Rancang Bangun Sistem Pengamanan Dokumen Pada Sistem Informasi Akademik Menggunakan *Digital Signature* dengan Algoritma Kurva Eliptik

Tesis

Untuk Memenuhi Sebagian Persyaratan Mencapai Derajat Sarjana S2
Program Studi Magister Sistem Informasi

oleh:

Ahmaddul Hadi

J4F009003

PROGRAM PASCASRAJANA
UNIVERSITAS DIPONEGORO
SEMARANG
2011

ABSTRAK

Pengamanan dokumen pada setiap lembaran informasi Sistem Informasi Akademik (SIA) UNP dengan menambahkan tanda tangan digital. Penggunaan tanda tangan digital (*Digital Signature*) kurva eliptik didasarkan atas *Elliptic Curve Discrete Logarithm Problem* (ECDLP) pada kurva eliptik modulo prima memiliki tingkat keamanan yang tinggi.

Aplikasi SIA pada penelitian ini menghasilkan output dokumen tercetak yang telah ditambahkan (*embedded*) dengan tanda tangan. Tanda tangan berupa Informasi (*plaintext*) yang di-enkripsi pada proses *signing* yaitu NIM, IP, jenis dokumen dan waktu cetak dimana ke empat variabel ini di-enkripsi dengan algoritma kurva eliptik pada bidang terbatas F_p^m dan menghasilkan kunci *public r* yang tersimpan pada sebuah tabel database, kunci *privat*e *s* serta *chipertext* (*ds code*). Pada aplikasi pengujian tandatangan (*verifying*) dengan mendekripsikan *chipertext (ds code)* yang diinputkan, jika nilai kunci publik dan ke empat variabel cocok maka ditampilkan informasi valid dari dokumen. Pengujian tingkat keamanan dan kehandalan tanda tangan dengan menggunakan program *sniffing wireshark* dan *chain & abel*. Dan pengujian kinerja laman web dengan menggunakan aplikasi *firebug*.

Penelitian tesis ini menghasilkan aplikasi SIA yang telah ditambahkan tanda tangan dan aplikasi pembaca keabsahan tanda tangan. Dari hasil uji coba tanda tangan tidak dapat dihacking dan dicracking dengan aplikasi pengendus, serta aplikasi *perifying* menunjukkan waktu akses rata-rata 110 ms. Aplikasi web *verifying* membutuhkan waktu yang lama untuk mendekripsikan digital signature, tetapi ini sebanding dengan keamanan dan kehandalan yang dihasilkan oleh sistim informasi dengan algoritma kurva eliptik ini.

Kata Kunci : Sistem Informasi Akademik (SIA), algoritma kurva eliptik, kunci publik, kunci private.

ABSTRACT

Improved document security on any information output at *Sistem Informasi Akademik (SIA)* of UNP is by adding a digital signature. The use of elliptic curve digital signature based on Elliptic Curve Discrete logarithm problem (ECDLP) on the elliptic curve modulo a prime has a high level of security.

SIA application in this thesis produce printed output of documents that have been added (embedded) with signature. The Signature of Information (Plaintext) which are encrypted in the process of signing the "NIM, IP, type of document and print time" where all four of these variables are encrypted with the elliptic curve algorithms in Finite Fields F_P^m and generate the *public key r* stored in a database table, *a private key s* and the *ciphertext* (*ds code*). In the test application signatures (verifying) with decrypt the ciphertext (ds code) is entered, if the public key value and the four variables is match than the application will be displayed information from the document valid. Testing the level of security and reliability of the signature using a sniffing program *Wireshark* and Chain & Abel. And performance testing web pages using firebug applications.

This thesis research resulted in the application "SIA" which was added a signature and the signature validity reader application. From the test results, the signature can not be hacked and cracked with application-sniffing, and the perifying application is show the average access time of 110 ms. The Verifying web application takes a long time to decrypt the digital signature, but is comparable to the security and reliability of information generated by the system with this elliptic curve algorithms.

Keywords : Sistem Informasi Akademik (SIA), elliptic curve algorithms, public key, private key.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Berbagai peralatan teknologi informasi telah berkembang dengan pesatnya, dan hampir segala lini kehidupan manusia menggunakan peralatan teknologi informasi. Dibidang Pendidikan, penerapan teknologi informasi telah memudahkan dan meningkatkan peran serta mutu dari sebuah lembaga pendidikan. Dengan terkoneksinya berbagai aplikasi pada sistem informasi di Perguruan Tinggi ke jaringan global (internet) dan implementasi penggunaannya oleh segenap *stake holder* merupakan salah satu indikator penilaian Akreditasi dan *Word Class University* (WCU) yang sedang dikembangkan oleh Dirjen Pendidikan Tinggi (DIKTI).

Berbagai aplikasi sistem informasi dibidang pendidikan, antara lain: Sistem Informasi Akademik (SIA), *e-learning*, *e-library*, *e-assesment*, *e-tutor*, portal pendidikan dan berbagai aplikasi lainnya memberikan kemudahan kepada *stake holder* (mahasiswa, dosen, staff administrasi, eksekutif dan bagian luar kampus) di sebuah Perguruan Tinggi untuk melaksanakan Tridharma Perguruan Tinggi dan meningkatkan kualitas pembelajaran dan pelayanan. SIA yang telah banyak diimplementasikan berbagai kampus di Indonesia telah memberikan dampak kemudahan dari berbagai kampus untuk mengelola administrasi kegiatan akademik mahasiswa dan kampus, seperti informasi data mahasiswa, nilai, informasi jadwal, materi kuliah dari setiap dosen pengajar dan beberapa informasi lainnya.

Setiap aplikasi pada Sistem Informasi Akademik memerlukan media dokumentasi tercetak baik yang menggunakan kertas (*paper*) maupun tidak menggunakan kertas (*paperless*) atau biasa juga dikenal dengan istilah *e-paper*. Penggunaan *e-paper* atau lembaran/dokumen digital yang digunakan untuk setiap lembar naskah yang dicetak dari SIA baik oleh mahasiswa, staf administrasi maupun pihak lainnya rentan terhadap pemalsuan dan

pembajakan oleh pihak-pihak yang tidak bertanggungjawab. Berbagai kasus seperti pemalsuan ijazah, manipulasi dan pemalsuan lembaran nilai oleh mahasiswa sering dilakukan. Beberapa kasus tersebut telah ditangani oleh pihak berwajib dan dipidana (Koran tempo edisi 4 Juni 2010).

Berbagai pengamanan terhadap berkas elektronik (*e-paper)* telah banyak dilakukan, seperti pengamanan berkas dengan *watermarking* (Chao-Yong Hsu: 2005). Pengamanan lain pada berkas elektronik seperti dengan memberikan tanda tangan elektronik atau *digital signature* untuk *e-voting (ÃLukasz Nitschke*: 2008). Pengamanan secara konvensional seperti yang dilakukan di Universitas Negeri Padang (UNP) dan kebanyakan Perguruan Tinggi lain di Indonesia dilakukan dengan melegalisasi setiap lembaran yang telah dicetak dan ditanda tangani oleh pejabat yang berwenang serta distempel. Beberapa jenis pengamanan ini masih rentan terhadap pemalsuan dan kurang efektif dengan jenis aplikasi SIA yang banyak diimplementasikan di berbagai Perguruan Tinggi.

Dari beberapa kasus yang ada pada *e-paper* (berkas elektronik) dalam Sistem Informasi Akademik (SIA), dan dari beberapa penelitian yang telah dilaksanakan, maka pada penelitian tesis ini akan dibahas tentang bagaimana mengamankan berkas elektronik (*e-paper*) dengan menambahkan (meng-*embedded*) *digital signature* pada setiap berkas yang akan dicetak baik yang menggunakan kertas (*paper*) maupun yang tidak menggunakan kertas (*paperless*) dalam format PDF maupun format digital lainnya. Tanda tangan digital (*digital signature*) akan dikonversikan dalam model ***barcode*** dari nilai hasil enkripsi *digital signature* dengan metode kurva eliptik, dan *barcode* ini akan dibaca kembali oleh *barcode reader* dengan menterjemahkan informasi yang ada pada *barcode* menjadi informasi yang dimengerti bagi yang membaca berkas.

1.2. Perumusan Masalah

Dari permasalah di atas maka pada tesis ini dirumuskan beberapa permasalahan berikut ini:

1. Bagaimana merancang pengamanan dokumen dengan menggunakan *digital signature* dan algoritma kurva eliptik pada aplikasi sistem informasi akademik (*signing*).
2. Bagaimana merancang sistem aplikasi pengamanan dokumen yang bersifat tambahan dan dapat diintegrasikan pada Sistem Informasi Akademik (SIA) yang telah ada.
3. Bagaimana merancang sistem aplikasi pembaca *barcode* dengan aplikasi dekripsi yang menggunakan algoritma sama pada proses enkripsi (*perifying*).

1.3. Batasan Masalah

Pada perancangan aplikasi pengamanan pada penelitian ini dibatasi sebagai berikut:

1. Studi kasus pada Sistem Informasi Akademik (SIA) Universitas Negeri Padang (UNP, http://portal.unp.ac.id/).
2. Disain pengamanan berkas elektronik dengan menggunakan tanda tangan digital (*digital signature*) dan menggunakan algoritma kurva eliptik.
3. Aplikasi yang dirancang akan bersifat tambahan pada sistem yang telah ada sehingga tidak mengganggu jalannya sistem.
4. Aplikasi *signing* akan berjalan hanya pada saat user akan menjalankan perintah *Print*, pada menu cetak (print) di aplikasi.

1.4. Keaslian Penelitian

Pada saat ini telah banyak aplikasi pengamanan berkas pada *e-paper* seperti yang dilakukan oleh ***Chao-Yong Hsu*** dan ***Chun-Shien Lu*** (2005). Penelitian ini mencari perbedaan berbagai model tipe kompresi terhadap kualitas warna dan mencari warna yang paling efektif untuk watermarking sebagai tanda pengamanan untuk berkas elektronik.

Selanjutnya penelitian yang dilakukan oleh ***El-Affendi*** (2008) yang meneliti tentang penerapan *watermarking* untuk mengamankan tanda tangan fisik yang disertai dengan

stempel dalam aplikasi *e-government*. Setiap berkas/surat elektronik yang telah ditanda tangani oleh pihak yang berwenang dan distempel, pada setiap tanda tangan dan stempelnya diselipkan kriptografi *watermarking*.

Lukasz Nitschke (2008) dalam penelitiannya berjudul "Remote Voting Using Smart Cards With Display", mengamati model-model keamanan yang telah ada serta gangguannya untuk diimplementasikan pada aplikasi pemilihan umum (pemilu) online sebagai jawaban dari tuntutan demokrasi modern yang menginginkan digitalisasi pada pemilu. Hasil yang didapatkan adalah dengan mengkombinasikan beberapa model keamanan seperti menggunakan saluran khusus dengan keamanan protokol yang tinggi, *e-paper* dengan sebuah *digital signature* dan sebuah smart card yang telah dimiliki oleh masing-masing pemilih (KTP digital).

Pada penelitian tesis ini dibahas tentang pengamanan sistem informasi yaitu pengamanan hasil output dari Sistem Informasi Akademik (SIA) Universitas Negeri Padang dengan menambahkan tanda tangan digital (*digital signature*) pada setiap lembaran tercetak. Tanda tangan digital menggunakan algoritma kurva eliptik dengan tampilan berupa *barcode*. Pada penelitian El-Efendi (2008) dilakukan pengamanan berkas elektronik dengan tanda tangan dan stempel yang telah di *scanner* kemudian diselipkan sebuah kriptografi watermarking. Pada penelitian yang dilakukan oleh ***Chao-Yong Hsu*** dan ***Chun-Shien Lu*** (2005) menggunakan watermarking untuk mengamankan berkas elektronik dengan mencari berbagai tipe kompresi dari beberapa kualitas warna. Sedangkan Penelitian yang dilakukan oleh ***Lukasz Nitschke*** (2008) yaitu pengamanan untuk e-voting, dengan mengkombinasikan beberapa model keamanan seperti menggunakan saluran khusus dengan keamanan protokol yang tinggi, *e-paper* dengan sebuah digital signature dan sebuah smart card yang telah dimiliki oleh masing-masing pemilih (KTP digital).

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah menghasilkan suatu aplikasi pengamanan berkas elektronik dengan *digital signature* pada sistem informasi akademik yang merupakan aplikasi tambahan dari aplikasi yang telah ada. Model aplikasi ini juga dapat diimplementasikan pada sistem informasi lainnya, tidak hanya terbatas SIA seperti surat elektronik, legalisir elektronik, dan lainnya. Selain itu juga menghasilkan aplikasi pembaca kembali (aplikasi dekripsi).

1.6. Tujuan Penelitian

Penelitian ini bertujuan untuk merancang aplikasi keamanan dokumen elektronik pada Sistem Informasi Akademik (SIA) menggunakan *digital signature* dengan algoritma kurva eliptik, dengan menghasilkan sebuah kode yang telah di-enkripsi kemudian dikonversikan menjadi *barcode*. Aplikasi dekripsi (program *perifying*) yang akan me-dekripsikan kembali kode pada *barcode* menjadi informasi yang dimengerti oleh pengguna.

BAB II

TINJAUAN PUSTAKA

2.1. TINJAUAN PUSTAKA

Menurut Rinaldi Munir (2006), Integritas berkas perangkat lunak berkaitan dengan keaslian berkas program, keutuhan, dan keabsahan pengembang perangkat lunak sangat rentan pada transaksi di internet. Berkas program dapat dimodifikasi oleh pihak ketiga (menjadi tidak asli) atau mengalami kerusakan (*corrupt*) oleh virus atau gangguan selama transmisi dari komputer *server* ke komputer *client* (menjadi tidak utuh). Selain itu, pengguna perangkat lunak perlu memastikan bahwa program yang ia *download* dibuat oleh pengembang program yang sah, dan bukan pengembang lain yang menyamar sebagai pengembang program yang asli. Masalah integritas berkas perangkat lunak ini dapat diselesaikan dengan menggunakan tanda tangan digital. Tanda tangan digital dibangkitkan dengan algoritma kriptografi kunci-publik. Tanda tangan digital bergantung pada isi berkas program dan kunci pengembang perangkat lunak. Melalui proses verifikasi, pengguna dapat membuktikan integritas berkas perangkat lunak yang ia *downlaod* dari situs web pengembang.

