

Bahan kuliah IF4020 Kriptografi

03 – Kriptografi Klasik

(Bagian 1)

Oleh: Rinaldi M

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2026**



Pendahuluan

- Kriptografi klasik (*classical cryptography*) merupakan kriptografi yang sudah tua, sudah ada sejak ribuan tahun yang lalu hingga ditemukan komputer digital
- *Cipher* pada kriptografi klasik, dinamakan *cipher* klasik, (*classical cipher*) hanya memproses pesan dari huruf-huruf alfabet saja
- Menggunakan alat tulis pena dan kertas saja, belum ada komputer
- Termasuk ke dalam jenis kriptografi kunci-simetri
- Tiga alasan mempelajari kriptografi klasik:
 1. Memahami konsep dasar kriptografi.
 2. Sebagai dasar algoritma kriptografi modern.
 3. Untuk memahami kelemahan sistem *cipher*.

- *Cipher* di dalam kriptografi klasik disusun oleh dua teknik dasar:

1. Teknik substitusi: mengganti huruf plainteks dengan huruf cipherteks.

Plainteks:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipherteks:	I	J	K	L	Q	R	S	T	U	V	W	D	C	B	A	Z	Y	X	P	O	N	M	H	G	F	E

Contoh: Plainteks: MENGANTUK

Cipherteks: CQBSIBONW

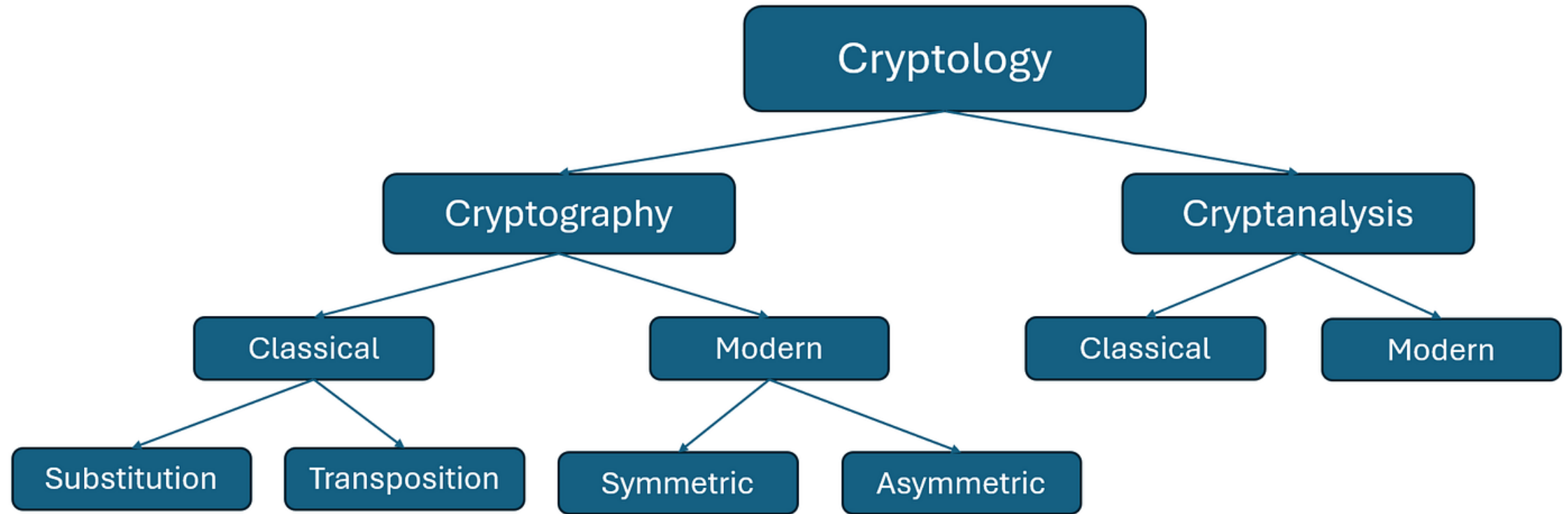
2. Teknik transposisi: mengubah susunan atau posisi huruf plainteks menjadi susunan huruf cipherteks.

