Bahan Kuliah ke-23

IF5054 Kriptografi

Public Key Infrastructure (*PKI*)

Disusun oleh:

Ir. Rinaldi Munir, M.T.

Departemen Teknik Informatika
Institut Teknologi Bandung
2004

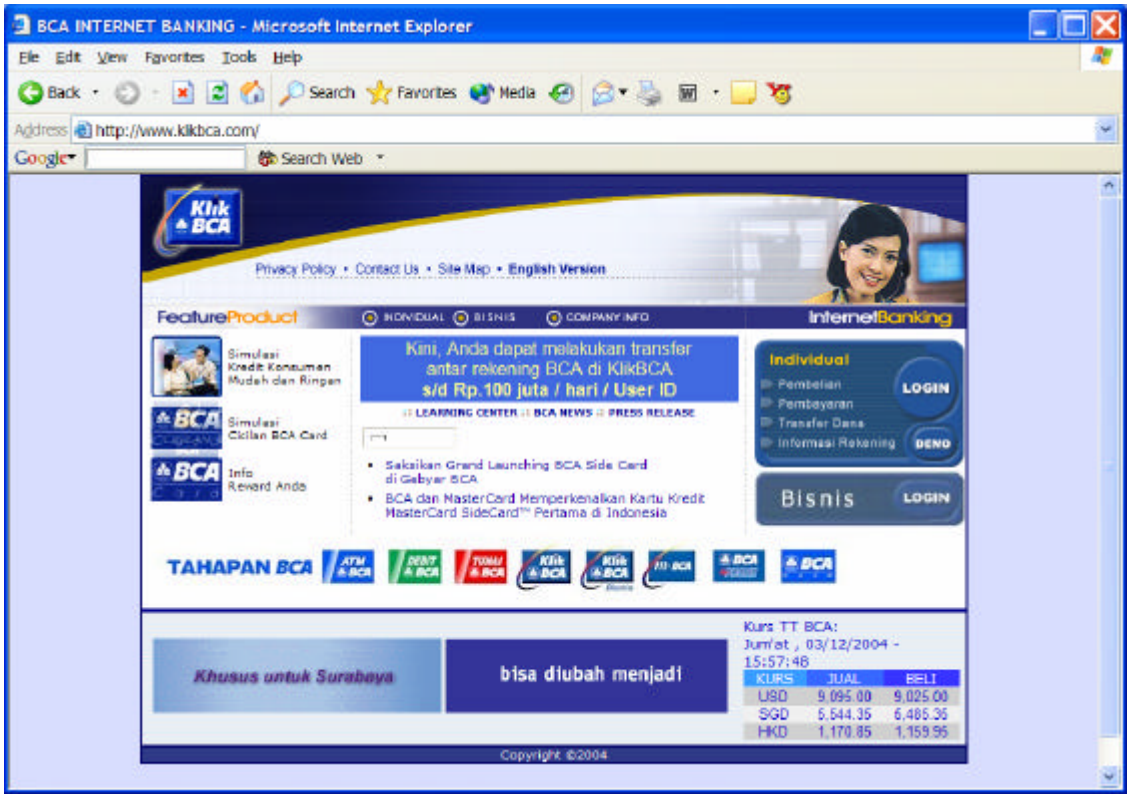
23. *Public Key Infrastructure* (PKI)

23.1 Sertifikat Digital

- Serangan yang umum terjadi pada kunci publik tanpa identitas adalah penyamaran (*impersonation attack*). Seseorang yang memiliki kunci publik orang lain dapat menyamar seolah-olah dialah pemilik kunci itu. Serangan semacam ini adalah masalah yang muncul dari penggunaan kriptografi kunci-publik.

- Contohnya, dalam teknologi *e-commerce*, pembayaran transaksi dilakukan dengan menggunakan kartu kredit. Pelanggan mengirimkan informasi kartu kreditnya (yang bersifat rahasia) melalui *website* pedagang *online*. Selama pengiriman, informasi kartu kredit tersebut dilindungi dengan cara mengenkripsinya dengan kunci publik pedagang *online*.

  Bagaimana pelanggan itu memastikan bahwa *website* pedagang *online* tersebut memang benar milik pedagang *online* dan bukan *website* pihak lain yang menyamar sebagai *website* pedagang asli dengan tujuan untuk mencuri informasi kartu kredit (ingat kasus *klikbca.com*). Gambar 23.1 memperlihatkan website `www.klikbca.com`.



Gambar 23.1 *Website* **www.klikbca.com**

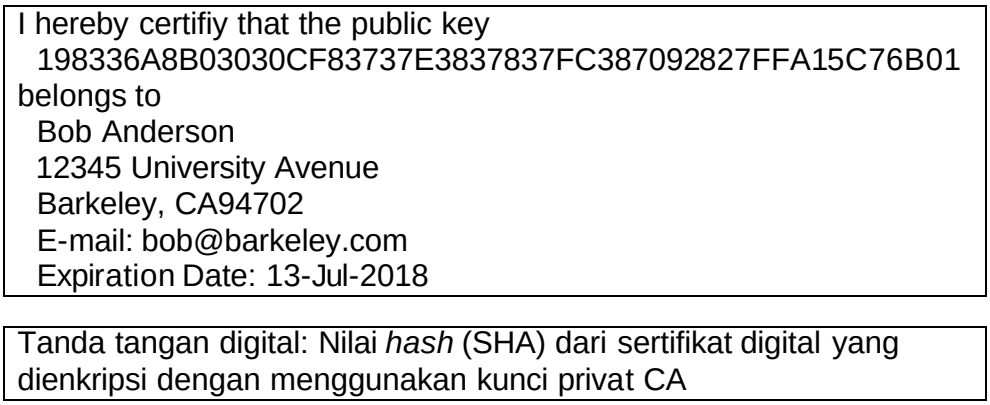
- Untuk menjawab masalah di atas, solusinya adalah dengan memberikan **sertifikat digital** pada kunci publik. Sertifikat digital dikeluarkan (*issued*) oleh pemegang otoritas sertifikasi (*Certification Authority* atau *CA*).

- *CA* biasanya adalah institusi keuangan (seperti bank) atau institusi yang terpercaya.

