Bahan kuliah II4021 Kriptografi

02 – Ragam Cipher Klasik

(Bagian 1)

Oleh: Rinaldi Munir

Program Studi Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2025

Pendahuluan

- Kriptografi klasik (*classical cryptography*) merupakan kriptografi yang sudah tua, sudah ada sejak ribuan tahun yang lalu sampai dtemukan computer digital
- *Cipher* pada kriptografi klasik, dinamakan *cipher* klasik, (*classical cipher*) hanya memproses pesan berupa 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

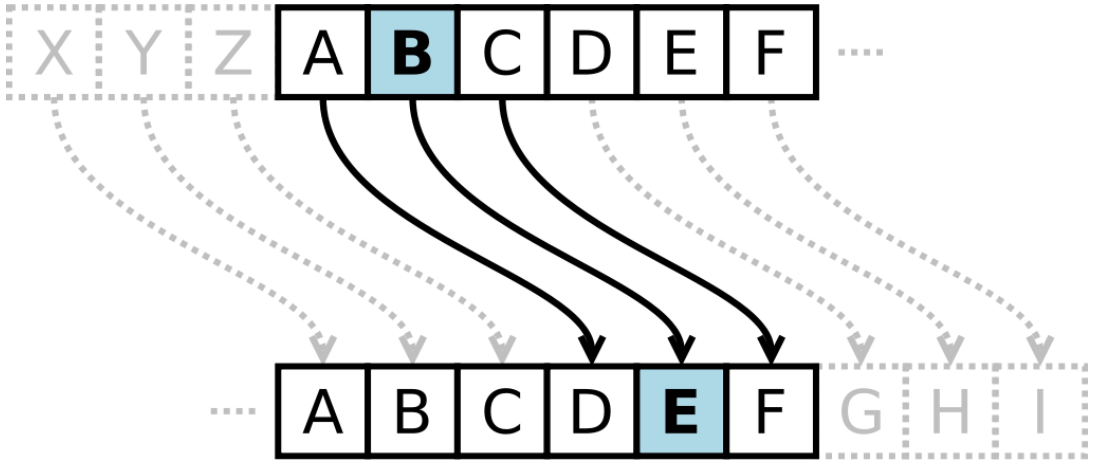
  2. *Cipher* Transposisi (*transposition Cipher*)

     - metode enkripsi dan dekripsi menggunakan teknik transposisi

- Kombinasi kedua teknik tersebut membentuk *product cipher* atau *super enkripsi*

