

deteksi :



CreativeBook

Karya 5
November 2024

Perancangan,
Pembuatan dan
Pengujian Software



Dewan V1.0
Deteksi Awan

Python, Computer
Vision, Digital Image
Processing

Zidan Rasyidi Lazuardani
Author



Programming



www.zidcreative.com
email : zidan@zidcreative.com

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

أَلَمْ تَرَ أَنَّ اللَّهَ يُزْجِي سَحَابًا ثُمَّ يُؤَلِّفُ بَيْنَهُ ثُمَّ يَجْعَلُهُ رُكَّامًا فَتَرَى الْوَدْقَ يَخْرُجُ مِنْ خِلَالِهِ وَيُنَزِّلُ مِنَ السَّمَاءِ مِنْ جِبَالٍ فِيهَا مِنْ بَرَدٍ فَيُصِيبُ بِهِ مَنْ يَشَاءُ وَيَصْرِفُهُ عَنْ مَنْ يَشَاءُ يَكَادُ سَنَا بَرْقُهُ يَذْهَبُ بِالْأَبْصَارِ ﴿٤٣﴾

Terjemah Kemenag 2019

43. Tidakkah engkau melihat bahwa sesungguhnya Allah mengarahkan awan secara perlahan, kemudian mengumpulkannya, lalu menjadikannya bertumpuk-tumpuk. Maka, engkau melihat hujan keluar dari celah-celahnya. Dia (juga) menurunkan (butiran-butiran) es dari langit, (yaitu) dari (gumpalan-gumpalan awan seperti) gunung-gunung. Maka, Dia menyimpakannya (butiran-butiran es itu) kepada siapa yang Dia kehendaki dan memalingkannya dari siapa yang Dia kehendaki. Kilauan kilatnya hampir-hampir menghilangkan penglihatan.

Pengantar

Cuaca merupakan sebuah fenomena keseharian yang sangat mempengaruhi kehidupan kita sejak berabad-abad lamanya. Akibat dari proses pengamatan yang terus menerus, kita dapat mengerti tentang berbagai bentuk awan beserta apa yang akan terjadi tentang perubahan cuaca di sekitar kita.

Di zaman modern, telah banyak dan akurat berbagai macam prediksi cuaca – contoh yang kita peroleh dari BMKG misalnya -, telah sangat membantu kita mengetahui lebih dini kondisi cuaca lengkap dengan perkiraan bentuk awannya sehingga kita lebih arang mengamati langit terlebih fenomena cuaca yang saat ini sedang terjadi. Meskipun begitu, dengan ketidakpastian perubahan iklim dan cuaca saat ini seolah membuat kita kembali harus memahami tentang tanda-tanda cuaca yang diberikan oleh alam, terutama awan.

Dalam perkembangan teknologi ini pula segala sesuatu sangat memudahkan salah satunya di bidang pengolahan visual, gambar atau citra sehingga menciptakan peluang besar menggunakannya dalam berbagai hal termasuk salah satunya cuaca. Oleh karena itulah pendeteksian tentang ada tidaknya awan di langit beserta luasannya tentu menjadi peluang besar untuk dikembangkan.

Software ini secara program sebagian besar dibuat dan digenerate oleh AI melalui ChatGPT namun dengan prompt atau perintah yang diberikan adalah berdasarkan hasil kajian, pengembangan dan penyesuaian dari penulis.

Daftar Isi

Pengantar	3
Daftar Isi	4
BAB I : Pendahuluan.....	5
a. Latar Belakang	5
b. Tujuan	6
c. Manfaat	6
d. Batasan Masalah	6
BAB II : Teori.....	7
a. Python	7
b. Digital Image Processing	7
c. Computer Vision.....	8
d. Segmentasi.....	8
BAB III Perancangan	9
a. Diagram Komponen Sistem.....	9
b. Flowchart.....	10
c. Diagram Arsitektur Komponen	12
BAB IV Analisa Program.....	14
a. Pemanggilan Library	14
b. Set Up Aplikasi	14
c. Fungsi - fungsi	18
d. Lokasi dan waktu pelaksanaan	26
BAB V Pengujian.....	28
a. Tampilan perangkat lunak.....	28
b. instalasi dan Penggunaan.....	29
c. Analisa.....	30
BAB VI Penutup	33
a. Kesimpulan.....	33
b. Saran pengembangan.....	33
Daftar Pustaka.....	34

