BAB II

TINJAUAN PUSTAKA

II.1. Media dan Jejaring Sosial

II.1.1. Jejaring Sosial

Situs jejaring sosial yang dalam bahasa Inggris disebut *social network sites* merupakan sebuah *web* berbasis pelayanan yang memungkinkan penggunanya untuk membuat profil, melihat daftar pengguna yang tersedia, serta mengundang atau menerima teman untuk bergabung dalam situs tersebut. Tampilan dasar situs jejaring sosial ini menampilkan halaman profil pengguna, yang di dalamnya terdiri dari identitas diri dan foto pengguna.(Christiany Juditha ; Volume. 13.No. 1, Juni 2011 : 5).

II.1.2. Media Sosial

Menurut (Errika Dwi Setya Watie ; Volume. 3. No. 1, Juli 2011 : 72), *New media* merupakan media yang menawarkan *digitisation, conver-gence, interactiviy, dan development of network* terkait pembuatan pesan dan penyampaian pesannya. Kemampuanya menawarkan interaktifitas ini memungkinkan pengguna dari *new media* memiliki pilihan informasi apa yang dikonsumsi, sekaligus mengendalikan keluaran informasi yang dihasilkan serta melakukan pilihan-pilihan yang diinginkannya. Kemampuan menawarkan suatu *interactivity* inilah yang merupakan konsep sentral dari pemahaman tentang *new media.*

Media sosial/*social media* atau yang dikenal juga dengan jejaring sosial merupakan bagian dari media baru. Jelas kiranya bahwa muatan interaktif dalam media baru sangatlah tinggi. Media sosial, dikutip dari Wikipedia, didefinisikan sebagai sebuah media *online*, dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi, dan menciptakan isi meliputi blog, jejaring sosial, wiki, forum dan dunia virtual. Blog, jejaring sosial dan wiki merupakan bentuk media sosial yang paling umum digunakan oleh masyarakat di seluruh dunia.

II.2 Facebook

Facebook (FB) adalah sebuah situs *web* jejaring sosial populer yang diluncurkan pada 4 Februari 2004. FB didirikan oleh Mark Zuckerberg, seorang mahasiswa Harvard kelahiran 14 Mei 1984 dan mantan murid Ardsley High School. Pada awal masa kuliahnya situs *web* jejaring sosial ini, keanggotaannya masih dibatasi untuk mahasiswa dari Harvard College. Dalam dua bulan selanjutnya, keanggotaannya diperluas ke sekolah lain di wilayah Boston (Boston College, Universitas Boston, MIT, Tufts), Rochester, Stanford, NYU, Northwestern, dan semua sekolah yang termasuk dalam Ivy League. Banyak perguruan tinggi lain yang selanjutnya ditambahkan berturut- turut dalam kurun waktu satu tahun setelah peluncurannya. Akhirnya, orang-orang yang memiliki alamat surat-e suatu universitas (seperti: .edu, .ac, .uk, dll) dari seluruh dunia dapat juga bergabung dengan situs jejaring sosial ini.

Pengguna dapat memilih untuk bergabung dengan satu atau lebih jaringan yang tersedia, seperti berdasar- kan sekolah, tempat kerja, atau wilayah geografis. Hingga Juli 2007, FB memiliki jumlah pengguna terdaftar paling besar di antara

situs-situs yang berfokus pada sekolah dengan lebih dari 34 juta anggota aktif yang dimilikinya dari seluruh dunia. Dari September 2006 hingga September 2007, peringkatnya naik dari posisi ke-60 ke posisi ke-7 situs paling banyak dikunjungi, dan merupakan situs nomor satu untuk foto di Amerika Serikat, mengungguli situs publik lain seperti Flickr, dengan 8,5 juta foto dimuat setiap harinya. Fitur hiburan dalam FB disebut aplikasi. Contohnya antara lain permain-an video, kuis, dan lain sebagainya.

