Bahan kuliah II4021 Kriptografi

05 – Kriptanalisis Cipher Klasik

Oleh: Rinaldi M

Program Studi Sistem dan Teknollogi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2026

Coincidence Index (CI)

- *Coincidence Index* (CI) atau lebih dikenal sebagai *Index of Coincidence* (IoC) adalah ukuran statistik yang menyatakan probabilitas bahwa dua huruf yang dipilih secara acak dari suatu teks adalah huruf yang sama.

- Konsep ini diperkenalkan oleh kriptolog Amerika, William F. Friedman, dan sangat penting dalam kriptanalisis cipher klasik.

- Rumus CI:

$$CI = \frac{\sum_{i=1}^{26} f_i\,(f_i - 1)}{N(N-1)}$$

  - N= jumlah total huruf dalam teks
  - f_i= frekuensi huruf ke-i (misalnya A, B, C, …)

Maknanya:
- Pembilang menghitung jumlah pasangan huruf yang sama
- Penyebut menghitung jumlah seluruh pasangan huruf yang mungkin

Contoh: Misalkan kita punya cipherteks: ABABAC

Frekuensi huruf: A = 3, B = 2, C = 1, total huruf $N = 6$

Hitung pembilang: $3(2) + 2(1) + 1(0) = 6 + 2 + 0 = 8$

Hitung penyebut: $6(5) = 30$

Maka: $CI = \frac{8}{30} = 0.2667$, nilai ini tinggi karena huruf A dan B sering berulang.

Nilai CI dalam Bahasa Alami

Untuk teks panjang:

- Bahasa Inggris *CI* ≈ 0.065
- Teks acak murni *CI* ≈ 0.038
- Bahasa Indonesia biasanya *CI* juga mendekati 0.06 – 0.07

Semakin alami suatu bahasa → distribusi huruf tidak merata → CI lebih besar.

Semakin acak huruf di dalam pesan → distribusi huruf merata → CI ≈ 1/26 ≈ 0.038.

- Mengapa CI efektif untuk kriptanalisis? Karena *cipher* substitusi klasik:
  - ✓ tidak mengubah distribusi frekuensi huruf di dalam cipherteks
  - ✓ hanya memetakan huruf ke huruf lain
  - ✓ Jadi struktur statistik bahasa masih terlihat di dalam cipherteks.

- Ini sangat berbeda dengan *cipher* modern seperti *Advanced Encryption Standard* (AES) yang menghasilkan distribusi karakter di dalam cipherteks mendekati acak sehingga CI ≈ 0.038.

- Pada teks (English) normal → banyak perulangan huruf E, T, A → CI tinggi
- Pada teks benar-benar acak → semua huruf hampir sama peluangnya → CI rendah

- Jadi, CI adalah cara matematis untuk mengukur “seberapa alami” distribusi huruf dalam suatu teks.

- CI sangat penting untuk menyerang *cipher* substitusi klasik, terutama untuk membedakan jenis cipher
  - Jika CI ≈ 0.065 → kemungkinan *monoalphabetic cipher*, contoh: Caesar Cipher
  - Jika CI ≈ 0.038 → kemungkinan *polyalphabetic cipher*, contoh: Vigenere Cipher
  - Untuk *Playfairc cipher*, IC ≈ 0,065 (mendekati plainteks bahasa Inggris), biasanya berada di kisaran 0.060 – 0.068

Cipher	Perkiraan IC
Teks Inggris biasa	≈ 0,065
Caesar	≈ 0,065
Monoalphabetic	≈ 0,065
Playfair	≈ 0,060–0,068
Vigenère (panjang kunci besar)	≈ 0,038
Teks acak	≈ 0,038

Contoh penggunaan CI untuk mengklasifikasikan Cipher

Diberikan ciphertext: LXFOPVEFRNHR

Ciphertext ini sebenarnya adalah enkripsi dengan Vigenere Cipher dari plainteks: ATTACKATDAWN dengan kunci: LEMON.

Hitung frekuensi huruf: F = 2, R = 2, huruf lainnya = 1. Total huruf (N) = 12

Hitung CI:

$$CI = \frac{\sum f_i(f_i-1)}{N(N-1)} = \frac{2(1)+2(1)+1(0)+1(0)\ldots+1(0)}{12(11)} = \frac{4}{132} = 0.0303$$

Nilai CI mendekati 0.038 → distribusi hampir acak → kemungkinan *polyalphabetic cipher.*

Asumsikan kita tidak mengetahui kuncinya. Hitung estimasi panjang kunci, coba k = 1, k = 2, k = 3, k = 4, k = 5, dst. Lebih lanjut dibahas nanti pada metode Kasiski

A. Kriptanalisis *Cipher* Abjad-Tunggal

- Review kembali, pada *cipher* abjad-tunggal (*monoalpabhetic cipher*), 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.
  - artinya struktur statistik bahasa masih muncul di dalam cipherteks

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

- Teknik yang digunakan untuk memecahkan *cipher* abjad-tunggal adalah kombinasi teknik analisis frekuensi dan menerka kata (berdasarkan kamus bahasa)

- 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 kriptanalis dapat mencari tabel substitusi huruf plainteks menjadi huruf cipherteks dengan menggunakan 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.