- Sertifikat digital adalah dokumen digital yang berisi informasi sebagai berikut:
  - nama subjek (perusahaan/individu yang disertifikasi)
  - kunci publik si subjek
  - waktu kadaluarsa sertifikat (*expired time*)
  - informasi relevan lain seperti nomor seri sertifikat, dll

- *CA* membangkitkan nilai *hash* dari sertifikat digital tersebut (misalnya dengan fungsi *hash* satu-arah *MD5* atau *SHA*), lalu menandatangani nilai *hash* tersebut dengan menggunakan kunci privat *CA*.

- Contoh sebuah sertifikat digital: Bob membawa kunci publiknya dan mendatangi *CA* untuk meminta sertifikat digital. *CA* mengeluarkan sertifikat digital dan menandatangani sertifikat tersebut dengan cara mengenkripsi nilai *hash* dari kunci publik Bob (atau nilai *hash* dari sertifikat digital keseluruhan) dengan menggunakan kunci privat *CA*. Contoh isi sertifikat digital dan tanda tangan digital dari *CA* kira-kira seperti Gambar 23.2.



Gambar 23.2 Contoh sebuah sertifikat digital

- Jadi, sertifikat digital mengikat kunci publik dengan identitas pemilik kunci publik. Sertifikat ini dapat dianggap sebagai 'surat pengantar' dari *CA*.

- Supaya sertifikat digital itu dapat diverifikasi (dicek kebenarannya), maka kunci publik *CA* harus diketahui secara luas. Seseorang yang memiliki kunci publik *CA* dapat memverifikasi bahwa tanda tangan di dalam suatu sertifikat itu sah dan karena itu mendapat jaminan bahwa kunci publik di dalam sertifikat itu memang benar.

- *VeriSign, Inc*. adalah *CA* yang terkemuka. Untuk informasi tentang *VeriSign*, kunjungi `www.verisign.com`

- Sertifikat digital sendiri tidak rahasia, tersedia secara publik, dan disimpan oleh *CA* di dalam *certificate repositories.* Salinan (*copy*) sertifikat tersebut juga dimiliki oleh pemohon sertifikat.

- Bob mungkin meletakkan sertifikat tersebut di dalam *homepage*-nya, dengan *link* ke halaman *web* yang menyatakan: *Klik ini untuk melihat sertifikat kunci publikku*. Hasil klik akan memperlihatkan sertifikat digital dan tanda-tangan dari *CA*.

- Misalkan Alice mengakses *homepage* Bob untuk mendapatkan kunci publik Bob. Misalkan Carol berhasil memintas (*ineterception*) *request* Alice (*client*) ke *homepage* Bob (*server*), sehingga *request* tersebut masuk ke *homepage* Bob palsu (yang dibuat oleh Carol) (Tujuan memintas adalah agar Alice mengira Carol adalah Bob, sehingga Carol dapat memperoleh informasi rahasia dari Alice, misalnya kunci).

  Carol sudah meletakkan sertifikat digitalnya di dalam halaman *web* palsu, tapi jika Alice membaca sertifikat tersebut dia langsung paham bahwa dirinya sedang tidak berkomunikasi dengan Bob asli karena identitas Bob tidak terdapat di dalam sertifikat tersebut.

- Misalkan Carol berhasil mengubah *homepage* Bob, mengganti kunci publik Bob di dalam sertifikat digital dengan kunci publiknya. Tetapi, jika Alice meng-*hash* sertifikat digital tersebut, dia memperoleh nilai *hash* yang tidak sama dengan nilai *hash* yang dihasilkan jika tanda-tangan digital diverifikasi dengan kunci publik *CA*.

  Karena Carol tidak mempunyai kunci privat *CA*, maka Carol tidak dapat membangkitkan tanda-tangan digital dari sertifikat Bob yang sudah diubah tersebut. Dengan cara ini, Alice dapat meyakini bahwa dia memiliki kunci publik Bob dan bukan kunci publik Carol. Lagipula, skema ini juga tidak membutuhkan *CA* harus *online* untuk melakukan verifikasi.

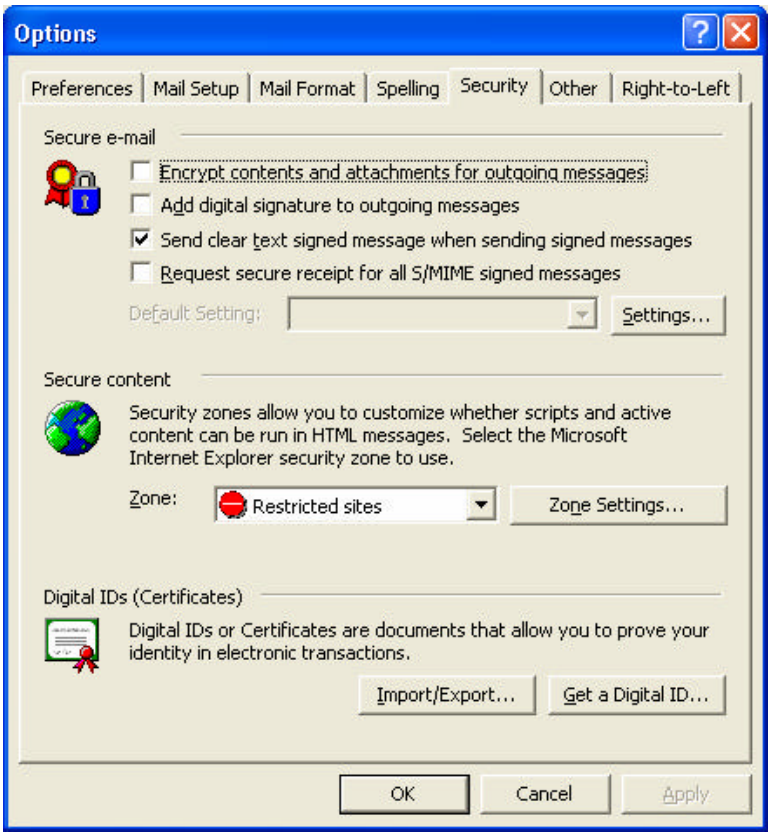
- Adanya atribut waktu kadualarsa pada sertifikat digital dimaksudkan agar pengguna mengubah kunci publik (dan kunci privat pasangannya) secara periodik. Makin lama penggunaan kunci, makin besar peluang kunci diserang dan dikriptanalisis. Jika pasangan kunci tersebut diubah, maka sertifikat digital yang lama harus ditarik kembali (*revoked*). Pada sisi lain, jika kunci privat berhasil diketahui pihak lain sebelum waktu kadualarsanya, sertifikat digital harus dibatalkan dan ditarik kembali, dan pengguna harus mengganti pasangan kuncinya.

