiliki fungsi yang sama dengan penerapan yang berbeda-beda [19].

2.2.13. Object Relational Mapping

Object Relational Mapping (ORM) adalah suatu teknik pemrograman yang digunakan untuk menghubungkan kode objek dengan *database* relasional. Kode objek ditulis dalam bahasa berorientasi objek pemrograman (OOP) seperti Java atau C#. Konsep ORM adalah melakukan pemetaan dari tabel menjadi objek. Kolom-kolom yang ada pada tabel nantinya akan menjadi variabel-variabel dalam objek tersebut. Satu objek mewakili satu baris. Karena merupakan objek, untuk mengakses beberapa baris sekaligus dapat disamakan dengan mengakses array dari objek tersebut. ORM memiliki kemampuan untuk menciptakan objek *database virtual*, yaitu suatu model *database* yang

direpresentasikan ke dalam sebuah objek pada bahasa pemrograman OOP. Adapun kelebihan yang dimiliki oleh ORM adalah mempercepat pengembangan program. Contohnya yaitu mengurangi perulangan kode *query*, dan memudahkan pemakaian karena tabel yang ada sudah tersusun dalam bentuk objek. Singkatnya ORM ini adalah sebuah konsep untuk menggabungkan metode pemrograman *database* dengan konsep *Object Oriented Programming* (OOP) [20].

2.2.14. Object Oriented Analysis and Design

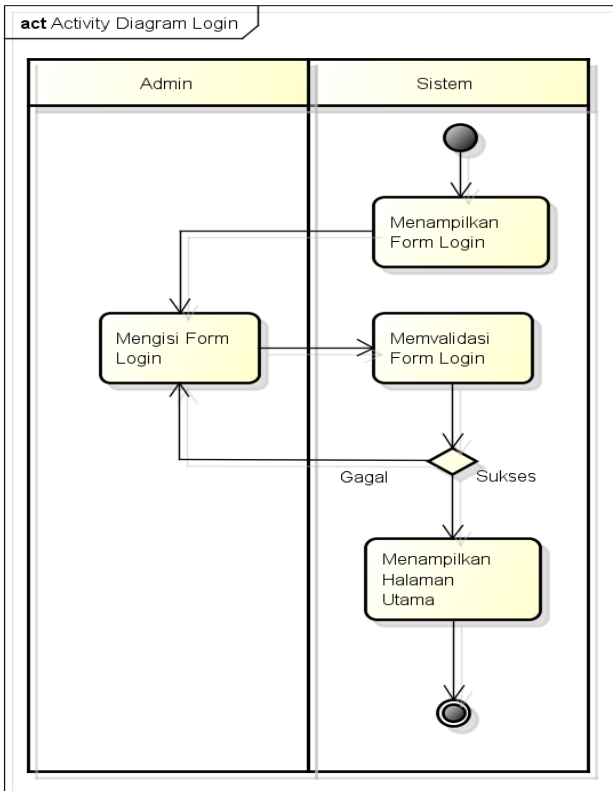
Analisis dan Desain Berorientasi Objek (*Object Oriented Analysis and Design*) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Dasar pembuatannya sendiri adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Alasan mengapa harus memakai metode beorientasi objek yaitu karena perangkat lunak itu sendiri yang bersifat dinamis, di mana hal ini disebabkan karena kebutuhan pengguna berubah dengan cepat. Selain itu bertujuan untuk menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak, karena pada pendekatan berorientasi objek, notasi yang digunakan pada tahap analisis perancangan dan implementasi relatif sama tidak seperti pendekatan konvensional yang dikarenakan notasi yang digunakan pada tahap analisisnya berbeda-beda hal itu menyebabkan transisi antar tahap pengembangan menjadi kompleks. Di samping itu dengan pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata itu sendiri yang sering pengguna lihat adalah objeknya bukan fungsinya. Beda ceritanya dengan pendekatan terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman berorientasi objek menekankan berbagai konsep seperti: *Class, Object, Abstract, Encapsulation, Polymorphism, Inheritance* dan tentunya UML (*Unified Modeling Language*). UML (*Unified Modeling Language*) sendiri merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman berorientasi objek. Selain itu UML merupakan *standard modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem (*developer*) dan

software agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan Testing. Dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented Programming*) [21].

Dokumentasi UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek yang 4 diantaranya adalah [22]:

1. *Activity Diagram*

Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. *Activity Diagram* ini sendiri memiliki struktur diagram yang mirip flowchart atau data flow diagram pada perancangan terstruktur. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*. Contoh activity diagram dapat dilihat pada Gambar 2.19 berikut.



Gambar 2.19 Contoh Activity Diagram Login

2. *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam use case diagram ini sendiri lebih ditekankan kepada apa yang diperbuat sistem dan bagaimana sebuah sistem itu bekerja. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan bentuk dari sebuah pekerjaan tertentu, misalnya login ke dalam sistem, posting, dan sebagainya, sedangkan seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [23].