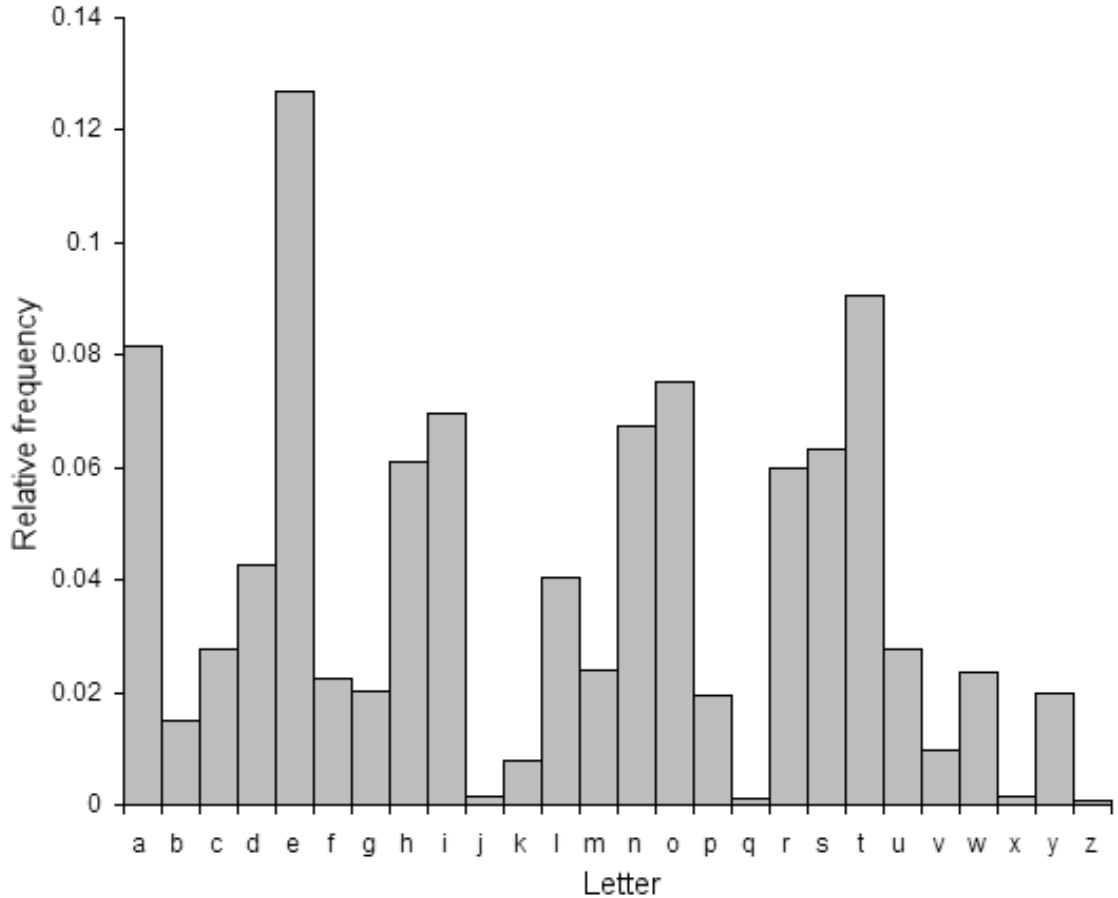
- Artinya, huruf yang sering muncul di dalam plainteks, maka huruf cipherteksnya juga sering muncul.