  Bagaimana *CA* memberitahu ke publik bahwa sertifikat digital ditarik? Caranya mudah saja. *CA* secara periodik mengeluarkan *CRL* (*Certificate Revocation List*) yang berisi nomor seri sertifikat digital yang ditarik. Sertifikat digital yang sudah kadaluarsa otomatis dianggap sudah tidak sah lagi dan ia juga dimasukkan ke dalam *CRL*. Dengan cara ini, maka *CA* tidak perlu memberitahu perubahan sertifikat digital kepada setiap orang.

- Sayangnya, keberadaan *CRL* menyebabkan pengguna yang memakai sertifikat digital harus memiliki *CRL* untuk memvalidasi apakah sertifikat tersebut telah ditarik. Sebagai alternatif *CRL* adalah *Online Certificate Status Protocol* (*OCSP*), yang memvalidasi sertifikat secara *real time.*

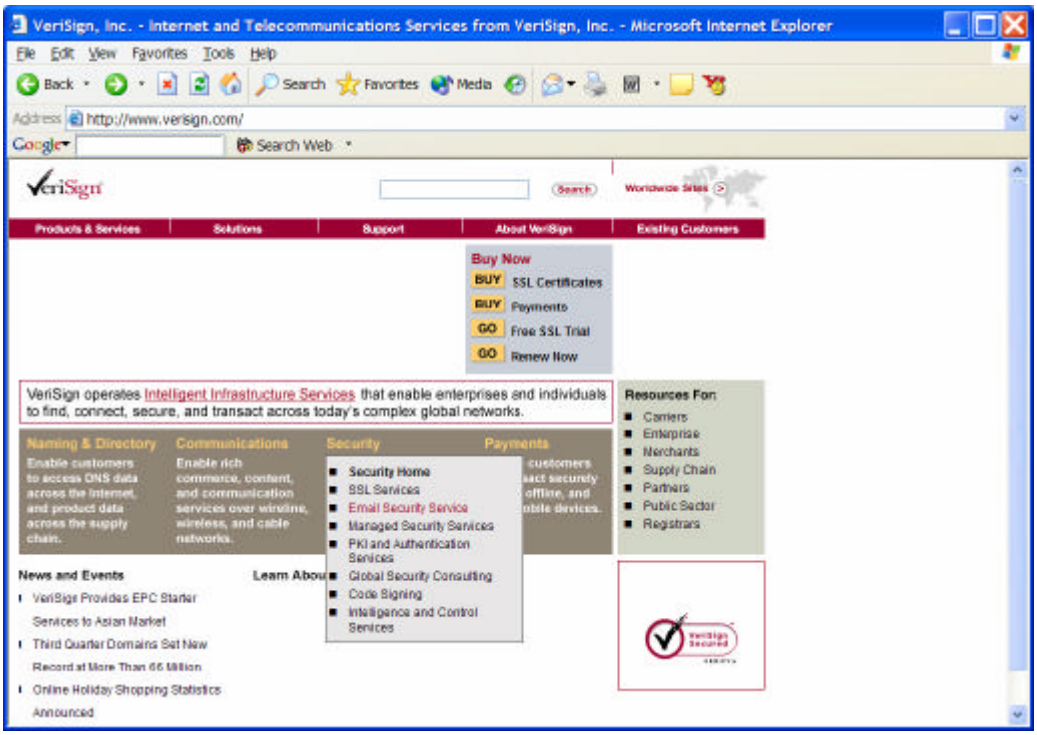
- Sertifikat digital dapat digunakan untuk keamanan *e-mail*. Sebagai contoh, di dalam *Microsoft Outlook*, pilih

