

**II4031 Kriptografi dan Koding**

# **Tanda-tangan Digital**



**Oleh: Rinaldi Munir**

**Program Studi Sistem dan Teknologi Informasi  
Sekolah Teknik Elektro dan Informatika  
ITB - 2023**



# Review materi awal

- Ingat kembali empat layanan keamanan yang disediakan oleh kriptografi:
  1. Kerahasiaan pesan (*confidentiality/secretcy*)
  2. Keaslian pesan (*data integrity*).
  3. Otentikasi (*authentication*)
  4. Anti-penyangkalan (*nonrepudiation*).
- Layanan 1 dilakukan dengan mengenkripsi/dekripsi pesan
- Layanan 2 dilakukan dengan fungsi hash
- Layanan 3 dan 4 dilakukan dengan menggunakan tanda-tangan digital (*digital signature*).

# Tanda-tangan

- Sejak zaman dahulu, tanda-tangan sudah digunakan untuk otentikasi dokumen cetak.
- Tanda-tangan mempunyai karakteristik sebagai berikut:
  1. Tanda-tangan adalah bukti yang otentik.
  2. Tanda tangan tidak dapat dilupakan.
  3. Tanda-tangan tidak dapat dipindah untuk digunakan ulang.
  4. Dokumen yang telah ditandatangani tidak dapat diubah.
  5. Tanda-tangan tidak dapat disangkal.

Hari: Senin, Tanggal 12 September 2016

Waktu: 14.00 Wib

Tempat: Aula

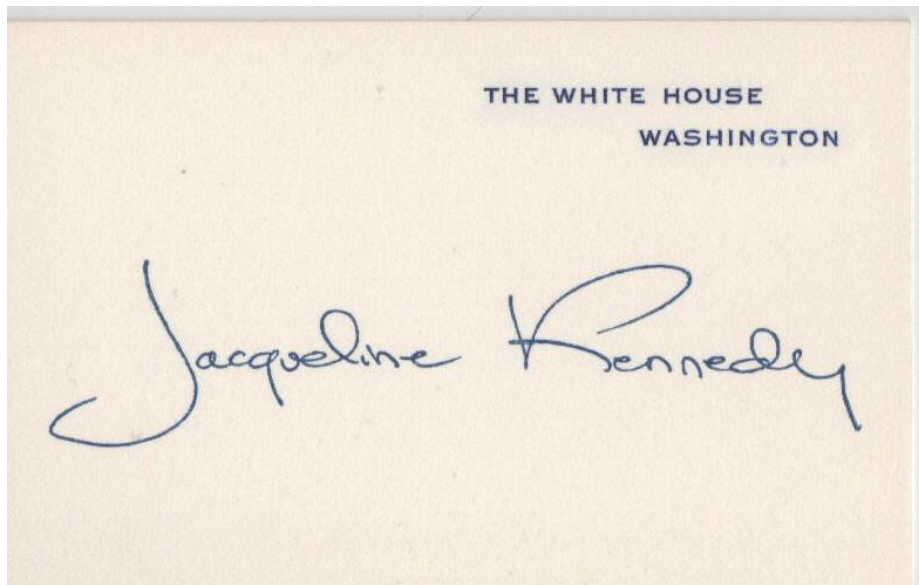
Demikian kami buat surat ini dengan sebenarnya. Harap semua peserta untuk hadir sesuai jadwal yang telah disebutkan di atas. Terima kasih.

Tertanda tangan

Mr Mukidi

Master of Ceremony  
Doktor Mukidi, Mpd

- Fungsi tanda tangan pada dokumen kertas juga diterapkan untuk otentikasi pada data digital (pesan, dokumen elektronik).
- Tanda-tangan untuk data digital dinamakan **tanda-tangan digital** (*digital signature*).
- Tanda-tangan digital di dalam konteks kriptografi tidak sama dengan tanda-tangan yang di-digitisasi (*digitized signature*) dengan cara dipindai atau difoto.



*digitized signature*



*digitized signature*

- Tanda-tangan digital adalah *nilai kriptografis* yang bergantung pada isi pesan dan kunci.
- Tanda-tangan seseorang pada dokumen cetak selalu sama, apa pun isi dokumennya.
- Sedangkan tanda-tangan digital selalu berbeda-beda antara satu pesan dengan pesan lain, dan/atau antara satu kunci dengan kunci yang lain.

Paris, 31 Desember 2018

Halo Alice

Sudah lama kita tidak berjumpa sejak lulus SMA. Saya sekarang tinggal di Paris sejak tahun 2016. Saya bekerja di sebuah perusahaan IT yang bernama Solution Express. Perusahaan ini memberikan layanan keamanan informasi berbasis cloud computing. Saya menjadi menejer Quality Control. Klien kami umumnya adalah bank-bank yang membutuhkan keamanan data nasabah.

