Data Acquisition, representation, and storage

Pertemuan 3
MK Data Mining II

M. N. Fakhruzzaman, S.Kom., M.Sc.
Ratih Ardiati Ningrung, S.Si., M.S., M.Stat.
Malikhah, S.Kom., M.Kom.

Program Studi S1 Teknologi Sains Data
Fakultas Teknologi Maju dan Multidisiplin
Universitas Airlangga Indonesia

Introduction

- Data acquisition adalah proses mengubah bentuk fisik objek menjadi bentuk yang dapat diproses secara digital

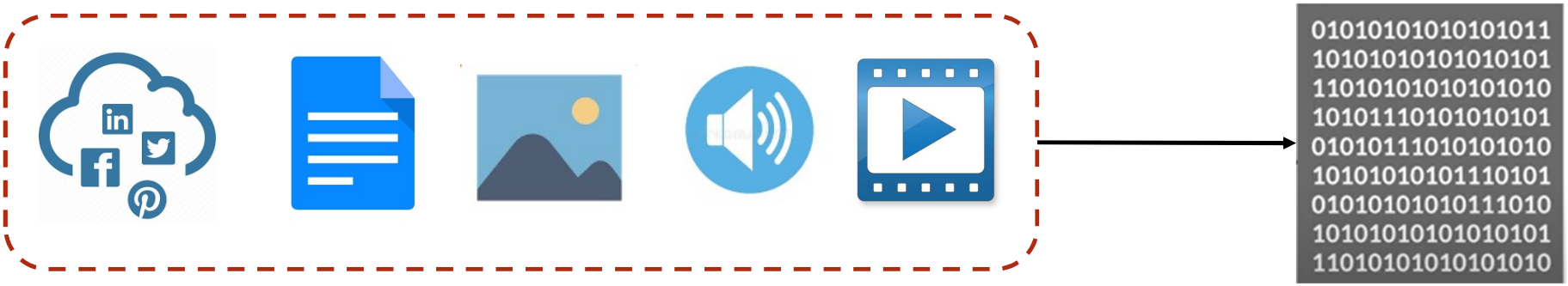
- Data representation merupakan bagaimana mengekstrak/mendapatkan informasi dari data agar bisa dimengerti oleh komputer

- Storage adalah bagaimana menyimpan sebanyak mungkin informasi dari data

How computers represent data

- Blok bangunan dasar di computer adalah system bilangan biner
- Bilangan biner adalah system bilangan yang hanya memiliki 2 nilai: **0** dan **1**
- Nilai 0 dan 1 bisa disebut dengan **low** dan **high** atau **off** dan **on** atau bisa diibartkan seperti switch.
- Suatu switch yang sedang ON bisa diibaratkan memiliki nilai 1, sedangkan yang sedang OFF diibartkan memiliki nilai 0.
- Central Processing Unit (CPU) adalah bagian dari computer yang membuat computer bisa berinteraksi dengan aplikasi dan program yang diinstall di computer. CPU mengartikan instrukti program dan menghasilkan output.

Sistem Bilangan Biner



- Semua informasi di sistem computer harus diubah menjadi data yang bisa dibaca oleh computer, atau diubah ke bilangan biner, sehingga komputer bisa mengerti dan memproses data tersebut.
- Sistem bilangan biner merupakan salah satu system bilangan matematika, yang nilainya hanya 0 dan 1.

  Contoh : 10011010 → bilangan biner

  2,3,4,5,6,7,8,9, dst → bukan bilangan biner

BIT and Byte

- **BIT** (binary digit): unit dasar informasi untuk computer
- 1 **Byte** = 8 bit

# of bits	# of different binary numbers
1 bit:	$2^1 = 2$
2 bits:	$2^2 = 4$
3 bits:	$2^3 = 8$
4 bits:	$2^4 = 16$
5 bits:	$2^5 = 32$
6 bits:	$2^6 = 64$
7 bits:	$2^7 = 128$
8 bits:	$2^8 = 256$
9 bits:	$2^9 = 512$
10 bits:	$2^{10} = 1024$
etc.	

Tugas kelas

- Ubah bilangan biner berikut menjadi bilangan decimal (berbasis 10):
  1. 1011
  2. 10100
  3. 10001001
  4. 11001101
- Ubah bilangan decimal berikut menjadi bilangan biner:
  1. 190
  2. 235
  3. 68

Biner ke Desimal

1 0 1 1

Cara:

1= 2^3 = 8

0= 2^2 = 4 (x)

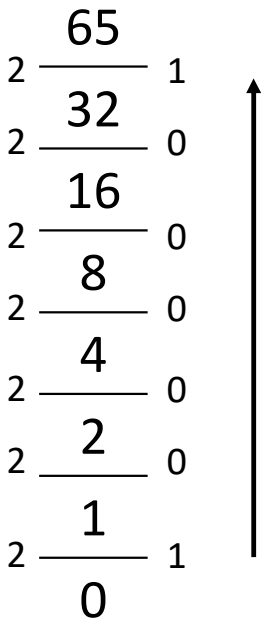
1= 2^1 = 2

1 = 2^0 = 1

8 + 0 + 2 + 1 = 11

Jadi, 1 0 1 1 = 11

Desimal ke Biner

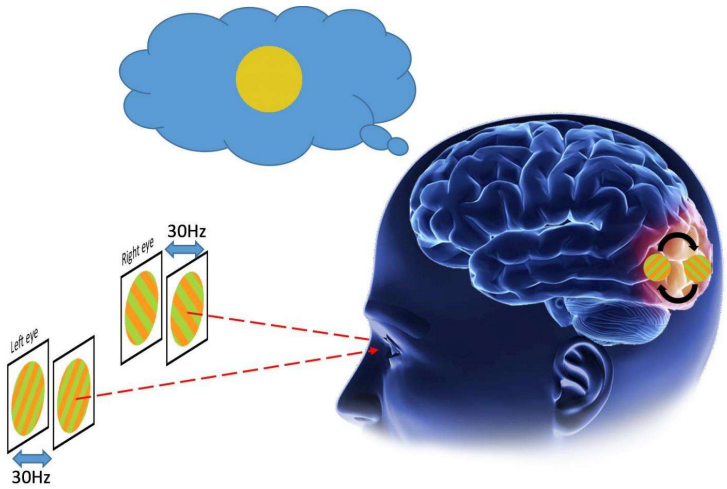


65 = 1 0 0 0 0 0 1

Image

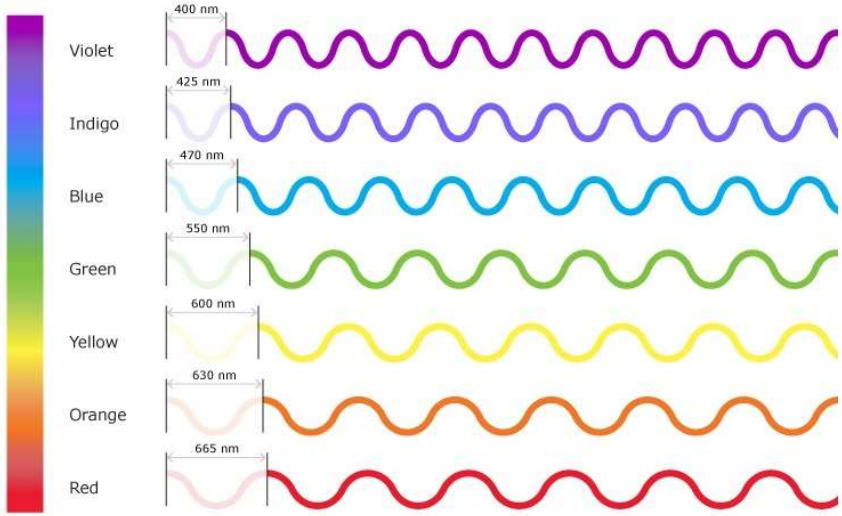
Human Visual System

- Komponen utama dari system visual manusia:
  - Mata → Menerima sensor (camera, scanner, dll)
  - Otak → unit pemroses informasi (computer)
  - Saraf optic → menghubungkan antara otak dan mata (kabel fisik, Bluetooth, wifi, dll)
- Fungsi penglihatan → membedakan objek dengan background, mendeteksi gerakan