Keistimewaan FB terletak pada fasilitas- nya yang variatif dan cenderung mudah dipelajari. Bahkan kini, FB menjadi *hosting* foto terbesar, mengalahkan situs foto seperti Flickr atau Picasso. Lebih dari sekadar mencari teman dan memasukkannya dalam friendlist, situs ini bisa menawarkan lebih dari itu. Sharing untuk media seperti audio, video, foto, dan *notes*, merupakan salah satu wujud kebebasan yang memungkinkan siapa saja dapat mengunggah apa saja dengan segala resiko yang juga ada. Sedang untuk jaminan keamanannya bisa diatur untuk foto dan profil dalam privacy setting. (Christiany Juditha ; Volume. 13. No. 1, Juni 2011 : 6)

II.3. Keamanan Data

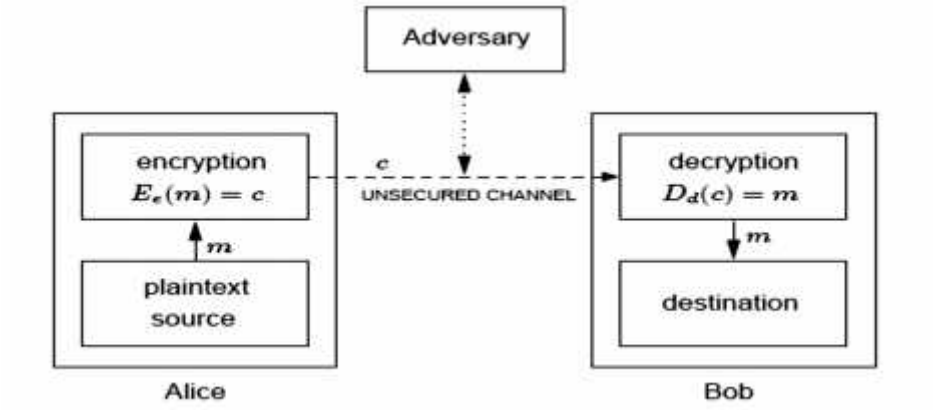
Kemanan data merupakan hal yang sangat penting dalam bidang bisnis komersial dan tradisional saat ini, contohnya penggunaan media pengiriman data elektronik melalui *e-mail* dan media lainnya yang sering digunakan dalam dunia bisnis. Data yang dikirm melalui email memiliki variasi, ada yang bersifat umum dan rahasia. Berdasarkan kondisi dan kenyataan tersebut, para ahli dan peneliti dibidang IT mengupayakan berbagai cara yaitu dengan membuat sistem

keamanan data. Salah satu metode keamanan data adalah metode kriptografi yang dapat digunakan untuk mengamankan data yang bersifat rahasia agar data tersebut tidak diketahui oleh orang lain yang tidak berkepentingan. (Budi Triandi ; Vol. 2 No. 2, Oktober 2013 : 124)

II.4. Kriptografi

Kriptografi (*cryptography*) berasal dari dua kata dalam Bahasa Yunani, yaitu "*cryptos*" yang berarti rahasia, dan "*graphein*" yang berarti tulisan. Kriptografi adalah ilmu mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi. Secara umum, kriptografi terdiri dua proses utama, yaitu enkripsi dan dekripsi. Proses enkripsi akan mengubah pesan asli (*plainteks*) menjadi pesan terenkripsi dengan menggunakan algoritma dankunci tertentu yang tidak dapat dibaca secaralangsung (*cipherteks*). Proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu proses untuk memperoleh kembali *plainteks* dari *cipherteks* menggunakan kunci dan algoritma tertentu. (Azanuddin ; Volume IV. No. 1, Agustus 2013 : 49)

Secara sederhana istilah-istilah diatas dapat digambarkan sebagai berikut:



Gambar:II.1 Proses Enkripsi/Dekripsi Sederhana
(Azanuddin ; Volume IV. No. 1, Agustus 2013 : 49)

II.4.1. Algoritma Kriptografi

Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua jenis yaitu algoritma simetrik (*symmetric*) dan asimetrik (*asymmetric*).

