

✓ Simulasi Monte Carlo

Simulasi Monte Carlo adalah sebuah metode komputasi yang digunakan untuk menghitung properti suatu kejadian probabilistik dengan menggunakan proses random sampling berkali-kali. Sebuah permasalahan probabilistik mungkin saja bisa diselesaikan secara analitis dengan bentuk tertutup, namun bisa sangat kompleks. Simulasi Monte Carlo kemudian menerapkan konsep randomness untuk menyelesaikan permasalahan seperti ini secara aproksimasi.

Sebagai contoh, Anda diberikan permasalahan berikut:

Sebuah dadu terdiri dari 6 buah mata nilai $\{1, 2, 3, 4, 5, 6\}$. Masing-masing mata nilai mempunyai peluang yang sama untuk muncul dalam sebuah lemparan dadu (equally likely). Seseorang kemudian melakukan eksperimen pelemparan dadu dengan cara **melempar dadu berkali-kali hingga angka genap muncul**. Diasumsikan kejadian lemparan sebuah dadu tidak bergantung dengan lemparan sebelumnya (independent). **Berapakah ekspektasi (rata-rata atau harapan) banyaknya lemparan dadu yang dilakukan hingga muncul angka genap pada akhirnya?**

Permasalahan di atas dapat diselesaikan secara analitis dengan cara berikut:

Pertama, kita perlu menghitung terlebih dahulu berapa peluang munculnya angka ganap pada sebuah lemparan dadu. Kejadian ini dihitung dengan:

$$P(\{2, 4, 6\}) = P(\{2\}) + P(\{4\}) + P(\{6\}) = \frac{3}{6} = 0,5.$$

Kedua, kita definisikan sebuah variabel acak X yang menyatakan "**banyaknya lemparan dadu yang dilakukan hingga muncul nilai genap (termasuk lemparan terakhir yang muncul nilai genap)**". Misal, kita notasikan juga $p = P(\text{Genap}) = p(\{2, 4, 6\})$. Nilai variabel acak X adalah selalu positif:

$$\begin{aligned} P(X = 1) &= p \\ P(X = 2) &= (1 - p) \times p \\ P(X = 3) &= (1 - p) \times (1 - p) \times p = (1 - p)^2 \times p \\ &\dots \\ P(X = n) &= (1 - p) \times \dots \times (1 - p) \times p = (1 - p)^{n-1} \times p \end{aligned}$$

Terakhir, nilai ekspektasi atau harapan dari X , yang dinotasikan dengan $\mathbf{E}[X]$, dihitung dengan:



$$\begin{aligned}
 \mathbf{E}[X] &= \sum_{i=1}^{\infty} i \cdot P(X = i) \\
 &= \sum_{i=1}^{\infty} i \cdot (1 - p)^{i-1} \cdot p \\
 &= \frac{1}{1 - p}
 \end{aligned}$$

Jadi, ekspektasi banyaknya lemparan dadu hingga muncul genap pertama kali adalah $1/(1 - p) = 1/(1 - 0,5) = 2$.

Permasalahan di atas juga bisa diselesaikan secara aproksimasi menggunakan Simulasi Monte Carlo. Sebuah eksperimen merefleksikan kegiatan melempar dadu berkali-kali hingga stop ketika bertemu sebuah lemparan yang menghasilkan nilai genap. Eksperimen ini dilakukan berkali-kali sebanyak N kali, dengan N biasanya harus cukup besar. Setelah itu, kita hitung rata-rata banyaknya lemparan dadu untuk semua eksperimen. Proses ini direfleksikan dengan kode Python berikut:

```
import random

# berikut adalah cara "kocok" dadu secara random; dengan masing-masing
# mata dadu mempunyai peluang yang sama
# jika Anda jalankan hal ini berkali-kali, hasilnya akan berbeda
mata_dadu = random.randint(1, 6)
print(mata_dadu)
```

4

```
# Simulasi Monte Carlo untuk menjawab:
# Berapakah ekspektasi (rata-rata atau harapan) banyaknya
# lemparan dadu yang dilakukan hingga muncul angka genap pada akhirnya?

N = 1000 # banyaknya eksperimen

hasil = []
for _ in range(N):

    # sekali eksperimen lempar dadu berkali-kali hingga muncul genap
    mata_dadu = random.randint(1, 6)
    outcomes = [mata_dadu]
    while mata_dadu % 2:
        mata_dadu = random.randint(1, 6)
        outcomes.append(mata_dadu)

    hasil.append(len(outcomes))
```

```
print(sum(hasil) / len(hasil)) # hitung rataan
```

```
1.983
```

Anda akan melihat bahwa hasil dari potongan program di atas adalah ≈ 2 .

✓ Soal A (30 poin)

Buatlah program Simulasi Monte Carlo untuk menjawab soal berikut:

Berapa ekspektasi banyaknya lemparan dadu yang dilakukan agar jumlah nilai mata dadu yang dihasilkan minimal 12?

Pada permasalahan ini, contoh beberapa eksperimen adalah seperti:

1. 2, 1, 2, 5, 6 $\rightarrow$ 5 kali lemparan dengan total 16
2. 6, 6 $\rightarrow$ 2 kali lemparan dengan total 12
3. ...

```
# Tulis program Anda di sini
```

✓ Soal B (70 poin)

Buatlah program Simulasi Monte Carlo untuk menjawab soal berikut:

Misal, kita melakukan eksperimen lemparan dadu dan berhenti ketika muncul angka 6. Jika diketahui kita melakukan sebanyak 5 lemparan dadu (lemparan yang terakhir bernilai 6), berapakah ekspektasi banyaknya angka ganjil yang muncul?

```
# Tulis program Anda di sini
```


Exploratory Data Analysis (EDA)

Dataset ini berisi informasi penjualan tiket bioskop untuk berbagai film yang tayang di beberapa kota besar di Indonesia. Data dikumpulkan dari tanggal 1 hingga 20 Juni 2025.

Setiap data terdiri dari informasi berikut:

🔍 Deskripsi Kolom Dataset (Film & Bioskop)

- `judul_film` : Judul film yang ditayangkan.
- `genre` : Kategori genre film.
- `harga_tiket` : Harga satu tiket film dalam Rupiah.
- `jumlah_tiket` : Jumlah tiket yang terjual untuk penayangan tertentu.
- `tanggal` : Tanggal penayangan film dalam format YYYY-MM-DD.
- `kota` : Kota tempat film ditayangkan.
- `jam_tayang` : Waktu penayangan film.
- `studio` : Kode studio tempat film ditayangkan.