Disebut juga teknik *scrambling*, permutasi, atau pengacakan

Contoh: Plainteks: MENGANTUK

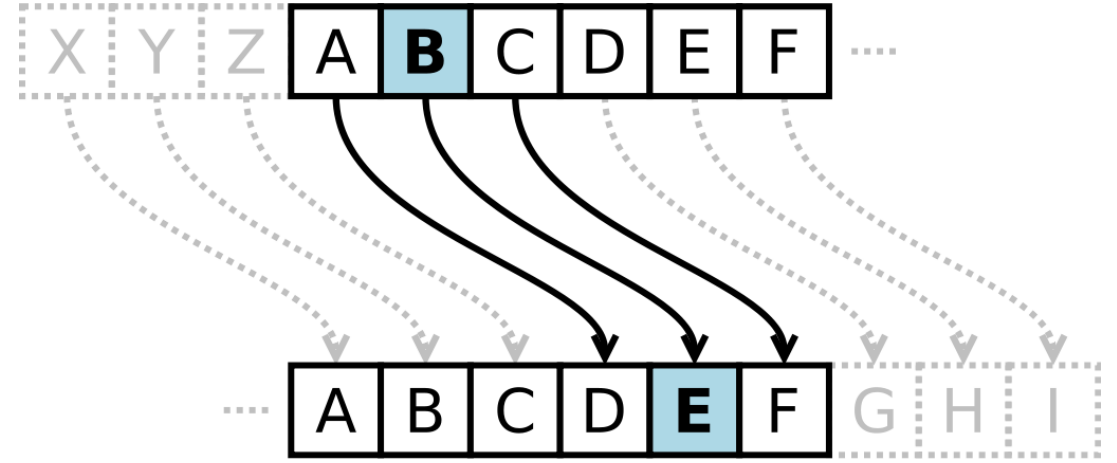
Cipherteks: TNEAKMNGU

- Oleh karena itu, dikenal dua macam *cipher* di dalam kriptografi klasik:
 1. *Cipher Substitusi (substitution Cipher)*
 - metode enkripsi dan dekripsi menggunakan teknik substitusi
 2. *Cipher Transposisi (transposition Cipher)*
 - metode enkripsi dan dekripsi menggunakan teknik transposisi
- Kombinasi kedua teknik tersebut membentuk *product cipher* atau *super enkripsi*
$$\text{product cipher} = \text{cipher substitusi} + \text{cipher transposisi}$$



A. Cipher Substitusi

- Contoh yang terkenal: *Caesar Cipher*
- Tiap huruf alfabet digeser 3 huruf ke kanan



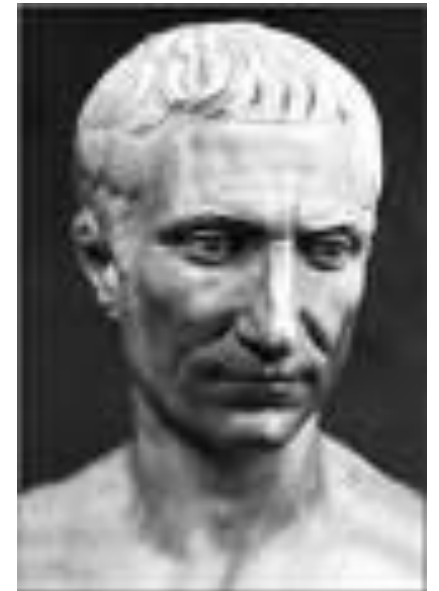
Plainteks : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Cipherteks : **D E F G H I J K L M N O P Q R S T U V W X Y Z A B C**

- Contoh:

Plainteks: temui saya di jembatan merah nanti malam

Cipherteks: WHPXL VDBD GL MHPEDWDQ PHUDK QDQWL PDODP



- Supaya lebih aman, cipherteks dikelompokkan ke dalam kelompok n -huruf, misalnya kelompok 4-huruf:

Semula: WHPXL VDBD GL MHPEDWDQ PHUDK QDQWL PDODP

Menjadi: WHPX LVDB DGLM HPED WDQP HUDK QDQW LPDO DP

- Atau membuang semua spasi:

WHPXLVDBDGLMHPEDWDQPHUDKQDQWLPDODP

- Tujuannya agar proses kriptanalisis menjadi lebih sulit dilakukan

- Enkripsi dan dekripsi Caesar Cipher dapat dilakukan secara matematis
- Misalkan setiap huruf alfabet dikodekan ke dalam integer dari 0 sampai 25 sebagai berikut:

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

maka, secara matematis Caesar Cipher dirumuskan sebagai:

Enkripsi: $c = E(p) = (p + 3) \bmod 26$

Dekripsi: $p = D(c) = (c - 3) \bmod 26$

Ket: p = plainteks; c = cipherteks

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

ENKRIPSI:

Plainteks: `temui saya di jembatan merah nanti malam`

- $p_1 = 't' = 19 \rightarrow c_1 = E(19) = (19 + 3) \bmod 26 = 22 = 'W'$
- $p_2 = 'e' = 4 \rightarrow c_2 = E(4) = (4 + 3) \bmod 26 = 7 = 'H'$
- $p_3 = 'm' = 12 \rightarrow c_3 = E(12) = (12 + 3) \bmod 26 = 15 = 'P'$
- $p_4 = 'u' = 20 \rightarrow c_4 = E(20) = (20 + 3) \bmod 26 = 23 = 'X'$
- $p_5 = 'i' = 8 \rightarrow c_4 = E(8) = (8 + 3) \bmod 26 = 11 = 'L'$
- dst...

Cipherteks: `WHPXL VDBD GL MHPEDWDQ PHUDK QDQWL PDODP`

DEKRIPSI:

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Cipherteks: WHPXL VDBD GL MHPEDWDQ PHUDK QDQWL PDODP

