

HANDOUT KULIAH  
**PENGOLAHAN SINYAL  
MULTIMEDIA**

Oleh  
**KOREDIANTO USMAN**  
Program Studi Sarjana Teknik Telekomunikasi  
Fakultas Teknik Elektro  
Universitas Telkom



**TELKOM UNIVERSITY PRESS**

HANDOUT KULIAH

# PENGOLAHAN SINYAL MULTIMEDIA

Handout ini adalah materi pengiring pada perkuliahan di mata kuliah:  
**TTI4V3 PENGOLAHAN SINYAL MULTIMEDIA (PSM)**  
di lingkungan program studi Sarjana Teknik Telekomunikasi  
Fakultas Teknik Elektro  
Universitas Telkom

Oleh

KOREDIANTO USMAN

EDISI 1

JULI 2021

TELKOM UNIVERSITY PRESS

# Kata Pengantar

Pada Kurikulum 2020 yang diberlakukan di lingkungan program studi Sarjana Teknik Telekomunikasi, dimunculkan mata kuliah baru yaitu **Pengolahan Sinyal Multimedia** yang disingkat juga dengan **PSM** dengan kode mata kuliah TTI4V3 sebagai mata kuliah pilihan pada rumpun keilmuan Pengolahan Sinyal Informasi (PSI). Pemunculan mata kuliah ini karena dirasakan perlu jembatan antara mata kuliah dasar yaitu Pengolahan Sinyal Waktu Diskrit (PSWD) (atau Pengolahan Sinyal Digital atau PSD) dengan mata kuliah aplikasi lanjut dari pengolahan sinyal seperti *speech signal processing*, *computer vision*, steganografi dan *watermarking*. Mata kuliah PSM akan dibahas konsep dasar yang lebih terperinci tentang Audio, Citra, dan Video. Ketiga media ini tidak dibahas pada MK PSWD atau PSD, di sisi lain, MK seperti *speech signal processing* mengasumsikan mahasiswa sudah memahami tentang konsep sinyal audio dan MK *Computer Vision* dan MK Steganografi dan *Watermarking* juga telah mengasumsikan bahwa mahasiswa telah memahami tentang konsep citra digital.

Dengan posisinya sebagai jembatan pada mata kuliah dasar pengolahan sinyal dan mata kuliah aplikasi lanjut pengolahan sinyal tersebut, PSM didesain untuk membekali mahasiswa agar memiliki konsep yang lebih kuat pada bidang audio, citra dan video tersebut. Idealnya, mata kuliah ini diambil setelah menyelesaikan mata kuliah PSWD atau PSD dan sebelum mengambil mata kuliah **speech signal processing**, **computer vision**, steganografi dan **watermarking**, atau mata kuliah lainnya yang memerlukan dasar pemahaman pada audio, citra dan video.

Materi pada handout ini disusun untuk keperluan mengajar satu semester dengan durasi **14 minggu** dan satu minggu satu (atau dua) pertemuan. Terdapat **14 slide** dalam handout ini, dengan penyampaian satu pertemuan **satu slide**. Sebaiknya keseluruhan materi handout ini diberikan kepada mahasiswa pada awal semester, sehingga mahasiswa dapat mempelajari setiap slide sebelum perkuliahan dimulai dan mencoba dulu beberapa latihan yang terdapat di slide. Jika ada pertanyaan dalam proses belajar tersebut, maka mahasiswa dapat menyampaikannya pada saat perkuliahan.

Slide pertama terdiri dari dua bagian yaitu bagian A dan bagian B. Bagian A berisi tentang peraturan perkuliahan dan mekanisme penilaian. Isi dari bagian A ini bisa disesuaikan dengan style penilaian yang disepakati bersama dengan dosen lain jika terdapat beberapa kelas. Bagian B mulai masuk ke materi PSM.

Untuk keperluan evaluasi, bersama slide ini juga, telah dibuat set soal kuis untuk 14 kali kuis (dikembangkan terpisah), dengan kata lain, setiap selesai perkuliahan,

maka diberikan kuis dengan durasi sekitar 10 menit dengan jumlah soal rata-rata 10 soal untuk setiap kuis.

Penulis menggunakan platform Quizziz (**quizziz.com**), dengan mengacak soal dan opsi jawaban, dan memastikan waktu kuis sekitar 10 menit, teknik ini cukup efektif dalam mengevaluasi pemahaman mahasiswa terhadap materi yang diberikan.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada Ibu Kaprodi S1 Teknik Telekomunikasi pada periode 2021 saat handout ini dikembangkan yaitu Ibu Dr. Levy Olivia Nur, atas kepercayaannya kepada penulis untuk mengampu dan mengembangkan handout perkuliahan ini, demikian juga terima kasih penulis sampaikan kepada Ibu Rita Magdalena, S.T., M.T., selaku ketua kelompok keahlian Pengolahan Sinyal Informasi, tempat penulis bernaung, atas semua bantuannya dalam mendukung penulisan handout ini. Tak lupa penulis mengucapkan terima kasih kepada rekan, Bapak Gelar Budiman yang juga berkenan *men-share* materi-materi terkait PSM.

Akhirul kalam, terdapat banyak kekurangan pada handout ini, penulis akan sangat senang hati menerima masukan dan saran untuk pengembangan handout ini agar lebih baik lagi.

Kab. Bandung, 23 April 2021.  
Penulis

# Petunjuk Penggunaan Handout

Untuk meningkatkan efektifitas pengajaran, berikut ini adalah saran penggunaan kepada tenaga pengajar dan mahasiswa dalam menggunakan handout ini:

## Bagi Tenaga Pengajar

1. Bagikan seluruh slide kepada para mahasiswa sebelum perkuliahan dimulai
2. Sampaikan materi perkuliahan satu slide untuk setiap minggunya
3. Untuk pengajaran online, gunakan perangkat lunak pencoret PDF seperti XODO PDF Annotator, atau XOURNAL yang bersifat *free (opensource)*
4. berikan Kuis setelah perkuliahan berdasarkan materi yang disampaikan.
5. 14 pertemuan dan 14 quiz perlu dilakukan untuk memastikan semua materi PSM tersampaikan dengan baik dan telah pula dipahami oleh mahasiswa.

## Bagi Para Mahasiswa

1. Peroleh handout lengkap perkuliahan dari dosen pengampu dan print serta jilid rapi handout tersebut
2. Baca slide satu atau dua hari sebelum perkuliahan dimulai
3. Kerjakan latihan atau contoh yang ada pada slide
4. Tandai bagian-bagian yang belum dipahami
5. Tanyakan bagian-bagian yang belum dipahami kepada dosen saat perkuliahan berlangsung
6. Buat soal-soal sendiri dan *share* ke teman-teman, dan bahas jawabannya dalam forum-forum belajar kelompok
7. Selama perkuliahan berlangsung, tambahkan catatan-catatan di sekeliling handout terkait informasi-informasi tambahan yang diberikan oleh dosen saat perkuliahan berlangsung.
8. Hadir pada setiap perkuliahan dan ikuti semua kuis.



# Daftar Isi

<b>I</b>	<b>AUDIO DIGITAL</b>	<b>9</b>
1	Pendahuluan	11
2	Review Pengolahan Sinyal Digital	23
3	Review ADC dan Sinyal Audio Digital	39
4	Spektrum Frekuensi Audio dan Windowing	59
5	Wavelet Pada Sinyal Audio	79
6	STFT dan Spektogram pada Sinyal Audio	103
<b>II</b>	<b>CITRA DIGITAL</b>	<b>123</b>
7	Perkembangan Citra Digital	125
8	Konsep Dasar Citra Digital	141
9	Filtering pada Citra Digital	163
10	Operasi Morfologi pada Citra Digital	189

<b>11 Wavelet pada Citra Digital</b>	<b>213</b>
<b>12 Ekstraksi Ciri dengan GLCM</b>	<b>235</b>
<b>III PENGANTAR VIDEO DIGITAL</b>	<b>255</b>
<b>13 Video Digital</b>	<b>257</b>

Bagian I

# AUDIO DIGITAL



Bab 1

# Pendahuluan

# Selamat Datang di Perkuliahan Pengolahan Sinyal Multimedia (PSM)

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa mengetahui tentang **tata tertib** dalam perkuliahan ini
- 2 Mahasiswa mengetahui tentang **tujuan** ingin dicapai dari MK PSM ini.
- 3 Mahasiswa memahami aspek **penilaian** MK
- 4 Mahasiswa mengetahui **materi-materi** bahasan dalam MK ini

## Tata tertib perkuliahan

- Perkuliahan dilaksanakan secara **full-online**
- Pada awal bergabung perkuliahan online, kamera **dinyalakan**.
- Saat materi mulai disampaikan, kamera dapat **dimatikan**.
- di akhir kuliah, saat presensi dilakukan, kamera **dinyalakan kembali**.

## Tata tertib perkuliahan

- Jadwal Perkuliahan: Setiap **Kamis** 08:30-11:30 **WIB**
- Pukul 08:30-11:30 akan dibagi menjadi 5 bagian:
  - **08:30-09:30** kuliah **bagian I**
  - **09:30-09:45** istirahat
  - **09:45-10:30** kuliah **bagian II**
  - **10:30-10:45** istirahat
  - **10:45-11:30** **presensi**, review, quiz (jika ada).

## Tambahan

- Jika **diperlukan**, diperkenankan makan dan minum selama perkuliahan
- Jika perlu meninggalkan kelas sebentar, sampaikan melalui fitur **chat** pada aplikasi
- Utamakan **kerja keras** dan **kejujuran**

# Tujuan Perkuliahan dan Kompetensi yang ingin dicapai

MK **PSM** ini bertujuan:

- 1 memberikan pengetahuan terkait **konsep dasar** dan **operasi** pada sinyal multimedia:
- 2 **Audio, Citra, dan Video**
- 3 Fokus pada: **Audio** dan **Citra** (karena Video = Audio+Stream of Images)

## Penilaian dan Indeks Hasil pembelajaran

### Komponen Penilaian

- 1 Kehadiran (7,5%)
- 2 Quiz, PR, Tugas (32,5%)
- 3 Ujian Kompre I (sebelum UTS) (30%).
- 4 Ujian Kompre II ( sebelum UAS) (30%)
- 5 Tidak ada UTS dan UAS, digantikan dengan ujian Kompre I dan Kompre II
- 6 Quiz dilakukan sebanyak 12 kali (Mulai minggu II), dan rata-rata **10 Quiz terbaik**.

Indeks Hasil Pembelajaran (**A**, **AB**, **B**, dst) : Mengikuti Standar Institusi

## Materi dan Rencana Perkuliahan

- ① Minggu I : Pendahuluan
- ② Minggu II : Konsep Dasar Audio
- ③ Minggu III : Operasi dasar Audio: Windowing dan filtering
- ④ Minggu IV : Transformasi Audio : DFT, DCT
- ⑤ Minggu V : Transformasi Audio : Wavelet
- ⑥ Minggu VI : Autokorelasi dan Mel Frequency Spectral
- ⑦ Minggu VII : Overview Citra digital

## Materi dan Rencana Perkuliahan - Lanjutan

- ⑧ Minggu VIII : Operasi dasar Citra: Kernel dan Filtering
- ⑨ Minggu IX : Operasi Morfologi pada Citra
- ⑩ Minggu X : Transformasi pada Citra: DFT dan DCT
- ⑪ Minggu XI : Transformasi pada Citra: Wavelet
- ⑫ Minggu XII : Ekstraksi ciri pada citra: GLCM
- ⑬ Minggu XIII : Overview Video I
- ⑭ Minggu XIV : Overview Video II

## Alat bantu

Perangkat Lunak:

- ➊ **Utama:** Octave (gunakan **Matlab**, jika punya license atau terpaksa/tidak ada alternatif lagi)
- ➋ **Sekunder:** Python

Perangkat Keras:

- ➊ PC atau
- ➋ Laptop
- ➌ Preferable specs: RAM 4 GB or **more**, **SSD** Harddisk, Processor **Core2Duo** or **newer**.



## Bab 2

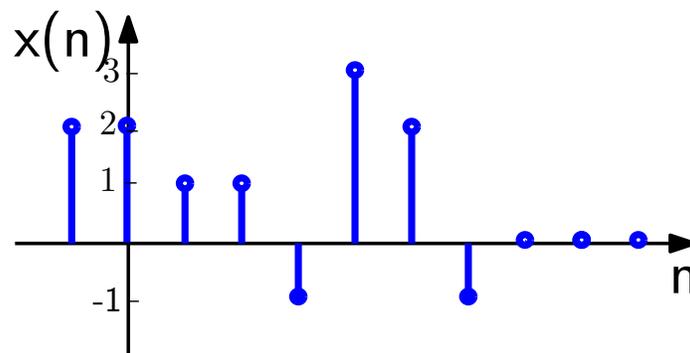
# Review Pengolahan Sinyal Digital

## Review Sinyal dan Sistem Linier

- 1 Slide ini berisi materi review dari sinyal dan sistem linier
- 2 Review ini diperlukan agar konsep-konsep dasar yang akan dipakai pada MK ini dapat diketahui oleh peserta didik
- 3 Terdapat dua aspek pada review ini yaitu aspek review sinyal dan aspek review sistem
- 4 Aspek review sinyal membahas tentang persamaan sinyal diskrit
- 5 Aspek review sistem meliputi definisi sistem Linier, Kausal dan Time Invariant, komponen dalam sistem LTI, serta representasi sistem

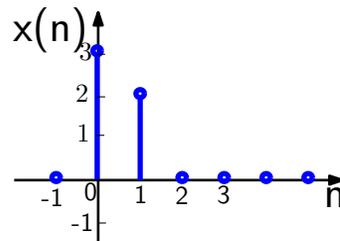
## Sinyal diskrit

- 1 Sinyal adalah besaran yang merupakan fungsi ruang dan atau waktu.
- 2 Sinyal dapat statis (tidak berubah terhadap fungsi waktu) dapat pula dinamis (berubah terhadap waktu)
- 3 Sinyal diskrit fungsi waktu adalah sinyal yang hanya ada pada waktu-waktu tertentu.
- 4 Contoh sinyal diskrit fungsi waktu (selanjutnya hanya disebut sebagai sinyal diskrit saja) adalah seperti gambar berikut:



## Persamaan Sinyal Diskrit

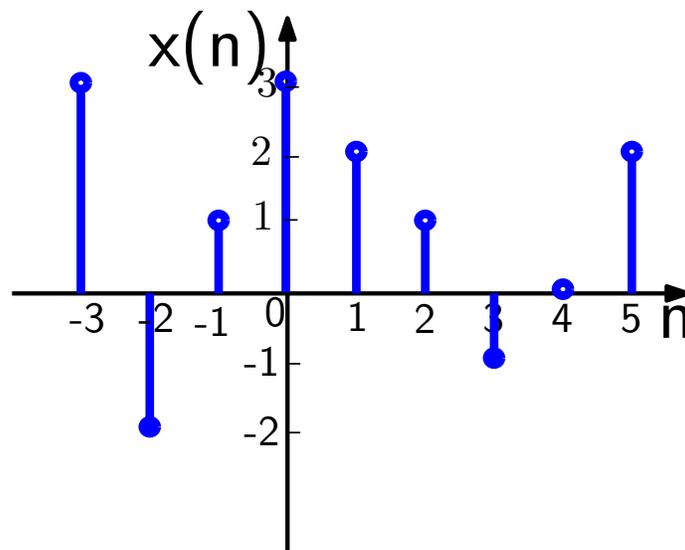
- 1 Persamaan sinyal diskrit dinyatakan dengan penjumlahan setiap elemennya
- 2 Untuk setiap elemen, persamaan dinyatakan dengan amplitudo dan waktu.
- 3 Perhatikan contoh sinyal dengan dua elemen berikut.



- 4 Elemen pertama **amplitudo 3** dan waktu di **0** →  
 $3\delta(n - 0) = 3\delta(n)$
- 5 Elemen kedua **amplitudo 2** dan waktu di **1** →  $2\delta(n - 1)$
- 6 Dengan demikian persamaan sinyal adalah:  
 $x(n) = 3\delta(n) + 2\delta(n - 1)$

## Latihan 1

- 1 Tuliskanlah persamaan sinyal berikut:



**Jawab:**

# Sistem

- 1 Sistem adalah perangkat pengolah sinyal
- 2 Terdapat sinyal masuk  $x(n)$
- 3 diolah oleh sistem
- 4 diperoleh sinyal keluaran  $y(n)$
- 5 Contoh ilustrasi sistem pengolah sinyal seperti gambar.

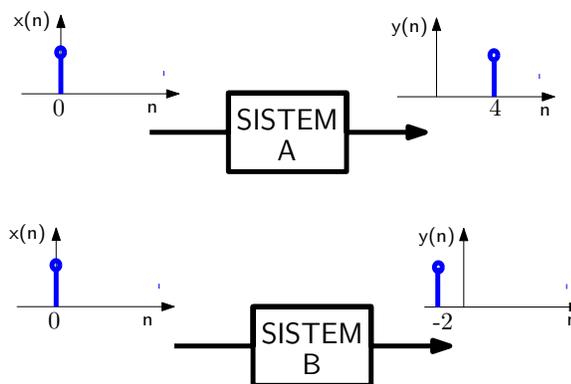


## Sistem

- ❶ Ada banyak sekali sistem pengolah sinyal
- ❷ Namun kita hanya tertarik pada subset dari sistem pengolah sinyal yang banyak tersebut.
- ❸ Subset sistem yang dipelajari adalah sistem yang bersifat:
- ❹ Kausal
- ❺ Linier
- ❻ Time Invariant
- ❼ Sistem Yang memenuhi ketiga sifat di atas disebut sistem yang Kausal-Linier-Time-Invariant (KLTI)
- ❽ Mari kita lihat definisi dari masing-masing sifat tersebut.

# Kausal

- 1 Sistem Kausal adalah sistem yang memiliki output yang tidak mendahului input secara waktu.
- 2 Analogi sistem kausal ini adalah sistem memasak air.
- 3 Jika kita memasak air jam 12:00, tentu saja air mendidih adalah setelah pukul 12:00 tersebut (tidak mungkin kurang dari itu)
- 4 Pada contoh ilustrasi di bawah, Sistem A adalah kausal dan sistem B adalah non-kausal. Jelaskan!



# Linier

- 1 Sistem Linier adalah sistem hubungan input-outputnya proporsional terhadap penskalaan amplitudo.
- 2 Contoh: Suatu sistem jika diberi input beramplitudo **2** mengeluarkan output beramplitudo **8**. Kemudian jika amplitudo input diduakalikan menjadi **4**, maka jika ternyata output juga menjadi dua kali yaitu **16**, maka sistem ini bersifat LINIER.
- 3 Secara matematis, sifat linier dapat ditulis sebagai:

Input	Output
$x_1(n)$	$\rightarrow y_1(n)$
$x_2(n)$	$\rightarrow y_2(n)$
$ax_1(n) + bx_2(n)$	$\rightarrow ay_1(n) + by_2(n)$
$a$ dan $b$ suatu konstanta	

## Time Invariant

- 1 Sistem Time Invariant (TI) adalah sistem yang memiliki proporsionalitas yang sama dalam skala waktu.
- 2 Analogi sederhana dari sistem TI adalah sistem memasak air.
- 3 Jika kita memasak air pukul 12:00 dan mendidih pukul 12:30, maka jika kita memasak air pukul 13:00 maka tentu ia akan mendidih pukul 13:30. Jadi proses bergeser mundur 1 jam, maka output juga bergeser mundur 1 jam.
- 4 Secara matematis, sifat Time Invariant dapat ditulis sebagai:

Input		Output
$x(n)$	→	$y(n)$
$x(n - p)$	→	$y(n - p)$

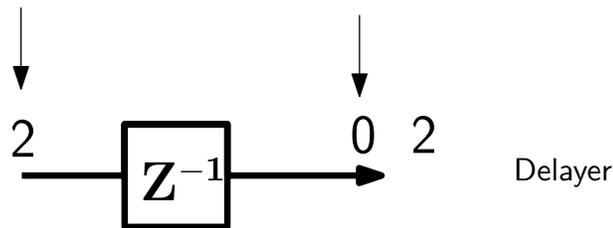
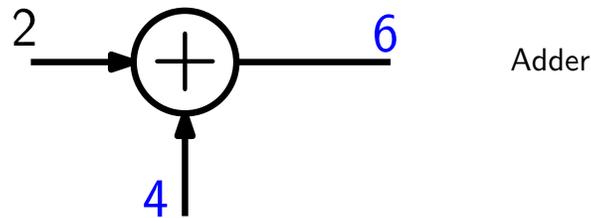
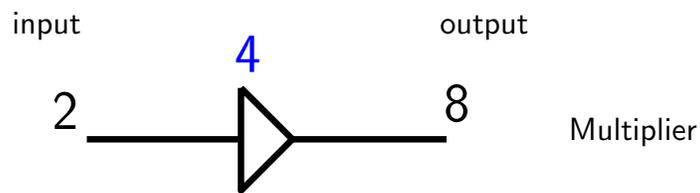
$p$  suatu konstanta

## Sistem Diskrit KLTl

- 1 Sistem Diskrit KLTl adalah sistem yang mengolah sinyal diskrit DAN
- 2 Bersifat Kausal-Linier-Time Invariant sekaligus
- 3 Komponen pada sistem KLTl ini terdiri dari **Adder**, **Delay**, **Multiplier**
- 4 masing-masing komponen ini bersifat KLTl
- 5 Sistem pengolah sinyal tersusun atas komponen-komponen ini dengan susunan tertentu agar memberikan sifat pengolahan yang diharapkan.
- 6 Sistem Diskrit KLTl ini juga sering disebut dengan **Filter**, karena ia memiliki response spesifik terhadap frekuensi dari sinyal input

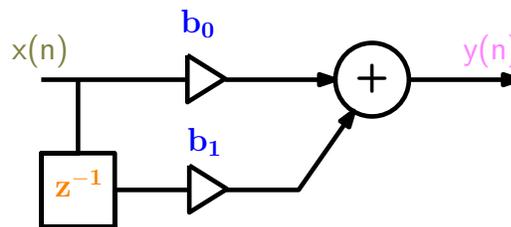
# Komponen Sistem Diskrit KLTl

Ilustrasi komponen Multiplier, Adder dan Delay



## Struktur Sistem Diskrit KLTl dan representasi

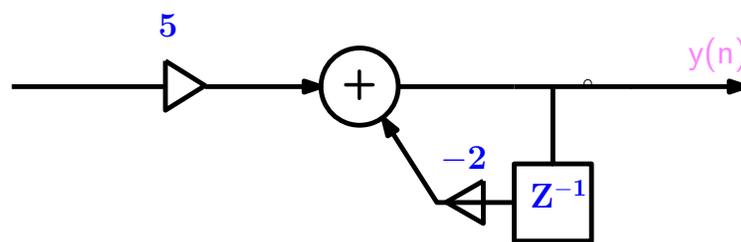
- 1 Tinjau contoh realisasi sistem KLTl berikut:



- 2 dengan contoh ini, maka sistem dapat dinyatakan dalam persamaan input dan output:  $y(n) = b_0x(n) + b_1x(n - 1)$
- 3 Jika kita lakukan transformasi Z dari persamaan input-output tersebut, maka kita peroleh:  $Y(z) = b_0X(z) + b_1z^{-1}X(z)$ , sehingga fungsi transfer sistem dapat ditulis:
 
$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1z^{-1}$$
- 4 Jika sistem kita beri input sinyal impuls:  $x(n) = \delta(n)$ , maka keluaran sistem adalah  $h(n) = y(n) = b_0\delta(n) + b_1\delta(n - 1)$

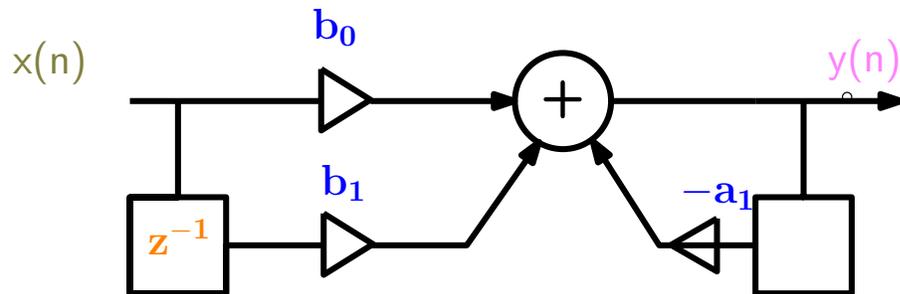
## Struktur Sistem Diskrit KLTl dan representasi

- 1 Pada contoh sebelumnya, terlihat bahwa 4 cara yang dapat digunakan untuk merepresentasikan sistem: **Realisasi diagram**, **Persamaan input-output**, **Fungsi Transfer** dan **Response Impulse**
- 2 Sebagai latihan, coba representasikan sistem berikut dalam : **Persamaan input-output**, **Fungsi Transfer** dan **Response Impulse**!



## Latihan Soal

- 1 Jelaskan bagaimana cara menyatakan persamaan sinyal?
- 2 Jelaskan apa yang dimaksud dengan **sistem Kausal, Linier, dan Time Invariant!**
- 3 Jelaskan tentang 3 komponen utama dari sistem KLTII!
- 4 Diberikan realisasi rangkaian berikut:



Tuliskan persamaan input-output, Fungsi transfer, dan response impuls dari sistem tersebut!