II.4.1.1. Algoritma Simetrik

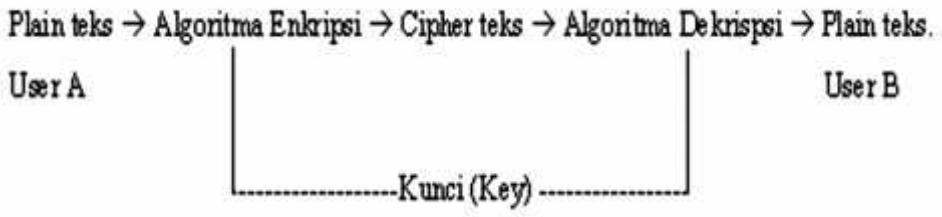
Algoritma simetris adalah salah satu jrnis kunci pada algoritma kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Istilah lain untuk kriptografi kunci simetri adalah kriptografi kunci privat (*private-key cryptography*). Sistem kriptografi kunci-simetri diasumsikan sebagai pengirim dan penerima pesan yang sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kuncinya.

Kriptografi simetri adalah jenis kriptografi yang diketahui masuk ke dalam catatan sejarah hingga tahun 1976. Semua algoritma kriptografi klasik termasuk

ke dalam sistem kriptografi simetri. Salah satu kelebihan pada algoritma simetris yaitu proses enkripsi dan deskripsinya jauh lebih cepat dibandingkan dengan algoritma asimetris. Sedangkan kelemahannya yaitu pada permasalahan distribusi kunci (*key distribution*).

Seperti yang telah dibahas sebelumnya, proses enkripsi dan deskripsi pada kriptografi simetri menggunakan kunci yang sama. Sehingga timbul persoalan untuk menjaga kerahasian kunci. Contohnya pada saat pengiriman kunci dilakukan melalui media yang tidak aman seperti internet. Jika kunci ini hilang atau sudah diketahui oleh orang yang tidak berhak, maka kriptosistem ini dinyatakan tidak aman lagi. Kelemahan lain adalah masalah efisiensi jumlah kunci. Jika terdapat *n* user, maka diperlukan $n(n-1)/2$ kunci, sehingga untuk jumlah user yang sangat banyak, sistem ini tidak efisien lagi.(Anandia Zelvina, dkk ; Volume. 1. No. 1, 2012 : 57)

Proses enkripsi dan dekripsi dengan algoritma simetrik dapat digambarkan sebagai berikut :



Gambar:II.2 Enkripsi dan Dekripsi Algoritma Simetrik
(Sumber: Munawar ; Volume. 1. Edisi. 1, Maret 2012 : 12)

Berikut ini algoritma yang memakai kunci simetri diantaranya:

a. Data Encryption Standard (DES),

b. RC2, RC4, RC5, RC6,

c. *International Data Encryption Algorithm* (IDEA),

d. *Advanced Encryption Standard* (AES),

e. *One Time Pad* (OTP),

f. A5, dan lain sebagainya.

II.4.1.2. Algoritma Asimetris

Algoritma asimetris atau dapat disebut juga dengan algoritma kunci *public*, didesain sebaik mungkin sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci dekripsinya. Dimana kunci untuk enkripsi tidak rahasia (diumumkan ke publik), sementara kunci dekripsinya bersifat rahasia (hanya diketahui oleh penerima pesan).

Pada kriptografi asimetris, setiap orang yang akan berkomunikasi harus mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim pesan akan mengenkripsi pesan menggunakan kunci publik si penerima pesan dan hanya penerima pesan yang dapat mendekripsi pesan tersebut karena hanya ia yang mengetahui kunci privatnya sendiri.