JALANKAN KODE DIBAWAH INI DAN JANGAN UBAH DATA SAMA SEKALI.

```
data = [
    {'judul_film': 'Train to Busan', 'genre': 'Drama', 'harga_tiket': 70000, 'jumlah_tiket': 121, 'tanggal': '2025-06-12', 'kota': 'MAKASSA'},
    {'judul_film': 'The Avengers: Endgame', 'genre': 'Drama', 'harga_tiket': 55000, 'jumlah_tiket': 130, 'tanggal': '2025-06-15', 'kota': 'SURABAYA'},
    {'judul_film': 'Spiderman: Homecoming', 'genre': 'Sci-Fi', 'harga_tiket': 70000, 'jumlah_tiket': 113, 'tanggal': '2025-06-12', 'kota': 'JOGJA'},
    {'judul_film': 'Doctor Strange: Multiverse of Madness', 'genre': 'Animasi', 'harga_tiket': 50000, 'jumlah_tiket': 74, 'tanggal': '2025-06-08', 'kota': 'MEDAN'},
    {'judul_film': 'Frozen II', 'genre': 'Animasi', 'harga_tiket': 65000, 'jumlah_tiket': 132, 'tanggal': '2025-06-08', 'kota': 'MAKASSA'},
    {'judul_film': 'Avengers: Infinity War', 'genre': 'Drama', 'harga_tiket': 75000, 'jumlah_tiket': 133, 'tanggal': '2025-06-01', 'kota': 'SURABAYA'},
    {'judul_film': 'Train to Busan', 'genre': 'Komedi', 'harga_tiket': 55000, 'jumlah_tiket': 123, 'tanggal': '2025-06-01', 'kota': 'JOGJA'},
    {'judul_film': 'Black Panther', 'genre': 'Action', 'harga_tiket': 55000, 'jumlah_tiket': 81, 'tanggal': '2025-06-04', 'kota': 'SURABAYA'},
    {'judul_film': 'Captain America', 'genre': 'Komedi', 'harga_tiket': 80000, 'jumlah_tiket': 147, 'tanggal': '2025-06-09', 'kota': 'JOGJA'},
    {'judul_film': 'Avengers: Infinity War', 'genre': 'Sci-Fi', 'harga_tiket': 55000, 'jumlah_tiket': 150, 'tanggal': '2025-06-02', 'kota': 'SURABAYA'},
    {'judul_film': 'Inception', 'genre': 'Action', 'harga_tiket': 55000, 'jumlah_tiket': 130, 'tanggal': '2025-06-07', 'kota': 'SURABAYA'},
    {'judul_film': 'Joker', 'genre': 'Animasi', 'harga_tiket': 75000, 'jumlah_tiket': 80, 'tanggal': '2025-06-14', 'kota': 'MEDAN'},
    {'judul_film': 'Iron Man', 'genre': 'Animasi', 'harga_tiket': 65000, 'jumlah_tiket': 67, 'tanggal': '2025-06-01', 'kota': 'MAKASSA'},
    {'judul_film': 'Captain America', 'genre': 'Horor', 'harga_tiket': 80000, 'jumlah_tiket': 140, 'tanggal': '2025-06-15', 'kota': 'JOGJA'},
    {'judul_film': 'Frozen II', 'genre': 'Sci-Fi', 'harga_tiket': 70000, 'jumlah_tiket': 65, 'tanggal': '2025-06-12', 'kota': 'JOGJA'},
    {'judul_film': 'Spiderman: No Way Home', 'genre': 'Komedi', 'harga_tiket': 80000, 'jumlah_tiket': 118, 'tanggal': '2025-06-12', 'kota': 'SURABAYA'},
    {'judul_film': 'Captain America', 'genre': 'Drama', 'harga_tiket': 70000, 'jumlah_tiket': 131, 'tanggal': '2025-06-14', 'kota': 'SURABAYA'},
    {'judul_film': 'Conjuring The Devil Made Me Do It', 'genre': 'Action', 'harga_tiket': 70000, 'jumlah_tiket': 103, 'tanggal': '2025-06-12', 'kota': 'SURABAYA'},
    {'judul_film': 'Conjuring The Devil Made Me Do It', 'genre': 'Sci-Fi', 'harga_tiket': 75000, 'jumlah_tiket': 63, 'tanggal': '2025-06-12', 'kota': 'SURABAYA'},
    {'judul_film': 'Doctor Strange: Multiverse of Madness', 'genre': 'Action', 'harga_tiket': 80000, 'jumlah_tiket': 66, 'tanggal': '2025-06-08', 'kota': 'SURABAYA'},
    {'judul_film': 'Inception', 'genre': 'Action', 'harga_tiket': 60000, 'jumlah_tiket': 131, 'tanggal': '2025-06-08', 'kota': 'SURABAYA'},
    {'judul_film': 'Spiderman: No Way Home', 'genre': 'Sci-Fi', 'harga_tiket': 70000, 'jumlah_tiket': 66, 'tanggal': '2025-06-02', 'kota': 'SURABAYA'},
    {'judul_film': 'Parasite', 'genre': 'Komedi', 'harga_tiket': 75000, 'jumlah_tiket': 85, 'tanggal': '2025-06-12', 'kota': 'SEMARANG'},
    {'judul_film': 'Frozen II', 'genre': 'Drama', 'harga_tiket': 75000, 'jumlah_tiket': 79, 'tanggal': '2025-06-11', 'kota': 'SURABAYA'},
    {'judul_film': 'Avenger Boy', 'genre': 'Komedi', 'harga_tiket': 55000, 'jumlah_tiket': 111, 'tanggal': '2025-06-08', 'kota': 'SURABAYA'},
    {'judul_film': 'Black Panther', 'genre': 'Action', 'harga_tiket': 60000, 'jumlah_tiket': 105, 'tanggal': '2025-06-14', 'kota': 'SURABAYA'},
    {'judul_film': 'Your Name', 'genre': 'Komedi', 'harga_tiket': 65000, 'jumlah_tiket': 77, 'tanggal': '2025-06-15', 'kota': 'SEMARANG'},
    {'judul_film': 'Spiderman: Far From Home', 'genre': 'Drama', 'harga_tiket': 65000, 'jumlah_tiket': 133, 'tanggal': '2025-06-06', 'kota': 'SURABAYA'},
    {'judul_film': 'Spiderman: No Way Home', 'genre': 'Sci-Fi', 'harga_tiket': 80000, 'jumlah_tiket': 104, 'tanggal': '2025-06-13', 'kota': 'SURABAYA'},
    {'judul_film': 'Laskar Pelangi', 'genre': 'Komedi', 'harga_tiket': 60000, 'jumlah_tiket': 139, 'tanggal': '2025-06-10', 'kota': 'SURABAYA'},
    {'judul_film': 'Conjuring The Devil Made Me Do It', 'genre': 'Horor', 'harga_tiket': 80000, 'jumlah_tiket': 91, 'tanggal': '2025-06-12', 'kota': 'SURABAYA'},
    {'judul_film': 'Inception', 'genre': 'Komedi', 'harga_tiket': 75000, 'jumlah_tiket': 78, 'tanggal': '2025-06-05', 'kota': 'MEDAN'},
    {'judul_film': 'Kimi no Na wa', 'genre': 'Komedi', 'harga_tiket': 80000, 'jumlah_tiket': 66, 'tanggal': '2025-06-04', 'kota': 'SURABAYA'},
    {'judul_film': 'Frozen II', 'genre': 'Komedi', 'harga_tiket': 50000, 'jumlah_tiket': 61, 'tanggal': '2025-06-11', 'kota': 'Bandung'},
    {'judul_film': 'Avenger Boy', 'genre': 'Action', 'harga_tiket': 65000, 'jumlah_tiket': 85, 'tanggal': '2025-06-04', 'kota': 'MAKASSA'},
    {'judul_film': 'Your Name', 'genre': 'Komedi', 'harga_tiket': 65000, 'jumlah_tiket': 135, 'tanggal': '2025-06-04', 'kota': 'JKT'},
    {'judul_film': 'The Avengers Assemble', 'genre': 'Horor', 'harga_tiket': 75000, 'jumlah_tiket': 132, 'tanggal': '2025-06-07', 'kota': 'SURABAYA'},
    {'judul_film': 'Parasite', 'genre': 'Horor', 'harga_tiket': 80000, 'jumlah_tiket': 74, 'tanggal': '2025-06-08', 'kota': 'MAKASSA'},
    {'judul_film': 'The Batman', 'genre': 'Komedi', 'harga_tiket': 70000, 'jumlah_tiket': 62, 'tanggal': '2025-06-10', 'kota': 'MEDAN'},
    {'judul_film': 'Spiderman: Far From Home', 'genre': 'Animasi', 'harga_tiket': 60000, 'jumlah_tiket': 60, 'tanggal': '2025-06-08', 'kota': 'SURABAYA'},
    {'judul_film': 'The Batman', 'genre': 'Animasi', 'harga_tiket': 75000, 'jumlah_tiket': 94, 'tanggal': '2025-06-02', 'kota': 'SM'},
    {'judul_film': 'Captain America', 'genre': 'Action', 'harga_tiket': 60000, 'jumlah_tiket': 116, 'tanggal': '2025-06-12', 'kota': 'SURABAYA'},
    {'judul_film': 'Laskar Pelangi', 'genre': 'Drama', 'harga_tiket': 50000, 'jumlah_tiket': 75, 'tanggal': '2025-06-02', 'kota': 'SURABAYA'},
    {'judul_film': 'Frozen II', 'genre': 'Horor', 'harga_tiket': 80000, 'jumlah_tiket': 123, 'tanggal': '2025-06-02', 'kota': 'JOGJA'},
    {'judul_film': 'Black Panther', 'genre': 'Komedi', 'harga_tiket': 80000, 'jumlah_tiket': 93, 'tanggal': '2025-06-04', 'kota': 'SURABAYA'},
    {'judul_film': 'Conjuring The Devil Made Me Do It', 'genre': 'Action', 'harga_tiket': 50000, 'jumlah_tiket': 68, 'tanggal': '2025-06-12', 'kota': 'SURABAYA'},
    {'judul_film': 'Interstellar', 'genre': 'Komedi', 'harga_tiket': 60000, 'jumlah_tiket': 142, 'tanggal': '2025-06-15', 'kota': 'SURABAYA'},
    {'judul_film': 'Your Name', 'genre': 'Sci-Fi', 'harga_tiket': 55000, 'jumlah_tiket': 136, 'tanggal': '2025-06-01', 'kota': 'Bandung'},
    {'judul_film': 'Parasite', 'genre': 'Action', 'harga_tiket': 60000, 'jumlah_tiket': 103, 'tanggal': '2025-06-07', 'kota': 'JAKA'},
    {'judul_film': 'Spiderman: Far From Home', 'genre': 'Komedi', 'harga_tiket': 80000, 'jumlah_tiket': 90, 'tanggal': '2025-06-02', 'kota': 'SURABAYA'}
]
```

```
{'judul_film': 'Black Panther', 'genre': 'Sci-Fi', 'harga_tiket': 80000, 'jumlah_tiket': 103, 'tanggal': '2025-06-11', 'kota': 'bandung'},
{'judul_film': 'Laskar Pelangi', 'genre': 'Komedi', 'harga_tiket': 65000, 'jumlah_tiket': 71, 'tanggal': '2025-06-07', 'kota': 'bandung'},
{'judul_film': 'Conjuring The Devil Made Me Do It', 'genre': 'Drama', 'harga_tiket': 60000, 'jumlah_tiket': 108, 'tanggal': '2025-06-08', 'kota': 'bandung'},
{'judul_film': 'Conjuring The Devil Made Me Do It', 'genre': 'Horor', 'harga_tiket': 60000, 'jumlah_tiket': 136, 'tanggal': '2025-06-09', 'kota': 'bandung'},
{'judul_film': 'The Avengers Assemble', 'genre': 'Drama', 'harga_tiket': 75000, 'jumlah_tiket': 80, 'tanggal': '2025-06-13', 'kota': 'bandung'},
{'judul_film': 'Spiderman: Homecoming', 'genre': 'Action', 'harga_tiket': 65000, 'jumlah_tiket': 100, 'tanggal': '2025-06-15', 'kota': 'bandung'},
{'judul_film': 'Toy Story 4', 'genre': 'Action', 'harga_tiket': 60000, 'jumlah_tiket': 73, 'tanggal': '2025-06-06', 'kota': 'bandung'},
{'judul_film': 'Frozen II', 'genre': 'Sci-Fi', 'harga_tiket': 55000, 'jumlah_tiket': 76, 'tanggal': '2025-06-01', 'kota': 'bandung'},
{'judul_film': 'Frozen II', 'genre': 'Komedi', 'harga_tiket': 60000, 'jumlah_tiket': 82, 'tanggal': '2025-06-02', 'kota': 'bandung'},
{'judul_film': 'Doctor Strange: Multiverse of Madness', 'genre': 'Drama', 'harga_tiket': 55000, 'jumlah_tiket': 107, 'tanggal': '2025-06-03', 'kota': 'bandung'}
```