## Bab 3

# Review ADC dan Sinyal Audio Digital

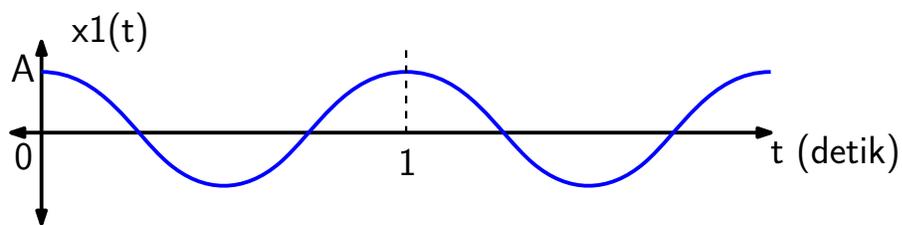
## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami konsep frekuensi pada sinyal Audio
- 2 Mahasiswa dapat memahami konsep windowing untuk memotong sinyal panjang dan menganalisis frekuensi setiap potongannya

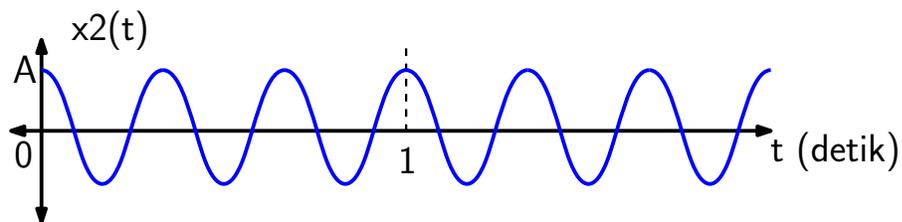
## Frekuensi analog

**Pada sinyal analog: Frekuensi** menyatakan banyaknya getaran yang terjadi pada setiap detik.

- 1 Sinyal frekuensi rendah



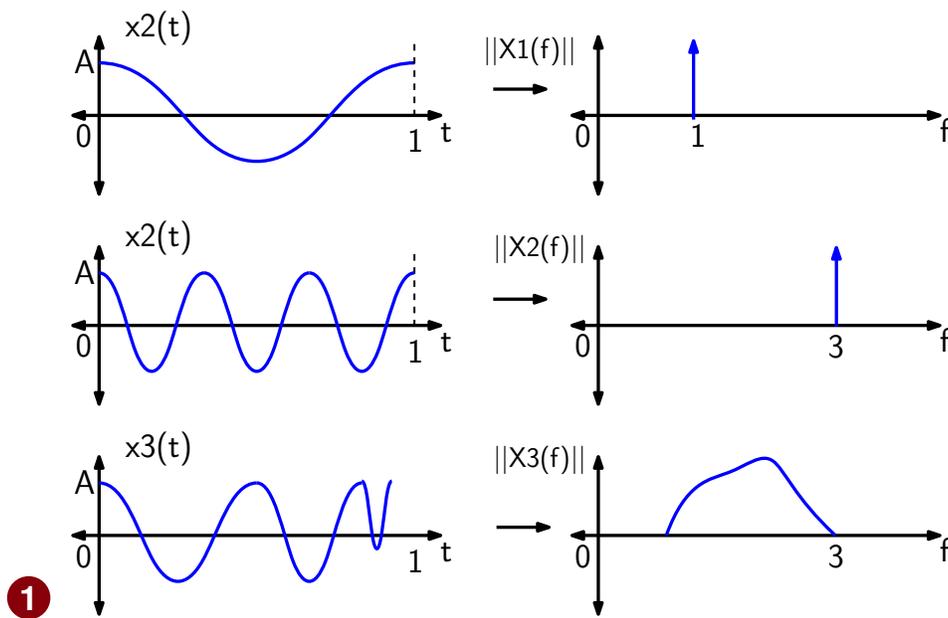
- 2 Sinyal dengan frekuensi tinggi



Tentukan frekuensi dan persamaan sinyal  $x_1(t)$  dan  $x_2(t)$ !

# Spektrum Frekuensi Analog

**Content Frekuensi** dari suatu gelombang dinyatakan dengan spektrum frekuensi.



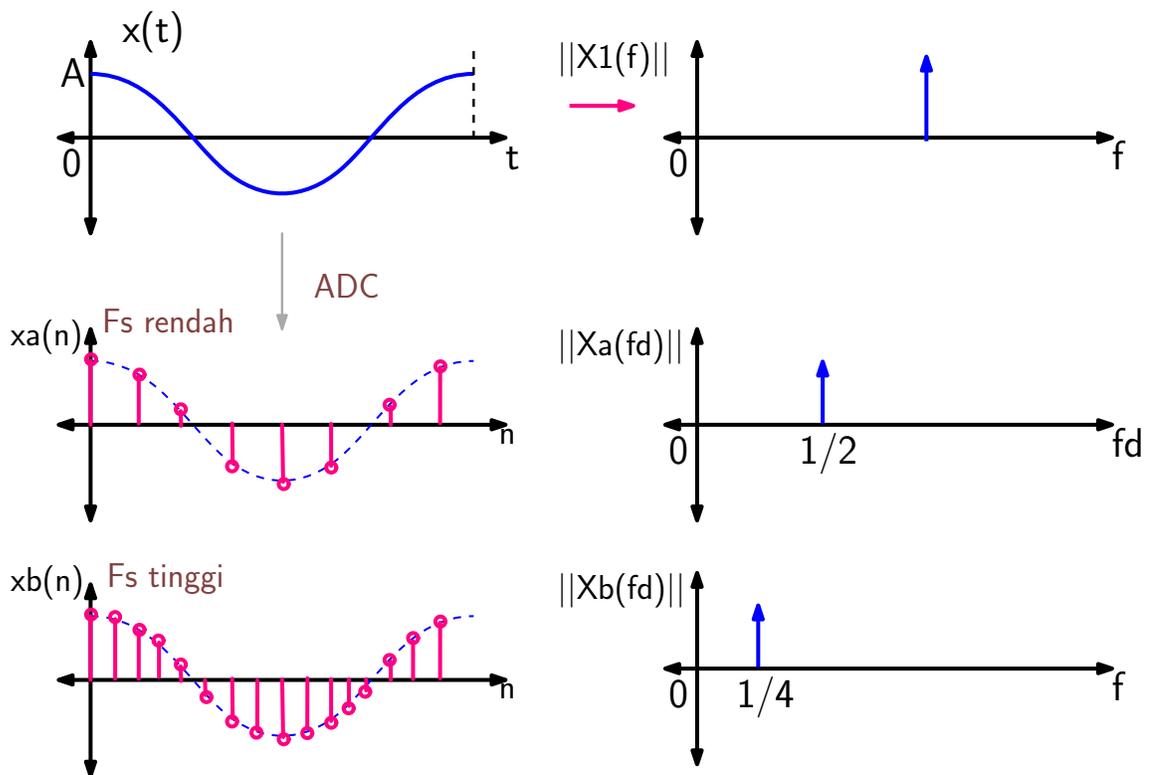
1

- 2 **Review:** Transformasi untuk melihat content frekuensi sinyal adalah ...
- 3 Alat untuk melihat content frekuensi sinyal adalah...

## Frekuensi Digital

- 1 Seperti yang telah dibahas pada slide sebelumnya, sinyal digital berasal dari sinyal analog dengan proses ADC
- 2 Dengan demikian, frekuensi frekuensi analog berubah menjadi frekuensi digital
- 3 Sinyal analog memiliki **frekuensi absolut**
- 4 Sinyal digital memiliki frekuensi **relatif** bergantung pada **Frekuensi Sampling** yang digunakan
- 5 Untuk sinyal analog yang sama, semakin tinggi frekuensi sampling, semakin rendah frekuensi digital yang dihasilkan!

# Frekuensi Digital



1

2 Mengingat Teorema Sampling Nyquist, maka **frekuensi tertinggi digital 1/2**

## Frekuensi Digital

- 1 Telah disampaikan bahwa frekuensi digital tertinggi adalah  $1/2$
- 2 Misal sinyal analog berfrekuensi  $f$ :

$$x(t) = A \cos 2\pi ft$$

- 3 Sinyal disampling dengan frekuensi sampling  $F_s$  ( $T_s = \frac{1}{F_s}$ )
- 4 Sinyal hasil sampling:

$$x(n) = A \cos 2\pi f(nT_s) = A \cos 2\pi f\left(\frac{n}{F_s}\right) = A \cos 2\pi \frac{f}{F_s} n$$

- 5 Atau:

$$x(n) = A \cos 2\pi f_d n$$

- 6 Dengan:  $f_d = \frac{f}{F_s}$  adalah frekuensi digital
- 7 Oleh karena  $F_s$  terkecil adalah  $2f_{max}$ , maka  $f_d$  terbesar adalah  $1/2$

## Contoh

Suatu sinyal Audio analog memiliki persamaan  $x(t) = 3 \cos 200\pi t$ .

**Tentukan:**

- ① frekuensi digitalnya jika frekuensi sampling yang digunakan adalah 200 sampel per detik adalah ...
- ② frekuensi digitalnya jika frekuensi sampling yang digunakan adalah 400 sampel per detik adalah ...
- ③ frekuensi digitalnya jika frekuensi sampling yang digunakan adalah 800 sampel per detik adalah ...
- ④ Bagaimana jika digunakan frekuensi sampling 100 sampel per detik?

## DFT dan FFT

- 1 Content frekuensi sinyal digital dapat diperoleh dari Discrete Fourier Transform
- 2 DFT N-titik dirumuskan:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$$

- 3 Nilai  $X(k)$  biasanya bernilai kompleks menunjukkan koefisien Fourier
- 4 DFT pada implementasinya dihitung dengan teknik Fast Fourier Transform (dengan butterfly diagram)
- 5 Nilai magnitudo FFT N titik diplot dari **k=0** sampai **k=N-1**

## Contoh Latihan

Diberikan sinyal  $\mathbf{x(n)}=[2 \ 3 \ 1 \ 1]$ . Tentukan koefisien DFT 2 titik dan 4 titik dari  $x(n)$ . **Note:** DFT 2 titik dapat dipandang sebagai DFT dari 2 komponen awal dari  $x(n)$

**Jawab:**

1 ...

## Octave Session

**Persiapan:** Jika belum dilakukan latihan pada Slide sebelumnya, lakukan kembali:

- 1 Sumber file WAV :  
<http://www0.cs.ucl.ac.uk/teaching/GZ05/samples/>
- 2 Buat folder PSM di komputer kita
- 3 Download file **The original 48KHz sampled file** dan situs di atas.
- 4 Simpan dengan nama file **tone.wav**

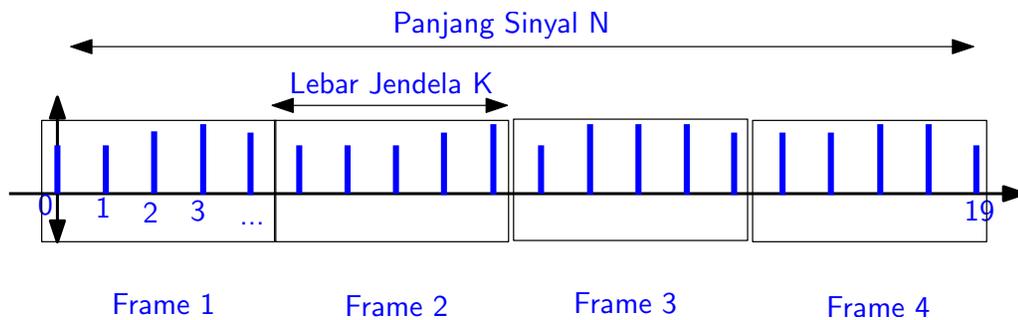
## Octave Session

### Percobaan:

- ① buka file **tone.wav** menggunakan perintah **audioread**:  
`[x fs nbits] = audioread('tone.wav')`
- ② Gunakan  $2 \times F_s$  sebagai jumlah titik pada DFT/FFT dan lakukan FFT pada sinyal **x**  
`Xfrek = fft(x, 2*fs)`
- ③ Nilai absolut **Xfrek** diplot dengan perintah **stem**:  
`figure(1)`  
`stem(abs(Xfrek))`
- ④ Sumbu datar dapat diperbaiki untuk memberikan informasi frekuensi  
`xaxis = [0:1:2*fs]`  
`figure(2)`  
`stem(xaxis,abs(Xfrek))`

## Windowing

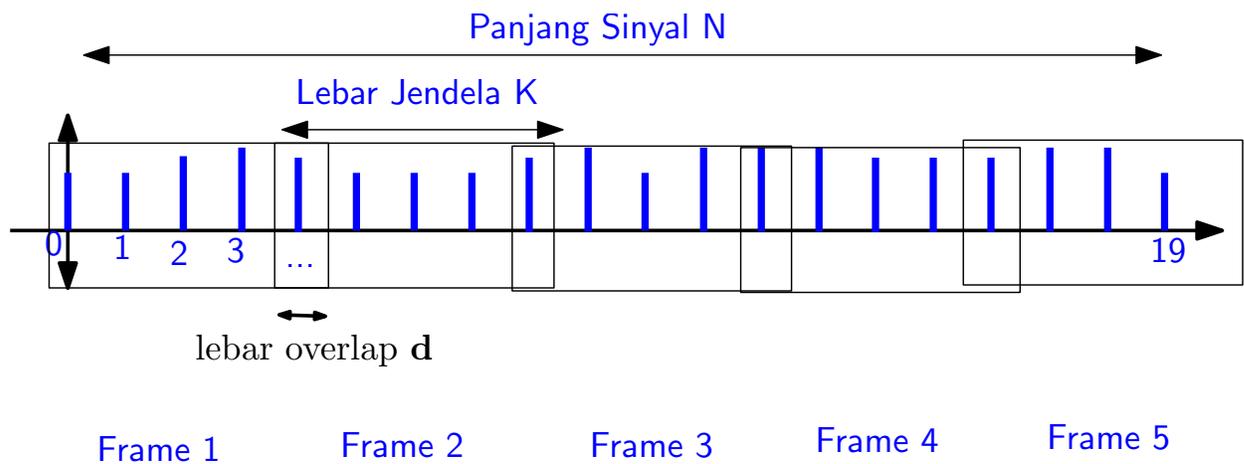
- 1 Untuk sinyal yang terlalu panjang, maka sinyal dapat dipotong-potong menjadi potongan yang lebih pendek selebar **K**
- 2 **Contoh:** Sinyal  $x(n)$  dengan panjang **N = 20** sampel
- 3 dapat dipotong menjadi **4 bagian** dengan panjang masing-masing **5 sampel**



- 4 Pembagian sinyal menjadi frame-frame ini disebut **Windowing**

## Window Overlap

- 1 Window dapat pula dibuat **overlap**
- 2 Contoh: Sinyal sepanjang 20 sampel, diwindow dengan lebar 5 sampel dan overlap 1 sampel
- 3 Hasil windowing adalah sbb:



- 4 Jika panjang sinyal adalah  $N$ , lebar jendela adalah  $K$ , dan lebar overlap adalah  $d$ , maka jumlah frame  $n_f = \dots \dots$

## Alternatif Window

- 1 Disamping bentuk window persegi seperti contoh sebelumnya, terdapat alternatif window lain:
- 2 Hamming
- 3 Hanning
- 4 ...

## Octave Session II

### Percobaan

- ① Gunakan file **tone.wav** yang telah diunduh sebelumnya.
- ② buka file **tone.wav** menggunakan perintah **audioread**:  
`[x fs nbits] = audioread('tone.wav')`
- ③ Potong audio menjadi 2 frame dengan lebar frame adalah panjang sinyal / 2  
`K = floor(length(x)/2)`  
`x1 = x(1:K)`  
`x2 = x(K+1:end)`
- ④ Gunakan  $2 \times F_s$  sebagai jumlah titik pada DFT/FFT dan lakukan FFT pada sinyal **x1** dan **x2**  
`Xfrek1 = fft(x1, 2*fs)`  
`Xfrek2 = fft(x2, 2*fs)`

## Octave Session II

Lanjutan

- ⑤ Nilai absolut **Xfrek1** dan **Xfrek2** diplot dengan perintah **stem**:  
`xaxis = [0:1:2*fs]`  
`figure(1)`  
`stem(xaxis,abs(Xfrek1))`  
`figure(2)`  
`stem(xaxis,abs(Xfrek2))`
- ⑥ Bandingkan antara plot spektrum frekuensi **x1** dan **x2**!

## Latihan OCTAVE

- 1 Baca kembali file audio **tone.wav**
- 2 Sekarang potong sinyal dalam 8 frame tanpa overlap dengan lebar frame adalah  $F_s$ .
- 3 Plot kembali spektrum frekuensi setiap frame!

## Pertanyaan Rekapitulasi

- 1 Informasi frekuensi sinyal ditampilkan dalam ...
- 2 Alat untuk mengamati spektrum sinyal disebut ...
- 3 Transformasi yang digunakan untuk memperoleh spektrum frekuensi sinyal adalah ...
- 4 Frekuensi tertinggi sinyal digital adalah ...
- 5 Teknik tercepat untuk menghitung transformasi fourier diskrit sinyal adalah menggunakan ...
- 6 Sinyal dengan lebar 1000 dilakukan windowing dengan lebar 50 tanpa overlap, maka jumlah frame yang dihasilkan adalah ...
- 7 Sinyal dengan lebar 1000 dilakukan windowing dengan lebar 50 dan overlap 10, maka jumlah frame yang dihasilkan adalah ...



## Bab 4

# Spektrum Frekuensi Audio dan Windowing

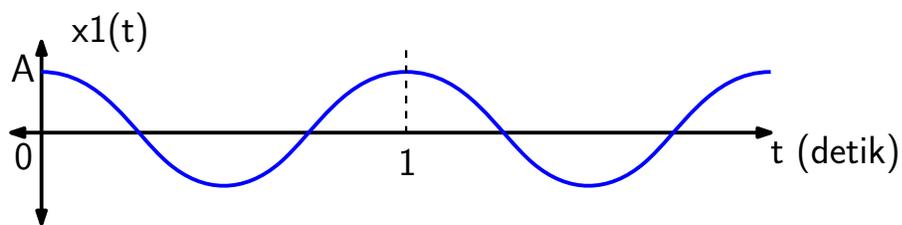
## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami konsep frekuensi pada sinyal Audio
- 2 Mahasiswa dapat memahami konsep windowing untuk memotong sinyal panjang dan menganalisis frekuensi setiap potongannya

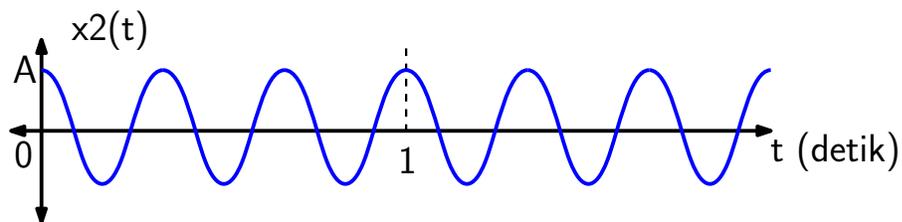
## Frekuensi analog

**Pada sinyal analog: Frekuensi** menyatakan banyaknya getaran yang terjadi pada setiap detik.

- 1 Sinyal frekuensi rendah



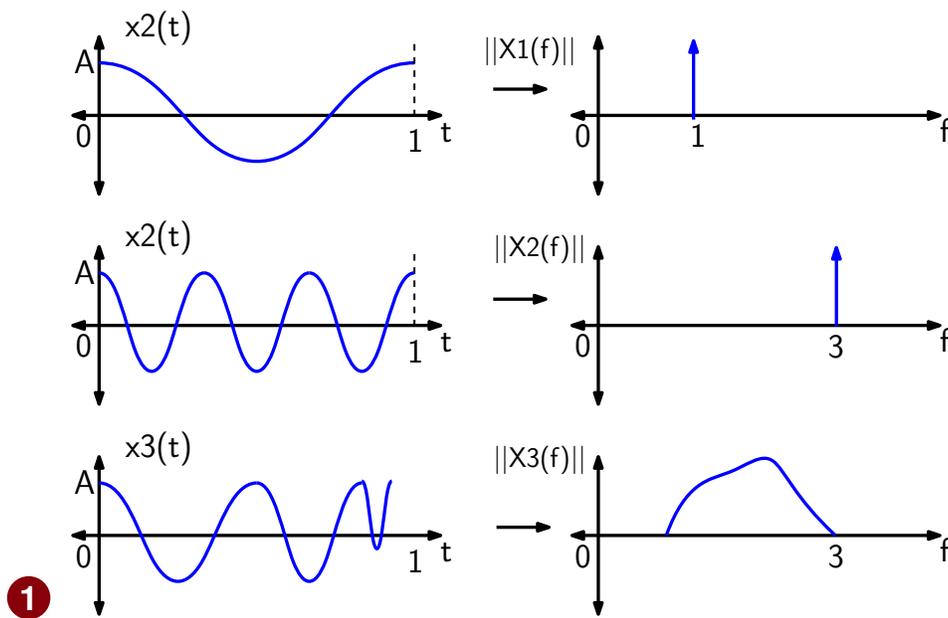
- 2 Sinyal dengan frekuensi tinggi



Tentukan frekuensi dan persamaan sinyal  $x_1(t)$  dan  $x_2(t)$ !

# Spektrum Frekuensi Analog

**Content Frekuensi** dari suatu gelombang dinyatakan dengan spektrum frekuensi.

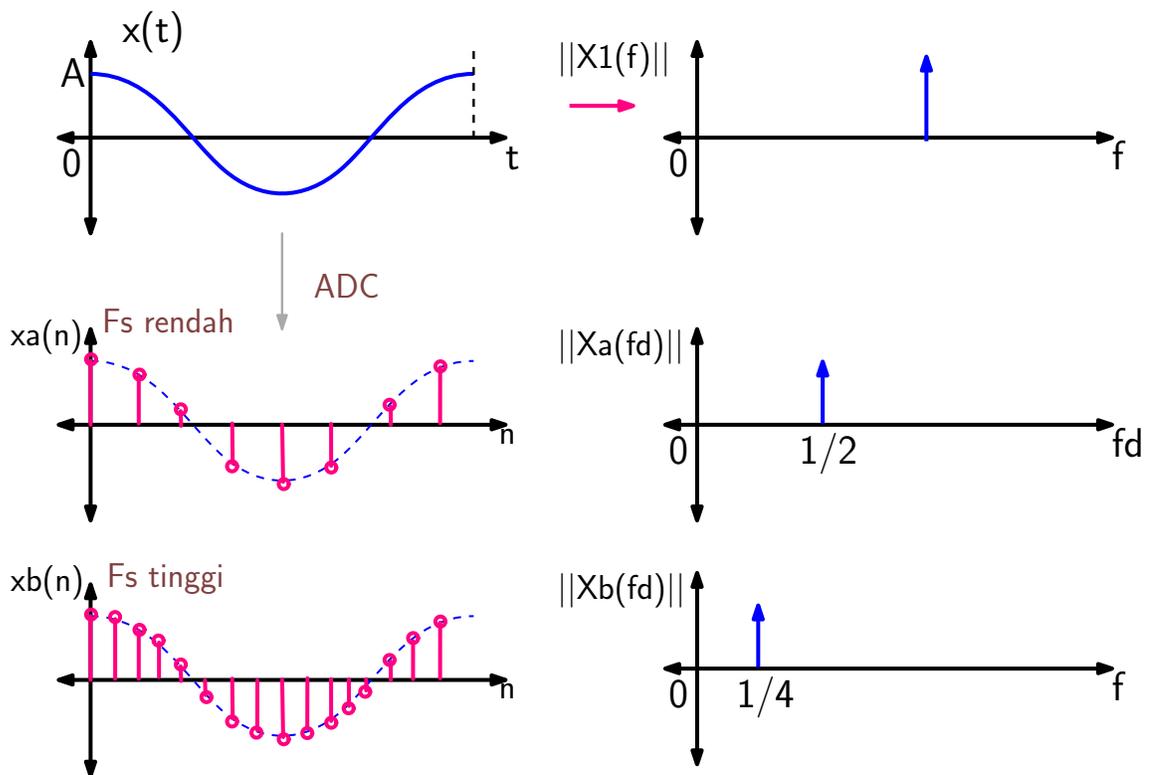


- 2 **Review:** Transformasi untuk melihat content frekuensi sinyal adalah ...
- 3 Alat untuk melihat content frekuensi sinyal adalah...

## Frekuensi Digital

- 1 Seperti yang telah dibahas pada slide sebelumnya, sinyal digital berasal dari sinyal analog dengan proses ADC
- 2 Dengan demikian, frekuensi frekuensi analog berubah menjadi frekuensi digital
- 3 Sinyal analog memiliki **frekuensi absolut**
- 4 Sinyal digital memiliki frekuensi **relatif** bergantung pada **Frekuensi Sampling** yang digunakan
- 5 Untuk sinyal analog yang sama, semakin tinggi frekuensi sampling, semakin rendah frekuensi digital yang dihasilkan!

# Frekuensi Digital



1

2 Mengingat Teorema Sampling Nyquist, maka **frekuensi tertinggi digital 1/2**

## Frekuensi Digital

- 1 Telah disampaikan bahwa frekuensi digital tertinggi adalah  $1/2$
- 2 Misal sinyal analog berfrekuensi  $f$ :

$$x(t) = A \cos 2\pi ft$$

- 3 Sinyal disampling dengan frekuensi sampling  $F_s$  ( $T_s = \frac{1}{F_s}$ )
- 4 Sinyal hasil sampling:

$$x(n) = A \cos 2\pi f(nT_s) = A \cos 2\pi f\left(\frac{n}{F_s}\right) = A \cos 2\pi \frac{f}{F_s} n$$

- 5 Atau:

$$x(n) = A \cos 2\pi f_d n$$

- 6 Dengan:  $f_d = \frac{f}{F_s}$  adalah frekuensi digital
- 7 Oleh karena  $F_s$  terkecil adalah  $2f_{max}$ , maka  $f_d$  terbesar adalah  $1/2$

## Contoh

Suatu sinyal Audio analog memiliki persamaan  $x(t) = 3 \cos 200\pi t$ .

**Tentukan:**

- ① frekuensi digitalnya jika frekuensi sampling yang digunakan adalah 200 sampel per detik adalah ...
- ② frekuensi digitalnya jika frekuensi sampling yang digunakan adalah 400 sampel per detik adalah ...
- ③ frekuensi digitalnya jika frekuensi sampling yang digunakan adalah 800 sampel per detik adalah ...
- ④ Bagaimana jika digunakan frekuensi sampling 100 sampel per detik?

