



# Video Processing

## Fundamentals

**Martin C.T. Manullang**

Institut Teknologi Sumatera

**Website:** mctm.web.id/course/if25-40305

2025

# OUTLINE

1. From Image to Video
2. Historical Keypoints
3. Digital Representation
4. Real World Challenges

# From Image to Video

# Dari Gambar ke Video - Membangun dari Fondasi Anda

## Pertanyaan Kunci

Apa perbedaan utama antara **1.000 gambar** dan **video 30 detik**?

**Aktivitas:** Silahkan rangkum jawaban anda dalam 1-2 kalimat!

*Mari kita diskusikan bersama...*

# Manfaatkan Pengetahuan Pemrosesan Gambar

Selama beberapa minggu terakhir, kita telah mempelajari berbagai teknik pemrosesan gambar:

1. Filter (penghalusan, penajaman)
2. Transformasi geometri
3. Operasi titik dan histogram
4. Ruang warna (RGB, HSV, YCbCr)
5. Deteksi tepi dan kontur
6. Morfologi dan segmentasi

## Kunci Pemahaman

Semua teknik ini dapat diterapkan pada **setiap frame** dalam video!

# Video: Dimensi Keempat dalam Multimedia

Video = Gambar + **Waktu**

Jika gambar memiliki 3 dimensi (lebar, tinggi, warna), maka:

## Gambar:

- Dimensi X (lebar)
- Dimensi Y (tinggi)
- Dimensi C (warna)

## Video:

- Dimensi X (lebar)
- Dimensi Y (tinggi)
- Dimensi C (warna)
- **Dimensi T (waktu)**

# Pengenalan Frame Rate (FPS)

## Definisi Frame Rate

Frame per Second (FPS) adalah jumlah **gambar yang ditampilkan** dalam satu detik untuk menciptakan ilusi gerakan.

**Analogi sederhana:** FPS seperti flipbook - semakin banyak halaman per detik, gerakan semakin halus!

### FPS Umum:

1. 24 FPS: Film bioskop
2. 30 FPS: TV, YouTube standar
3. 60 FPS: Video game, olahraga

### Konsep Penting:

- Sampling waktu, bukan ruang
- FPS lebih tinggi = gerakan lebih halus
- FPS lebih rendah = ukuran file lebih kecil

# FPS: Pencuplikan Waktu dalam Video

## Sampling Time vs Sampling Space

### Perbedaan Pencuplikan

Pada gambar kita mencuplik **ruang** (piksel), pada video kita juga mencuplik **waktu** (frame)!

### Diskusi kelompok (2-3 orang):

1. Apa yang terjadi jika FPS terlalu rendah?
2. Apa yang terjadi jika FPS terlalu tinggi?
3. Kapan kita membutuhkan FPS tinggi?

*Waktu diskusi: 3 menit, lalu presentasi 1 kelompok*

# Kompromi Resolusi: Kualitas Spasial vs Temporal

## Tantangan Penyimpanan

Dengan ruang penyimpanan terbatas, kita harus memilih: resolusi **tinggi** dengan FPS rendah, atau resolusi rendah dengan FPS **tinggi**?

### Kualitas Spasial Tinggi:

1. Resolusi 4K (3840x2160)
2. Detail gambar sangat tajam
3. FPS mungkin hanya 24-30
4. Cocok untuk: film, fotografi

### Kualitas Temporal Tinggi:

1. Resolusi 720p atau 1080p
2. FPS 60-120
3. Gerakan sangat halus
4. Cocok untuk: game, olahraga

# Latihan: Menghitung Kebutuhan Penyimpanan Video

**Skenario:** Anda ingin merekam video dengan spesifikasi berikut:

- Durasi: 30 detik
- Resolusi: 1920x1080 piksel
- Warna: 24 bit per piksel (3 byte)
- Frame rate: 30 FPS

## Tugas (Individu/Berpasangan)

Hitunglah ukuran file video **tanpa kompresi!** Gunakan rumus:

$$\text{Ukuran} = \text{Lebar} \times \text{Tinggi} \times \text{Byte/piksel} \times \text{FPS} \times \text{Durasi}$$

# Historical Keypoints in Video Technology

# Dari Gambar Diam ke Gambar Bergerak

## Awal Mula Ilusi Gerak

Manusia sudah mencoba menciptakan ilusi gerakan sejak **ribuan tahun** yang lalu, jauh sebelum teknologi modern ada!