Lukasz Nitschke (2008) dalam penelitiannya berjudul "*Remote Voting Using Smart Cards With Display*", mengamati model-model keamanan yang telah ada serta gangguannya untuk diimplementasikan pada aplikasi pemilihan umum (pemilu) online sebagai jawaban dari tuntutan demokrasi modern yang menginginkan digitalisasi pada pemilu. Hasil yang didapatkan adalah dengan mengkombinasikan beberapa model keamanan seperti menggunakan saluran khusus dengan keamanan protokol yang tinggi, *e-paper* dengan sebuah digital signature dan sebuah *smart card* yang telah dimiliki oleh masing-masing pemilih (KTP digital) sebagai autentifikasi validitas pemilih untuk menghindari pemilih

palsu. Model digital signature yang digunakan pada *remote vote* ini yaitu setiap suara yang digunakan pada saat menentukan satu pilihan, maka pilihan ini akan dikonversikan menjadi pesan dalam besaran digital dengan enkripsi algoritma RSA, sehingga yang keluar dari mesin pemilih adalah hasil dengan nilai yang telah terenkripsi. Ini berguna untuk menghindari pemalsuan elektroral yang mungkin terjadi dan hanya orang tertentu serta komputer server yang bisa mendekripsikan kembali pilihan dalam bentuk pesan dan enkripsi yang telah di pilih oleh *voter*.

Chao-Yong Hsu dan **Chun-Shien Lu** (2005) meneliti tentang perbedaan berbagai model tipe kompresi terhadap kualitas warna dan mencari warna yang paling efektif untuk watermarking. Percobaan yang dilakukan dengan mengambil sebuah gambar dan melakukan manipulasi pada gambar tersebut dengan metode memberikan tekanan pada *halftone* warna serta kualitas resolusi dan menghasilkan beberapa model gambar yang telah di modifikasi. Hasil dari modifikasi kemudian di analisis.

El-Affendi (2008) yaitu penerapan watermarking untuk mengamankan tanda tangan fisik yang disertai dengan stempel dalam aplikasi *e-government*. Setiap berkas/surat elektronik yang telah ditanda tangani oleh pihak yang berwenang dan distempel, pada setiap tanda tangan dan stempelnya diselipkan kriptografi watermarking. Setiap dokumen dikirim bersamaan dengan contoh hasil tanda tangan dan stempel fisik yang tidak disandikan dengan watermarking yang berfungsi sebagai pembanding dari tandatang dan stempel pada dokumen yang telah di enkripsi dengan algoritma tertentu.

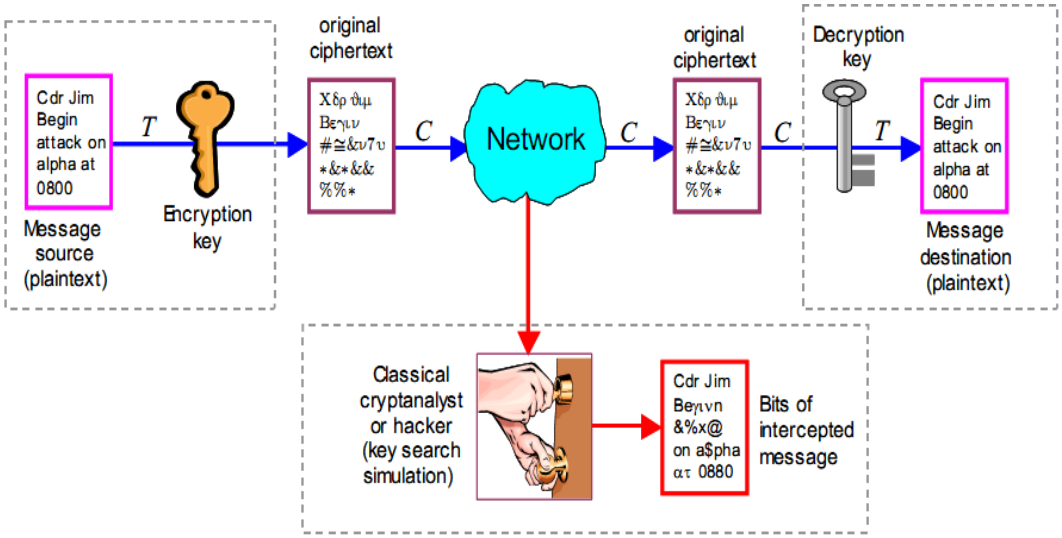
Don Johnson dan **Alfred Menezes** (2001) mencoba menganalisis kelebihan dari model *Elliptic Curve Digital Signature Algorithm (ECDSA)*, dari beberapa perbandingan parameter dengan persamaan diskrit biasa akan didapat beberapa kemudahan dalam faktorisasi. Selain itu keamanan yang tinggi serta implementasi yang mudah menjadikan algoritma ini bisa dijadikan dalam satu model algoritma pada digital signature yang handal.

2.2. LANDASAN TEORI

2.2.1. Konsep Kriptografi

Kriptografi klasik yang mulai digunakan dari jaman Yunani kuno seperti oleh raja Julius Caesar, menggunakan metode subtitusi yang paling sederhana yaitu *Caesar cipher*. Kriptografi klasik lain yang digunakan oleh raja Yunani kuno menggunakan *Scytale* terdiri dari sebuah kertas panjang dari daun papyrus yang dililitkan pada sebuah silinder dengan diameter tertentu (diameter silender menyatakan kunci penyandian), dan masih banyak kriptografi klasik lainnya.

Kriptografi pada abad modern menggunakan matematika untuk melakukan enkripsi dan dekripsi. Data teks asli yang dapat dibaca atau dipahami disebut *plaintext*. Metode penyandian teks asli dengan menggunakan kunci sebagai informasi tambahan sedemikian hingga isi data aslinya tersembunyi disebut enkripsi. Enkripsi digunakan untuk menyembunyikan informasi dari orang yang tidak dikehendaki. Data hasil enkripsi disebut *ciphertext*. Proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi atau de-enkripsi.



Gambar 2.1. Konsep Dasar Proses Enkripsi dan Dekripsi
(Sumber: Rabah, 2009)

Kriptografi dibagi menjadi dua golongan besar yaitu kriptografi kunci simetris (*symmetric-key cryptography*) dan kriptografi kunci asimetris (*asymmetric-key cryptography*).

2.2.1.1. *Symmetric Algorithms*

Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi.

Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu byte data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau byte data (per blok). Contoh algoritma kunci simetris yang terkenal adalah DES (*Data Encryption Standard*).

2.2.1.2. *Asymmetric Algorithms*

Algoritma kriptografi asimetrik adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA dan ECC.

2.2.2. Aspek-aspek Keamanan pada Kriptografi

Inti dari kriptografi adalah menjaga kerahasiaan *plaintext* atau kunci dari penyadapan. Penyadap berusaha mendapatkan data yang digunakan untuk kegiatan pencurian data atau biasa disebut kriptanalisis (*cryptanalysis*). Kriptanalisis bertujuan untuk memecahkan cipherteks menjadi *plainteks* semula tanpa memiliki akses ke kunci yang digunakan hingga

berhasil menemukan kelemahan dari sistem kriptografi yang pada akhirnya mengarah untuk menemukan kunci dan mengungkap *plainteks*.

Aspek-aspek yang diamankan pada sistem kriptografi agar sistem dapat berjalan sempurna menurut Dony Ariyus (2006) ada delapan aspek yang perlu diperhatikan antara lain:

1. *Authentifikasi*: agar penerima informasi dapat memastikan pesan tersebut datang dari orang yang dimintai informasi, dengan kata lain informasi tersebut benar-benar datang dari orang yang dikehendaki.
2. *Integrity*: keaslian pesan yang dikirim melalui sebuah jaringan dan dapat dipastikan bahwa informasi yang dikirim tidak dimodifikasi oleh orang lain yang tidak berhak dalam perjalanan informasi tersebut.
3. *Nonrepudiation*: menyatakan pesan yang dikirim dari orang yang asli, artinya si pengirim pesan tidak dapat mengelak bahwa dialah yang mengirimkan informasi tersebut.
4. *Authority*: informasi yang berada pada sistem jaringan tidak dapat dimodifikasi oleh pihak yang tidak berhak atas akses tersebut.
5. *Confidentiality*: merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses.
6. *Privacy*: merupakan data-data yang sifatnya rahasia dan tidak boleh diketahui oleh pihak lain.
7. *Availability*: Sistem yang diserang atau di jebol dapat menghambat atau meniadakan akses ke informasi.
8. *Access Control*: Aspek ini berhubungan dengan cara pengaturan siapa-siapa saja yang berhak mengakses sistem, mengetahui sistem keamanannya.

2.2.3. Teori Bilangan Modulo

Pada proses pembangkitan kunci pada kurva eliptik terdapat operator modulo. Aritmatika modulo merupakan salah satu dari teori bilangan bulat yang penting yang digunakan untuk perhitungan bilangan bulat pada kurva ellipstik.

Adapun fungsi modulo didefinisikan sebagai berikut:

1. Misalkan *a* adalah bilangan bulat, dan *m* bilangan bulat > 0. Operasi *a mod m* memberikan sisa jika *a* dibagi dengan *m*.
2. *a mod m* dibaca 'a modulo m'
3. notasi *a mod m = r* sehingga $a = mq + r$, dengan $0 \leq r < m$.
4. *m* disebut modulus atau modulo, dan hasil modulo *m* terletak di dalam himpunan {0,1,2, ..., m+1}

Contoh dari fungsi modulo misalnya 23 mod 5 = 3, 27 mod 3 = 0.

Untuk dua buah bilangan a dan b yang berbeda, bisa saja memiliki sisa yang sama jika dibagi dengan bilangan positif m. Hal ini bisa disebut bahwa a dan b kongruen dalam modulo m, yang dilambangkan dengan $a \equiv b \pmod{m}$.

Misalnya 38 mod 5 dan 13 mod 5, hasil dari dua operasi tersebut adalah 3. Maka dapat dikatakan $38 \equiv 13 \pmod{5}$.

2.2.4. Teori Bilangan Prima

Pada proses pembangkitan kunci pada kurva eliptik terdapat bilangan prima. Bilangan prima adalah bilangan integer positif, $p > 1$ adalah prima jika hanya dapat dibagi oleh 1 dan *p* (bilangan prima itu sendiri). Karena bilangan prima harus lebih besar dari 1, maka barisan bilangan prima dimulai dari 2, yaitu 2,3,5,7,11,13,…. Seluruh bilangan prima adalah bilangan ganjil kecuali 2.

Bilangan selain prima disebut bilangan komposit. Misal 20 adalah bilangan komposit karena 20 dapat dibagi oleh 2, 4, 5 dan 10 selain 1 dan 20 sendiri.

Menurut Stalling (2004) Dua buah bilangan bulat *a* dan *p* dikatakan relatif prima/koprima (*coprime*) jika faktor persekutuan terbesar (FPB) (*a*, *p*) = 1. Sehingga jika *a* dan *b* relatif prima, maka terdapat bilangan bulat *m* dan *n* sedemikian. Sehingga *ma* + *nb* = 1. Contoh : Bilangan 20 dan 3 adalah relatif prima karena PBB (20, 3) =1, atau dapat ditulis 2 . 20 + (–13) . 3 = 1

2.2.5. Fungsi HASH

Dalam peneltian ini digunakannya fungsi Hash untuk menentukan string yang berukuran apapun diubah menjadi message digest dengan bit yang konstan. Secara sederhana fungsi hash ditujukan untuk pengalamatan record di memori. Bentuk dari fungsi hash adalah sebagai berikut:

$$h(k) = k \bmod m \qquad (2.20)$$

dengan *k* adalah kunci bilangan bulat dan *m* adalah jumlah lokasi memori yang tersedia. Sedangkan hasilnya *h(k)* adalah lokasi memori untuk record dengan kunci *k*. (Munir: 2005).

Contoh: m = 11 mempunyai sel+sel memori yang diberi indeks 0 sampai 10. Akan disimpan data record yang masing+masing mempunyai kunci 15, 558, 32, 132, 102, dan 5. Maka:

h(15) = 15 mod 11 = 4
h(558) = 558 mod 11 = 8
h(32) = 32 mod 11 = 10
h(132) = 132 mod 11 = 0
h(102) = 102 mod 11 = 3
h(5) = 5 mod 11 = 5

penempatan record pada memori dengan fungsi hash menjadi:

132			102	15	5			32		558
0	1	2	3	4	5	6	7	8	9	10

Fungsi hash sering juga disebut sebagai cryptographic checksum karena bisa digunakan untuk mentransformasi masukan string dengan panjang sembarang menjadi sebuah string lain dengan panjang tetap. Hasil dari transformasi tersebut biasanya berukuran lebih pendek dibanding string masukannya. Hasil transformasi ini disebut juga nilai hash atau message digest. Jika dituliskan dalam notasi matematis akan jadi seperti:

$$MD = Hash(M) \tag{2.21}$$

dengan MD adalah message digest, dan M adalah string masukan.

Oleh fungsi hash sebuah string yang berukuran apapun diubah menjadi message digest yang berukuran tetap (128+512 bit). Adapun sifat+sifat yang dimiliki oleh fungsi hash adalah sebagai berikut :

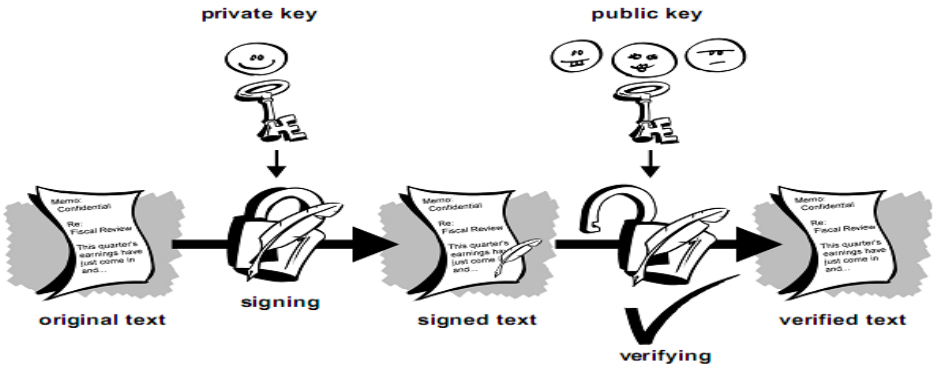
- Fungsi H dapat diterapkan pada blok data yang berukuran berapa saja.
- Nilai hash yang dihasilkan memiliki panjang yang tetap.
- Untuk setiap h yang diberikan, tidak mungkin menemukan suatu x sedemikian sehingga H(x)=h. Fungsi H tidak dapat mengembalikan nilai hash menjadi masukan awal.
- Untuk setiap x yang diberikan, tidak mungkin mencari pasangan $x \neq y$ sedemikian sehingga H(x)=H(y).