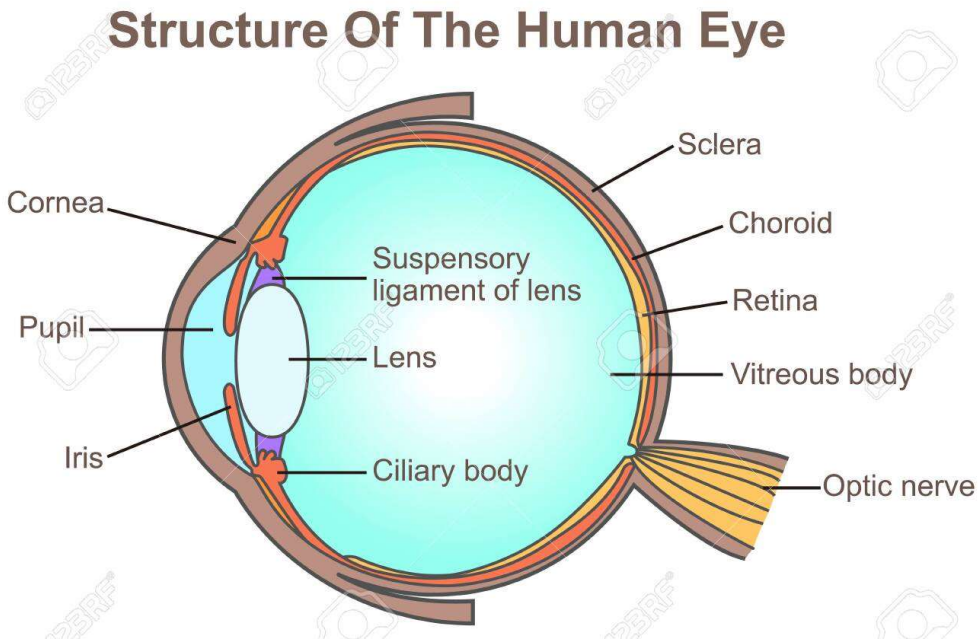
Visible Spectrum

- Manusia dapat mendeteksi Panjang gelombang cahaya antara 400-700 nm
  - Warna (hue) berhubungan dengan panjang gelombang cahaya
  - Kecerahan (brightness) berhubungan dengan intensitas cahaya
- Spekrum yang terlihat oleh mata manusia terbagi tiga:
  - **Blue** (400 – 500 nm)
  - **Green** (500 – 600 nm)
  - **Red** (600 – 700 nm)



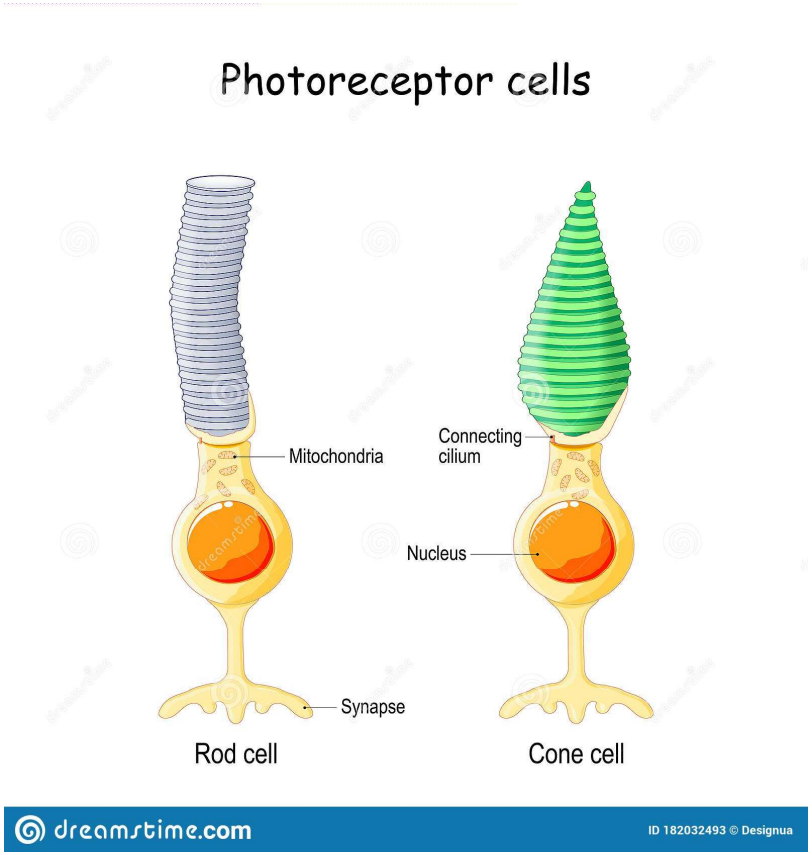
Structure of Human Eye

- Rata-rata diameter mata manuisa adalah $\pm 20\ mm$
- 3 membran / selaput yang membungkus mata manusia:
  - Kornea dan sklera
  - Choroid
  - Retina

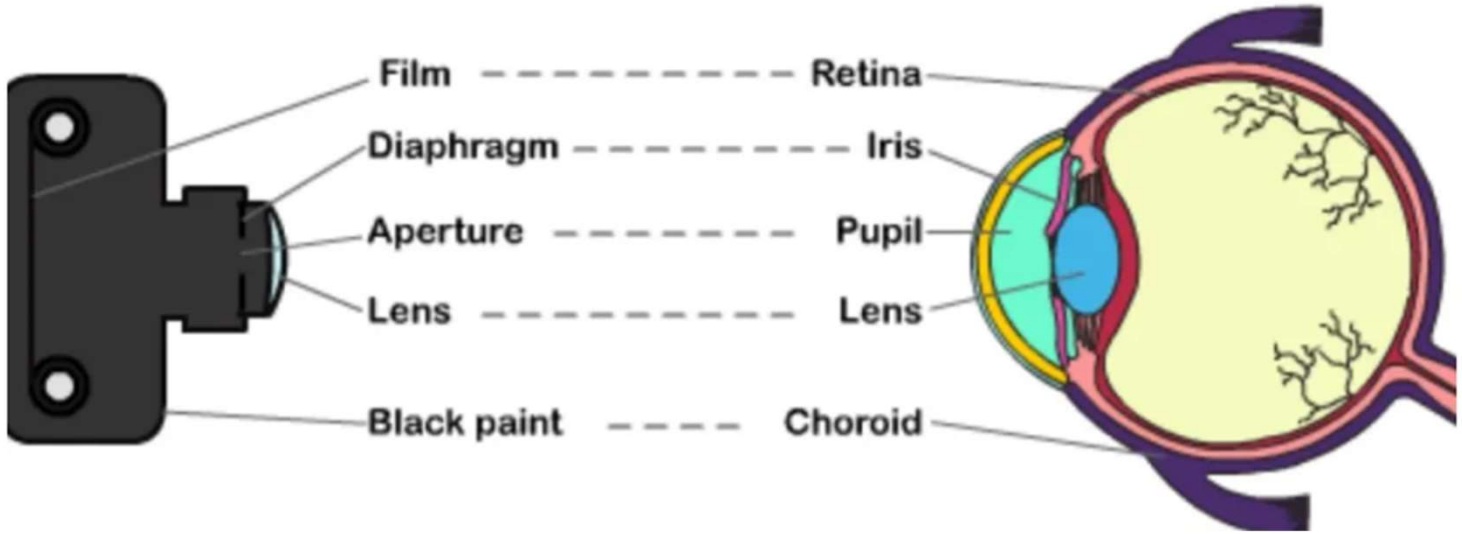


Receptors

- Ada 2 reseptor warna di mata:
  - Cone cell (sel kerucut) → terletak di bagian tengah retina, berjumlah 6- 7 juta di tiap mata. Sensitive terhadap warna dan bekerja dengan baik pada kondisi yang cukup terang.
  - Rod cell (sel batang) → terletak di permukaan retina, berjumlah 75 – 150 juta. Memberikan gambaran keseluruhan objek (hanya melihat kecerahan atau gray level, bukan warna) dan bekerja dengan baik pada cahaya yang redup.