### Penemuan-penemuan awal:

1. **Zaman Prasejarah:** Lukisan gua menunjukkan hewan dengan banyak kaki (ilusi gerakan)
2. **1824 - Thaumatrope:** Cakram berputar dengan gambar di dua sisi
3. **1834 - Zoetrope:** Silinder berputar dengan celah untuk melihat urutan gambar
4. **1872 - Eadweard Muybridge:** Fotografi kuda berlari dengan 12 kamera
5. **1878:** Membuktikan kuda mengangkat keempat kakinya saat berlari

**Prinsip dasar:** Persistensi penglihatan - mata kita "menahan" gambar sebentar setelah hilang!

# Era Film: Kelahiran Industri Perfilman

Tonggak penting di akhir abad ke-19:

1. **1888 - Louis Le Prince:** Film pertama "Roundhay Garden Scene" (2 detik)
2. **1891 - Thomas Edison:** Kinetoscope - mesin untuk menonton film secara individual
3. **1895 - Lumière Brothers:** Cinematograph - proyektor film untuk penonton banyak
4. **1895:** Pemutaran film publik pertama di Paris (10 film pendek)
5. **1927:** "The Jazz Singer" - film bicara pertama yang sukses komersial

## Fakta Menarik

Film awal hanya **16 FPS** dan hitam-putih. Penonton sudah merasa terpukau dengan ilusi geraknya!

# Revolusi Televisi: Video Masuk ke Rumah

## Dari Layar Lebar ke **Layar Kecil**

### Era Mekanis (1920-an):

- 1925: John Logie Baird - TV mekanis
- Gambar sangat kasar (30 baris)
- Menggunakan cakram berputar
- Tidak praktis untuk produksi massal

### Era Elektronik (1930-an):

- 1927: Philo Farnsworth - TV elektronik
- Menggunakan tabung katoda (CRT)
- Kualitas gambar lebih baik
- 1939: Siaran TV komersial pertama

**Dampak sosial:** Untuk pertama kali, orang bisa menonton peristiwa langsung dari rumah mereka!

# Standarisasi Video: NTSC, PAL, dan SECAM

## Mengapa Perlu Standar?

Berbagai negara mengembangkan sistem TV sendiri, sehingga dibutuhkan **standar internasional** agar peralatan kompatibel.

Tiga standar utama yang bertahan hingga era analog:

1. **NTSC (1953):** National Television System Committee
  - 525 baris, 30 FPS
  - Amerika Serikat, Jepang, Korea
2. **PAL (1967):** Phase Alternating Line
  - 625 baris, 25 FPS
  - Eropa, Australia, sebagian Asia (termasuk Indonesia)
3. **SECAM (1967):** Séquentiel Couleur à Mémoire
  - 625 baris, 25 FPS
  - Prancis, Rusia, beberapa negara Afrika

# Era Video Rumahan: VHS vs Betamax

Perang format tahun 1970-an yang legendaris:

## Betamax (Sony, 1975):

- Kualitas gambar lebih baik
- Ukuran kaset lebih kecil
- Durasi rekam lebih pendek (60 menit)
- Harga lebih mahal
- **Kalah di pasar**

## VHS (JVC, 1976):

- Kualitas gambar cukup baik
- Kaset lebih besar
- Durasi rekam lebih lama (120-240 menit)
- Harga lebih murah
- **Menang di pasar**

## Pelajaran Penting

Teknologi yang **lebih baik** tidak selalu menang - faktor harga dan praktikalitas juga penting!