2.2.6. Digital Signature

Tanda tangan digital (*Digital Signature*) adalah suatu mekanisme untuk menggantikan tanda tangan secara manual pada dokumen kertas. Yang dimaksud dengan tanda tangan digital menurut Rinaldi Munir (2005) bukanlah tanda tangan yang di-digitalisasi dengan alat *scanner*, tetapi suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan. Kegunaan tanda tangan digital adalah menyatakan pengesahan (*data integrity*) atas apa yang tercatum dalam dokumen tersebut, dan menyatakan pertanggung jawaban penandatangan

(*data originality*) atas apa yang tertulis dalam dokumen tersebut, serta untuk mencegah satu saat penandatangan mengingkari apa yang tertulis didokumen bertanda tangan (*non repudiation*).

Adapun aspek keamanan kerahasiaan (*confidentality*) bukan disediakan dengan sistem tanda tangan digital, tetapi tanda tangan yang telah dienkripsikan terlebih dahulu dan menghasilkan sebuah *publick key* serta tanda tangan dengan algoritma tertentu. Jika Digital Signature yang telah di enkripsi menggunakan kunci publik X, maka pada proses mendeskripsikan kembali dengan kunci pribadi X. Tidak akan terbuka dengan kunci pribadi Y.



Gambar 2.2. Proses Pengabsahan Pada Tanda Tangan Digital.
(Sumber: http://www.nai.com/An_Introduction_to_Cryptography.html)

Penandatanganan pesan dengan cara mengenkripsikannya selalu memberikan dua fungsi berbeda, yaitu kerahasiaan pesan dan otentifikasi. Pada beberapa kasus, seringkali otentifikasi yang diperlukan tetapi kerahasiaan tidak. Maksudnya pesan tidak perlu dienkripsikan, sebab yang diperlukan hanya otentikasi saja.

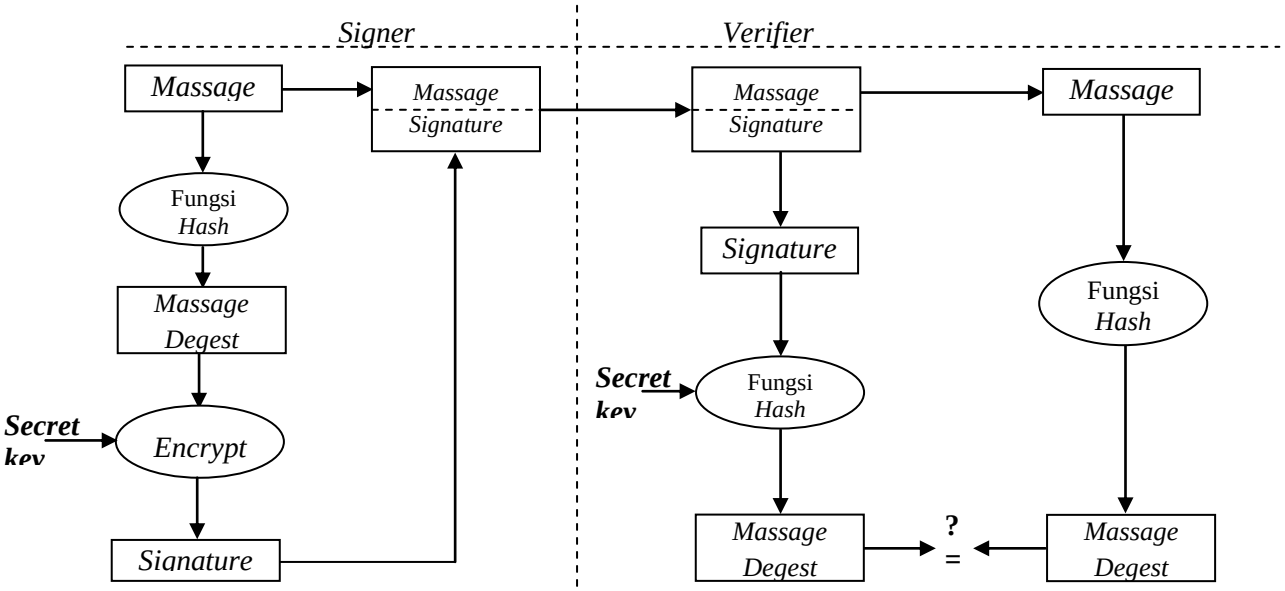
Hanya sistem kriptografi kunci publik yang cocok dan alami untuk pemberian tanda tangan digital dengan menggunakan fungsi *hash*. Hal ini karena disebabkan karena skema tanda tangan digital berbasis sistem kunci publik dapat menyediakan masalah *non-repudiation* (baik penerima dan pengirim pesan mempunyai pasangan kunci masing-masing).

Teknik yang umum digunakan untuk membentuk tanda tangan digital adalah dengan fungsi hash dan melibatkan algoritma kriptografi kunci-publik. Mula-mula pesan

M ditransformasi oleh fungsi hash H menjadi pesan ringkas h. Pesan ringkas tersebut dienkripsi dengan kunci *private* (PK) pengirim pesan: S = ESK(h). Hasil enkripsi (S) inilah yang disebut tanda-tangan digital. Tanda-tangan digital dapat ditambahkan pada pesan atau terpisah dari pesan dan dikirim secara bersamaan. Di tempat penerima, tanda tangan diverifikasi untuk dibuktikan keotentikannya dengan cara berikut:

a. Tanda tangan digital *S* didekripsi dengan menggunakan kunci publik (PK) pengirim pesan, menghasilkan pesan-ringkas semula, h, sebagai berikut: h = DPK(S)
b. Pengirim kemudian mengubah pesan M menjadi pesan ringkas h' dengan menggunakan fungsi hash satu-arah yang sama dengan fungsi hash yang digunakan oleh pengirim.
c. Jika h' = h, berarti tanda-tangan yang diterima otentik dan berasal dari pengirim yang benar.

Gambar 2.3 memperlihatkan proses pembangkitan tanda tangan digital (*signing*) dan verifikasi tanda tangan digital (*verivying)*.



Gambar 2.3. Otentifikasi dengan tanda tangan digital yang menggunakan fungsi hash satu-arah (Munir, 2005)

Otentikasi pesan dapat dijelaskan sebagai berikut:

a. Apabila pesan M yang diterima sudah berubah, maka h' yang dihasilkan dari fungsi hash berbeda dengan h semula. Ini berarti pesan tidak asli lagi.

b. Apabila pesan M tidak berasal dari orang yang sebenarnya, maka h yang dihasilkan berbeda dengan h' yang dihasilkan pada proses verifikasi (hal ini karena kunci publik yang digunakan oleh penerima pesan tidak berkoresponden dengan kunci privat pengirim).

c. Bila h = h', ini berarti pesan yang diterima adalah pesan yang asli dan orang yang mengirim adalah orang yang sebenarnya.

Beberapa parameter dasar pada *Digital Standard Algorithm* (DSA) seperti berikut ini: (Munir: 2005)

1. p, adalah bilangan prima dengan panjang L bit, yang dalam hal ini $512 \leq L \leq 1024$ dan L harus kelipatan 64. Parameter p bersifat publik dan dapat digunakan bersama-sama oleh orang di dalam kelompok.
2. q, bilangan prima 160 bit, merupakan faktor dari $p - 1$. Dengan kata lain, $(p - 1) \bmod q = 0$. Parameter q berisfat publik.
3. $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini $h < p - 1$ sedemikian sehingga $h^{(p-1)/q} \bmod p > 1$. Parameter g bersifat publik.
4. x, adalah bilangan bulat kurang dari q. Parameter x adalah kunci rahasia.
5. $y = g^x \bmod p$, adalah kunci publik.
6. m, pesan yang akan diberi tanda tangan.

Proses pembangkitan sepasang kunci adalah sebagai berikut:

1. Pilih bilangan prima p dan q, yang dalam hal ini $(p - 1) \bmod q = 0$.
2. Hitung $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini
   $1 < h < p - 1$ dan $h^{(p-1)/q} \bmod p > 1$.

3. Tentukan kunci rahasia x, yang dalam hal ini $x < q$.
4. Hitung kunci publik $y = g^x \bmod p$.

Prosedur di atas menghasilkan:

1. Kunci publik dinyatakan sebagai (p, q, g, y)
2. Kunci *private* dinyatakan sebagai (p, q, g, x).

Prosedur pembangkitan tanda tangan (*Signing*):

1. Ubah pesan m menjadi *massage digest* dengan fungsi *hash SHA*, H.
2. Tentukan bilangan acak $k < q$.
3. Tanda tangan dari pesan m adalah bilangan r dan s. Hitunglah r dan s sebagai berikut:

   $r = (g^k \bmod p) \bmod q$

   $s = (k^{-1}(H(m) + x * r)) \bmod q$
4. Kirimkan pesan m beserta tanda tangan r dan s.

Prosedur verifikasi keabsahan tanda tangan (*Verifing*)

1. Hitung

   $w = s^{-1} \bmod q$

   $u_1 = (H(m) * w) \bmod q$

   $u_2 = (r * w) \bmod q$

   $v = ((q^{u1} * y^{u2}) \bmod p) \bmod q)$
2. Jika v = r, maka tanda tangan sah, yang berarti bahwa pesan masih asli dan dikirim oleh pengirim yang benar.

2.2.7. Algoritma Tanda Tangan Digital Kurva Eliptik.

Kriptosistem kurva eliptik (*Elliptic Curves Cryptosystem*) di perkenalkan oleh Neil Koblitz dan Viktor Miller pada tahun 1985 yang menggunakan masalah logaritma diskrit pada titik-titk kurva eliptik yang disebut dengan ECDLP (*Elliptic Curves Discrete Logarithm*

Problem) (www.wikipedia.org). Kriptosistem kurva eliptik ini dapat digunakan pada beberepa keperluan antara lain Skeme enkripsi (ElGamal ECC) dan Tanda tangan digital (ECDSA–*Elliptic Curves Digital Signature*).

Pada tanda tangan digital, pendekatan yang dilakukan untuk menghasilkan algoritma kurva eliptik adalah dengan meggunakan struktur matematika yang sangat unik yang memungkinkan memrosesan titik dengan memiliki dua buah titik dalam sebuah kurva eliptik dan menghasilkan sebuah titik lain yang ada pada kurva tersebut. Struktur yang unik ini memberikan keuntungan dalam kriptografi dikarenakan kesulitan untuk menemukan 2 buah titik yang menentukan sebuah titik tertentu tersebut tidak dapat ditemukan dengan mudah. Tingkat kesulitan untuk menemukan 2 buah titik termasuk dalam golongan yang rumit sama seperti kesulitan untuk memperhitungkan variasi eksponensial yang digunakan dalam algoritma RSA yang telah banyak diimplementasikan. Sehingga tanda tangan digital dengan algoritma kurva eliptik ini lebih aman terhadap *sniffing* yang mencoba untuk mendapatkan informasi dari pesan yang terenkripsi.

Menurut Certicom, parameter-parameter domain kriptografi kurva eliptik pada bidang F_p didefinisikan sebagai six-tuple T, yaitu:

$$\boldsymbol{T = (p, a, b, G, n, h).}$$

Dimana:

F_p *:* Lapangan berhingga prima yang memiliki p elemen. $F_p=\{0,1,\ldots,p-1\}$

p *:* bilangan prima

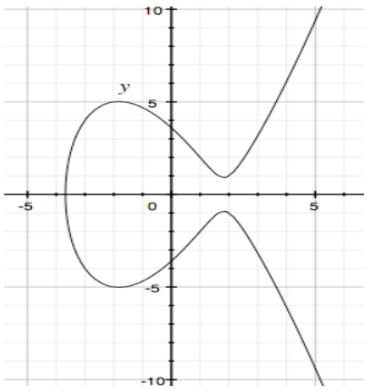
a,b *:* koefisien persamaan kurva eliptik $y^2 = x^3 + ax + b \pmod p$ (2.1)

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

n *:* order basic point, yaitu bilangan bulat positip terkecil $n.G = \boldsymbol{O}$

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

Setiap perubahan nilai dari '*a*' dan'*b*' akan menghasilkan kurva eliptik yang berbeda. Contoh pada persamaan $y^2 = x^3 - 10x + 13$, maka akan menghasilkan grafik sebagai berikut:



Gambar 2.4. Grafik Kurva Eliptik dengan Persamaan $y^2 = x^3$-10x + 13

Setiap kurva eliptik akan mendefinisikan kumpulan titik pada bidang dan dapat membentuk kumpulan albelian (kumpulan titik dengan titik tak hingga sebagai elemen identitas). Jika nilai *x* dan *y* yang dipilih adalah daerah finit yang besar, solusi akan menghasilkan suatu *albelian finite*.

Proses pembuatan tanda tangan, hingga pengujian tanda tangan digital dengan algoritma kurva eliptik menurut aturan standar (**certicom,** 2000) adalah sebagai berikut:

1. Proses pembangkitan sepasang kunci:

a. Menentukan sebuah bilangan bulat random d_A, yang nilainya diantara *[1,n-1]*

b. Menghitung $Q_A = d_A * G \rightarrow G[(x_1, y_1)]$ (2.2)

dengan $y^2 = x^3 + ax + b$ (mod *p*).

d_A = kunci rahasia

Q_A = kunci publik.