Soal 1 (10 poin)

Hitung **total pendapatan tiket** untuk setiap **genre**. (Pendapatan = harga_tiket × jumlah_tiket)

```
pendapatan_per_genre = {}

"""
    Insert your code here.
"""

print("Total Pendapatan per Genre:")
for genre, total in pendapatan_per_genre.items():
    print(f"- {genre}: {total}")
```

Total Pendapatan per Genre:

Soal 2 (15 poin)

Tentukan **rata-rata harga tiket** untuk setiap **genre**.

```
rata_rata_harga_per_genre = {}

"""
    Insert your code here.
"""

print("Rata-rata Harga Tiket per Genre:")
for genre, rata_rata in rata_rata_harga_per_genre.items():
    print(f"- {genre}: {rata_rata:.2f}")
```

Rata-rata Harga Tiket per Genre:

Soal 3 (20 poin)

Cari **tanggal** dengan jumlah tiket terbanyak (akumulasi semua film).

```
tanggal_terbanyak = None

"""
    Insert your code here.
"""

print("\nTanggal dengan jumlah tiket terbanyak:", tanggal_terbanyak)
```

Tanggal dengan jumlah tiket terbanyak: None

Soal 4 (30 poin)

Tentukan kota yang menayangkan film berbeda terbanyak (jumlah judul unik).

```
kota_variasi_tertinggi = None

"""
    Insert your code here.
"""

print("\nKota dengan variasi film paling tinggi:", kota_variasi_tertinggi)
```

Kota dengan variasi film paling tinggi: None

✓ Soal 5 (25 poin)

Hitung total pendapatan = harga_tiket * jumlah_tiket untuk setiap film. Tentukan film dengan pendapatan tertinggi.

```
film_teruntung = None
"""
    Insert your code here.
"""
print("\nFilm paling menguntungkan:", film_teruntung)
```

Film paling menguntungkan: None



Soal Bonus: Temukan Insight Menarik dari Dataset (10 poin)

Lakukan eksplorasi bebas terhadap dataset ini, dan temukan **satu insight menarik** yang tidak ditanyakan pada soal-soal sebelumnya.

Contoh jenis insight:

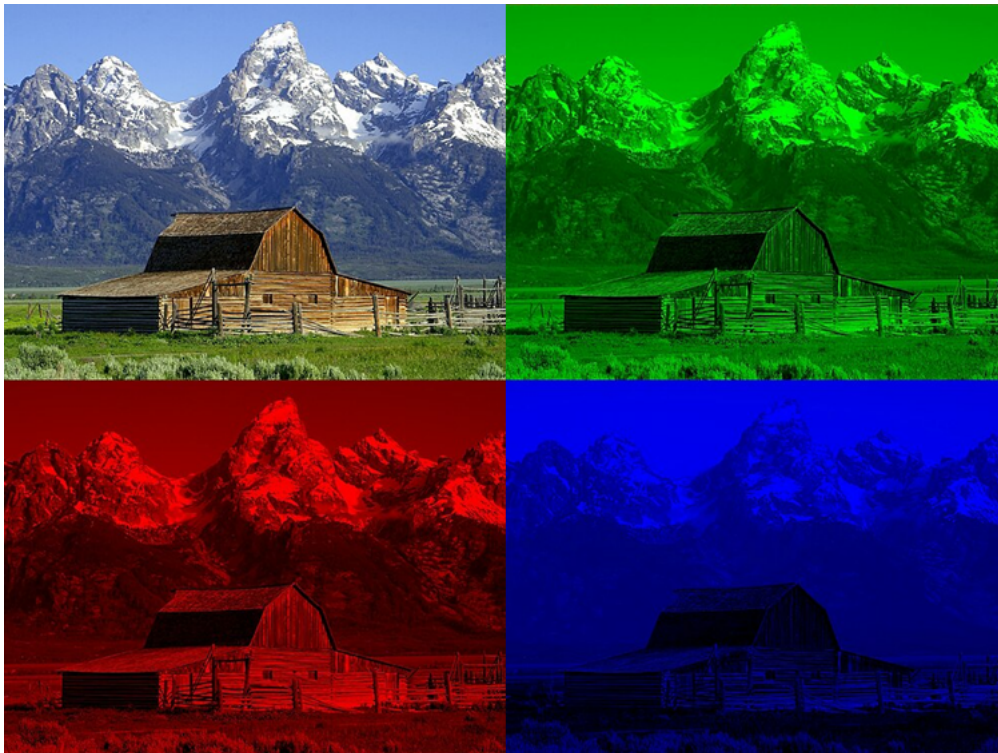
- Studio dengan rata-rata pendapatan tertinggi?

👉 🤖 Computer Vision Task: Face or Flag?

Bayangkan kamu seorang **kurator emoji di sebuah pameran besar**. Setiap ikon terlihat mirip, kecil, warna-warni, dan seragam ukurannya, tetapi sebenarnya mereka berasal dari **dua kelas** yang berbeda: **Face (wajah)** dan **Flag (bendera)**. Tugasmu sederhana, yaitu membuat pemilah emoji otomatis untuk tiap-tiap emoji yang diberikan, memutuskan apakah sebuah emoji termasuk kelas *Face* atau *Flag* seakurat mungkin.

Piksel dan Gambar Digital

Gambar digital disusun dari kotak-kotak kecil yang disebut **piksel (pixel)**. Setiap piksel menyimpan informasi warna. Untuk gambar digital, warna biasanya disimpan di dalam sebuah **model RGB**. RGB adalah singkatan dari *Red* (merah), *Green* (hijau), *Blue* (biru), sehingga **setiap piksel** direpresentasikan oleh tiga angka $[R, G, B]$ untuk tiap-tiap *channel* (kanal). Ketiga angka itu menunjukkan seberapa terang warna merah, hijau, dan biru “menyala” di masing-masing kanal RGB secara bersamaan di piksel tersebut. Nilai pada masing-masing kanal RGB umumnya bernilai diantara $0-255$, dimana nilai 0 berarti kanal tersebut **mati (gelap)** sedangkan nilai 255 berarti kanal tersebut **menyala dengan intensitas maksimal**.