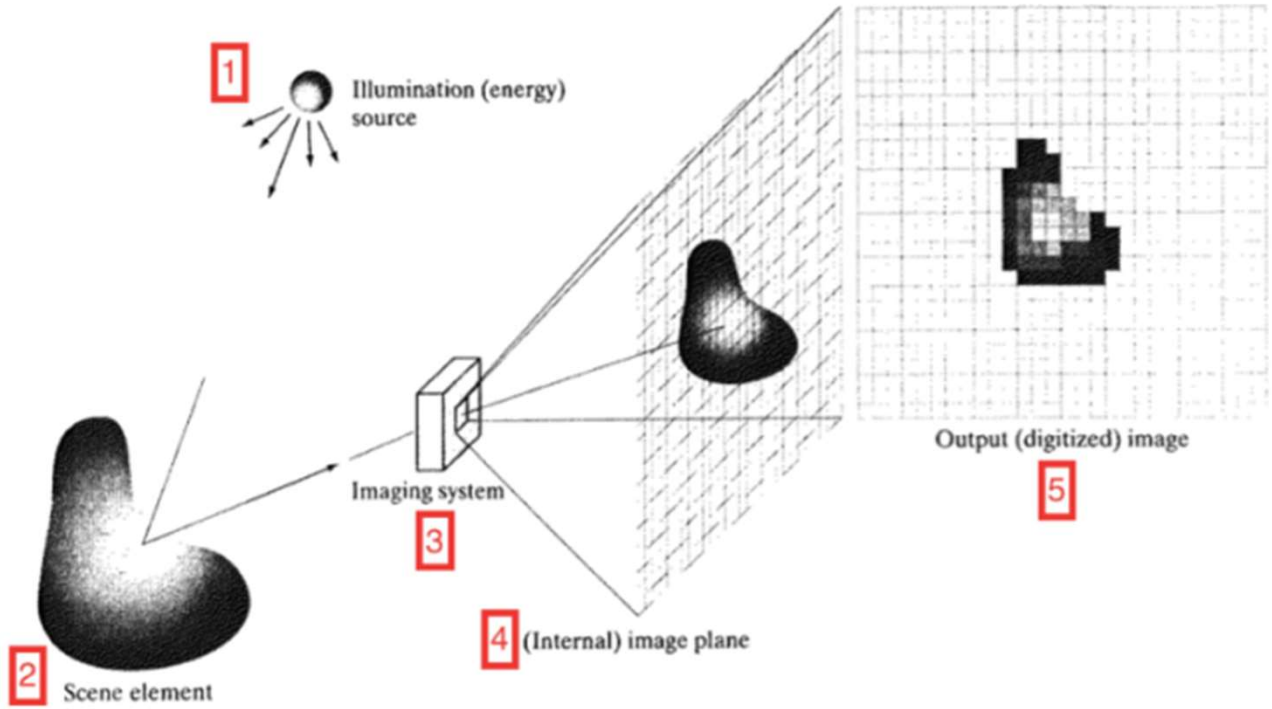


Human Eye and Camera



Human eye can be viewed as a type of camera

Image Acquisition

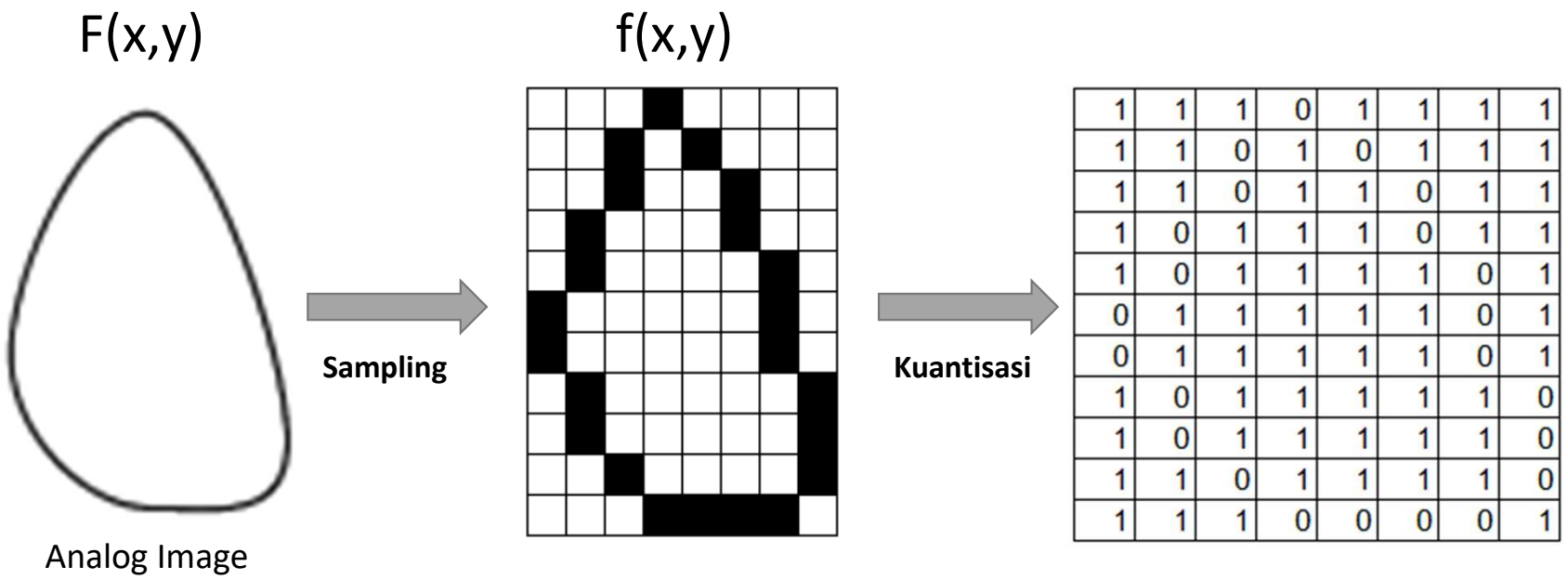


Analog and Digital Image

- 2 jenis citra:
  - Citra analog → diperoleh dari system optic yang menerima sinyal analog, seperti mata manusia dan kamera analog
  - Citra digital: citra 2 dimensi yang disimpan dan diproses oleh computer.

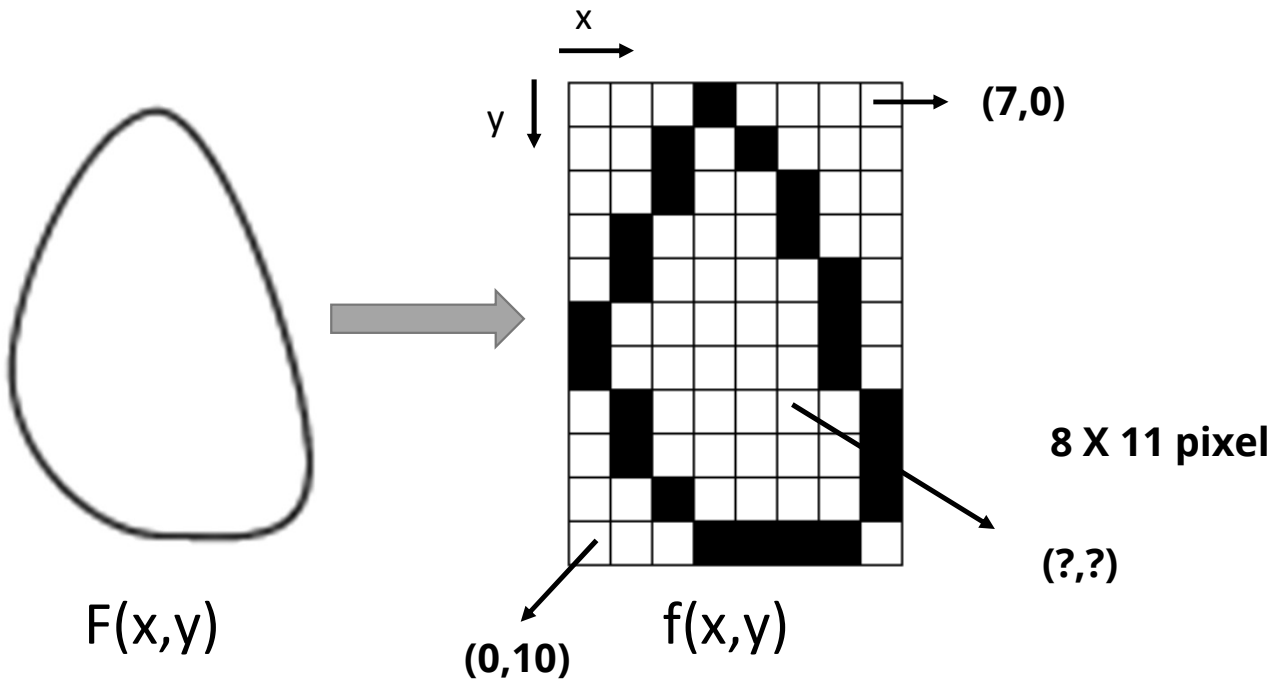
    Dibentuk oleh kumpulan titik yang dinamakan pixel/picture element, dimana tiap pixel memiliki koordinat posisi dan memiliki nilai intensitas cahaya
- Agar Citra dapat diolah oleh computer, citra analog harus diubah ke citra digital

Analog and Digital Image



1	1	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	0	1	1	0	1	1
1	0	1	1	1	0	1	1
1	0	1	1	1	1	0	1
0	1	1	1	1	1	0	1
0	1	1	1	1	1	0	1
1	0	1	1	1	1	1	0
1	0	1	1	1	1	1	0
1	1	0	1	1	1	1	0
1	1	1	0	0	0	0	1