2. Prosedur pembangkitan tanda tangan (*Signing*):

a. Memilih sebuah bilangan bulat random *k*, yang nilainya diantara *[1, n-1]*

b. Menghitung $Q_A = k * G = (x_1,y_1)$ (2.3)

dan $r = x_1 \bmod n$ (2.4)

jika r = 0 maka kembali ke langkah a

c. Menghitung $k^{-1} \bmod n$ (2.5)

d. Menghitung $e = HASH(m)$ (2.6)

(dimana m adalah pesan yang akan di *signing*)

e. Menghitung $s = k^{-1} \{e + d_A * r\} \bmod n$ (2.7)

Tanda tangan untuk *massage m* adalah (r, s)

3. Prosedur verifikasi keabsahan tanda tangan (*Verifing*)

a. Memverifikasi bahwa r dan s adalah bilangan bulat antara $[1, n-1]$

b. Menghitung $e = HASH(m)$

c. Menghitung $w = s^{-1} \bmod n$ (2.8)

d. Menghitung $u_1 = ew \bmod n$ (2.9)

dan $u_2 = rw \bmod n$ (2.10)

e. Menghitung $u_1 * G + u_2 * Q_A = (x_1, y_1)$ (2.11)

f. Menghitung $v = x_1 \bmod n$ (2.12)

Jika v = r, maka tanda tangan adalah sah

Seperti dengan kriptografi pada umumnya, ukuran bit dari kunci publik diyakini diperlukan untuk tanda tangan digital kurva eliptik adalah sekitar dua kali ukuran tingkat keamanan dalam bit. Sebagai perbandingan, pada tingkat keamanan 80 bit, berarti penyerang memerlukan sekitar setara dengan sekitar 280 generasi tanda tangan untuk menemukan kunci pribadi, ukuran kunci DSA publik setidaknya 1024 bit, sedangkan ukuran sebuah kunci publik ECDSA akan menjadi 160 bit.

Tabel 2.1. Penggunaan Panjang Bit *Publick Key* dan *Private Key* antara ECDSA dan RSA

(Sumber: http://www.rsa.com/rsalabs/node.asp?id=2013)

	ECDSA and ECES over GF(q)	RSA 1024-bit n and e=216+1
system parameters	(4 x 160)+1 = 641	0
public key	160+1 = 161	1024 + 17 = 1041
private key	160 (801 with system parameters)	2048 (or 2560 with CRT information)

2.2.8. Bidang Terbatas

Untuk membentuk kurva eliptik dalam persamaan $y^2 = x^3 + ax + b \ (mod\ p)$ pada bidang terbatas F_p, maka diperlukan pengetahuan tentang bidang terbatas untuk menentukan titik-titik pada kurva eliptik pada penelitian ini.

Bidang terbatas (*finite field*) atau yang biasa disebut dengan *Galois Field* (GF) adalah bidang yang hanya memiliki elemen bilangan yang terbatas. Derajat (*order*) dari *finite field* adalah banyaknya elemen yang ada di dalam bidang. Jika *q* adalah pangkat prima (*prime power*), maka hanya ada satu bidang terbatas dengan derajat *q*. Bidang tersebut dilambangkan dengan F_q atau *GF(q).* Banyak cara untuk merepresentasikan elemen dari *Fq*, jika $q = pm$, dimana *p* adalah bilangan prima dan *m* adalah bilangan integer positif, maka *p* disebut sebagai karakteristik dari *Fq* dan *m* disebut sebagai derajat perluasan (*extension degree*) dari *Fq*. Bidang terbatas yang digunakan dalam kriptografi adalah $q = p$, dimana *p* adalah bilangan prima ganjil, yang dilambangkan dengan *Fp* (*odd prime*), dan $q=2^m$, dimana *m* adalah integer lebih besar dari satu, yang dilambangkan dengan F_2^m.

2.2.8.1. Bidang Terbatas F_p

Bidang Terbatas *Fp* merupakan sebuah bidang yang beranggotakan bilangan integer {0,1,...,*p-1*}, dan p merupakan bilangan prima, setiap perhitungan dikalkulasikan dengan modulo *p* agar hasilnya tetap berada dalam daerah *Fp*. Operasi yang berlaku dalam bidang terbatas *Fp* adalah:

1. Penjumlahan (*Addition*), jika $a,b \in Fp$, maka $a + b = r$, dimana r adalah sisa pembagian $a + b$ dengan bilangan prima p, $0 \leq r \leq p-1$. penjumlahan seperti ini disebut penjumlahan modulo p (mod p).
2. Perkalian (*Multiplication*), *jika* $a,b \in Fp$, maka $a * b = s$, dimana s adalah sisa pembagian $a * b$ dengan bilangan prima p, $0 \leq s \leq p-1$. Perkalian seperti ini disebut perkalian modulo p (mod p).

2.2.8.2. Bidang Terbatas F_2^m

Bidang terbatas F_2^m biasa disebut dengan bidang terbatas biner (*binary finite field*), dapat dipandang sebagai ruang vektor berdimensi m pada F2. Karena itu ada himpunan yang beranggotakan m elemen {α0,α1,...αm-1}di dalam F2m sedemikian rupa sehingga setiap a∈ F_2^m dapat ditulis secara unik ke dalam bentuk:

$$a = a_0 \alpha_0 + a1\alpha1+...+ a_{m-1} \alpha_{m-1} \text{ , untuk ai} \in \{0,1\} \quad (2.13)$$

Salah satu cara untuk merepresentasikan elemen-elemen pada F_2^m adalah dengan representasi basis polinomial. Pada representasi basis polinomial elemen pada F_2^m merupakan polinomial dengan derajat lebih kecil dari m, dengan koefisien bilangan 0 atau 1.

$$\{a_{m-1}x^{m-1} + ... + a_2x^2 + a_1x^1 + a_0x^0 \mid a_i : 0,1\} \quad (2.14)$$

Operasi yang berlaku dalam bidang terbatas F_2^m representasi basis polinomial:

1. Penjumlahan (*Addition*), $(a_{m-1}...a_1a_0) + (b_{m-1}..b_1b_0) = (c_{m-1}...c_1c_0)$ dimana $c_i = a_i + b_i$. Operasi penjumlahan dapat menggunakan deretan komponen $(a_{m-1}...a_1a_0)$ yang di-XOR-kan dengan $(b_{m-1}..b_1b_0)$.
2. Perkalian (*Multiplication*), $(a_{m-1}...a_1a_0) * (b_{m-1}..b_1b_0) = (r_{m-1}...r_1r_0)$ dimana $r_{m-1}x^{m-1} +... + r_1x + r_0$ adalah sisa dari pembagian $(a_{m-1}x^{m-1} +...+ a_1x+ a_0)*$ $(b_{m-1}x^{m-1} + ... + b_1x + b_0)$ dibagi dengan polinomial $f(x)$ pada F_2 (setiap koefisien polinomial di reduksi ke modulo 2).

2.2.9. Kurva Eliptik Pada Bidang Terbatas

Ada beberapa cara untuk mendefinisikan persamaan kurva eliptik bergantung kepada bidang terbatas yang digunakan apakah F_p atau $F_2{}^m$. Persamaan *Weierstrass* yang digunakan untuk kedua bidang terbatas tersebut berbeda.

2.2.9.1. Kurva Eliptik Pada Bidang Terbatas F_p

Misalkan $p > 3$ adalah bilangan prima ganjil, dan $a,b \in F_p$ memenuhi $4a^3 + 27b^2 \neq 0 \pmod p$, maka sebuah kurva eliptik $E(F_p)$ pada F_p merupakan himpunan titik-titik $P(x,y)$, dimana $x,y \in F_p$,yang memenuhi persamaan $y^2 = x^3 + ax + b$, dan sebuah titik khusus $\phi(\infty,\infty)$ yang merupakan titik tak hingga. Operasi penjumlahan pada $E(F_p)$ didefinisikan sebagai berikut :

1. $P + \phi = \phi + P = P$ untuk setiap $P \in E(F_p)$

   Jika $P(x.y) \in E(F_p)$, maka $(x,y) + (x,-y) = \phi$ titik $(x,-y) \in E(F_p)$ dinotasikan sebagai $-P$, disebut sebagai negatif dari P

2. Misalkan $P(x_1,y_1) \in E(F_p)$, $Q(x_2,y_2) \in E(F_p)$, dan $P \neq \pm Q$, maka $P + Q = (x_3,y_3)$ dimana :

   $$x_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)^2 - x_1 - x_2 \qquad (2.15)$$

   $$y_3 = \left(\frac{y_2-y_1}{x_2-x_1}\right)(x_1 - x_3) - y_1 \qquad (2.16)$$

3. Misalkan $P(x_1,y_1) \in E(F_p)$, maka $P + P = 2P = (x_3,y_3)$, dimana :

   $$x_3 = \left(\frac{3{x_1}^2+a}{2y_1}\right)^2 - 2x_1$$

   $$y_3 = \left(\frac{3{x_1}^2+a}{2y_1}\right)(x_1 - x_3) - y_1$$

Operasi di atas disebut dengan penggandaan titik (doubling a point).

Kehebatan dari operasi penjumlahan pada kurva eliptik adalah jika menjumlahkan dua buah titik yang merupakan elemen dari kelompok kurva eliptik, maka hasil penjumlahannya adalah titik lain yang juga merupakan elemen dari kelompok kurva eliptik tersebut.

2.2.9.2. Kurva Eliptik Pada Bidang Terbatas F_2^m

Sebuah kurva eliptik E pada F_2^m didefinisikan sebagai sebagai sebuah persamaan dalam bentuk :

$y^2 + xy = x^2 + ax^2 + b$

dimana $a,b \in F_2^m$, dan $b \neq 0$. Set E (F_2^m) terdiri dari seluruh titik (x,y), $x \in F_2^m$, $y \in F_2^m$ yang memenuhi persamaan kurva eliptik tersebut, bersamaan dengan titik khusus $\phi(\infty,\infty)$ yang disebut titik tak hingga (*point at infinity*).

Sebagaimana kurva-kurva eliptik pada F_p, ada aturan-aturan untuk menjumlahkan titik–titik pada kurva eliptik $E(F_2^m)$ untuk mendapatkan sebuah titik ketiga kurva eliptik. Rumus aljabar untuk menjumlahkan dua titik dan menggandakan dua titik adalah sebagai berikut:

1. $P + \phi = \phi + P = P$ untuk seluruh $P \in E(F_2^m)$.

   Jika $P = (x,y) \in E(F_2^m)$, kemudian $(x,y) + (x, x+y) = \phi$. (Titik $(x,x+y)$ dinotasikan dengan $-P$, dan disebut negatif P.

2. Misalkan $P = (x_1,y_1) \in E(F_2^m)$ dan $Q = (x_2,y_2) \in E(F_2^m)$, dimana $P \neq \pm Q$. Kemudian $P + Q = (x3,y3)$, dimana:

   $$x_3 = \left(\frac{y_1+y_2}{x_1+x_2}\right)^2 + \frac{y_1+y_2}{x_1+x_2} + x_1 + x_2 + a \qquad (2.17)$$

   $$y_3 = \left(\frac{y_1+y_2}{x_1+x_2}\right)(x_1 + x_3) + x_3 + y_1 \qquad (2.18)$$

3. Penggandaan titik (Point doubling) Misalkan $P = (x_1,y_1) \in E(F_2^m)$, kemudian $2P = (x_3,y_3)$, dimana :

   $$x_3 = x_1{}^2 + \frac{b}{x_1{}^2} \qquad (2.19)$$

$$y_3 = {x_1}^2 + \left(x_1 + \frac{y_1}{x_1}\right) x_3 + x_3$$

2.2.10. Barcode

Barcode adalah sebuah perwakilan informasi yang dapat terbaca oleh mesin, biasanya berupa baris warna hitam di atas latar belakang putih yang terpola jaraknya, yang mewakili 1 dan 0 dalam bilangan biner. Baris-baris ini mengandung informasi yang mudah terbaca oleh mesin (Wahyono, 2010).

Pada awal perkembangannya, penggunaan barcode dilakukan untuk membantu proses pemeriksaan barang-barang secara otomatis pada pasar-pasar swalayan. Namun, pada saat ini barcode sudah banyak digunakan pada kartu identitas, kartu kredit, maupun untuk pemeriksaan secara otomatis pada peralatan yang memerlukan penomoran dan pengkodean yang otomatis.

Ada beberapa tipe dan jenis barcode yang digunakan, antara lain; 1). barcode satu dimensi yang biasa juga disebut kode baris linier; 2) barcode dua dimensi. yang merupakan kombinasi kode matriks bujur sangkar. Barcode dua dimensi ini diantaranya adalah PDF Code, QRCode, Matrix Code dan lain-lain. Dengan menggunakan barcode dua dimensi jumlah karakter yang bisa kita masukkan ke barcode bisa semakin banyak dibandingkan barcode satu dimensi.

Alat yang digunakan untuk membaca barcode adalah barcode reader atau barcode scanner yang berfungsi untuk membaca barcode dengan memancarkan sinar laser yang mengenai barcode kemudian dibaca kembali dan diterjemankan kedalam besaran karakter yang dimengerti oleh komputer.

Gambar 2.5. Contoh Barcode dan Barcode Reader.
(Sumber: http://www.barcoderesource.com/index.shtml)

2.2.11. Jenis-jenis Barcode

2.2.11.1. Code 39

Code 39 dapat mengkodekan karakter alphanumeric yaitu angka desimal dan huruf besar serta tambahan karakter spesial -.*$/%+. Satu karakter dalam Code 39 terdiri dari 9 elemen yaitu 5 bar (garis vertikal hitam) dan 4 spasi (garis vertikal putih) yang disusun bergantian antara bar dan spasi. 3 dari 9 elemen tersebut memiliki ketebalan lebih tebal dari yang lainnya oleh karenanya kode ini biasa disebut juga *code 3 of 9*, 3 elemen yang lebih tebal tersebut terdiri dari 2 bar dan 1 spasi. Elemen yang lebar mewakili digit biner 1 dan elemen yang sempit mewakili digit biner 0. Tabel karakter code 39 beserta nilai seperti berikut ini:

Tabel 2.2. Peta Karakter Code 39
(Sumber: http://www.barcoderesource.com/code39_barcode_map.html)

Data Character	Barcode Font Character	Code39 Value	Data Character	Barcode Font Character	Code39 Value
'0'	'0'	0	'M'	'M'	22
'1'	'1'	1	'N'	'N'	23
'2'	'2'	2	'O'	'O'	24
'3'	'3'	3	'P'	'P'	25
'4'	'4'	4	'Q'	'Q'	26
'5'	'5'	5	'R'	'R'	27
'6'	'6'	6	'S'	'S'	28
'7'	'7'	7	'T'	'T'	29
'8'	'8'	8	'U'	'U'	30
'9'	'9'	9	'V'	'V'	31
'A'	'A'	10	'W'	'W'	32
'B'	'B'	11	'X'	'X'	33
'C'	'C'	12	'Y'	'Y'	34
'D'	'D'	13	'Z'	'Z'	35
'E'	'E'	14	'-'	'-'	36

'F'	'F'	15	'.'	'.'	37
'G'	'G'	16	' '(space)	' '(space)	38
'H'	'H'	17	'$'	'$'	39
'I'	'I'	18	'/'	'/'	40
'J'	'J'	19	'+'	'+'	41
'K'	'K'	20	'%'	'%'	42
'L'	'L'	21	'*'	'*' (Start/Stop Character)	