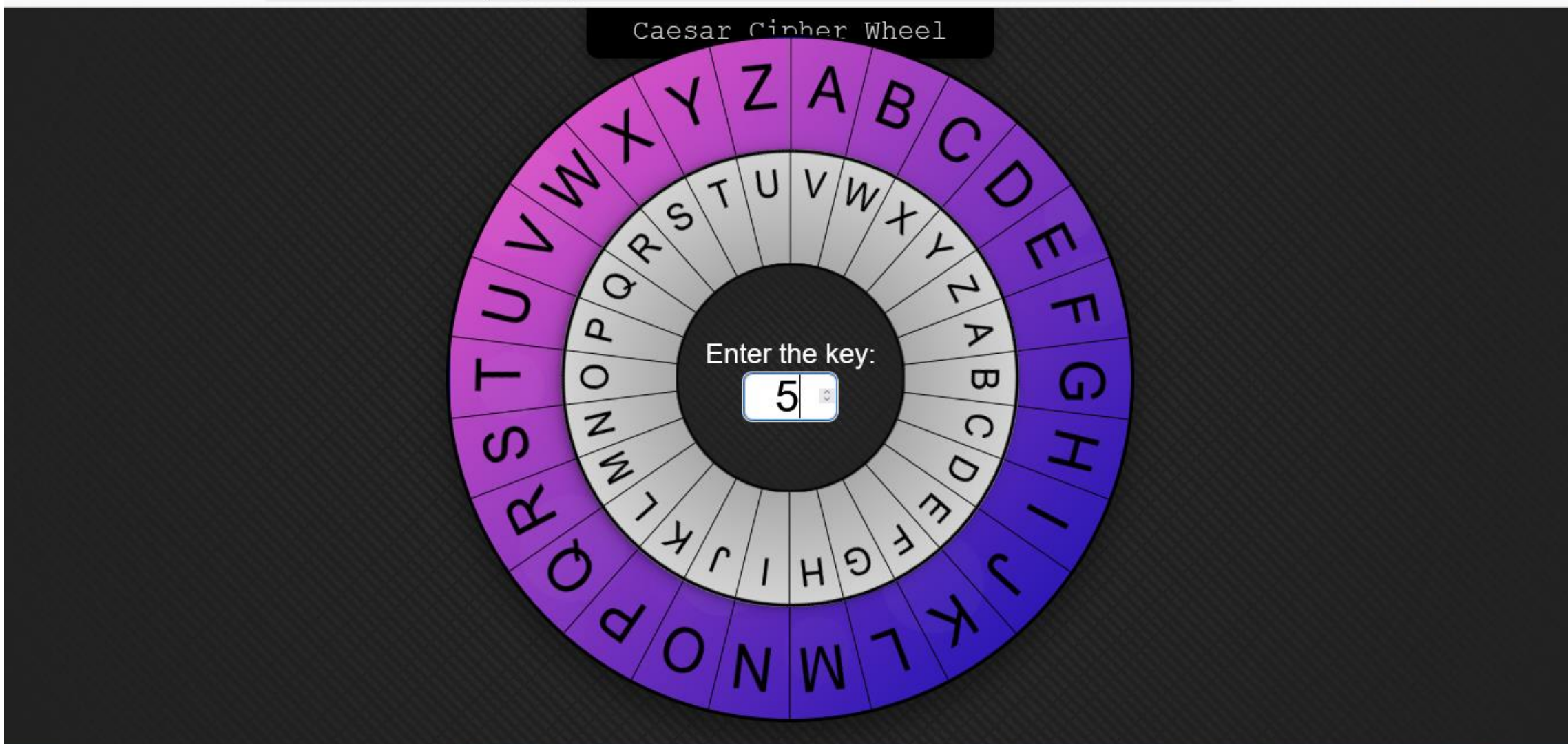
- $c_1 = 'W' = 22 \rightarrow p_1 = D(22) = (22 - 3) \bmod 26 = 19 = 't'$
- $c_2 = 'H' = 7 \rightarrow p_2 = D(7) = (7 - 3) \bmod 26 = 4 = 'e'$
- $c_3 = 'P' = 15 \rightarrow p_3 = D(15) = (15 - 3) \bmod 26 = 12 = 'm'$
- $c_4 = 'X' = 23 \rightarrow p_4 = D(23) = (23 - 3) \bmod 26 = 20 = 'u'$
- $c_5 = 'L' = 11 \rightarrow p_5 = D(11) = (11 - 3) \bmod 26 = 8 = 'i'$
- ...dst

Plainteks: temui saya di jembatan merah nanti malam



Caesar wheel untuk membentuk tabel substitusi huruf alfabet

Lihat demo online: <https://www.101computing.net/cipher-wheel.html>



- Secara umum, jika pergeseran huruf sejauh k , maka:

Enkripsi: $c = E(p) = (p + k) \bmod 26$

Dekripsi: $p = D(c) = (c - k) \bmod 26$

k = kunci rahasia

- Untuk alfabet berupa 256 karakter ASCII, maka:

Enkripsi: $c = E(p) = (p + k) \bmod 256$

Dekripsi: $p = D(c_i) = (c - k) \bmod 256$

k = kunci rahasia

Latihan

Enkripsi kalimat berikut dengan Caesar Cipher, $k = 5$

ADA RENCANA PENYELUNDUPAN NARKOBA DI BANDARA

Tentukan cipherteksnya

Demo Caesar Cipher Online: <https://cryptii.com/pipes/caesar-cipher>

The screenshot shows a web browser window with the URL <https://cryptii.com/pipes/caesar-cipher>. The page features the Cryptii logo in the top left and a red Adobe Creative Cloud banner in the top right. The main interface consists of three panels: a 'Plaintext' panel on the left containing the text 'The quick brown fox jumps over the lazy dog', a central 'Caesar cipher' panel with settings for a shift of 8 and 'Maintain case', and a 'Ciphertext' panel on the right containing the encoded text 'Bpm ycqks jzwev nwf rcuxa wdmz bpm ting lwo'. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons. The system clock in the bottom right corner displays '9:08 PM 2/3/2024'.