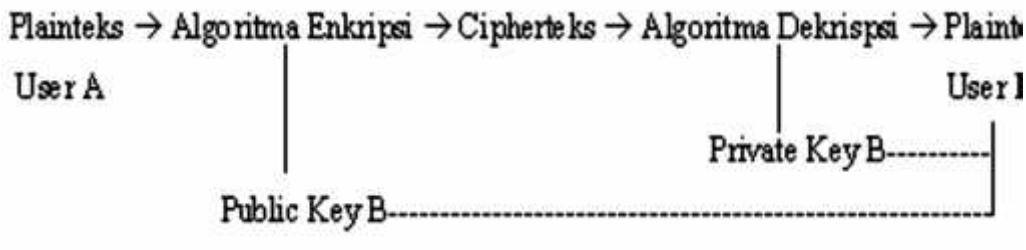
Kriptografi kunci-publik dapat dianalogikan seperti kotak surat yang terkunci dan memiliki lubang untuk memasukkan surat. Setiap orang dapat memasukkan surat ke dalam kotak surat tersebut, tetapi hanya pemilik kotak yang dapat membuka kotak dan membaca surat di dalamnya karena ia yang memiliki kunci. Sistem ini memiliki dua keuntungan. Yang pertama yaitu, tidak ada kebutuhan untuk mendistribusikan kunci privat sebagaimana pada sistem kriptografi simetri. Kunci publik dapat dikirim ke penerima pesan melalui saluran

yang sama dengan saluran yang digunakan untuk mengirim pesan. Saluran untuk mengirim pesan umumnya tidak aman.

Kedua, jumlah kunci yang digunakan untuk berkomunikasi secara rahasia dengan banyak orang tidak perlu sebanyak jumlah orang tersebut, cukup membuat dua buah kunci, yaitu kunci publik bagi para koresponden untuk mengenkripsi pesan, dan kunci privat untuk mendekripsi pesan. Berbeda dengan kriptografi kunci-simetris yang membuat kunci sebanyak jumlah pihak yang diajak berkorespondensi.

Meski masih terbilang baru (sejak 1976), kriptografi kunci-publik mempunyai kontribusi yang luar biasa dibandingkan dengan sistem kriptografi simetri. Kontribusi yang paling penting adalah tanda-tangan digital pada pesan untuk memberikan aspek keamanan otentikasi, integritas data, dan nirpenyangkalan. Tanda-tangan digital adalah nilai kriptografis yang bergantung pada isi pesan dan kunci yang digunakan. Pengirim pesan mengenkripsi pesan (yang sudah diringkas) dengan kunci privatnya, hasil enkripsi inilah yang dinamakan tanda-tangan digital. Tanda-tangan digital dilekatkan (embed) pada pesan asli. Penerima pesan memverifikasi tanda-tangan digital dengan menggunaklan kunci publik.(Anandia Zelvina, dkk ; Volume. 1. No. 1, 2012 : 58)

Proses enkripsi dan dekripsi dengan algoritma asimetrik dapa digambarkan sebagai berikut :



Gambar:II.3 Enkripsi dan Dekripsi Algoritma Asimetri
(Sumber: Munawar ; Volume. 1. Edisi. 1, Maret 2012 : 12)

Berikut ini algoritma yang memakai kunci asimetrik diantaranya adalah:

a. *Digital Signature Algorithm (DSA)*

b. *RSA*,

c. *Elliptic Curve Crtpyography (ECC)*

d. Kriptografi Quantum, dan lain sebagainya.

II.4.2. Algoritma *Gronsfeld Cipher*

Salah satu *cipher* substitusi sederhana *polyalphabetic* adalah *Gronsfeld*. Gaspar Schot adalah seorang kriptografer abad ke-17 di Jerman, yang belajar cipher ini selama perjalanan antara Mainz dan Frankfurt dengan menghitung Gronsfeld,maka terciptalah nama dari cipher tersebut yaitu Gronsfeld**.** (Azanuddin ; Volume. IV. No. 1, Agustus 2013 : 49)

Gronsfeld identik dengan *vigenere cipher*, tetapi perbedaan nya adalah pada kunci cipher ini yang menggunakan angka 0-9. Kelebihan *gronsfeld cipher* adalah kuncinya bukan kata, tetapi kelemahannya cipher ini hanya mempunyai 10

kemungkinan, tidak seperti *vigenere* yang memiliki 26 kemungkinan karena menggunakan alfabet. *Gronsfeld* sendiri umumnya lebih banyak digunakan di Jerman dan beberapa negara lain di Eropa.