Adapun komponen-komponen dalam use case diagram diantaranya:

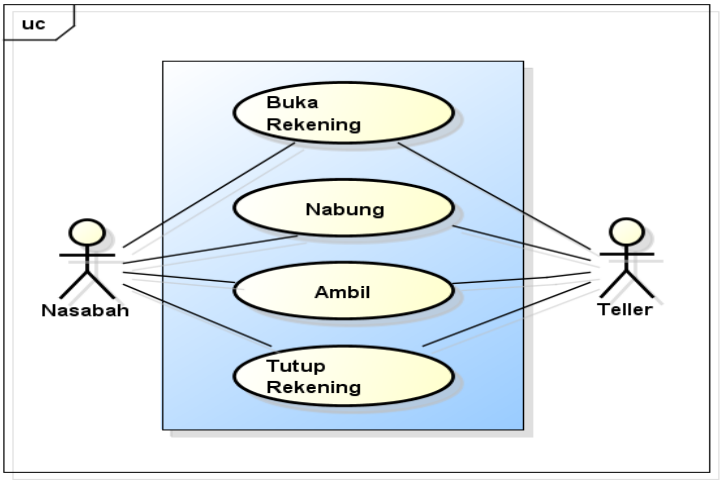
a. Aktor

   Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berada diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut.

b. Use Case

   Use case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada use case, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. Use case menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem.

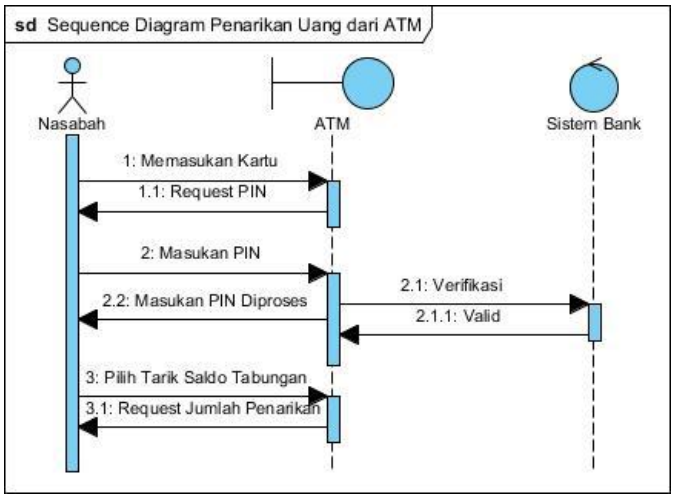
   Contoh use case diagram dapat dilihat pada Gambar 2.20 berikut.



Gambar 2.20 Contoh Use Case Diagram aktivitas nasabah di Bank

3. *Sequence Diagram*

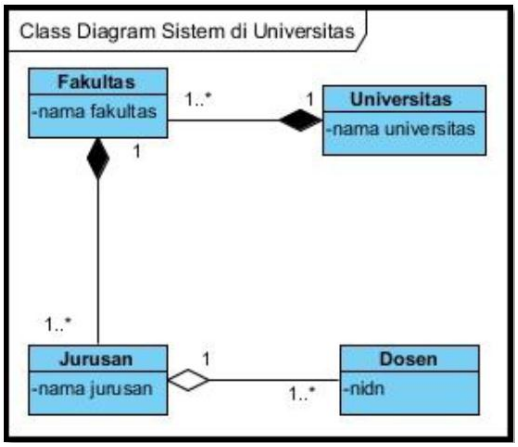
Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram jenis ini memberikan kejelasan seumlah objek dan pesan-pesan yang diletakan diantaranya di dalam sebuah *use case*. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, *message* yang digambarkan dengan garis penuh, dan waktu yang ditunjukan dengan *progress vertical*. Manfaat dari *sequence diagram* adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya. Contoh *sequence diagram* dapat dilihat pada Gambar 2.21 berikut.



Gambar 2.21 Contoh *sequence diagram* penarikan uang dari ATM

4. *Class Diagram*

Class Diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain. *Class diagram* juga menjelaskan hubungan antara *class* serta keseluruhan di dalam sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolabirasi untuk mencapai sebuah tujuan. Contoh *class diagram* dapat dilihat pada Gambar 2.22 berikut.



Gambar 2.22 Contoh *Class Diagram* sistem di universitas

2.2.15. NetBeans IDE 8.0.2

NetBeans adalah Integrated Development Environment (IDE) berbasiskan Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat bejalan di berbagai macam platforms seperti Windows, Linux, Mac OS X and Solaris.

Netbeans merupakan software development yang Open Source, dengan kata lain software ini di bawah pengembangan bersama, bebas biaya NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun Graphic User Interface

(GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger.

The NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. NetBeans IDE mendukung pengembangan semua tipe 16 aplikasi Java (J2SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan refactoring [24].