Caesar cipher: Encode and decode online

Open in **cryptii** ciphereditor

Program Caesar Cipher dalam Bahasa Python

```
[38]: def Caesar_cipher_encrypt(plaintexts, k):
    ciphertexts = ""
    for char in plaintexts:
        if char.isalpha():          # Hanya memproses huruf alfabet saja
            start = ord('A') if char.isupper() else ord('a')
            c = (ord(char) - start + k) % 26    # Kodekan huruf ke angka 0 s/d 25, lalu enkripsi dengan Caesar Cipher
            c = c + start                # Kembalikan ke posisi semula
            ciphertexts = ciphertexts + chr(c)    # sambung setiap huruf ciphertexts
        else:
            ciphertexts = ciphertexts + char    # Pesan yang bukan huruf tidak dienkripsi, dibiarkan saja
    return ciphertexts

[39]: def Caesar_cipher_decrypt(ciphertexts, k):
    plaintexts = ""
    for char in ciphertexts:
        if char.isalpha():          # Hanya memproses huruf alfabet saja
            start = ord('A') if char.isupper() else ord('a')
            c = (ord(char) - start - k) % 26    # Kodekan huruf ke angka 0 s/d 25 lalu dekripsi dengan Caesar Cipher
            c = c + start                # Kembalikan ke posisi semula
            plaintexts = plaintexts + chr(c)    # sambung setiap huruf plaintexts
        else:
            plaintexts = plaintexts + char    # Pesan yang bukan huruf tidak dienkripsi, dibiarkan saja
    return plaintexts
```

Run program

```
[40]: pesan = input("ketikkan pesan anda: ")
```

```
ketikkan pesan anda: Halo kawan, ini nomor PIN-ku: 123456, tolong jaga kerahasiannya
```

```
[41]: kunci = int(input("kunci: "))
```

```
kunci: 10
```

```
[42]: cipherteks = Caesar_cipher_encrypt(pesan, kunci)
```

```
[43]: print(f"Pesan terenkripsi: {cipherteks}")
```

```
Pesan terenkripsi: Rkvy ukgkx, sxs xywyb ZSX-ue: 123456, dyvyxq tkqk uobkrkcskxxik
```

```
[44]: plainteks = Caesar_cipher_decrypt(cipherteks, kunci)
```

```
[46]: print(f"Pesan hasil dekripsi: {plainteks}")
```

```
Pesan hasil dekripsi: Halo kawan, ini nomor PIN-ku: 123456, tolong jaga kerahasiannya
```

Program Caesar Cipher dalam Bahasa Python (256 karakter ASCII)

```
def Caesar_cipher_encryptV2(plaintexts, k):  
    cipherteks = ""  
    for char in plaintexts:  
        c = (ord(char) + k) % 256 # Kodekan huruf ke angka, lalu enkripsi dengan Caesar Cipher  
        cipherteks = cipherteks + chr(c) # sambung setiap huruf cipherteks  
    return cipherteks
```

```
def Caesar_cipher_decryptV2(cipherteks, k):  
    plaintexts = ""  
    for char in cipherteks:  
        c = (ord(char) - k) % 256 # Kodekan huruf ke angka, lalu dekripsi dengan Caesar Cipher  
        plaintexts = plaintexts + chr(c) # sambung setiap huruf plaintexts  
    return plaintexts
```

Run program

```
pesan = input("ketikkan pesan anda: ")
```

```
ketikkan pesan anda: Nomor telponku 08234654890
```

```
kunci = int(input("kunci (1-255): "))
```

```
kunci (1-255): 189
```

```
cipherteks = Caesar_cipher_encryptV2(pesan, kunci)
```

```
print(f"Pesan terenkripsi: {cipherteks}")
```

```
Pesan terenkripsi: ,*,/Ý1")-,+(2Ýíðïðñóðñðïí
```

```
plainteks = Caesar_cipher_decryptV2(cipherteks, kunci)
```

```
print(f"Pesan hasil dekripsi: {plainteks}")
```

```
Pesan hasil dekripsi: Nomor telponku 08234654890
```

Program Caesar Cipher dalam Bahasa C++

```
/ Program enkripsi dan dekripsi pesan dengan Caesar Cipher dalam Bahasa C++
#include <iostream>
#include <string.h>
using namespace std;

void enkripsi()
{
    string plainteks, cipherteks;
    int i, k;
    char c;

    cout << "Ketikkan pesan:";
    cin.ignore(); getline (cin, plainteks);
    cout << "Masukkan jumlah pergesaran (0-25): "; cin >> k;
    cipherteks = ""; // inisialisasi cipherteks dengan null string

    for (i=0; i < plainteks.length(); i++) {
        c = plainteks[i];
        if (isalpha(c)) { //hanya memproses huruf alfabet saja
            c = toupper(c); // ubah menjadi huruf kapital
            c = c - 65;      // kodekan huruf ke angka 0 s/d 25
            c = (c + k) % 26; // enkripsi, geser sejauh k ke kanan
            c = c + 65;      // kodekan kembali ke huruf semula
        }
        cipherteks = cipherteks + c; // sambungkan ke cipherteks
    }
    cout << "Cipherteks: "<<cipherteks<< endl; // cetak cipherteks
}
```

```

void dekripsi()
{
    string plainteks, cipherteks;
    int i, k;
    char c;

    cout << "Ketikkan cipherteks: ";
    cin.ignore();getline (cin, cipherteks);
    cout << "Masukkan jumlah pergesaran (0-25): ";
    cin >> k;
    plainteks = ""; // inisialisasi plainteks dengan null string

    for (i=0; i < cipherteks.length(); i++) {
        c = cipherteks[i];
        if (isalpha(c)) { //hanya memproses alfabet
            c = toupper(c); // ubah karakter ke huruf besar
            c = c - 65; // kodekan huruf ke angka 0 s/d 25
            if (c - k < 0) // kasus pembagian bilangan negatif
                c = 26 + (c - k);
            else
                c = (c - k) % 26;
            c = c + 65; // kodekan kembali ke huruf semula
            c = tolower(c); // plainteks dinyatakan sebagai huruf kecil
        }
        plainteks = plainteks + c; // sambungkan ke plainteks
    }
    cout << "Plainteks: " << plainteks << endl; // cetak plainteks
}