2.2.11.2. Code 128

Code 128 adalah standar *barcode* (kode baris) dengan kerapatan yang tinggi, dapat mengkodekan seluruh karakter ASCII (128 karakter) dalam lingkup yang paling minimum diantara jenis kode baris yang lain. Hal ini dikarenakan Code 128 menggunakan 4 ketebalan elemen (bar atau spasi) yang berbeda-beda (jenis yang lain biasanya hanya menggunakan 2 macam ketebalan elemen). Setiap karakter pada Code 128 dikodekan dengan 3 bar dan 3 spasi (6 bar keseluruhan) dengan ketebalan masing-masing elemen 1 sampai 4 kali ketebalan minimum (module). Jika dihitung dengan satuan module, maka tiap karakter Code 128 terdiri dari 11 module, kecuali untuk stop character yang terdiri dari 4 bar dan 3 spasi (13 module). Jumlah total module untuk bar selalu genap, sedangkan untuk spasi selalu ganjil. Selain itu, Code 128 memiliki 3 start character yang berbeda, sehingga Code 128 memiliki 3 sub-set character yang bersesuaian dengan start character-nya. Seperti yang diperlihatkan pada tabel berikut ini:

Tabel 2.3. Peta Karakter Code 128

(Sumber: http://www.barcoderesource.com/code128_barcode_map.html)

Value	Code 128 A	Code 128 B	Code 128 C	Value	Code 128 A	Code 128 B	Code 128 C	Value	Code 128 A	Code 128 B	Code 128 C
0	' '	' '	0	36	'D'	'D'	36	72	BS	'h'	72
1	'!'	'!'	1	37	'E'	'E'	37	73	HT	'i'	73
2	'"'	'"'	2	38	'F'	'F'	38	74	LF	'j'	74
3	'#'	'#'	3	39	'G'	'G'	39	75	VT	'k'	75
4	'$'	'$'	4	40	'H'	'H'	40	76	FF	'l'	76
5	'%'	'%'	5	41	'I'	'I'	41	77	CR	'm'	77
6	'&'	'&'	6	42	'J'	'J'	42	78	SO	'n'	78
7	''	''	7	43	'K'	'K'	43	79	SI	'o'	79
8	'('	'('	8	44	'L'	'L'	44	80	DLE	'p'	80
9	')'	')'	9	45	'M'	'M'	45	81	DC1	'q'	81
10	'*'	'*'	10	46	'N'	'N'	46	82	DC2	'r'	82
11	'+'	'+'	11	47	'O'	'O'	47	83	DC3	's'	83
12	','	','	12	48	'P'	'P'	48	84	DC4	't'	84
13	'-'	'-'	13	49	'Q'	'Q'	49	85	NAK	'u'	85
14	'.'	'.'	14	50	'R'	'R'	50	86	SYN	'v'	86
15	'/'	'/'	15	51	'S'	'S'	51	87	ETB	'w'	87
16	'0'	'0'	16	52	'T'	'T'	52	88	CAN	'x'	88

17	'1'	'1'	17	53	'U'	'U'	53	89	EM	'y'	89
18	'2'	'2'	18	54	'V'	'V'	54	90	SUB	'z'	90
19	'3'	'3'	19	55	'W'	'W'	55	91	ESC	'{'	91
20	'4'	'4'	20	56	'X'	'X'	56	92	FS	'|'	92
21	'5'	'5'	21	57	'Y'	'Y'	57	93	GS	'}'	93
22	'6'	'6'	22	58	'Z'	'Z'	58	94	RS	'~'	94
23	'7'	'7'	23	59	'['	'['	59	95	US	DEL	95
24	'8'	'8'	24	60	'\'	'\'	60	96	FNC3	FNC3	96
25	'9'	'9'	25	61	']'	']'	61	97	FNC2	FNC2	97
26	':'	':'	26	62	'^'	'^'	62	98	Shift	Shift	98
27	';'	';'	27	63	'_'	'_'	63	99	Code C	Code C	99
28	'<'	'<'	28	64	NUL	'`'	64	100	Code B	FNC4	Code B
29	'='	'='	29	65	SOH	'a'	65	101	FNC4	Code A	Code A
30	'>'	'>'	30	66	STX	'b'	66	102	FNC1	FNC1	FNC1
31	'?'	'?'	31	67	ETX	'c'	67	103	Start A	Start A	Start A
32	'@'	'@'	32	68	EOT	'd'	68	104	Start B	Start B	Start B
33	'A'	'A'	33	69	ENQ	'e'	69	105	Start C	Start C	Start C
34	'B'	'B'	34	70	ACK	'f'	70	106	Stop	Stop	Stop
35	'C'	'C'	35	71	BEL	'g'	71				

Code 128 memiliki fitur untuk dapat beralih dari satu sub-set ke sub-set yang lain dengan menggunakan karakter CODE dan SHIFT. CODE X menyebabkan seluruh message beralih menjadi sub-set X (misalnya, CODE A pada sub-set B membuat message beralih ke sub-set B). Sedangkan SHIFT menyebabkan satu karakter di depannya beralih sub-set (ini hanya berlaku untuk sub-set A ke sub-set B atau sebaliknya).

Struktur dari kode baris Code 128 seperti terlihat pada gambar berikut ini:

Gambar 2.6. Contoh Nilai Terbaca Pada Barcode.
(Sumber: Wahyono, 2010)

Tinggi kode baris minimum adalah 0.15 kali lebar baris kode. Lebar kode baris dinyatakan dengan aturan di bawah ini:

L=X(11C+35) untuk alpha-numerik (CODE A & B)
L=X(5.5C+35) untuk double density numeric (C)

Dimana:

L = lebar kode baris termasuk Quiet Zone

C = jumlah karakter

X = lebar module

2.2.11.2. Barcode Dua Dimensi

Kode baris dua dimensi telah dikembangkan lebih dari sepuluh tahun yang lalu, namun baru sekarang mulai populer. Kode baris dua dimensi ini memiliki beberapa keunggulan dibandingkan kode baris satu dimensi (linier barcode). Yang paling utama adalah: kode baris dua dimensi ini dapat menyimpan data atau informasi dalam suatu ruang yang lebih kecil. Contoh kode baris dua dimensi adalah Symbology PDF417 yang dapat menyimpan lebih dari 2000 karakter dalam sebuah ruang yang berukuran 4 inch persegi.

Pada penelitian ini, jenis barcode yang digunakan adalah code 128, karena karakter yang dihasilkan pada proses *signing* menggunakan karekter angka, huruf besar dan kecil.

2.2.12. Bahasa Pemograman PHP

Aplikasi yang di buat pada penelitian ini menggunakan bahasa pemograman PHP, karena bahasa pemograman ini bekerja pada sisi server sehingga dari segi keamanan kriptografi yang di gunakan lebih terjamin kerahasiaannya karena user hanya terhubung di sisi *client* nya. Untuk disain tampilan digunakan juga *cascading style sheets* (CSS) sebagai dinamisasi tampilan web.

PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C.

Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessing. Pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Saat ini banyak aplikasi yang telah dibuat dengan PHP. Baik sifatnya yang komersil maupun *free*. Salah satu aplikasi yang cukup terkenal dilingkungan *open source* adalah web portal PHP Nuke yang dibuat oleh komunitas *open source* dapat diperoleh secara gratis di internet. Software ini dapat di download di http://www.phpnuke.org. Selain PHP Nuke banyak lagi software web portal yang sejenis seperti Post Nuke yang merupakan turunan dari PHP Nuke. Aplikasi-aplikasi seperti e-commerce juga banyak dikembangkan dengan PHP, aplikasi e-learning, aplikasi search engine, bahkan aplikasi ERP (Enterprise Resource Plan) yang banyak digunakan oleh perusahaan-perusahaan besar juga sudah ada yang dikembangkan dengan PHP.

Ketika akan membuat aplikasi dengan PHP. Supaya PHP dapat dijalankan tentunnya kita perlu memiliki software pendukung yang biasanya sering digunakan. Seperti web server, database server, teks editor, dan web browser. Istilah yang sering gunakan untuk kombinasi software untuk keempat aplikasi di atas di dunia open source dikenal dengan LAMP (Linux

Apache MySQL dan PHP). Jika kita menggunakan sistem operasi windows, kita bisa menggunakan PHP Triad. Software ini telah menyertakan ketiga komponen software untuk pemrograman PHP. Yakni PHP itu sendiri, Apache dan MySQL.

2.2.13. Web Server

Web server adalah software yang menjadi tulang punggung dari *world wide web* (www). Web server menunggu permintaan dari client yang menggunakan browser seperti Netscape Navigator, Internet Explorer, Modzilla, dan program browser lainnya. Jika ada permintaan dari browser, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang standar, disebut dengan format SGML (*standar general markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser tersebut. Contohnya, bila data yang dikirim berupa gambar, browser yang hanya mampu menampilkan teks (misalnya *lynx*) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja. Web server, untuk berkomunikasi dengan client-nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*). Dengan protokol ini, komunikasi antar *web server* dengan client-nya dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan diatas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari. Kata *HyperText* mempunyai arti bahwa seorang pengguna internet dengan *web browser*nya dapat membuka dan membaca dokumen-dokumen yang ada dalam komputernya atau bahkan jauh tempatnya sekalipun. Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa

bagian dari halaman-halaman dokumen (web) itu. Proses yang dimulai dari permintaan *webclient* (browser), diterima web server, diproses, dan dikembalikan hasil prosesnya oleh web server ke web client lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap-tiap proses. Secara garis besarnya *web server* hanya memproses semua masukan yang diperolehnya dari *web client*nya.

Apache merupakan *web server* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya Apache mengeluarkan program yang dapat dijalankan di Windows. Apache mempunyai program pendukung yang cukup banyak, hal ini memberikan layanan yang cukup lengkap bagi penggunanya.

2.2.14. Aplikasi *Sniffing Wireshark*

Wireshark adalah sebuah *Network Paket Analizer* yang berfungsi menangkap dan menganalisa paket-paket data dijaringan komputer dan kemudian menampilkan semua informasi dipaket tersebut secara detail. Wireshark dapat membaca paket data secara langsung dari *Ethernet*, *Token-Ring*, *FDDI*, komunikasi serial (*PPP* and *SLIP*), 802.11 *wireless LAN* , koneksi ATM, dan berbagai macam protokol jaringan yang mampu di analisa oleh wireshark (http://www.wireshark.org/docs/relnotes/wireshark-0.99.8.html).

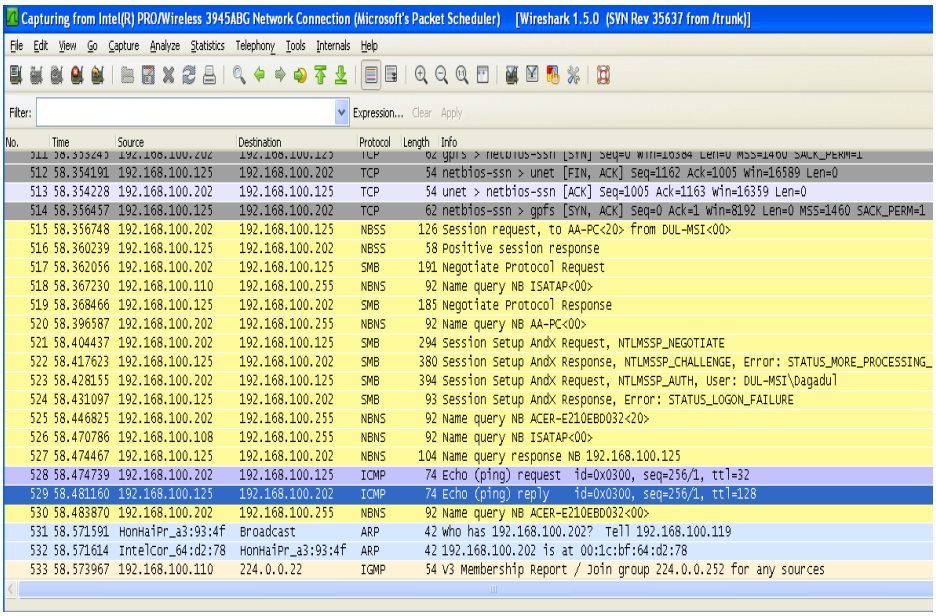
Beberapa contoh penggunaan dari aplikasi sniffing wireshark sebagai berikut:

- Administrator sebuah jaringan menggunakan *wireshark* untuk menganalisa troubleshooting masalah-masalah di jaringan yang dikelola.
- Teknisi keamanan jaringan menggunakannya untuk memeriksa tingkat keamanan jaringan dan memeriksa celah yang nampak di jaringan.
- Pengembang software (*Software development*) bisa menggunakannya untuk men-*debug* implementasi protokol jaringan dan keamanan dalam software.
- Pengguna yang akan mempelajari protocol-protokol jaringan secara detail.

- Pengintip (Sniffer) yang melakukan aktifitas pengendusan (sniffing) untuk mengintip data-data privasi di jaringan.

Prinsip kerja dari aplikasi ini dengan menangkap datagram data yang terdapat pada *switch* atau *hub*. Tabel ruting yang terdapat pada *switch* menjadi referensi dari aplikasi ini untuk menangkap datagram data dengan mengambil *IP Address* sumber dan target dari proses komunikasi pada sebuah *switch*. Datagram data dari suatu *IP Address* sumber dan target berisikan transaksi data. Transaksi data inilah yang mampu dianalisa dan ditampilkan menjadi penggalan informasi-informasi yang bisa dimengerti oleh pengguna. Misalkan pada proses pencarian inputan (*login*) dari sebuah aplikasi *form*, maka wireshark mampu mencari dan menampilkan informasi inputan yang telah dilakukan pada satu proses transaksi login dari sebuah IP Address.

Berikut ini adalah tampilan dari aplikasi wireshark saat melakukan analisa suatu *network adapter*.