BAB I : Pendahuluan

a. Latar Belakang

Pentingnya informasi cuaca bagi masyarakat tidak pernah lepas dari perkembangan teknologi yang ada. Dengan semakin canggihnya teknologi, kondisi cuaca dapat dideteksi dan dipantau dengan lebih baik secara cepat, akurat, dan akurat. Untuk mengetahui kondisi cuaca dan mengubahnya menjadi informasi, diperlukan cara dan metode untuk mengolahnya hingga menjadi informasi yang berguna bagi suatu wilayah [1].

Awan adalah suatu kumpulan partikel air atau es tampak di atmosfer. Kumpulan partikel tersebut termasuk partikel yang lebih besar, juga partikel kering seperti terdapat pada asap atau debu, juga terdapat di dalam awan [2]. Pergerakan awan sangat mempengaruhi berbagai aktivitas seperti pengamatan langit [3] hingga dapat menjadi indikator kondisi udara dan cuaca di suatu tempat.

Pemantauan dan pengamatan terhadap awan sudah lama dilakukan baik yang dilakukan dengan pengamatan langsung maupun dengan menggunakan citra satelit [4]. Dan dengan berkembangnya teknik pengolahan citra saat ini maka identifikasi awan dapat dimudahkan [5] sehingga penelitian tentang identifikasi awan berpeluang besar untuk diidentifikasi.

Dengan memanfaatkan teknik pengolahan citra digital dengan menggunakan Computer Vision Python, maka dalam karya ini akan dilakukan uji coba mendeteksi liputan awan melalui pengambilan citra / gambar awan secara langsung dalam bentuk video yang dikemas dalam bentuk perangkat lunak sederhana yang bernama **Dewan** – Deteksi Awan -.

Sekilas Wawasan

Kecepatan awan



Awan yang berada di udara tenang memiliki kecepatan jatuh terminal rata-rata sebesar 1,3 cm per detik.

Awan bergerak lebih cepat di ketinggian yang lebih tinggi karena angin lebih kencang di sana.

Awan panas yang dikeluarkan oleh gunung berapi dapat bergerak dengan kecepatan hingga 700 km/jam.

Awan bergerak karena angin yang membawa sebidang udara berawan. Angin dapat ditemukan di semua tingkat atmosfer, mulai dari tanah hingga ketinggian yang lebih tinggi.

b. Tujuan

Tujuan dari pembuatan perangkat lunak **Dewan** ini adalah :

1. Memberikan informasi adanya awan dan/atau liputan awan
2. Memberikan informasi presentase liputan awan dan kondisi langit
3. Menguji keakuratan pendeteksian dengan metode pengolahan citra secara manual

c. Manfaat

Adapun potensi manfaat dari dari pembuatan perangkat lunak **Dewan** ini adalah :

Manfaat umum :

1. memudahkan pengamatan liputan awan beserta presentasinya liputannya
2. memberikan informasi kondisi cuaca / langit berdasarkan kondisi awan

Manfaat bagi peneliti / pengembang :

1. sebagai informasi tentang bagaimana keakuratan pengamatan awan dengan menggunakan metode pengolahan citra digital.

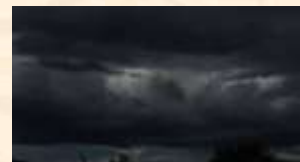
d. Batasan Masalah

Terdapat batasan-batasan pada pembuatan dan pengembangan perangkat lunak **Dewan** ini, diantaranya adalah :

1. Pengujian hanya dilakukan untuk kondisi langit yang perbedaan langit dan awannya bisa terdeteksi dengan jelas
2. Difungsikan hanya untuk mendeteksi keberadaan awan dan liputannya saja.