```

```
main()
{
    int pil; bool stop;
    stop = false;

    while (!stop) {
        cout << "Menu: " << endl;
        cout << "1. Enkripsi " << endl;
        cout << "2. Dekripsi " << endl;
        cout << "3. Exit      " << endl;
        cout << "Pilih menu: "; cin >> pil;
        switch (pil) {
            case 1 : enkripsi(); break;
            case 2 : dekripsi(); break;
            case 3 : stop = true; break;
        }
    }
}
```

```
Command Prompt

C:\data\Dataku\Buku\Buku Kriptografi\Edisi kedua>caesar
Menu:
1. Enkripsi
2. Dekripsi
3. Exit
Pilih menu: 1
Ketikkan pesan: the quick brown fox jumps over the lazy dog
Masukkan jumlah pergesaran (0-25): 18
Cipherteks:  LZW IMAUC TJGOF XGP BMEHK GNWJ LZW DSRQ UGY
Menu:
1. Enkripsi
2. Dekripsi
3. Exit
Pilih menu: 2
Ketikkan cipherteks: LZW IMAUC TJGOF XGP BMEHK GNWJ LZW DSRQ UGY
Masukkan jumlah pergesaran (0-25): 18
Plainteks: the quick brown fox jumps over the lazy dog
Menu:
1. Enkripsi
2. Dekripsi
3. Exit
Pilih menu: 3

C:\data\Dataku\Buku\Buku Kriptografi\Edisi kedua>
```

Kriptanalisis Caesar Cipher

- *Caesar cipher* mudah dipecahkan dengan *exhaustive key search (brute force)* karena jumlah kuncinya sangat sedikit (hanya ada 26 kunci).
- Coba lakukan dekripsi dengan berbagai nilai k dari 0 sampai 25, lalu periksa apakah hasil dekripsi merupakan kata atau kalimat yang bermakna. Jika ya, maka diduga k adalah kuncinya.
- Untuk memastikan k adalah kunci yang benar, maka cobakan k untuk potongan kriptogram lainnya.

Contoh: kriptogram XMZVH

Tabel 1. Contoh *exhaustive key search* terhadap cipherteks XMZVH

Kunci (k) <i>ciphering</i>	‘Pesan’ hasil dekripsi	Kunci (k) <i>ciphering</i>	‘Pesan’ hasil dekripsi	Kunci (k) <i>ciphering</i>	‘Pesan’ hasil dekripsi
0	XMZVH	17	GVIEQ	8	PERNZ
25	YNAWI	16	HWJFR	7	QFSOA
24	ZOBXJ	15	IXKGS	6	RGTPB
23	APCYK	14	JYLHT	5	SHUQC
22	BQDZL	13	KZMIU	4	TIVRD
21	CREAM	12	LANJV	3	UJWSE
20	DSFBN	11	MBOKW	2	VKXTF
19	ETGCO	10	NCPLX	1	WLYUG
18	FUHDP	9	ODQMY		

Plainteks yang potensial adalah CREAM dengan $k = 21$.

Kunci ini digunakan untuk mendekripsikan potongan cipherteks lainnya.

Contoh lain:

Cipherteks: PHHW PH DIWHU WKH WRJD SDUWB

	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
k						
0	phhw	ph	diwhu	wkh	wrjd	sduwb
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrpc	rfc	rmey	nyprw
6	...					
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

(Sumber: William Stallings)

Cipherteks: VIVBQ SQBI SMBMUC LQ ICTI

k	Hasil dekripsi
0	vivbq sqbi smbduc lq icti
1	uhup rpah rlaltb kp hbsh
2	tgtzo qozg qkzksa jo garg
3	sfsyn pnyf pjyjrz in fzqf
4	rerxm omxe oixiqy hm eyep
5	qdqwl nlwd nhwhpx gl dxod
6	pcpuk mkvc mgvgow fk cwnc
7	obouj ljub lfufnu ej bvmb
8	nanti kita ketemu di aula
9	mzmsh jhsz jdsdlt ch ztkz
10	lylrg igry icrcks bg ysyy
11	kxkqf hfqx hbqbjr af xrix
12	jwjpe gepw gapaiq ze wqhw
13	iviod fdov fzozhp yd vpgv
14	huhnc ecnu eynygo xc uofu
15	gtgmb dbmt dxmxfn wb tnet
16	fsfla cals cwlwem va smds
17	erekz bzkr bvkvd1 uz r1cr
18	dqdjy ayjq aujuck ty qkbq
19	cpcix zxip ztitbj sx pjap
20	bobhw ywho yshsai rw oizo
21	anagv xvgx xrfqyg pu mgxm
22	xmzfu wufm wqfqyg pu mgxm
23	ylzet vtet vpepxf ot lfwl
24	xkxds usdk uodowe ns kevk
25	wjwcr trcj tncnvd mr jduj

- Bagaimana jika terdapat dua atau lebih nilai k yang menghasilkan pesan-pesan bermakna?