- 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



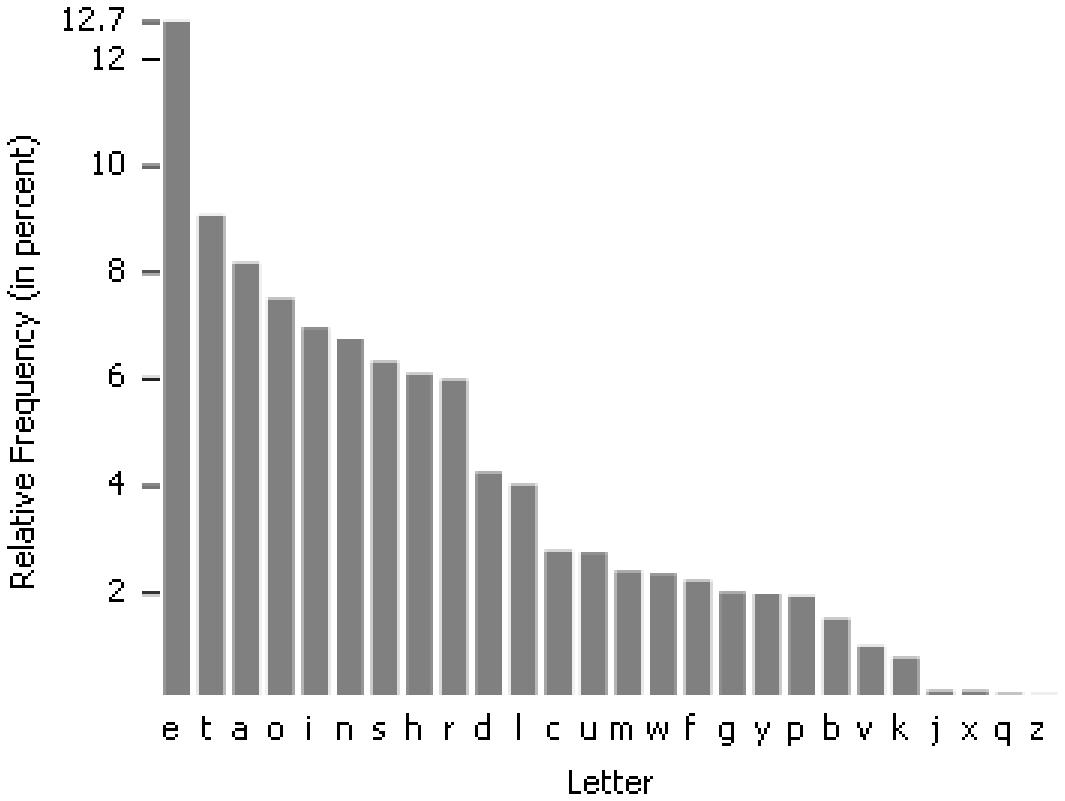
- *Top* 10 huruf yang sering muncul dalam teks Bahasa Inggris: E, T, A, O, I, N, S, H, R, D, L, U

- 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:

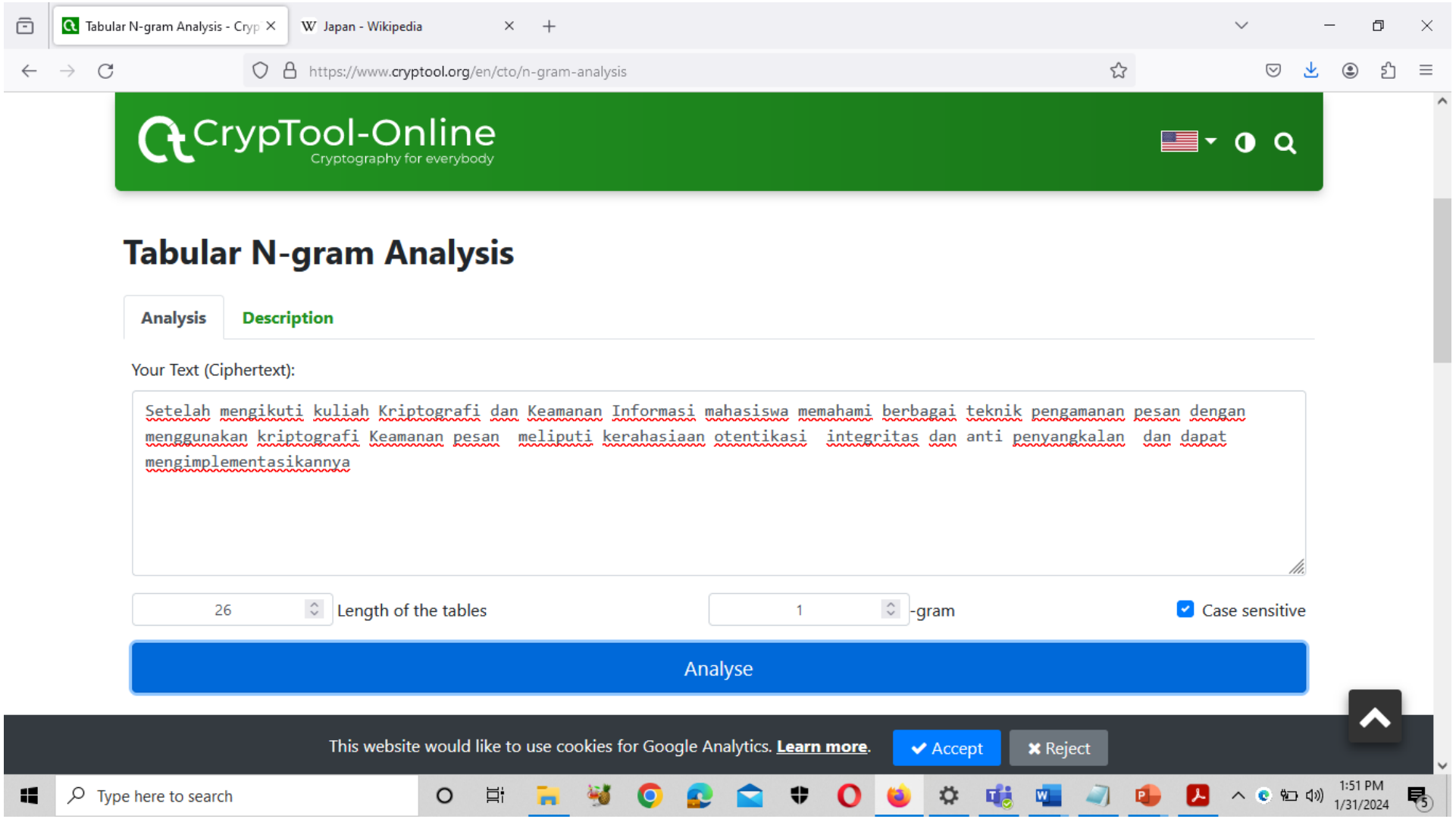
1. Hitung frekuensi kemunculan relatif huruf-huruf di dalam cipherteks.