# Revolusi Digital: Dari Analog ke Digital

## Lompatan Kuantum dalam **Kualitas Video**

Tonggak penting digitalisasi video:

1. **1982**: CD (Compact Disc) - audio digital pertama yang sukses
2. **1995**: DVD (Digital Versatile Disc) - video digital untuk rumahan
3. **1996**: MPEG-2 menjadi standar untuk DVD dan TV digital
4. **2003**: H.264/AVC - codec yang merevolusi streaming video
5. **2006**: Blu-ray Disc - video HD untuk rumahan

**Keuntungan digital:** Tidak ada degradasi kualitas saat disalin, mudah diedit, file lebih kecil dengan kompresi!

# Era Internet dan Streaming

## Perubahan Paradigma

Internet mengubah cara kita menonton video dari **broadcast** (satu arah) menjadi **on-demand** (kapan saja, di mana saja).

## Evolusi platform video online:

1. **2005:** YouTube diluncurkan - demokratisasi pembuatan konten
2. **2007:** Netflix mulai layanan streaming
3. **2010:** Instagram dan video mobile mulai populer
4. **2013:** H.265/HEVC - codec untuk video 4K
5. **2016:** TikTok - video pendek vertikal
6. **2020:** Pandemi COVID-19 mempercepat adopsi video call massal

*Saat ini, lebih dari 80% trafik internet adalah video!*

# Masa Depan Video: AI dan Beyond

Teknologi video terkini dan masa depan:

## Sudah Ada Sekarang:

1. Video 8K (7680 x 4320)
2. HDR (High Dynamic Range)
3. 120 FPS untuk gaming
4. AI upscaling
5. Virtual Reality (VR)

## Sedang Dikembangkan:

1. AI-generated video
2. Holographic displays
3. Brain-computer interface
4. 16K resolution
5. Real-time deepfake

## Pertanyaan Etis

Dengan kemampuan AI membuat video realistik yang palsu, bagaimana kita bisa **memverifikasi** kebenaran video di masa depan?

# Digital Representation of Video

# Masalah "Longsoran Data" dalam Video Digital

## Tantangan Utama

Video mentah (tanpa kompresi) menghasilkan file yang **sangat besar** dan tidak praktis untuk disimpan atau ditransmisikan.

### Mari kita buktikan dengan perhitungan:

Video HD (1920x1080) dengan 30 FPS selama 1 menit:

- $1920 \times 1080 \times 3 \text{ byte} \times 30 \text{ FPS} \times 60 \text{ detik} = 11.1 \text{ GB!}$
- Film 2 jam = lebih dari 1.3 TB
- Mustahil untuk streaming atau penyimpanan normal

*Solusinya? Kita perlu **kompresi!***

# Codec: Jantung Pemrosesan Video Digital

**Codec = Compressor + Decompressor**

## Definisi Codec

Codec adalah algoritma yang **mengompresi** video untuk penyimpanan dan **mendekompresi** untuk pemutaran.

## Contoh codec populer:

1. H.264 (AVC) - paling umum digunakan
2. H.265 (HEVC) - lebih efisien, file lebih kecil
3. VP9 - codec terbuka dari Google
4. AV1 - codec terbaru, sangat efisien

# Codec vs Container: Jangan Tertukar!

## Perbedaan Penting

**Codec** adalah cara kompresi video, sedangkan **Container** adalah format file yang menyimpan video, audio, dan metadata.

### Container (Format File):

- MP4 (.mp4)
- AVI (.avi)
- MKV (.mkv)
- MOV (.mov)

### Codec (Kompresi):

- H.264
- H.265
- VP9
- AV1

**Analogi:** Container seperti kotak, codec seperti cara kita melipat barang di dalam kotak!

# Aktivitas: Identifikasi Codec dan Container

## Aktivitas Berpasangan:

1. Buka file video di laptop/HP anda
2. Cek properties/informasi file
3. Identifikasi: apa container-nya? apa codec-nya?
4. Diskusikan: mengapa kombinasi tertentu digunakan?

### Pertanyaan Diskusi

Apakah file dengan ekstensi yang sama (misal .mp4) selalu menggunakan codec yang sama?