Contoh: Misalkan kriptogram `HSPPW` menghasilkan dua kemungkinan kunci yang potensial, yaitu:

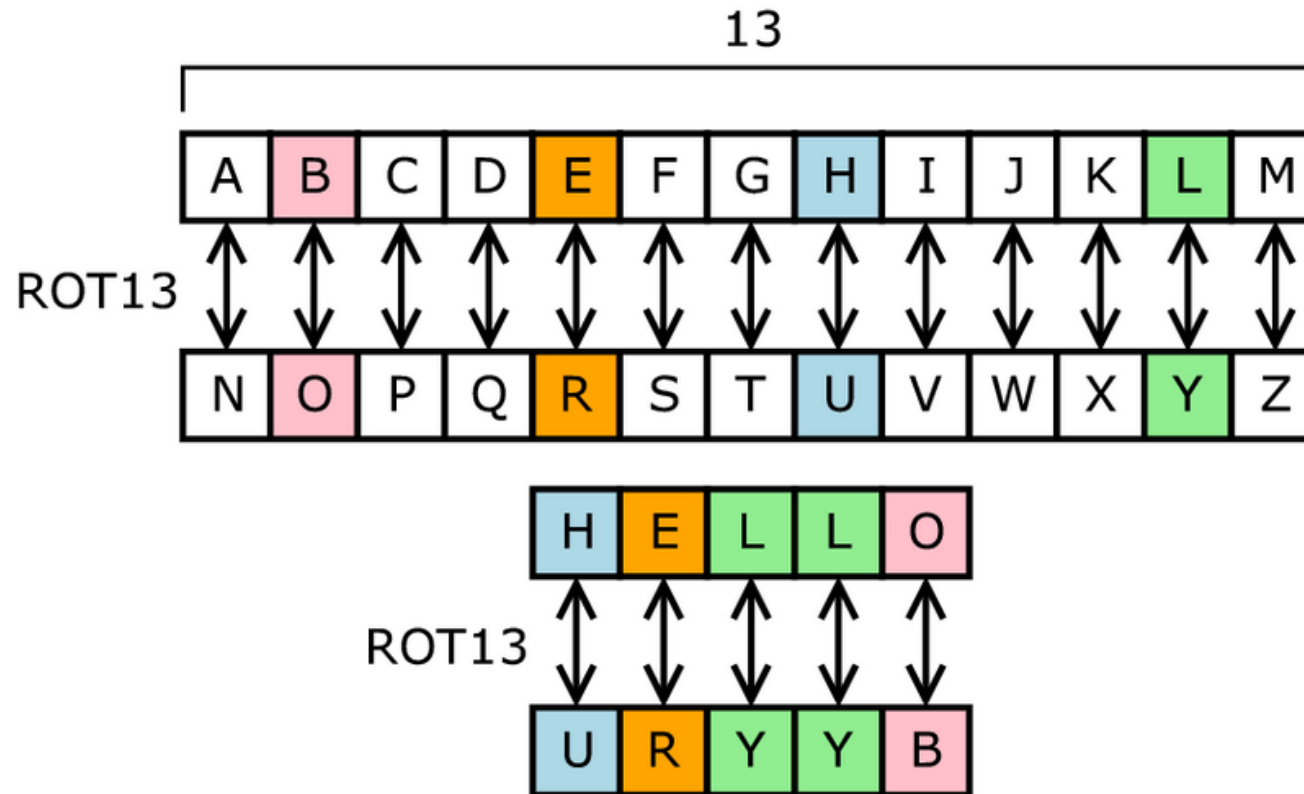
$k = 4$ menghasilkan pesan `dolls` (boneka)

$k = 11$ menghasilkan `wheel` (roda) .

Nilai k mana yang benar?

Jika kasusnya demikian, maka lakukan dekripsi terhadap potongan cipherteks lain tetapi cukup menggunakan $k = 4$ dan $k = 11$ agar dapat disimpulkan kunci mana yang benar.

- Di dalam sistem operasi Unix, ROT13 adalah fungsi menggunakan *Caesar cipher* dengan pergeseran $k = 13$



Sumber gambar: Wikipedia

- Contoh: ROT13 (ROTATE) = EBGNGR
- Nama “ROT13” berasal dari *net.jokes*
(<http://groups.google.com/group/net.jokes>) (tahun 1980)
- ROT13 biasanya digunakan di dalam forum *online* untuk menyandikan jawaban teka-teki, kuis, canda, dsb
- Enkripsi arsip dua kali dengan ROT13 menghasilkan pesan semula:

$$P = \text{ROT13}(\text{ROT13}(P))$$
 sebab $\text{ROT}_{13}(\text{ROT}_{13}(x)) = \text{ROT}_{26}(x) = x$
- Jadi dekripsi cukup dilakukan dengan mengenkripsi cipherteks kembali dengan ROT13

B. Cipher Transposisi

- Cipherteks diperoleh dengan mengubah posisi huruf di dalam plainteks.
- Dengan kata lain, algoritma ini melakukan *transpose* terhadap rangkaian huruf di dalam plainteks.
- Nama lain untuk metode ini adalah *cipher permutasi*, karena *transpose* setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut.
- Contoh cipher transposisi: *Columnar Transposition Cipher, Rail Fence Transposition Cipher*