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



Tabular N-gram Analysis - Cryp
Japan - Wikipedia
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

This website would like to use cookies for Google Analytics. Learn more.
Accept
Reject
Type here to search
1:51 PM
1/31/2024

Contoh 1: 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 malakukan 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:

Rank	1-gram	Abs.	Rel.
1	P	16	13.333
2	Z	14	11.667
3	U	10	8.333
4	S	10	8.333
5	O	9	7.500
6	M	8	6.667
7	H	7	5.833
8	E	6	5.000
9	D	6	5.000
10	V	5	4.167
11	X	5	4.167

Rank	1-gram	Abs.	Rel.
12	W	4	3.333
13	F	4	3.333
14	Q	3	2.500
15	T	3	2.500
16	G	2	1.667
17	B	2	1.667
18	A	2	1.667
19	Y	2	1.667
20	I	1	0.833
21	J	1	0.833

Jumlah total huruf, N = 120

$$CI = \frac{\sum_{i=1}^{21} f_i(f_i-1)}{N(N-1)} = \frac{916}{120(119)} = 0{,}0641$$

Nilai CI sangat dekat dengan 0,065 (teks bahasa Inggris), jauh dari 0,038 (teks acak/polyalphabetic cipher), maka kemungkinan besar ini adalah cipherteks dari *monoalphabetic cipher*

Rank	1-gram	Abs.	Rel.
1	P	16	13.333
2	Z	14	11.667
3	U	10	8.333
4	S	10	8.333
5	O	9	7.500
6	M	8	6.667
7	H	7	5.833
8	E	6	5.000
9	D	6	5.000
10	V	5	4.167
11	X	5	4.167