Oh ya, saya belum menanyakan bagaimana keadaanmu sekarang. Di mana kamu bekerja atau malah melanjutkan studi S2 di mana? Saya ingat kamu dulu jago sekali pelajaran kimia. Apakah kamu masih menekuni bidang kimia saat ini?

Oke deh, jika kamu jalan-jalan ke Eropa jangan lupa mampir ke kota Paris. Nanti saya akan ajak kamu mengunjungi Menara Eiffel. Bisa naik sampai ke atas lho.

Salam dari temanmu di Paris

Bob

-- BEGIN SIGNATURE—

**13706B6D42442620B2FD1098BD4D54ADFA9F7DC27576954ADCE5E5FC901**

-- END SIGNATURE--

# Dua proses dalam tanda-tangan digital

1. Menandatangani pesan (*signing*)  
Memberi tanda-tangan pada pesan
2. Memverifikasi tanda-tangan (*verification*)  
Memeriksa keabsahan tanda-tangan

# Bagaimana cara menandatangani pesan?

Ada dua cara yang dilakukan untuk menandatangani pesan:

1. Mengenkripsi pesan
  - Khusus untuk pesan rahasia
2. Menggunakan kombinasi fungsi *hash* (*hash function*) dan kriptografi kunci-publik
  - Untuk pesan yang **tidak perlu** rahasia



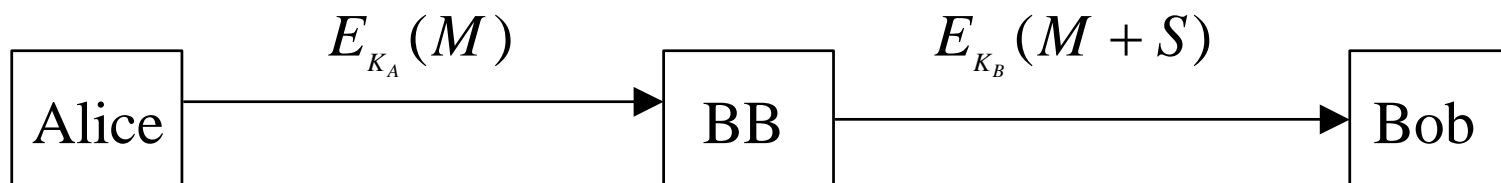
# Penandatanganan dengan Cara Mengenkripsi Pesan

## a. Enkripsi menggunakan algoritma kriptografi kunci-simetri

- Pesan yang dienkripsi dengan algoritma simetri sudah memberikan solusi untuk otentikasi pengirim karena kunci simetri hanya diketahui oleh pengirim dan penerima.
- Namun cara ini tidak menyediakan cara untuk melakukan anti-penyangkalan (*non-repudiation*).
- Jika Alice menyangkal telah mengirim pesan kepada Bob, maka Bob tidak punya cara untuk membantah sangkalan Alice. Alice dapat saja menuduh bahwa pesan tersebut dibuat oleh Bob sendiri karena hanya Alice dan Bob yang mengetahui kunci untuk enkripsi dan dekripsi.
- **Kesimpulan:** menandatangani pesan dengan kriptografi kunci simetri tidak dapat dilakukan.

- Agar kriptografi kunci-simetri dapat mengatasi masalah penyangkalan, maka diperlukan pihak ketiga yang dipercaya oleh pengirim/penerima.
- Pihak ketiga ini disebut penengah (*arbitrase*).
- Misalkan BB (*Big Brothers*) adalah otoritas arbitrase yang dipercaya oleh Alice dan Bob.
- BB memberikan kunci rahasia  $K_A$  kepada Alice dan kunci rahasia  $K_B$  kepada Bob.
- Hanya Alice dan BB yang mengetahui  $K_A$ , begitu juga hanya Bob dan BB yang mengetahui  $K_B$ .