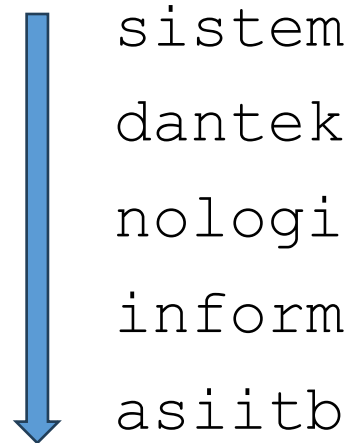
1. Columnar Transposition Cipher

Contoh: Misalkan plainteks adalah

sistem dan teknologi informasi itb

Panjang kunci = 6

Enkripsi: (*buang spasi)



sistem
dantek
nologi
inform
asiitb

Cipherteks: (baca secara vertical kolom per kolom)

SDNIAIAONSSNLFITTOOIEEGR

(tanpa spasi)

TMKIMB
SDNI AIAO NSSN LFIT TOOI EEGR TMKI MB

(atau blok 4 huruf)

Dekripsi: Bagi panjang cipherteks dengan panjang kunci (Pada contoh ini, $30 / 6 = 5$) .

Cipherteks: **SDNIAIAONSSNLFITTOOIEEGRTMKIMB**

Tulis cipherteks secara vertical sepanjang 5 kolom:

SDNIA

IAONS

SNLFI

TTOOI

EEGRT

MKIMB

Plainteks: (baca secara vertikal)

sistemdanteknologiinformasiitb

sistem dan teknologi informasi itb

(tambahkan spasi)

- Untuk membuat enkripsi menjadi lebih kompleks, gunakan kata kunci sepanjang n dengan huruf-huruf berbeda. Urutan huruf di dalam kata kunci menentukan urutan pembacaan secara vertikal.
- Contoh: kata kunci = TOMBAK (urutan huruf sesuai alfabet: 6 5 4 2 1 3)

Plainteks: sistem dan teknologi informasi itb

Enkripsi:

123456

TOMBAK

sistem

dantek

nologi

inform

asiitb

Cipherteks: (baca secara vertikal sesuai urutan huruf di dalam kata kunci → 5, 4, 6, 3, 2, 1)

EEGRTTTOOIMKIMBSNLFIIAONSSDNIA

Demo online: <https://www.dcode.fr/columnar-transposition-cipher>

The screenshot shows a web browser window with the URL <https://www.dcode.fr/columnar-transposition-cipher>. The page features a search bar at the top left with the text "Search for a tool" and a search input field containing "e.g. type 'boolean'". Below the search bar, there are results for "TOLONGJEMPUT...LAM" and "TGUKNLOJTKTALAUIMOMNNMNPAAA".

The main content area is titled "COLUMNAR TRANSPOSITION CIPHER" and includes a sub-header "Cryptography > Transposition Cipher > Columnar Transposition Cipher". The tool is divided into two main sections: "COLUMNAR TRANSPOSITION DECODER" and "COLUMNAR TRANSPOSITION ENCODER".

The "COLUMNAR TRANSPOSITION DECODER" section has a text input field for "COLUMNAR TRANSPOSITION CIPHERTEXT" containing the text "AGTAMKDAHAUULLPRSAUIEMYLRITASAAEDINNLI". Below this, there are checkboxes for "KEEP SPACES, PUNCTUATION (AND OTHER CHARACTERS)" and a dropdown menu for "PLAINTEXT (PRESUMED) LANGUAGE" set to "English". The "DECRYPTION METHOD" section has two radio buttons: "WITH THE ENCRYPTION KEY OR PERMUTATION" (selected) and "TRY SOME PERMUTATIONS (BRUTEFORCE UP TO SIZE 6)". The "WITH THE ENCRYPTION KEY OR PERMUTATION" method has a text input field for the key containing "12345678" and a formula $\rightarrow (1,2,3,4,5,6,7,8) \Rightarrow (1,2,3,4,5,6,7,8)^{-1}$. The "GRID WRITING/READING ENCRYPTION DIRECTIONS" section has a dropdown menu for "MODE" set to "Write by rows, read by columns (by default)". A "DECRYPT" button is located below the decryption method section.

The "COLUMNAR TRANSPOSITION ENCODER" section has a text input field for "COLUMNAR TRANSPOSITION PLAIN TEXT" containing the text "TOLONGJEMPUTANAKKUNANTIMALAM".

On the right side of the page, there is a "Summary" section with a "Feedback" button. The summary includes links to "Columnar Transposition Decoder", "Columnar Transposition Encoder", "What is a Columnar Transposition cipher? (Definition)", "How to encrypt using a Columnar Transposition cipher?", "How to decrypt with a Columnar Transposition cipher?", "How to recognize a Columnar Transposition ciphertext?", and "How to decipher a Columnar Transposition without the key?". Below the summary, there is a "Similar pages" section with links to "Caesar Box Cipher", "Mono-alphabetic Substitution", "ADFGVX Cipher", "ADFGX Cipher", "Spiral Cipher", "Skip Cipher", "Redefence Cipher", and "dCode's TOOL S.I.U.S.T".

The bottom of the page shows a Windows taskbar with various application icons and a search bar.