- 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 -**

Contoh 2: Diberikan cipherteks berikut ini, yang dihasilkan dari cipher abjad-Tunggal. Lakukan dekripsi untuk mengungkap plainteksnya.

```
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.

B. Kriptanalisis Vigenere Cipher

- Friedrich Kasiski adalah orang yang pertama kali memecahkan *Vigènere cipher* pada Tahun 1863.

> Friedrich Kasiski
> Born: November 29, 1805 @ Schlochau, Kingdom of Prussia
> Died: May 22, 1881 (aged 75) @ Neustettin, German Empire
> Nationality: German

- Metodenya dinamakan metode Kasiski

Metode Kasiski

- Metode Kasiski tidak secara langsung menemukan kunci Vigenere Cipher, tetapi membantu menemukan estimasi panjang kunci *Vigenere cipher*.

- Metode Kasiski memanfaatkan keuntungan bahwa bahasa Inggris tidak hanya mengandung perulangan huruf,

- tetapi juga perulangan pasangan huruf atau tripel huruf, seperti `TH`, `THE`, `EN`, dsb.

- Perulangan kelompok huruf ini ada kemungkinan menghasilkan kriptogram yang berulang.

Contoh 3:

Plainteks : **crypto**isshortfor**crypto**graphy

Kunci : abcdabcdabcdabcdabcdabcdabcdabcd

Cipherteks : **CSASTP**KVSIQUTGQU**CSASTP**IUAQJB

- Pada contoh ini, crypto dienkripsi menjadi kriptogram yang sama, yaitu **CSATP**.

- Tetapi kasus seperti ini tidak selalu demikian, misalnya pada contoh berikut ini….

Contoh 4:

Plainteks : **crypto**isshortfor**crypto**graphy

Kunci : abcdefabcdefabcdefabcdefabcd

Cipherteks : **CSASXT**ITUKWSTGQU**CWYQVR**KWAQJB

- Pada contoh di atas, crypto tidak dienkripsi menjadi kriptogram yang sama.

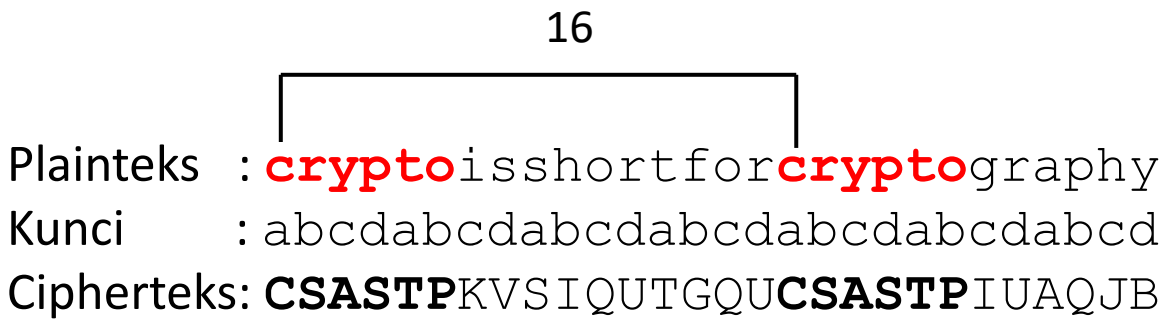
- Mengapa bisa demikian?

- Secara intuitif: jika jarak antara dua buah *string* yang berulang di dalam plainteks merupakan kelipatan dari panjang kunci,

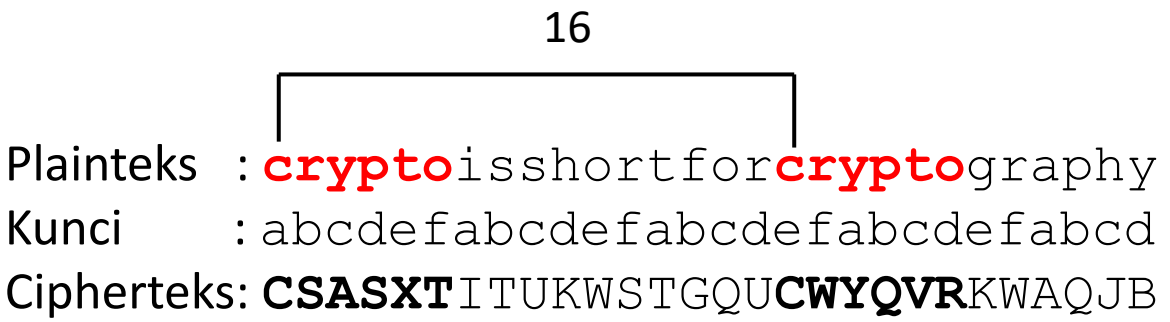
- maka *string* yang sama tersebut akan muncul menjadi kriptogram yang sama pula di dalam cipherteks.

- Pada Contoh 1,

  - kunci = `abcd`
  - panjang kunci = 4
  - jarak antara dua `crypto` yang berulang = 16
  - 16 = kelipatan 4



∴ `crypto` dienkripsi menjadi kriptogram yang sama



- Pada Contoh 2,
  - kunci = `abcdef`
  - panjang kunci = 6
  - jarak antara dua `crypto` yang berulang = 16
  - 16 bukan kelipatan 6

∴ `crypto` tidak dienkripsi menjadi kriptogram yang sama

- Goal metode Kasiski: mencari dua atau lebih kriptogram yang berulang untuk menentukan panjang kunci.

Langkah-langkah metode Kasiski:

1. Temukan semua kriptogram yang berulang di dalam cipherteks (pesan yang panjang biasanya mengandung kriptogram yang berulang).
2. Hitung jarak antara kriptogram yang berulang
3. Hitung semua faktor (pembagi) dari jarak tersebut (faktor pembagi menyatakan panjang kunci yang mungkin).
4. Tentukan irisan dari himpunan faktor pembagi tersebut. Nilai yang muncul di dalam irisan menyatakan angka yang muncul pada semua faktor pembagi dari jarak-jarak tersebut . Nilai tersebut mungkin adalah panjang kunci.

- Contoh:

  DYDUXRMHTVDV**NQD**QNW**DYDUXRMH**ARTJGW**NQD**

  Kriptogram yang berulang: **DYUDUXRMH** dan **NQD**.

  Jarak antara dua buah perulangan **DYUDUXRMH** = 18.

  Semua faktor pembagi 18 : {18, 9, 6, 3, 2}