Sekilas Wawasan**Kondisi cuaca berdasarkan awan**

Awan cirrus: Awan yang menandakan cuaca cerah dan baik. Awan ini memiliki partikel air yang halus seperti serat kapas atau bulu burung.



Awan padat, gelap, dan menjulang tinggi: Menunjukkan bahwa cuaca akan berubah atau memburuk.



Awan terisolasi, tipis, atau sangat tinggi: Menunjukkan bahwa cuaca akan cerah.



Persentase tutupan awan: Tutupan awan 100% berarti hari itu benar-benar berawan, sedangkan tutupan awan 0% berarti hari itu cerah tanpa awan.

BAB II : Teori

a. Python

python adalah bahasa pemrograman yang banyak digunakan untuk melakukan administrasi sistem operasi dan jaringan komputer, pengembangan aplikasi desktop dan web, pengolahan data (*data science* dan *Machine Learning*), dan pembuatan program antarmuka ke perangkat-perangkat keras dan mikrokontroller [6].

Keunggulan python diantaranya memiliki konsep yang bagus dan sederhana, yang berfokus pada kemudahan dalam penggunaan. Kode python dirancang untuk mudah dibaca, dipelajari, digunakan ulang, dan dirawat. Selain itu, python juga mendukung pemrograman berorientasi objek dan fungsional.

b. Digital Image Processing

Citra digital adalah salah satu bentuk representasi visual dari dunia nyata dalam bentuk digital yang dapat dipahami dan diolah oleh komputer. Citra ini terdiri dari elemen-elemen titik yang disebut piksel, yang tersusun dalam baris dan kolom. Setiap piksel memiliki nilai numerik yang menggambarkan tingkat kecerahan atau warna pada posisi tertentu dalam citra [7].

Tujuan utama dari pengolahan citra digital adalah memproses dan memanipulasi citra digital untuk berbagai keperluan. Pengolahan citra ini melibatkan serangkaian operasi yang bertujuan untuk meningkatkan kualitas citra, mengekstraksi informasi penting, dan membuat citra lebih mudah dipahami atau digunakan dalam berbagai aplikasi.

Sekilas Wawasan

Fakta Unik Bahasa Pemrograman Python 1



- Python dinamai bukan dari ular, tapi dari acara komedi Inggris "Monty Python's Flying Circus."
- Python pertama kali dirilis pada tahun 1991, membuatnya lebih tua dari bahasa-bahasa populer seperti Java (1995) dan JavaScript (1995). Meskipun sudah lama, Python terus berkembang dan tetap relevan hingga sekarang.
- Python terkenal karena menggunakan indentasi (whitespace) untuk menentukan blok kode. Ini membuat kode Python lebih rapi dan mudah dibaca, tetapi juga membuatnya berbeda dari bahasa lain yang menggunakan kurung kurawal {}.
- Python termasuk dalam tiga bahasa pemrograman utama yang digunakan di Google, bersama C++ dan Java. Banyak proyek internal Google, termasuk YouTube, dikembangkan dengan Python.

c. Computer Vision

Computer vision adalah bidang yang menggabungkan AI, Machine Learning, Deep Learning, dan CNN supaya komputer dapat melihat dunia, menganalisis data visual dan kemudian membuat keputusan yang sesuai [8].

Penerapan pada computer vision dalam kehidupan sudah banyak seperti Mobil self- driving, google penerjemah, absensi sidik jari, absensi menggunakan pengenalan wajah, dan menganalisis menggunakan sinar-X.

Dalam pengembangan **Dewan** ini, Computer Vision digunakan untuk segmentasi awan, pengukuran liputan awan, penggambaran kontur awan dan visualisasi penyajian data melalui library OpenCV.

d. Segmentasi

Teknik untuk membagi atau memisahkan citra ke dalam beberapa daerah (region) berdasarkan kemiripan atribut yang dimilikinya disebut dengan segmentasi. segmentasi juga disebut sebuah proses yang membagi sebuah citra menjadi sejumlah bagian atau objek [9].