## DFT dan FFT

- 1 Content frekuensi sinyal digital dapat diperoleh dari Discrete Fourier Transform
- 2 DFT N-titik dirumuskan:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$$

- 3 Nilai  $X(k)$  biasanya bernilai kompleks menunjukkan koefisien Fourier
- 4 DFT pada implementasinya dihitung dengan teknik Fast Fourier Transform (dengan butterfly diagram)
- 5 Nilai magnitudo FFT N titik diplot dari **k=0** sampai **k=N-1**

## Contoh Latihan

Diberikan sinyal  $\mathbf{x(n)}=[2 \ 3 \ 1 \ 1]$ . Tentukan koefisien DFT 2 titik dan 4 titik dari  $x(n)$ . **Note:** DFT 2 titik dapat dipandang sebagai DFT dari 2 komponen awal dari  $x(n)$

**Jawab:**

1 ...

## Octave Session

**Persiapan:** Jika belum dilakukan latihan pada Slide sebelumnya, lakukan kembali:

- 1 Sumber file WAV :  
<http://www0.cs.ucl.ac.uk/teaching/GZ05/samples/>
- 2 Buat folder PSM di komputer kita
- 3 Download file **The original 48KHz sampled file** dan situs di atas.
- 4 Simpan dengan nama file **tone.wav**

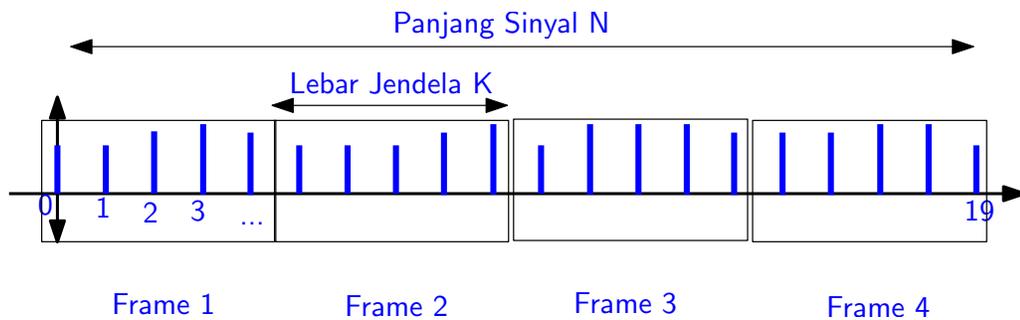
## Octave Session

### Percobaan:

- ① buka file **tone.wav** menggunakan perintah **audioread**:  
`[x fs nbits] = audioread('tone.wav')`
- ② Gunakan  $2 \times F_s$  sebagai jumlah titik pada DFT/FFT dan lakukan FFT pada sinyal **x**  
`Xfrek = fft(x, 2*fs)`
- ③ Nilai absolut **Xfrek** diplot dengan perintah **stem**:  
`figure(1)`  
`stem(abs(Xfrek))`
- ④ Sumbu datar dapat diperbaiki untuk memberikan informasi frekuensi  
`xaxis = [0:1:2*fs]`  
`figure(2)`  
`stem(xaxis,abs(Xfrek))`

# Windowing

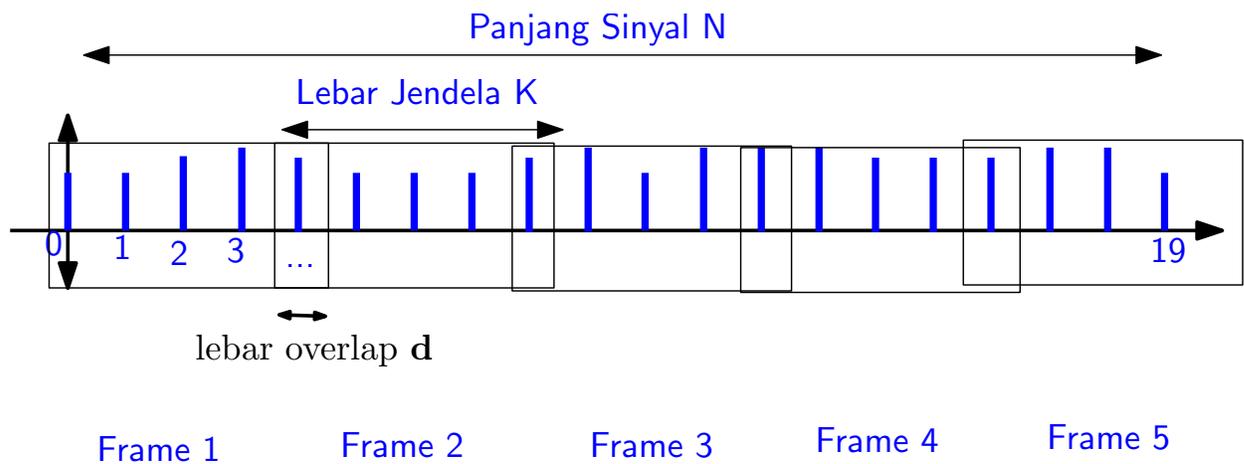
- ① Untuk sinyal yang terlalu panjang, maka sinyal dapat dipotong-potong menjadi potongan yang lebih pendek selebar **K**
- ② **Contoh:** Sinyal  $x(n)$  dengan panjang **N = 20** sampel
- ③ dapat dipotong menjadi **4 bagian** dengan panjang masing-masing **5 sampel**



- ④ Pembagian sinyal menjadi frame-frame ini disebut **Windowing**

## Window Overlap

- 1 Window dapat pula dibuat **overlap**
- 2 Contoh: Sinyal sepanjang 20 sampel, diwindow dengan lebar 5 sampel dan overlap 1 sampel
- 3 Hasil windowing adalah sbb:



- 4 Jika panjang sinyal adalah  $N$ , lebar jendela adalah  $K$ , dan lebar overlap adalah  $d$ , maka jumlah frame  $n_f = \dots$

## Alternatif Window

- 1 Disamping bentuk window persegi seperti contoh sebelumnya, terdapat alternatif window lain:
- 2 Hamming
- 3 Hanning
- 4 ...

## Octave Session II

### Percobaan

- ① Gunakan file **tone.wav** yang telah diunduh sebelumnya.
- ② buka file **tone.wav** menggunakan perintah **audioread**:  

$$[x \text{ fs } \text{nbits}] = \text{audioread}('tone.wav')$$
- ③ Potong audio menjadi 2 frame dengan lebar frame adalah panjang sinyal / 2  

$$K = \text{floor}(\text{length}(x)/2)$$

$$x1 = x(1:K)$$

$$x2 = x(K+1:\text{end})$$
- ④ Gunakan  $2 \times F_s$  sebagai jumlah titik pada DFT/FFT dan lakukan FFT pada sinyal **x1** dan **x2**  

$$X_{\text{frek1}} = \text{fft}(x1, 2*fs)$$

$$X_{\text{frek2}} = \text{fft}(x2, 2*fs)$$

## Octave Session II

Lanjutan

- ⑤ Nilai absolut **Xfrek1** dan **Xfrek2** diplot dengan perintah **stem**:  
`xaxis = [0:1:2*fs]`  
`figure(1)`  
`stem(xaxis,abs(Xfrek1))`  
`figure(2)`  
`stem(xaxis,abs(Xfrek2))`
- ⑥ Bandingkan antara plot spektrum frekuensi **x1** dan **x2**!

## Latihan OCTAVE

- 1 Baca kembali file audio **tone.wav**
- 2 Sekarang potong sinyal dalam 8 frame tanpa overlap dengan lebar frame adalah  $F_s$ .
- 3 Plot kembali spektrum frekuensi setiap frame!

## Pertanyaan Rekapitulasi

- 1 Informasi frekuensi sinyal ditampilkan dalam ...
- 2 Alat untuk mengamati spektrum sinyal disebut ...
- 3 Transformasi yang digunakan untuk memperoleh spektrum frekuensi sinyal adalah ...
- 4 Frekuensi tertinggi sinyal digital adalah ...
- 5 Teknik tercepat untuk menghitung transformasi fourier diskrit sinyal adalah menggunakan ...
- 6 Sinyal dengan lebar 1000 dilakukan windowing dengan lebar 50 tanpa overlap, maka jumlah frame yang dihasilkan adalah ...
- 7 Sinyal dengan lebar 1000 dilakukan windowing dengan lebar 50 dan overlap 10, maka jumlah frame yang dihasilkan adalah ...



## Bab 5

# Wavelet Pada Sinyal Audio

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami proses dekomposisi menggunakan teknik Wavelet
- 2 Mahasiswa dapat memahami proses rekonstruksi menggunakan teknik Wavelet

# Wavelet

**Wavelet** berarti small wave atau gelombang kecil.

- 1 Istilah wavelet mulanya dipopulerkan oleh Morlet dan Grossmann awal tahun 1980an.
- 2 Istilah yang digunakan oleh Morlet dan Grossmann adalah **Ondelette**
- 3 **Ondelette** adalah istilah Perancis untuk gelombang kecil atau Wavelet
- 4 Wavelet adalah teknik mendekomposisi sinyal ke dalam komponen-komponen **gelombang kecil**
- 5 Sejak 1980an Teknik Dekomposisi sinyal dengan Wavelet banyak diteliti
- 6 Antara lain oleh **Daubechies, Haar, Gabor**, dll.

# Bagian Wavelet

Wavelet terdiri dari 2 bagian:

1 Bagian Dekomposisi

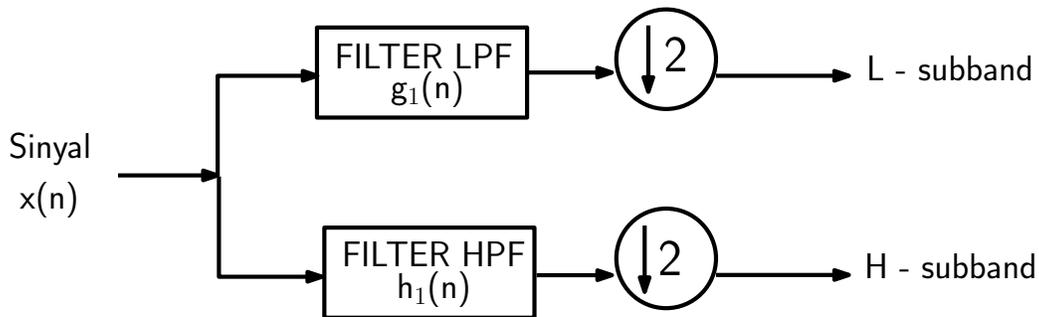


2 Bagian Rekonstruksi



## Struktur Dekomposisi Wavelet

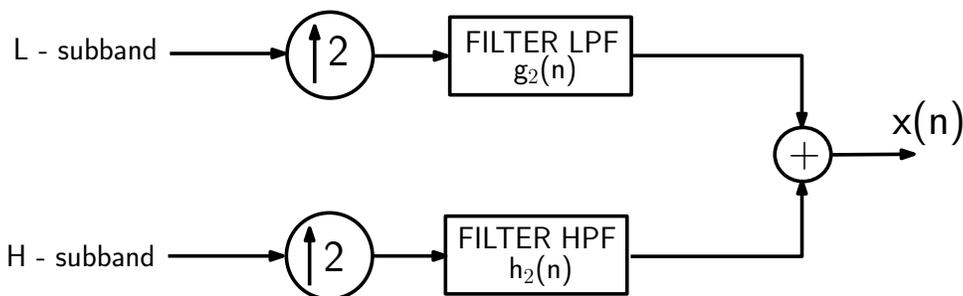
Struktur Dekomposisi Wavelet adalah sebagai berikut:



- ❶ Sinyal  $x(n)$  dialirkan ke dua jalur: Jalur **Subband L** dan **Subband H**
- ❷ Pada **Jalur L**, sinyal **difilter** dengan LPF  $g_1(n)$  kemudian di-**downsampling** dengan faktor 2.
- ❸ Pada **Jalur H**, sinyal **difilter** dengan HPF  $h_1(n)$  kemudian di-**downsampling** dengan faktor 2.

## Struktur Rekonstruksi Wavelet

Struktur Dekomposisi Wavelet adalah sebagai berikut:



- ❶ Sinyal **L** dilakukan up-sampling dengan faktor 2, kemudian difilter dengan LPF  $g_2(n)$
- ❷ Sinyal **H** dilakukan up-sampling dengan faktor 2, kemudian difilter dengan HPF  $h_2(n)$
- ❸ Sinyal keluaran kedua jalur dijumlahkan untuk memperoleh  $x(n)$  rekonstruksi.

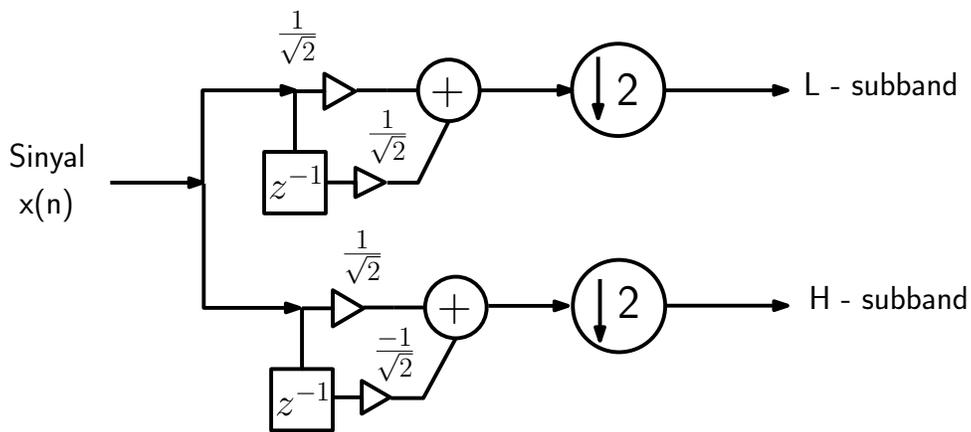
## Pasangan Filter $[g_1(n), h_1(n)]$ dan $[g_2(n), h_2(n)]$

- ➊ Pasangan filter  $[g_1(n), h_1(n)]$  harus ditentukan dengan hati-hati agar proses **rekonstruksi** dapat **sempurna**
- ➋ **Rekonstruksi sempurna** berarti sinyal hasil rekonstruksi adalah sama persis dengan sinyal semula
- ➌ Pasangan filter ini disebut juga dengan **Mother Wavelet** atau **wavelet induk**
- ➍ Beberapa peneliti berhasil menemukan pasangan filter ini: **Haar**, **Daubechies**, dll.
- ➎ Sehingga ada Mother **Wavelet Haar**, (Mother) **Wavelet Daubechies**, (Mother) **Wavelet Coiflets**, (Mother) **Wavelet Symlets**, ...

## Pasangan Filter $[g_1(n), h_1(n)]$ dan $[g_2(n), h_2(n)]$

- ❶ Contoh pasangan  $g_1(n)$  dan  $h_1(n)$  pada Wavelet Haar
- ❷  $g_1(n) = [\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}}]$
- ❸  $h_1(n) = [\frac{1}{\sqrt{2}} \quad -\frac{1}{\sqrt{2}}]$

Struktur Dekomposisi Wavelet Haar dengan demikian adalah:



## Hubungan $g_1(n)$ dengan $h_1(n)$

- ❶  $g_1(n)$  adalah response impulse filter low-pass
- ❷  $h_1(n)$  adalah response impuls filter high-pass yang menjadi pasangan  $g_1(n)$
- ❸  $h_1(n)$  dan  $h_1(n)$  memiliki hubungan yaitu:

$$h_1(n) = (-1)^{1-n}g(N - n - 1)$$

**N** panjang filter  
 $n=0,1,\dots,N-1$

## Hubungan $g_1(n)$ dengan $h_1(n)$

**Contoh 1:** Jika  $g_1(n) = [\frac{1}{\sqrt{2}} \quad \frac{1}{\sqrt{2}}]$ , tentukan  $h_1(n)$ !

**Jawab:**

## Hubungan $g_1(n)$ dengan $h_1(n)$

**Contoh 2:** Jika  $g_1(n) = [0, 48 \ 0, 84 \ 0, 22 \ -0, 13]$ ,  
tentukan  $h_1(n)$ !

**Jawab:**

## Latihan

Gambarkan struktur dekomposisi wavelet jika:

$$g_1(n) = [0,48 \ 0,84 \ 0,22 \ -0,13]$$

① Jawab:

## Pasangan filter $g_2(n)$ dan $h_2n$

- ① Setelah pasangan filter dekomposisi  $g_1(n)$  dan  $h_1(n)$  dipilih,
- ② maka pasangan filter rekonstruksi akan mengikuti, yaitu
- ③  $g_2(n) = g_1(N - n - 1)$
- ④  $h_2(n) = h_1(N - n - 1)$
- ⑤  $N$  adalah panjang filter
- ⑥  $n=0, 1, \dots, N-1$
- ⑦ Dengan kata lain,  $g_2(n)$  adalah pencerminan waktu dari  $g_1(n)$
- ⑧ dan  $h_2(n)$  adalah pencerminan waktu dari  $h_1(n)$

## Hubungan $g_2(n)$ dengan $h_2(n)$

**Contoh 3:** Jika  $\mathbf{g}_1(n) = [0, 48 \ 0, 84 \ 0, 22 \ -0, 13]$ ,  
tentukan  $h_1(n)$ ,  $g_2(n)$ , dan  $h_2(n)$ !

**Jawab:**

## Hubungan $g_2(n)$ dengan $h_2(n)$

**Contoh 4:** Gambarkan Struktur Dekomposisi dan Rekonstruksi dari Wavelet pada **Contoh 3!**

**Jawab:**

## Down Sampling

- ① **Down Sampling** adalah proses pengurangan sampel
- ② **Contoh:**
- ③ misalkan  $\mathbf{x}=[1\ 2\ 5\ 4\ 4\ 2]$
- ④ Downsampling dengan faktor 2 dari  $\mathbf{x}$  memberikan:
- ⑤  $\mathbf{x1}=[1\ 5\ 4]$  (pembuangan posisi ganjil), **atau**
- ⑥  $\mathbf{x2}=[2\ 4\ 2]$  (pembuangan posisi genap)

## Up Sampling

- ① Up Sampling adalah proses penambahan sampel
- ② Penambahan sampel pada proses **Up Sampling** biasanya dilakukan dengan zero-padding
- ③ **Contoh:**
- ④ misalkan  $\mathbf{x1}=[1\ 5\ 4]$
- ⑤ Up sampling dengan faktor 2 dari  $\mathbf{x1}$  memberikan:
- ⑥  $\mathbf{x1a}=[1\ 0\ 5\ 0\ 4\ 0]$  (penambahan posisi ganjil), **atau**
- ⑦  $\mathbf{x2}=[0\ 1\ 0\ 5\ 0\ 4]$  (penambahan posisi genap)
- ⑧ Pada proses Dekomposisi dan Rekonstruksi Wavelet, antara down-sampling dan up-sampling harus dilakukan pada posisi yang sama!

## Contoh 5: Down dan Up Sampling

- ① misalkan  $\mathbf{x}=[5\ 4\ 3\ 2\ 1\ 2]$
- ② Dilakukan downsampling dengan membuang sinyal pada posisi genap untuk menghasilkan  $\mathbf{x1}$ , kemudian  $\mathbf{x1}$  dilakukan up sampling pada posisi genap menghasilkan  $\mathbf{x1a}$
- ③ Tentukan  $\mathbf{x1}$  dan  $\mathbf{x1a}$ !
- ④ **Jawab:**

## Contoh 6: Wavelet Dekomposisi

- 1 misalkan  $\mathbf{x} = [2\ 4\ 1\ 4\ 5\ 3\ 3\ 3]$
- 2 akan dilakukan dekomposisi wavelet Haar pada  $\mathbf{x}$  tersebut, tentukan sinyal  $\mathbf{L}$  dan  $\mathbf{H}$ !
- 3 **Jawab:...**

## Contoh 7: Wavelet Rekonstruksi

- 1 Rekonstruksi kembali sinyal asli dari Sinyal **L** dan **H** yang diperoleh dari **Contoh 6!**
- 2 **Jawab:...**

## Dekomposisi Bertingkat

- 1 Pada prinsipnya, Dekomposisi Wavelet dapat dilakukan lagi setelah dekomposisi pertama selesai
- 2 Misal, Jika dekomposisi dilakukan dalam 2 tahap, maka, keluaran dari tahap 1 yaitu L dan H di dekomposisi lagi
- 3 Keluaran tahap ke dua yaitu LL, LH yakni hasil dekomposisi dari L
- 4 Keluaran lainnya yaitu HL dan HH yakni dekomposisi dari H
- 5 Demikian dapat dilakukan untuk tahap selanjutnya.
- 6 Panjang keluaran dari setiap tahap adalah setengah dari inputnya.

## Octave Session

### Percobaan I:

- ① Fungsi wavelet tersimpan pada library atau package yang bernama **lfat** (large time frequency analysis toolbox)
- ② Oleh karena itu aktifkan dulu package ini dengan perintah:  
**pkg load lfat**
- ③ Buatlah sinyal sebarang, misalnya:  $\mathbf{x} = [2\ 4\ 1\ 4\ 5\ 3\ 3\ 3]$
- ④ Lakukan dekomposisi Wavelet Haar alias daubechies 1 (db1)  
 $[\mathbf{LH}] = \mathbf{fwt}(\mathbf{x}, 'db1', 1)$
- ⑤ 1 menunjukkan level dekomposisi
- ⑥ Pisahkan **L** dan **H** dari **LH**; plotlah sinyal **L** dan sinyal **H**, diamati dari **L** dan **H** tsb?

**Catatan:** **fwt** adalah singkatan dari **fast wavelet transform**. **Matlab** memiliki perintah **dwt** untuk melakukan dekomposisi wavelet.

## Octave Session

### Percobaan II:

- 1 Rekonstruksi kembali subband L dan H pada Percobaan I untuk memperoleh kembali sinyal  $x$ .
- 2 `xrec = ifwt(LH, 'db1', 1, length(LH))`
- 3 1 menyatakan tingkat dekomposisi
- 4 `length(LH)` berarti kita ingin panjang sinyal hasil rekonstruksi sama panjang dengan panjang LH

## Octave Session

**Percobaan III:** Lakukan dekomposisi Wavelet pada sinyal audio

- 1 Download file **tone.wav** dari <http://www0.cs.ucl.ac.uk/teaching/GZ05/samples/>
- 2 Bacalah file **tone.wav** ini dengan menggunakan OCTAVE dan simpan ke dalam variabel **x**
- 3 Dekomposisilah **x** ke dalam **L** dan **H** dengan Wavelet Haar.

## Bab 6

# STFT dan Spektogram pada Sinyal Audio

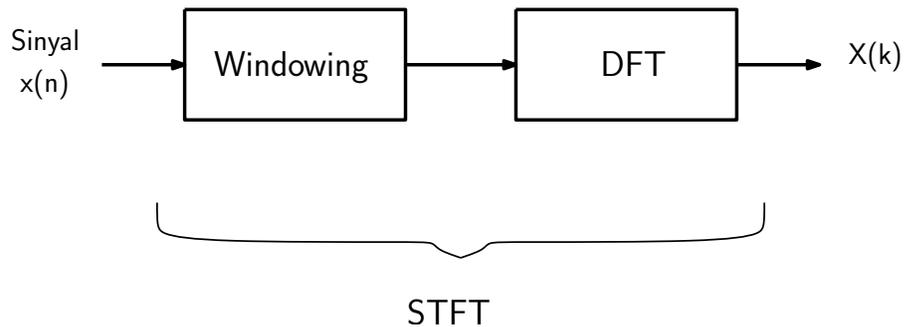
## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami Short Time Fourier Transform
- 2 Mahasiswa dapat memahami tentang Spektogram
- 3 Mahasiswa dapat memahami cara mengambil ciri dengan teknik STFT

# STFT

**STFT** adalah singkatan dari **Short Time Fourier Transform**.

- 1) **STFT** terdiri dari 2 tahapan yaitu: 1). **Windowing** dan 2) **DFT**
- 2) Skema dari **STFT** adalah sebagai berikut:

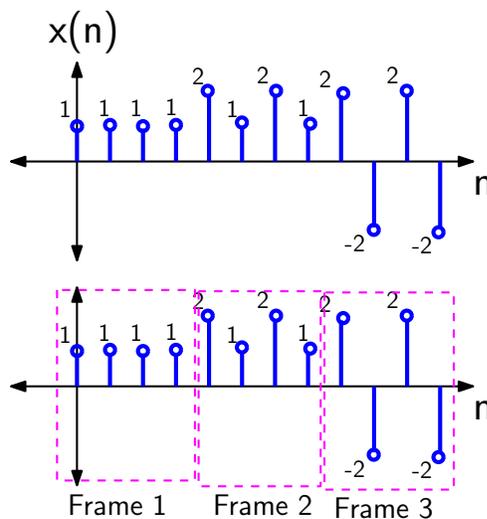


- 3) sebagai ilustrasi, perhatikan contoh berikut:

# STFT

**Contoh:** Diketahui sinyal:  $x(n)=[1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ -2 \ 2 \ -2]$

- ❶ Sinyal tsb diwindowing dengan window natural, dan lebar window 4 tanpa overlapping,
- ❷ Tentukan koefisien STFT dari sinyal dengan ketentuan windowing tsb. **Jawab:**
- ❸ Pertama kita gambarkan dulu sinyal tersebut setelah diwindowing:



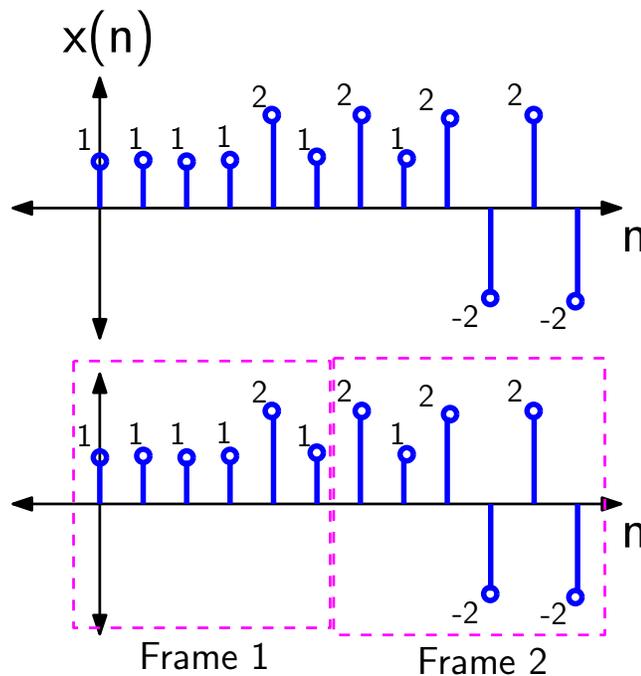
## STFT

- 1 Selanjutnya, setiap frame dilakukan DFT.
- 2 Untuk keperluan DFT ini, kita dapat menggunakan OCTAVE dengan perintah **fft**
- 3 Frame 1:  $x_1(n) = [1 \ 1 \ 1 \ 1] \Rightarrow X_1(k) = [4 \ 0 \ 0 \ 0]$
- 4 Frame 2:  $x_2(n) = [2 \ 1 \ 2 \ 1] \Rightarrow X_2(k) = [6 \ 0 \ 2 \ 0]$
- 5 Frame 3:  $x_3(n) = [2 \ -2 \ 2 \ -2] \Rightarrow X_3(k) = [0 \ 0 \ 8 \ 0]$
- 6  $X_1(k)$ ,  $X_2(k)$ ,  $X_3(k)$  ini menjadi koefisien STFT dari  $x(n)$  dengan lebar window 4 dan tanpa overlapping

## STFT

Mari kita ulangi lagi STFT dari  $x(n)=[1\ 1\ 1\ 1\ 2\ 1\ 2\ 1\ 2\ 2\ -2\ 2\ -2]$ , namun kali ini kita gunakan lebar window **6** dan **tanpa overlapping**.