  Jarak antara dua buah perulangan **NQD** =20.

  Semua faktor pembagi 20 : {20, 10, 5, 4, 2}.

  Irisan dari kedua buah himpunan tersebut adalah 2

  Panjang kunci kemungkinan besar adalah 2.

- Setelah panjang kunci diketahui, maka langkah berikutnya menentukan huruf-huruf kunci

- Huruf-huruf kunci dapat ditentukan dengan menggunakan *exhaustive key search*

- Jika panjang kunci = *p*, maka jumlah kunci yang harus dicoba sampai menemukan kunci yang benar adalah adalah maksimal 26^p kali.

- Namun lebih sangkil menemukan huruf-huruf kunci dengan menggunakan teknik analisis frekuensi.

Langkah-langkahnya sbb:

1. Misalkan panjang kunci yang sudah berhasil dideduksi adalah *n.* Setiap huruf pada modulus 5 pasti dienkripsi dengan huruf kunci yang sama. Kelompokkan setiap huruf posisi modulus 5 bersama-sama sehingga kriptanalis memiliki n buah “pesan”, masing-masing “pesan” dienkripsi dengan huruf kunci yang sama dengan *cipher* abjad-tunggal (dalam hal ini *Caesar cipher*).

2. Tiap-tiap pesan dari hasil langkah 1 dapat dipecahkan dengan metode analisis frekuensi.

3. Dari hasil langkah 2 kriptanalis dapat menyusun huruf-huruf kunci. Atau, kriptanalis dapat menerka kata yang membantu untuk memecahkan cipherteks

- Contoh:

1 2 3 4 5

LJVBQSTNEZLQMED**LJV**MAMPKAUFAVAT**LJV**DAYYVNFJQLNP**LJV**HKVTRNF**LJV**CMLKETA
LJVHUYJVSFKRFTTWEFUXVHZNP

6

Kriptogram yang berulang adalah **LJV**

Jarak **LJV** ke-1 dengan **LJV** ke-2 = 15

Jarak **LJV** ke-2 dengan **LJV** ke-3 = 15

Jarak **LJV** ke-3 dengan **LJV** ke-4 = 15

Jarak **LJV** ke-4 dengan **LJV** ke-5 = 10

Jarak **LJV** ke-5 dengan **LJV** ke-6 = 10

Faktor pembagi 15 = {3, 5, 15}

Faktor pembagi 10 = {2, 5, 10}

Irisan kedua himpunan ini = 5. Jadi, panjang kunci diperkirakan = 5

- Kelompokkan “pesan” setiap modulus 5, dimulai dari huruf cipherteks pertama, kedua, dan seterusnya. Setiap huruf pada posisi modulus 5 pasti dienkripsi dengan huruf kunci yang sama.

123456789$_{10}$11…

LJVBQSTNEZLQMEDLJVMAMPKAUFAVATLJVDAYYVNFJQLNPLJVHKVTRNFLJVCM
LKETALJVHUYJVSFKRFTTWEFUXVHZNP

Kelompok	Pesan	Huruf paling sering muncul
1	LSLLMFLYJLVLLLYKWV	L
2	JTQJP AJYQJTJKJJREH	J
3	VNMVKVVVLVRVEVVFFZ	V
4	BEEMAADNNHNCTHSTUN	N
5	QZDAUTAFPKFMAUFTXP	A

- Dalam Bahasa Inggris, 10 huruf yang yang paling sering muncul adalah E, T, A, O, I, N, S, H, R, dan D,

- Triplet yang paling sering muncul adalah THE. Karena **LJV** paling sering muncul di dalam cipherteks, maka dari 10 huruf tsb semua kemungkinan kata 3-huruf dibentuk dan kata yang yang cocok untuk **LJV** adalah THE.

- Jadi, kita dapat menerka bahwa **LJV** mungkin adalah THE.

- Huruf-huruf kunci lainnya dicoba dengan menerka dan menguji coba.

- Dari sini kita buat tabel yang memetakan huruf plainteks dengan cipherteks dan huruf-huruf kuncinya (ingatlah bahwa setiap nilai numerik dari huruf kunci menyatakan jumlah pergeseran huruf pada *Caesar cipher*):

Kelompok	Huruf plainteks	Huruf cipherteks	Huruf kunci
1	T	L	S (=18)
2	H	J	C (=2)
3	E	V	R (=17)
4	N	N	A (=0)
5	O	A	M (=12)

Jadi, kuncinya adalah SCRAM

- Dengan menggunakan kunci SCRAM cipherteks berhasil didekripsi menjadi:

```
THEBE ARWEN TOVER THEMO UNTAI NYEAH
THEDO GWENT ROUND THEHY DRANT THECA
TINTO THEHI GHEST SPOTH ECOUL DFIND
```

- Lakukan post-processing untuk menghasilkan kalimat yang lebih jelas:

```
THE BEAR WENT OVER THE MOUNTAIN YEAH
THE DOG WENT ROUND THE HYDRANT
THE CAT INTO THE HIGHEST SPOT HE COULD FIND
```

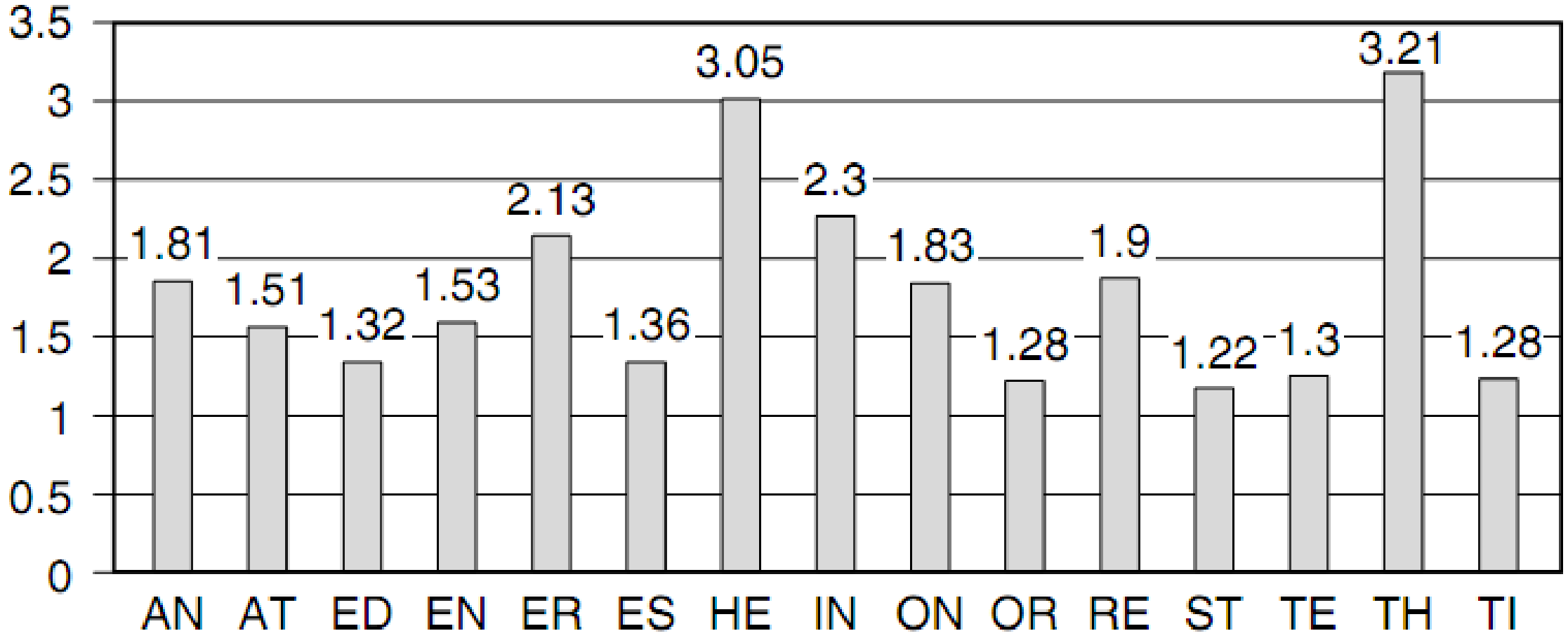
Kriptanalisis Playfair Cipher

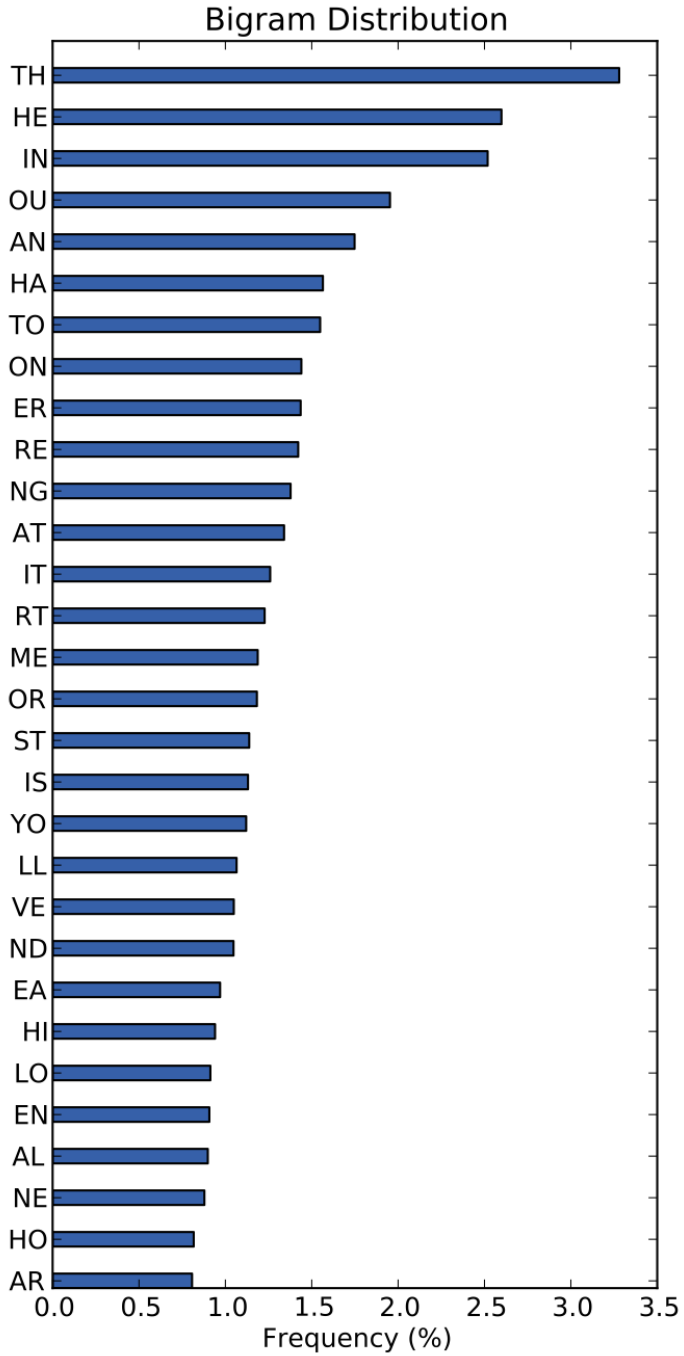
- Karena ada 26 huruf abjad, maka terdapat 26 x 26 = 677 bigram, sehingga identifikasi bigram individual lebih sukar.

- Sayangnya ukuran poligram di dalam *Playfair cipher* tidak cukup besar, hanya dua huruf sehingga *Playfair cipher* tetap tidak aman.

- Meskipun *Playfair cipher* sulit dipecahkan dengan analisis frekuensi relatif huruf-huruf, namun ia dapat dipecahkan dengan analisis frekuensi pasangan huruf.

- Dalam Bahasa Inggris kita bisa mempunyai frekuensi kemunculan pasangan huruf, misalnya pasangan huruf TH dan HE paling sering muncul.


3.5
3
2.5
2
1.5
1
0.5
0
1.81
1.51
1.32
1.53
2.13
1.36
3.05
2.3
1.83
1.28
1.9
1.22
1.3
3.21
1.28
AN
AT
ED
EN
ER
ES
HE
IN
ON
OR
RE
ST
TE
TH
TI



- Dengan menggunakan tabel frekuensi kemunculan pasangan huruf di dalam Bahasa Inggris dan cipherteks yang cukup banyak, *Playfair cipher* dapat dipecahkan.

- Kelemahan lainnya, bigram dan kebalikannya (misal AB dan BA) akan didekripsi menjadi pola huruf plainteks yang sama (misal RE dan ER). Di dalam bahasa Inggris terdapat banyak kata yang mengandung bigram terbalik seperti REceivER dan DEpartED.

- Contoh:

  - misalkan bigram yang sering muncul di dalam cipherteks adalah BM.

  - bigram yang sering muncul di dalam b ahasa Inggris adalah TH

  - jadi, kita duga TH dipetakan menjadi BM

  - misalkan TH dienkripsi dengan aturan segiempat, maka jika T dan H membentuk segiempat dengan B dan M, maka posisi mereka harus konsisten dalam matriks.

  Artinya: T satu baris dengan B, dan H satu baris dengan M

  T -------- **B**

  | |

  M-------- H

  - Jika TH dienkripsi dengan aturan kolom, maka B di bawah T, M di bawah H