  product cipher = cipher substitusi + cipher transposisi

A. Cipher Substitusi



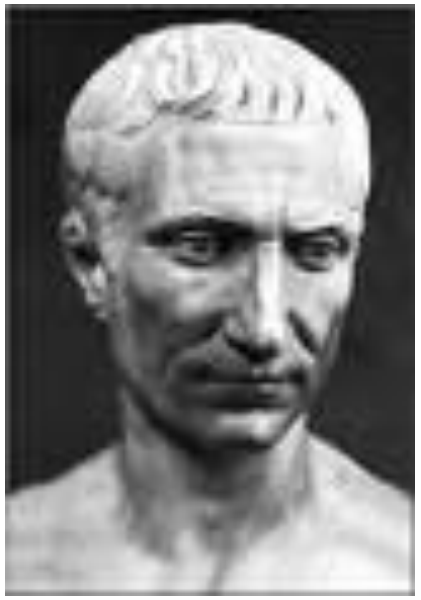
- Contoh yang terkenal: *Caesar Cipher*
- Tiap huruf alfabet digeser 3 huruf ke kanan

```
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: `awasi asterix dan temannya obelix`

Cipherteks: `DZDVL DVWHULA GDQ WHPDQQBA REHOLA`



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

  Semula: `DZDVL DVWHULA GDQ WHPDQQBA REHOLA`

  Menjadi: `DZDV LDVW HULA GDQW HPDQ QBAR EHOL A`

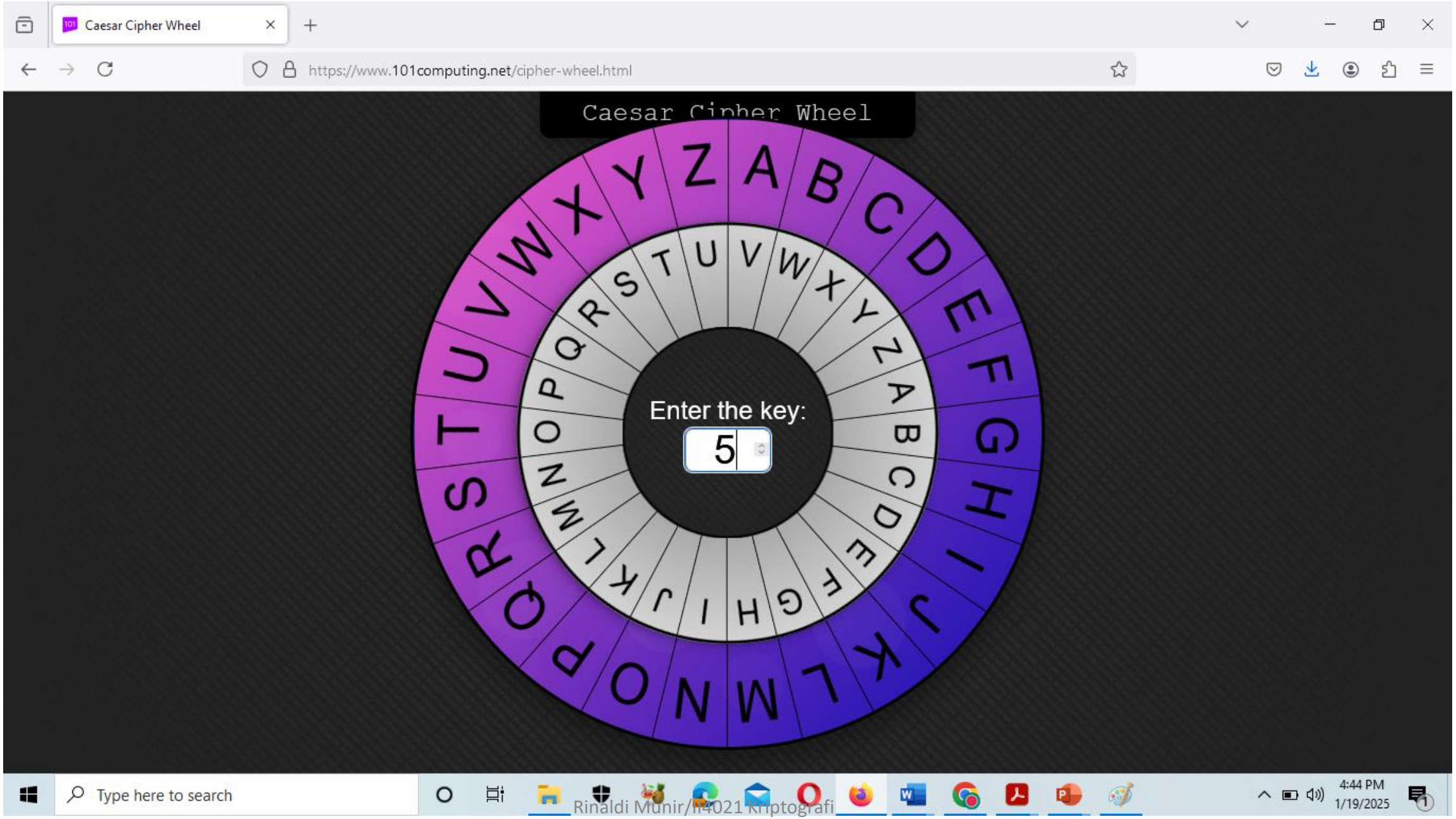
- Atau membuang semua spasi:

  `DZDVLDVWHULAGDQWHPDQQBAREHOLA`

- Tujuannya agar proses kriptanalisis menjadi lebih sulit dilakukan

Caesar wheel untuk membentuk tabel substitusi huruf alfabet

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


Caesar Cipher Wheel
https://www.101computing.net/cipher-wheel.html
Caesar Cipher Wheel
Enter the key:
5
Type here to search
4:44 PM
1/19/2025

Latihan

Enkripsi kalimat berikut dengan Caesar Cipher, k = 5

ADA RENCANA PENYELUNDUPAN NARKOBA DI BANDARA

- Misalkan setiap huruf alfabet dikodekan ke dalam integer dari 0 sampai 25 sebagai berikut:

  A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8, J = 9, K = 10

  L = 11, M = 12, N = 13, O = 14, P = 15, Q = 16, R = 17, S = 18, T = 19, U = 20

  V = 21, W = 22, X = 23, Y = 24, Z = 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

ENKRIPSI:

Plainteks: `awasi asterix dan temannya obelix`

- p_1 = ‘a’ = 0 → c_1 = $E(0)$ = (0 + 3) **mod** 26 = 3 = ‘D’
- p_2 = ‘w’ = 22 → c_2 = $E(22)$ = (22 + 3) **mod** 26 = 25 = ‘Z’
- p_3 = ‘a’ = 0 → c_3 = $E(0)$ = (0 + 3) **mod** 26 = 3 = ‘D’
- p_4 = ‘s’ = 18 → c_4 = $E(18)$ = (18 + 3) **mod** 26 = 21 = ‘V’
- p_5 = ‘i’ = 8 → c_4 = $E(8)$ = (8 + 3) **mod** 26 = 11 = ‘L’
- dst…

Cipherteks: `DZDVL DVWHULA GDQ WHPDQQBA REHOLA`

DEKRIPSI:

Cipherteks: `DZDVL DVWHULA GDQ WHPDQQBA REHOLA`

- c_1 = 'D' = 3 → $p_1 = D(3) = (3 - 3)$ **mod** 26 = 0 = '`a`'
- c_2 = 'Z' = 25 → $p_2 = D(25) = (25 - 3)$ **mod** 26 = 22 = '`w`'
- c_3 = 'D' = 3 → $p_3 = D(3) = (3 - 3)$ **mod** 26 = 0 = '`a`'
- ...
- c_{12} = 'A' = 0 → $p_{12} = D(0) = (0 - 3)$ **mod** 26 = –3 **mod** 26 = (26 – 3) mod 26 = 23 = '`x`'

- Plainteks ditemukan kembali: `awasi asterix dan temannya obelix`

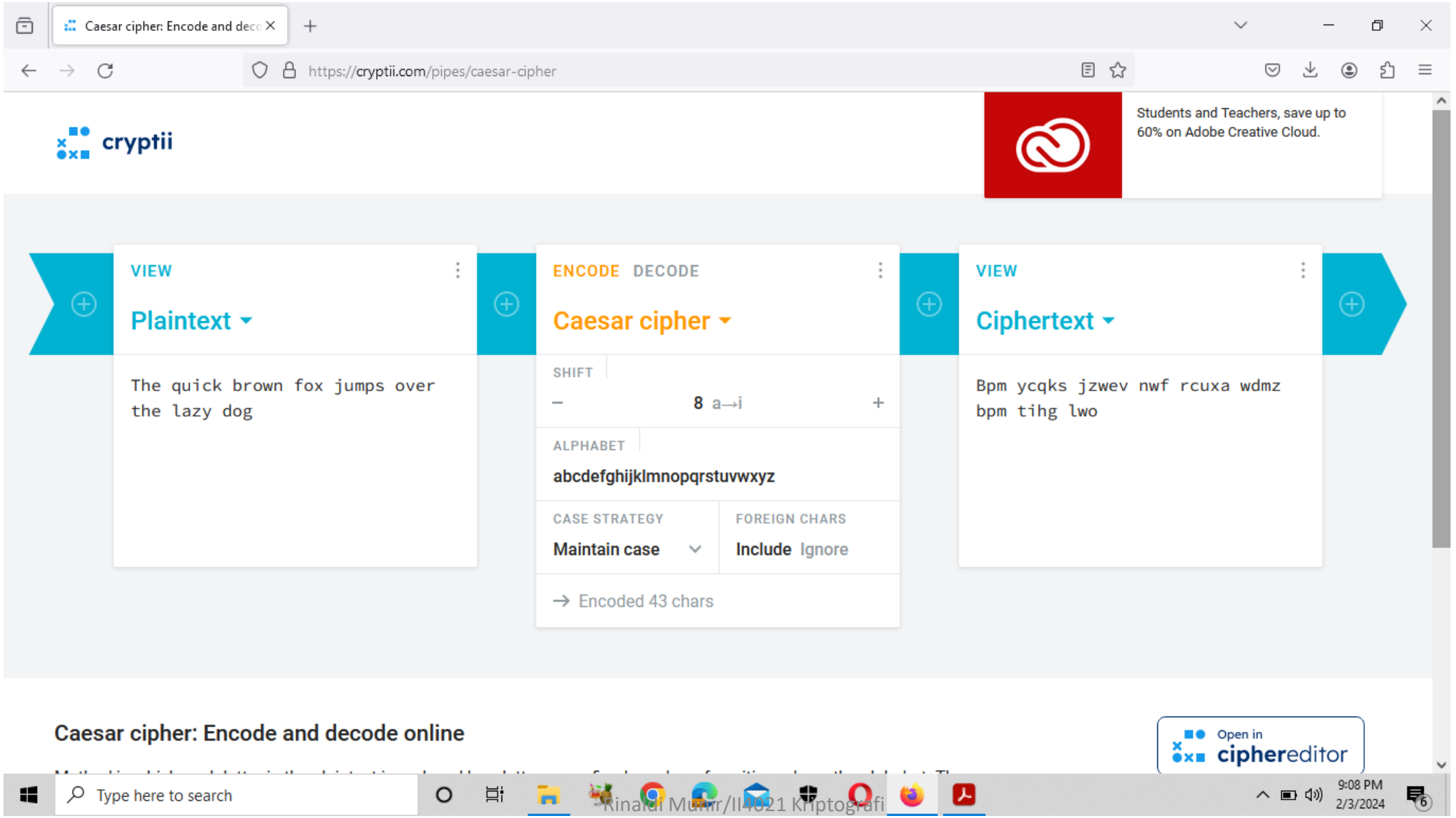
- Secara umum, jika pergeseran huruf sejauh k, maka:

  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

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



Program Caesar Cipher dalam Bahasa Python

[38]:
```python
def Caesar_cipher_encrypt(plainteks, k):
    cipherteks = ""
    for char in plainteks:
        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
           cipherteks = cipherteks + chr(c)      # sambung setiap huruf cipherteks
        else:
           cipherteks = cipherteks + char     # Pesan yang bukan huruf tidak dienkripsi, dibiarkan saja
    return cipherteks
```

[39]:
```python
def Caesar_cipher_decrypt(cipherteks, k):
    plainteks = ""
    for char in cipherteks:
        if char.isalpha():           # Hnaya 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
           plainteks = plainteks + chr(c)   #  sambung setiap huruf plainteks
        else:
           plainteks = plainteks + char    # Pesan yang bukan huruf tidak dienkripsi, dibiarkan saja
    return plainteks
```

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 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
}
```