① Gambar ilustrasi:



## STFT

- 1 Selanjutnya, setiap frame dilakukan DFT.
- 2 Untuk keperluan DFT ini, kita dapat menggunakan OCTAVE dengan perintah **fft**
- 3 Frame 1:  $x_1(n) = [1 \ 1 \ 1 \ 1 \ 2 \ 1] \Rightarrow X_1(k) = [7 \ -0.5 + 0.866i \ -0.5 - 0.866i \ 1 \ -0.5 + 0.866i \ -0.5 - 0.866i]$
- 4 Frame 2:  $x_2(n) = [2 \ 1 \ 2 \ -2 \ 2 \ -2] \Rightarrow X_2(k) = [3 \ 1.5 - 2.598i \ -1.5 - 2.598i \ 9 \ -1.5 + 2.598i \ 1.5 + 2.598i]$
- 5 sekarang,  $X_1(k)$ ,  $X_2(k)$  ini menjadi koefisien STFT dari  $x(n)$  dengan lebar window 6 dan tanpa overlapping

# STFT

## Catatan

- 1 Berbeda dengan DFT yang melakukan transformasi sepanjang sinyal,
- 2 STFT melakukan DFT untuk setiap potongan sinyal atau frame
- 3 Nilai koefisien STFT akan bergantung pada panjang window

## STFT-Latihan

Diberikan  $\mathbf{x(n)}=[1\ 1\ 1\ 1\ 2\ 1\ 2\ 1\ 2\ -2\ 2\ -2]$ , tentukan koefisien STFT dengan lebar window 4 sampel dan **overlapping** sebanyak 1 sampel.

① Jawab:

## Spektogram

- 1 **Spektogram** adalah penggambaran **amplituda** atau **nilai absolut STFT** dalam kurva **waktu-frekuensi**.
- 2 Ambil contoh sebelumnya, yakni:  $\mathbf{x(n)}=[1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ -2 \ 2 \ -2]$
- 3 Dengan panjang window 4 dan tanpa overlapping, maka kita peroleh koefisien STFT:
- 4 **Frame 1:**  $x_1(n) = [1 \ 1 \ 1 \ 1] \Rightarrow X_1(k) = [4 \ 0 \ 0 \ 0]$
- 5 **Frame 2:**  $x_2(n) = [2 \ 1 \ 2 \ 1] \Rightarrow X_2(k) = [6 \ 0 \ 2 \ 0]$
- 6 **Frame 3:**  $x_3(n) = [2 \ -2 \ 2 \ -2] \Rightarrow X_3(k) = [0 \ 0 \ 8 \ 0]$
- 7 nilai absolut  $X_1(k) \rightarrow |X_1(k)| = [4 \ 0 \ 0 \ 0]$
- 8 nilai absolut  $X_2(k) \rightarrow |X_2(k)| = [6 \ 0 \ 2 \ 0]$
- 9 nilai absolut  $X_3(k) \rightarrow |X_3(k)| = [0 \ 0 \ 8 \ 0]$

## Spektogram

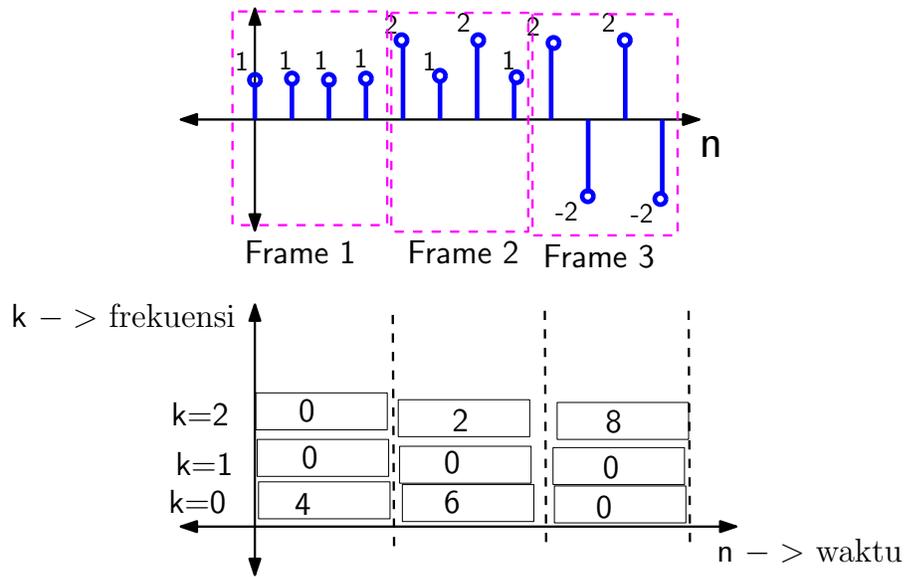
- 1 **Spektogram** adalah penggambaran **amplituda** atau **nilai absolut STFT** dalam kurva **waktu-frekuensi**.
- 2 Ambil contoh sebelumnya, yakni:  $\mathbf{x(n)}=[1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ -2 \ 2 \ -2]$
- 3 Dengan panjang window 4 dan tanpa overlapping, maka kita peroleh koefisien STFT:
- 4 **Frame 1:**  $x_1(n) = [1 \ 1 \ 1 \ 1] \Rightarrow X_1(k) = [4 \ 0 \ 0 \ 0]$
- 5 **Frame 2:**  $x_2(n) = [2 \ 1 \ 2 \ 1] \Rightarrow X_2(k) = [6 \ 0 \ 2 \ 0]$
- 6 **Frame 3:**  $x_3(n) = [2 \ -2 \ 2 \ -2] \Rightarrow X_3(k) = [0 \ 0 \ 8 \ 0]$
- 7 nilai absolut  $X_1(k) \rightarrow |X_1(k)| = [4 \ 0 \ 0 \ 0]$
- 8 nilai absolut  $X_2(k) \rightarrow |X_2(k)| = [6 \ 0 \ 2 \ 0]$
- 9 nilai absolut  $X_3(k) \rightarrow |X_3(k)| = [0 \ 0 \ 8 \ 0]$

## Spektogram

- 1 Sebelum kita memplot spektogram dari nilai absolut tadi, maka kita lakukan beberapa penyesuaian berikut:
- 2 Membuang elemen  $X(k)$  yang merupakan cermin dari yang lain
- 3  $|X_1(k)| = [4 \ 0 \ 0 \ 0]$  menjadi  $|X_1(k)| = [4 \ 0 \ 0]$
- 4  $|X_2(k)| = [6 \ 0 \ 2 \ 0]$  menjadi  $|X_2(k)| = [6 \ 0 \ 2]$
- 5  $|X_3(k)| = [0 \ 0 \ 8 \ 0]$  menjadi  $|X_3(k)| = [0 \ 0 \ 8]$

# Plot Spektogram

1 Plot spektogram:

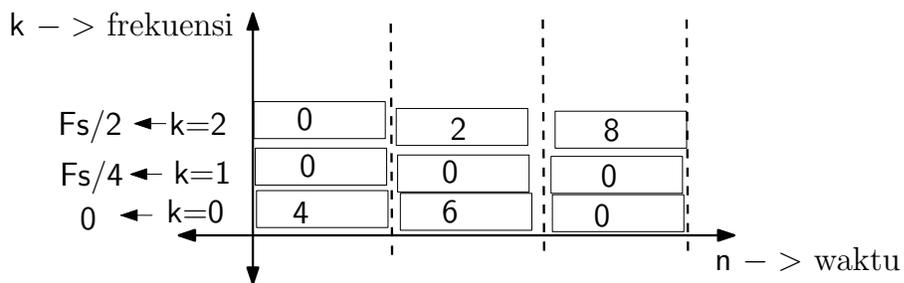
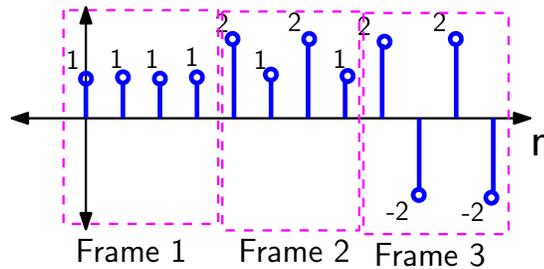


- 2 Spektogram memberikan kita informasi dengan cepat komponen frekuensi pada setiap frame
- 3 **Contoh:** Frame 1 memiliki frekuensi rendah
- 4 Frame 2 memiliki frekuensi rendah dan tinggi

# Plot Spektrogram

**Catatan:**

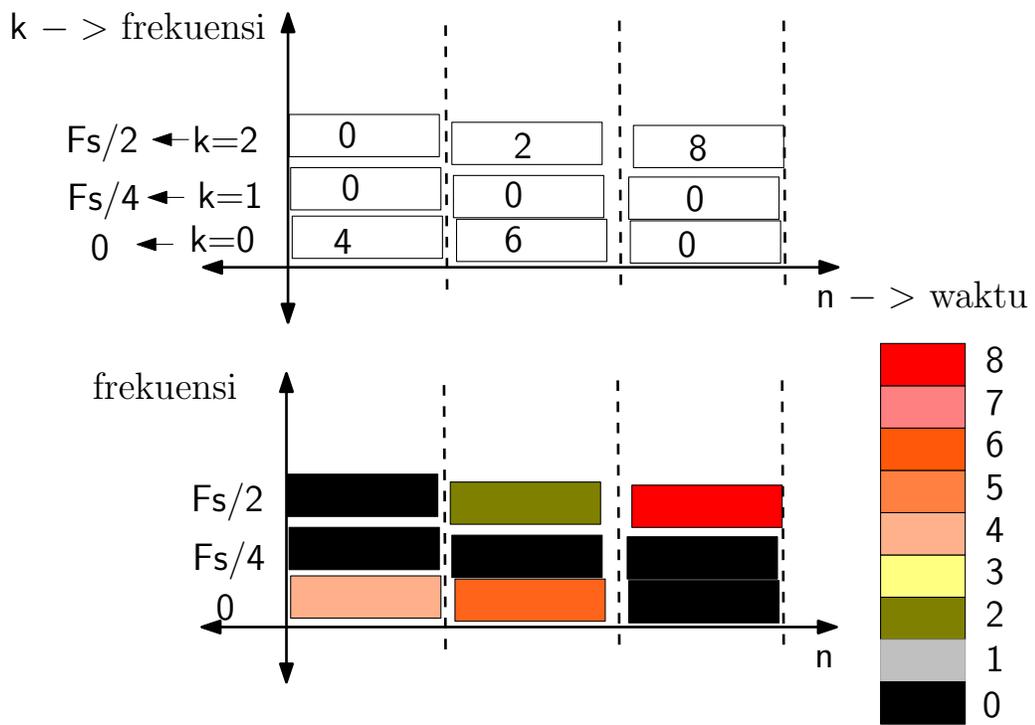
- ❶ Frekuensi tertinggi digital adalah  $1/2$ , berkorespondensi dengan frekuensi analog  **$F_s/2$**
- ❷ dengan informasi ini, kita dapat juga menambah informasi pada sumbu frekuensi pada spektrogram



# Plot Spektrogram

## Color representation

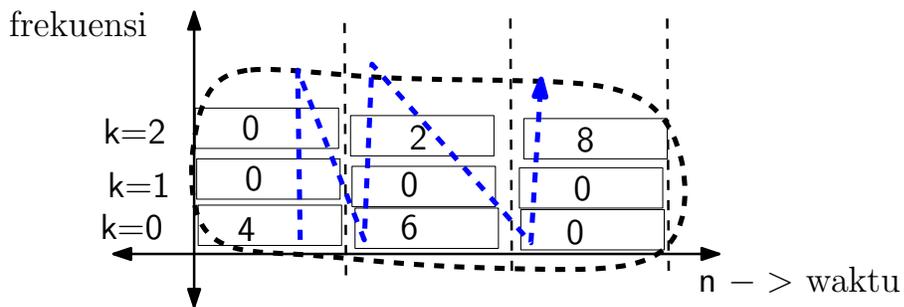
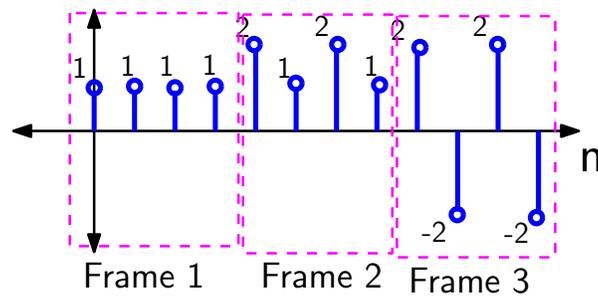
- 1 Alih-alih menggunakan angka, spektrogram lebih sering digambarkan dengan menggunakan warna



# Ekstraksi Ciri dengan STFT

STFT dapat digunakan untuk melakukan ekstraksi ciri dari suatu sinyal.

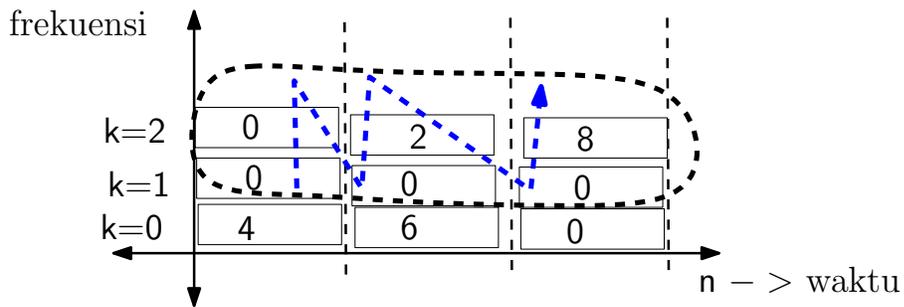
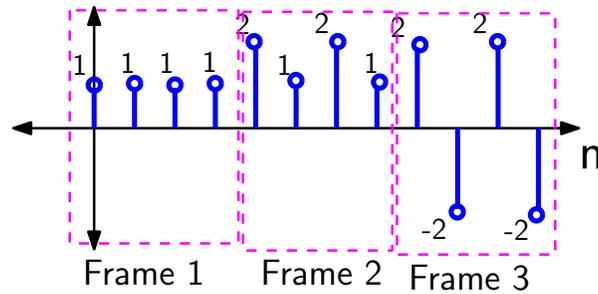
**1 Contoh:**



CIRI: [4 0 0 6 0 2 0 0 8]

# Ekstraksi Ciri dengan STFT

- 1 Sering komponen DC tidak masuk dalam ekstraksi ciri.  
**Contoh:**



CIRI: [0 0 0 2 0 8]

## LATIHAN 1

Ambil ciri sinyal  $x(n)=[1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ -2 \ 2 \ -2]$  dengan mengambil nilai absolut dari STFTnya, dengan lebar window **6** dan **tanpa overlapping**, jika:

- 1 komponen DC ikut diambil,
- 2 komponen DC tidak ikut diambil.
- 3 **Jawab:**

## LATIHAN 2

Ambil ciri sinyal  $x(n)=[1 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ -2 \ 2 \ -2]$  dengan mengambil nilai absolut dari STFTnya, dengan lebar window **4** dan **overlapping** 1 sampel, jika:

- ① komponen DC ikut diambil,
- ② komponen DC tidak ikut diambil.
- ③ **Jawab:**



**Bagian II**

**CITRA DIGITAL**



## Bab 7

# Perkembangan Citra Digital

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami perkembangan citra digital

## Citra Analog

- 1 Sebelum citra digital muncul, teknologi pencitraan dikuasai oleh citra analog
- 2 Citra analog utamanya dihasilkan oleh kamera analog dengan film analognya
- 3 Akuisisi citra analog memiliki perangkat utama yaitu: kamera analog + film



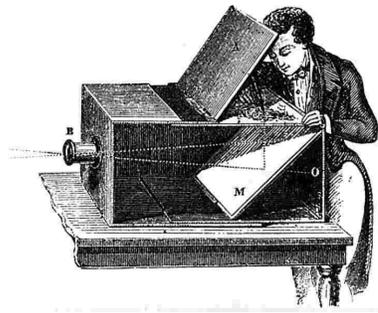
## Citra Analog

- 1 Penemuan Kamera Analog didahului dengan pengamatan Garam Perak bersifat berubah warna menjadi gelap jika terkena cahaya.
- 2 Fenomena di atas diamati secara teliti oleh ilmuan Jerman : Johann Heinrich Schulze, tahun 1717
- 3 Garam perak yang diamati oleh Schulze adalah Perak Nitrat ( $AgNO_3$ )
- 4 Schulze mengamati bahwa perubahan warna menjadi gelap pada Perak Nitrat terjadi jika ia terkena cahaya.



## Citra Analog

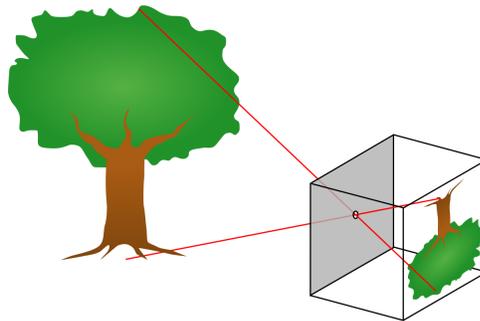
- 1 Kamera analog sendiri mulanya berupa perangkat proyeksi objek berupa **kamar gelap** (**Camera Obscura**, **Camera** = Kamar, **Obscura** = Gelap)



- 2 Tulisan awal tentang Camera Obscura adalah oleh **Mozi** (sekitar 470 - 391 SM).
- 3 Mozi menyatakan bahwa **citra** yang ditangkap di layar Camera Obscura adalah **terbalik**
- 4 Teori Camera Obscura ditulis lebih komprehensif oleh **Ibn Al-Haytham** tahun 1000an Masehi

## Citra Analog

- 1 **Ibn Al-Haytham** (dikenal juga dengan nama **Alhazen**) juga mengembangkan kamera **pin-hole**
- 2 Kamera pin-hole adalah penyederhanaan dari Camera Obscura
- 3 Kamera pin-hole tersusun atas box dengan **satu lubang** di salah satu dindingnya.
- 4 Image yang terproyeksi pada sisi dalam kamera pin-hole bersifat **terbalik**



**Figure :** Kamera Pin-Hole

## Citra Analog

- 1 Kombinasi antara kamera dengan film untuk menghasilkan foto analog dilakukan oleh **Joseph Niepce** tahun **1825 Masehi** (bersamaan Awal perang Diponegoro).
- 2 Niepce menggunakan campuran timah dan semacam aspal untuk filmnya
- 3 Pada mulainya Niepce memerlukan eksposur sebanyak 8 jam.
- 4 Proses ini disempurnakan oleh **Niepce** dengan bantuan **Louis Daguerre** untuk eksposur yang lebih singkat
- 5 Untuk keperluan ini, **Niepce** dan **Daguerre** menggunakan Ion Nitrat.



**Figure :** Joseph Niepce

## Citra Analog

- 1 Berikut ini contoh citra awal yang dibuat Niepce dari jendela kamarnya



**Figure :** Citra awal dari tahun 1825 oleh Niepce (**Sumber:** Wikipedia)

## Film Celluloid

- 1 Untuk meningkatkan kepraktisan film, para ilmuwan memikirkan film yang berbentuk sangat tipis dan dapat ditempel di dinding dalam kamera
- 2 Film Celluloid dikembangkan untuk keperluan ini
- 3 Film Celluloid dikembangkan oleh Alexander Parkes tahun 1855
- 4 Pemakaian film celluloid ini meluas dan secara praktis diadopsi oleh dunia fotografi secara umum.



Film Celluloid



Film Celluloid modern

## Citra Analog

- 1 Bisnis Fotografi Film Analog dapat dikatakan dipelopori oleh **George Eastman**, membuat kertas film tahun 1885
- 2 Kertas film digantikan oleh **seluloid** tahun in 1888–1889.
- 3 Dengan film seluloid George Eastman mengeluarkan kamera pertama : "**Kodak**" tahun 1888.
- 4 Kodak adalah simple box camera dengan lensa fokus tetap dan single shutter speed
- 5 Tahun 1900 Kodak disempurnakan dengan **Brownie**, sebuah simple portable kamera awal di dunia.



**Figure :** Kamera Brownie buatan George Eastman (**Sumber:** Wikipedia)

## Film Warna

- 1 Proses film warna dipelopori dengan teknik yang disebut **Three Color Process**
- 2 Teknik ini diusulkan oleh **James Clerk Maxwell** ilmuwan Skotlandia tahun 1855.
- 3 Teknik usulan Maxwell ini didasarkan bahwa mata manusia melihat warna sebagai kombinasi dari tiga warna: **Red**, **Green** dan **Blue** ⇒ **RGB**
- 4 **Thomas Sutton**, tahun **1861** berhasil membuat foto warna dengan prinsip **Maxwell** ini.



**Figure :** Foto warna awal oleh **Thomas Sutton** tahun 1861 berdasarkan prinsip **Maxwell** (Sumber: Wikipedia)

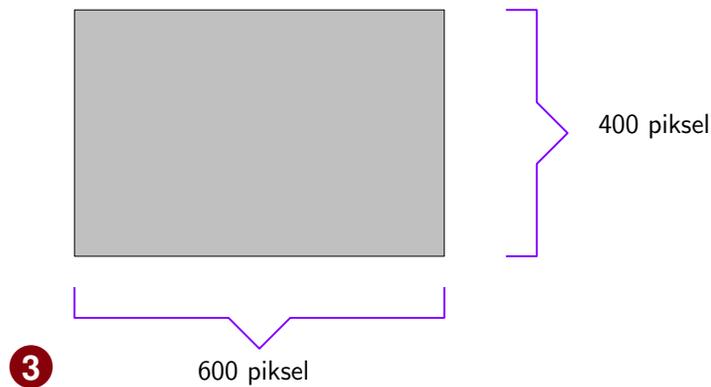
## Citra Digital

- 1 Citra analog mencapai puncaknya pada tahun 1980-an
- 2 Era citra digital dimulai sejak berkembangnya komputer tahun 1960-1970 an.
- 3 Pertama kali, citra digital harus ditampilkan pada Monitor Komputer
- 4 Monitor komputer tersusun atas **pixel** (piksel) = **picture element**. (Gambar di bawah (Sumber: <https://onlinepercentagecalculators.com/>))



## Citra Digital

- 1 Jumlah Piksel (baris×kolom) disebut dengan resolusi layar.
- 2 **Contoh:** resolusi  $400 \times 600$  berarti terdapat 400 baris piksel dan 600 kolom piksel.



**Figure :** Resolusi layar

- 4 Rasio layar untuk resolusi  $400 \times 600$  adalah **2:3**

## Citra Digital

- ① Pada masa lalu, layar komputer didesain untuk **bekerja**.
- ② Untuk keperluan ini, banyak digunakan rasio resolusi layar 3:4
- ③ **Misal:**
  - $240 \times 320$  (Resolusi CGA)
  - $480 \times 640$  (Resolusi EGA)
  - $600 \times 800$  (resolusi VGA)
  - $768 \times 1024$  (resolusi SVGA)
  - dst
- ④ Pada masa kini, desain komputer juga diarahkan untuk keperluan menonton
- ⑤ untuk keperluan **menonton**, layar komputer dibuat lebar (**Widescreen**)
- ⑥ Rasio resolusi Widescreen adalah **9:16**

# Citra Digital

## ① Contoh Resolusi Widescreen:

- $720 \times 1280$  (**Resolusi HD** atau **720p**)
- $768 \times 1366$
- $900 \times 1.600$
- $1080 \times 1920$  (Resolusi **Full HD** atau **1080p** atau **1080i**)



## Penutup

Demikian pada **Bagian I** ini kita telah mempelajari tentang perkembangan Citra Analog dan Pengantar perkembangan citra digital dari layar monitor komputer. Dengan rekapitulasi

- 1 Sejarah perkembangan Kamera analog
- 2 Sejarah perkembangan Film analog
- 3 Layar komputer
- 4 Resolusi
- 5 Rasio resolusi
- 6 Format 3:4 dan 9:16

Konsep dasar citra lebih lanjut akan dibahas pada **Slide 09**.