Sampling

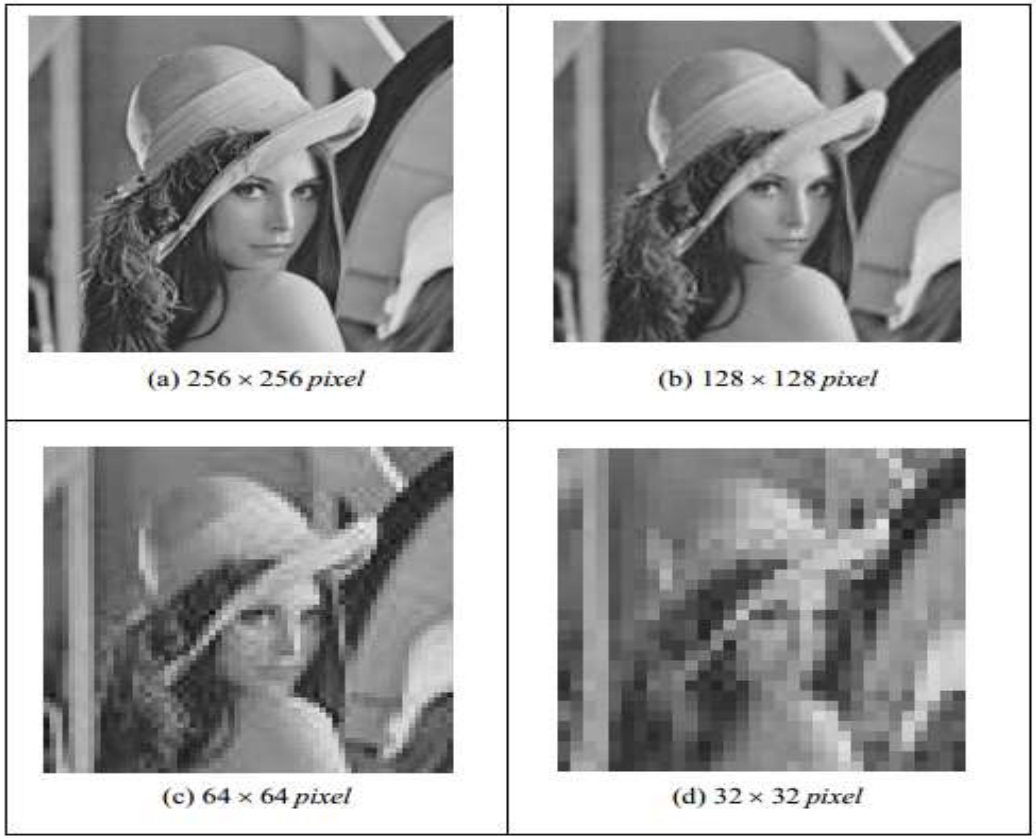


Sampling

- Proses sampling:
  - Downsampling

    Menurunkan jumlah pixel atau resolusi citra spasial
  - Upsampling

    Menaikkan jumlah pixel atau meningkatkan resolusi gambar

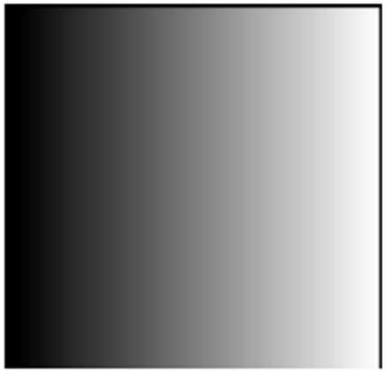


Quantization

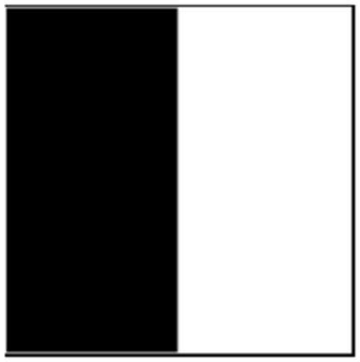
- Kuantisasi adalah memberi nilai pada tiap pixel berdasarkan intensitas citra (gray level/skala keabuan)
- Gray level adalah proses mengasosiasikan warna dengan tingkat warna tertentu / berdasarkan intensitas atau kecerahan.
- Rentang nilai gray level pixel dinyatakan dalam pixel depth. Nilai maksimal gray level bergantung pada kedalaman pixel (pixel depth).

Skala Keabuan	Rentang Nilai Keabuan	*Pixel Depth*
2^1 (2 nilai)	0, 1	1 bit
2^2 (4 nilai)	0 sampai 3	2 bit
2^3 (16 nilai)	0 sampai 15	3 bit
2^8 (256 nilai)	0 sampai 255	8 bit

Sampling and Quantization



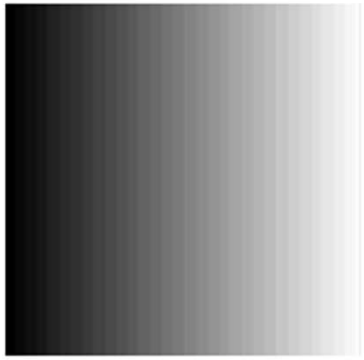
Original image



2 graylevel

4 graylevel

16 graylevel



32 graylevel

Calculate Memory

Jika citra digital berukuran *M* x *N* dan setiap pixel memiliki kedalaman *b* bit, maka kebutuhan memori untuk merepresentasikan citra adalah

$$M \times N \times b$$

(a) 256×256 *pixel*

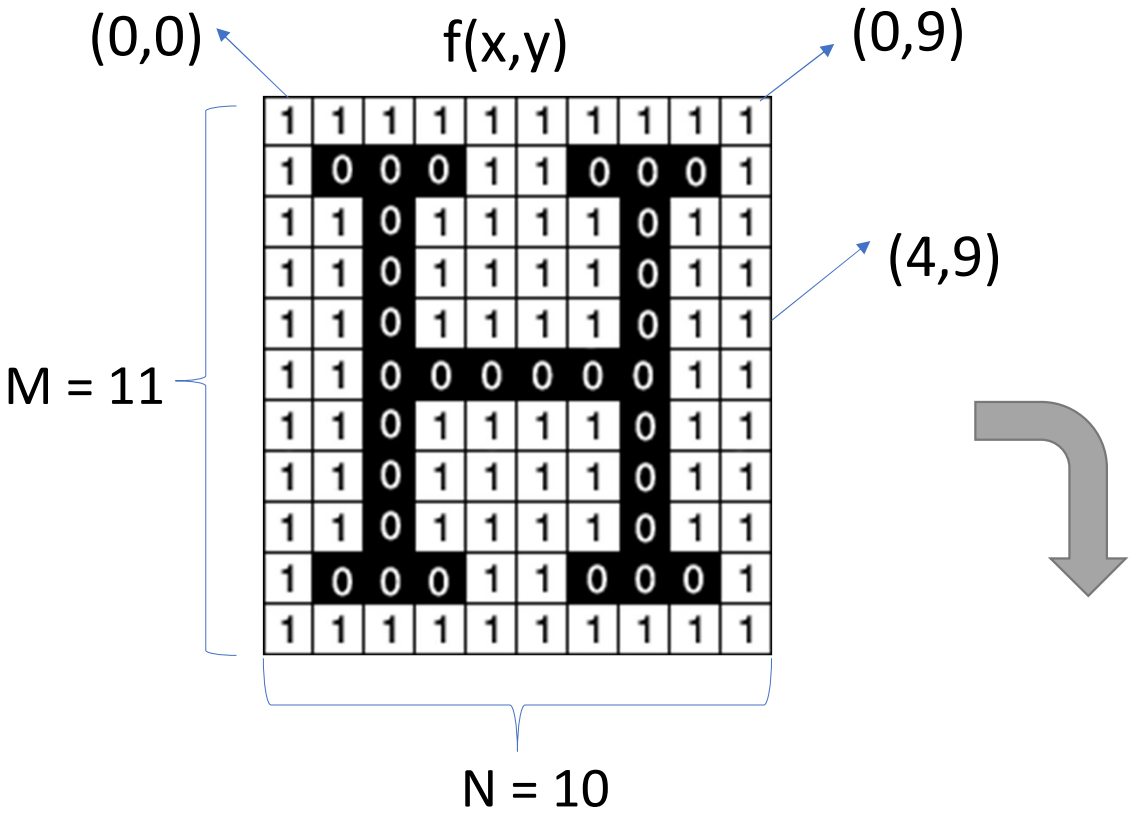
Citra berukuran 256 x 256 pixel dengan kedalaman 8 bit membutuhkan memori sebesar:

$$memory = 256 \times 256 \times 8 = 524.288\ bit$$
$$= 65.536\ byte = \pm 66\ Kilo\ byte$$

Image Representation

F(x,y)

Analog Image

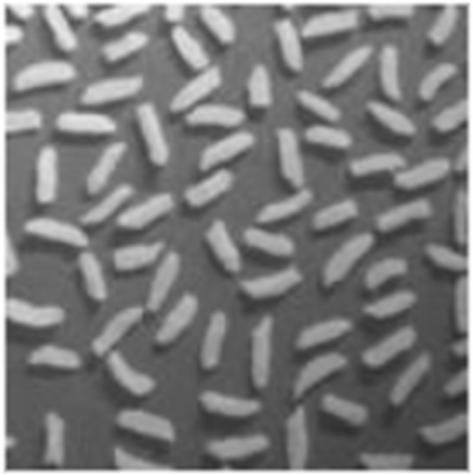


Digital Image

Binary Image

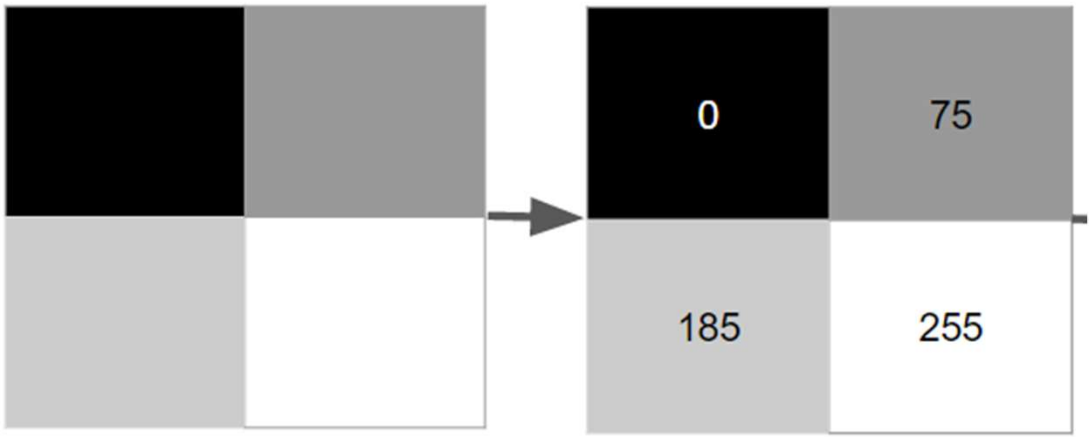
- Citra biner hanya memiliki 2 nilai: Hitam dan Putih atau 0 dan 1
- Citra 1 bit
- Dihasilkan dengan melakukan operasi *threshold* (*T*).