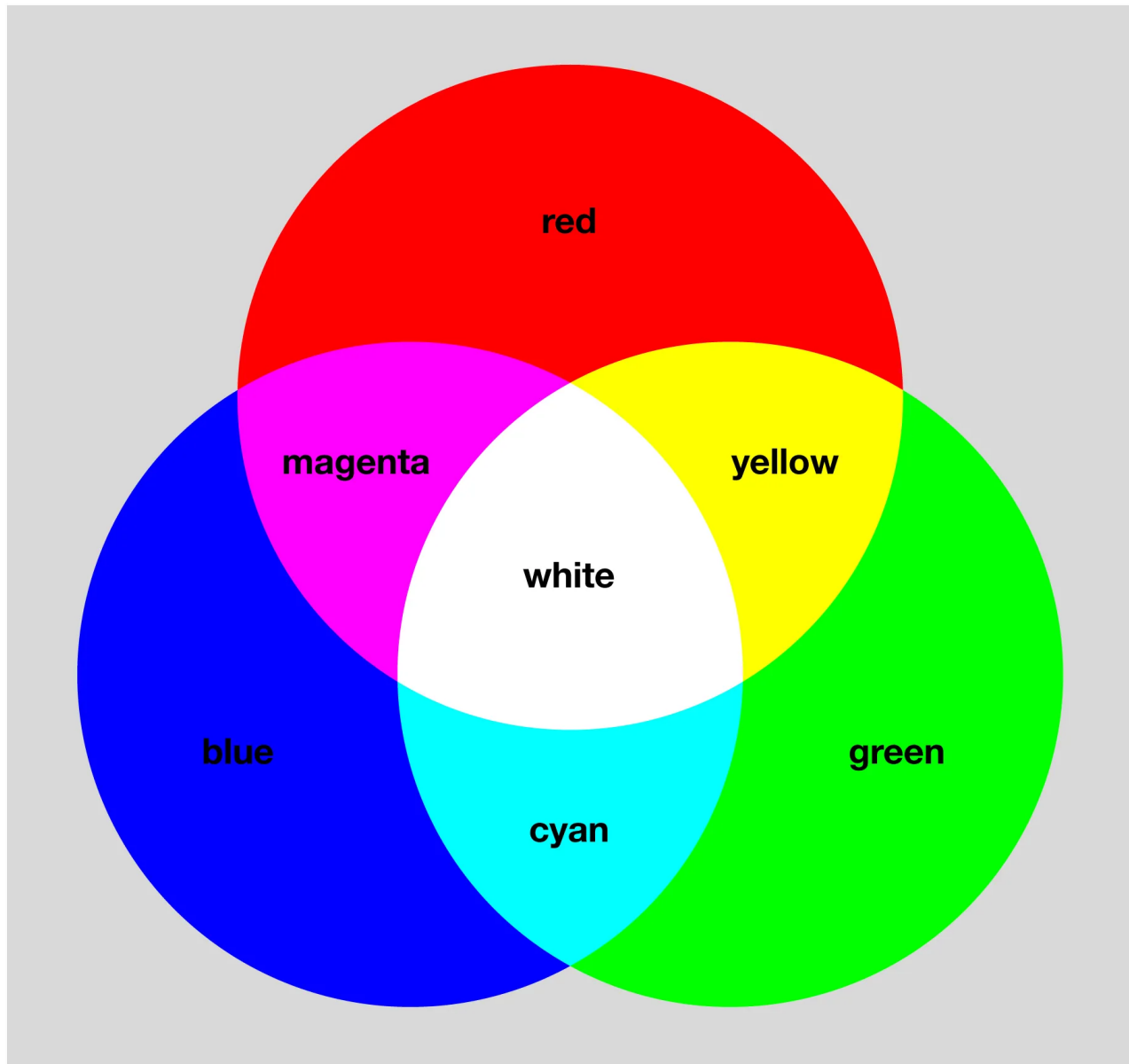


Contoh:

1. Hitam = $[0, 0, 0]$ (semua mati)
2. Putih = $[255, 255, 255]$ (semua menyala penuh)
3. Merah murni = $[255, 0, 0]$
4. Hijau murni = $[0, 255, 0]$
5. Biru murni = $[0, 0, 255]$
6. Kuning = $[255, 255, 0]$
7. Biru muda (*cyan*) = $[0, 255, 255]$
8. Ungu muda (*magenta*) = $[255, 0, 255]$
9. Oranye = $[255, 165, 0]$



RGB color model

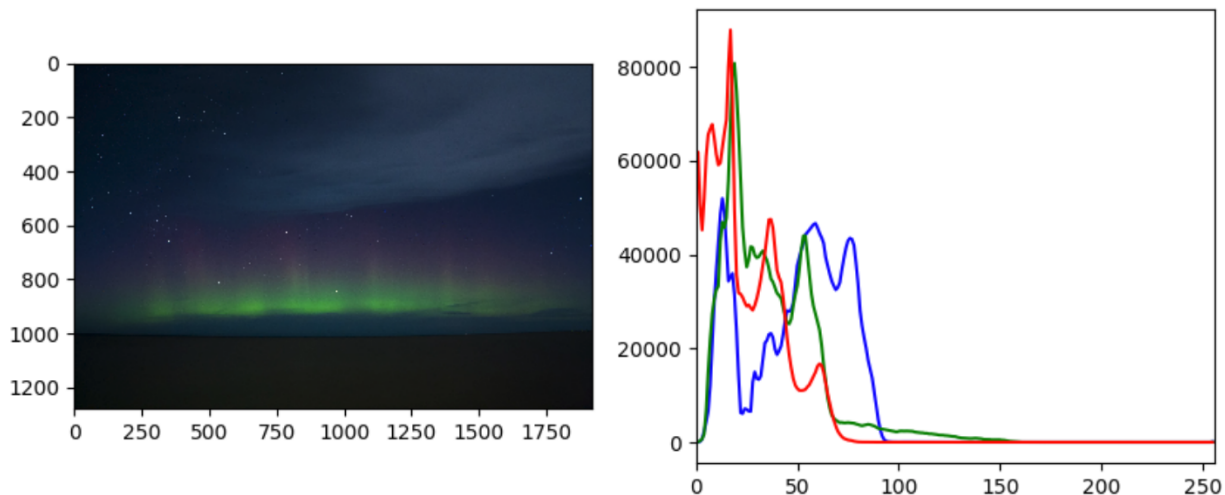


© Encyclopædia Britannica, Inc.

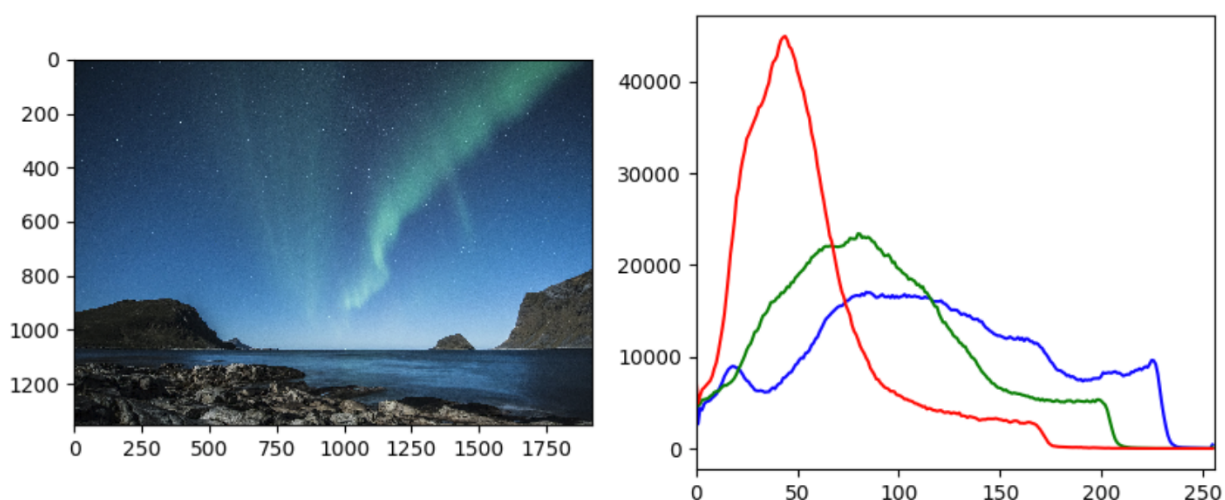
Histogram dari gambar digital RGB

Histogram kanal RGB adalah ringkasan statistik yang menunjukkan seberapa sering setiap nilai intensitas warna R, G, dan B (umumnya pada rentang 0–255) muncul di seluruh piksel sebuah gambar. Sumbu-X mewakili nilai intensitas (dari gelap dengan nilai 0 sampai ke terang dengan nilai 255), sedangkan sumbu-Y menunjukkan jumlah kemunculan (atau probabilitas jika dinormalisasi, sehingga total area = 1). Dengan menggambarkan ketiga kanal untuk warna merah, hijau, dan biru secara bersama-sama, kita bisa melihat warna mana yang dominan (memiliki puncak tinggi), seberapa “kontras” gambar (sebaran intensitas yang sempit vs lebar), adanya pencahayaan berlebih/kurang (penumpukan di ujung kanan/kiri histogram), serta pola campuran warna (puncak ganda/multimodal). Perbandingan bentuk histogram antarkanal membantu memahami karakter warna gambar tanpa melihatnya secara visual, berguna untuk deskripsi data, pemilihan nilai ambang sederhana, dan deteksi perbedaan objek pada gambar.

sample image with dark tones and histogram (colored),



Sample image with mid tones and histogram (colored)



Dataset yang tersedia

- **Train:** Gambar emoji terdapat di dalam variabel `X_train` yang terdiri dari 160 gambar emoji dengan ukuran 64x64 dengan 3 *channel* RGB. Kelas untuk masing-masing gambar bisa ditemukan di variabel `y_train` dimana nilai `1` = Faces dan `2` = Flags.
- **Test:** Variabel `X_test` terdiri dari 310 gambar emoji dengan ukuran dan jumlah *channel* yang sama dengan `X_train`. Akan tetapi, gambar-gambar emoji di `X_test` tidak memiliki informasi kelas untuk masing-masing emoji. **Dataset Test akan diberikan sebelum 1 jam kontes berakhir.**

Aturan

- Anda bebas memilih dan menggunakan pra-proses dan pendekatan pemodelan yang terbaik.
- Anda tidak diperkenankan untuk menggunakan data eksternal yang mengandung emoji serupa.
- Penilaian akhir akan menggunakan akurasi pada set uji `X_test`.