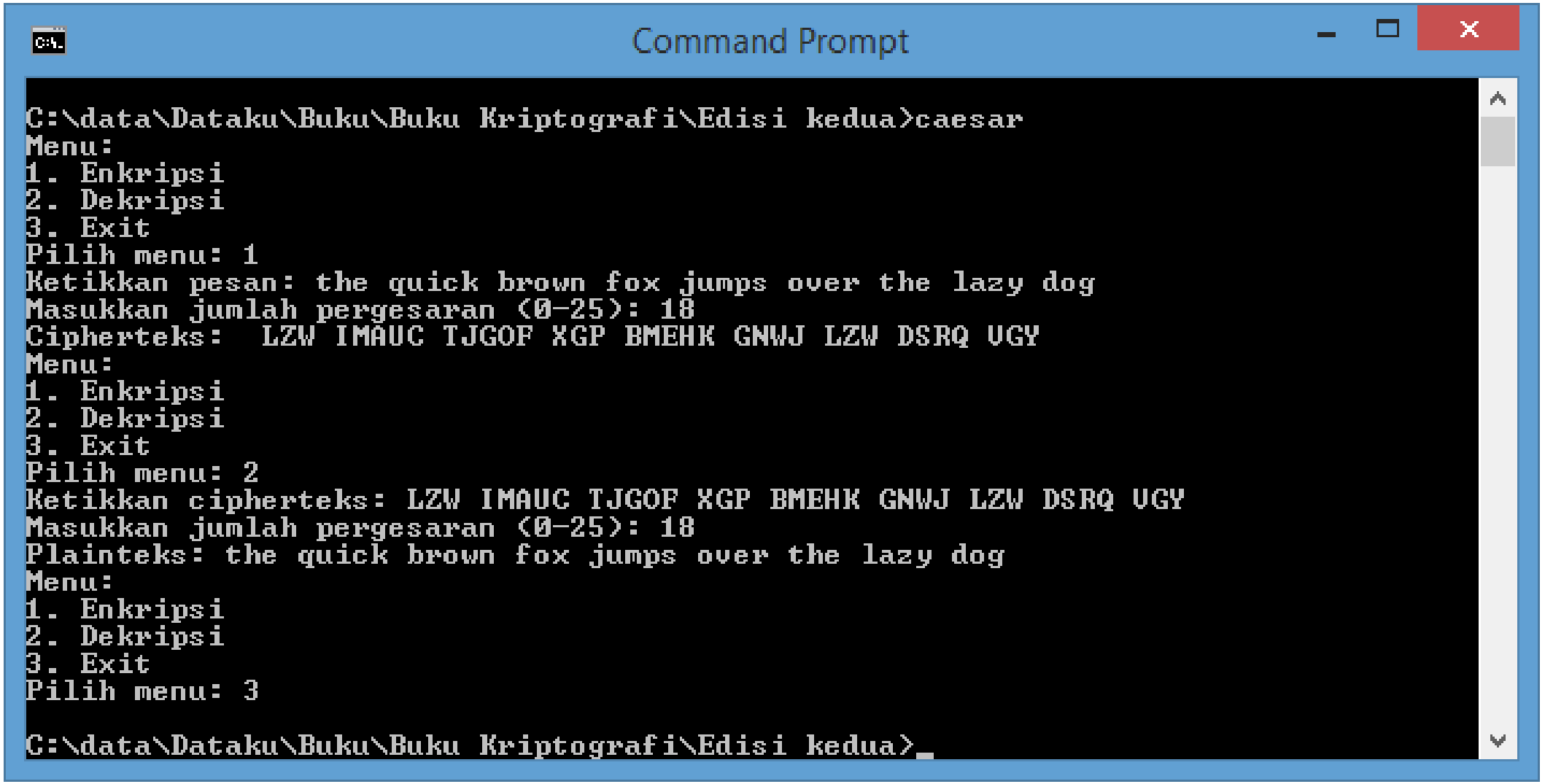
```
void dekripsi()
{
  string plainteks, cipherteks;
  int i, k;
  char c;

  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 VGY
Menu:
1. Enkripsi
2. Dekripsi
3. Exit
Pilih menu: 2
Ketikkan cipherteks: LZW IMAUC TJGOF XGP BMEHK GNWJ LZW DSRQ VGY
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*) *ciphering*	'Pesan' hasil dekripsi	Kunci (*k*) *ciphering*	'Pesan' hasil dekripsi	Kunci (*k*) *ciphering*	'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

k	PHHW PH DIWHU WKH WRJD SDUWB
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 ydrcp 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 smbmuc lq icti
1	uhuap rpah rlaltb kp hbsh
2	tgtzo qozg qkzksa jo garg
3	sfsyn pnyf pjyjrz in fzqf
4	rerxm omxe oixiqy hm eype
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 ysjy
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 bvkvdl uz rlcr
18	dqdjy ayjq aujuck ty qkbq
19	cpcix zxip ztitbj sx pjap
20	bobhw ywho yshsai rw oizo
21	anagv xvgn xrfqyg pu mgxm
22	xmzfu wufm wqfqyg pu mgxm
23	ylyet vtel 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:

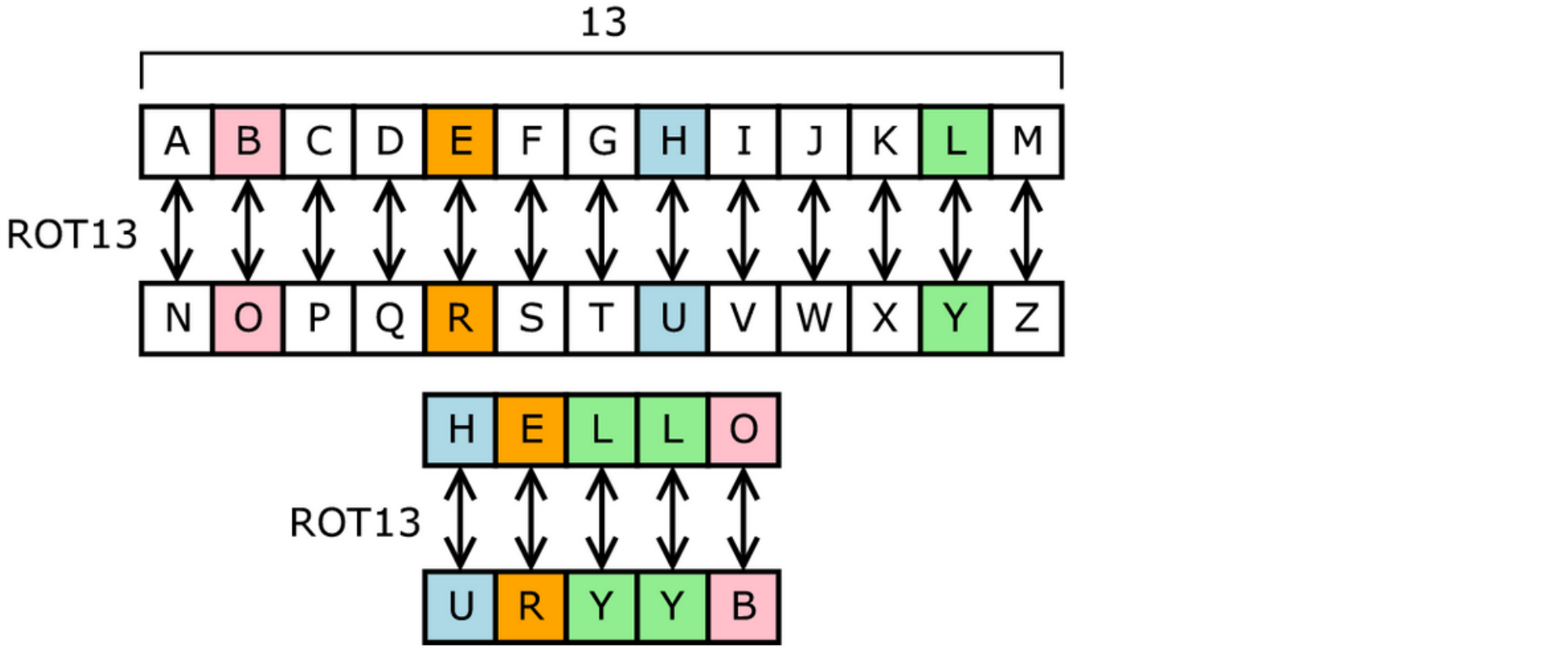
k = 4 menghasilkan pesan `dolls` `(boneka)`

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* (*hhtp://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

Jenis-jenis *Cipher* Substitusi

1. ***Cipher* abjad-tunggal** (*monoalphabetic cipher*)

   - setiap huruf plainteks diganti dengan satu huruf cipherteks

2. ***Cipher* substitusi homofonik** (*Homophonic substitution cipher*)

   - setiap huruf plainteks diganti dengan salah satu huruf atau pasangan huruf cipherteks yang mungkin.

3. ***Cipher* abjad-majemuk** (*Polyalpabetic substitution cipher*)

   - setiap huruf plainteks diganti menggunakan kunci yang berbeda.

4. **Cipher substitusi poligram** (*Polygram substitution cipher*)

   - setiap pasangan huruf plainteks diganti dengan pasangan huruf cipherteks

1. Cipher abjad-tunggal (*monoalphabetic cipher*)

- Pada cipher abjad-tunggal, satu huruf plainteks diganti dengan satu huruf cipherteks yang bersesuaian.

- *Caesar cipher* adalah salah satu *cipher* yang tergolong ke dalam *cipher* abjad-tunggal dengan tabel substitusi berupa hasil dari pergeseran tiga huruf ke kanan.

- Secara umum, kita dapat membentuk tabel subtitusi sembarang. Jumlah kemungkinan tabel substitusi yang dapat dibuat pada sembarang *cipher* abjad-tunggal adalah sebanyak

  26! = 403.291.461.126.605.635.584.000.000

  karena ada 26! cara mempermutasikan 26 huruf alfabet.

- Tabel substitusi dapat dibentuk secara acak:

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

- Atau berdasarkan kalimat yang mudah diingat:

  Contoh: `di bawah sinar bulan purnama hati resah jadi senang`

  Buang duplikasi huruf menjadi: `dibawhsnrulpmtejg`

  Sambung dengan huruf lain yang belum ada:

  `dibawhsnrulpmtejgcfkoqvwxyz`

  Tabel substitusi:

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

2. Cipher Substitusi Homofonik

(*Homophonic substitution cipher*)

- Setiap huruf plainteks dipetakan ke dalam salah satu huruf atau salah satu pasangan huruf cipherteks yang mungkin.

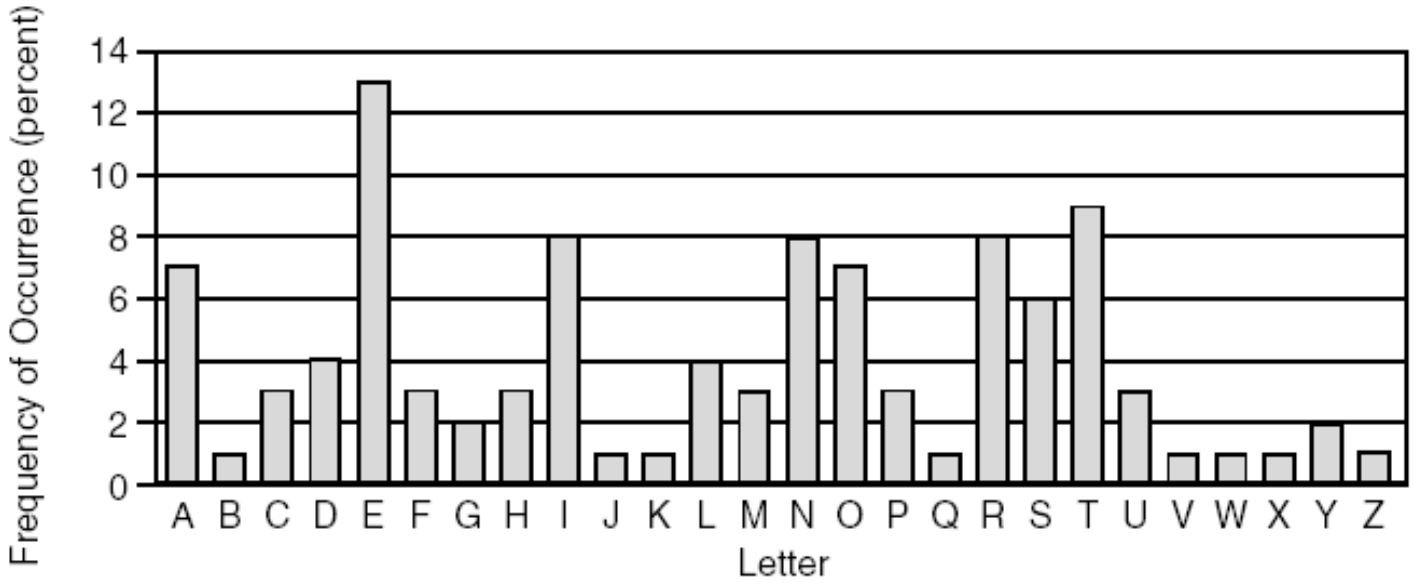
- Tujuan: menyembunyikan hubungan statistik antara plainteks dengan cipherteks

- Fungsi *ciphering* memetakan satu-ke-banyak (*one-to-many*).