$$g(x,y) = \begin{cases} 1, jika\ f(x,y) \geq T \\ 0, jikaf(x,y)\ < T \end{cases}$$



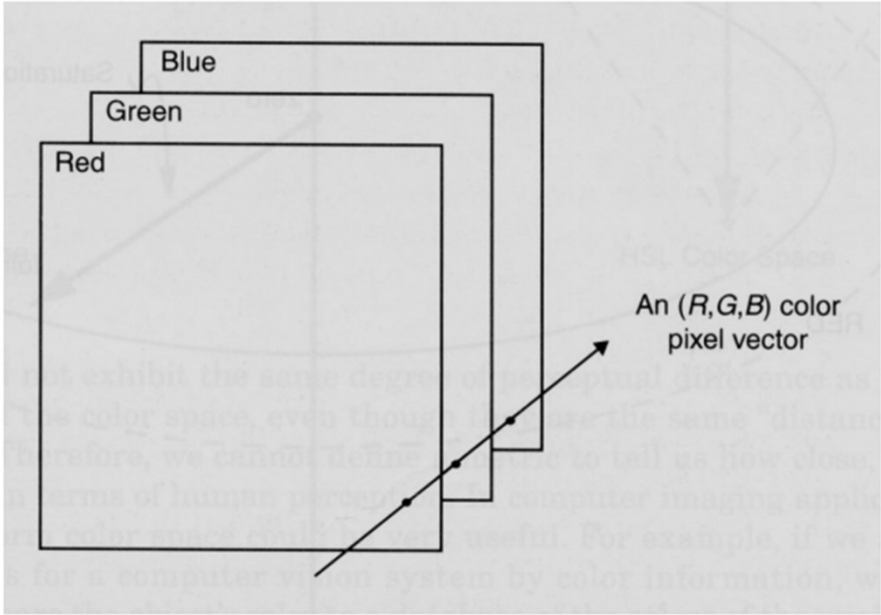
Grayscale Image

- Citra grayscale hanya memiliki satu nilai warna (monokrom) pada tiap pixel
- Citra grayscale tidak memberikan informai apapun tentang warna.
- Tiap pixel menentukan skala keabuan yang berbeda.
- Citra grayscale normal berisi 8 bit.



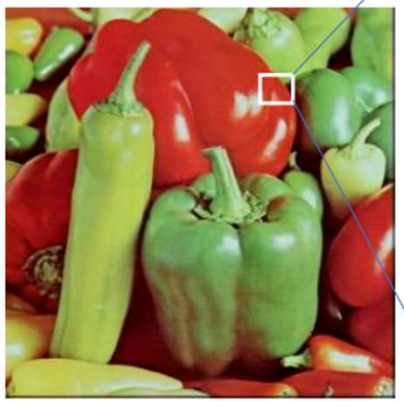
Color Image

- Terdiri dari 3 kanal (channel) warna RGB: Red, Green, dan Blue.
- Intensitas suatu titik pada citra berwarna merupakan kombinasi dari tiga intensitas: derajat keabuan merah (fmerah(x,y)), hijau (fhijau(x,y)), dan biru (fbiru(x,y))



Color Image

Color Image



Red

148	162	175	182	189	194	195	193	195	195	197
148	164	174	176	185	189	191	191	196	194	195
144	159	167	176	178	185	188	191	196	194	197
128	147	157	168	173	179	182	184	191	191	192
119	134	148	160	164	170	179	176	181	189	185
145	124	142	151	160	168	169	174	180	182	183
172	120	140	153	157	169	171	178	180	182	182
196	120	129	144	152	158	167	170	177	176	178
204	144	116	134	142	149	155	165	165	170	171

Green

42	43	48	50	53	56	56	53	54	54	54
50	49	51	47	53	55	56	55	59	55	54
51	48	47	49	49	51	50	52	54	51	54
53	48	45	49	50	52	50	48	51	50	50
59	43	43	48	47	48	54	47	49	55	50
100	42	41	42	44	46	45	46	50	52	50
142	47	43	42	39	46	44	48	49	51	49
185	65	44	42	42	43	48	46	50	48	49
209	106	44	42	41	42	44	50	48	50	49

Blue

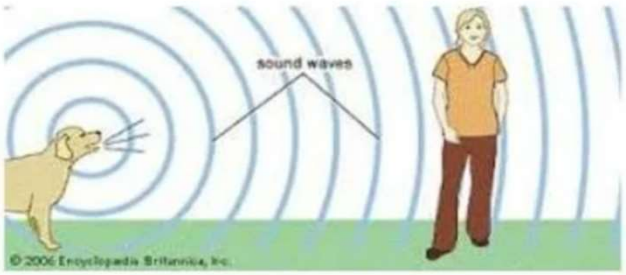
16	24	32	35	37	40	40	37	37	38	36
19	25	31	28	34	37	38	37	40	35	33
17	23	27	33	32	35	33	36	39	35	37
20	19	23	31	33	34	34	32	36	35	35
29	16	24	33	32	34	39	30	31	38	34
71	11	18	24	30	33	30	30	34	36	34
113	14	16	21	24	32	30	32	33	35	33
156	32	13	20	25	28	33	31	35	33	32
177	72	9	16	22	26	30	35	32	33	32

Image Format

- Penyimpanan dan pengambilan citra dilakukan melalui file
- Penyimpanan citra membutuhkan memori yang banyak. Contoh: sebuah citra grayscale (8 bit) berukuran 1024 x 1024 membutuhkan 1024 x 1024 x 8 bit atau 8,388,608 bit ≈ 1MByte.
- Karena membutuhkan memori yang besar, tiap citra disimpan dalam bentuk yang sudah terkompres. 2 format file citra:
  - No lossy image file format: tidak ada informasi yang hilang pada tahap kompresi. contoh: Tagged Image File Format (TIFF), Portable Network Graphics (PNG), Graphics Interchange Format (GIF)
  - Lossy image file format: terdapat informasi yang hilang pada tahap kompresi. contoh: Joint Photographic Expert Group (JPEG)

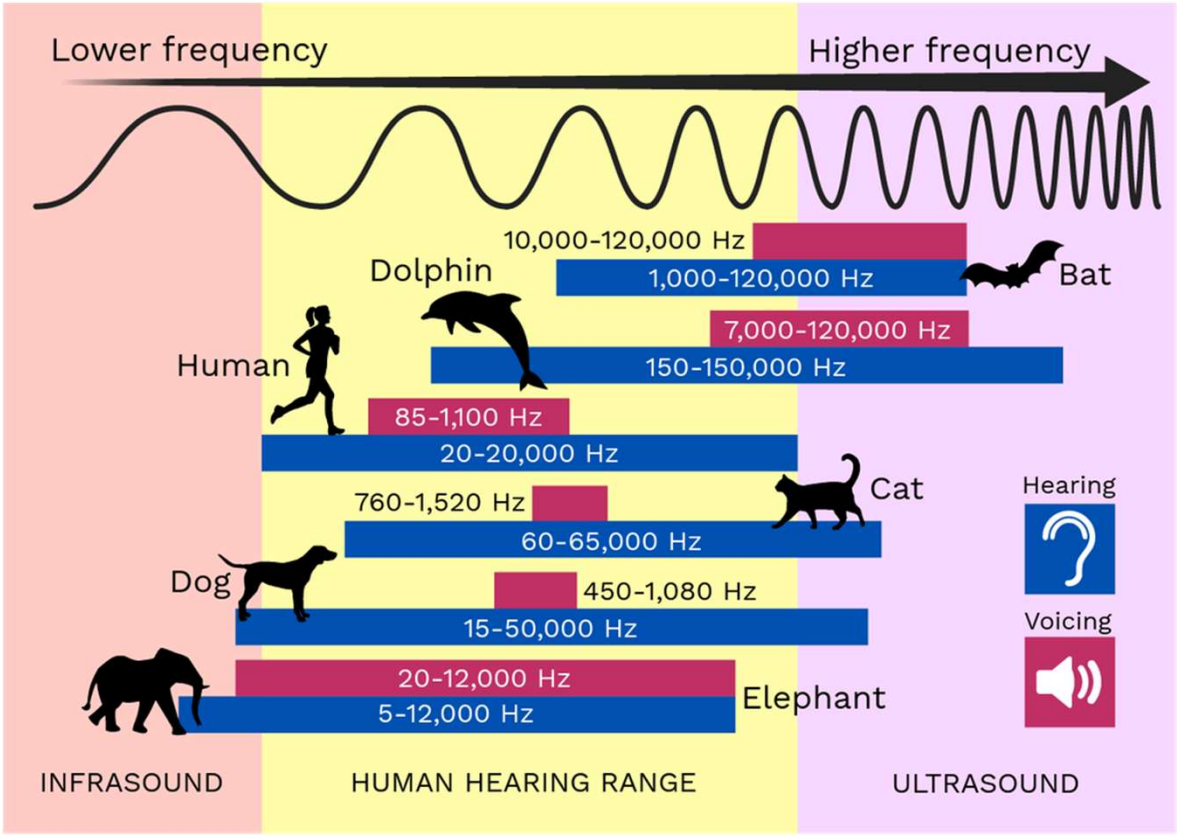
Audio

Audio Sound

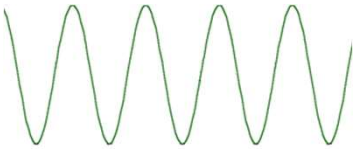


- Bunyi terjadi karena adanya getaran dari suatu objek, yang menyebabkan udara di sekitarnya bergetar. Pola osilasi yang terjadi dinamakan sebagai gelombang suara (sound wave).
- Getaran udara ini menyebabkan gendang telinga manusia bergetar dan otak menginterpretesaikan sebagai bunyi
- Bunyi tidak bisa merambat melalui ruang hampa