> Download data (TIDAK PERLU DIUBAH)

↳ 1 cell hidden

> Fungsi util untuk evaluasi (TIDAK PERLU DIUBAH)

↳ 1 cell hidden

Memperlihatkan contoh gambar emoji dan kelasnya dalam grid (TIDAK PERLU DIUBAH)

```
print("Ukuran (shapes) dari dataset:")
print(" X_train:", X_train.shape, " y_train:", y_train.shape)
print(" X_test :", X_test.shape)

# Label yang tersedia di train (harus 1..5)
print("Label unik yang tersedia di dataset:", sorted(set(y_train.tolist())))

# Tampilkan beberapa contoh gambar + label (matplotlib)
import numpy as np
import matplotlib.pyplot as plt

def show_grid(images, labels=None, n=12, cols=6):
    n = min(n, len(images))
    rows = (n + cols - 1) // cols
    plt.figure(figsize=(1.8*cols, 1.8*rows))
    for i in range(n):
        ax = plt.subplot(rows, cols, i+1)
        ax.imshow(images[i])
        ax.axis("off")
        if labels is not None:
            ax.set_title(str(labels[i]))
    plt.tight_layout()
    plt.show()

# Acak indeks hanya untuk visualisasi
rng = np.random.RandomState(0)
idx = rng.choice(len(X_train), size=12, replace=False)
show_grid(X_train[idx], y_train[idx])
```

Ukuran (shapes) dari dataset:
 X_train: (160, 64, 64, 3) y_train: (160,)
 X_test : (310, 64, 64, 3)
 Label unik yang tersedia di dataset: [1, 2]



CONTOH Mengakses 1 buah gambar emoji dan memperlihatkan masing-masing kanal RGB

```
# Contoh: melihat 1 gambar TRAIN + memecah kanal R, G, B
from emoji_train import X_train, y_train
import matplotlib.pyplot as plt

# --- Pilih indeks gambar yang ingin dilihat ---
INDEX = 0 # ganti sesuai kebutuhan, 0 <= INDEX < len(X_train)

# --- Ambil satu gambar & labelnya ---
# X_train[INDEX] adalah array (H, W, 3) bertipe uint8; kita ubah ke list bersarang agar mudah dipahami
img = X_train[INDEX].tolist() # ukuran: H x W x 3; setiap piksel = [R, G, B] (0..255)
label = int(y_train[INDEX]) # label integer (1 atau 2)

# --- Informasi dimensi gambar ---
H = len(img) # jumlah baris (tinggi)
W = len(img[0]) if H > 0 else 0 # jumlah kolom (lebar)
C = len(img[0][0]) if (H > 0 and W > 0) else 0 # jumlah kanal (harusnya 3 = R,G,B)
```

```

print(f"Index: {INDEX} | Label: {label} | Ukuran: {H}x{W} | Kanal: {C}")

# --- Akses contoh piksel (baris=10, kolom=20; sesuaikan agar masih dalam rentang) ---
y, x = min(10, H-1), min(20, W-1)
Rpx, Gpx, Bpx = img[y][x] # ini adalah nilai RGB 0..255 pada piksel tersebut
print(f"Contoh piksel di (baris={y}, kolom={x}): R={Rpx}, G={Gpx}, B={Bpx}")

# --- Pecah kanal menjadi 2D (grayscale) ---
# Setiap kanal akan berupa list 2D berukuran H x W berisi angka 0..255
R_plane = [[img[r][c][0] for c in range(W)] for r in range(H)]
G_plane = [[img[r][c][1] for c in range(W)] for r in range(H)]
B_plane = [[img[r][c][2] for c in range(W)] for r in range(H)]

# --- Buat visualisasi berwarna "single-channel" ---
# R-only: salin nilai R ke kanal merah, nolkan G dan B
R_only_color = [[[img[r][c][0], 0, 0] for c in range(W)] for r in range(H)]
# G-only: salin nilai G ke kanal hijau, nolkan R dan B
G_only_color = [[[0, img[r][c][1], 0] for c in range(W)] for r in range(H)]
# B-only: salin nilai B ke kanal biru, nolkan R dan G
B_only_color = [[[0, 0, img[r][c][2]] for c in range(W)] for r in range(H)]

# Jika Anda juga ingin melihat B-only, tampilkan terpisah:
# Gambar asli
plt.figure(figsize=(4, 4))
plt.imshow(img)
plt.axis('off')
plt.title("Gambar Asli (RGB)")
plt.show()

# --- Tampilkan semuanya dalam grid subplot ---
fig, axes = plt.subplots(2, 3, figsize=(12, 8))

# (a) Visualisasi R-only berwarna
axes[0, 0].imshow(R_only_color)
axes[0, 0].set_title("R-only (RGB)")
axes[0, 0].axis("off")

# (b) Visualisasi G-only & B-only
axes[0, 1].imshow(G_only_color)
axes[0, 1].set_title("G-only (RGB)")
axes[0, 1].axis("off")

# (c) Visualisasi B-only
axes[0, 2].imshow(B_only_color)
axes[0, 2].set_title("B-only (RGB)")
axes[0, 2].axis("off")

# (d) Kanal R sebagai grayscale
axes[1, 0].imshow(R_plane, cmap="gray", vmin=0, vmax=255)
axes[1, 0].set_title("Kanal R (grayscale)")
axes[1, 0].axis("off")

# (e) Kanal G sebagai grayscale
axes[1, 1].imshow(G_plane, cmap="gray", vmin=0, vmax=255)
axes[1, 1].set_title("Kanal G (grayscale)")
axes[1, 1].axis("off")

# (f) Kanal B sebagai grayscale
axes[1, 2].imshow(B_plane, cmap="gray", vmin=0, vmax=255)
axes[1, 2].set_title("Kanal B (grayscale)")
axes[1, 2].axis("off")

plt.tight_layout()
plt.show()

```

Index: 0 | Label: 1 | Ukuran: 64x64 | Kanal: 3
 Contoh piksel di (baris=10, kolom=20): R=255, G=252, B=190

Gambar Asli (RGB)



R-only (RGB)



G-only (RGB)



B-only (RGB)



Kanal R (grayscale)



Kanal G (grayscale)



Kanal B (grayscale)



✓ CONTOH Mengakses 1 buah emoji dan menampilkan histogram intensitas untuk tiap-tiap channel (kanal)

```
# Tampilkan 1 emoji + histogram kanal R/G/B dalam satu figur
# - Data gambar diperlakukan sebagai list bersarang: img[baris][kolom] -> [R, G, B] (0..255)
# - Histogram dihitung manual (tanpa NumPy)

from emoji_train import X_train, y_train
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec
```

```

# --- Pilih indeks gambar yang ingin dilihat ---
INDEX = 0 # ubah sesuai kebutuhan (0 <= INDEX < len(X_train))

# Ambil gambar & label; ubah ke list bersarang (HxWx3)
img = X_train[INDEX].tolist()
label = int(y_train[INDEX])

# Info dimensi
H = len(img)
W = len(img[0]) if H > 0 else 0
C = len(img[0][0]) if (H > 0 and W > 0) else 0
print(f"Index: {INDEX} | Label: {label} | Ukuran: {H}x{W} | Kanal: {C}")

# --- Hitung histogram kanal R/G/B (0..255) dengan loop murni ---
def channel_histograms(img_3d):
    """Kembalikan (histR, histG, histB) – masing2 list panjang 256."""
    H = len(img_3d)
    W = len(img_3d[0]) if H > 0 else 0
    histR = [0]*256
    histG = [0]*256
    histB = [0]*256
    for r in range(H):
        row = img_3d[r]
        for c in range(W):
            R, G, B = row[c] # [R,G,B] 0..255
            histR[int(R)] += 1
            histG[int(G)] += 1
            histB[int(B)] += 1
    return histR, histG, histB

histR, histG, histB = channel_histograms(img)

# (Opsional) Normalisasi ke probabilitas agar skala ketiganya sebanding
total_pixels = H * W if (H > 0 and W > 0) else 1
probR = [h/total_pixels for h in histR]
probG = [h/total_pixels for h in histG]
probB = [h/total_pixels for h in histB]
bins = list(range(256)) # 0..255

# --- Tata letak figur: atas gambar RGB, bawah satu axes berisi 3 kurva histogram ---
fig = plt.figure(figsize=(12, 8))
gs = GridSpec(nrows=2, ncols=1, height_ratios=[2.2, 1.5], hspace=0.28)

# Atas: gambar asli
ax_img = fig.add_subplot(gs[0, 0])
ax_img.imshow(img)
ax_img.set_title("Gambar Asli (RGB)")
ax_img.axis("off")

# Bawah: histogram kanal R/G/B (dioverlay)
ax_hist = fig.add_subplot(gs[1, 0])
ax_hist.plot(bins, probR, label="Kanal R", color="red")
ax_hist.plot(bins, probG, label="Kanal G", color="green")
ax_hist.plot(bins, probB, label="Kanal B", color="blue")
ax_hist.set_xlim(0, 255)
ax_hist.set_xlabel("Nilai Intensitas (0..255)")
ax_hist.set_ylabel("Probabilitas (p)")
ax_hist.set_title("Histogram Kanal RGB (dinormalisasi)")
ax_hist.grid(alpha=0.25)
ax_hist.legend()

plt.tight_layout()
plt.show()

# (Opsional) Jika ingin versi 'bar' (batang) bukan garis – hati-hati bisa padat:
plt.figure(figsize=(12,4))
plt.bar(bins, probR, color="red", alpha=0.4, label="R")
plt.bar(bins, probG, color="green", alpha=0.4, label="G")
plt.bar(bins, probB, color="blue", alpha=0.4, label="B")
plt.xlim(0, 255); plt.xlabel("Nilai Intensitas"); plt.ylabel("Probabilitas"); plt.legend(); plt.show()