Misal: huruf `E` → `AB`, `TQ`, `YT`, `UX` (homofon)

huruf `B` → `EK`, `MF`, `KY` (homofon)

- Contoh: Sebuah teks dengan frekuensi kemunculan huruf sbb:



- Huruf E muncul 13 % → E dapat dikodekan dengan 13 homofon

Huruf Plainteks	Pilihan untuk unit cipherteks
A	BU, TX, YR, MB, OP, TF, QA
B	ER, FY
C	IU, CW, PL
D	NQ, VT, OA, GP
E	ZX, BR, JO, EW, HT, KC, ND, SO, BO, VE, KL, JU, HR
F	EP, MS
G	TW, HL
H	OU, HE, JK, AT, KY, IQ
I	GT, UA, CN, HI, WO, ZF, FI
J	OC
K	LV
L	TY, JO, DR, ML
M	GR, KU
N	BE, TF, XO, LG, PS, CD, IE
O	YA, HU, VS, KP, BD, JZ, OL
P	IR, JA
Q	SP
R	UL, XP, TA, RL, LW, DO
S	EQ, IF, TK, PN, GL, TB
T	SI, GD, KI, MA, EL, ET, MS, MT, TL
U	FA, BI, SF
V	GM
W	TG, AS
X	FI, TM
Y	SR, DS
Z	AR

- Pasangan huruf di dalam tabel substitusi ini dibentuk secara acak
- Tidak boleh ada pasangan huruf yang sama
- Enkripsi dan dekripsi menggunakan tabel
- Tabel substitusi tersebut berlaku sebagai kunci, harus dirahasiakan

- Unit cipherteks mana yang dipilih diantara semua homofon ditentukan secara acak.
- Contoh:

  Plainteks: `kripto`

  Cipherteks: `LV TA FI JA MS KP`

- Enkripsi: satu-ke-banyak
- Dekripsi: satu-ke-satu
- Dekripsi menggunakan tabel homofon yang sama.

Huruf Plainteks	Pilihan untuk unit cipherteks
A	BU, TX, YR, MB, OP, TF, QA
B	ER, FY
C	IU, CW, PL
D	NQ, VT, OA, GP
E	ZX, BR, JO, EW, HT, KC, ND, SO, BO, VE, KL, JU, HR
F	EP, MS
G	TW, HL
H	OU, HE, JK, AT, KY, IQ
I	GT, UA, CN, HI, WO, ZF, FI
J	OC
K	LV
L	TY, JO, DR, ML
M	GR, KU
N	BE, TF, XO, LG, PS, CD, IE
O	YA, HU, VS, KP, BD, JZ, OL
P	IR, JA
Q	SP
R	UL, XP, TA, RL, LW, DO
S	EQ, IF, TK, PN, GL, TB
T	SI, GD, KI, MA, EL, ET, MS, MT, TL
U	FA, BI, SF
V	GM
W	TG, AS
X	FI, TM
Y	SR, DS
Z	AR

3. Cipher Abjad-Majemuk

(*Polyalpabetic substitution cipher*)

- *Cipher* abjad-tunggal: satu kunci untuk semua huruf plainteks
- *Cipher* abjad-majemuk: setiap huruf menggunakan kunci berbeda.
- *Cipher* abjad-majemuk dibuat dari sejumlah *cipher* abjad-tunggal, masing-masing dengan kunci yang berbeda.

Contoh 1: (spasi dibuang)

P : kriptografiklasikdengancipheralfabetmajemuk

K : LAMPIONLAMPIONLAMPIONLAMPIONLAMPIONLAMPIONL

C : VRUEBCTCARXSZNDIWSMBTLNOXXVRCAXUIPREMMYMAHV

Perhitungan:

(K + L) mod 26 = (10 + 11) mod 26 = 21 = **V**

(R + A) mod 26 = (17 + 0) mod 26 = 17 = **R**

(I + M) mod 26 = (8 + 12) mod 26 = 20 = **U**

dst

Contoh 2: (dengan spasi)

P: she sells sea shells by the seashore

K: KEY KEYKE YKE YKEYKE YK EYK EYKEYKEY

C: CLC CIJVW QOE QRIJVW ZI XFO WCKWFYVC

Bentuk umum cipher abjad-majemuk:

- Kunci:

  $K = k_1 k_2 \ldots k_m$ (ket: m = panjang kunci)

- Plainteks:

  $P = p_1 p_2 \ldots p_m p_{m+1} \ldots p_{2m} \ldots$

- Cipherteks:

  $C = E_K(P) = f_{k1}(p_1)\, f_{k2}(p_2) \ldots f_{km}(p_m)\, f_{k1}(p_{m+1}) \ldots f_{km}(p_{2m}) \ldots$

- Untuk m = 1, *cipher*-nya ekivalen dengan *cipher* abjad-tunggal.

4. Cipher substitusi poligram

(*Polygram substitution cipher*)

- Blok huruf plainteks disubstitusi dengan blok cipherteks.
- Misalnya `AS` diganti dengan **`RT`**, `BY` diganti dengan **`SL`**

- Jika unit huruf plainteks/cipherteks panjangnya 2 huruf, maka ia disebut digram (*bigram*), jika 3 huruf disebut ternari-gram, dst
- Tujuannya: distribusi kemunculan poligram menjadi *flat* (datar), dan hal ini menyulitkan analisis frekuensi.

- Contoh: Playfair cipher (akan dijelaskan pada kuliah selanjutnya)

2. 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)