Bab 8

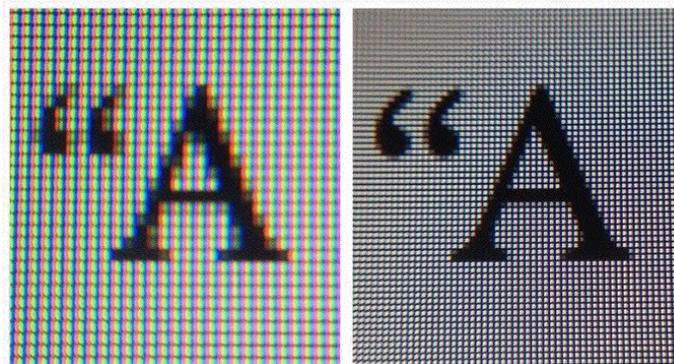
# Konsep Dasar Citra Digital

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami format penyimpanan citra **BW**, **Greyscale**, dan **RGB**
- 2 Mahasiswa dapat memahami teknik **konversi** citra RGB ke Greyscale dan ke BW
- 3 Mahasiswa mengetahui pula adanya basis warna **CYMK**

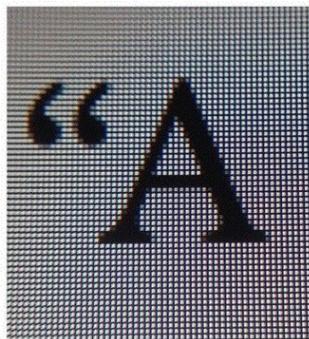
## Citra Digital

- 1 Citra digital tersusun atas komponen citra yang disebut piksel
- 2 Jumlah baris  $\times$  kolom ( $M \times N$ ) disebut sebagai resolusi citra
- 3 Semakin tinggi resolusi, semakin detil citra yang dihasilkan
- 4 Contoh berikut memperlihatkan citra dengan resolusi rendah dan resolusi tinggi (**sumber:** <http://teknoSrc.com/>)



## Citra Digital

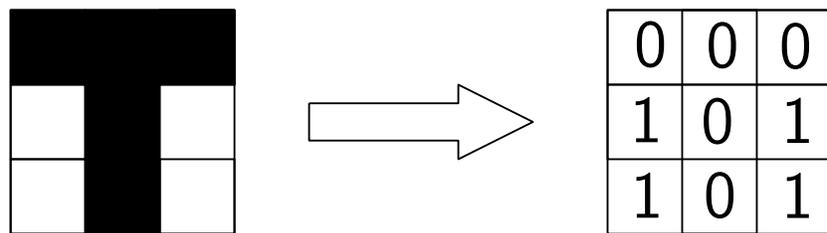
- 1 Setiap piksel pada citra memiliki intensitas tertentu
- 2 Konfigurasi intensitas piksel keseluruhan yang membuat terbentuknya citra



- 3 Untuk menyatakan intensitas, maka komputer menggunakan nilai intensitas
- 4 Nilai kecil untuk **intensitas rendah (gelap)** dan nilai besar untuk **intensitas tinggi (terang)**
- 5 Karena citra komputer adalah persegi, maka nilai intensitas **dapat disimpan** dalam **matriks**  $M \times N$ .

## Citra BW

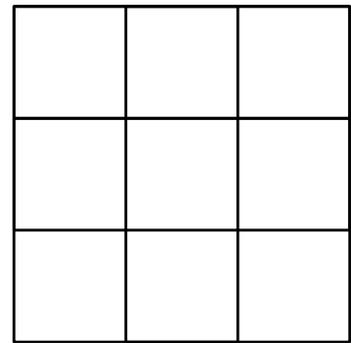
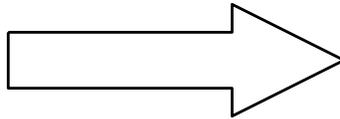
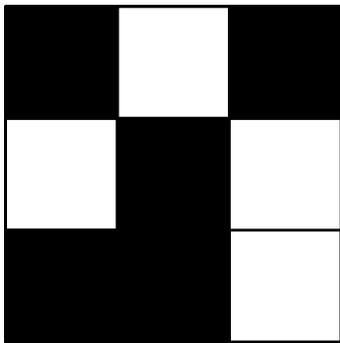
- 1 Citra BW hanya memiliki 2 warna yaitu **Hitam (Black)** dan **Putih (White)** disingkat sebagai Citra **BW**
- 2 Oleh karena hanya ada 2 warna, maka intensitas citra BW hanya ada 2 nilai yakni:  
**0** ⇒ **hitam**  
**1** ⇒ **putih**



**Figure :** Citra BW 3×3 dan nilai intensitasnya

## Citra BW (Latihan)

1 Coba tentukan nilai intensitas dari citra BW berikut:

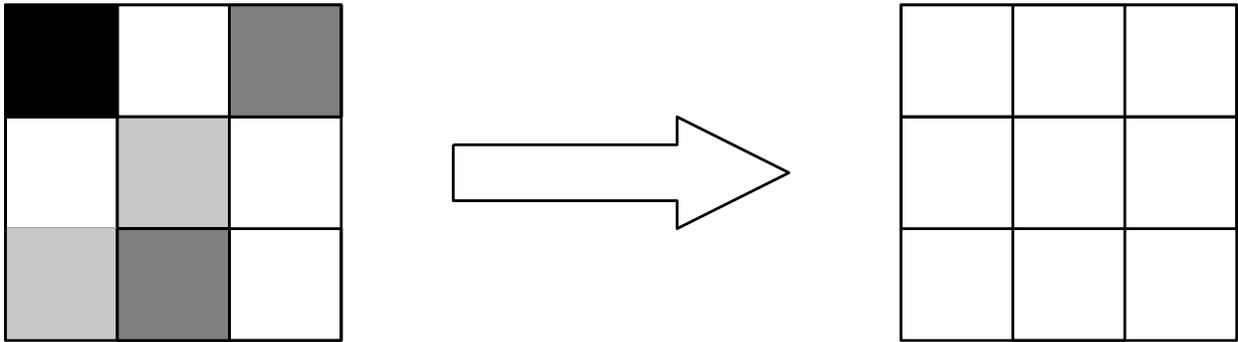


## Citra Greyscale

- 1 Jika citra BW hanya memiliki 2 intensitas,
- 2 Citra Greyscale memiliki **banyak** intensitas
- 3 Jumlah nilai intensitas, tergantung pada jumlah bit penyimpanan citra yang komputer gunakan. **Contoh:**
- 4 **Jika** komputer menyimpan nilai intensitas dengan **2 bit**, maka akan ada 4 nilai intensitas, yakni: **00**, **01**, **10** dan **11**.
- 5 Intensitas dalam desimal: **00 = 0**, **01 = 1**, **10 = 2** dan **11 = 3**
- 6 Nilai intensitas **0** mewakili warna **hitam**
- 7 Nilai Intensitas **3** mewakili warna **putih**
- 8 Nilai Intensitas **1** akan mewakili warna **Abu-abu gelap**
- 9 Nilai Intensitas **2** akan mewakili warna **Abu-abu terang**

## Citra Greyscale (Latihan)

- 1 Misalkan komputer menggunakan **2 bit** untuk menyimpan intensitas piksel, tentukan nilai intensitas untuk citra  $3 \times 3$  berikut:

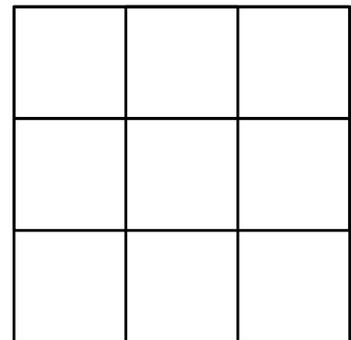
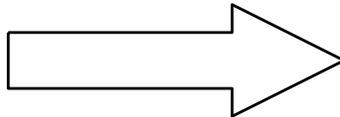
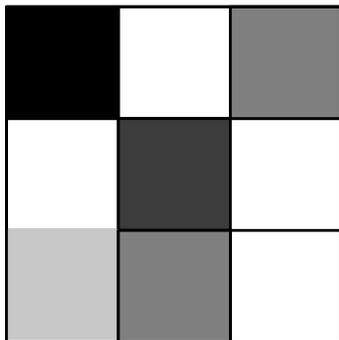


## Citra Greyscale 8-bit

- 1 Standard penyimpanan citra greyscale adalah 8 bit.
- 2 Dengan 8 bit ini, maka intensitas citra adalah dari **00000000** sampai dengan **11111111**
- 3 atau secara desimal nilai intensitas adalah dari **0** sampai **255**
- 4 Nilai intensitas **0** berkorespondensi dengan warna hitam
- 5 Nilai intensitas **255** berkorespondensi dengan warna putih.
- 6 Nilai intensitas **30** berkorespondensi dengan warna ...
- 7 Nilai intensitas **200** berkorespondensi dengan warna ...

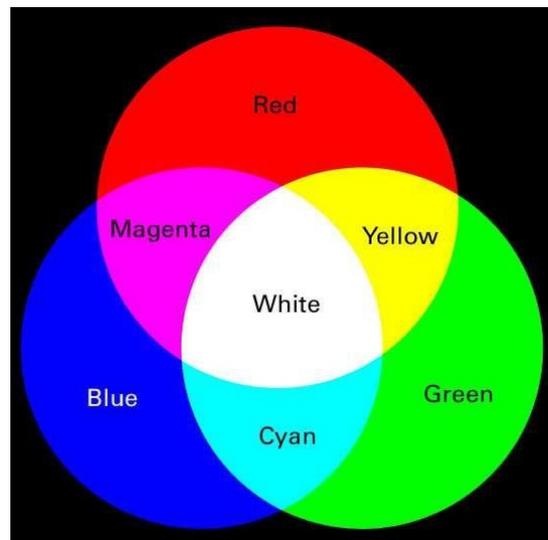
## Citra Greyscale 8-bit (Latihan)

- 1 Dengan format penyimpanan greyscale **8 bit**, perkirakan nilai intensitas dari citra  $3 \times 3$  berikut!



## Citra Warna RGB

- 1 Citra warna RGB terdiri dari **3 komponen**:
- 2 **Red, Green** dan **Blue**
- 3 Warna lain dapat diperoleh dengan campuran dari ketiga komponen ini.
- 4 Diagram campuran RGB (**sumber**: quora.com)

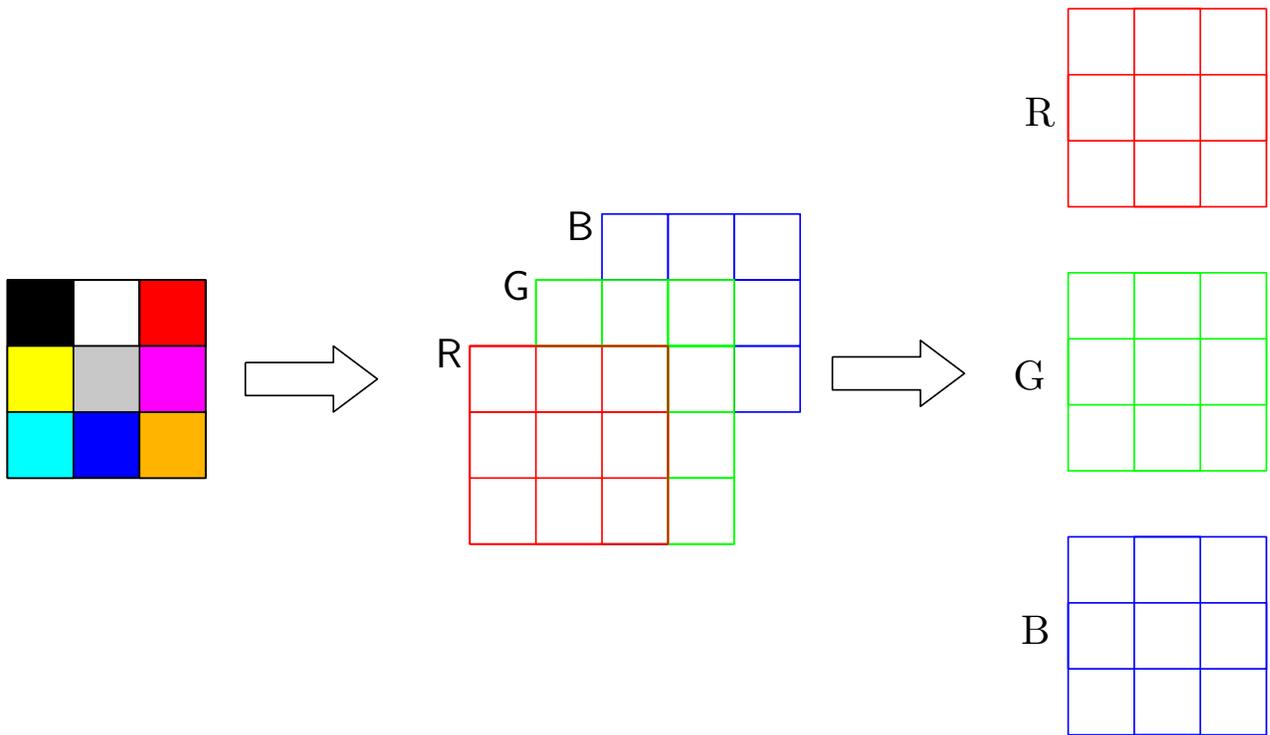


## Citra Warna RGB

- 1 Setiap piksel pada citra RGB memiliki **3 komponen warna** yaitu komponen **R**, komponen **G**, dan komponen **B**.
- 2 Jika setiap komponen intensitasnya dinyatakan dengan 8 bit (Red 8 bit, Green 8 bit, dan Blue 8 bit = 24 bit total),
- 3 maka setiap komponen bernilai **0** sampai **255**.
- 4 Perhatikan contoh latihan citra RGB  $3 \times 3$  piksel pada halaman berikut:

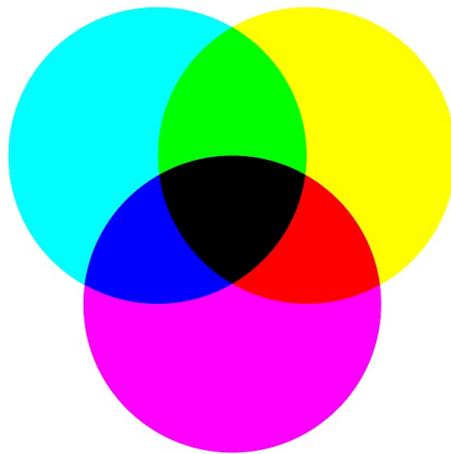
# Citra Warna RGB - Latihan

1 Tentukan nilai intensitas piksel pada citra RGB  $3 \times 3$  berikut:



## Catatan - Basis Warna CYMK

- 1 Disamping basis warna RGB yang standar pada representasi citra,
- 2 terdapat juga basis warna CYM (Cyan Yellow Magenta)
- 3 Kadang disebut dengan lengkap yaitu CYMK (Cyan Yellow Magenta Key)
- 4 Basis warna ini biasanya digunakan pada printer warna.



## Konversi RGB ke Greyscale

- 1 RGB 3 layer dapat dikonversi ke Greyscale 1 layer dengan rumus:

$$GREY = a \times R + b \times G + c \times B$$

dengan

$$a + b + c = 1$$

dan

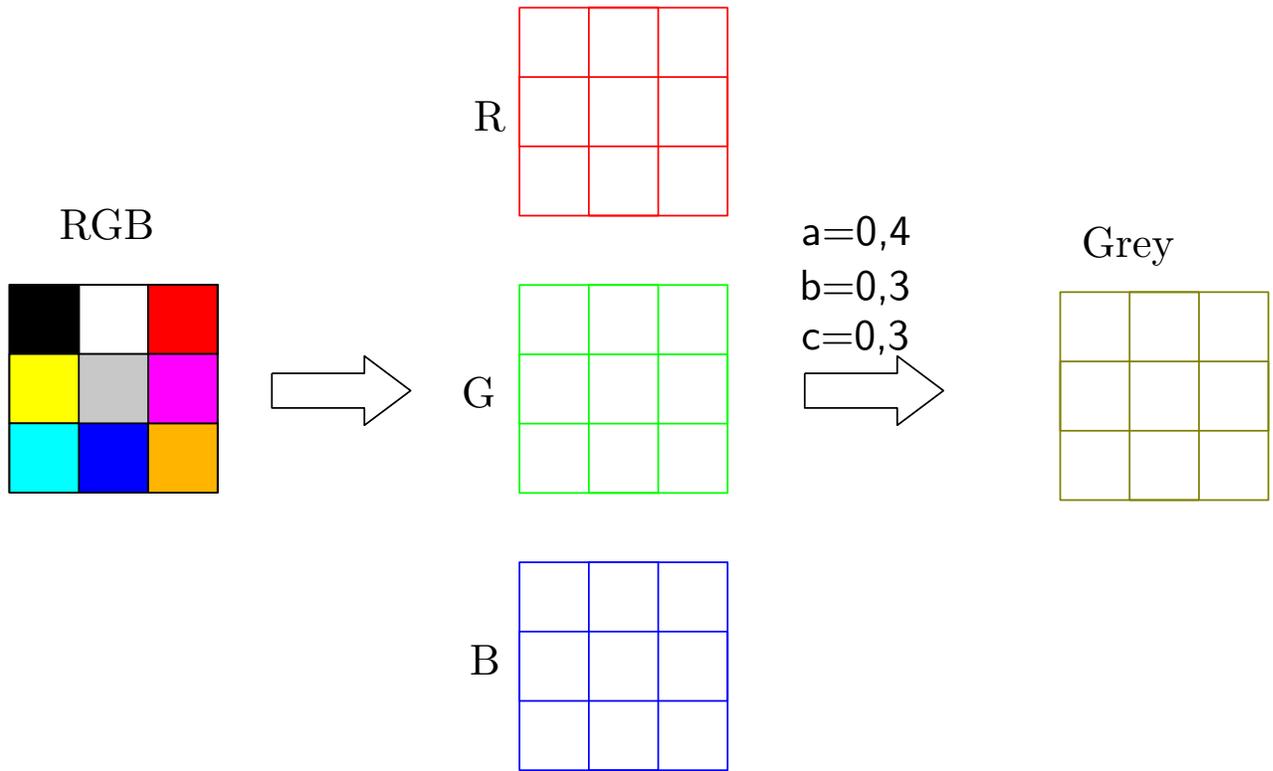
$$0 \leq a \leq 1$$

$$0 \leq b \leq 1$$

$$0 \leq c \leq 1$$

## Latihan konversi RGB ke Greyscale

- 1 Diberikan Citra RGB  $3 \times 3$  berikut. Jika dipilih  $a = 0,4$ ,  $b = 0,3$  dan  $c = 0,3$ , tentukan nilai intensitas citra Grey-nya!



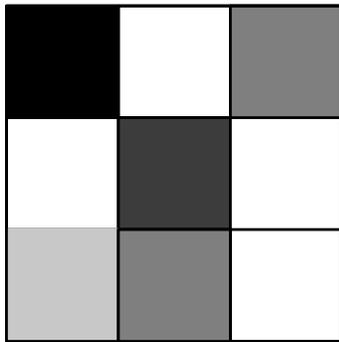
## Konversi Greyscale ke BW

- 1 Citra greyscale 1 layer dapat dikonversi ke citra BW 1 layer dengan teknik **thresholding**.
- 2 **Jika** ditetapkan nilai **threshold = Th**,
- 3 **maka** intensitas piksel yang **kurang dari Th** dijadikan **0**,
- 4 intensitas piksel yang **di atas atau sama dengan Th** dijadikan **1**
- 5 Perhatikan contoh latihan pada slide berikut ini:

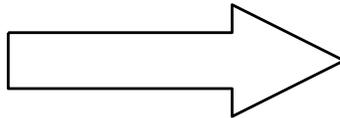
## Konversi Greyscale ke BW (Latihan)

- 1 Diberikan citra greyscale 8-bit di bawah:
- 2 perkirakan nilai intensitasnya!
- 3 Jika dipilih nilai threshold sebesar 120, tentukan intensitas citra BW-nya!

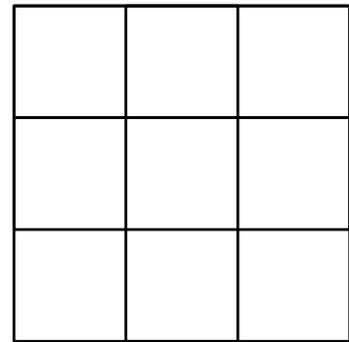
Greyscale



Th=120



BW



## Latihan OCTAVE

### Persiapan:

- 1 Unduh citra RGB **peppers.png** dari situs:  
<https://www.mathworks.com/help/matlab/ref/rgb2gray.html>
- 2 simpan citra ini dengan nama **peppers.png** pada folder latihan octave di komputer Anda.
- 3 Buka file **peppers.png** ini di **MS-Paint**, kemudian **save-as** sebagai **BMP** dengan nama **peppers2.bmp**



## Latihan OCTAVE

- 1 Perintah membaca citra di OCTAVE:  
**`x1 = imread('peppers2.bmp')`**
- 2 Perintah menampilkan citra di figure(1):  
**`figure(1); imshow(x1)`**
- 3 Lanjutkan dengan konversi ke greyscale:  
**`R = x1(:,:,1); %ambil layer 1 simpan ke R`**  
**`G = x1(:,:,2); %ambil layer 2 simpan ke G`**  
**`B = x1(:,:,3); %ambil layer 1 simpan ke B`**  
**`a=0.4; b=0.3; c=0.3; % set nilai a,b,dan c`**  
**`GREY = a*R+b*G+c*B; %hitung GREY`**
- 4 Tampilkan GREY di figure 2:  
**`figure(2); imshow(GREY)`**

## Latihan OCTAVE

- 1 Lanjutkan konversi GREY ke BW  
**Th = 120;** %tetapkan threshold  
**BW = (GREY >= Th);** %lakukan thresholding,  $\geq$  Th menjadi 1  
%kurang dari Th menjadi 0
- 2 Tampilkan BW di figure 3:  
**figure(3); imshow(BW)**

### Catatan:

- OCTAVE menyediakan toolbox juga untuk mengubah citra RGB ke Grey, yaitu dengan perintah: **rgb2gray**
- Untuk konversi dari Grey ke BW: **im2bw**
- **im2bw** juga dapat digunakan untuk mengubah dari RGB langsung ke BW

## Penutup

- 1 Pada Slide 09 ini, kita telah membahas tentang citra BW, Greyscale, dan RGB
- 2 Dilanjutkan dengan mengkonversi RGB ke Greyscale dan Greyscale ke BW
- 3 Terdapat pula basis warna **CYMK** disamping **RGB** yang populer di printer warna.
- 4 Terdapat pula basis warna **YCbCr** yang diberikan kepada para Mahasiswa untuk mengeksplorasi basis warna tersebut.

## Bab 9

# Filtering pada Citra Digital

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami teknik pengolahan citra pada bidang Filtering

## Konvensi

Pada pengolahan citra pada slide ini, **diasumsikan** hal-hal berikut:

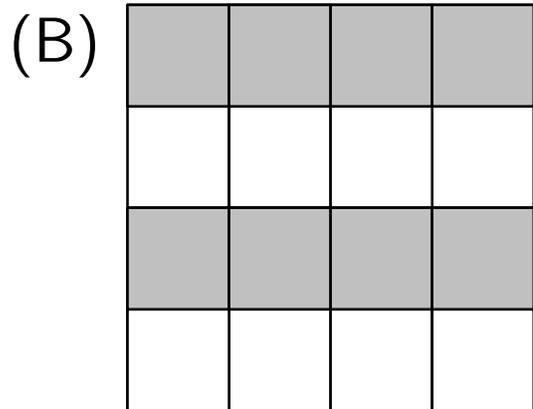
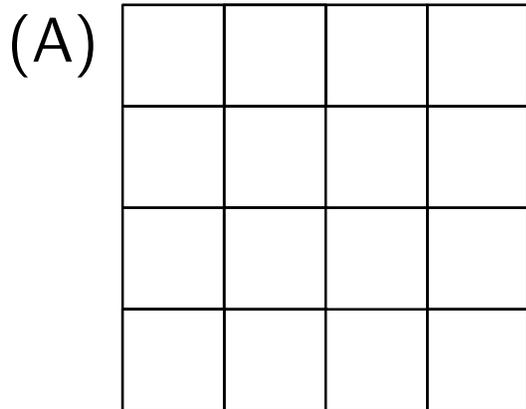
- ① Citra yang digunakan adalah citra **Greyscale** 8 bit (0-255)
- ② Pengolahan yang mengeluarkan nilai pecahan, hasil akhirnya **dibulatkan** dengan **aturan pembulatan**
- ③ Misal:
  - **round**(2,3) = 2
  - **round**(4,7) = 4
  - **round**(5,5) = 6
- ④ Hasil **operasi negatif** dijadikan **0**
- ⑤ Hasil **operasi** yang **lebih dari 255** dijadikan **255**.