```

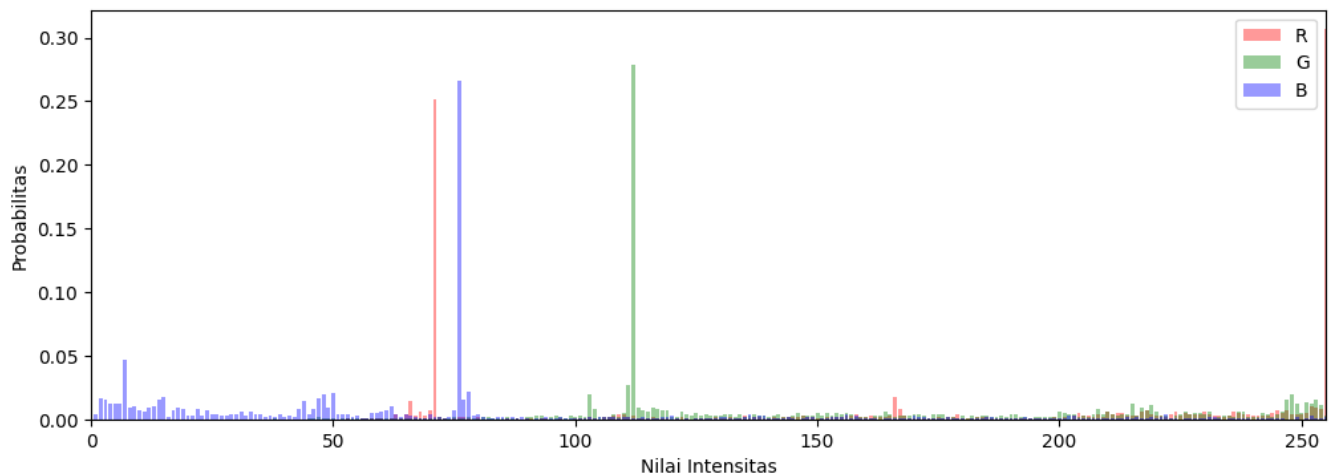
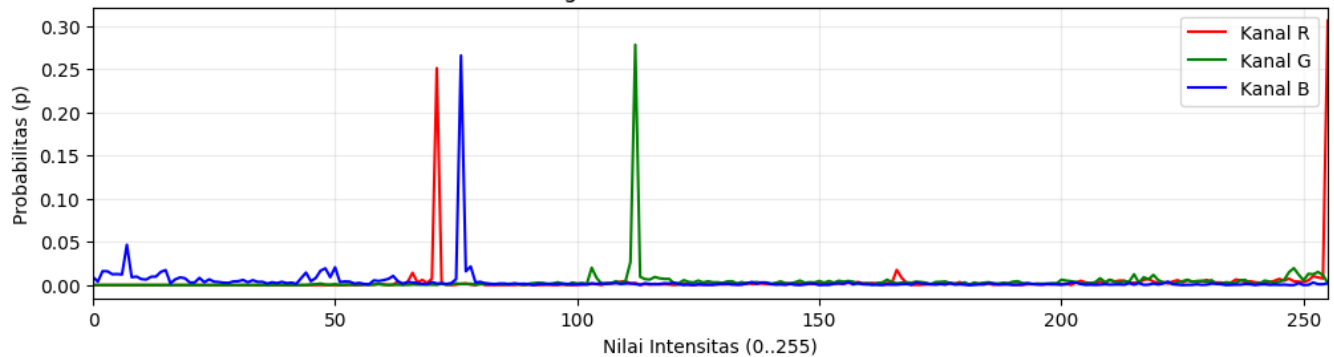
Index: 0 | Label: 1 | Ukuran: 64x64 | Kanal: 3

/tmp/ipython-input-785755323.py:70: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be different.
plt.tight_layout()

Gambar Asli (RGB)



Histogram Kanal RGB (dinormalisasi)



TUGAS ANDA

Lengkapilah fungsi ini untuk memprediksi apakah sebuah gambar emoji termasuk ke dalam kelas *Faces* (kelas = 1) atau *Flags* (kelas = 2).

```
def prediksi_gambar_img(img):
    """
    Template fungsi prediksi untuk 1 gambar.
    Parameter
    -----
    img : list[list[list[int]]] atau array setara
        Gambar berukuran HxWx3 dengan nilai 0..255 (RGB).
        Jika yang masuk berupa NumPy array, fungsi akan mencoba mengubahnya ke list.

    Return
```

```

-----
int
    Label prediksi {1, 2}, dimana 1 = Faces, 2 = Flags.
"""
# Pastikan berbentuk list bersarang (jika masih array, ubah ke list)
if hasattr(img, "tolist"):
    img = img.tolist()

# Validasi bentuk data dasar
H = len(img)
W = len(img[0]) if H > 0 else 0
C = len(img[0][0]) if (H > 0 and W > 0) else 0
if H == 0 or W == 0 or C != 3:
    raise ValueError(f"Bentuk gambar tidak valid. Ditemukan: H={H}, W={W}, C={C} (harus HxWx3)")

# ----- KODE ANDA MULAI DI SINI -----
# Tulis logika Anda di bawah untuk menghitung fitur/keputusan dari img (list bersarang).
# Contoh pola penggunaan:
#
# some_value = 0.0
# for r in range(H):
#     for c in range(W):
#         R, G, B = img[r][c] # akses piksel di baris=r, kolom=c
#         # ... tulis logika Anda di sini ...
#
# pred_label = 1 (untuk Faces) # atau 2 (untuk Flags), sesuai keputusan Anda
# ----- KODE ANDA SELESAI DI SINI -----

# Validasi & kembalikan label
try:
    pred_label
except NameError:
    raise RuntimeError("Anda belum menetapkan variabel 'pred_label' di blok KODE ANDA.")

if pred_label not in (1, 2):
    raise ValueError("pred_label harus bernilai 1 atau 2.")

return int(pred_label)

```

Melakukan Prediksi Pada Data Latih

Jawaban Anda akan dinilai sebagian terhadap data latih yang terdiri dari 160 gambar emoji. Data latih ini bisa digunakan untuk menentukan ekstraksi fitur pada gambar emoji dan membuat aturan untuk memprediksi apakah sebuah gambar termasuk emoji *Faces* atau *Flags*. Ekstraksi fitur dan aturan prediksi untuk sebuah gambar bisa diimplementasikan di fungsi `prediksi_gambar_img` (cell sebelum ini).