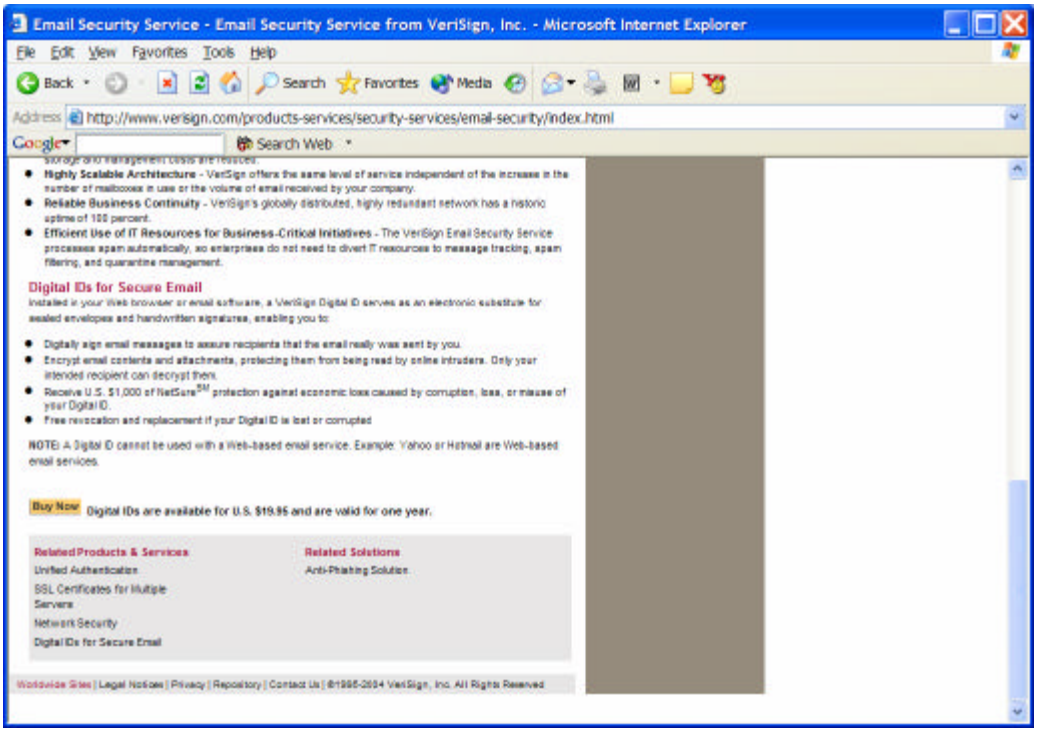
  Tools → *Options* → *Security*



Pada bagian bawah kotak dialog, anda akan melihat opsi untuk memperoleh *digital ID*. Dengan memilih opsi tersebut, anda akan dibawa ke situs *web Microsoft* dengan *link* ke beberapa *CA*. Sekali anda mempunyai sertifikat digital, anda dapat menandatangani *e-mail* secara digital

Untuk memperoleh sertifikat digital untuk *e-mail* pribadi, kunjungi `www.verisign.com` atau `www.thawte.com`.





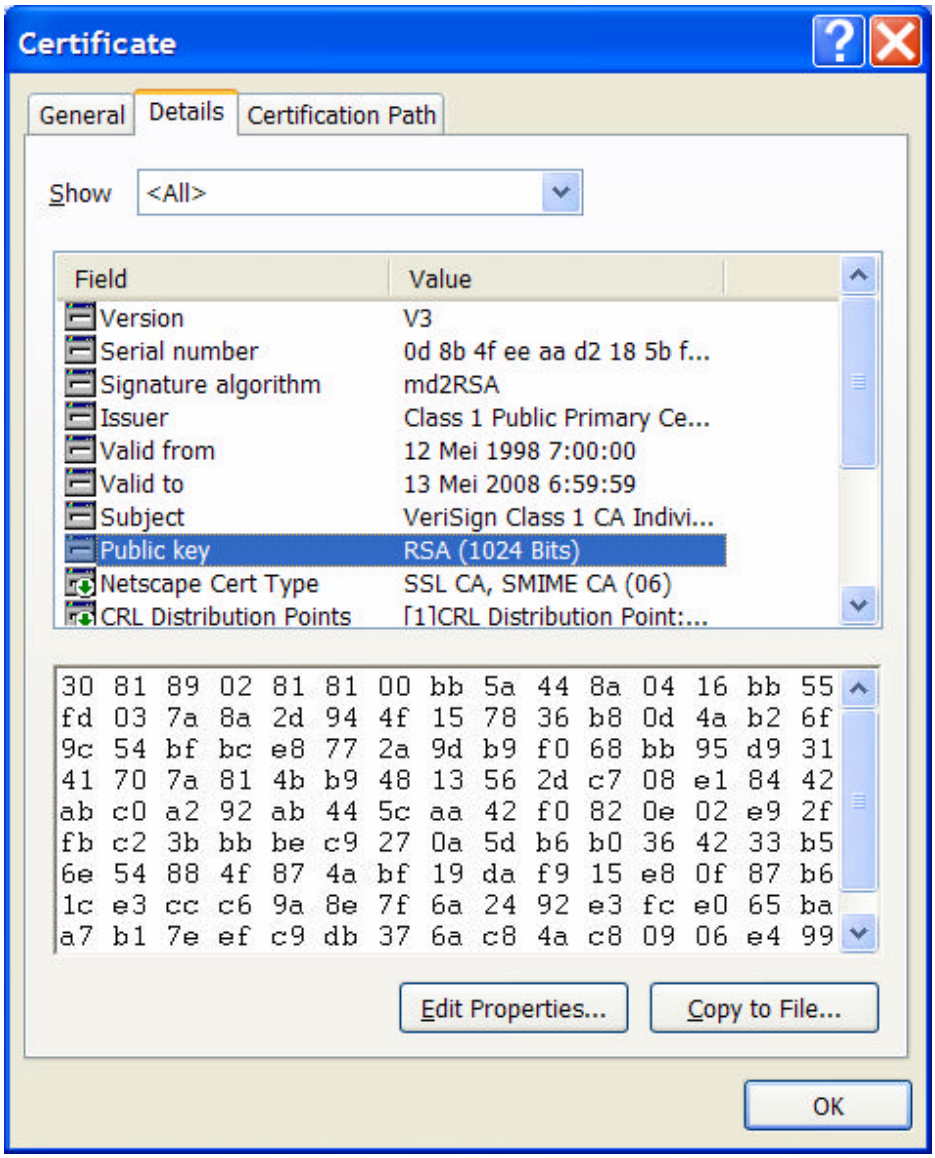
23.2 X.509

- Standard untuk sertifikat telah ditetapkan dan disetujui oleh ITU. Standard tersebut dinamakan X.509 dan digunakan secara luas di internet. Ada tiga versi standard X.509, yaitu V1, V2, dan V3.

- X.509 adalah cara mendeskripsikan sertifikat. *Field-field* utama di dalam sertifikat standard X.509 adalah sebagai berikut:

Field	Arti
Version	Versi X.509
Serial Number	Nomor ini plus nama *CA* secara unik digunakan untuk mengidentifikasi sertifikat
Signature Algorithm	Algoritma yang digunakan untuk menandatangani sertifikat.
Issuer	Nama pemberian X.509 untuk *CA*
Validity period	Waktu awal dan akhir periode valid
Subject name	Entitas (individu atau organisasi) yang disertifikasi
Public Key	Kunci publik subjek dan *ID* dari algoritma yang menggunakannya.
Issuer ID	*ID* opsional yang secara unik mengidentifikasi *certificate's issuer.*
Subject ID	*ID* opsional yang secara unik mengidentifikasi *certificate's subject*
Extensions *Signature*	Bayak ekstensi yang telah didefinisikan. Tanda-tangan sertifikat (ditandatangani dengan kunci privat CA).