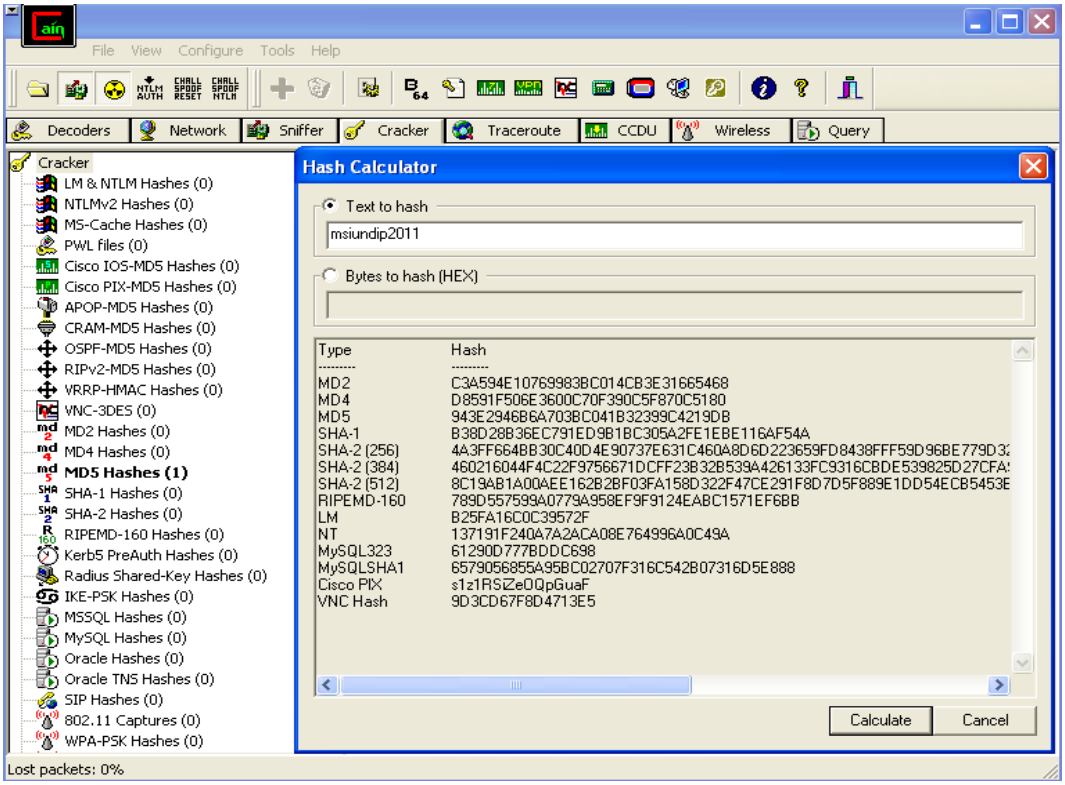


Gambar 2.7. Tampilan Aplikasi Wireshark

2.2.15. Aplikasi *Chain & Abel*

Chain & Abel merupakan aplikasi *sniffing* yang dirancang khusus untuk melihat dan menganalisa *password* serta enkripsi dan dekripsi dari berbagai algoritma kriptografi dengan metode membuat *Dictionary*, *Brute-Force* dan *Kriptanalisis* (http://www.oxid.it/cain.html). Sedikit berbeda dengan wireshark, aplikasi ini dirancang khusus hanya untuk sistem operasi Windows. Wireshark berfungsi menampilkan datagram data yang berhasil di tangkap dan menampilkan informasi-informasi yang di butuhkan oleh pengguna, sedangkan chain & abel dikhususkan untuk menangkap data login dan password serta data-data yang ter-enkripsi.

Pada *chain & abel* versi terakhir terdapat fungsi kalkulator yang berfungsi untuk mendekripsikan kembali data atau password yang terenkripsi. Hampir semua enkripsi dengan algoritma fungsi Hash generasi terakhir mampu di dekripsikan oleh aplikasi ini, sehingga aplikasi *chain & abel* dapat dijadikan sebagai *cracking* yang ampuh untuk menterjemahkan (mendekripsikan) *password* dan data yang terenkripsi.



Gambar 2.8. Tampilan Aplikasi Chain & Abel.

BAB III

CARA PENELITIAN

3.1. Bahan Penelitian

Bahan penelitian yang digunakan dalam proses penelitian tentang pengamanan dokumen elektronik pada Sistem Informasi Akademik (SIA) menggunakan *digital signature* dengan algoritma kurva eliptik sebagai berikut :

3.1.1 Obyek Penelitian

Dalam penelitian ini, data yang diamankan adalah lembaran informasi akhir aplikasi SIA Universitas Negeri Padang (http://portal.unp.ac.id) seperti Bio Data Mahasiswa Baru, Kartu Rencana Studi (KRS) baik sementara dan permanen, Laporan Hasil Studi (LHS), Transkrip Nilai Sementara, Transkrip Nilai Permanen baik dalam format elektronik (*e-paper*) berupa file .pdf maupun format tercetak dengan printer (*paperless*).

3.1.2. Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini adalah sebagai berikut:

3.1.2.1.Observasi

Merupakan metode pengumpulan data dengan cara melakukan pengamatan secara langsung pada obyek yang diteliti yaitu hasil cetakan SIA UNP, baik yang berupa file maupun dalam media yang telah tercetak, pada lembaran ini akan di modifikasi yaitu dengan memberikan *digital signature* pada salah satu bagian pada lembaran yang di cetak berupa kode angka dan barcode. Contoh salah satu lembaran cetakan SIA berupa lembar hasil studi seperti pada gambar berikut:

Universitas Negeri Padang
Fakultas Teknik

LEMBAR HASIL STUDI
Semester: Genap 2009 / 2010

Nama Mahasiswa	:	Yulia Eka Syafitri
NIM	:	74271
Angkatan	:	2006
Program Studi	:	Pendidikan Kesejahteraan Keluarga
Dosen PA	:	

No.	Matakuliah			SKS	KE	Nilai	Bobot	Nilai SKS
	Seksi	Kode	Nama					
1	72388	FTE007	Statistika	2	1	B	3.00	6
2	72384	FTE107	Metode Mengajar Khusus 2	3	1	E	0.00	0
3	72394	FTE034	Tata Tulis Karya Ilmiah dan Seminar	2	1	B	3.00	6
4	49224	KKE046	Kewirausahaan	2	1	B	3.00	6
5	49079	KKE081	Gambar Anatomi	2	1	B	3.00	6
6	63227	UNP104	Profesi Kependidikan	3	1	B	3.00	9
7	63191	UNP101	Pengantar Pendidikan	3	1	A	4.00	12
8	62895	UNP003	Pendidikan Kewarganegaraan	2	1	B	3.00	6
Jumlah				**19**				**51**

IP Semester (IPS) : 2.68
IP Kumulatif (IPK) : 2.33
Beban SKS Semester Genap 2010/2011 : 22

Catatan : Nilai BL Semester Lalu dan Nilai T tidak mempengaruhi jumlah SKS dan IP Semester

Kabag. Pendidikan dan
Kerjasama,

Azhari Suwir, SE
NIP : 130780909

Gambar 3.1. Dokumen Hasil Cetakan SIA UNP (LHS)

3.1.2.2.Studi Pustaka

Merupakan metode pengumpulan data dengan cara mengumpulkan data-data dari berbagai sumber yang mendukung penelitian ini dengan melakukan pencarian informasi di internet maupun berupa sumber buku, jurnal ilmiah, makalah, prosiding maupun artikel lainnya yang mendukung penelitian. Hasil dari studi pustaka berupa teori dan perkembangan terkini mengenai kriptografi, tanda tangan digital, keamanan sistem informasi dan teori pendukung lainnya.

3.2. Alat Penelitian

Dalam penelitian ini diperlukan peralatan perangkat lunak dan perangkat keras sebagai berikut:

3.2.1. Perangkat Lunak:

1. Web Server, yang digunakan dalam penelitian ini adalah XAMPP for Windows versi 1.7.3 dengan versi PHP My Admin 3.2.4 dan My SQL Versi Server: 5.1.41 bawaan dari XAMPP.
2. Sistem Operasi yang digunakan adalah Windows XP Profesional 32 Bit.
3. PHP editor yang digunakan adalah Adobe Dreamweaver CS4.
4. MySql editor yang digunakan adalah Navicat (MySql GUI) versi 7.1.14
5. *PDF creator* dan *viewer* Nitro PDF Professional versi 5.3.1.8 atau PDF creator lainnya yang telah terinstal pada sistem printer di windows.

3.2.2. Perangkat Keras:

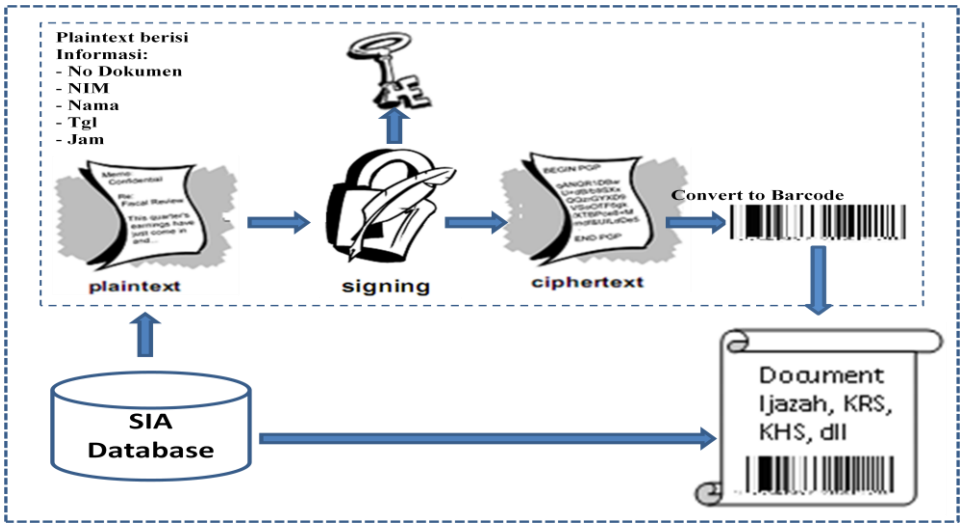
1. PC-P4/*Compatible*/Notebook (Intel P4/1,6G/Memory 1G)
2. *Barcode Reader* (Jenis Laser)
3. Printer, tinta dan kertas.

3.3. Jalan Penelitian

Adapun jalannya penelitian tentang pengamanan dokumen elektronik pada Sistem Informasi Akademik UNP menggunakan *digital signature* dengan algoritma kurva eliptik, maka dibuat skema seperti pada gambar 3.2 dan 3.3. Sistem rancangan pada tesis ini ada dua bagian, pertama sistem hasil modifikasi Aplikasi SIA yang telah ada yaitu dengan menambahkan kode *digital signature* dan *barcode*. Dalam hal ini peneliti hanya mendapatkan

izin untuk memodifikasi SIA hanya dari sisi *client*, yaitu dengan cara mendownload setiap tampilan saat menjalankan aplikasi SIA sedangkan database dan data (informasi lainnya) merupakan rancangan dari peneliti. Pada aplikasi yang kedua, yaitu aplikasi pembaca keabsahan merupakan hasil rancangan dari peneliti. Kedua aplikasi ini secara umum prinsip kerjanya adalah sebagai berikut:

1. Aplikasi pembentuk tanda tangan (*Created Signature*)

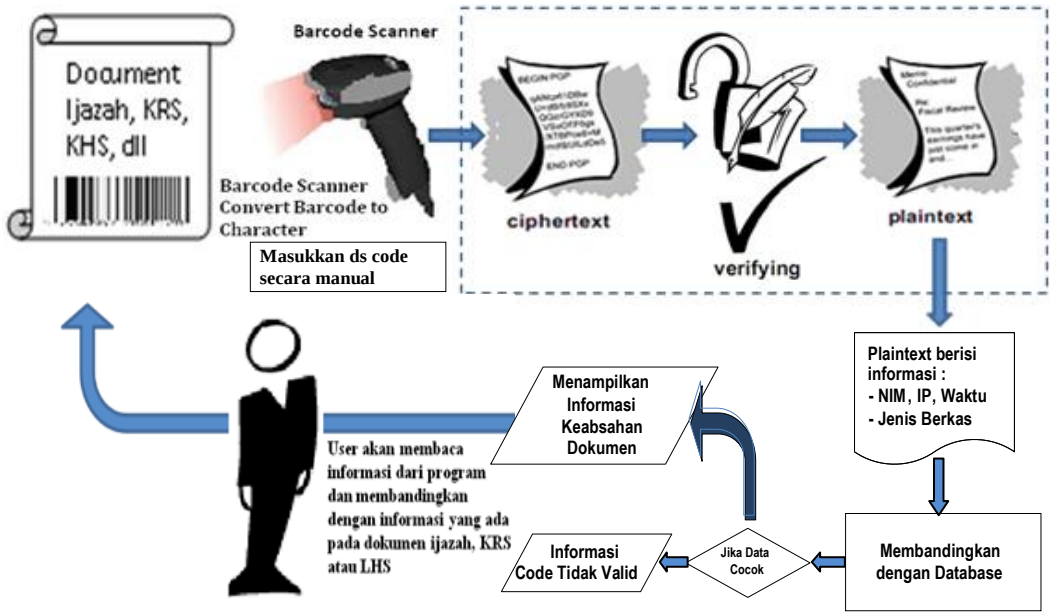


Gambar 3.2. Diagram Proses Pembuatan Digital Signature pada Dokumen

Pada gambar 3.2. terjadi proses pembentukan tanda tangan. Informasi plaintext yang diambil dari database SIA yang menjadi referensi adalah NIM, index prestasi, jenis dokumen dan waktu cetak dokumen. Informasi ini pada saat perintah cetak di eksekusi maka keempat informasi ini akan di *signing* atau dikalkulasikan dengan metode kurva eliptik dan menghasilkan *chipertext* atau kode *digital signature* (Kode ds) dengan kunci publik *r* dan kunci private *s*. Informasi akademik (nama, nim, program studi, penasehat akademik, tahun masuk, jenis dokumen, ip dan waktu cetak) dari user akan digunakan kembali pada aplikasi pembaca keabsahan tanda tangan digital juga tersimpan pada tabel database. *Chipertext* (Kode ds) hasil enkripsi di konversikan ke dalam format barcode versi 128, kode ds dan barcode akan tercetak bersamaan dengan lembaran informasi dari aplikasi SIA. Barcode ini akan digunakan untuk dibaca kembali oleh *barcode reader* pada proses pembacaan keabsahan tanda tangan

digital. *Barcode reader* berfungsi menggantikan *keyboard* yang diketik dengan tangan, berguna untuk mengurangi kesalahan pengetikan *chipertext* yang memiliki banyak karakter.