```
SDNIAIAONSSNLFITTOOIEEGRTMKIMB          (tanpa spasi)
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`

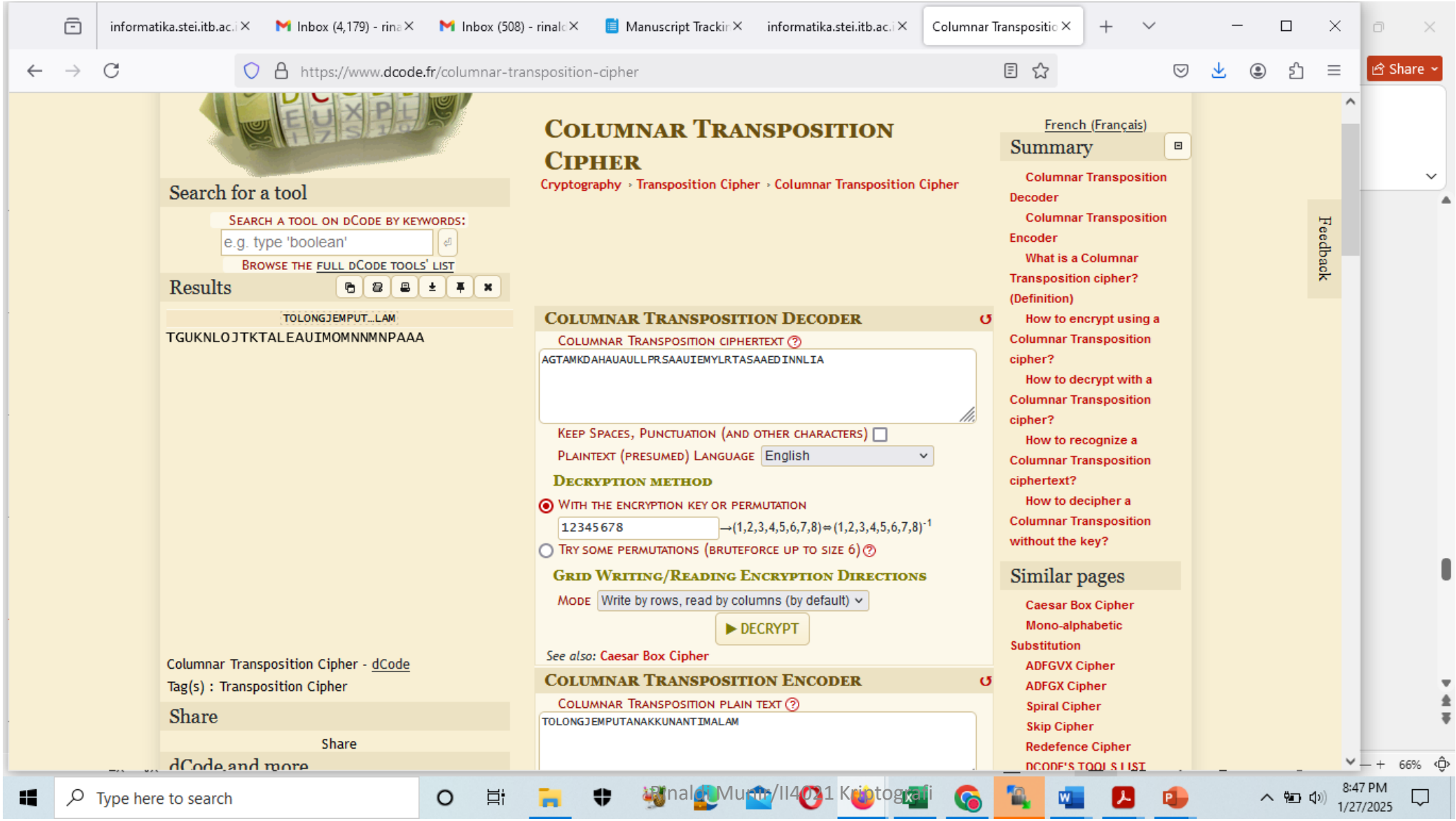
  Enkripsi:

```
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



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:

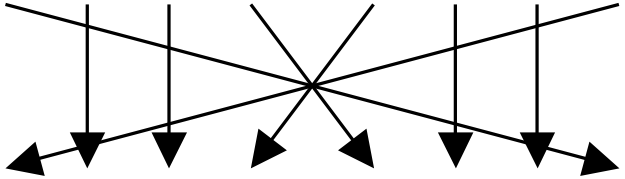
```
C       T       A       A       A       E       I
 R   P   O   R   P   Y   N   D   T   S   C   R   T
   Y       G       H       D       A       U       Y
```

maka cipherteksnya adalah