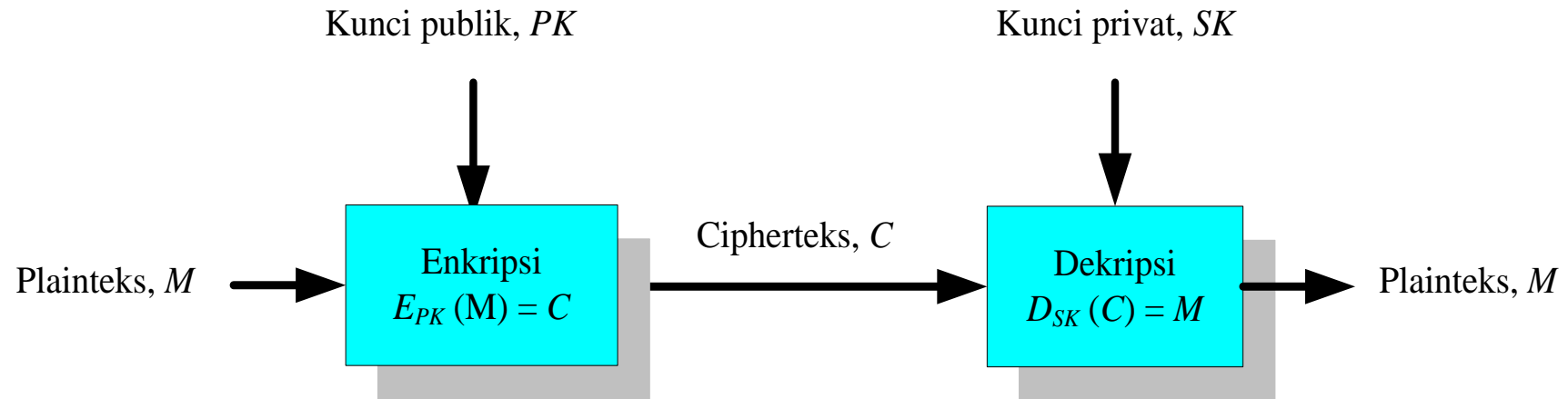
- Jika Alice bekirim pesan  $M$  kepada Bob, maka langkah-langkahnya adalah sebagai berikut:
  1. Alice mengenkripsi pesan  $M$  untuk Bob dengan  $K_A$ , lalu mengirim cipherteksnya ke BB.
  2. BB melihat bahwa pesan dari Alice, lalu mendekripsi pesan dari Alice dengan  $K_A$ .
  3. BB membuat pernyataan  $S$  bahwa ia menerima pesan dari Alice, lalu menambahkan pernyataan tersebut pada plainteks dari Alice.
  4. BB mengenkripsi bundel pesan  $(M + S)$  dengan  $K_B$ , lalu mengirimkannya kepada Bob.
  5. Bob mendekripsi bundel pesan dengan  $K_B$ . Ia dapat membaca pesan dari Alice ( $M$ ) dan pernyataan ( $S$ ) dari BB bahwa Alice yang mengirim pesan tersebut.



- Jika Alice menyangkal telah mengirim pesan tersebut, maka pernyataan dari BB pada pesan yang diterima oleh Bob digunakan untuk menolak penyangkalan Alice.
- Bagaimana BB tahu bahwa pesan tersebut dari Alice dan bukan dari Charlie?
- Karena hanya BB dan Alice yang mengetahui kunci rahasia, maka hanya Alice yang dapat mengenkripsi pesan dengan kunci tersebut.
- **Kelemahan:** pelibatan pihak ketiga dalam penandatanganan pesan membuatnya menjadi lebih rumit, tidak praktis, dan tidak sangkil (*efficient*) sehingga tidak digunakan di dalam praktek dunia nyata.
- Solusinya adalah menandatangani pesan dengan menggunakan kriptografi kunci-publik seperti penjelasan berikut ini.

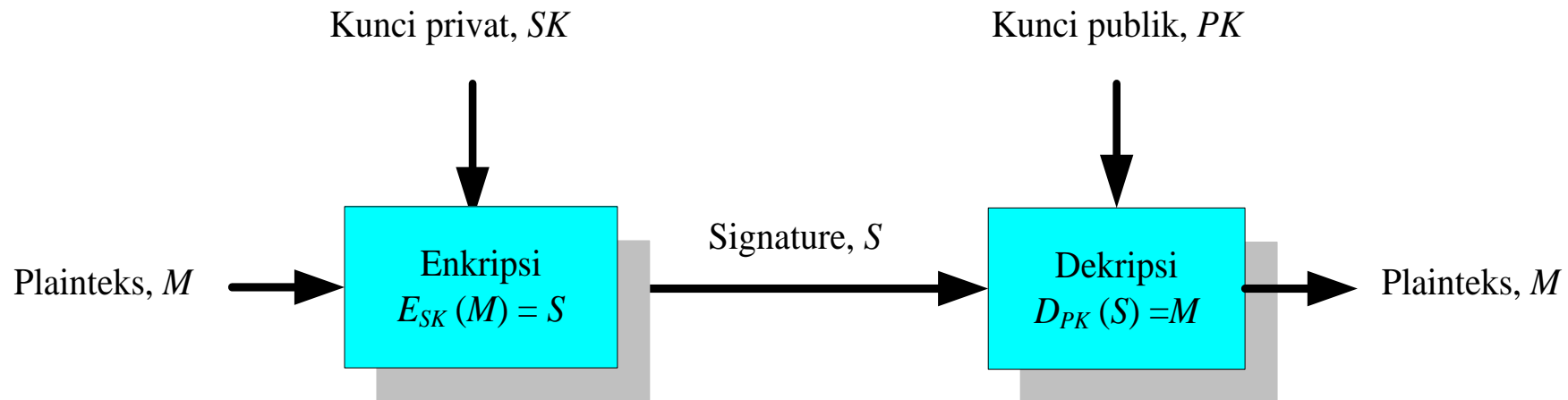
## b. Enkripsi menggunakan algoritma kriptografi kunci-publik

- Mengenkripsi pesan dengan menggunakan kunci publik penerima pesan dan mendekripsinya dengan kunci privat penerima pesan adalah proses yang normal di dalam kriptografi kunci-publik.