Program Transposition Cipher dalam Bahasa Python (26 huruf alfabet)

```
import math

def transposition_cipher_encrypt(plaintext, k):
    ciphertext = [''] * k          # membuat sebuah list dengan panjang k, yang setiap
                                   # elemennya berupa string kosong.
    for kolom in range(k):
        index = kolom
        while index < len(plaintext):
            ciphertext[kolom] = ciphertext[kolom] + plaintext[index]
            index = index + k
    return ''.join(ciphertext)
```

```

def transposition_cipher_decrypt(ciphertext, k):
    jumlah_baris = math.ceil(len(ciphertext) / k)
    jumlah_sel_kosong = (jumlah_baris * k) - len(ciphertext)

    plaintext = [''] * jumlah_baris      # membuat sebuah list dengan panjang k, yang
                                         # setiap elemennya berupa string kosong.

    kolom = 0
    baris = 0

    for karakter in ciphertext:
        plaintext[baris] = plaintext[baris] + karakter
        baris = baris + 1

        if (baris == jumlah_baris) or (baris == jumlah_baris-1 and
                                         kolom >= k - jumlah_sel_kosong):
            baris = 0
            kolom += 1

    return ''.join(plaintext)

```

Run program

```
pesan = input("ketikkan pesan anda: ")
k = int(input("Masukkan panjang kunci: "))
ciphertext = transposition_cipher_encrypt(pesan, k)
print(ciphertext)
```

```
ketikkan pesan anda: banjir bandang melanda sumatera
Masukkan panjang kunci: 5
brdmdmaa aeaanbnl tjagasein nur
```

```
pesan = input("ketikkan ciphertext: ")
k = int(input("Masukkan panjang kunci: "))
plaintext = transposition_cipher_decrypt(pesan, k)
print(plaintext)
```

```
ketikkan ciphertext: brdmdmaa aeaanbnl tjagasein nur
Masukkan panjang kunci: 5
banjir bandang melanda sumatera
```

2. Rail Fence Transposition Cipher

Contoh lain. Misalkan plainteks adalah

CRYPTOGRAPHY AND DATA SECURITY

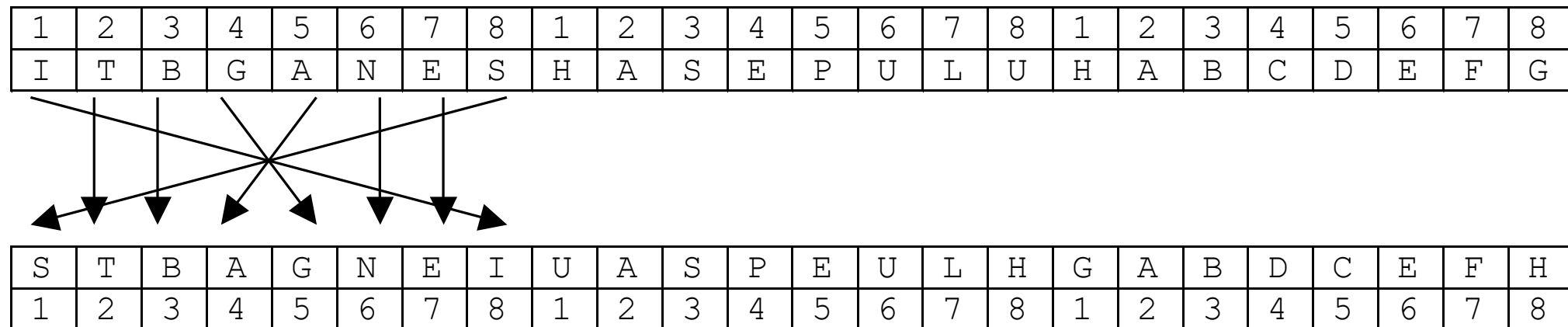
Plainteks disusun menjadi 3 baris ($k = 3$) seperti di bawah ini:

C		T		A		A		A		E		I	
R	P	O	R	P	Y	N	D	T	S	C	R	T	
	Y		G			H		D		A		U	Y

maka cipherteksnya adalah

CTAAAEIRPORPYNDTSCRTYGHDAUY

- Kita pun dapat membuat variasi cipher transposisi dengan aturan yang kita definisikan sendiri.
- Contohnya seperti berikut. Misalkan plainteks: ITB GANESHA SEPULUH
- Bagi menjadi blok-blok 8-huruf. Jika < 8 , tambahkan huruf *dummy*.



- Cipherteks: **STBAGNEIUASPEULHGABDCEFH**

Super-enkripsi

- Menggabungkan *cipher* substitusi dengan *cipher* transposisi.
- Disebut juga *product cipher*
- Mula-mula pesan dienkripsi dengan *cipher* substitusi, selanjutnya hasilnya dienkripsi dengan *cipher* transposisi (atau sebaliknya).

Contoh. Plainteks `hello world`

- ✓ Mula-mula dienkripsi dengan *caesar cipher* ($k = 3$) menjadi `KHOOR ZRUOG`
- ✓ kemudian hasil ini dienkripsi lagi dengan *cipher* transposisi ($k = 4$):

`KHOO`

`RZRU`

`OGZZ`

→ Cipherteks akhir adalah: **KROHZGORZOUZ**

- *Cipher* modern menggunakan konsep kombinasi *cipher* substitusi dan *cipher* transposisi, namun operasinya dibuat sekompleks mungkin

Bersambung