Berikut ini adalah gambaran dari proses algoritma *Gronsfeld cipher*, dimana *gronsfeld chiper* ini memiliki cara kerja seperti vigenere, namun menggunakan kunci terbatas yaitu dalam bentuk angka.

Ct=K+pt [cipher = key +plaintext]

Keterangan: CT = Ciphertext

K = Key (Kunci)

Pt = Plaintext

Dimana : *Ct*= nilai desimal karakter *ciphertext*

K = nilai desimal karakter *key*

Pt= nilai desimal karakter *plaintext*

Tabel:II.1. Tabel Gronsfeld Cipher Sederhana

Teks → 0 1 2 3 4 5 6 7 8 9	
0	**0 1 2 3 4 5 6 7 8 9**
1	**1 2 3 4 5 6 7 8 9 0**
2	**2 3 4 5 6 7 8 9 0 1**
3	**3 4 5 6 7 8 9 0 1 2**
4	**4 5 6 7 8 9 0 1 2 3**
5	**5 6 7 8 9 0 1 2 3 4**
6	**6 7 8 9 0 1 2 3 4 5**
7	**7 8 9 0 1 2 3 4 5 6**

(Sumber : Azanuddin ; Volume IV. No. 1, Agustus 2013 : 50)

II.4.2.1. Proses Enkripsi *Gronsfeld Cipher*

Untuk mengenkripsi, hanya menambahkan jumlah huruf yang akan dienkripsi sesuai dengan jumlah kunci tetapi terlebih dahulu pesan tersebut diubah ke kode nilai desimal, plainteks yang dihasilkan akan menjadi chiperteks.

Langkah-langkah proses enkripsi adalah sebagai berikut :

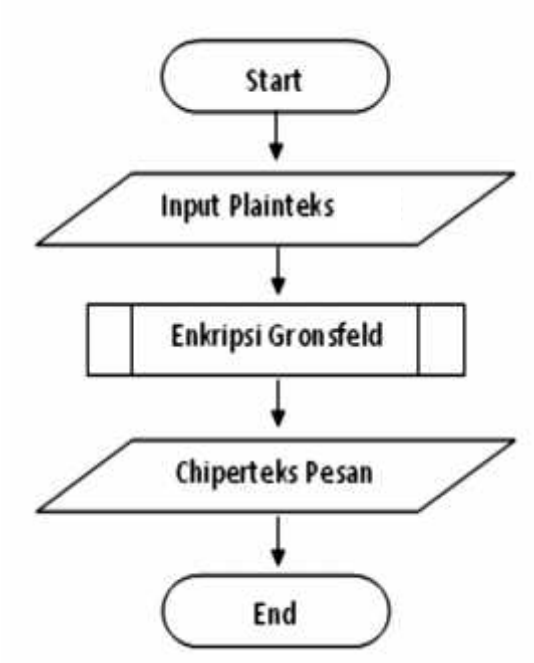
1. Tentukan plainteks yang akan dienkripsi beserta kunci.
2. Jika panjang kunci tidak sama dengan panjang plainteks maka kunci yang ada diulang secara priodik sehingga jumlah karakter kuncinya sama dengan jumlah plainteks nya.
3. Selanjutnya ubah plainteks ke bentuk nilai desimal kemudian ditambahkan dengan kunci.Jika penambahan lebih besar dari jumlah mod, maka diambil nilai sisa hasil bagi nya.
4. Setelah dijumlahkan dengan kunci maka langkah berikutnya adalah mengubah kembali ke bentuk karakter.