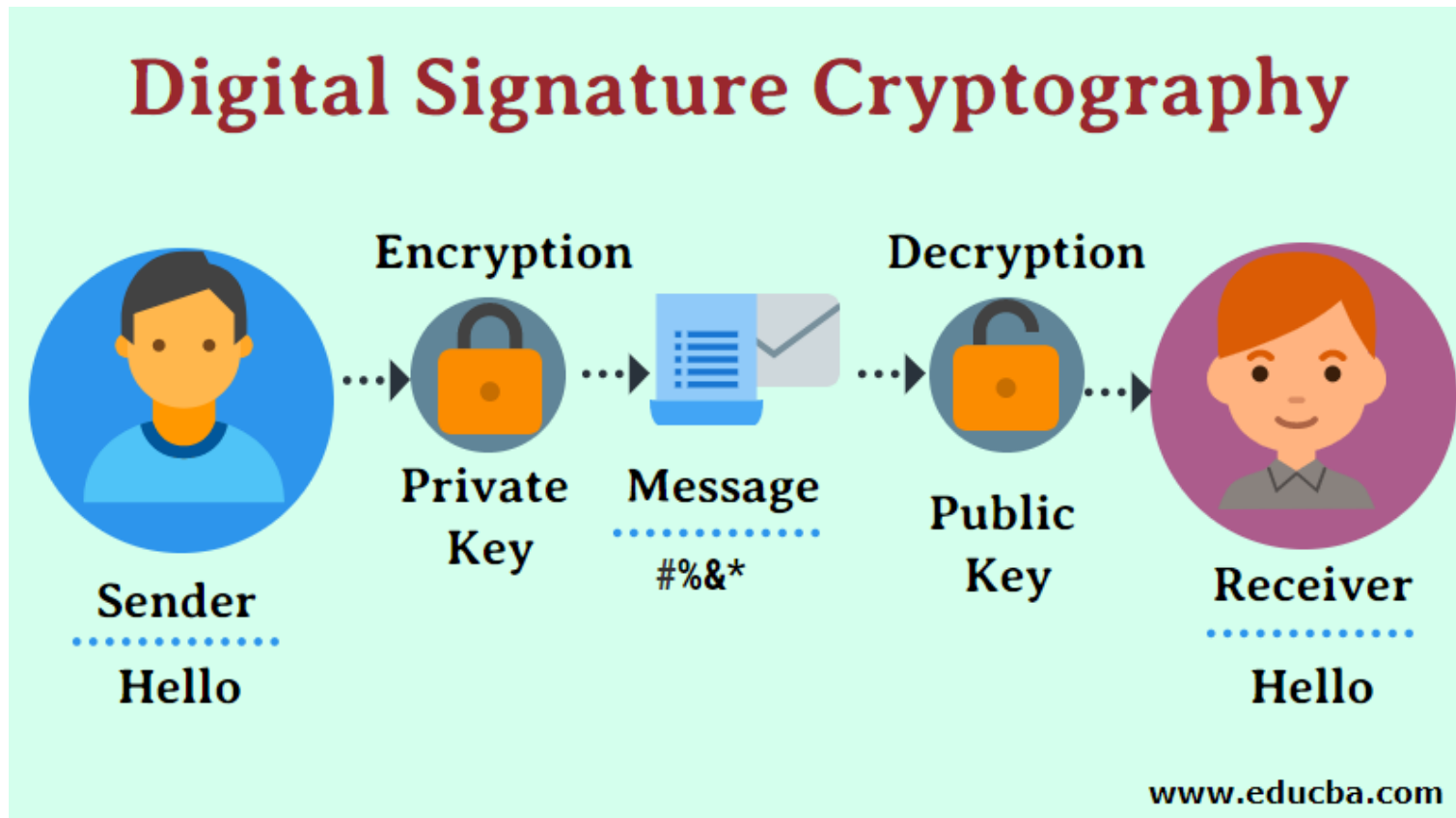
- Namun cara ini tidak dapat mengotentikasi si pengirim pesan karena kunci publik diketahui oleh siapapun. Karena itu cara ini tidak dapat digunakan untuk menandatangani pesan.

- Oleh karena itu, agar dapat berfungsi sebagai tanda-tangan digital, maka prosesnya dibalik:
  - pesan dienkripsi dengan kunci privat si pengirim.
  - pesan didekripsi dengan kunci publik si pengirim.



- Dengan cara ini, maka kerahasiaan pesan dan otentikasi si pengirim pesan keduanya dicapai sekaligus. Penerima pesan dapat mengotentikasi pengirim pesan karena kunci publik dan kunci privat adalah berpasangan.
- Ide ini ditemukan oleh Diffie dan Hellman.