  - Jika TH dienkripsi dengan aturan baris, maka B di kanan T, M di kanan H

  - Mulai bangun matriks parsial 5 x 5 dengan hipotesis seperti di atas

Kriptanalisis Affine Cipher

- *Affine Cipher:*

  Enkripsi: $C \equiv mP + b \pmod{n}$

  Dekripsi: $P \equiv m^{-1}(C - b) \pmod{n}$

- *Affine cipher* mudah diserang dengan *known-plaintext attack* (sebagian plainteks dari pesan diketahui)

- Misalkan kriptanalis mempunyai dua buah plainteks, P_1 dan P_2, yang berkoresponden dengan cipherteks C_1 dan C_2,

- maka m dan b mudah dihitung dari buah kekongruenan simultan berikut ini:

  $C_1 \equiv mP_1 + b \pmod{n}$

  $C_2 \equiv mP_2 + b \pmod{n}$

- Contoh: Misalkan kriptanalis menemukan

  cipherteks **C** dan plainteks berkorepsonden K

  cipherteks **E** dan plainteks berkoresponden O.

- Kriptanalis m dan n dari kekongruenan berikut:

  $2 \equiv 10m + b \pmod{26}$ (i)

  $4 \equiv 14m + b \pmod{26}$ (ii)

- Kurangkan (ii) dengan (i), menghasilkan

  $2 \equiv 4m \pmod{26}$ (iii)

  Solusi: $m = 7$

  Substitusi $m = 7$ ke dalam (i),

  $2 \equiv 70 + b \pmod{26}$ (iv)

  Solusi: $b = 10$.

Kriptanalisis Hill Cipher

Enkripsi: C = KP mod 26
Dekripsi: P = K^{-1}C mod 26

- *Hill cipher* mudah dipecahkan dengan *known-plaintext attack*.
- Misalkan untuk *Hill cipher* dengan *m* = 2 diketahui:
  - P = (19, 7) → *C* = (0, 23)
  - P = (4, 17) → *C* = (12, 6)
  - Jadi, **K**(19, 7) = (0, 23) dan **K**(4, 17) = (12, 6)

$$\begin{pmatrix} 0 & 12 \\ 23 & 6 \end{pmatrix} = K \begin{pmatrix} 19 & 4 \\ 7 & 17 \end{pmatrix} \bmod 26 \qquad \rightarrow \mathbf{K} = \mathbf{CP}^{-1} \bmod 26$$

$\leftarrow C \rightarrow \qquad \leftarrow P \rightarrow$

- Matriks balikan dari *P* adalah P^{-1} = $\begin{pmatrix} 19 & 4 \\ 7 & 17 \end{pmatrix}^{-1} \bmod 26 = \begin{pmatrix} 25 & 14 \\ 5 & 5 \end{pmatrix}$
- Sehingga

$$\mathbf{K} = \mathbf{CP}^{-1} \bmod 26 = \begin{pmatrix} 0 & 12 \\ 23 & 6 \end{pmatrix} \begin{pmatrix} 25 & 14 \\ 5 & 5 \end{pmatrix} \bmod 26 = \begin{pmatrix} 8 & 8 \\ 7 & 14 \end{pmatrix}$$