2. Aplikasi pembaca keabsahan tanda tangan digital



Gambar 3.3. Diagram Proses Otentifikasi *Digital Signature* pada Dokumen

Pada gambar 3.3, terlihat proses pembacaan keabsahan tanda tangan. Aplikasi pembaca keabsahan tanda tangan membaca nilai yang dimasukkan dari kode ds (*digital signature code*) pada menu inputan atau dengan menggunakan *barcode reader* yang akan membaca barcode dari lembaran dokumen SIA. Nilai yang terbaca merupakan nilai *ciphertext* yang kemudian pada proses eksekusi aplikasi pembacaan akan di-*dekripsi*-kan kembali (proses *verifying*) dengan metode kurva eliptik (kebalikan dari proses pembuatan tanda tangan) dengan kunci publik dan akan menghasilkan kode *plaintext*. Kode *plaintext* ini dikalkulasikan menjadi informasi NIM, IP, jenis dokumen dan waktu cetak dokumen.

Keempat informasi ini dibandingkan dengan keempat *field* database digital signature NIM, jenis dokumen, waktu cetak dokumen dan IP (Index Prestasi), jika cocok maka informasi yang diperlukan akan ditampilkan, dan jika tidak cocok, maka akan ditampilkan informasi data tidak ada atau pemberitahuan tentang status ketidak cocokan dokumen (dokumen palsu).

3.3.1. Pengembangan Aplikasi dengan Metode *Waterfall*

Pada penelitian tesis ini peneliti akan membangun suatu sistem pengamanan dokumen elektronik pada Sistem Informasi Akademik (SIA) menggunakan *digital signature* dengan algoritma kurva eliptik yang terdiri dari modifikasi perangkat lunak SIA yang telah ada dan membuat baru perangkat lunak pembaca keabsahan tanda tangan. Sedangkan metode yang digunakan untuk membangun perangkat lunak adalah dengan metode *waterfall* yaitu suatu metode pengembangan *software* yang bersifat sekuensial dan terdiri dari 5 tahapan yang saling terkait dan mempengaruhi. Kelima tahapan tersebut yaitu (Kendal: 2002):

3.3.1.1. Analisa Kebutuhan (*Requirement Analysis*)

Pada tahapan awal penelitian ini dilakukan pengumpulan informasi tentang Sistem Informasi Akademik baik yang di UNP maupun kampus lainnya, model-model pengamanan sistem informasi, pengamanan dokumen elektronik, algoritma kriptografi tentang kelebihan dan kekurangannya, model *digital signature*. Kemudian dilanjutkan dengan mengidentifikasi masalah pada kasus pengamanan sistem informasi akademik dan dilanjutkan dengan perumusan masalah serta solusi alternatif untuk mengatasi masalah tersebut. Dalam hal ini pengamanan dengan membuat *digital signature* pada setiap transaksi dokumen elektronik pada SIA yang tercetak. *Digital Signature* yang dirasakan aman dengan kecepatan relatif tinggi dengan menggunakan algoritma kurva elipstik.

Pada sistem yang akan di rancang nanti ada 2 aplikasi, pertama aplikasi SIA yang telah ada dengan dimodifikasi dan ditambahkan pembuat tanda tangan pada saat perintah cetak dokumen SIA dieksekusi, tanda tangan menggunakan enkripsi dengan algoritma kurva elipstik. Aplikasi kedua merupakan aplikasi pembaca keabsahan tanda tangan dengan meng-dekripsi kembali tanda tangan dari lembar dokumen SIA yang telah tercetak.

3.3.1.2. Perancangan Sistem (*System Design*)

Disain atau perancangan sistem adalah proses penterjemahan rancangan sistem sesuai dengan algoritma yang telah direncanakan pada tahap awal. Pada tahapan perencanaan dilakukan penyusunan proses sistem, aliran proses dan hubungan antar kelompok program sehingga didapatkan perancangan yang paling optimal. Pada tahapan ini akan dihasilkan *flowchart* sebagai model perencanaan dari sistem yang akan di bangun. Secara garis besar ada 2 *flowchart* yang akan di rancang, pertama *flowchart* untuk menghasilkan tanda tangan digital pada saat proses cetak pada aplikasi SIA, yang kedua flowchart untuk aplikasi pembaca keabsahan tanda tangan. Selain *flowchart* tersebut juga dirancang *flowchart* untuk menjelaskan proses enkripsi.

Setelah *flowchart* dihasilkan, langkah selanjutnya adalah disain dan perancangan tampilan. Pada tahapan ini, disain tampilan hanya dilakukan pada aplikasi pembaca keabsahan tanda tangan (aplikasi perifying). Sedangkan pada aplikasi untuk menghasilkan tanda tangan hanya menggunakan tampilan yang ada pada SIA UNP, modifikasi hanya dilakukan pada perintah cetak yaitu dengan menyisipkan sintak untuk menghasilkan tanda tangan. Untuk lebih jelasnya pembahasan tantang perancangan sistem ini akan dibahas pada sub bab 3.3.2.

3.3.1.3. Pembuatan Kode Program (*Implementation*)

Proses pembuatan kode program merupakan proses implementasi atau proses penulisan kode program dengan menggunakan bahasa pemograman PHP dan database menggunakan MySql sesuai dengan blok-blok *flowchart* (aliran program) yang ditetapkan pada langkah disain. Dalam koding juga dibuat antarmuka sistem untuk aplikasi pembacaan keabsahan tanda tangan guna mempermudah interaksi antara sistem dengan user.

3.3.1.4. Pengujian (*Testing*)

Pengujian dilakukan untuk melihat sistem yang telah dibuat apakah bekerja sesuai dengan tujuan program yang telah ditentukan atau tidak, dengan memberikan masukan seperti layaknya program dijalankan oleh user atau dengan memberikan perlakuan khusus seperti masukan-masukan yang semestinya dilakukan oleh user. Pengujian tingkat keamanan dan kehandalan aplikasi digunakan program pengendus (*sniffer*) *wireshark* dan *chain & abel* yang akan menangkap aliran datagram pada saat proses pengujian keabsahan tanda tangan dijaringan pada saat user menginputkan dan menjalankan aplikasi pembaca keabsahan tanda tangan (*perifying*). Pengujian kinerja aplikasi pembaca keabsahan tanda tangan dengan *tools firebug* yang merupakan alat tambahan dari *browser firefox*. *Firebug* ini berfungsi untuk melihat waktu akses dan proses sebuah laman website. Hasil pengujian untuk setiap kasus akan diamati dan dianalisa untuk diambil hasilnya dan digunakan sebagai acuan untuk perbaikan dan pengembangan sistem pada tesis ini.

3.3.1.5. Penerapan Sistem (*Implementation*)

Penerapan program merupakan tahapan dimana peneliti menerapkan atau menjalankan program ke sistem yang mendekati kondisi sebenarnya yaitu dengan menjalankan serfis *web server* untuk mengaktifkan aplikasi yang menggunakan *script* PHP dan database MySql sehingga pada sisi *client* aplikasi yang dijalankan bisa aktif dengan benar untuk kedua aplikasi yang telah di bangun.

3.3.2. Perancangan Aplikasi

Sistem pada tesis ini secara umum dibangun dari dua buah aplikasi yang bekerja terpisah, pertama aplikasi untuk menghasilkan tanda tangan dan kedua aplikasi penguji keabsahan tanda tangan dengan menggunakan satu database yang tersimpan di server. Pondasi utama dari sistim ini adalah algoritma kurva eliptik yang digunakan untuk menghasilkan tanda tangan yaitu proses enkripsi data referensi dalam hal ini NIM, IP, Jenis Dokumen dan waktu cetak di kalkukasi dengan algoritma kurva eliptik dan akan menghasilkan kode digital signature dan sepasang kunci. Begitu juga pada proses sebaliknya (pada aplikasi penguji keabsahan tanda tangan) terjadi proses dekripsi data dari input *chipertext* yang dimasukkan kemudian data di dekripsi dan akan menghasilkan *plaintext* yaitu nilai nim, ip, jenis dokumen dan waktu cetak. Keempat data ini kemudian dibandingkan dengan data di database, jika keempat data sama, maka dokumen benar dan jika tidak maka sebaliknya.

3.3.2.1. Perancangan Model *Digital Signature* Algoritma Kurva Ellipstic

Algoritma kurva eliptik yang digunakan merupakan algoritma pengembangan dari algoritma tanda tangan digital sebelumnya, seperti RSA, DSA. Digunakannya algoritma kurva eliptik pada penelitian tesis ini dikarenakan tingkat kesulitan yang tinggi untuk lebar bit yang rendah. Seperti yang telah di bahas pada bab 2 bahwa algoritma kurva eliptik hanya membutuhkan 160 bit untuk tingkat keamanan yang sama pada algoritma RSA yang membutuhkan panjang bit sebesar 1024 (perbandingan di http://www.rsa.com/rsalabs/ node.asp?id=2013).

Proses pembuatan tanda tangan, enkripsi, hingga pengujian keabsahan tanda tangan dengan algoritma kurva eliptik menurut aturan standar (**certicom,** 2000) dalam aturan pada prosedur algoritma kurva elipstik dari proses pembentukan tanda tangan dan pengujian keabsahan tanda tangan adalah sebagai berikut:

1. Prosedur penentuan kunci.

Setiap pengguna SIA pada saat melakukan transaksi (mencetak dokumen SIA) akan menghasilkan *public key* dan *private key* yang akan digunakan juga pada proses pembacaan keabsahan tanda tangan. Langkah-langkah proses pembuatan kedua kunci ini adalah:

a. Menentukan sebuah bilangan bulat random d_A, yang nilainya diantara *[1,n-1]*

b. Menghitung $Q_A = d_A * G \rightarrow G[(x_1, y_1)]$ dengan $y^2 = x^3 + ax + b$ (mod p).

c. Kunci rahasia = d_A , dan kunci publik = Q_A

2. Prosedur pembangkitan tanda tangan (*Signing*):

Langkah-langkah yang dilakukan pada proses signing atau pembentukan tanda tangan adalah sebagai berikut:

a. Memilih sebuah bilangan bulat random *k*, yang nilainya diantara *[1, n-1]*

b. Menghitung $Q_A = k * G = (x_1,y_1)$ *dan* $r = x_1 \bmod n$,

jika r = 0 maka kembali ke langkah 1

c. Menghitung $k^{-1} \bmod n$

d. Menghitung $e = HASH(m)$

e. Menghitung $s = k^{-1} \{e+d_A * r\} \bmod n$

Tanda tangan untuk *massage m* adalah *(r, s)*

3. Prosedur verifikasi keabsahan tanda tangan (*Verifing*)

Setelah tanda tangan dihasilkan (proses *signing*), Algoritma selanjutnya yang diperlukan adalah pengujian keabsahan tanda tangan (*verifying*). Algoritma verifying adalah sebagai berikut:

a. Memverifikasi bahwa *r* dan *s* adalah bilangan bulat antara *[1, n-1]*

b. Menghitung $e = HASH(m)$

c. Menghitung $w = s^{-1} \bmod n$

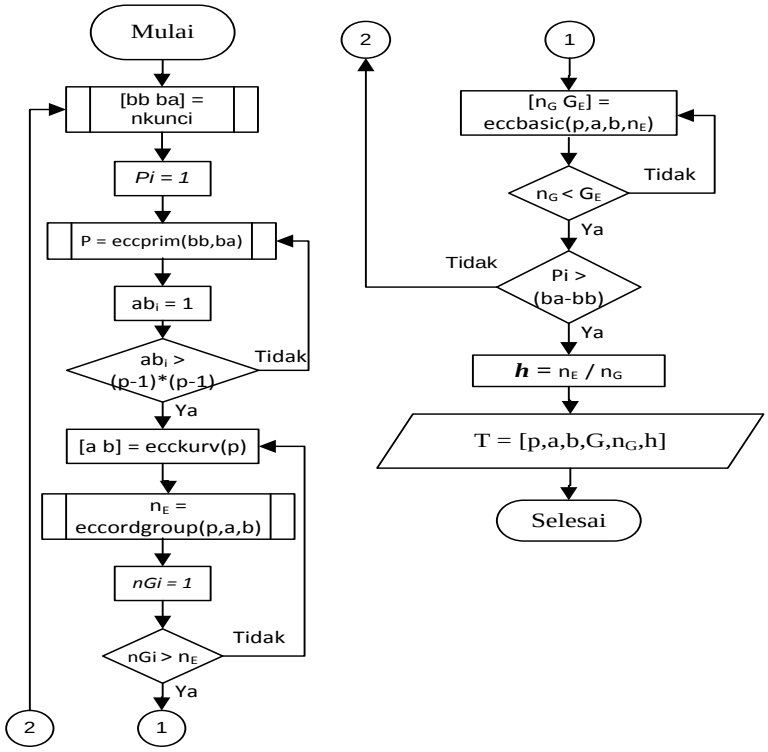
d. Menghitung $u_1 = ew \bmod n$ *dan* $u_2 = rw \bmod n$

e. Menghitung $u_1 * G + u_2 * Q_A = (x_1, y_1)$

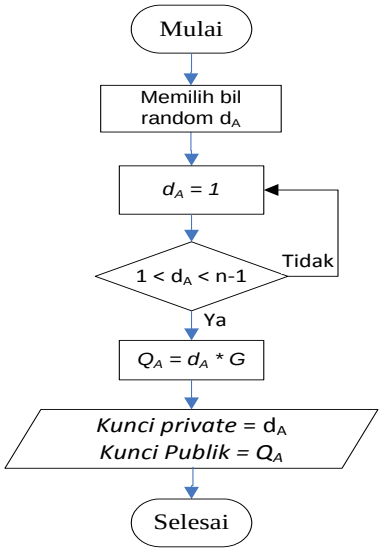
f. Menghitung $v = x_1 \bmod n$

Jika v = r, maka tanda tangan adalah sah

Maka aturan di atas dikonversikan menjadi sebuah *flowchart* seperti gambar-gambar berikut ini. Sebelum menghasilkan *flowchart* untuk penentuan *private key* dan *public key* perlu ditentukan dibuat sebuah fungsi untuk menentukan parameter domain kurva eliptik seperti pada *flowchart* berikut ini:

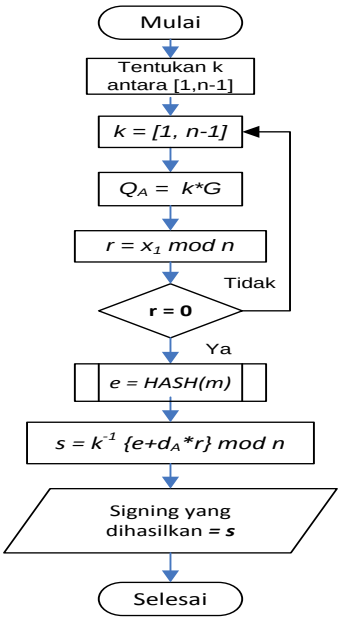


Gambar 3.4. Flowchart fungsi untuk mencari parameter domain kurva eliptik



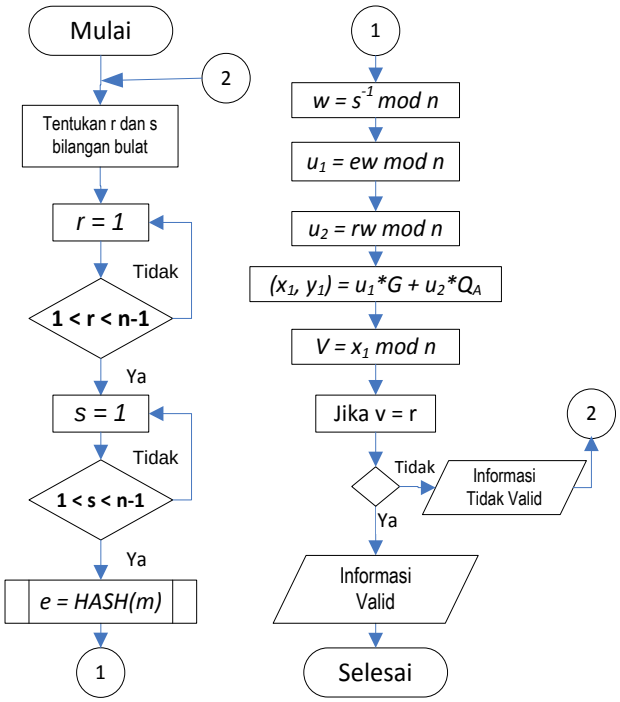
Gambar 3.5. Flowchart fungsi penentuan *Public Key* dan *Private Key*

Setelah *flowchart* pembangkit sepasang kunci dihasilkan, maka proses selanjutnya adalah menghasilkan tanda tangan digital (*signing*) seperti yang dilihatkan pada *flowchart* berikut ini:



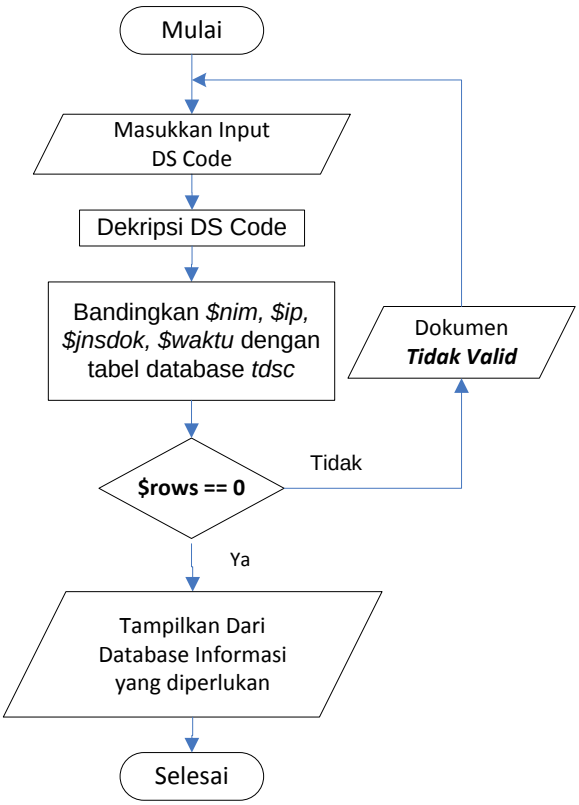
Gambar 3.6. Flowchart proses *signing*

Pada aplikasi berikutnya yaitu aplikasi pembaca keabsahan tanda tangan, terjadi proses dekripsi atau pembacaan kembali keabsahan tanda tangan (*verifying*). *Flowchart* yang menggambarkan proses *verifying* adalah sebagai berikut:



Gambar 3.7. Flowchart proses *verifying*

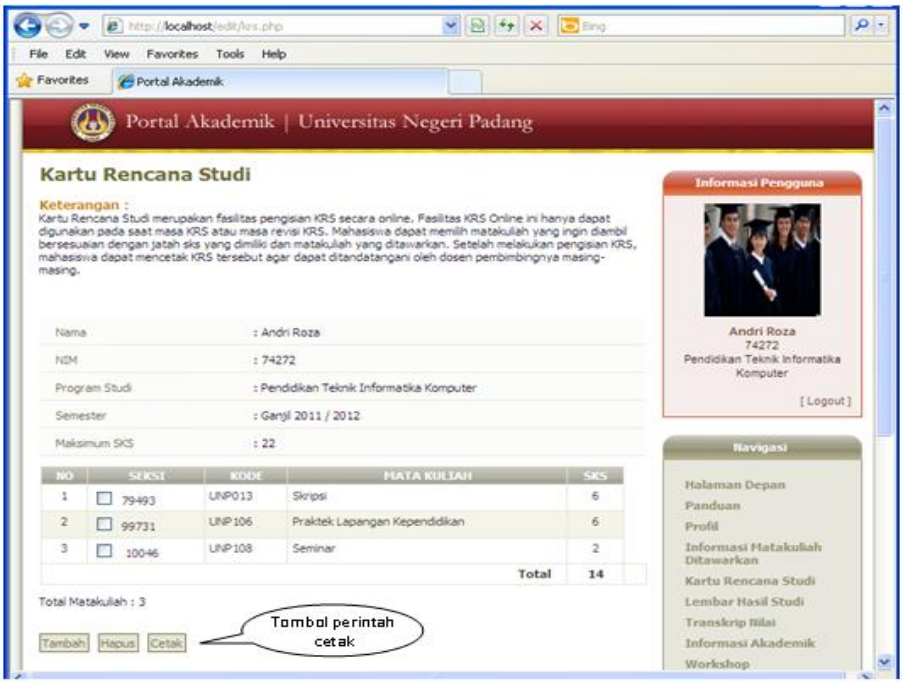
Sedangan *flowchart* untuk program pembaca keabsahan tanda tangan dari dokumen secara keseluruhan adalah sebagai berikut:



Gambar 3.8. Flowchart program pembacaan keabsahan tanda tangan keseluruhan

3.3.2.2.Perancangan Model Tampilan

Perancangan model tampilan atau antar muka program berguna untuk interaksi antara pengguna dengan aplikasi. Terdapat dua tampilan dari sistem yang dirancang, pertama tampilan pada perintah cetak SIA UNP. Untuk tampilan ini peneliti tidak melakukan perubahan tampilan, hanya menyisipkan perintah *coding generate digitalsignature* pada perintah cetak seperti pada gambar berikut:



Gambar 3.9. Tampilan Aplikasi SIA UNP untuk menjalankan perintah cetak

Universitas Negeri Padang

Fakultas Teknik

KARTU RENCANA STUDI

Semester: Ganjil 2011 / 2012

NIM : 74272
Nama : Andri Roza
Program Studi : Pendidikan Teknik Informatika Komputer
Dosen PA : Ahmaddul Hadi, S.Pd. (5326)
Tahun Masuk : 2007

No.	Seksi	Matakuliah		SKS	Jadwal							Dosen 1
		Kode	Nama		Sn	Sl	Rb	Km	Jm	Sb	Mg	
1	79493	UNP013	Skripsi	6								5633
2	99731	UNP106	Praktek Lapangan Kependidikan	6								1413
3	10046	UNP108	Seminar	3						EP5 07:00-11:30		5625
Total :				15								

IP Semester Lalu : 3.00
Max Sks : 22
Waktu Cetak : 10:15:00
DS Code : fd43f5054a2e6190d32cb79e8ee5a243

Catatan: Jadwal yang masih kosong silakan kontak Jurusan

Padang, 21-September-2011

Mengetahui
a.n. Dosen PA

Mahasiswa

Untuk Melihat Keaslian Dokumen, Silahkan buka laman http://localhost/edit/cek_ds.php

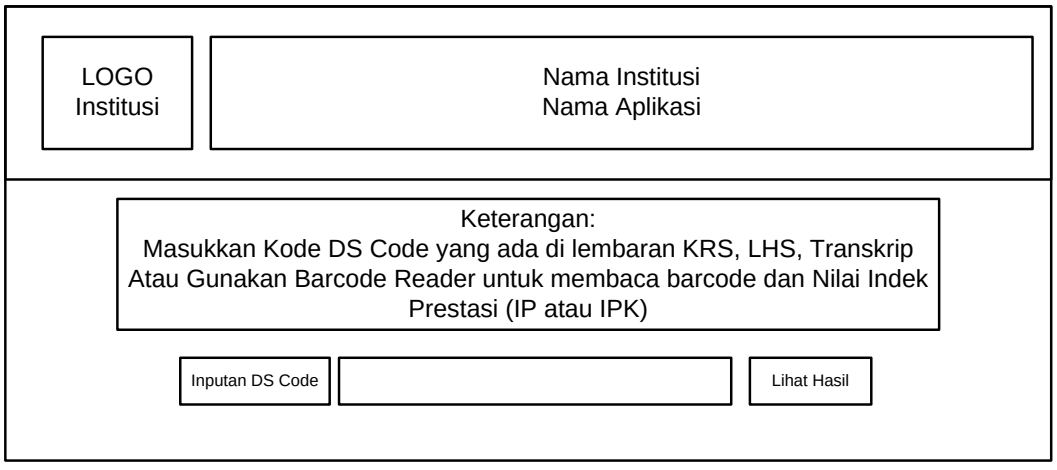
Ahmaddul Hadi, S.Pd. (5326)

Andri Roza

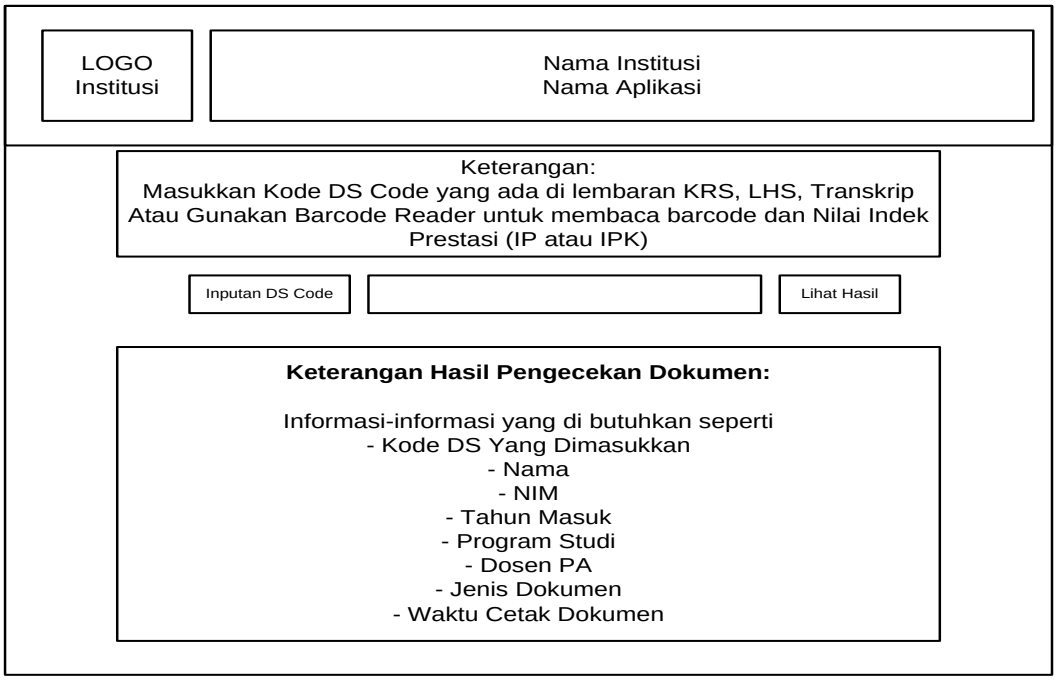
BarCode:

Gambar 3.10. Tampilan Hasil Cetakan Dengan Tampilan DS Code

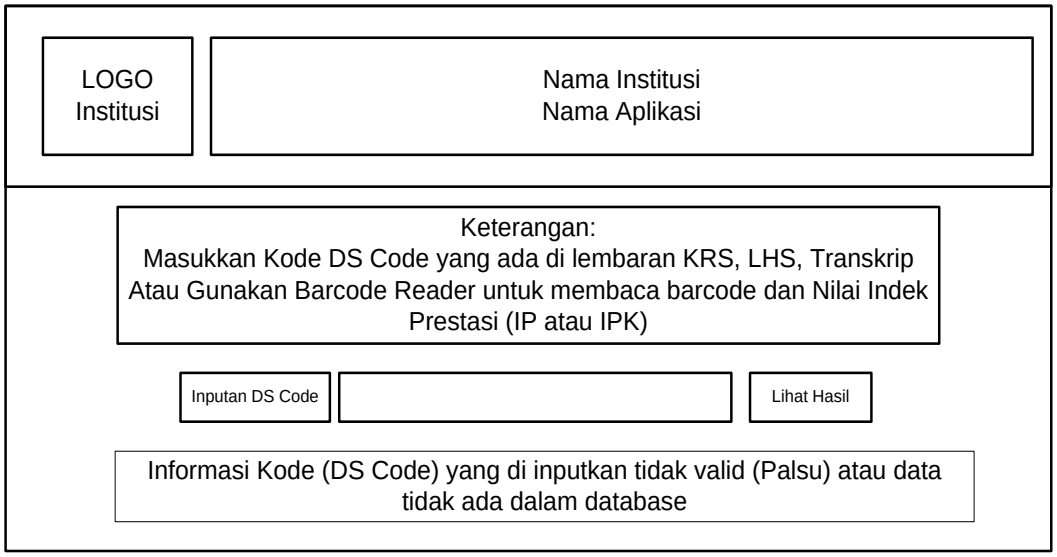
Sedangkan pada tampilan aplikasi pengecekan keabsahan tanda tangan, rancangan tampilan aplikasi adalah sebagai berikut:



Gambar 3.11. Rancangan Tampilan Aplikasi Pembaca Keabsahan



Gambar 3.12. Rancangan Tampilan Aplikasi Pembaca Keabsahan Jika Valid



Gambar 3.13. Rancangan Tampilan Aplikasi Pembaca Keabsahan Jika Tidak Valid