**Kesimpulan:** Jadi untuk menandatangani pesan, maka pesan dienkripsi dengan kunci privat si pengirim, penerima pesan mendekripsinya dengan kunci publik si pengirim pesan.



Sumber: <https://www.educba.com/digital-signature-cryptography/>

- Proses menandatangani pesan (*signing* oleh pengirim):

$$S = E_{SK}(M)$$

- Proses memverifikasi tanda-tangan (*verification* oleh penerima):

$$M = D_{PK}(S)$$

Keterangan:

$SK$  = *secret key* = kunci privat pengirim

$PK$  = *public key* = kunci publik pengirim

$E$  = fungsi enkripsi;       $D$  = fungsi dekripsi

$M$  = pesan;       $S$  = *signature* = cipherteks (hasil enkripsi pesan)

- Jadi, mengenkripsi pesan dengan menggunakan kunci privat sama artinya dengan menandatangani pesan. Mendekripsi pesan dengan kunci publik sama artinya dengan memverifikasi tanda-tangan .
- Dengan cara seperti ini, tidak lagi dibutuhkan pihak penengah (arbitrase).



- Beberapa algoritma kunci-publik dapat digunakan untuk menandatangani pesan dengan cara mengenkripsinya, asalkan algoritma tersebut memenuhi sifat:

$$D_{SK}(E_{PK}(M)) = M \text{ dan } D_{PK}(E_{SK}(M)) = M ,$$

Keterangan:

$PK$  = kunci publik ;

$SK$  = kunci privat (*secret key*).

$E$  = fungsi enkripsi;

$D$  = fungsi dekripsi;

$M$  = pesan

- Contoh algoritma yang memenuhi sifat ini adalah RSA, karena persamaan enkripsi dan dekripsi identik (dapat dipertukarkan)

Enkripsi/dekripsi biasa dengan RSA:

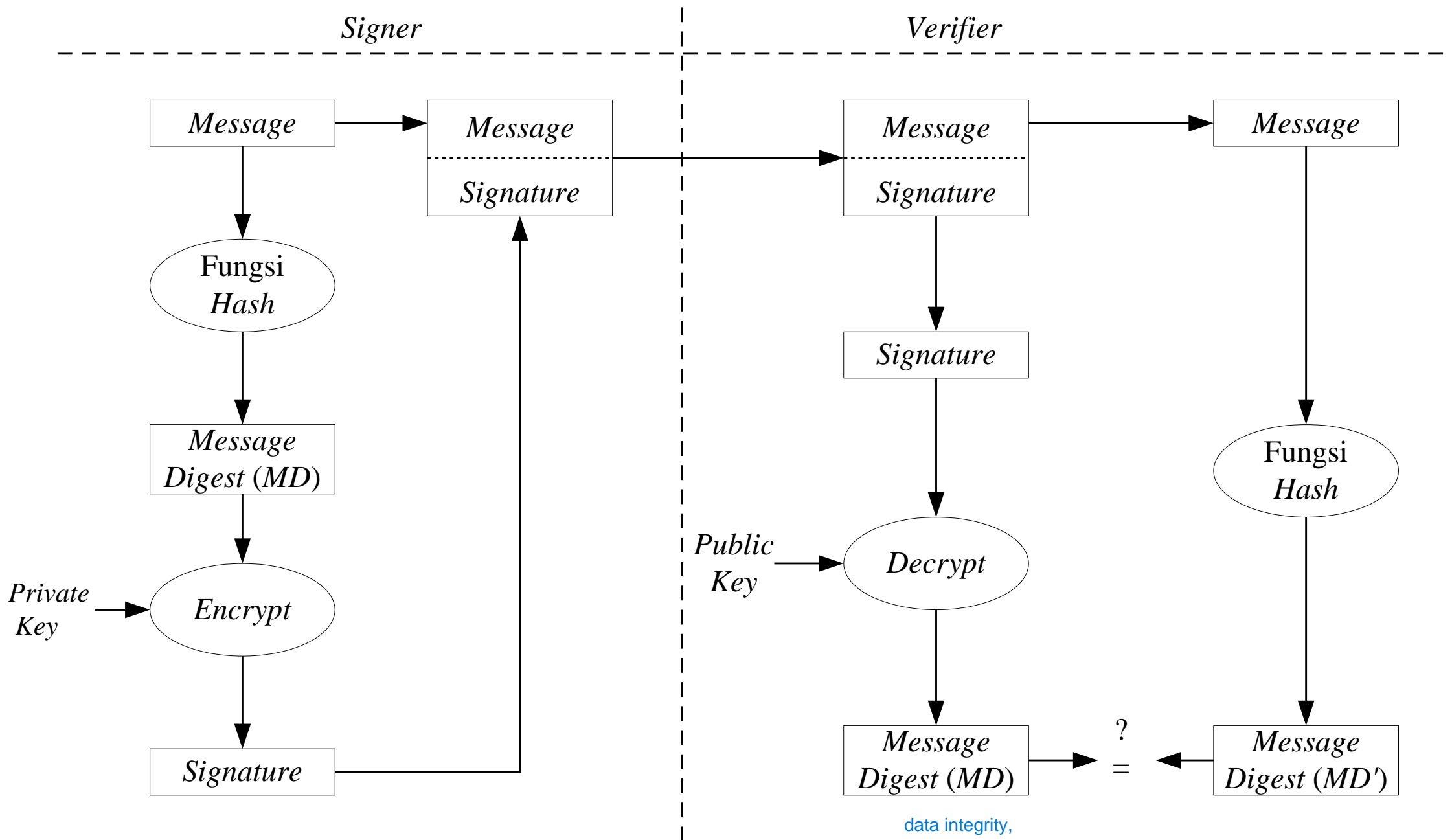
- Enkripsi:  $c = m^e \bmod n$
- Dekripsi:  $m = c^d \bmod n$