## Konsep Frekuensi Pada Citra

- 1 Frekuensi terkait dengan perubahan
- 2 Semakin cepat terjadi perubahan semakin tinggi frekuensi
- 3 Frekuensi 0 berarti tidak ada perubahan
- 4 Perhatikan 2 contoh citra dengan dimensi  $4 \times 4$  piksel pada slide berikut:

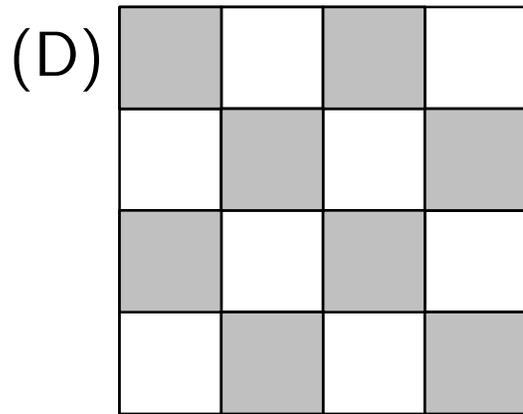
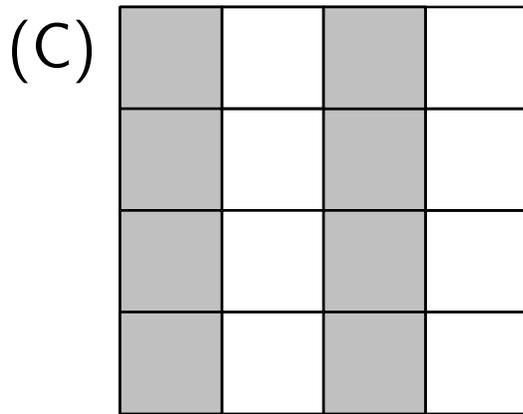
# Konsep Frekuensi Pada Citra

- 1 Warna putih memiliki intensitas 255
- 2 Warna abu-abu anggap memiliki intensitas 150



# Konsep Frekuensi Pada Citra

- 1 Warna putih memiliki intensitas 255
- 2 Warna abu-abu anggap memiliki intensitas 150



## Konsep Frekuensi Pada Citra

- 1 Oleh karena nilai intensitas piksel berubah dari piksel ke piksel
- 2 Sedangkan piksel-piksel memiliki koordinat ruang
- 3 Maka frekuensi pada citra disebut juga dengan **Frekuensi Spatial** (**space** = ruang)
- 4 Audio, di sisi lain, memiliki amplitudo berbeda dari waktu ke waktu
- 5 Oleh karena itu frekuensi pada audio disebut juga **Frekuensi Temporal** (**tempo** = waktu)

## Konsep Frekuensi Pada Citra - Contoh

- 1 Perhatikan citra Lena-Khan di bawah!
- 2 Tandai bagian yang berfrekuensi rendah dan tinggi!



## Latihan

- 1 Perhatikan citra Lena-Soederberg<sup>1</sup> di bawah!
- 2 Tandai bagian yang berfrekuensi rendah dan tinggi!



<sup>1</sup>Citra Lena Soederberg ini terkenal karena digunakan pertama kali pada format kompresi jpg

## Low Pass dan High Pass pada Citra

- 1 Dampak LPF pada citra adalah citra menjadi lebih brur
- 2 Sebaliknya HPF pada citra menghasilkan citra tajam



asli



hasil LPF



asli



hasil HPF

## Struktur Element atau Kernel

- 1 Untuk melakukan filter pada citra, maka kita memerlukan Struktur Element (Strel) atau Kernel
- 2 Kernel adalah matriks ukuran persegi  $M \times M$ , dengan  $M > 1$
- 3 Contoh Strel  $2 \times 2$  :

$$K = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

- 4 Strel  $3 \times 3$

$$K = \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

- 5 dan sebagainya

# Filtering

- 1 Untuk melakukan **filtering**, maka Citra **dikonvolusikan** dengan **Strel** atau **kernel**.
- 2 Mengkonvolusikan citra dengan kernel adalah **mengoverlay** kernel ke citra, kemudian **dikalikan** setiap elemennya dan **dijumlahkan**.
- 3 Agar **dimensi citra hasil sama** dengan **dimensi citra asal**, maka dapat dilakukan **zero padding** pada tepi citra asli sebelum dilakukan konvolusi dengan kernel
- 4 Perhatikan **contoh** konvolusi halaman-halaman selanjutnya.

# Filtering - Contoh

**Kernel dimensi genap:**

- 1 Konvolusikan antara citra  $4 \times 4$  dengan kernel  $2 \times 2$  berikut!

30	50	50	30
30	50	50	30
30	50	50	30
30	50	50	30

citra

1	1
0	1

kernel

# Filtering

**Jawab:**

- 1 Agar **dimensi** citra hasil konvolusi sama dengan dimensi citra semula, maka pertama dilakukan **zero padding** dulu sbb:

30	50	50	30	0
30	50	50	30	0
30	50	50	30	0
30	50	50	30	0
0	0	0	0	0

citra

1	1
0	1

kernel

# Filtering

## Jawab - lanjutan

- 1 Selanjutnya **kernel** di-**overlay** ke citra padding
- 2 telah itu digeser ke piksel selanjutnya

1	30	1	50	50	30	0
0	30	1	50	50	30	0
	30		50	50	30	0
	30		50	50	30	0
	0		0	0	0	0

hasil konvolusi

130			

## CEK Hasil perhitungan

Dengan OCTAVE (atau MATLAB) kita bisa mengecek hasil perhitungan (fungsi Octave yang diperlukan: **filter2**)

- 1 Pertama deklarasikan citra:  
 $\mathbf{x} = [30 \ 50 \ 50 \ 30; 30 \ 50 \ 50 \ 30; 30 \ 50 \ 50 \ 30; 30 \ 50 \ 50 \ 30]$
- 2 Kemudian deklarasikan kernel:  
 $\mathbf{kernel} = [1 \ 1; 0 \ 1]$
- 3 Lanjutkan dengan filter/konvolusi:  
 $\mathbf{y} = \mathbf{filter2}(\mathbf{kernel}, \mathbf{x})$

# Latihan

1 Konvolusikan antara citra  $4 \times 4$  dengan kernel  $2 \times 2$  berikut!

30	50	50	30
30	50	50	30
30	50	50	30
30	50	50	30

citra

0	1
0	1

kernel

**Jawab: ...**

# Filtering-Contoh

Kernel dimensi **Ganjil**

① Konvolusikan antara citra  $4 \times 4$  dengan kernel  $3 \times 3$  berikut!

30	50	50	30
30	50	50	30
30	50	50	30
30	50	50	30

citra

1	0	0
1	0	0
1	1	1

kernel

**Jawab:** Zero Padding citra:

0	0	0	0	0	0
0	30	50	50	30	0
0	30	50	50	30	0
0	30	50	50	30	0
0	30	50	50	30	0
0	0	0	0	0	0

1	0	0
1	0	0
1	1	1

kernel




# Latihan

Kernel dimensi **Ganjil**

① Konvolusikan antara citra  $4 \times 4$  dengan kernel  $3 \times 3$  berikut!

30	50	50	30
30	50	50	30
30	50	50	30
30	50	50	30

citra

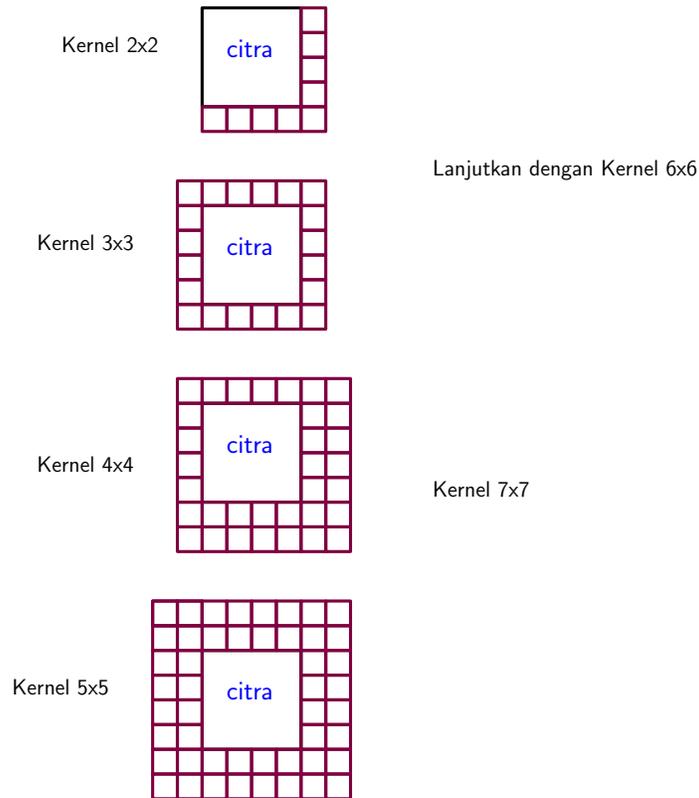
1	0	0
1	0	0
1	1	1

kernel

**Jawab:** Zero Padding citra:

# Konvensi Pola penyisipan Zero Padding

OCTAVE dan MATLAB menyepakati pola penyisipan Zero Padding sebagai berikut:



# Struktur Kernel LPF

- ❶ Struktur Kernel LPF memiliki ciri khas yakni semua (atau mayoritas) nilai elemennya tak negatif
- ❷ Perhatikan contoh kernel LPF di bawah:

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

1	1	1			
0	0	0			
0	0	0			

## Struktur Kernel HPF

- 1 Struktur Kernel HPF memiliki ciri yaitu jika semua nilai elemen pada kernel dijumlahkan, hasilnya 0 atau mendekati 0
- 2 Perhatikan contoh kernel HPF di bawah:

1	-1
0	0

1	0
-1	0

-2	1
1	0

1	-1
1	-1

1	-2	1
0	0	0
0	0	0


## Addendum: Filter Median

- 1 Filter Median memiliki cara kerja unik
- 2 Kernelnya tidak berisi apa-apa, namun digunakan untuk menyeleksi blok piksel pada citra
- 3 Dilakukan **zero padding** sebelum pembuatan blok
- 4 Nilai tengah (median) dari blok diambil sebagai hasil filter
- 5 **Filter Median** digunakan untuk menghilangkan derau tipe speckles
- 6 **Contoh:**
- 7 Lakukan **Median Filter** pada citra  $4 \times 4$  dengan **kernel** berukuran  $2 \times 2$  berikut!

# Contoh Median Filter

Citra asli dan Kernel:

30	32	33	30
30	200	29	30
30	28	30	30
30	34	98	30

citra


kernel

**1 Jawab:** Pertama lakukan zero padding

30	32	33	30	
30	200	29	30	
30	28	30	30	
30	34	98	30	

citra-padding


kernel




# Latihan

Lakukan Median filter untuk citra  $4 \times 4$  dengan kernel  $3 \times 3$

30	32	33	30
30	200	29	30
30	28	30	30
30	34	98	30

citra


kernel

**Jawab:**

## Penutup

- ➊ Pada **Slide 10** ini, kita telah membahas tentang pengolahan Citra **Bagian I**
- ➋ **Bagian I** ini adalah tentang **filtering**
- ➌ Diawali dengan konsep **frekuensi spatial** pada citra
- ➍ dilanjutkan dengan **Struktur Elemen** atau **Kernel**
- ➎ Diberikan contoh konvolusi citra dan kernel
- ➏ **Sifat** kernel LPF dan HPF
- ➐ dan diilustrasikan pula tentang Median Filtering
- ➑ Pada **Slide 11** akan dibahas tentang **operasi Morfologi** Pada Citra.

## Bab 10

# Operasi Morfologi pada Citra Digital

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami operasi Morfologi pada Citra Digital, khususnya Citra BW

# Pendahuluan

- 1 Operasi morfologi dilakukan pada Citra **Black and White (BW)** atau disebut juga dengan citra Biner
- 2 Citra Biner biasanya berasal dari citra Greyscale
- 3 Citra Greyscale biasanya berasal dari citra warna (RGB)



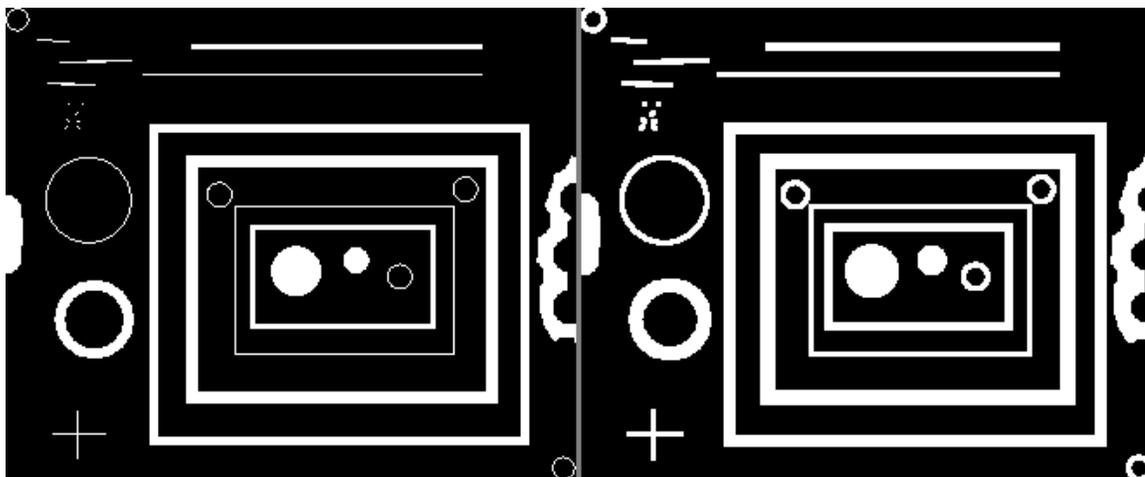
**Figure :** Citra RGB, Grey, BW (Sumber: mathworks.com)

## Arti Morfologi

- ① Morfologi, atau **morphology** berasal dari bahasa Yunani: **Morphe + logos**
- ② **Morphe** berarti bentuk (atau **form** atau **shape**)
- ③ **Logos** berarti **ilmu**
- ④ Jadi morfologi adalah ilmu tentang bentuk.
- ⑤ Operasi Morfologi pada citra adalah operasi untuk **memodifikasi bentuk** yang terdapat pada citra.
- ⑥ Operasi Morfologi meliputi: **Dilasi, Erosi, Opening, Closing,**  
...
- ⑦ Operasi morfologi ini dapat dikembangkan lebih lanjut dengan operasi labeling, dsb...

## Dilasi

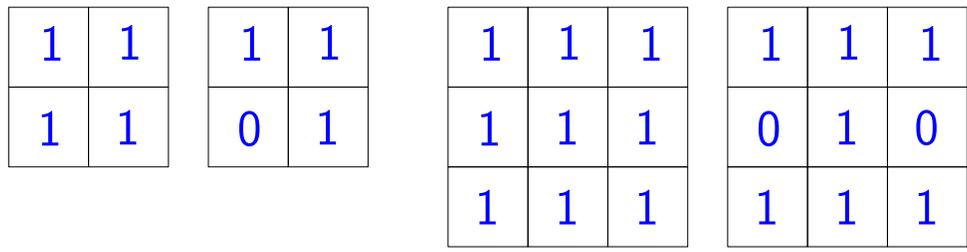
- 1 Dilasi atau **dilation** adalah operasi untuk mempertebal warna putih pada citra BW.
- 2 Contoh hasil dari operasi Dilasi:



**Figure :** Kiri: citra asal, Kanan: citra hasil dilasi

# Dilasi

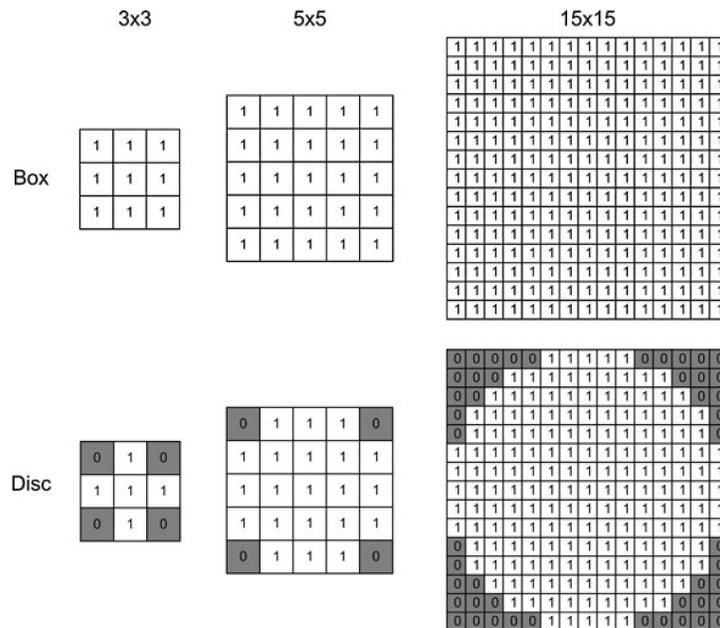
- ① Langkah pertama dilasi adalah mendefinisikan **kernel** atau **structuring elemen (strel)**
- ② Contoh Kernel:



**Figure :** Contoh kernel ukuran 2×2 dan 3×3

# Dilasi

- 1 Untuk mendeskripsikan **strel** atau **kernel**, orang mengistilahkan beberapa bentuk **kernel**
- 2 **Contoh:** kernel **box** dan kernel **disc** seperti contoh pada gambar (**Sumber:himnicksn.medium.com**).

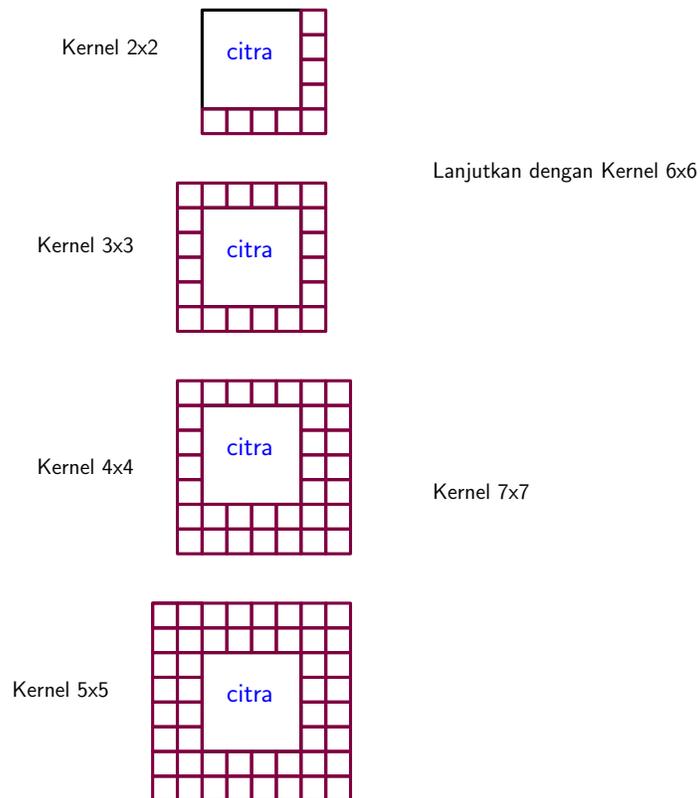


## Dilasi

- 1 Setelah kernel dideklarasikan, langkah kedua adalah **zero-padding**
- 2 **Zero padding** diperlukan agar dimensi citra hasil sama dengan dimensi citra asal.
- 3 teknik **zero padding** pada operasi morfologi adalah sama dengan teknik **zero padding** pada filtering (slide berikut)

# Dilasi

## Aturan standard **zero-padding**



## Dilasi

- 1 Setelah zero-padding dilakukan, maka langkah selanjutnya adalah proses **over-lay** dan **sliding**
- 2 Proses Overlay adalah meletakkan kernel pada citra asal
- 3 Titik yang ter-**over-lay** dikalikan dan dijumlahkan
- 4 Jika hasil penjumlahan  $> 0$  maka hasil akhir adalah **1**
- 5 Jika hasil penjumlahan  $= 0$ , maka hasil akhir adalah **0**.
- 6 Setelah selesai over-lay pada satu posisi, selanjutnya kernel digeser (di-sliding) ke piksel berikutnya.
- 7 Proses ini diulang sampai semua piksel pada citra asal selesai dioverlay
- 8 Perhatikan **contoh** berikut.

## Dilasi - Contoh

- 1 **Soal:** Diberikan citra BW  $4 \times 4$  berikut. Dengan kernel **box**  $2 \times 2$ , tentukan hasil keluaran dari proses dilasi antara citra asal dengan kernel tersebut!

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

Citra asli

- 2 **Jawab:** Kernel **box**  $2 \times 2$  memiliki bentuk:

1	1
1	1

## Dilasi - Contoh

- 1 Selanjutnya, zero-padding pada citra semula kita peroleh:

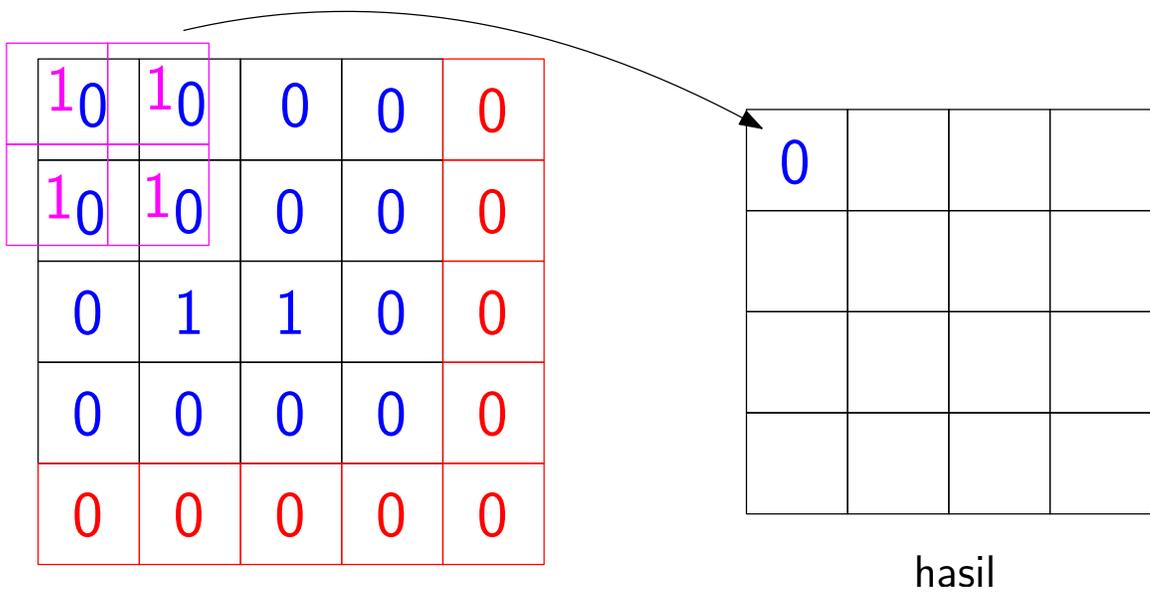
0	0	0	0	0
0	0	0	0	0
0	1	1	0	0
0	0	0	0	0
0	0	0	0	0

1	1
1	1

**Figure :** Hasil zero-padding (kiri) dan kernel (kanan)

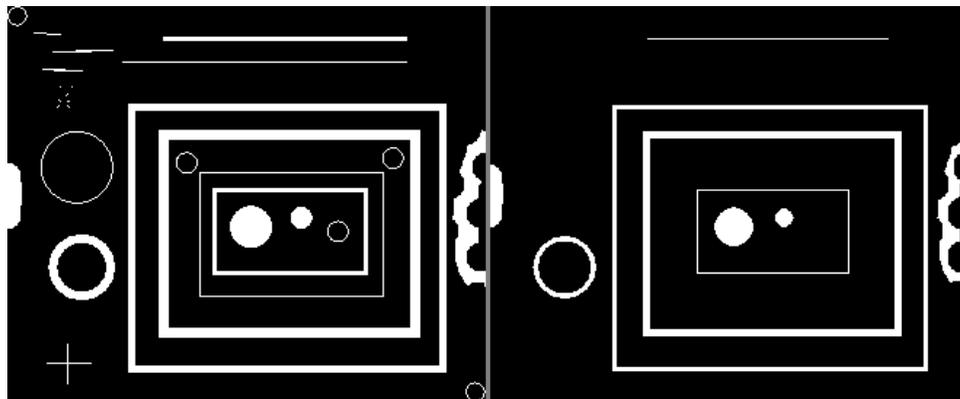
# Dilasi - Contoh

② Lakukan overlay dan sliding, kita peroleh:



## Dilasi

- 1 Jika dilasi mempertebal garis atau area berwarna putih,
- 2 Erosi sebaliknya, mempertipis garis atau area berwarna putih pada citra BW.



## Erosi

- ➊ Tahapan dari proses erosi adalah sama dengan dilasi, yakni:
- ➋ Mendefinisikan kernel
- ➌ Proses Overlay, dengan aturan:
- ➍ Titik yang ter-**over-lay** dikalikan dan dijumlahkan
- ➎ Jika hasil penjumlahan  $>$  jumlah 1 pada kernel dibagi 2, maka hasil akhir adalah 1,
- ➏ Jika hasil penjumlahan  $\leq$  jumlah 1 pada kernel dibagi 2, maka hasil akhir adalah **0**.
- ➐ Setelah selesai over-lay pada satu posisi, selanjutnya kernel digeser (di-sliding) ke piksel berikutnya.
- ➑ Proses ini diulang sampai semua piksel pada citra asal selesai dioverlay
- ➒ Perhatikan **contoh** berikut.

## Erosi - Contoh

- ① **Soal:** Diberikan citra BW  $4 \times 4$  berikut. Dengan kernel **box**  $2 \times 2$ , tentukan hasil keluaran dari proses erosi antara citra asal dengan kernel tersebut!