Tabel:II.2 Proses Enkrispi

Plainteks	G	R	O	N	S	F	E	L	D
Desimal plainteks	71	82	79	78	83	70	69	76	68
Kunci	7	3	4	1	7	3	4	1	7
Desimal Enkripsi	78	85	83	79	90	73	73	77	75
Cipherteks	N	U	S	O	Z	I	I	M	K

(Sumber : Azanuddin ; Volume IV. No. 1, Agustus 2013 : 50)

Proses kerja enkripsi dapat digambarkan seperti *flowchart* pada gambar berikut :



Gambar: II.4 *Flowchart* Enkripsi Gronsfeld
(Sumber : Azanuddin ; Volume IV. No. 1, Agustus 2013 : 53)

II.4.2.2. Proses Dekripsi

Dekripsi adalah proses sebaliknya, dimana chiperteks nya diubah menjadi nilai desimal dan dikurangi dengan jumlah kunci kemudian dikembalikan ke karakter.

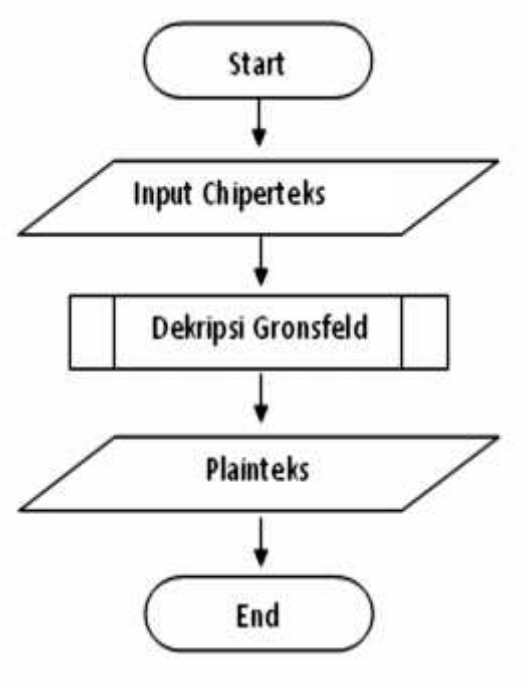
Langkah-langkah proses dekripsi adalah sebagai berikut :

1. Terlebih dahulu mengubah chiperteks ke nilai
   desimal.
2. Kemudian nilai desimal chiperteks nya dikurangi sesuai dengan kunci
3. Setelah dikurangi dengan kunci maka langkah berikutnya adalah mengubah kembali kebentuk karakter.

Tabel:II.3 Proses Dekripsi

Cipherteks	N	U	S	O	Z	I	I	M	K
Desimal CipherTeks	78	85	83	79	90	73	73	77	75
Kunci	7	3	4	1	7	3	4	1	7
Plainteks	G	R	O	N	S	F	E	L	D
Desimal plainteks	71	82	79	78	83	70	69	76	68

(Sumber : Azanuddin ; Volume IV. No. 1, Agustus 2013 : 50)



Gambar: II.5 *Flowchart* Dekripsi Gronsfeld
(Sumber : Azanuddin ; Volume IV. No. 1, Agustus 2013 : 55)

II.4.3. Algoritma *Rivest Code* 4 (RC4)

Algoritma RC4 mengenkripsi dengan mengombinasikannya dengan plainteks dengan menggunakan bit-wise Xor (Exclusive-or). RC4 menggunakan panjang kunci dari 1 sampai 256 byte yang digunakan untuk menginisialisasikan tabel sepanjang 256 byte. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan plaintext untuk menghasilkan *ciphertext*. Masing - masing elemen dalam tabel saling ditukarkan minimal sekali. Proses dekripsinya dilakukan dengan cara yang sama (karena Xor merupakan fungsi simetrik).