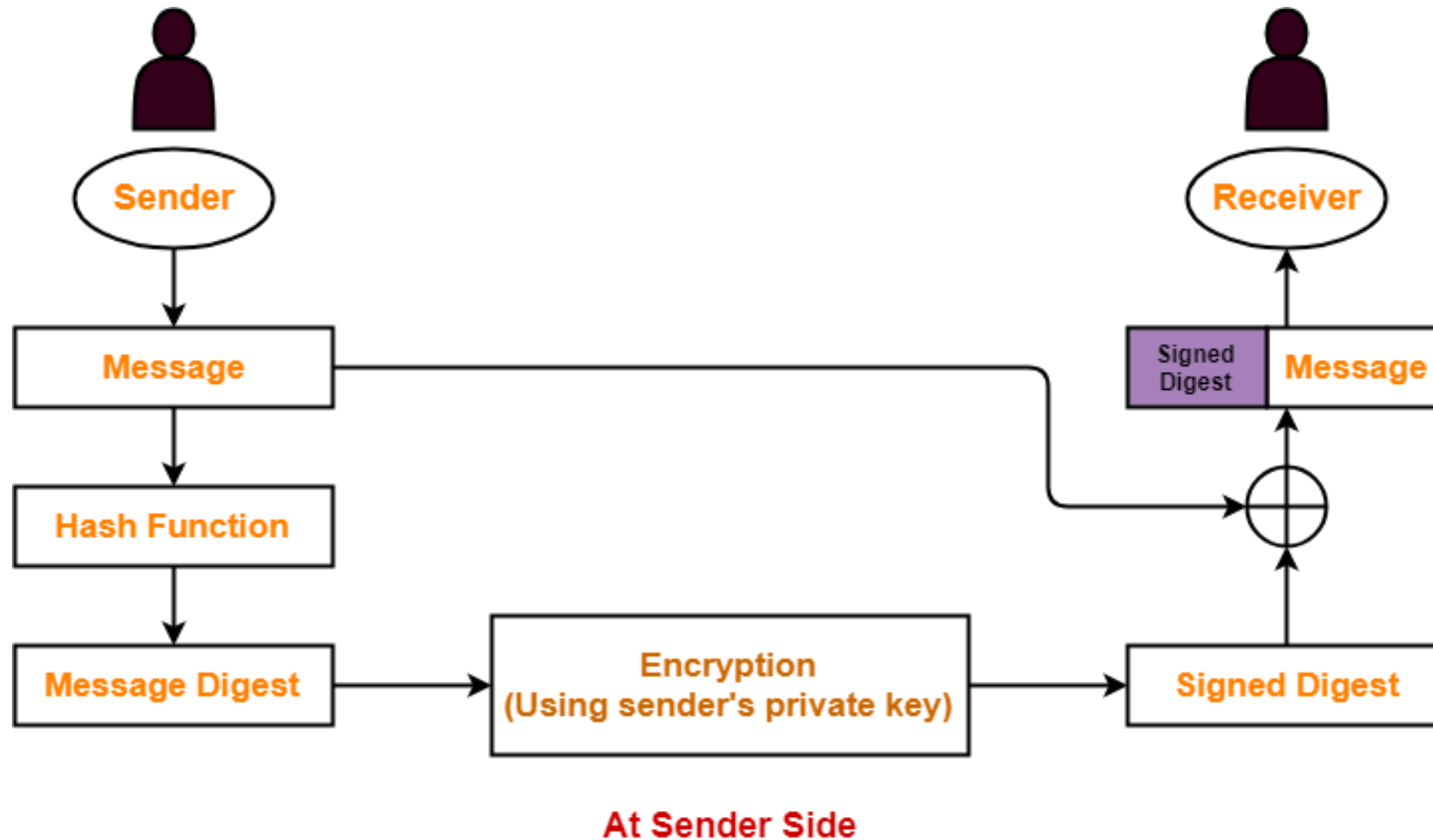
Tanda-tangan digital dengan RSA:

- Signing:  $c = m^d \bmod n$
- Verification:  $m = c^e \bmod n$

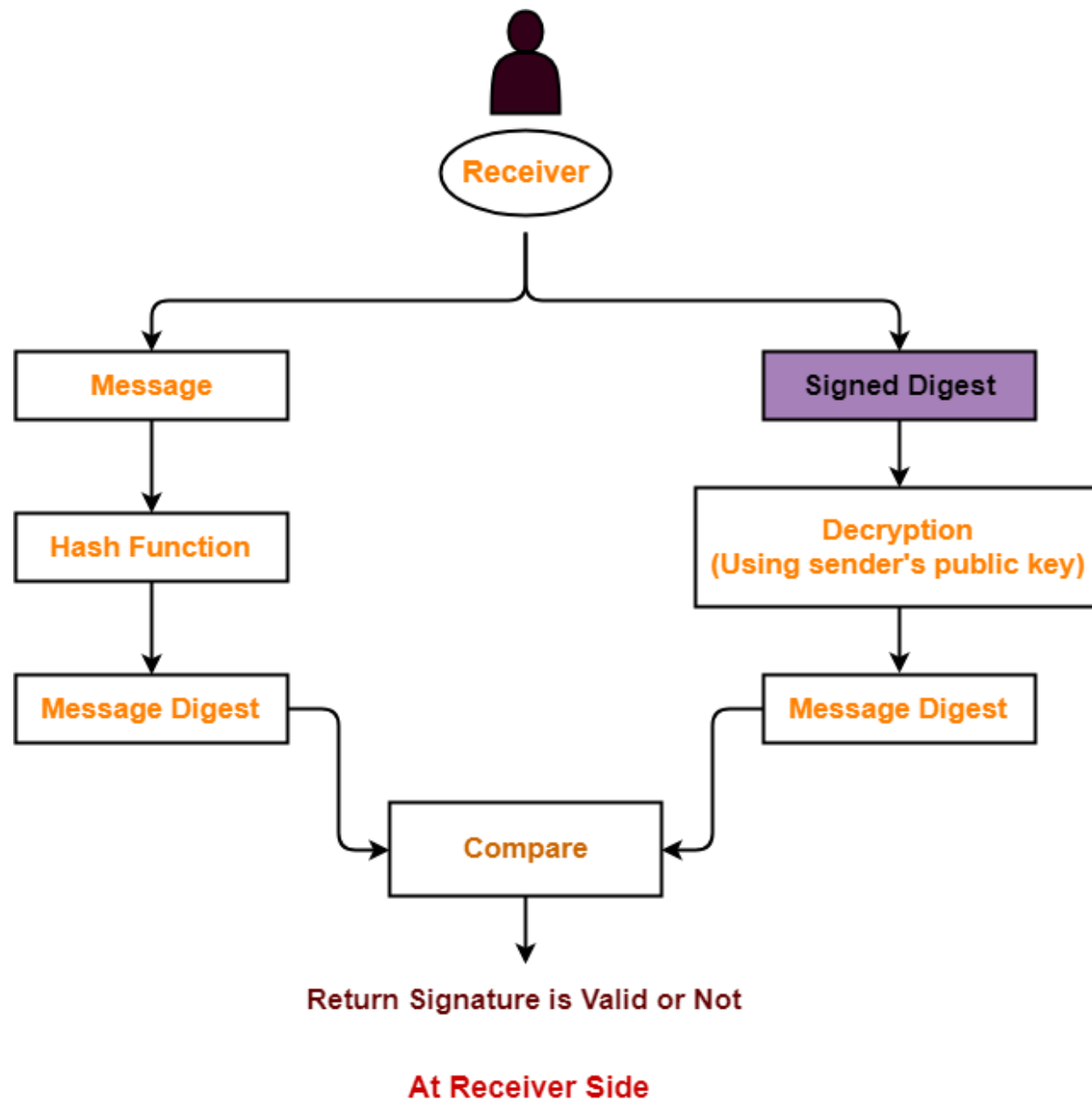
# Penandatanganan dengan Menggunakan Kombinasi Kriptografi kunci-publik dan Fungsi *Hash*

- Penandatanganan pesan dengan cara mengenkripsinya selalu memberikan dua fungsi berbeda: kerahasiaan pesan dan otentikasi pesan.
- Pada beberapa kasus, seringkali otentikasi yang diperlukan, tetapi kerahasiaan pesan tidak perlu. Maksudnya, pesan tidak perlu dienkripsi karena tidak rahasia, sebab yang dibutuhkan hanya keotentikan pesan saja.
- Kombinasi algoritma kunci-publik dan fungsi *hash* dapat digunakan untuk kasus seperti ini.