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

Citra asli

- ② **Jawab:** Kernel **box**  $2 \times 2$  memiliki bentuk:

1	1
1	1

Jumlah 1 pada kernel adalah  $1+1+1+1 = 4$ ; dan setelah dibagi 2 menjadi  $4/2=2$  (threshold)

## Erosi - Contoh

- 1 Selanjutnya, zero-padding pada citra semula kita peroleh:

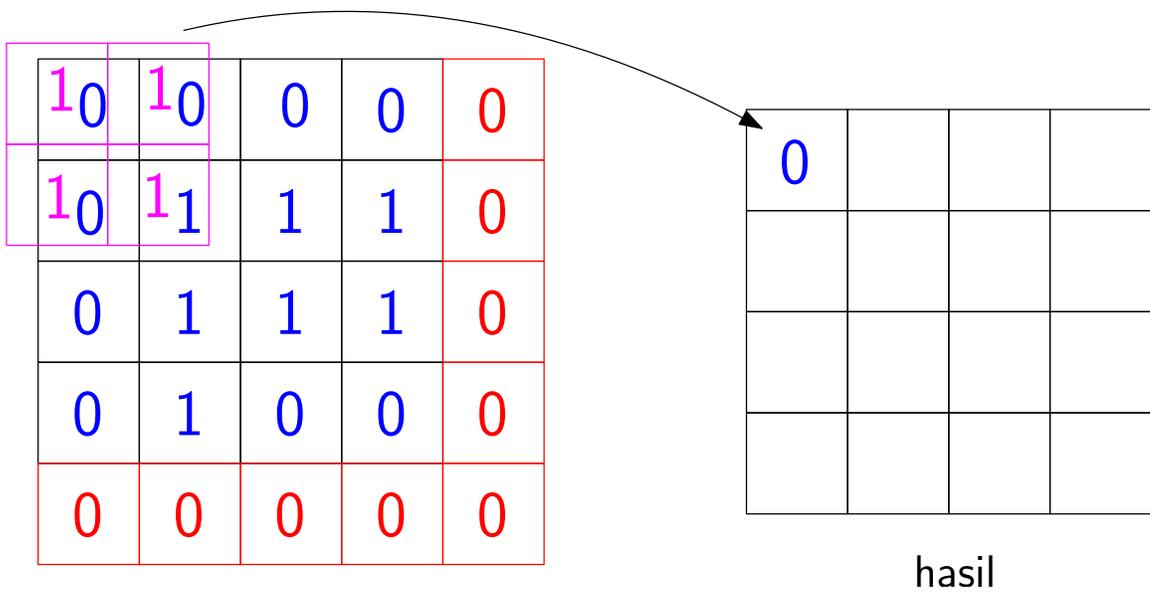
0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	0	0	0
0	0	0	0	0

1	1
1	1

**Figure :** Hasil zero-padding (kiri) dan kernel (kanan)

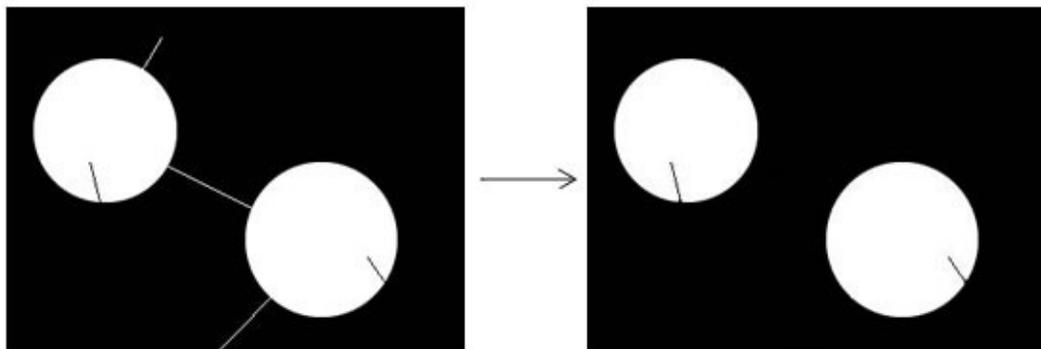
# Erosi - Contoh

② Lakukan overlay dan sliding, kita peroleh:



## Opening

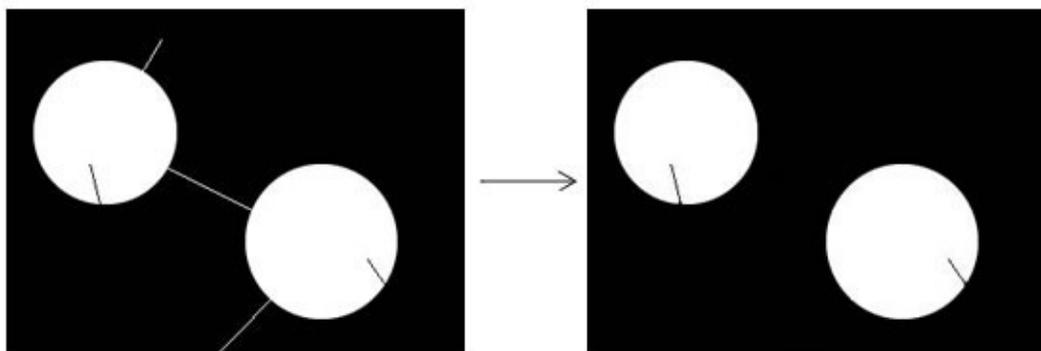
- 1 Operasi **opening** adalah operasi komposit yang tersusun dari operasi EROSI dilanjutkan dengan operasi DILASI
- 2 Dampak dari operasi opening, setiap ada warna hitam dalam kluster putih akan tetap ada (terbuka)
- 3 Namun warna putih yang kecil di dalam hitam akan hilang



Morphological Opening

## Opening

- 1 Operasi **opening** adalah operasi komposit yang tersusun dari operasi EROSI dilanjutkan dengan operasi DILASI
- 2 Dampak dari operasi opening, setiap ada warna hitam dalam kluster putih akan tetap ada (terbuka)
- 3 Namun warna putih yang kecil di dalam hitam akan hilang



Morphological Opening

## Operasi Opening - Latihan

- 1 **Soal:** Diberikan citra BW  $4 \times 4$  berikut. Dengan kernel **box**  $2 \times 2$ , tentukan hasil keluaran dari operasi opening antara citra asal dengan kernel tersebut!

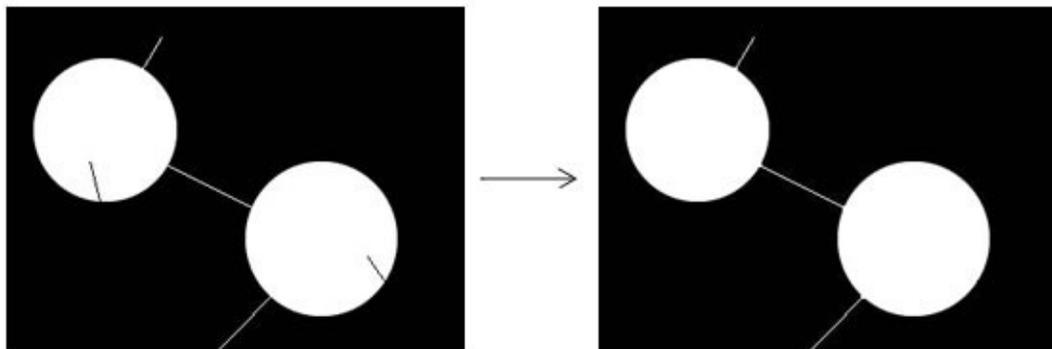
0	0	0	0
0	1	1	1
0	1	1	1
0	1	0	0

Citra asli

- 2 **Jawab:** ....

## Closing

- 1 Operasi **closing** adalah operasi komposit yang tersusun dari operasi DILASI dilanjutkan dengan operasi EROSI
- 2 Dampak dari operasi closing, setiap ada warna hitam dalam kluster putih akan hilang (tertutup)
- 3 sedangkan warna putih yang kecil di dalam hitam akan tetap ada



Morphological Closing

## Operasi Closing - Latihan

- 1 **Soal:** Diberikan citra BW  $4 \times 4$  berikut. Dengan kernel **box**  $2 \times 2$ , tentukan hasil keluaran dari operasi closing antara citra asal dengan kernel tersebut!

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

Citra asli

- 2 **Jawab:** ....



**Bab 11**

# **Wavelet pada Citra Digital**

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa dapat memahami proses dekomposisi menggunakan teknik Wavelet Pada Citra
- 2 Mahasiswa dapat memahami proses rekonstruksi menggunakan teknik Wavelet Pada Citra

## Pendahuluan

- 1 Teknik Dekomposisi dan Rekonstruksi wavelet pada Audio telah kita bahas pada Pertemuan 5 (Slide 5)
- 2 Wavelet terdiri dari 2 bagian:
- 3 Bagian Dekomposisi (berfungsi **memecah** sinyal menjadi **komponen frekuensi Rendah L** dan **komponen frekuensi Tinggi H**)
- 4 Bagian Rekonstruksi (berfungsi menggabungkan **L** dan **H** menjadi sinyal semula)
- 5 Mari kita **review** sejenak Wavelet pada Audio

## Wavelet Pada Audio

Wavelet terdiri dari 2 bagian:

### 1 Bagian Dekomposisi

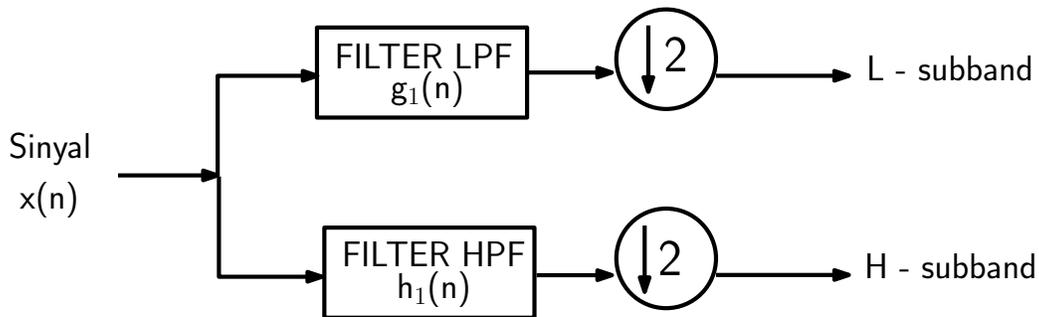


### 2 Bagian Rekonstruksi



## Struktur Dekomposisi Wavelet Audio

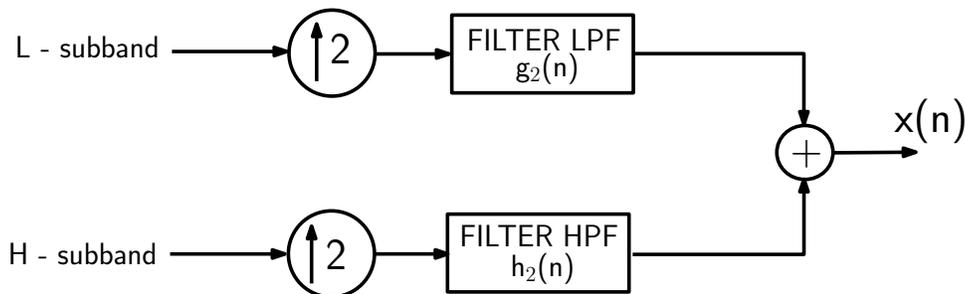
Struktur Dekomposisi Wavelet adalah sebagai berikut:



- 1 Sinyal  $x(n)$  dialirkan ke dua jalur: Jalur **Subband L** dan **Subband H**
- 2 Pada **Jalur L**, sinyal **difilter** dengan LPF  $g_1(n)$  kemudian di-**downsampling** dengan faktor 2.
- 3 Pada **Jalur H**, sinyal **difilter** dengan HPF  $h_1(n)$  kemudian di-**downsampling** dengan faktor 2.

## Struktur Rekonstruksi Wavelet Audio

Struktur Dekomposisi Wavelet adalah sebagai berikut:



- 1 Sinyal **L** dilakukan up-sampling dengan faktor 2, kemudian difilter dengan LPF  $g_2(n)$
- 2 Sinyal **H** dilakukan up-sampling dengan faktor 2, kemudian difilter dengan HPF  $h_2(n)$
- 3 Sinyal keluaran kedua jalur dijumlahkan untuk memperoleh  $x(n)$  rekonstruksi.

## Wavelet Pada Citra

Asumsi:

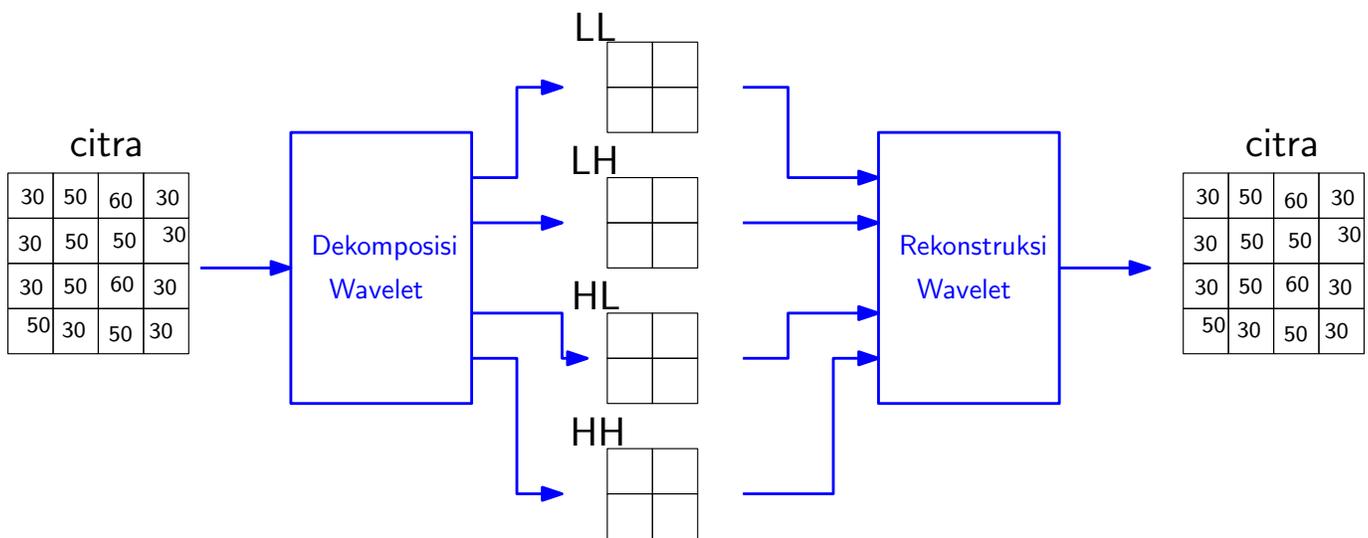
- ➊ Meskipun Wavelet dapat diterapkan pada citra RGB dan WB, namun pada pembahasan ini kita menggunakan citra **Greyscale**
- ➋ **Greyscale** yang digunakan adalah **8 bit**.

Dengan asumsi di atas, maka contoh tipikal nilai intensitas citra, misalnya, adalah

30	50	60	30
30	50	50	30
30	50	60	30
30	50	50	30

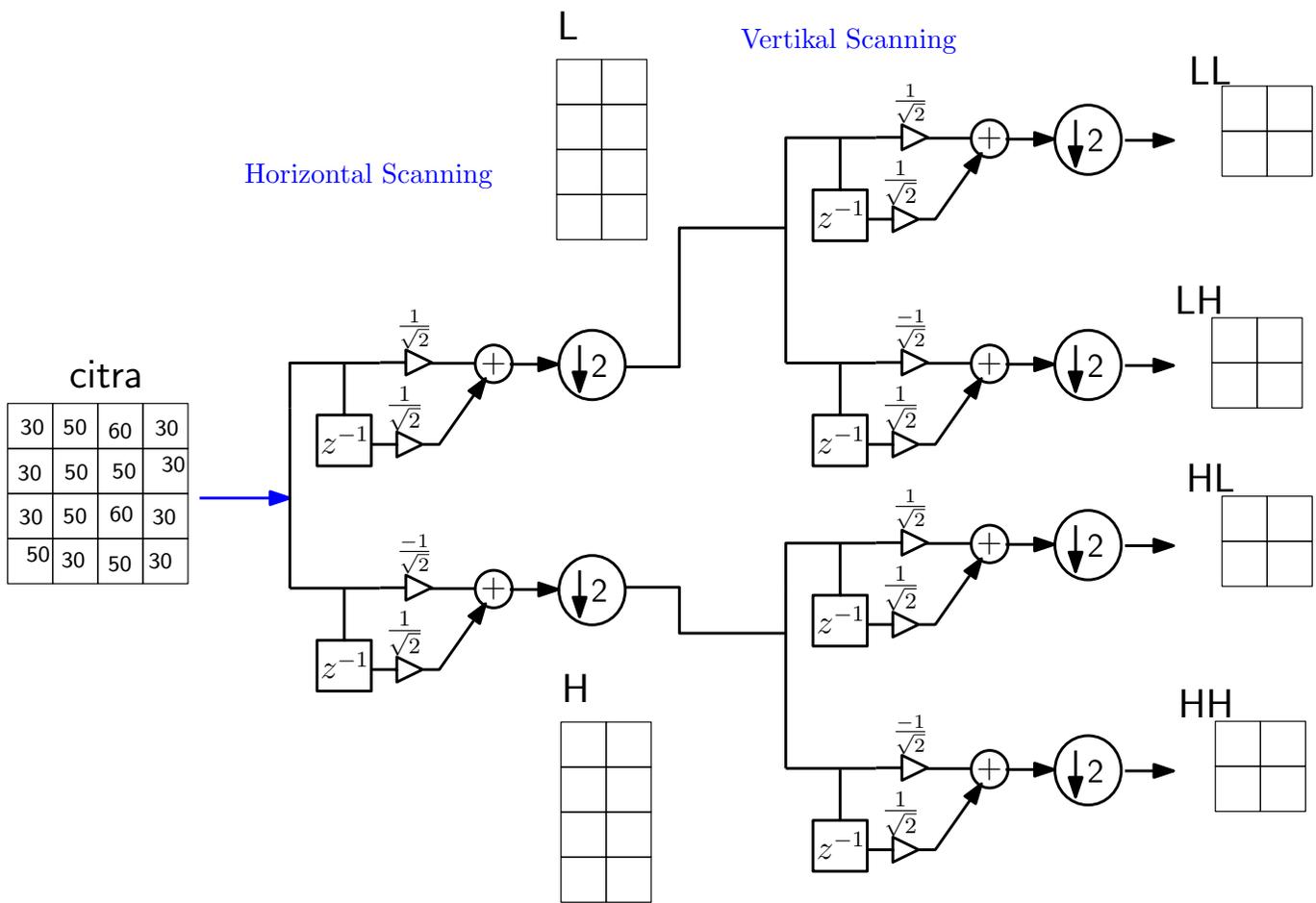
citra

## Blok Diagram Wavelet pada Citra

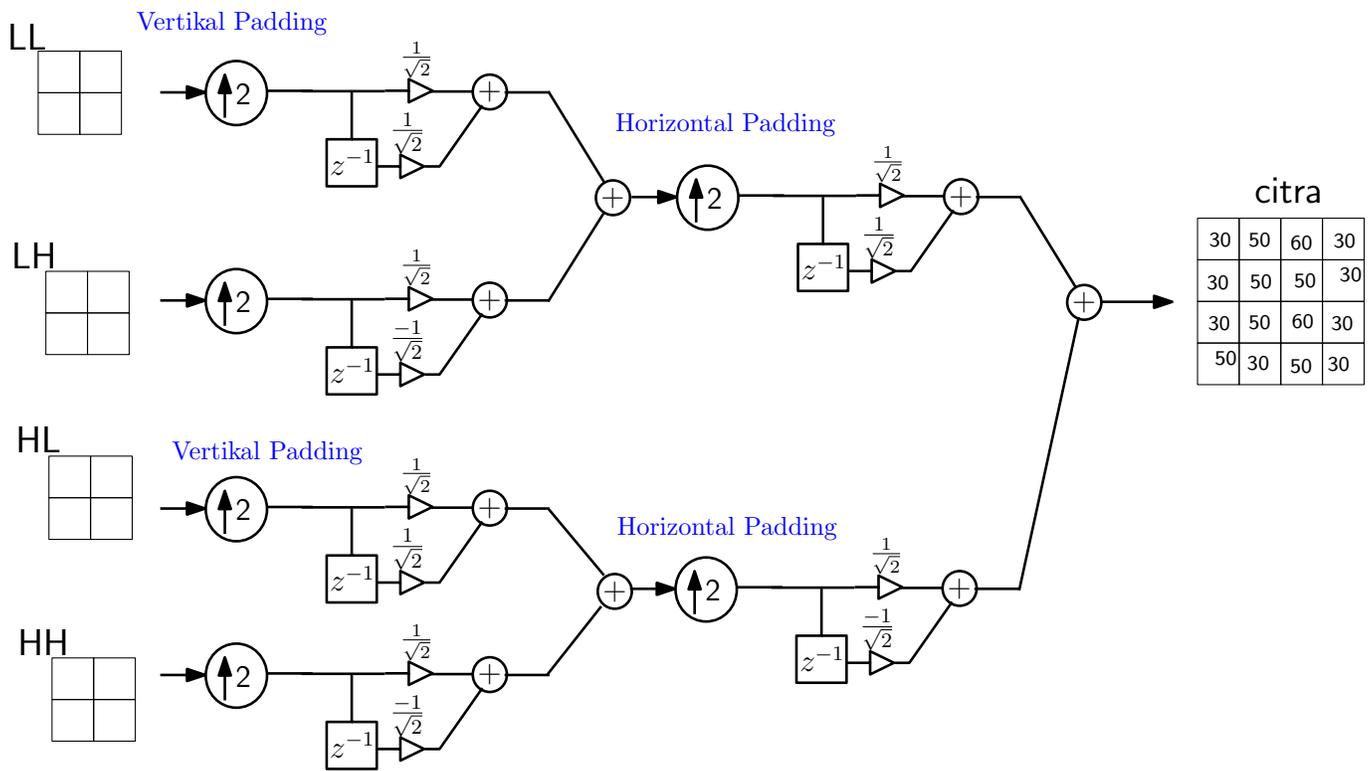


- 1 Perhatikan dimensi dari subband LL, LH, HL dan HH adalah  $1/4$  dari dimensi citra semula
- 2 Modifikasi, penyisipan, dll. biasanya dilakukan di salah satu dari subband LL, LH, HL, HH

# Rincian Blok Diagram Dekomposisi - Haar Wavelet



# Rincian Blok Diagram Rekonstruksi - Haar Wavelet



## Perhatikan Pasangan Filter Dekomposisi dan Rekonstruksi

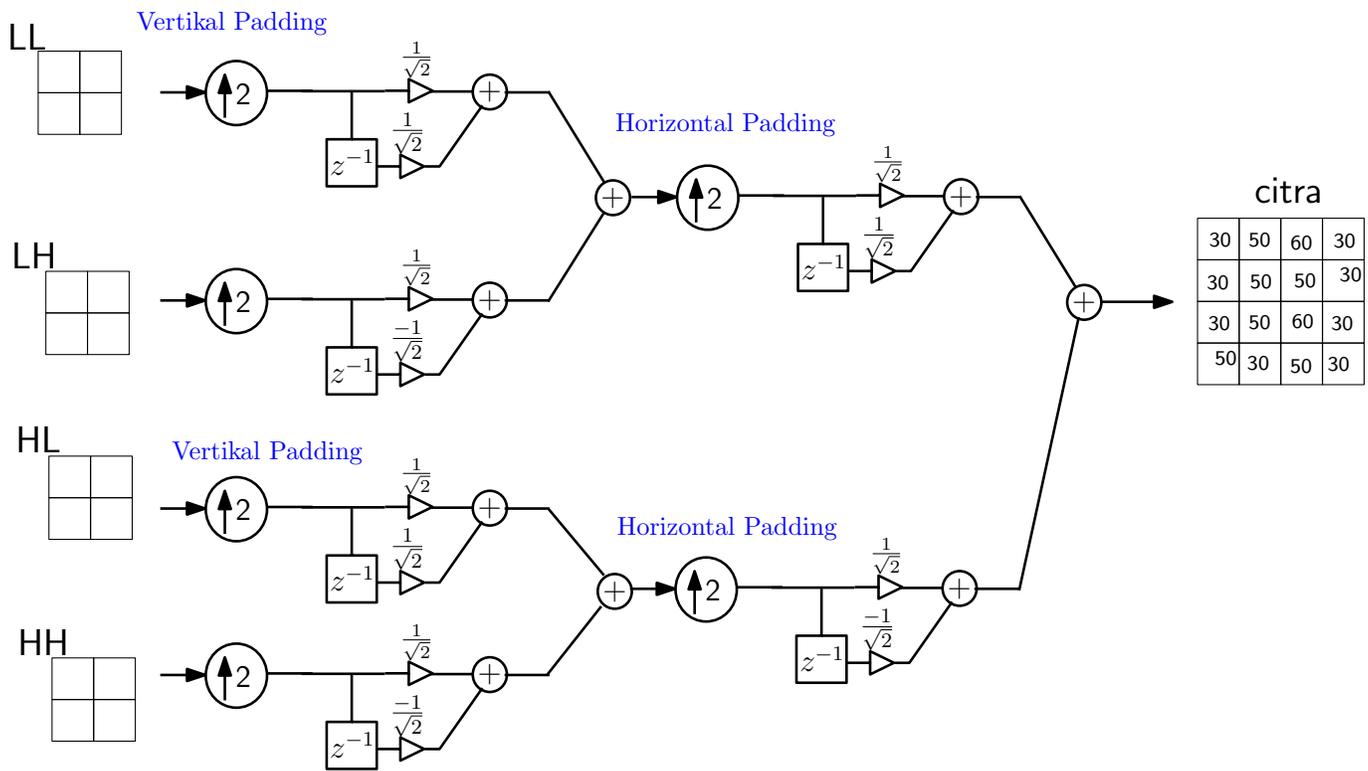
**Contoh:** Diberikan filter

$$g_1(n) = [1 \ 1 \ 2 \ 2]/\sqrt{10}$$

Tentukan  $h_1(n)$ ,  $g_2(n)$ , dan  $h_2(n)$ !

**Jawab:...**

# Rincian Blok Diagram Rekonstruksi - Haar Wavelet



## Contoh Perhitungan Dekomposisi

**Contoh:** Diberikan filter

$$g_1(n) = [1 \ 1]/\sqrt{2}$$

serta citra dengan dimensi  $4 \times 4$ :

30	50	60	30
30	50	50	30
30	50	60	30
30	50	50	30

citra .