Catatan:

1. **Anda tidak perlu mengubah bagian kode di bawah ini, selama Anda mengimplementasikan fungsi `prediksi_gambar_img` di atas.** Kode di bawah ini akan otomatis berjalan dengan baik selama Anda mengimplementasikan fungsi `prediksi_gambar_img`.

```

# Template TRAIN (tanpa NumPy di kode Anda):
# - Mengiterasi contoh latih satu per satu
# - Mengonversi gambar menjadi list bersarang: img[baris][kolom] -> [R, G, B] (nilai 0..255)
# - Menampilkan informasi ukuran/kanal + visualisasi
# - Menyediakan tempat kosong untuk menulis fitur/keputusan Anda sendiri
#
# Catatan:
#   X_train: array gambar berukuran (N, H, W, 3) bertipe uint8
#   y_train: label berukuran (N,) bernilai {1, 2}

from emoji_train import X_train, y_train
import matplotlib.pyplot as plt

THRESH_N = 160 # batasi berapa banyak sampel yang diproses (agar cepat saat eksplorasi)

# (Opsional) list untuk menyimpan prediksi Anda (jika ingin mengevaluasi/submit nanti)
preds = [] # isi dengan 1 atau 2 untuk tiap sampel (panjangnya bebas di tahap eksplorasi)

N = min(THRESH_N, len(X_train))
for i in range(N):
    # Ambil satu gambar dan ubah ke list bersarang (tanpa bergantung pada NumPy di logika Anda)
    img = X_train[i].tolist() # ukuran: H x W x 3, masing-masing piksel = [R, G, B] (0..255)

```

```

# Ambil label (1 atau 2) untuk data TRAIN
label = int(y_train[i])

# Informasi struktur gambar
H = len(img)                # jumlah baris (tinggi)
W = len(img[0]) if H > 0 else 0 # jumlah kolom (lebar)
C = len(img[0][0]) if (H > 0 and W > 0) else 0 # jumlah kanal (harusnya 3 = RGB)

# Memanggil fungsi `prediksi_gambar_img` yang Anda buat untuk memprediksi kelas gambar img.
prediksi_label = prediksi_gambar_img(img)
preds.append(prediksi_label)

print(f"[TRAIN] idx={i} | size={H}x{W} | channels={C} | label={label} | prediksi={prediksi_label}")

# (Opsional) Visualisasi untuk pengecekan cepat
plt.imshow(img)
plt.axis('off')
plt.title(f"train idx={i} | label={label} | prediksi={prediksi_label}")
plt.show()

# Anda bisa menggunakan variabel 'preds' untuk mengevaluasi TRAIN,
# dengan membandingkan 'preds' vs 'y_train'.

# -----
# CEK & EVALUASI
# -----
if len(preds) != N:
    raise RuntimeError(
        f"Prediksi belum lengkap: len(preds)={len(preds)} sementara N={N}. "
        "Silakan isi blok 'KODE ANDA MULAI DI SINI' untuk menghasilkan prediksi."
    )

acc, f1_macro = evaluate_binary_12(y_train[:N], preds)

print(f"Evaluasi pada {N} sampel test:")
print(f"Akurasi      : {acc:.4f}")
print(f"F1 Macro     : {f1_macro:.4f}")

```

```

-----
NameError                                Traceback (most recent call last)
/tmp/ipython-input-351055284.py in prediksi_gambar_img(img)
    40     try:
--> 41         pred_label
    42     except NameError:

NameError: name 'pred_label' is not defined

During handling of the above exception, another exception occurred:

RuntimeError                                Traceback (most recent call last)
-----
1 frames
/tmp/ipython-input-351055284.py in prediksi_gambar_img(img)
    41         pred_label
    42     except NameError:
--> 43         raise RuntimeError("Anda belum menetapkan variabel 'pred_label' di blok KODE ANDA.")
    44
    45     if pred_label not in (1, 2):

RuntimeError: Anda belum menetapkan variabel 'pred_label' di blok KODE ANDA.

```

PENILAIAN: Melakukan Prediksi Pada Data Uji

Jawaban Anda akan dinilai sebagian terhadap data uji yang terdiri dari 310 gambar emoji. Ekstraksi fitur dan aturan prediksi untuk sebuah gambar bisa diimplementasikan di fungsi `prediksi_gambar_img`, yang juga digunakan untuk memprediksi data latih.

Catatan:

1. File pengujian hanya akan dibuka 1 jam sebelum kontes selesai!
2. Anda tidak perlu mengubah bagian kode di bawah ini, selama Anda mengimplementasikan fungsi `prediksi_gambar_img` di atas. Kode di bawah ini akan otomatis berjalan dengan baik selama Anda mengimplementasikan fungsi `prediksi_gambar_img`.

```

# Template TRAIN (tanpa NumPy di kode Anda):
# - Mengiterasi contoh latih satu per satu

```

```

# - Mengonversi gambar menjadi list bersarang: img[baris][kolom] -> [R, G, B] (nilai 0..255)
# - Menampilkan informasi ukuran/kanal + visualisasi
# - Menyediakan tempat kosong untuk menulis fitur/keputusan Anda sendiri

FILE_ID = "1sgh5ofZ1Eg9J2TchSzdf1W0jB9PIeo04" # WITHOUT THE KEYS
!gdown --id {FILE_ID} -O OSNAI2025_privateGT.zip
!unzip -o OSNAI2025_privateGT.zip -d /content/

# Import path
import sys
sys.path.append("/content/OSNAI2025_pack")

from emoji_test_gt import X_test, y_test # private ground-truth module

import matplotlib.pyplot as plt

THRESH_N = 310 # batasi berapa banyak sampel yang diproses (agar cepat saat eksplorasi)

# (Optional) list untuk menyimpan prediksi Anda (jika ingin mengevaluasi/submit nanti)
preds = [] # isi dengan 1 atau 2 untuk tiap sampel (panjangnya bebas di tahap eksplorasi)

N = min(THRESH_N, len(X_test))
for i in range(N):
    # Ambil satu gambar dan ubah ke list bersarang (tanpa bergantung pada NumPy di logika Anda)
    img = X_test[i].tolist() # ukuran: H x W x 3, masing-masing piksel = [R, G, B] (0..255)

    # Ambil label (1 atau 2) untuk data TRAIN
    label = int(y_test[i])

    # Informasi struktur gambar
    H = len(img) # jumlah baris (tinggi)
    W = len(img[0]) if H > 0 else 0 # jumlah kolom (lebar)
    C = len(img[0][0]) if (H > 0 and W > 0) else 0 # jumlah kanal (harusnya 3 = RGB)

    # Memanggil fungsi `prediksi_gambar_img` yang Anda buat untuk memprediksi kelas gambar img.
    prediksi_label = prediksi_gambar_img(img)
    preds.append(prediksi_label)

    # (Optional) Visualisasi untuk pengecekan cepat
    print(f"[TEST] size={H}x{W} | channels={C} | label={label} | prediksi={prediksi_label}")
    plt.imshow(img)
    plt.axis('off')
    plt.title(f"label={label} | prediksi={prediksi_label}")
    plt.show()

# Anda bisa menggunakan variabel 'preds' untuk mengevaluasi TRAIN,
# dengan membandingkan 'preds' vs 'y_test'.

# -----
# CEK & EVALUASI
# -----
if len(preds) != N:
    raise RuntimeError(
        f"Prediksi belum lengkap: len(preds)={len(preds)} sementara N={N}. "
        "Silakan isi blok 'KODE ANDA MULAI DI SINI' untuk menghasilkan prediksi."
    )

acc, f1_macro = evaluate_binary_12(y_test[:N], preds)

print(f"Evaluasi pada {N} sampel test:")
print(f"Akurasi : {acc:.4f}")
print(f"F1 Macro : {f1_macro:.4f}")

```

```
/usr/local/lib/python3.12/dist-packages/gdown/__main__.py:140: FutureWarning: Option `--id` was deprecated in version 4.3.1 and will be removed in version 5.0.0. Use `--url` instead.
warnings.warn(
Failed to retrieve file url:
```

Cannot retrieve the public link of the file. You may need to change the permission to 'Anyone with the link', or have had many accesses. Check FAQ in <https://github.com/wkentaro/gdown?tab=readme-ov-file#faq>.

You may still be able to access the file from the browser:

<https://drive.google.com/uc?id=1sgh5ofZ1Eg9J2TchSzdf1W0jB9PIeo04>

but Gdown can't. Please check connections and permissions.

```
Archive: OSNAI2025_privateGT.zip
  inflating: /content/label_map.json
  inflating: /content/emoji_test_gt.py
  inflating: /content/test_private_gt.npz
  inflating: /content/build_stats_private.json
```

```
-----
NameError                                Traceback (most recent call last)
/tmp/ipython-input-351055284.py in prediksi_gambar_img(img)
     40     try:
--> 41         pred_label
     42     except NameError:
```

Acak Dokumen

Aji ditugaskan untuk mengirimkan dokumen-dokumen berisi informasi penting yang akan menjadi bagian dari upaya mendokumentasikan pengetahuan dan pengalaman manusia, jika suatu saat peradaban ini lenyap. Dokumen ini mencakup tulisan berbahasa Indonesia dari berbagai topik, dan **dipastikan** terdiri atas 2 paragraf. Dengan alasan keamanan, paragraf pertama dan kedua disimpan dalam list yang berbeda, yaitu list `first` dan `second`.

Namun, sistem komputer Aji diusili oleh Ammar, dan list untuk paragraf kedua kini diacak-acak! Dengan kata lain, pasangan `first[i]` tidak lagi dipastikan `second[i]`. Kini Aji pun meminta bantuan Anda untuk mengembalikan pasangan awal dokumen-dokumen tersebut!