```
CTAAAEIRPORPYNDTSCRTYGHDAUY
```

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

1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
I	T	B	G	A	N	E	S	H	A	S	E	P	U	L	U	H	A	B	C	D	E	F	G



S	T	B	A	G	N	E	I	U	A	S	P	E	U	L	H	G	A	B	D	C	E	F	H
1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8

- 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`

  ✓ 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

Kriptanalisis pada *cipher* abjad-tunggal

- Review kembali, pada *cipher* abjad-tunggal satu huruf plainteks diganti dengan satu huruf cipherteks.

- *Caesar cipher* adalah salah satu *cipher* abjad-tunggal dengan melakukan substitusi huruf berdasarkan hasil pergeseran huruf-huruf alfabet sejauh *k* huruf. Namun *Caesar Cipher* bukan satu-satunya *cipher* abjad-tunggal.

- Secara umum, kita dapat membentuk tabel subtitusi sembarang. Jumlah kemungkinan tabel substitusi yang dapat dibuat pada sembarang *cipher* abjad-tunggal adalah sebanyak

  26! = 403.291.461.126.605.635.584.000.000

  karena ada 26! cara mempermutasikan 26 huruf alfabet.

- Tabel substitusi dapat dibentuk secara acak seperti contoh berikut:

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

- Atau berdasarkan kalimat kunci yang mudah diingat:

  Contoh: `di bawah sinar bulan purnama hati resah jadi senang`

  Buang duplikasi huruf menjadi: `dibawhsnrulpmtejg`

  Sambung dengan huruf lain yang belum ada:

  `dibawhsnrulpmtejgcfkoqvwxyz`

  Tabel substitusi yang dihasilkan:

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

- Sudah kita ketahui bahwa *cipher* abjad-tunggal (*monoalphabetic cipher*) memetakan sebuah huruf plainteks ke sebuah huruf cipherteks.

- Kelemahan *cipher* abjad-tunggal: tidak dapat menyembunyikan hubungan statistic antara plainteks dengan cipherteks.
  - Huruf yang sama dienkripsi menjadi huruf cipherteks yang sama
  - Huruf yang sering muncul di dalam plainteks, juga sering muncul di dalam huruf cipherteksnya yang berkoresponden .

- Oleh karena itu, cipherteks dapat didekripsi tanpa mengetahui kuncinya sekalipun dengan menggunakan Teknik kriptanalisis sederhana.

- Teknik yang digunakan untuk memecahkan *cipher* abjad-tunggal:
  1. teknik analisis frekuensi
  2. terkaan kata atau

- dengan asumsi kriptanalis mengetahui bahasa yang digunakan di dalam plainteks (misalnya Bahasa Inggris).

- Meskipun kriptanalis tidak mengetahui kunci yang digunakan di dalam proses enkripsi, namun dapat mencari tabel substitusi huruf plainteks menjadi huruf cipherteks dengan kombinasi kedua Teknik di atas.

Teknik Analisis Frekuensi

- Pada cipher abjad-tunggal, perulangan huruf di dalam plainteks tercermin pula pada perulangan huruf yang berkoresponden di dalam cipherteksnya.

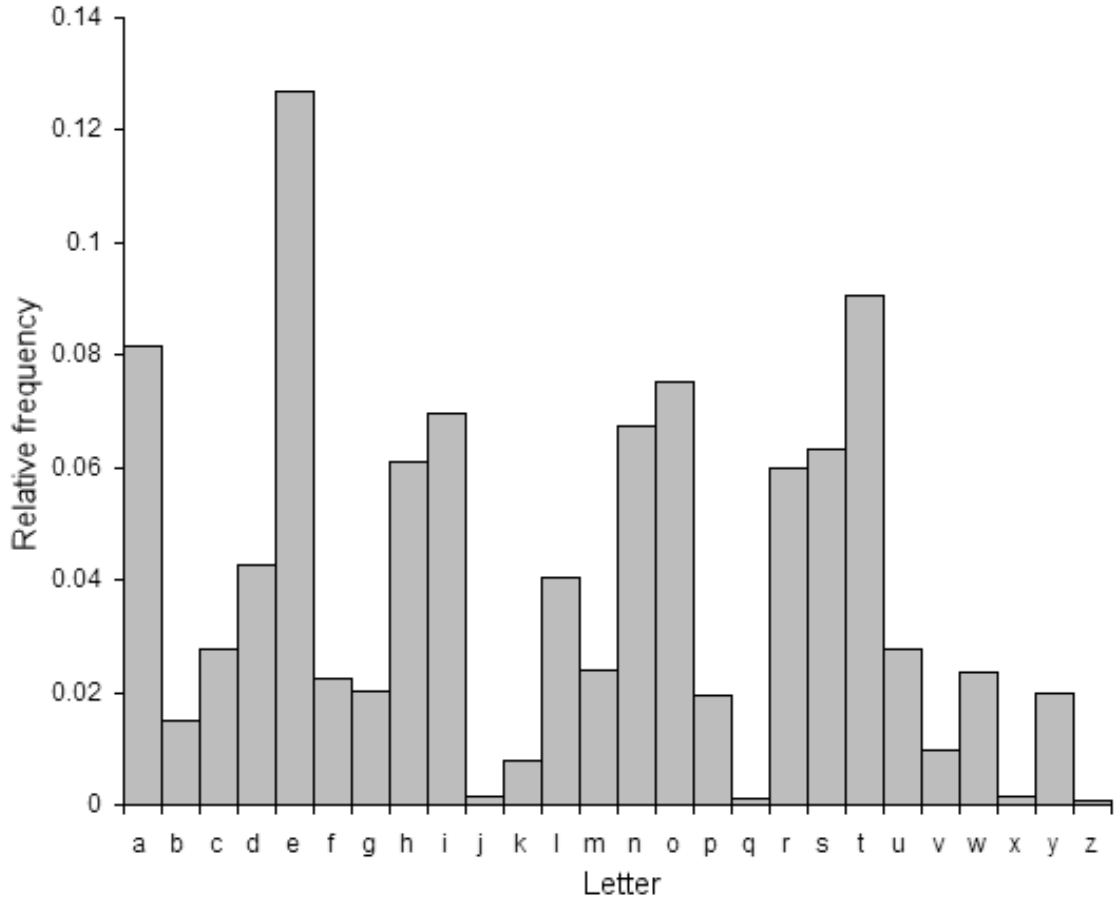
- Artinya, huruf yang sering muncul di dalam plainteks, maka huruf cipherteksnya juga sering muncul.

- Hubungan statistik antara huruf-huruf di dalam plainteks dengan huruf-huruf di dalam cipherteks menjadi peluang bagi kriptanalis untuk memecahkan cipherteks.

- Dengan memanfaatkan frekuensi kemunculan huruf, atau pasangan huruf (bigram), atau tiga huruf (trigram) di dalam suatu bahasa natural, kriptanalis dapat menemukan plainteks dengan mudah.