Audio Frequency Range

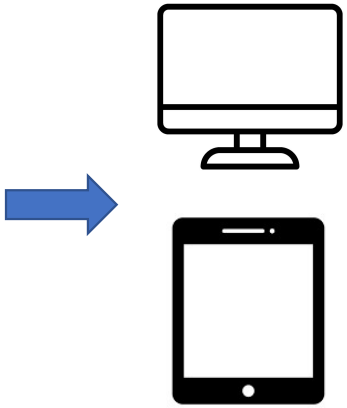


Audio Sound



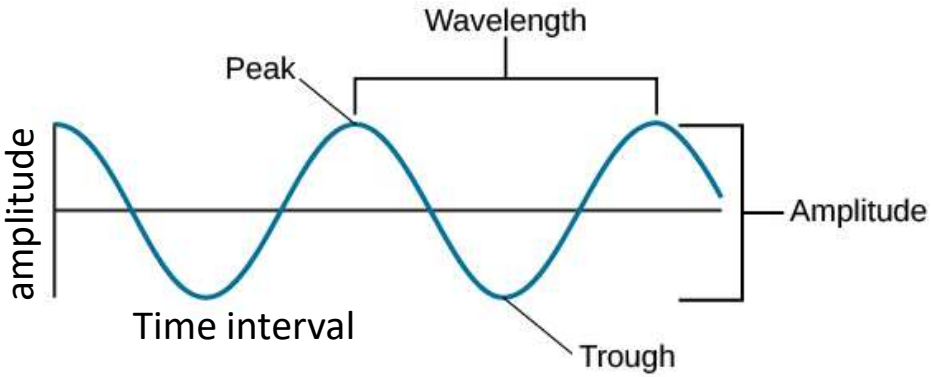
Sound wave

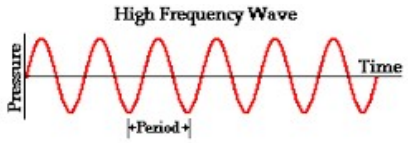
Konversi Analog ke Digital (ADC)



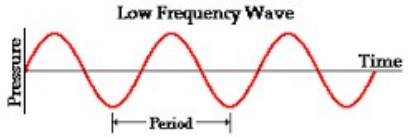
Sound wave

- Sound wave (gelombang bunyi): sinyal analog yang bergerak di udara
- Tiga sifat penting gelombang:
  - Amplitudo
  - Frekuensi
  - Wavelength (panjang gelombang)
- Sinyal analog memiliki amplitude yang berubah secara kontinyu terhadap waktu





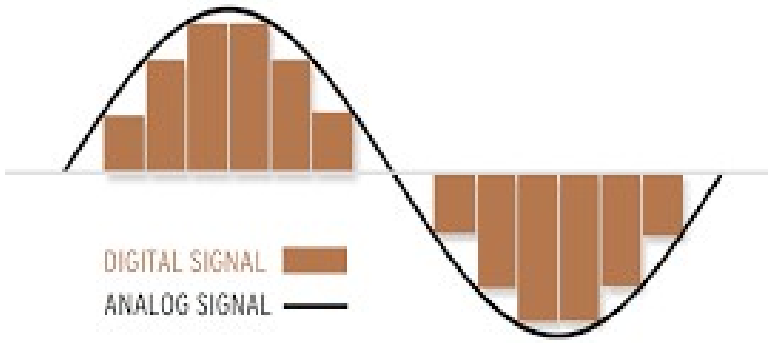
High frequency



Low frequency

Audio Acquisition

- Gelombang bunyi analog tidak dapat langsung direpresentasikan pada computer
- Agar dapat diproses oleh computer gelombang bunyi analog harus diubah menjadi digital
- Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan **sample**.

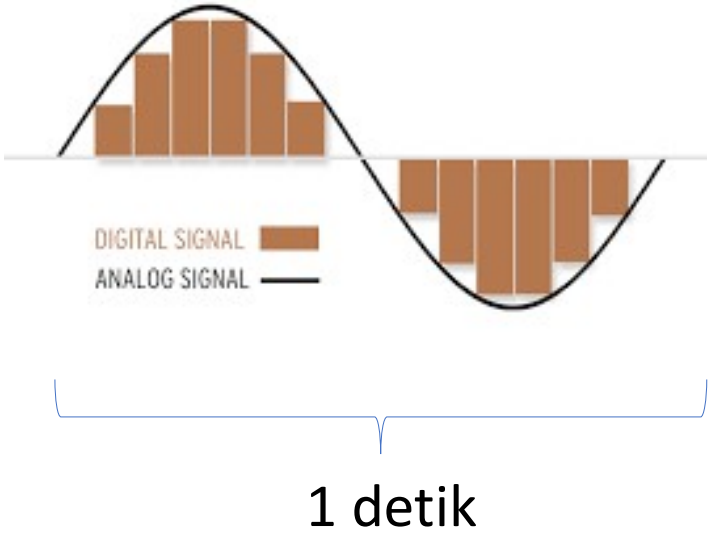


Digital Audio

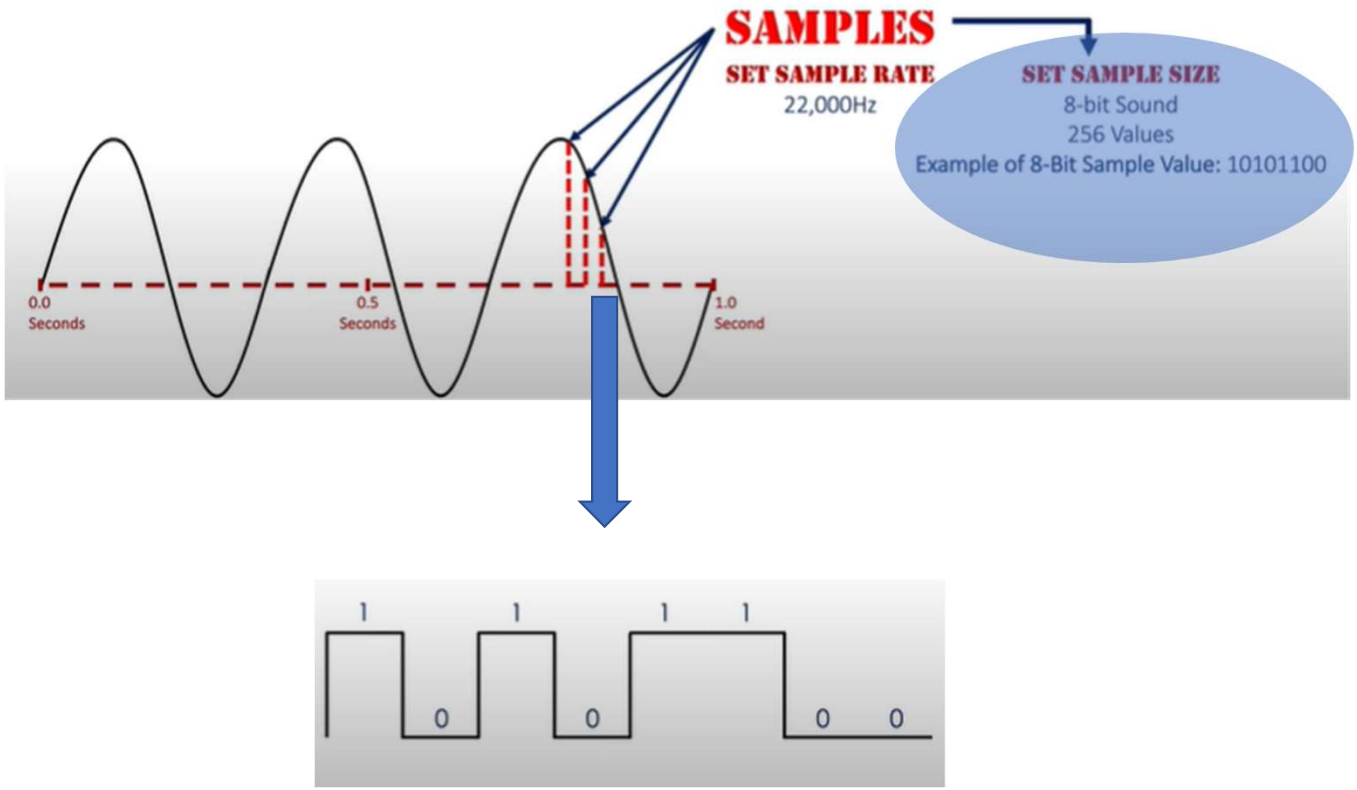
- Digital audio adalah sistem dimana kita bisa menyimpan, membuat ulang, dan memanipulasi informasi audio dalam sistem komputer
- Ribuan sampel perlu diambil dari gelombang bunyi analog untuk mewakili data audio secara memadai dalam bentuk digital.
- Beberapa parameter harus diatur Ketika mengubah gelombang bunyi analog ke digital:
  - Sampling rate: berapa kali sampel (irisan) diambil dari gelombang bunyi. Contoh rekomendasi sample rates: 44.100 Hz untuk Suara kualitas CD
  - Sampling resolution (bit-depth): jumlah bits per sampel (banyaknya amplitude pada tiap sampel), umumnya adalah 8-bit dan 16-bit