Untuk menghasilkan *keystream*, *ciphe*r menggunakan *state internal* yang meliputi dua bagian :

1. Tahap *key scheduling* dimana diberi nilai awal berdasarkan kunci enkripsi.State yang diberi nilai awal berupa array yang merepresentasikan suatu permutasi dengan 256 elemen, jadi hasil dari algoritma KSA adalah permutasi awal. Array yang mempunyai 256 elemen ini (dengan indeks 0 sampai dengan 255) dinamakan S. Berikut adalah algoritma KSA dalam bentuk *pseudo-code* dimana key adalah kunci enkripsi dan *keylength* adalah besar kunci enkripsi dalam *bytes* (untuk kunci 128 bit, *keylength* = 16).

```
for i = 0 to 255
  S[i] := i
j := 0
for i = 0 to 255
  j := (j + S[i] + key[i mod keylength]) mod 256
  swap(S[i],S[j])
```

2. Tahap *pseudo-random generation* dimana *state automaton* beroperasi dan outputnya menghasilkan *keystream.* Setiap putaran, bagian *keystream* sebesar 1 *byte* (dengan nilai antara 0 sampai dengan 255) dioutput oleh PRGA berdasarkan state S. Berikut adalah algoritma PRGA dalam bentuk pseudo-code:

```
i := 0
j := 0
loop
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap(S[i],S[j])
  output S[(S[i] + S[j]) mod 256]
```

Setelah terbentuk *keystream*, kemudian *keystream* tersebut dimasukkan dalam operasi XOR dengan *plaintext* yang ada, dengan sebelumnya pesan dipotong-potong terlebih dahulu menjadi *byte-byte*.(Okie Setiawan, dkk ; Pros iding SNATIF. No.1, 2014 : 117)

II.4.3.1. Keamanan sistem sandi RC4

Menurut Rifki Sadikin (2012 : 214), Ukuran sandi *RC4* sangat berpengaruh terhadap keamanan sistem Sandi *RC4. RC4* telah dibuktikan tidak aman untuk ukuran kunci yang kecil, yaitu dibawah 5 *byte*. Rekomendasi penggunaan sistem sandi *RC4* agar memiliki keamanan yang kuat adalah :

1. Ukuran kunci sama atau lebih besar daripada 256 *bit* (16 *byte*)
2. Setiap sesi baru membangkitkan kunci yang baru(dengan pembangkitkan kunci yang baru menghingdari penyerang untuk melakukan analisis sandi difernsial pada sistem sandi).

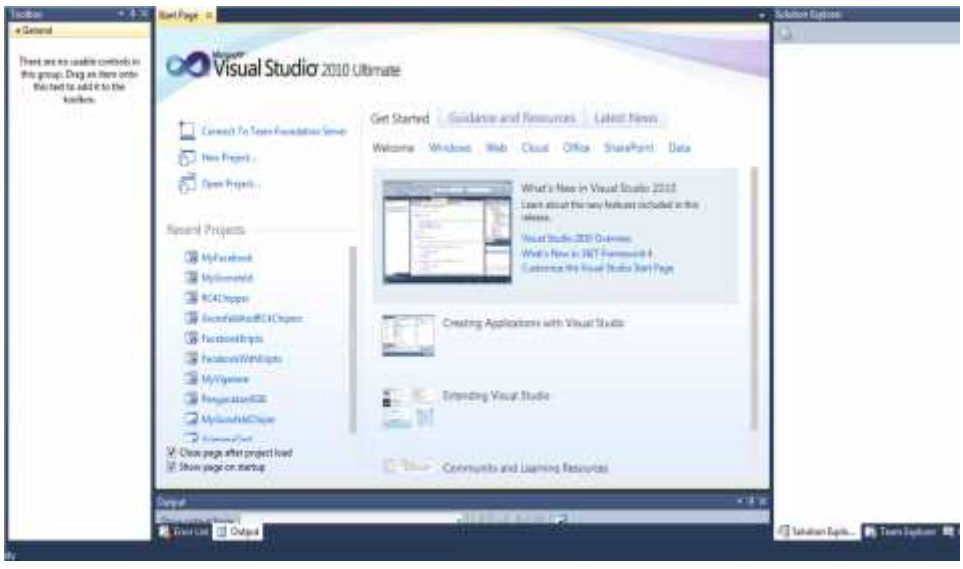
II.5. Aplikasi

Aplikasi atau juga disebut program aplikasi adalah program yang dibuat oleh pemakai yang ditujukan hanya untuk melakukan suatu tugas khusus.(Suriski Sitinjak, dkk ; SEMNASIF, 2010)