Gambar 23.3 adalah contoh standard X.509 untuk sebuah sertifikat digital yang terdapat di dalam *Internet Explorer.*



Gambar 23.2 Contoh sertifikat digital yang mengikuti standard X.509

23.2 *Public Key Infrastructure* (*PKI*)

- Jika hanya ada satu *CA* untuk melayani sertifikat digital dari seluruh dunia, jelas *CA* tersebut akan kolaps karena *load* yang sangat besar.

- Solusi yang mungkin adalah mempunyai banyak *CA*, semua *CA* dijalankan oleh organisasi yang sama. Setiap CA bekerja dengan menggunakan kunci privat yang sama untuk menandatangani sertifikat.

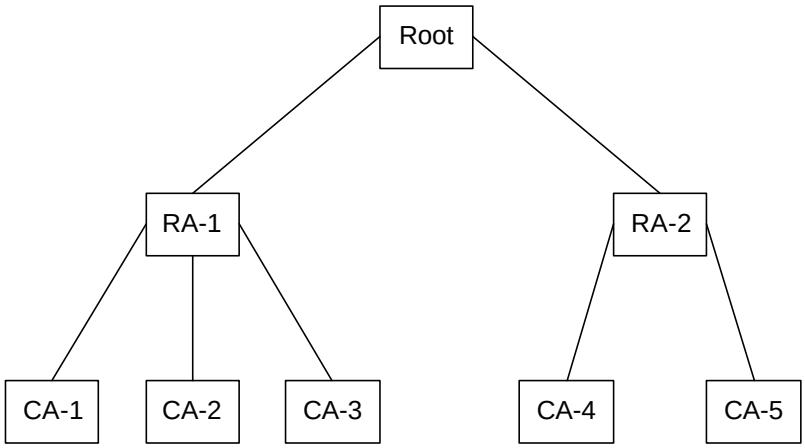
  Tapi hal ini menimbulkan masalah; jika kunci privat dicuri, maka ribuan sertifikat digital harus diganti. Lagipula, organisasi mana yang akan mengoperasikan *CA*? Sukar membayangkan suatu otoritas yang dapat diterima seluruh dunia.

- Untuk mengatasi masalah-masalah di atas, maka didefinisikan suatu cara yang berbeda untuk mensertifikasi kunci publik. Cara tersebut dinyatakan di dalam *PKI* (*Public Key Infrastructure*). *PKI* mengintegrasikan kriptografi kunci publik dengan sertifikat digital dan *CA* untuk mengotentikasi pihak-phak dalam suatu transaksi.

- *PKI* terdiri atas komponen-komponen:
  - pengguna (pemohon sertifikat dan pemakai sertifikat)
  - sertifikat digital
  - *CA*
  - Direktori (menyimpan sertifikat digital dan *CRL*)

- *PKI* menyediakan cara penstrukturan komponen-komponen di atas dan mendefinisikan standard bermacam-macam dokumen dan protokol.

- Bentuk *PKI* yang sederhana adalah hirarkhi *CA* dalam struktur pohon pada Gambar 23.4.



Gambar 23.4 Hirarkhi *CA* di dalam *PKI*

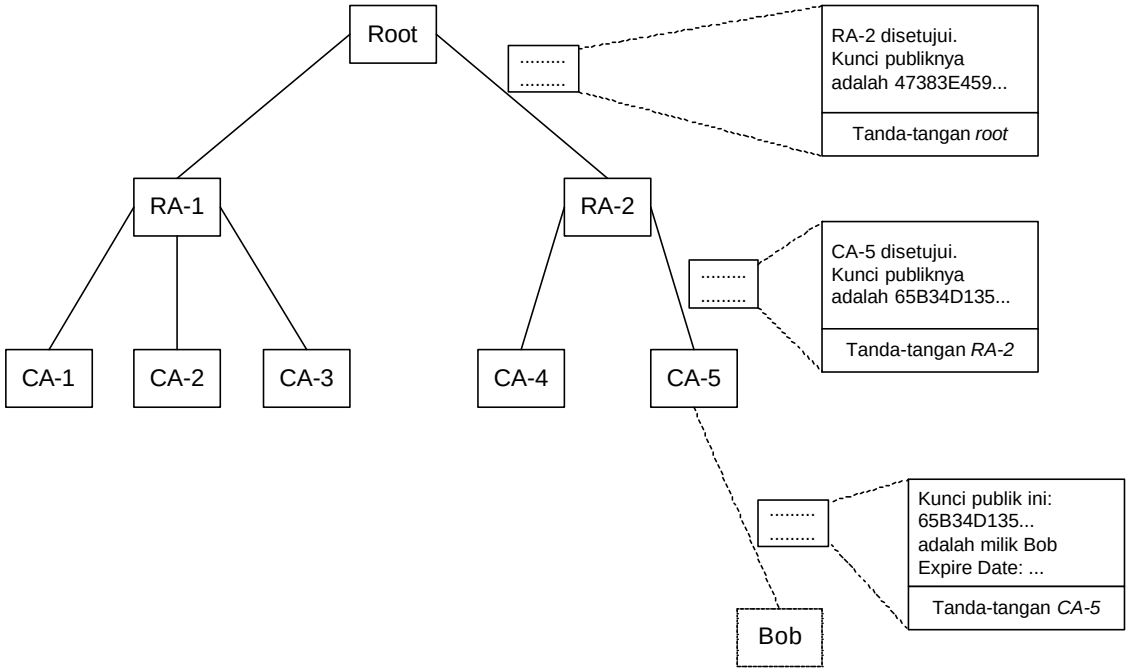
Aras ke-nol adalah *root*. *Root* merupakan *root certificate authority*, yang mana adalah *Internet Policy Registration Authority* (*IPRA*).

Root mensertifikasi *CA* aras satu dengan menggunakan privat *root* yang disebut *root key*. *CA* aras satu disebut *RA* (*Regional Authorities*), yang bertindak sebagai *policy creation authority*, yaitu oganisasi yang membuat kebijakan untuk memperoleh sertifikat digital. Sebuah *RA* mungkin mencakup beberapa area geografis, seperti negara bagian, negara, atau benua.

RA menandatangai sertifikat digital untuk *CA* di bawahnya dengan menggunakan kunci privat *RA*.