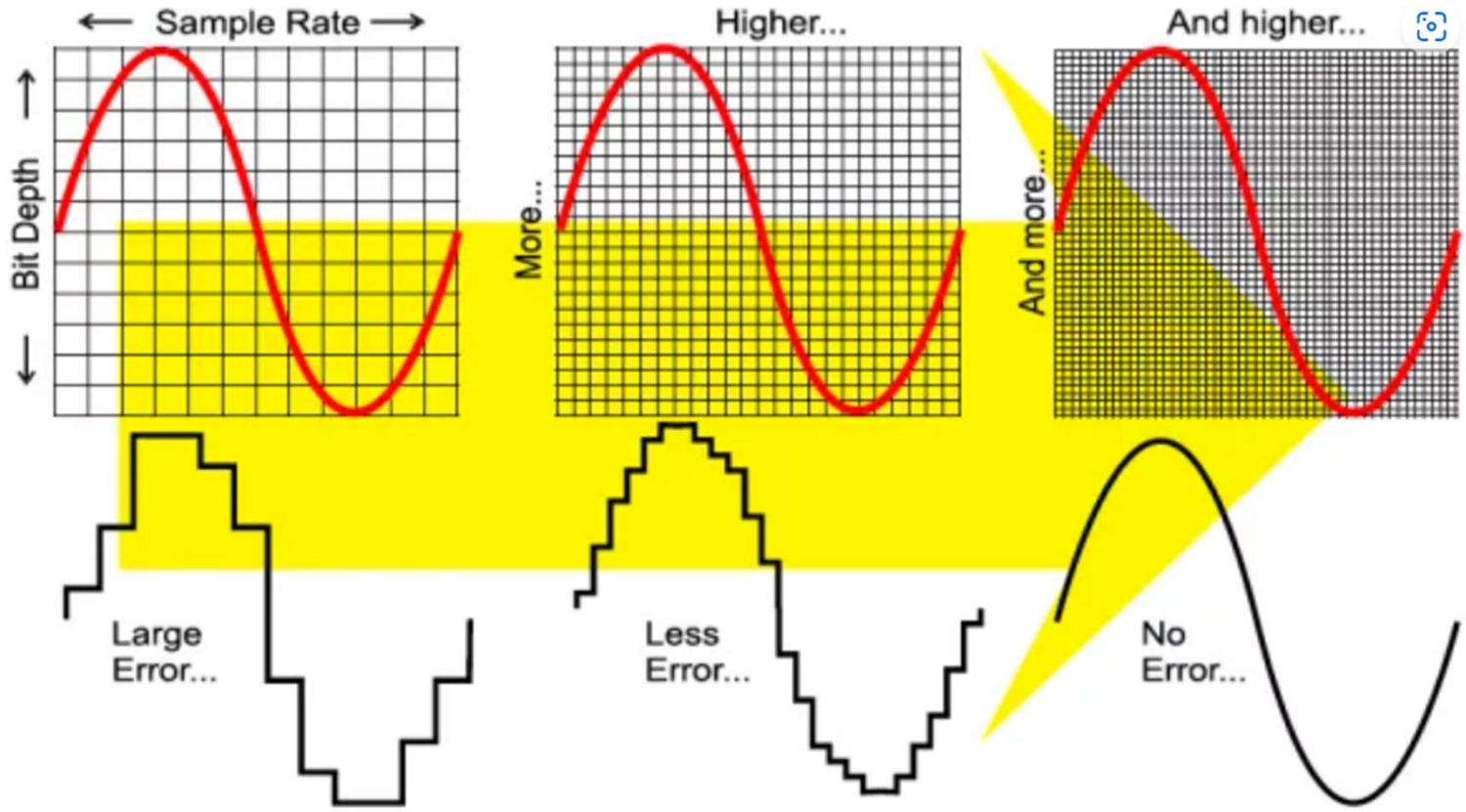
Analog to Digital Audio

- Proses merubah analog ke digital audio:
  - **Sampling:** proses mengambil potongan dari gelombang bunyi dan mengubah data gelombang menjadi data digital untuk digunakan oleh sistem komputer.
  - **Kuantisasi**: proses pembulatan nilai amplitudo ke nilai terdekat. Sinyal sampel dikuantisasi menjadi nilai diskrit.



Analog to Digital Audio - Quantization




Sample Rate
Higher...
And higher...
Bit Depth
More...
And more...
Large Error...
Less Error...
No Error...

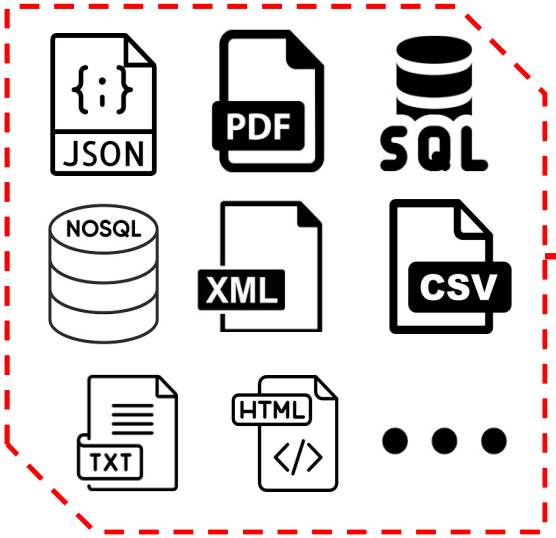
Audio Format

- Jumlah bit yang digunakan unuk merepresentasikan sample bunyi memiliki peran penting pada transmisi dan penyimpanan.
- Semakin besar bit maka data yang ditransmisi semakin besar dan memori yang diperlukan untuk menyimpan juga semakin besar.
- Bit rate : jumlah bit per waktu
- Format penyimpanan audio:
  - Lossless: tidak ada informasi yang hilang pada tahap kompresi. contoh: Linear PCM dan Compact Disc
  - Lossy: terdapat informasi yang hilang pada tahap kompresi. contoh: MPEG, mp3

Text

Text

- Data teks adalah jenis data tidak terstruktur yang jumlahnya semakin meningkat dengan pesat.
- Contoh data teks: Media sosial, journal penelitian, review aplikasi, dll



ASCII

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	U+0000 NUL 0	U+0001 SOH 1	U+0002 STX 2	U+0003 ETX 3	U+0004 EOT 4	U+0005 ENQ 5	U+0006 ACK 6	U+0007 BEL 7	U+0008 BS 8	U+0009 HT 9	U+000A LF 10	U+000B VT 11	U+000C FF 12	U+000D CR 13	U+000E SO 14	U+000F SI 15
1_	U+0010 DLE 16	U+0011 DC1 17	U+0012 DC2 18	U+0013 DC3 19	U+0014 DC4 20	U+0015 NAK 21	U+0016 SYN 22	U+0017 ETB 23	U+0018 CAN 24	U+0019 EM 25	U+001A SUB 26	U+001B ESC 27	U+001C FS 28	U+001D GS 29	U+001E RS 30	U+001F US 31
2_	U+0020 SP 32	U+0021 ! 33	U+0022 " 34	U+0023 # 35	U+0024 $ 36	U+0025 % 37	U+0026 & 38	U+0027 ' 39	U+0028 (40	U+0029) 41	U+002A * 42	U+002B + 43	U+002C , 44	U+002D - 45	U+002E . 46	U+002F / 47
3_	U+0030 0 48	U+0031 1 49	U+0032 2 50	U+0033 3 51	U+0034 4 52	U+0035 5 53	U+0036 6 54	U+0037 7 55	U+0038 8 56	U+0039 9 57	U+003A : 58	U+003B ; 59	U+003C < 60	U+003D = 61	U+003E > 62	U+003F ? 63
4_	U+0040 @ 64	U+0041 A 65	U+0042 B 66	U+0043 C 67	U+0044 D 68	U+0045 E 69	U+0046 F 70	U+0047 G 71	U+0048 H 72	U+0049 I 73	U+004A J 74	U+004B K 75	U+004C L 76	U+004D M 77	U+004E N 78	U+004F O 79
5_	U+0050 P 80	U+0051 Q 81	U+0052 R 82	U+0053 S 83	U+0054 T 84	U+0055 U 85	U+0056 V 86	U+0057 W 87	U+0058 X 88	U+0059 Y 89	U+005A Z 90	U+005B [91	U+005C \ 92	U+005D] 93	U+005E ^ 94	U+005F _ 95
6_	U+0060 ` 96	U+0061 a 97	U+0062 b 98	U+0063 c 99	U+0064 d 100	U+0065 e 101	U+0066 f 102	U+0067 g 103	U+0068 h 104	U+0069 i 105	U+006A j 106	U+006B k 107	U+006C l 108	U+006D m 109	U+006E n 110	U+006F o 111
7_	U+0070 p 112	U+0071 q 113	U+0072 r 114	U+0073 s 115	U+0074 t 116	U+0075 u 117	U+0076 v 118	U+0077 w 119	U+0078 x 120	U+0079 y 121	U+007A z 122	U+007B { 123	U+007C | 124	U+007D } 125	U+007E ~ 126	U+007F DEL 127