*Waktu diskusi: 5 menit*

# Dua Strategi Kompresi Video

## Kompresi **Spasial** vs **Temporal**

### Kompresi Intra-frame (Spasial):

- Kompresi *di dalam* satu frame
- Seperti kompresi JPEG pada gambar
- Menghilangkan redundansi dalam satu gambar
- Setiap frame dikompresi sendiri

### Kompresi Inter-frame (Temporal):

- Kompresi *antar* frame
- Memanfaatkan kesamaan frame berurutan
- Hanya menyimpan perbedaan
- Lebih efisien untuk video

# Kompresi Spasial: Fokus pada Satu Frame

## Prinsip Kompresi Intra-frame

Mengurangi ukuran data dengan menghilangkan informasi yang **berlebihan** atau **tidak terlihat** dalam satu frame.

### Teknik yang digunakan:

1. Transformasi DCT (seperti JPEG)
2. Kuantisasi untuk membuang detail halus
3. Pengkodean entropi (Huffman, arithmetic)

**Keuntungan:** Setiap frame dapat diakses independen

**Kerugian:** Kompresi tidak seefisien kompresi temporal

# Kompresi Temporal: Memanfaatkan Kesamaan Antar Frame

## Ide Kunci

Frame yang berurutan dalam video biasanya sangat **mirip**, jadi kita hanya perlu menyimpan **perbedaannya** saja!

### Analogi sederhana:

Bayangkan video seseorang berbicara:

- Frame 1: Simpan gambar lengkap
- Frame 2: Hanya simpan "mulut bergerak sedikit"
- Frame 3: Hanya simpan "mulut bergerak lagi"
- Frame 100: Simpan gambar lengkap lagi

Hasil: Ukuran file jauh lebih kecil!

# Tiga Jenis Frame: I, P, dan B

Untuk kompresi temporal yang efisien, video dibagi menjadi tiga jenis frame:

1. **I-frame (Intra-frame)**: Frame kunci yang berisi gambar lengkap
2. **P-frame (Predicted)**: Frame yang diprediksi dari frame sebelumnya
3. **B-frame (Bi-directional)**: Frame yang diprediksi dari frame sebelum dan sesudahnya

## Catatan Penting

Kita akan mempelajari detail teknis I, P, B frame di pertemuan berikutnya. Untuk sekarang, pahami konsep dasarnya!

# Analogi Sederhana: I, P, B Frame seperti Komik

Bayangkan Anda menggambar komik dengan 10 panel yang menunjukkan seseorang berjalan:

## Cara Tidak Efisien:

- Panel 1: Gambar lengkap orang
- Panel 2: Gambar lengkap orang (kaki sedikit maju)
- Panel 3: Gambar lengkap orang (kaki lebih maju)
- ...dan seterusnya

Butuh 10x usaha gambar lengkap!

## Cara Efisien (seperti video):

- Panel 1 (I): Gambar lengkap orang
- Panel 2 (P): "kaki kiri maju 5cm"
- Panel 3 (B): "kaki kanan maju 3cm"
- Panel 4 (P): "kaki kiri maju lagi"
- Panel 10 (I): Gambar lengkap lagi

Jauh lebih hemat!

# Mengapa Perlu Tiga Jenis Frame?

**Pertanyaan yang sering muncul:** Kenapa tidak pakai I-frame saja atau P-frame saja?

## 1. Mengapa perlu I-frame?

- Titik awal untuk dekoding (tanpa ini P dan B tidak bisa diproses)
- Memungkinkan jumping/seeking ke momen tertentu
- Recovery jika terjadi error transmisi
- Kualitas gambar tidak degradasi seiring waktu

## 2. Mengapa perlu P-frame?

- Lebih kecil dari I-frame (menghemat ruang)
- Lebih mudah diproses daripada B-frame
- Titik referensi untuk B-frame

## 3. Mengapa perlu B-frame?

- Kompresi paling efisien (ukuran terkecil)
- Menghasilkan kualitas terbaik untuk ukuran file tertentu