CA menandatangani sertifikat digital untuk individu atau organisasi dengan mengguankan kunci privat *CA*.

CA bertanggung jawab untuk otentikasi sertifikat digital, sehingga *CA* harus memeriksa informasi secara hati-hati sebelum mengeluarkan sertifikat digital. Gambar 23.5 memperlihatkan rantai sertifikat di dalam *PKI*.



Gambar 23.5 Contoh rantai sertifikat digital

- Misalkan Alice memerlukan kunci publik Bob untuk berkomunikasi, lalu dia mencari dan menemukan sertifikat Bob ditandatangani oleh *CA*-5.

  Alice kemudian mendatangani *CA*-5 dan meminta bukti legitimasi *CA*-5. *CA*-5 merespon dengan memperlihatkan sertifikat digital yang diperoleh dari *RA*-2, di dalamnya ada kunci publik *CA*-5 yang ditandangani oleh *RA*-2. Dengan menggunakan kunci publik *CA*-5, Alice dapat memverifikasi sertifikat digital Bob yang ditandatangani oleh *CA*-5 dan mendapatkan hasil bahwa sertifikat Bob sah.

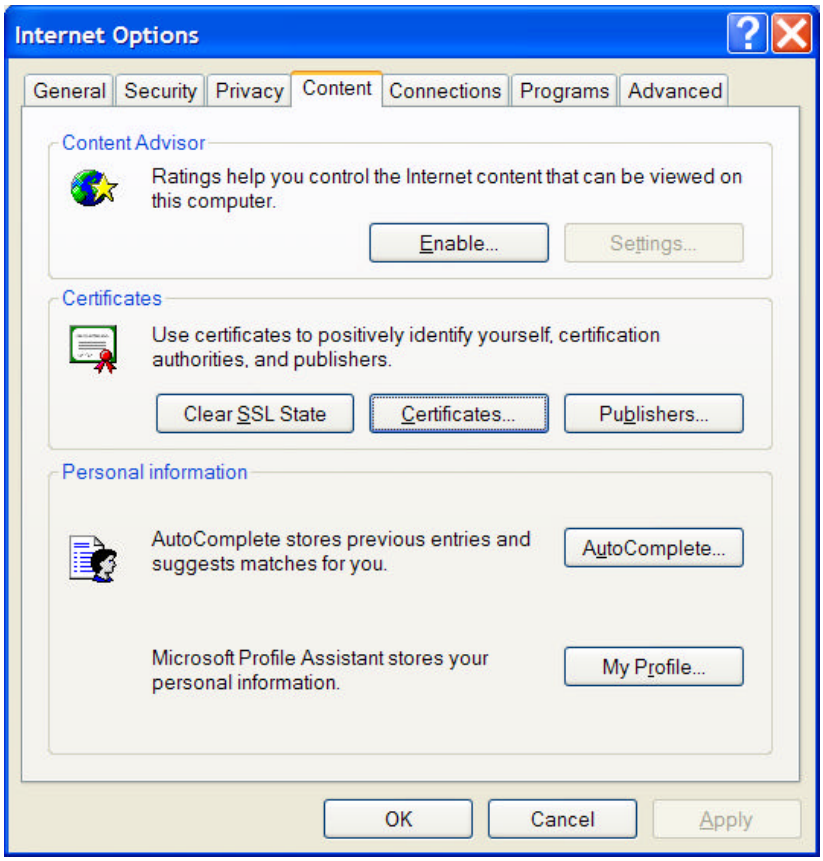
Langkah berikutnya, Alice mendatangi *RA*-2 dan meminta bukti legitimasi *RA*-5. *RA*-2 merespon dengan memperlihatkan sertifikat digital yang diperoleh dari *root*, di dalamnya ada kunci publik *RA*-2 yang ditandantangani oleh *root.* Alice memverifikasi sertifikat digital *RA*-2 dengan menggunakan kunci publik *root* dan mendapatkan hasil bahwa sertifikat tersebut sah. Sekarang Alice benar-benar yakin bahwa dia sudah memiliki kunci publik Bob.

Rantai sertifikat yang menuju ke *root* seperti ini disebut *chain of trust* atau *certification path*.

Tentu saja kita masih mempunyai masalah siapa yang bertindak sebagai *root*. Solusinya bukanlah dengan mempunyai sebuah *root* tunggal, tetapi mempunyai banyak *root,* masing-masing dengan sejumlah *RA* dan *CA*-nya sendiri.

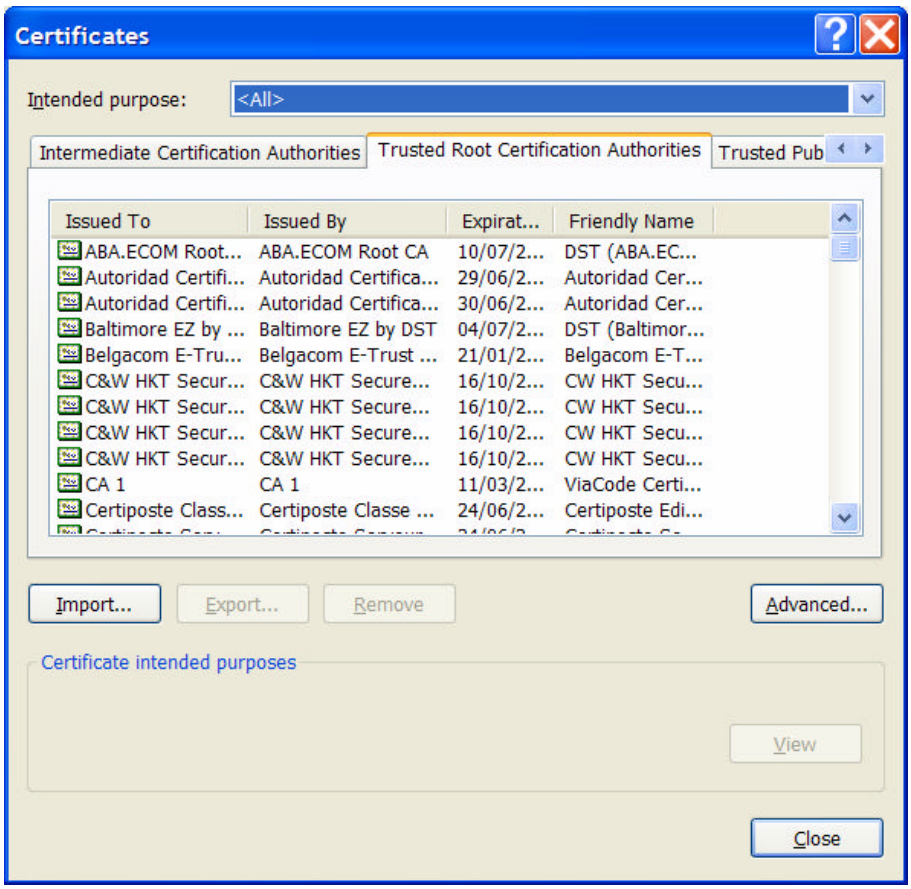
- Untuk melihat *CA* dan sertifikat digitalnya yang yang telah dipasang di dalam *Internet Explorer* (*IE*), lakukan hal sebagai berikut. Pilih:

Tools → *Internet Options* → *Contents*



Kemudian, klik tab:

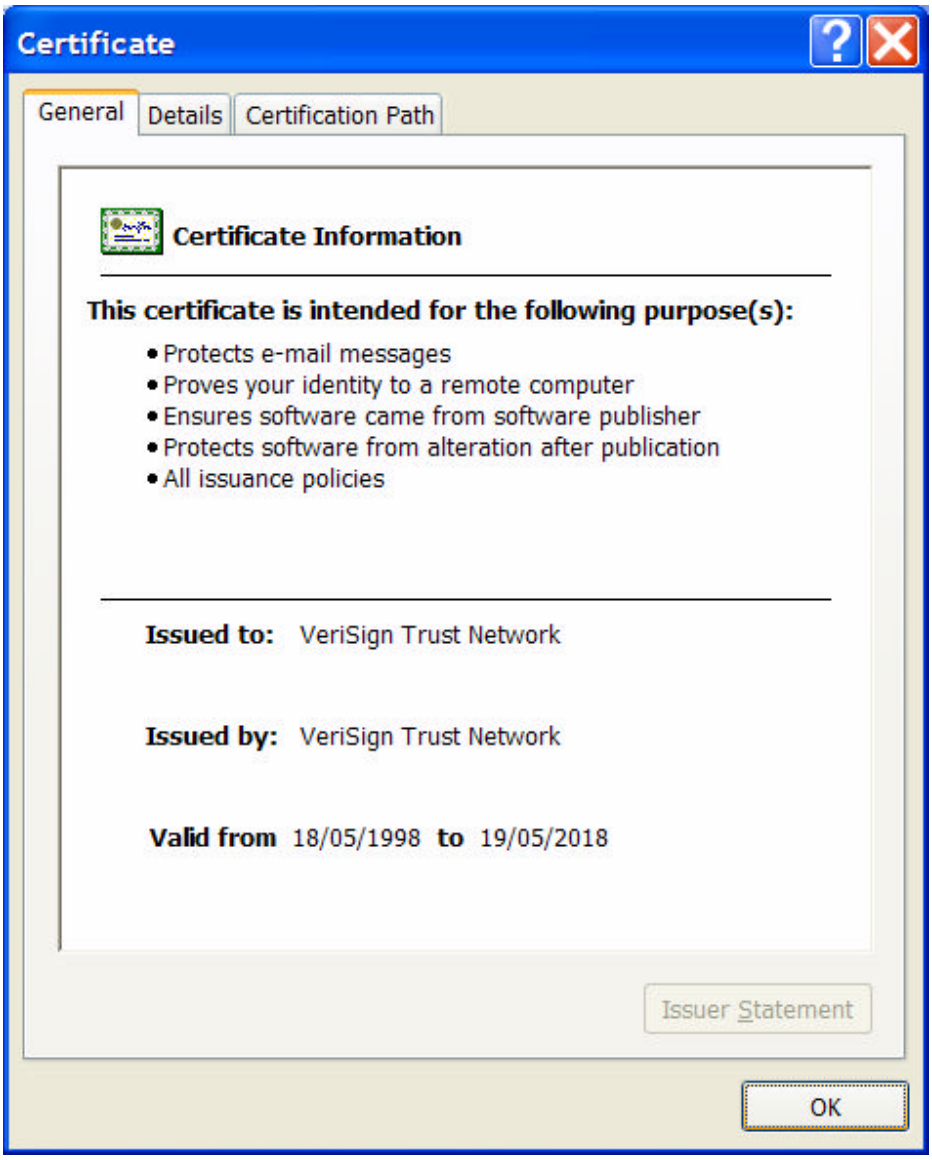
Certificates → *Trusted Root Certification Authorities*