<https://www.gatevidyalay.com/how-digital-signature-works-algorithm/>



**At Receiver Side**

- Dua algoritma *signature* yang digunakan secara luas adalah *RSA* dan *ElGamal Signature*.
- Pada *RSA*, algoritma enkripsi dan dekripsi identik, sehingga proses *signature* dan verifikasi juga identik.
- Selain *RSA*, terdapat algoritma yang dikhususkan untuk tanda-tangan digital, yaitu *Digital Signature Algorithm* (DSA), yang merupakan bakuan (*standard*) untuk *Digital Signature Standard* (DSS).
- Pada *DSA*, algoritma *signature* dan verifikasi berbeda

# Tanda-tangan digital dengan algoritma RSA

## Langkah-langkah pemberian tanda-tangan (*signing*)

1. Pengirim menghitung nilai *hash* dari pesan  $M$ :

$$h = H(M)$$

2. Pengirim mengenkripsi  $h$  dengan kunci privatnya (SK) menggunakan persamaan enkripsi *RSA*, hasilnya adalah signature  $S$ :

$$S = h^{SK} \bmod n \quad (n \text{ adalah modulus, } n = pq).$$

3. Pengirim mentransmisikan  $M + S$  ke penerima

## Langkah-langkah verifikasi tanda-tangan (*verifying*)

1. Penerima menghitung nilai *hash* dari pesan  $M$  yang diterima:

$$h = H(M)$$

2. Penerima melakukan dekripsi terhadap tanda-tangan  $S$  dengan kunci publik si pengirim (PK) menggunakan persamaan dekripsi *RSA*:

$$h' = S^{PK} \bmod n$$

3. Penerima membandingkan  $h$  dengan  $h'$ . Jika  $h = h'$  maka tanda-tangan digital adalah otentik. Jika tidak sama, maka tanda-tangan tidak otentik sehingga pesan dianggap tidak asli lagi atau pengirimnya





Alice

$M =$

Pada wisuda sarjana baru ITB, ternyata ada seorang wisudawan yang paling muda. Umurnya baru 19 tahun. Ini berarti dia masuk ITB pada umur 15 tahun. Zaman sekarang banyak sarjana masih berusia muda belia. Mungkin masuk sekolah pada usia dini dan mengikuti kelas akselerasi pada tingkatan SD, SMP, dan SMA. Masuk SD umur 6 tahun dan ikut kelas aksel sehingga selesai dalam waktu lima tahun pada umur 11. SMP diselesaikan dalam waktu dua tahun dan SMA dalam waktu dua tahun, sehingga lulus SMA pada umur 15 tahun. Kuliah di ITB selama empat tahun sehingga wajar saja menjadi sarjana pada umur 19 tahun.

$$h = H(M) = \text{A4C05176E1440FC879C06C72FA603A24} \quad (\text{heksadesimal})$$

$$= 218991964599382371228554013295471770148 \quad (\text{desimal})$$

$$S = h^{\text{PrivK}} \bmod n \quad (n = 223427, \text{PrivK} = 171635)$$

$$= (218991964599382371228554013295471770148)^{171635} \bmod (223427) = 46489$$

$M + S =$

Pada wisuda sarjana baru ITB, ternyata ada seorang wisudawan yang paling muda. Umurnya baru 19 tahun. Ini berarti dia masuk ITB pada umur 15 tahun. Zaman sekarang banyak sarjana masih berusia muda belia. Mungkin masuk sekolah pada usia dini dan mengikuti kelas akselerasi pada tingkatan SD, SMP, dan SMA. Masuk SD umur 6 tahun dan ikut kelas aksel sehingga selesai dalam waktu lima tahun pada umur 11. SMP diselesaikan dalam waktu dua tahun dan SMA dalam waktu dua tahun, sehingga lulus SMA pada umur 15 tahun. Kuliah di ITB selama empat tahun sehingga wajar saja menjadi sarjana pada umur 19 tahun.

46489



Bob

Pada wisuda sarjana baru ITB, ternyata ada seorang wisudawan yang paling muda. Umurnya baru 19 tahun. Ini berarti dia masuk ITB pada umur 15 tahun. Zaman sekarang banyak sarjana masih berusia muda belia. Mungkin masuk sekolah pada usia dini dan mengikuti kelas akselerasi pada tingkatan SD, SMP, dan SMA. Masuk SD umur 6 tahun dan ikut kelas aksel sehingga selesai dalam waktu lima tahun pada umur 11. SMP diselesaikan dalam waktu dua tahun dan SMA dalam waktu dua tahun, sehingga lulus SMA pada umur 15 tahun. Kuliah di ITB selama empat tahun sehingga wajar saja menjadi sarjana pada umur 19 tahun.

46489

$$\begin{aligned}h &= H(M) = A4C05176E1440FC879C06C72FA603A24 && \text{(heksadesimal)} \\&= 218991964599382371228554013295471770148 && \text{(decimal)} \\&\equiv 125468 \pmod{223427}\end{aligned}$$

$$\begin{aligned}h' &= S^{PubK} \pmod{n} \quad (PubK = 731) \\&= (46489)^{731} \pmod{(223427)} \\&= 125468\end{aligned}$$

sama

Tandda tangan valid!

# SELAMAT BELAJAR