# Contoh Kasus Nyata: Video Seseorang Berbicara

**Skenario:** Video close-up wajah presenter berbicara selama 10 detik (300 frame dengan 30 FPS)

## Analisis Perubahan Antar Frame

- Background: hampir tidak berubah (99% sama)
- Wajah: sedikit berubah (gerakan kepala kecil)
- Mulut: berubah cukup signifikan (sedang berbicara)
- Mata: kadang berkedip

## Bagaimana I, P, B bekerja:

- **I-frame** (Frame 0, 150, 300): Simpan gambar lengkap
- **P-frame** (Frame 30, 60, 90...): "Background sama, mulut bergerak sedikit ke kiri"
- **B-frame** (Frame 1-29, 31-59...): "Mulut di antara posisi frame sebelum dan sesudah"

## Hasil

Ukuran file bisa **50-100x lebih kecil** dibanding menyimpan semua frame sebagai I-frame!

# Latihan Pemahaman: Identifikasi Frame Type

**Aktivitas (Berpasangan):** Untuk setiap skenario, tentukan jenis frame mana yang paling cocok:

1. Video adegan aksi dengan kamera bergerak cepat

- Banyak I-frame
- Banyak P-frame
- Banyak B-frame

2. Video konferensi dengan pembicara diam

- Banyak I-frame
- Banyak P-frame
- Banyak B-frame

3. Video timelapse awan bergerak lambat

- Banyak I-frame
- Banyak P-frame
- Banyak B-frame

# I-frame: Frame Kunci dalam Video

## Definisi I-frame

I-frame adalah frame yang mengandung gambar **lengkap** dan dapat didekripsi secara **independen** tanpa informasi dari frame lain.

## Karakteristik I-frame:

- Ukuran file paling besar
- Kualitas gambar terbaik
- Digunakan sebagai titik referensi
- Ditempatkan secara periodik
- Penting untuk seeking/jumping
- Mirip dengan bookmark

**Pertanyaan:** Mengapa kita perlu I-frame jika ukurannya besar?

# P-frame dan B-frame: Frame Prediksi

## P-frame (Predicted):

- Diprediksi dari frame *sebelumnya*
- Hanya menyimpan perbedaan
- Ukuran lebih kecil dari I-frame
- Lebih umum digunakan

## Contoh urutan:

I P P P I

## B-frame (Bi-directional):

- Diprediksi dari frame sebelum *dan* sesudah
- Kompresi paling efisien
- Ukuran paling kecil
- Memerlukan proses lebih kompleks

## Contoh urutan:

I B B P B B I

*Detail teknis akan dipelajari di pertemuan selanjutnya!*

# Visualisasi: Struktur GOP (Group of Pictures)

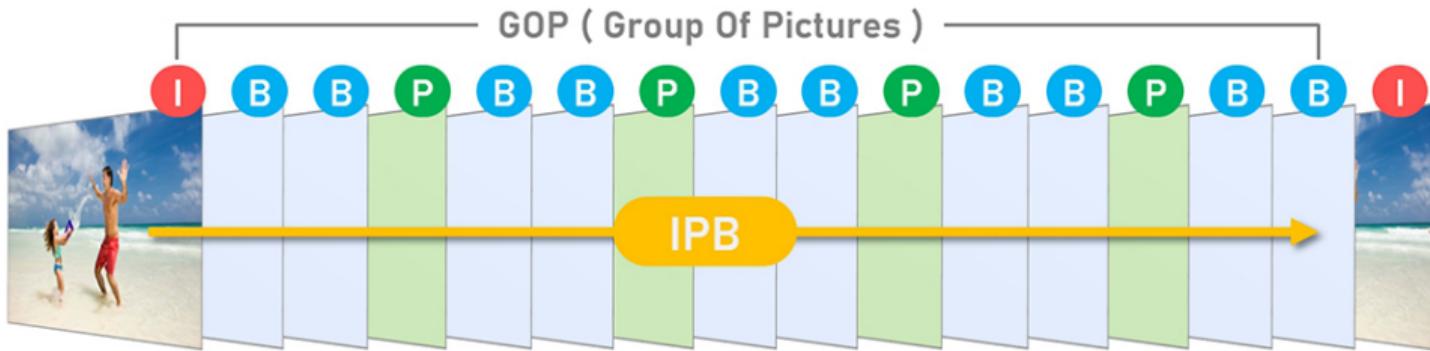
## Urutan I-P-B dalam Video

Contoh struktur GOP sederhana:

I B B P B B P B B I

### Hubungan Antar Frame

- I: Frame kunci (gambar lengkap)
- P: Diprediksi dari I
- B, B: Diprediksi dari I dan P
- I: Frame kunci berikutnya



Credit: Snapshot Canon Asia

# Real World Challenges in Video Processing

# Kompleksitas Komputasi: Mengapa Video Sangat Berat?

## Fakta Menarik

Memproses video **jauh lebih berat** daripada memproses audio atau gambar secara terpisah!

Mari kita bandingkan beban komputasi:

### Audio (1 menit):

1. 44.100 sampel per detik
2. 2 channel (stereo)
3. Total: ~5 MB data

### Video HD (1 menit):

1. 1920 x 1080 piksel
2. 30 frame per detik
3. Total: ~11.000 MB data

Video = Audio + Gambar kuadrat dalam hal **kompleksitas**!

# Mengapa "Audio + Gambar Kuadrat"?

## Penjelasan sederhana:

Bayangkan Anda harus menyelesaikan tugas berikut setiap detik:

1. Memproses 30 gambar HD (seperti mengedit foto)
2. Memproses audio yang menyertainya
3. Menyinkronkan audio dengan video
4. Mengompresi semuanya secara real-time
5. Mengirim hasilnya melalui internet

## Ilustrasi Beban Kerja

Jika memproses 1 gambar butuh 1 detik, maka memproses 1 detik video butuh **30 detik** hanya untuk gambarnya saja!

# Keterbatasan Penyimpanan dan Bandwidth

## Realitas **Rasio Kompresi** Codec

**Contoh nyata:** Video YouTube kualitas 1080p

- Ukuran mentah (tanpa kompresi): 11 GB per menit
- Ukuran setelah kompresi H.264: 100-200 MB per menit
- Rasio kompresi: sekitar **50:1 hingga 100:1**

### Pertanyaan Penting

Apa yang hilang saat kita mengompresi video sampai 100x lebih kecil?

# Trade-off Kompresi: Apa yang Dikorbankan?

Ketika codec mengompresi video hingga 100x lebih kecil, beberapa hal dikorbankan:

## Yang Hilang:

1. Detail halus pada tekstur
2. Variasi warna yang subtle
3. Ketajaman pada gerakan cepat
4. Informasi pada area gelap

## Yang Dipertahankan:

1. Struktur objek utama
2. Gerakan yang dapat dikenali
3. Warna dominan
4. Informasi penting untuk mata

**Prinsip codec:** Buang apa yang mata manusia tidak terlalu peka!

# Kebutuhan Real-Time: Tantangan Streaming dan Video Call

## Definisi Processing Real-Time

Video harus diproses dan dikirim **secepat** atau **lebih cepat** dari kecepatan pemutarannya.

### Contoh konkret:

Untuk video 30 FPS (frame per second):

- Setiap frame harus diproses dalam waktu maksimal  $1/30$  detik = 33 milidetik
- Termasuk: encode, transmit, decode, display
- Jika lebih lambat → video patah-patah atau buffering

*Inilah mengapa spesifikasi komputer penting untuk streaming!*

# Latensi vs Kualitas: Dilema yang Harus Dipilih

Dalam aplikasi video real-time, kita selalu menghadapi pilihan sulit:

## Prioritas Latensi Rendah:

1. Video call, gaming online
2. Kompresi lebih agresif
3. Resolusi lebih rendah
4. Frame rate mungkin turun
5. Response cepat lebih penting

## Prioritas Kualitas Tinggi:

1. Streaming film, YouTube
2. Kompresi lebih hati-hati
3. Resolusi maksimal
4. Frame rate stabil
5. Buffering dapat diterima