II.6. Bahasa Pemrograman *Visual Basic .Net*

Visual Basic merupakan pengembangan dari *BASIC* yang dibuat sebagai bahasa pemrograman yang mudah digunakan. *Visual Basic* memungkinkan proses *Rapid Application Development* (RAD) dari aplikasi antarmuka, mengakses *database*, dan membuat kontrol dan objek. (Jubilee Enterprise ; 2015 : 2).

Berikut adalah gambar tampilan awal dari Aplikasi *Visual BASIC.Net 2010* :



Gambar:II.6. Tampilan Awal *Visual BASIC.Net* 2010
(Sumber : Jubilee Enterprise ; 2015)

II.6. UML *(Unified Modelling Language)*

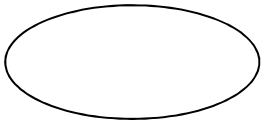
Penjadwalan telah lama diteliti, contohnya dalam penghasilan tenaga oleh *Windu Gata* dan *Grace Gata* (2013), UML *(Unified Modelling Language)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML *(Unified Modelling Language)* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

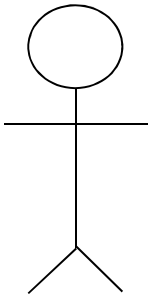
II.6.1. *UseCase Diagram*

UseCase Diagram merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *UseCase Diagram*, yaitu:

Tabel II.4 *Diagram UseCase*

Gambar	Keterangan
	*UseCase* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama *UseCase*.
	Actor atau Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor,

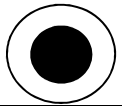
	harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *usecase*, tetapi tidak memiliki kontrol terhadap *usecase*.
	Asosiasi antara aktor dan *usecase*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan *usecase* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

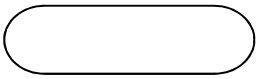
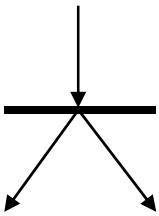
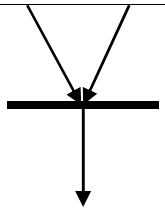
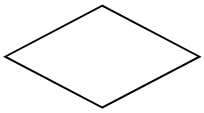
(Sumber : Windu Gata dan Grace Gata ; 2013)

II.6.2. *Activity Diagram* (Diagram Aktivitas)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

Tabel II.5 *Diagram Aktivitas*

Gambar	Keterangan
	*Start Point* diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End Point*, akhir aktifitas.

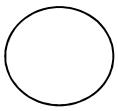
	*Activities*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	*Join* (penggabungan) atau *Rake*, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true* atau *false*.

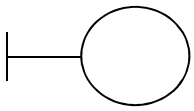
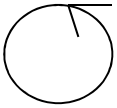
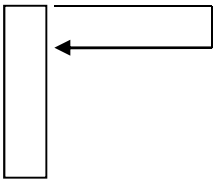
(Sumber : Windu Gata dan Grace Gata ; 2013)

II.6.3. *Sequence Diagram* (Diagram Urutan)

Sequence Diagram menggambarkan kelakuan objek pada *usecase* dengan mendekripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

Tabel II.6 *Diagram Urutan*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *formentry* dan *form* cetak.
	*Control Class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. Control *object* mengkoordinir pesan antara *boundary* dengan entitas.
	Message, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Windu Gata dan Grace Gata ; 2013)