Tabel Frekuensi kemunculan (relatif) huruf-huruf dalam teks Bahasa Inggris (sampel mencapai 300.000 karakter di dalam sejumlah novel dan suratkabar

Huruf	%	Huruf	%
A	8,2	N	6,7
B	1,5	O	7,5
C	2,8	P	1,9
D	4,2	Q	0,1
E	12,7	R	6,0
F	2,2	S	6,3
G	2,0	T	9,0
H	6,1	U	2,8
I	7,0	V	1,0
J	0,1	W	2,4
K	0,8	X	2,0
L	4,0	Y	0,1
M	2,4	Z	0,1



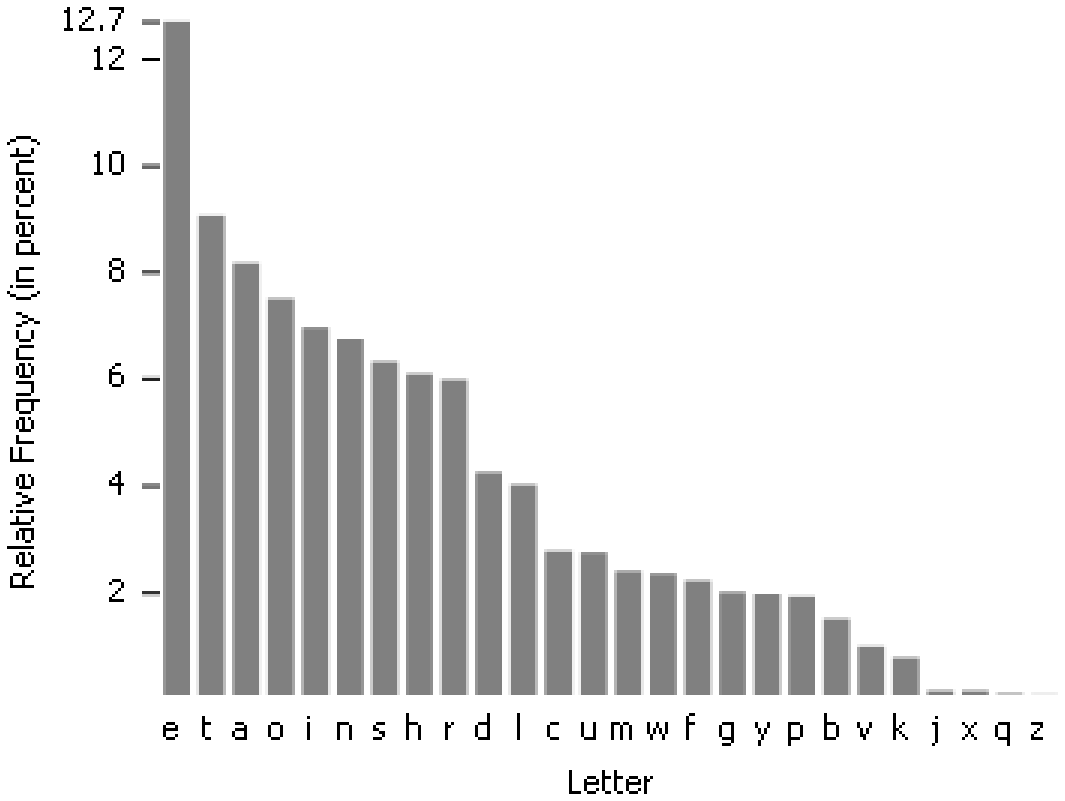
- *Top* 10 huruf yang sering muncul dalam teks Bahasa Inggris: E, T, A, O, I, N, S, H, R, D, L, U

- Top 10 huruf *bigram* yang sering muncul dalam teks B. Inggris: TH, HE, IN, EN, NT, RE, ER, AN, TI, dan ES

- Top 10 huruf *trigram* yang sering muncul dalam teks B. Inggris: THE, AND, THA, ENT, ING, ION, TIO, FOR, NDE, dan HAS

- Kriptanalis menggunakan tabel frekuensi kemunculan huruf dalam B. Inggris sebagai kakas bantu melakukan dekripsi.

- Misalnya, jika huruf "R" paling sering muncul di dalam cipherteks, maka kemungkinan besar itu adalah huruf "E" di dalam plainteksnya.



Bigram Frequencies

TH :	2.71	EN :	1.13	NG :	0.89
HE :	2.33	AT :	1.12	AL :	0.88
IN :	2.03	ED :	1.08	IT :	0.88
ER :	1.78	ND :	1.07	AS :	0.87
AN :	1.61	TO :	1.07	IS :	0.86
RE :	1.41	OR :	1.06	HA :	0.83
ES :	1.32	EA :	1.00	ET :	0.76
ON :	1.32	TI :	0.99	SE :	0.73
ST :	1.25	AR :	0.98	OU :	0.72
NT :	1.17	TE :	0.98	OF :	0.71

Trigram Frequencies

THE :	1.81	ERE :	0.31	HES :	0.24
AND :	0.73	TIO :	0.31	VER :	0.24
ING :	0.72	TER :	0.30	HIS :	0.24
ENT :	0.42	EST :	0.28	OFT :	0.22
ION :	0.42	ERS :	0.28	ITH :	0.21
HER :	0.36	ATI :	0.26	FTH :	0.21
FOR :	0.34	HAT :	0.26	STH :	0.21
THA :	0.33	ATE :	0.25	OTH :	0.21
NTH :	0.33	ALL :	0.25	RES :	0.21
INT :	0.32	ETH :	0.24	ONT :	0.20

- Perbandingan: top 10 huruf yang paling sering muncul dalam Bahasa Indonesia:

Huruf	Peluang (%)
A	17,50
N	10,30
I	8,70
E	7,50
K	5,65
T	5,10
R	4,60
D	4,50
S	4,50
M	4,50

Langkah-langkah kriptanalisis dengan teknik analisis frekuensi adalah sbb:

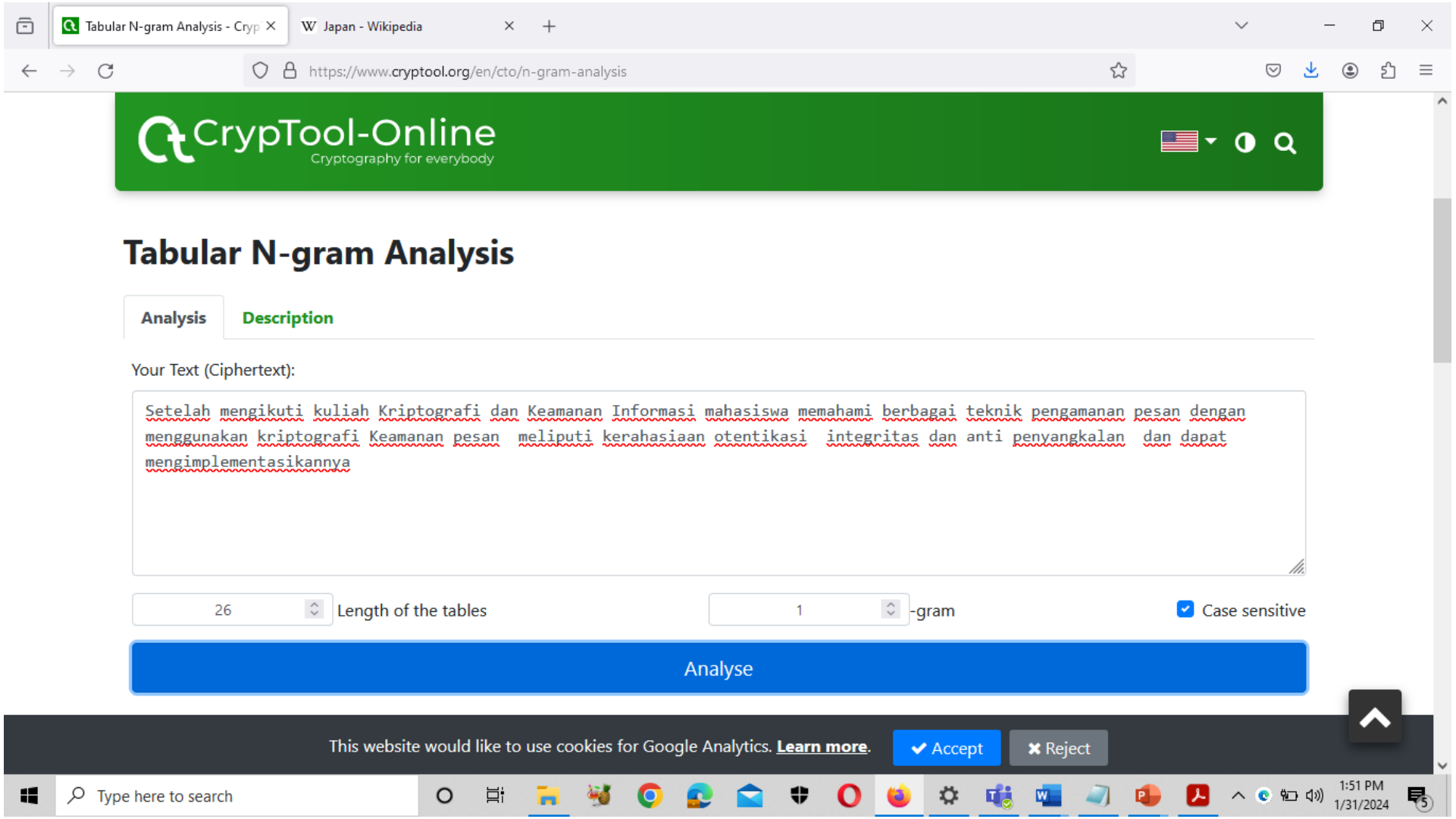
1. Hitung frekuensi kemunculan relatif huruf-huruf di dalam cipherteks.

2. Bandingkan hasil langkah 1 dengan tabel frekuensi kemunculan huruf, tabel kemunculan bigram, trigram, dsb. Mengingat huruf yang paling sering muncul dalam teks Bahasa Inggris adalah huruf E, maka huruf yang paling sering muncul di dalam cipherteks kemungkinan besar adalah huruf E di dalam plainteksnya.

3. Langkah 2 diulangi untuk huruf dengan frekeuensi terbanyak berikutnya. (biasanya hanya terpakai untuk 2 sampai 3 huruf pertama di dalam tabel frekuensi).

4. Ulangi langah 1 dan 2 dengan menggunakan bigram, trigram, dst, yang sering muncul.

- Kakas online untuk menghitung frekuensi kemunculan huruf, bigram, trigram dsb: https://www.cryptool.org/en/cto/n-gram-analysis



CrypTool-Online
Cryptography for everybody

N-gram tables

Rank	1-gram	Abs.	Rel.
1	a	44	19.298
2	n	31	13.596
3	i	22	9.649
4	e	21	9.211
5	m	14	6.140
6	t	13	5.702
7	g	11	4.825
8	k	10	4.386
9	p	9	3.947
10	s	9	3.947
11	r	8	3.509

- Contoh: Diberikan cipherteks berikut ini (Stalling, 2011), spasi tidak dibuang:

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
```