Tentukan  $LL$ ,  $LH$ ,  $HL$  dan  $HH$ !

**Jawab:...**

# Menghitung LL dan LH

...

30	50	60	30
30	50	50	30
30	50	60	30
30	50	50	30

citra

# Menghitung HL dan HH

...

30	50	60	30
30	50	50	30
30	50	60	30
30	50	50	30

citra .

# RESUME - Dekomposisi

30	50	60	30
30	50	50	30
30	50	60	30
30	50	50	30

citra

## Rekonstruksi - Contoh

Mari kita rekonstruksi **LL**, **LH**, **HL** dan **HH** untuk menjadi citra semula!

# Rekonstruksi - Lanjutan

...

## Latihan

Diberikan filter

$$g_1(n) = [1 \ 1]/\sqrt{2}$$

serta citra dengan dimensi  $4 \times 4$ :

20	50	60	20
30	40	50	30
20	50	60	20
30	40	50	30

citra .

**Tentukan:**

- $LL$ ,  $LH$ ,  $HL$  dan  $HH$ !
- Rekonstruksi** kembali  $LL$ ,  $LH$ ,  $HL$  dan  $HH$  tersebut untuk menghasilkan citra semula!

**Jawab:...**

# Jawab

...

# Jawab - lanjutan

...



## Bab 12

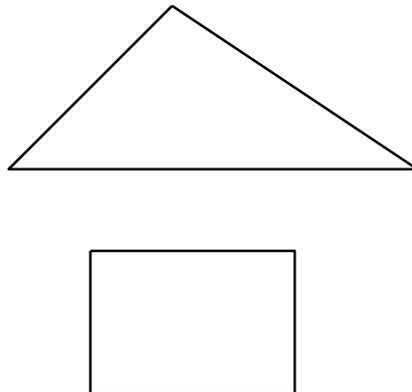
# Ekstraksi Ciri dengan GLCM

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa Memahami tentang Konsep Ekstraksi Ciri
- 2 Mahasiswa Memahami tentang Teknik Dasar Ekstraksi Ciri dengan GLCM pada citra Greyscale

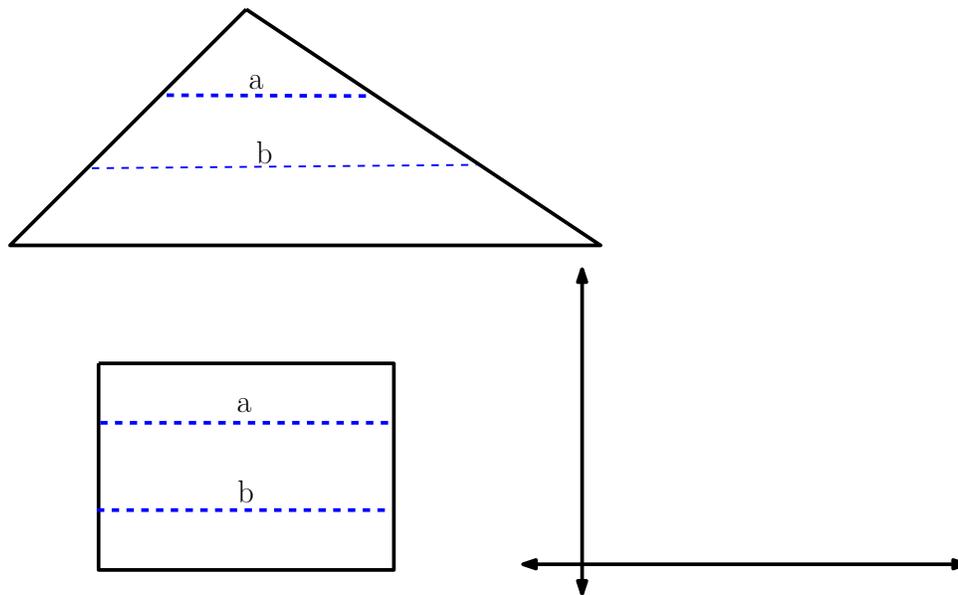
## Ekstraksi Ciri

- 1 Ekstraksi Ciri adalah teknik atau proses untuk **mengambil ciri** dari suatu objek (Audio, Citra, Video, Data)
- 2 Ekstraksi ciri menghasilkan satu set ciri yang disebut dengan **Vektor Ciri**
- 3 Perhatikan gambar segitiga dan dan segiempat berikut: **Apa** yang bisa diambil sebagai ciri dari keduanya? **Apa pula** yang tidak bisa?



## Plot Ciri

- 1 Set ciri dapat diplot
- 2 Objek berbeda akan berada pada cluster berbeda (terpisah)
- 3 Jika objek berbeda berada dalam cluster yang sama, maka ekstraksi ciri gagal!
- 4 Akan diplot set ciri dari 2 gambar berikut:



## Ekstraksi Ciri yang baik

- 1 Ekstraksi ciri yang baik ditandai dengan:
- 2 Objek **berbeda** menghasilkan **set ciri berbeda**
- 3 Ekstraksi ciri yang baik tidak terpengaruh oleh gangguan seperti rotasi, resizing, translasi dari citra.

Salah satu teknik ekstraksi ciri yang baik adalah GLCM (**Gray Level Co-occurrence Matrix**).

- Teknik GLCM ini berguna untuk membedakan tekstur pada citra



# GLCM

- 1 GLCM diusulkan tahun **1973**<sup>1</sup>
- 2 Oleh Robert Haralick
- 3 Profesor pada Jurusan **Computer Science**, City University of New York (CUNY)
- 4 Fungsi GLCM ini ditanamkan pada fungsi **graycomatrix** pada **Octave** dan **Matlab**



**Robert Haralick**, Computer Science, CUNY.

---

<sup>1</sup>Terbit dengan judul: **Textural Features for Image Classification**, IEEE Transactions on Cybernetics; 1973

## Ekstraksi ciri dengan GLCM

Ekstraksi Ciri citra Greyscale dengan Matriks GLCM terdiri dari 2 tahapan yaitu:

- 1 Pembentukan matriks GLCM, dan
- 2 Pengambilan ciri

## Pembentukan matriks GLCM

Pembentukan matriks GLCM terdiri dari beberapa langkah yaitu:

- 1 Kuantitasi
- 2 Penentuan **sudut** dan **jarak** ketetanggaan
- 3 Tabulasi ketetanggaan

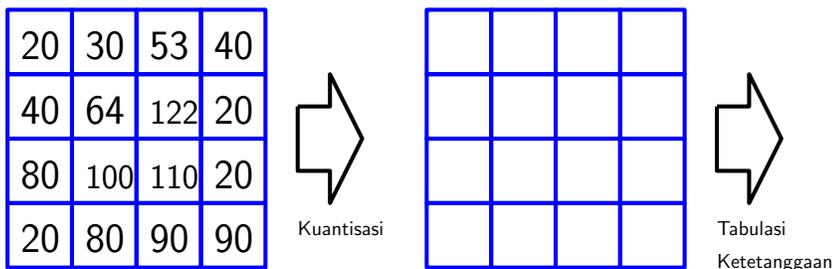
Untuk keperluan ilustrasi, kita tinjau **citra grey** dimensi  $4 \times 4$  berikut:

20	30	53	40
40	64	122	20
80	100	110	20
20	80	90	90

## Kuantisasi

- 1 Kuantisasi **bertujuan** untuk mengurangi rentang sinyal
- 2 **Misal** kita lakukan kuantisasi dengan rentang **32**
- 3 Dengan rentang ini maka:
  - Nilai 0-31  $\rightarrow$  0
  - Nilai 32-63  $\rightarrow$  1
  - Nilai 64-127  $\rightarrow$  2
  - Nilai 128-159  $\rightarrow$  3
  - Nilai 160-191  $\rightarrow$  4
  - Nilai 192-223  $\rightarrow$  5
  - Nilai 223-255  $\rightarrow$  6

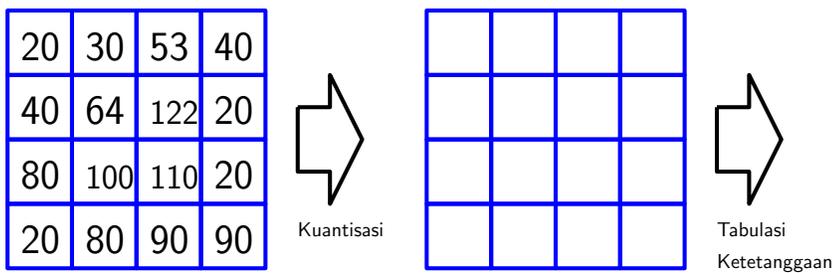
Hasil kuantisasi:



## Penentuan Jarak dan Sudut

- 1 Penentuan **Jarak** dan **Sudut** adalah untuk menentukan ketetanggaan
- 2 **Jarak**: 1 piksel, 2 piksel, ...
- 3 **Sudut**:  $0^0$ ,  $45^0$ ,  $90^0$ ,  $135^0$ ,  $180^0$ ,  $225^0$ ,  $270^0$ ,  $315^0$ .
- 4 **Contoh**: Jarak 1 piksel, sudut  $0^0$
- 5 Titik (1,1) bertetangga dengan (1,2), titik (1,2) bertetangga dengan ...

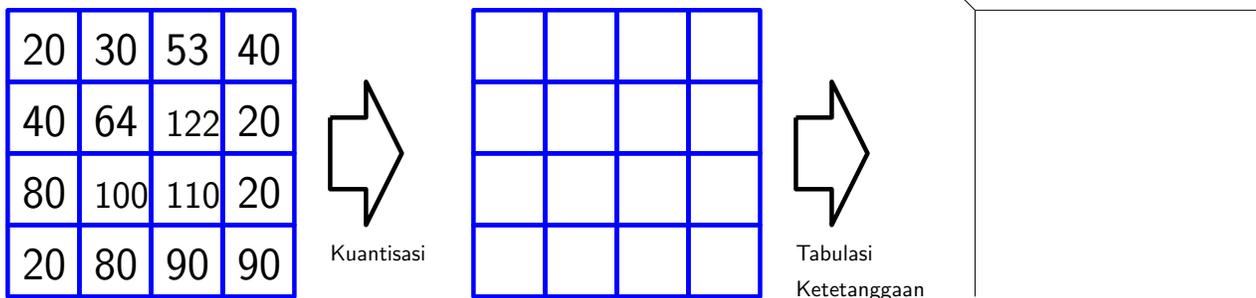
Hasil kuantisasi:



## Tabulasi Ketetanggaan

- 1 Setelah **Jarak** dan **Sudut** telah ditentukan (**misal**: Jarak 1 piksel, sudut  $0^{\circ}$ )
- 2 Berikutnya adalah membuat tabulasi ketetanggaan:

Hasil Tabulasi Ketetanggaan:



Tabel tabulasi ketetanggaan inilah yang disebut sebagai **Matriks GLCM**

## Tahapan ekstraksi ciri GLCM

- 1 Setelah pembentukan matriks GLCM selesai, berikutnya dilakukan tahapan ekstraksi ciri
- 2 Tahapan ekstraksi ciri terdiri dari:
  - **Mirroring** atau **Simetrifikasi**
  - **Normalisasi**
  - Perhitungan Ciri: **Energi**, **Kontras**, **Homogeneity**, **Korelasi**, **Dissimilarity**, dan Angular Second Moment (ASM), **Entropy**, dll

## Mirroring

- 1 Mirroring atau Simetrifikasi adalah menjumlahkan matriks GLCM dengan transpose-nya
- 2 Identy adalah: **jika** piksel (1,1) bertetangga dengan piksel (1,2) maka ini sama dengan piksel (1,2) yang bertetangga dengan piksel (1,1)
- 3 **Contoh:**

Proses Mirroring:

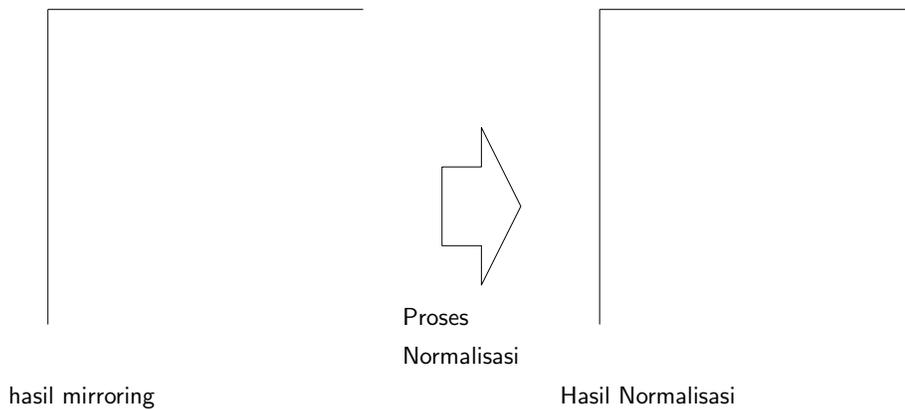
$$\square + \square = \square$$

# Normalisasi

❶ Normalisasi idenya adalah membatasi nilai matriks sehingga menjadi antara **0** sampai **1**

❷ 
$$x(i, j) \Leftarrow \frac{x(i, j)}{\sum_{i, j} x(i, j)}$$

Contoh Proses Normalisasi:



## Tahapan ekstraksi ciri GLCM

- 1 Setelah diperoleh matriks hasil Normalisasi, maka matriks ini digunakan untuk menghitung atau mengekstraksi ciri-ciri, **antara lain:**
  - **Energi, Kontras, Homogeneity, Dissimilarity, Korelasi,** dan Angular Second Moment (ASM), **Entropy, dll**
- 2 **Energi:**  $E = \sum_{i,j} [x(i,j)]^2$
- 3 **Kontras:**  $C = \sum_{i,j} [i - j]^2 x(i,j)$
- 4 **Homogeneity:**  $H = \sum_{i,j} \frac{x(i,j)}{1+|i-j|}$
- 5 **Dissimilarity:**  $D = \sum_{i,j} |i - j| x(i,j)$
- 6 **dst ...**

## Contoh Perhitungan Energi

- 1 **Energi:**  $E = \sum_{i,j} [x(i,j)]^2$
- 2 Matriks Normalisasi sebelumnya ...
- 3 ...

## Contoh Perhitungan Kontras

- 1 **Kontras:**  $C = \sum_{i,j} [i - j]^2 x(i, j)$
- 2 Matriks Normalisasi sebelumnya ...
- 3 ...

## Contoh Perhitungan Homogeneity

- 1 Homogeneity:  $H = \sum_{i,j} \frac{x(i,j)}{1+|i-j|}$
- 2 Matriks Normalisasi sebelumnya ...
- 3 ...

## Contoh Perhitungan Dissimilarity

- 1 **Dissimilarity:**  $D = \sum_{i,j} |i - j| x(i, j)$
- 2 Matriks Normalisasi sebelumnya ...
- 3 ...



## Bagian III

# PENGANTAR VIDEO DIGITAL



**Bab 13**

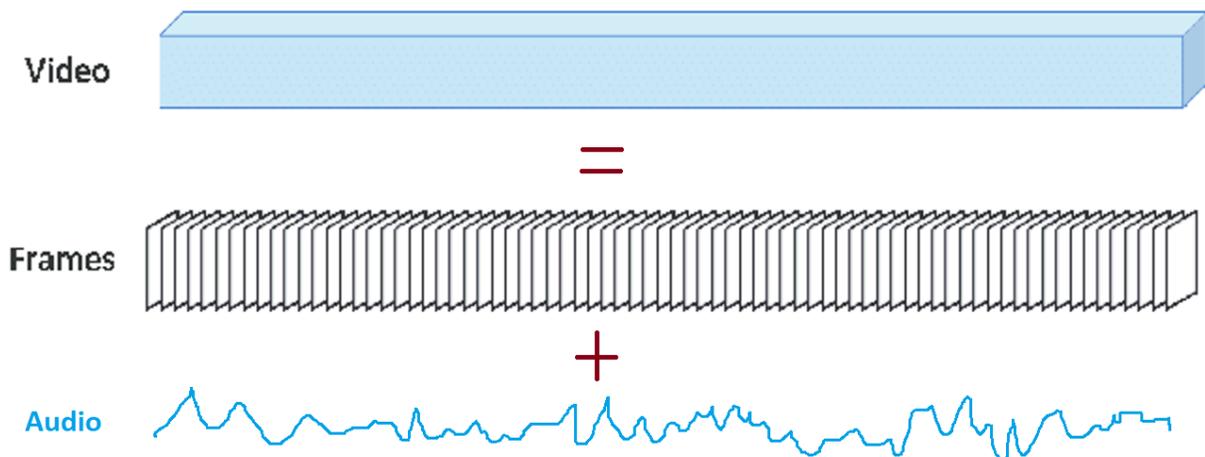
# **Video Digital**

## Tujuan yang ingin dicapai pada pertemuan ini

- 1 Mahasiswa Memahami Struktur Video Digital
- 2 Mahasiswa Memahami Parameter Video Digital, Resolusi Spasial, Resolusi Temporal, Ukuran file Video dan Kompresi Video

## Video Digital

- 1 Video berasal dari bahasa Latin yakni **Videre**
- 2 **Videre = to see** = melihat atau menonton
- 3 Video terdiri atas 2 komponen yaitu **streaming citra** atau **frames** dan **audio**

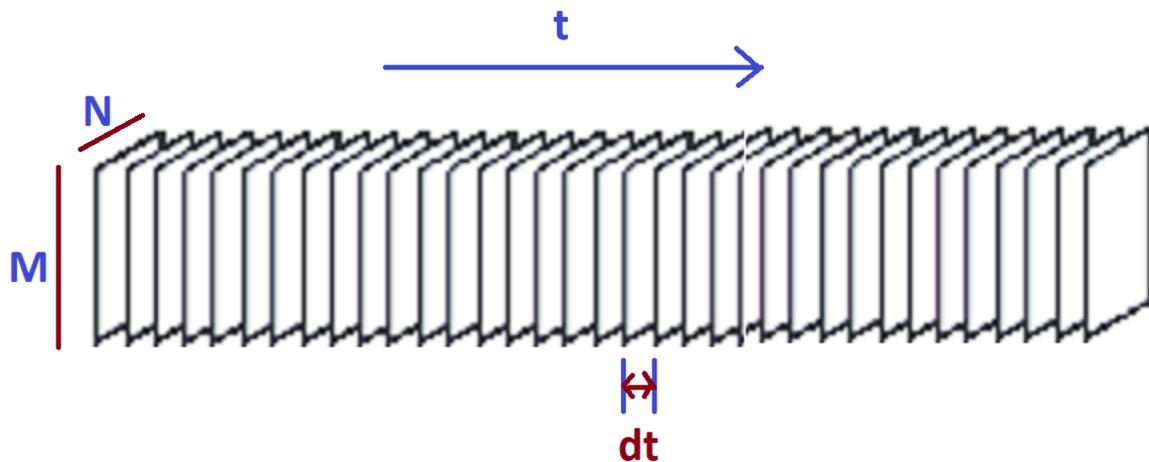


# Video Digital

- 1 Ukuran **file video** (KB atau MB atau GB) oleh karena itu adalah **ukuran frames + ukuran audio**
- 2 Biasanya **ukuran frames** >> **ukuran audio**
- 3 Oleh karena itu, pembicaraan tentang video biasanya **difokuskan** pada pembicaraan tentang **frames**

## Resolusi Video Digital

- 1 Terdapat 2 resolusi video, yakni:
- 2 **Resolusi Spasial** dan **Resolusi Temporal**
- 3 Resolusi Spasial menyatakan ukuran panjang kali lebar 1 frame
- 4 Resolusi Temporal menyatakan **jarak waktu** dari suatu frame ke frame berikutnya



## Frames Per Second

- 1 FPS menyatakan resolusi temporal video
- 2 **Semakin tinggi** nilai FPS, semakin halus video terlihat
- 3 **Semakin rendah** nilai FPS, video akan terlihat **lebih patah-patah**.
- 4 FPS **makin tinggi** maka ukuran video akan **semakin besar**.
- 5 **Standar** perekaman kamera HP dan Webcam adalah **30 FPS**
- 6 Untuk **kamera digital** yang **high spec**, biasanya nilai FPSnya adalah **60**.



Webcam



DSLR

## Contoh Menghitung Ukuran File Video

- 1 Suatu Webcam menangkap video dengan format RGB 24 bit, **Resolusi Spasial 600 × 800** piksel, Resolusi Temporal **30 FPS**. Tentukan Ukuran Video dengan durasi **1 menit!**
- 2 **JAWAB:**

## Latihan 1:

- 1 Suatu Webcam menangkap video dengan format RGB 24 bit, **Resolusi Spasial 768 × 1024** piksel, Resolusi Temporal **30 FPS**. Tentukan Ukuran Video dengan durasi **1 menit!**
- 2 **JAWAB:**

## Latihan 2:

- 1 Suatu kamera DSLR menangkap video dengan format RGB 24 bit, **Resolusi Spasial full HD** (lihat **Slide 08**), Resolusi Temporal **60 FPS**. Tentukan Ukuran Video berdurasi **1 menit!**
- 2 **JAWAB:**

# Kompresi Video

- 1 Permasalahan utama pada Video Digital adalah **besar**nya **ukuran file**
- 2 Berbagai upaya dilakukan untuk mengecilkan ukuran video ⇒ **Kompresi**
- 3 Kompresi video dapat dilakukan **secara spasial** atau **temporal** atau **kombinasi keduanya**

## Kompresi Spasial

- 1 Kompresi Spasial adalah kompresi yang dilakukan pada setiap frame pada video
- 2 Contoh, misal suatu video dengan FPS 30, setiap framenya memiliki ukuran 1 MB, maka dalam **1 detik** ukuran video adalah  $30 \times 1 \text{ MB} = \mathbf{30 \text{ MB}}$ .
- 3 Jika **setiap** frame yang berukuran **1 MB** tersebut **dikompresi** menjadi **100 KB**, maka ukuran video setelah dikompres adalah :  $30 \times 100 \text{ KB} = 3000 \text{ KB} = \mathbf{3 \text{ MB}}$ .

## Kompresi Temporal

- ➊ Biasanya beberapa **frame berurutan** pada video isinya **hampir sama**
- ➋ Oleh karena itu **salah satu** atau **beberapa frame dapat dibuang** untuk mengurangi ukuran video
- ➌ Pada proses pemutaran video, frame yang telah dibuang dapat dibuat kembali dengan melakukan **interpolasi** dari **frame antara**

## Kompresi Temporal

- 1 **Contoh:** Misalkan **Frame 1**, **Frame 2** dan **Frame 3** isinya hampir sama.
- 2 Untuk mengurangi ukuran video, maka **frame 2** dibuang.
- 3 Dengan membuang frame 2, maka kita sekarang hanya menyimpan 2 frame alih-alih 3 frame.
- 4 Ketika akan diputar videonya, maka Frame 2 dapat dibuat kembali dengan **interpolasi Frame 1** dan **Frame 3**.
- 5 Interpolasi dapat dilakukan misalnya dengan mengambil **nilai rata-rata:**

$$Frame\ 2 = \frac{Frame\ 1 + Frame\ 3}{2}$$

## Ilustrasi Kompresi Temporal

- ① Diberikan 3 frame berurutan dari suatu video **Greyscale 8 bit** dengan ukuran  $2 \times 2$  sebagai berikut:

$$\begin{bmatrix} 100 & 102 \\ 110 & 130 \end{bmatrix}, \quad \begin{bmatrix} 101 & 105 \\ 110 & 132 \end{bmatrix}, \quad \begin{bmatrix} 102 & 106 \\ 114 & 134 \end{bmatrix}$$

- ② **Pertanyaan:**
- ③ Berapa Byte-kah ukuran satu frame?
- ④ Berapa Byte-kah ukuran video dengan 3 frame tersebut?
- ⑤ Jika Frame kedua dibuang, berapa Byte-kah ukuran video sekarang?
- ⑥ Jika Frame kedua yang dibuang akan direkonstruksi kembali, tentukan nilai dari frame kedua!
- ⑦ **Jawab:**

# Ilustrasi Kompresi Temporal

① Jawab (lanjutan):

## Kompresi Lossless dan Lossy

- ➊ Karena ukuran file yang besar, format penyimpanan video **hampir selalu** dalam format terkompresi.
- ➋ Format video terkompresi yang populer contohnya adalah : **h.264, avi, mp4, mpeg, mov**
- ➌ Ada 2 kategori kompresi yakni **Lossless** dan **Lossy**
- ➍ Kompresi Lossless artinya, suatu video dikompres, kemudian ketika didekompres, maka hasilnya adalah **sempurna** seperti video semula
- ➎ Kompresi Lossy artinya, suatu video dikompres, kemudian ketika didekompres, maka hasilnya **tidak persis sama** dengan video semula.

## Kompresi Lossless dan Lossy

- 1 Contoh **Lossless compression**: h.264, avi
- 2 Contoh **Lossy compression**, mp4, mpeg, mov
- 3 Kualitas video dengan **Lossless compression** adalah lebih baik dari pada **Lossy compression**
- 4 Namun **Lossless compression** memiliki ukuran file yang lebih besar dibandingkan dengan ukuran file pada **Lossy compression**

## Penutup: Pertanyaan Review

- 1 Sebutkan 2 komponen pada Video!
- 2 Sebutkan dan jelaskan 2 macam resolusi pada video!
- 3 Parameter apa yang umum digunakan untuk mendeskripsikan resolusi temporal dari video?
- 4 FPS 30 artinya adalah ...
- 5 Cara untuk mengurangi ukuran file video adalah ...
- 6 Kompresi spasial dilakukan dengan cara ...
- 7 Kompresi Temporal dilakukan dengan cara ...
- 8 Frame yang dibuang pada kompresi temporal direkonstruksi dengan cara ...
- 9 Kompresi Lossless artinya ...
- 10 Kompresi Lossy artinya ...
- 11 Contoh format Kompresi Lossless video ...
- 12 Contoh format Kompresi Lossy video ...

ISBN 978-623-97190-8-1