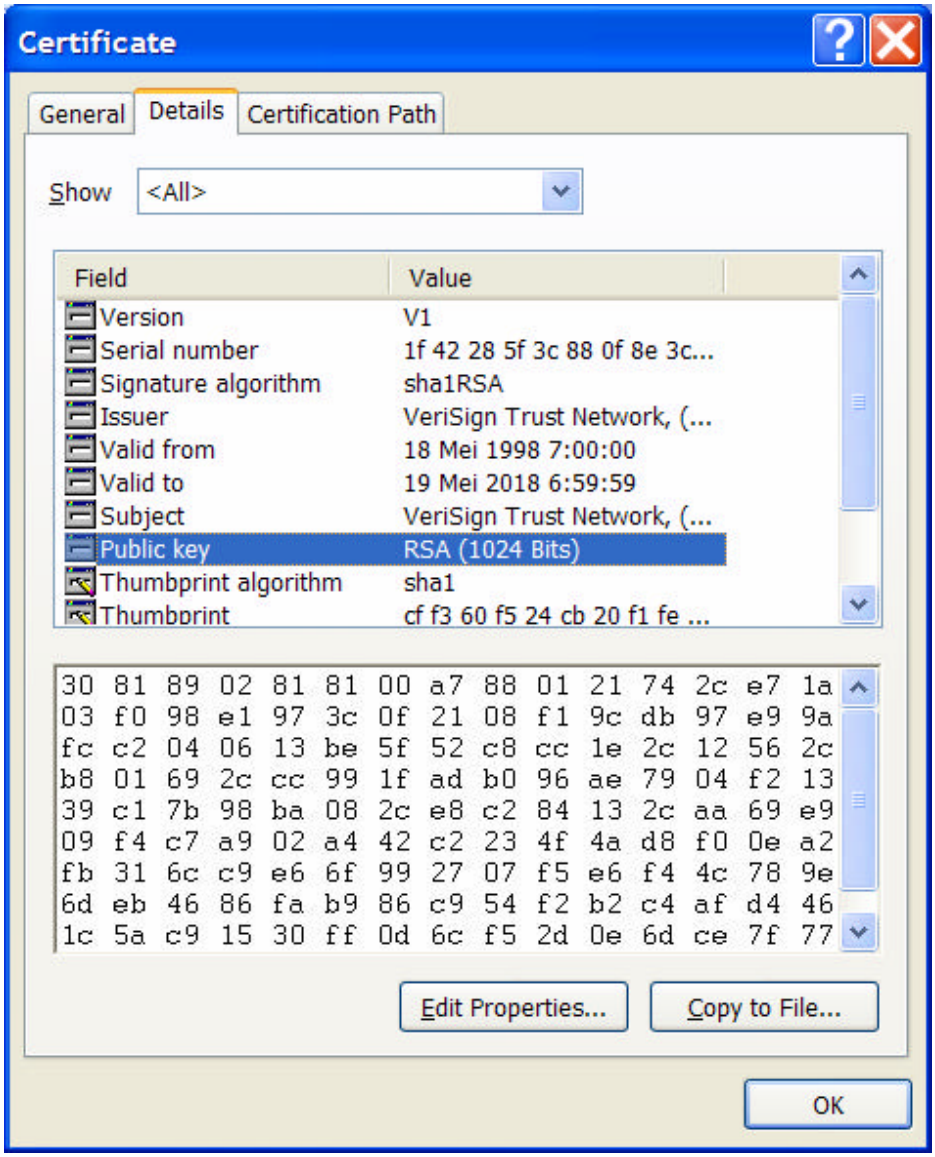
Trusted Root CA adalah *root* di dalam *PKI* dan memiliki cabang berupa *Intermediate CA*.

Bila terdapat *server* di internet yang diberi sertifikat oleh perusahaan yang tidak tercantum di dalam daftar *CA* di atas, maka *IE* akan memperingatkan bahwa *IE* tidak mengenal *CA* tersebut. Jika pengguna mempercayai *server* tersebut, maka *CA* tersebut akan ditambahkan ke dalam *IE*.

Untuk melihat isi sertifikat digital (misalnya sertifikat yang dikeluarkan oleh *VeriSign Trusted Network* dan diberikan untuk *VeriSign Trusted Network* sendiri), klik salah satu sertifikat, lalu tekan *View*:



Untuk melihat isi sertifikat lebih rinci, tekan *Details*. Anda dapat melihat kunci publik *VeriSign Trusted Network*:



Anda dapat meng-klik *Certification Path* untuk melihat *chain of trust*.

23.3 *Wireless PKI*

- *Wireless PKI* (*WPKI*) adalah protokol keamanan yang dispesifikasikan untuk transmisi nirkabel (*wireless*). Seperti *PKI*, *WPKI* mengotentikasi pengguna dengan sertifikat digital dan mengenkripsi pesan dengan kriptografi kunci-publik. CA WPKI melibatkan Certicom (www.certicom.com) dan RSA (www.rsasecurity.com).

23.4 *Microsof Authenticode*

- Banyak perusahaan piranti lunak (*software companies*) yang menawarkan produknya secara *on-line*, sedemikian sehingga pembeli dapat men-*download* piranti lunak langsung ke komputernya.

- Beberapa pertanyaan yang sering muncul jika kita men-*download* piranti lunak dari internet:
  - Bagaimana kita tahu bahwa program yang kita beli dari internet adalah aman dan tidak mengalamai perubahan (misalnya berubah karena gangguan virus)?
  - Bagaimana kita yakin bahwa kita tidak men-*download* virus komputer?
  - Bagaimana kita mempercayai situs *web* yang menjual program (software publisher) tersebut?

- Teknologi keamanan digunakan untuk menjamin bahwa piranti lunak yang di-*download* dapat dipercaya dan tidak mengalami perubahan. *Microsoft Auhenticode*, dikombinasikan dengan sertifikat digital *VeriSign* (atau *digital ID*), mengotentikasi penerbit piranti lunak (*software publisher*) dan mendeteksi apakah piranti lunak mengalami

perubahan. *Auhenticode* adalah fitur sekuriti yang dibangun di dalam *Internet Explorer.*

- Untuk menggunakan *Microsoft Auhenticode*, setiap penerbit harus mempunyai sertifikat digital yang dirancang untuk tujuan penerbitan piranti lunak. Sertifikat semacam itu dapat diperoleh dari *CA* seperti *VeriSign.* Untuk memperoleh sertifikat, penerbit piranti lunak harus menyediakan kunci publik dan informasi identifikasi lainnya dan menandatangai perjanjian bahwa ia tidak mendistribusikan virus.

- *Microsoft Auhenticode* menggunakan teknologi tanda-tangan digital untuk menandatangani piranti lunak. Piranti lunak yang ditandatangani dan sertifikat digital penerbit memberikan bukti bahwa piranti lunak tersebut aman dan tidak mengalami perubahan.

- Bila pembeli mencoba men-*downoad file*, kotak dialog yang muncul di layar menampilkan sertifikat digital dan nama *CA*-nya. *Link* ke penerbit piranti lunak dan *link* ke *CA* juga disediakan sehingga pembeli dapat mempelajari lebih jauh mengenai penerbit dan *CA* sebelum ia menyetujui untuk men-*download* piranti lunak. Jika *Microsoft Auhenticode* menyatakan bahwa piranti lunak tersebut membahayakan, maka transaksi dihentikan.