Kita akan makukakan kriptanalisis dengan metode analisis frekuensi untuk memperoleh plainteks.

Asumsi: bahasa yang digunakan adalah Bahasa Inggris dan *cipher* yang digunakan adalah *cipher* abjad-tunggal.

Hitung frekuensi kemunculan huruf di dalam cipherteks tersebut sbb:

Huruf	%	Huruf	%
P	13,33	Q	2,50
Z	11.67	T	2,50
S	8,33	A	1,67
U	8,33	B	1,67
O	7,50	G	1,67
M	6.67	Y	1,67
H	5,83	I	0,83
D	5,00	J	0,83
E	5,00	C	0,00
V	4,17	K	0,00
X	4,17	L	0,00
F	3,33	N	0,00
W	3,33	R	0,00

- Dua huruf yang paling sering muncul di dalam cipherteks: huruf P dan Z.
- Dua huruf yang paling sering muncul di dalam B. Inggris: huruf E dan T.
- Kemungkinan besar,
  - P adalah pemetaan dari e
  - Z adalah pemetaan dari t

- Tetapi kita belum dapat memastikannya sebab masih diperlukan cara *trial and error* dan pengetahuan tentang Bahasa Inggris.

- Tetapi ini adalah langkah awal yang bagus.

Iterasi 1:

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
 t           e   e te    t  t  e e             t

VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
   e t    t  t     e  ee    e   t      t

EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
 e  e e t t  e    t e   et
```

- ZWP dan ZWSZ dipetakan menjadi t*e dan t**t
- Kemungkinan besar W adalah pemetataan dari H sehingga kata yang mungkin untuk ZWP dan ZWSZ adalah the dan that

Iterasi 1:

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
 t           e   e te    t  t  e e              t

VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
   e t    t  t    e  ee    e   t     t

EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
 e  e e t t  e    t e   et
```

- `ZWP` dan `ZWSZ` dipetakan menjadi `t*e` dan `t**t`
- Kemungkinan besar `W` adalah pemetataan dari `H` sehingga kata yang mungkin untuk `ZWP` dan `ZWSZ` adalah `the` dan `that`

- Diperoleh pemetaan (cipherteks → plainteks):

  P → e
  Z → t
  W → h
  S → a

- **Iterasi 2:**

```
UZ QSO VUOHXMOPV GPOZPEVSG ZWSZ OPFPESX UDBMETSX AIZ
 t  a         e   e te  a  that  e e a        a    t

VUEPHZ HMDZSHZO WSFP APPD TSVP QUZW YMXUZUHSX
   e t    ta t  ha e  ee   a e   th     t  a

EPYEPOPDZSZUFPO MB ZWP FUPZ HMDJ UD TMOHMQ
 e  e e tat  e     the   et
```

- `WSFP` dipetakan menjadi `ha*e`.

- Dalam Bahasa Inggris, kata yang mungkin untuk `ha*e` hanyalah `have`, `hate`, `hale`, dan `haze`

- Dengan mencoba mengganti semua `F` di dalam cipherteks dengan `v`, `t`, `l`, dan `z`, maka huruf yang cocok adalah `v` sehingga `WSFP` dipetakan menjadi `have`

- Dengan mengganti `F` menjadi `v` pada kriptogram `EPYEPOPDZSZUFPO` sehingga menjadi `*e*e*e*tat*ve*`, maka kata yang cocok untuk ini adalah `representatives`

- Diperoleh pemetaan:

E → r Y → p

U → I O → s

D → n

- Hasil akhir bila diselesaikan seluruhnya:

It was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in Moscow

- Tabel substitusi yang dihasilkan:

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: **S A H V P B J W U - - X T D M Y Z E O Z I F Q – G -**

- Teknik analisis frekuensi tetap bisa dilakukan meskipun spasi dihilangkan.

- Contoh:

```
LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHKVSTYLXZIX
LIKIIXPIJVSZEYPERRGERIMWQLMGLMXQERIWGPSRIHMXQEREKIETXMJT
PRGEVEKEITREWHEXXLEXXMZITWAWSQWXSWEXTVEPMRXRSJGSTVRIEYVI
EXCVMUIMWERGMIWXMJMGCSMWXSJOMIQXLIVIQIVIXQSVSTWHKPEGARCS
XRWIEVSWIIBXVIZMXFSJXLIKEGAEWHEPSWYSWIWIEVXLISXLIVXLIRGE
PIRQIVIIBGIIHMWYPFLEVHEWHYPSRRFQMXLEPPXLIECCIEVEWGISJKTV
WMRLIHYSPHXLIQIMYLXSJXLIMWRIGXQEROIVFVIZEVAEKPIEWHXEAMWY
EPPXLMWYRMWXSGSWRMHIVEXMSWMGSTPHLEVHPFKPEZINTCMXIVJSVLMR
SCMWMSWVIRCIGXMWYMX
```

- Hasil perhitungan frekuensi kemunculan huruf, bigram, dan trigram:
  - huruf `I` paling sering muncul,
  - `XL` adalah bigram yang paling sering muncul,
  - `XLI` adalah trigram yang paling sering muncul.

Ketiga data terbanyak ini menghasilkan dugaan bahwa

`I` berkoresponden dengan huruf plainteks `e`,

`XLI` berkoresponden dengan `the`,

`XL` berkoresponden dengan `th`

Pemetaan:

`I` → `e`

`X` → `t`

`L` → `h`

- XLEX dipetakan menjadi th*t.
- Kata yang cocok untuk th*t. adalah that.
- Jadi kita memperoleh: E → a
- Hasil iterasi pertama:

```
heVeTCSWPeYVaWHaVSReQMthaYVaOeaWHRtatePFaMVaWHKVSTYhtZe
theKeetPeJVSZaYPaRRGaReMWQhMGhMtQaReWGPSReHMtQaRaKeaTtM
JTPRGaVaKaeTRaWHatthattMZeTWAWSQWtSWatTVaPMRtRSJGSTVRea
YVeatCVMUeMWaRGMeWtMJMGCSMWtSJOMeQtheVeQeVetQSVSTWHKPaG
ARCStRWeaVSWeeBtVeZMtFSJtheKaGAaWHaPSWYSWeWeaVtheStheVt
heRGaPeRQeVeeBGeeHMWYPFhaVHaWHYPSRRFQMthaPPtheaCCeaVaWG
eSJKTVWMRheHYSPHtheQeMYhtSJtheMWReGtQaROeVFVeZaVAaKPeaW
HtaAMWYaPPthMWYRMWtSGSWRMHeVatMSWMGSTPHhaVHPFKPaZeNTCMt
eVJSVhMRSCMWMSWVeRCeGtMWYMt
```

- Selanjutnya,

  `Rtate` mungkin adalah `state`,

  `atthattMZE` mungkin adalah `atthattime`,

  `heVe` mungkin adalah `here`.

- Jadi, kita memperoleh pemetaan baru:

  `R` → `s`

  `M` → `i`

  `Z` → `m`

  `V` → `r`

- Hasil iterasi ke-2:

```
hereTCSWPeYraWHarSseQithaYraOeaWHstatePFairaWHKrSTYhtm
etheKeetPeJrSmaYPassGaseiWQhiGhitQaseWGPSseHitQasaKeaT
tiJTPsGaraKaeTsaWHatthattimeTWAWSQWtSWatTraPistsSJGSTr
seaYreatCriUeiWasGieWtiJiGCSiWtSJOieQthereQeretQSrSTWH
KPaGAsCStsWearSWeeBtremitFSJtheKaGAaWHaPSWYSWeWeartheS
therthesGaPesQereeBGeeHiWYPFharHaWHYPSssFQithaPPtheaCC
earaWGeSJKTrWisheHYSPHtheQeiYhtSJtheiWseGtQasOerFremar
AaKPeaWHtaAiWYaPPthiWYsiWtSGSWsiHeratiSWiGSTPHharHPFKP
ameNTCiterJSrhisSCiWiSWresCeGtiWYit
```

- Teruskan, dengan menerka kata-kata yang sudah dikenal, misalnya `remarA` mungkin `remark`, dsb

- Hasil iterasi 3:

```
hereuponlegrandarosewithagraveandstatelyairandbr
oughtmethebeetlefromaglasscaseinwhichitwasenclos
editwasabeautifulscarabaeusandatthattimeunknownt
onaturalistsofcourseagreatprizeinascientificpoin
tofviewthereweretworoundblackspotsnearoneextremi
tyofthebackandalongoneneartheotherthescalesweree
xceedinglyhardandglossywithalltheappearanceofbur
nishedgoldtheweightoftheinsectwasveryremarkablea
ndtakingallthingsintoconsiderationicouldhardlybl
amejupiterfor*hisopinionrespectingit
```

- Tambahkan spasi, tanda baca, dll

```
Here upon Legrand arose, with a grave and stately
air, and brought me the beetle from a glass case in
which it was enclosed. It was a beautiful
scarabaeus, and, at that time, unknown to
naturalists—of course a great prize in a scientific
point of view. There were two round black spots
near one extremity of the back, and a long one near
the other. The scales were exceedingly hard and
glossy, with all the appearance of burnished gold.
The weight of the insect was very remarkable, and,
taking all things into consideration, I could
hardly blame Jupiter for his opinion respecting it.
```

Bersambung