Tidak Ada Solusi Sempurna

Kita harus memilih sesuai **kebutuhan aplikasi!**

# Aktivitas: Analisis Trade-off dalam Aplikasi Nyata

## Diskusi Kelompok (3-4 orang):

Untuk setiap skenario, tentukan prioritas (latensi atau kualitas) dan jelaskan alasannya:

1. Dokter melakukan operasi jarak jauh dengan robot
2. Menonton pertandingan sepak bola live streaming
3. Video conference rapat kantor
4. Menonton film Netflix
5. Main game online multiplayer

*Waktu diskusi: 5 menit, presentasi 2 kelompok*

# Motion Artifacts: Ketika Kompresi Bertemu Gerakan Cepat

## Apa itu Motion Artifacts?

Gangguan visual yang muncul saat video dengan **gerakan cepat** dikompresi terlalu agresif.

### Contoh yang sering kita lihat:

1. Blocking: gambar seperti kotak-kotak saat kamera bergerak cepat
2. Blurring: objek bergerak cepat menjadi kabur
3. Ghosting: bayangan ganda pada objek bergerak
4. Banding: gradasi warna tidak halus

**Mengapa ini terjadi?** Codec tidak punya cukup data untuk memprediksi perubahan mendadak dengan akurat!

# Demonstrasi Motion Artifacts

## Kapan Motion Artifacts Muncul?

**Skenario yang rentan:**

**Gerakan Kamera:**

- Pan cepat (geser horizontal)
- Zoom in/out tiba-tiba
- Kamera goyang (shaky cam)

**Gerakan Objek:**

- Mobil balap melaju
- Bola dilempar dengan keras
- Orang berlari sprint

**Solusi**

Gunakan bitrate lebih tinggi atau frame rate lebih tinggi untuk mengurangi artifacts!

# Aplikasi Video Processing dalam Kehidupan Sehari-hari

Video processing ada di mana-mana! Mari kita lihat berbagai domain aplikasinya:

1. **Entertainment & Media:** YouTube, Netflix, TikTok
2. **Komunikasi:** Zoom, Google Meet, WhatsApp video call
3. **Keamanan:** CCTV, surveillance systems
4. **Medis:** Endoskopi, CT scan, MRI video
5. **Otomotif:** Kamera mundur, dashcam, autonomous driving
6. **Pendidikan:** E-learning, recorded lectures

*Setiap domain punya kebutuhan dan tantangan unik!*

# Streaming Video: Tantangan dan Solusi

## Kebutuhan Utama Streaming

Video harus dapat **dikirim** dan **diputar** secara bersamaan tanpa menunggu download selesai.

### Teknologi yang digunakan:

- 1. Adaptive bitrate streaming
- 2. Buffering untuk stabilitas
- 3. Codec efisien (H.265, AV1)
- 4. CDN (Content Delivery Network)
- 5. Prediksi bandwidth
- 6. Multiple quality levels

**Contoh:** Netflix otomatis menurunkan kualitas saat internet lambat untuk mencegah buffering.

# Surveillance dan Keamanan: Video 24/7

## Tantangan khusus sistem CCTV:

### Kebutuhan Penyimpanan:

- Rekam 24 jam per hari
- Multiple kamera (puluhan)
- Simpan 30-90 hari
- Butuh kompresi sangat tinggi

### Kebutuhan Analisis:

- Deteksi gerakan
- Pengenalan wajah
- Tracking objek
- Alert otomatis

## Inovasi Terkini

Sistem CCTV modern menggunakan **AI** untuk hanya menyimpan frame yang ada aktivitas mencurigakan!

# Autonomous Vehicles: Video Processing untuk Keselamatan

## Tantangan Kritis

Mobil otonom harus memproses video dari multiple kamera secara **real-time** untuk keputusan yang menentukan keselamatan.

### Kebutuhan spesifik:

1. Latensi ultra-rendah (< 100 ms)
2. Processing 8+ kamera simultan
3. Deteksi objek: pejalan kaki, kendaraan, rambu
4. Estimasi jarak dan kecepatan
5. Bekerja dalam berbagai kondisi cahaya
6. Reliability 99.9999

*Ini adalah salah satu aplikasi video processing paling demanding!*