ASCII CODE

ASCII Code

- Karakter ASCII code terdiri dari 7 bit atau merepresentasikan 128 karakter (nilainya memiliki rentang 0 s/d 127 pada bilangan decimal)
- ASCII code merepresentasikan huruf, angka, dan karakter yang ada di keyboard standar

  Contoh: 72 69 76 76 79

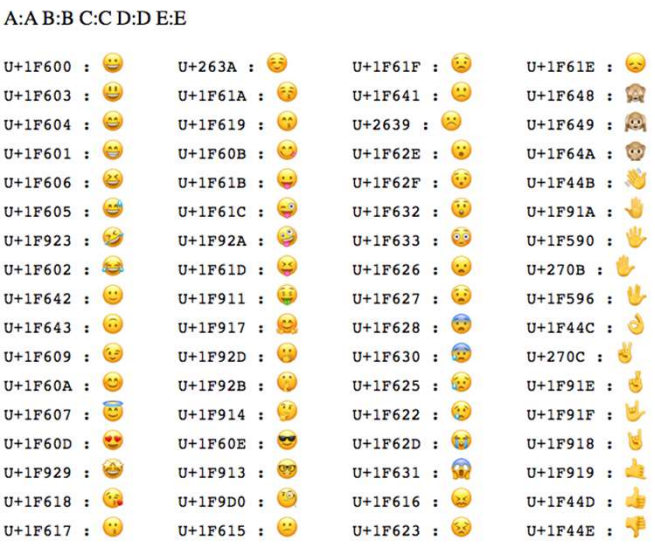
  HELLO
- Extended ASCII code terdiri dari 8 bit kumpulan karakter yang merepresentasikan 256 karakter, sehingga karakter ε, θ, σ, á, bisa diwakili juga oleh Extended ASCII code.

Char.	ASCII	Char.	ASCII	Char.	ASCII
@	64	U	85	j	106
A	65	V	86	k	107
B	66	W	87	l	108
C	67	X	88	m	109
D	68	Y	89	n	110
E	69	Z	90	o	111
F	70	[	91	p	112
G	71	\	92	q	113
H	72	]	93	r	114
I	73	^	94	s	115
J	74	_	95	t	116
K	75	`	96	u	117
L	76	a	97	v	118
M	77	b	98	w	119
N	78	c	99	x	120
O	79	d	100	y	121
P	80	e	101	z	122
Q	81	f	102	{	123
R	82	g	103	|	124
S	83	h	104	}	125
T	84	i	105	~	126

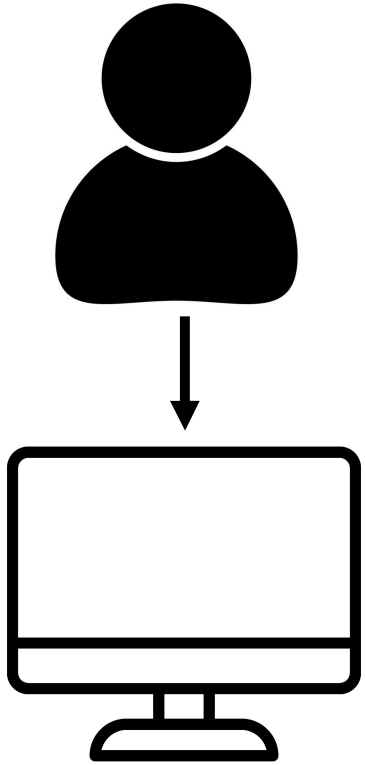
UNICODE

- UNICODE bisa menggunakan 8, 16, atau 32 bit untuk tiap karakternya, sehingga UNICODE bisa merepresentasikan karakter dan semua bahasa di dunia.
- UNICODE lebih besar dari ASCII code sehingga UNICODE membutuhkan penyimpanan yang lebih besar dari ASCII code.
- 32 bit → bisa merepresentasikan 2.147.483.647 karakter yang berbeda, termasuk emojis

128.514



Challenges in Processing text data



- Peningkatan jumlah data teks yang signifikan
- Untuk memproses data teks membutuhkan waktu yang lama

- Komputer tidak mampu belajar sendiri
- Komputer memahami data numerik tapi tidak memahami Bahasa, kata, dll
- Komputer memproses biner bukan karakter

Text Representation

- Mengubah bahasa, kata dalam data teks menjadi data numerik agar dapat dibaca/diproses oleh computer (ASCII code dan UNICODE)
- Famous text representation dalam text-mining:
  - One-Hot encoding
  - Basic Bag-of-words representation - CountVectorizer
  - Advanced Bag-of-words (BOW) - TF-IDF
- Istilah penting → “Korpus”

  Kumpulan teks yang tersimpan secara elektronik untuk berbagai kebutuhan

One-Hot Encoding

- Memberikan nilai 0 untuk semua elemen dalam vector kecuali untuk satu elemen, dimana memilki nilai 1.
- Jumlah array berdasarkan jumlah kata dalam korpus.
- Permasalahan: bagaimana jika ada 100.000 kata unik dalam korpus?

- Contoh:

korpus: “I ate an apple”

langkah:

1. Buat index posisi kata dari kalimat yang diberikan, index biasanya urut sesuai dengan abjadnya.

I	ate	an	apple
0	3	1	2

2. Buat Vektor One-Hot Encoding

	0	1	2	3
I	1	0	0	0
ate	0	0	0	1
an	0	1	0	0
apple	0	0	1	0

sehingga untuk keseluruhan korpus:

[[1 0 0 0] [0 0 0 1] [0 1 0 0][0 0 1 0]]

One-Hot Encoding

- Kelebihan:
  - Mudah dipahami dan diimplementasikan
- Kekurangan:
  - Ruang fitur (feature space) akan sangat besar jika jumlah kategori sangat tinggi
  - Representasi vector kata adalah orthogonal dan tidak bisa mengukur hubungan antara kata-kata yang berbeda
  - Tidak bisa mengukur pentingnya sebuah kata dalam kalimat, hanya memahami ada/tidaknya sebuah kata dalam kalimat
  - Membutuhkan memori besar dan biaya komputasi yang tinggi

Bag-of-words (BOW) - CountVectorizer

- BOW → menempatkan kata dalam 'tas' (bag) dan menghitung frekuensi kemunculan setiap kata
- Tidak memperhitungkan urutan kata atau informasi leksikal untuk representasi teks
- CountVectorizer → menghitung frekuensi kemunculan sebuah kata dalam dokumen
- Contoh:

Data1 = the red dog
Data2 = cat eat dog
Data3 = dog eat food
Data4 = red cat eat

	the	red	dog	cat	eats	food
Data1	1	1	1	0	0	0
Data2	0	0	1	1	1	0
Data3	0	0	1	0	1	1
Data4	0	1	0	1	1	0

CountVectorizer

- Kelebihan:
  - Dapat memberikan frekuensi kata-kata dalam dokumen teks/kalimat yang tidak bisa diberikan oleh One-Hot Encoding
  - Penjang vector yang disandikan adalah Panjang kamus
- Kekurangan
  - Mengabaikan informasi lokasi kata.
  - Tidak bisa memahami arti kata
  - Kata-kata yang berfrekuensi tinggi biasanya malah kata-kata yang tidak penting, seperti “is”, “are”, “an”, “I”, dll

TF-IDF

- Perlu memberikan bobot dari kata-kata
- TF-IDF → Term Frequency-Inverse Document Frequency. Bobot yang diberikan untuk setiap kata tidak hanya bergantung pada frekuensi kata, tetapi juga seberapa sering kata tersebut berada di seluruh korpus.
- TF-IDF merupakan perkalian dari 2 factor:

$$TF - IDF = TF(w, d) * (IDF(w))$$

$$IDF(w) = \log\left(\frac{N}{df(w)}\right)$$

dimana:

$TF(w, d)$ = frekuensi kata 'w' dalam dokumen 'd'

N = jumlah total dokumen

$df(w)$ = frekuensi dokumen yang mengandung kata 'w'

- Kata-kata yang sering muncul (stopwords) memiliki bobot rendah

TF-IDF

- Kelebihan:
  - Implementasi sederhana, mudah dipahami, dan diartikan
  - Memberikan bobot yang rendah untuk kata-kata yang sering muncul
- Kekurangan
  - Informasi posisi kata masih belum bisa ditangkap
  - Sangat bergantung pada korpus. Contoh: representasi matriks yang dihasilkan oleh data covid tidak bisa digunakan untuk data gigi.

Terima Kasih

Referensi:

Camastra, F. dan Vinciarelli, A., 2015, Machine Learning for Audio, Image, and Video Analysis Theory and Applications 2nd edition, Springer, London.
Marinai, S. dan Fujisawa, H., 2008, Machine Learning in Document Analysis and Recognition, Springer, Berlin Heidelberg.
Bird, S., Klein, E., dan Loper, E., 2009, Natural Language Processing with Python, O'Reilly Media Inc., USA.