✓ Membaca Data

Potongan kode di bawah ini adalah untuk membantu mendownload file. Tidak perlu diganti dan cukup jalankan saja!

```
from datasets import load_dataset

def read_wikipair_texts(base):
    ds = load_dataset(base)

    first = [str(x) for x in list(ds["first_part"]["text"]) if x is not None]
    second = [str(x) for x in list(ds["second_part"]["text"]) if x is not None]

    return first, second
```

Berikut adalah proses untuk mendownload datanya. Dalam kasus ini, kita menggunakan development set, atau data **dev** yang hanya berisi 25 dokumen.

Ini **BUKAN** merupakan dokumen yang akan dipakai untuk penilaian.

```
first, second = read_wikipair_texts("afaji/wikipair-dev")
```

Mari kita lihat contohnya.

Bisa dilihat bahwa data `first` dan `second` tidak cocok pasangannya! Tugas Anda adalah untuk mencocokkannya kembali.

```
import textwrap

print("First 1\n", textwrap.fill(first[0], width=80))
print("First 2\n", textwrap.fill(first[1], width=80))
print("---")
print("Second 1\n", textwrap.fill(second[0], width=80))
print("Second 2\n", textwrap.fill(second[1], width=80))
```

First 1

Teknik pembuatan batik tradisional melibatkan penggunaan canting untuk menggambar motif halus dan cap (temple) untuk mencetak motif berulang secara lebih cepat. Motif-motif batik sarat makna, mulai dari motif keraton yang melambangkan status dan aturan adat hingga motif rakyat yang mencerminkan kehidupan sehari-hari dan alam. Batik juga memiliki fungsi ritual dan simbolis, banyak dipakai dalam upacara adat, pernikahan, dan kegiatan resmi. Perawatan, pewarnaan alami, dan keterampilan pengerjaan menjadi bagian penting dalam menjaga kualitas batik tradisional.

First 2

Olimpiade Sains Nasional (OSN) adalah kompetisi tahunan yang bertujuan mengembangkan bakat dan kemampuan siswa dalam berbagai bidang sains. Kegiatan ini diselenggarakan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi bekerja sama dengan dinas pendidikan provinsi dan kabupaten/kota. OSN terbuka bagi siswa jenjang SD, SMP, dan SMA yang tertarik menguji kemampuan akademik di bidang seperti matematika, fisika, kimia, biologi, komputer, dan ilmu bumi. Kompetisi ini juga berfungsi sebagai sarana seleksi untuk pembinaan lebih lanjut dan representasi daerah di tingkat nasional maupun internasional.

Second 1

Kesempatan muncul ketika Jepang menyerah kepada Sekutu pada Agustus 1945, menciptakan kekosongan kekuasaan di Indonesia. Pemuda nasionalis menekan tokoh seperti Soekarno dan Hatta untuk segera memproklamasikan kemerdekaan, termasuk insiden Rengasdengklok yang memicu percepatan rencana. Pada 7 Agustus 1945, Soekarno membacakan Proklamasi Kemerdekaan Republik Indonesia yang

ditandatangani bersama Mohammad Hatta. Proklamasi itu kemudian diikuti pembentukan pemerintahan awal dan komite persiapan kemerdekaan.

Second 2

Thailand merupakan sebuah monarki konstitusional dengan raja sebagai kepala negara dan perdana menteri sebagai kepala pemerintahan. Sejarahnya mencakup kerajaan-kerajaan besar seperti Sukhothai dan Ayutthaya, dan wilayah ini pernah dikenal dengan nama Siam. Berbeda dengan banyak negara tetangga, Thailand berhasil mempertahankan kemerdekaan dari kolonialisasi Barat selama era imperialisme. Politik modern Thailand telah mengalami sejumlah perubahan konstitusional dan periode ketidakstabilan, termasuk kudeta militer dan demonstrasi pro-demokrasi.

Misalnya, untuk contoh data dev, pasangan dari `first[0]` adalah `second[11]`. Bisa dilihat berikut:

```
print(textwrap.fill(first[0], width=80))
print()
print(textwrap.fill(second[11], width=80))
```

Teknik pembuatan batik tradisional melibatkan penggunaan canting untuk menggambar motif halus dan cap (temple) untuk mencetak motif berulang secara lebih cepat. Motif-motif batik sarat makna, mulai dari motif keraton yang melambangkan status dan aturan adat hingga motif rakyat yang mencerminkan kehidupan sehari-hari dan alam. Batik juga memiliki fungsi ritual dan simbolis, banyak dipakai dalam upacara adat, pernikahan, dan kegiatan resmi. Perawatan, pewarnaan alami, dan keterampilan pengerjaan menjadi bagian penting dalam menjaga kualitas batik tradisional.

Pada masa kolonial batik mendapat perhatian luas dan mengalami perubahan produksi serta pasar karena permintaan ekspor dan pengaruh luar. Setelah kemerdekaan, batik terus berkembang baik sebagai identitas nasional maupun komoditas ekonomi, dengan munculnya perancang kontemporer yang menggabungkan unsur tradisi dan modernitas. Pada 2009 batik Indonesia diakui oleh UNESCO sebagai Warisan Budaya Takbenda, yang meningkatkan upaya pelestarian dan promosi. Saat ini batik tetap hidup melalui pendidikan, industri kreatif, dan inisiatif pelestarian yang melibatkan komunitas pengrajin.

TUGAS ANDA

Tentukan pasangan yang benar antara paragraf pada list `first` dan `second`. Lengkapilah fungsi berikut: diberikan daftar `first_list` dan `second_list`, kembalikan list berisi indeks.

Nilai pada posisi `i` dalam list jawaban harus berisi indeks dari `second` yang merupakan pasangan dari `first[i]`.

Contohnya untuk data **dev**, keluaran pada posisi `0` adalah `11`

```
import random

# TODO: COMPLETE THIS FUNCTION
# given a first paragraph text and a list of second paragraphs, return the most likely pair index
def predict_pair(first_list: list[str], second_list: list[str]):
    answer = []
    N = len(first_list)
    for i in range(N):
        answer.append(random.randint(0, N - 1))

    return answer
```

✓ Melakukan Prediksi Pada Data Dev

Potongan fungsi di bawah ini akan melakukan prediksi di semua data, dan jika labelnya diberikan, maka juga akan menghitung akurasinya.

```
def run_and_evaluate(first, second, golds):
    preds = predict_pair(first, second)

    # print accuracy from a list of gold label
    if golds is not None:
        correct = 0
        for i in range(len(preds)):
            if preds[i] == golds[i]:
                correct += 1
        print("Accuracy:", 100 * correct / len(preds), "%")
```

Mari kita panggil fungsi evaluasinya. Perhatikan bahwa kunci jawaban hanya tersedia untuk data **dev** saja.

```
# golds is indexed from 1 - N, so we offset by 1
golds = [(int(x) - 1) for x in load_dataset("afaji/wikipair-dev-answer")['answer']['label']]

run_and_evaluate(first, second, golds)

Accuracy: 4.0 %
```

PENILAIAN: Melakukan Prediksi Pada Data Test

Tugas Anda sebenarnya adalah untuk mencocokkan dokumen di data **test** yang berisi 300 dokumen total.

Namun karena alasan keamanan dan agar Ammar tidak iseng lagi, Aji hanya akan membuka akses ke data ini di 1 jam terakhir kompetisi!

Untuk sementara, Anda bisa bermain-main dengan data **dev** di atas dulu!

Jadi, bagian program berikut hanya bisa dijalankan 1 jam terakhir saja!

```
first_test, second_test = read_wikipair_texts("afaji/wikipair-test")
run_and_evaluate(first_test, second_test, None)
```

```
-----
DatasetNotFoundError                                Traceback (most recent call last)
/tmp/ipython-input-3701705708.py in <cell line: 0>()
----> 1 first_test, second_test = read_wikipair_texts("afaji/wikipair-test")
      2 run_and_evaluate(first_test, second_test, None)

-----
↳ 4 frames -----
/usr/local/lib/python3.12/dist-packages/datasets/load.py in dataset_module_factory(path, revision, download_config, download_mode,
data_dir, data_files, cache_dir, **download_kwargs)
   978         ) from e
   979     except RepositoryNotFoundError as e:
--> 980         raise DatasetNotFoundError(f"Dataset '{path}' doesn't exist on the Hub or cannot be accessed.") from e
   981     try:
   982         api.hf_hub_download(

DatasetNotFoundError: Dataset 'afaji/wikipair-test' doesn't exist on the Hub or cannot be accessed.
```

Penilaian

Peserta dengan akurasi terbaik akan mendapatkan nilai sempurna (100). Baseline akurasi adalah 0%. Peserta dengan performa akurasi sama atau lebih kecil dari baseline akan mendapatkan 0.

Selain itu, peserta mendapatkan nilai yang dinormalisasi:

$100 * (\text{akurasi_peserta} - \text{baseline}) / (\text{best_peserta} - \text{baseline})$

Baseline adalah 0%