Dalam pengembangan **Dewan**, Segmentasi HSV pada kode tersebut berfungsi untuk memisahkan area yang kemungkinan adalah awan dari langit berdasarkan karakteristik warna dan kecerahan. Penggunaan warna HSV (Hue, Saturation, Value) sangat berguna karena mampu mendeteksi objek secara lebih akurat dalam berbagai kondisi pencahayaan.

Sekilas Wawasan

Fakta Unik Bahasa Pemrograman Python 2

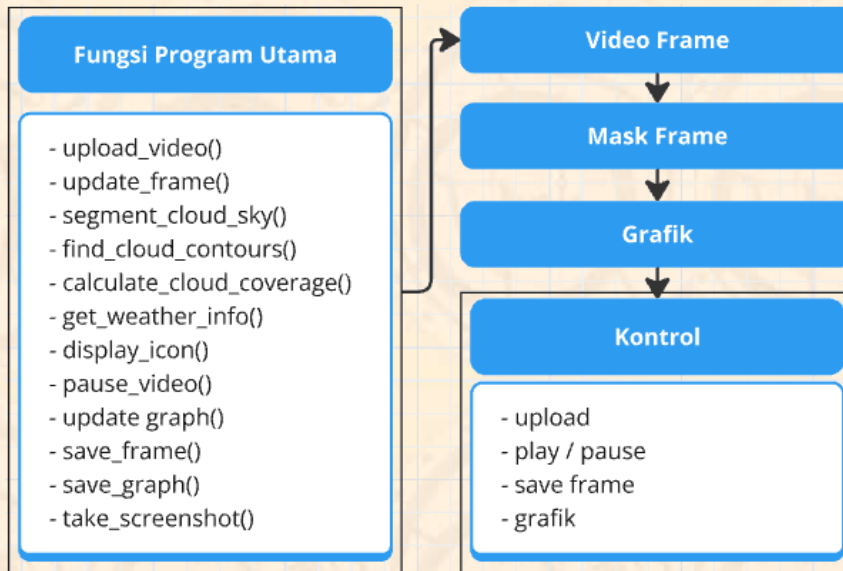


- Python punya versi yang bisa berjalan di microcontroller bernama MicroPython, yang memungkinkan pengembangan perangkat keras seperti IoT menggunakan Python dengan perangkat kecil seperti ESP8266 dan Raspberry Pi Pico.
- Berkas struktur bahasanya, Python bisa digunakan untuk membuat kode yang terbaca seperti puisi atau kalimat alami. Bahkan ada kompetisi untuk membuat code poetry menggunakan Python.
- NASA menggunakan Python untuk banyak tugas di laboratorium dan simulasi. Bahkan, beberapa alat analisis data di ISS dibangun menggunakan Python.
- Python mendukung beberapa paradigma pemrograman: prosedural, berorientasi objek, dan fungsional. Ini membuat Python sangat fleksibel dan dapat disesuaikan dengan kebutuhan pengembang.

BAB III Perancangan

a. Diagram Komponen Sistem

Berikut ini adalah diagram komponen sistem dari **Dewan**



GAMBAR 1, DIAGRAM KOMPONEN SISTEM

Berikut adalah penjelasannya :

1. Fungsi-fungsi pada utama aplikasi ini yang mencakup metode-metode penting untuk memproses dan menampilkan data deteksi awan (detail akan dibahas di bab berikutnya)
2. Video Frame: Frame video diambil langsung dari file video dan diolah melalui metode `update_frame()` dan `segment_cloud_sky_hsv()`
3. sebuah mask frame dihasilkan, yang memvisualisasikan area awan yang terdeteksi. Mask ini adalah hasil pemrosesan video frame menggunakan segmentasi warna HSV.
4. Graph Plot: Diagram ini menunjukkan grafik yang memplot cakupan awan (dalam persentase) terhadap waktu, diperbarui setiap kali frame baru diproses. Hal ini memungkinkan pengguna melihat perubahan cakupan awan dari waktu ke waktu.

Sekilas Wawasan

Fakta unik tentang PyQt



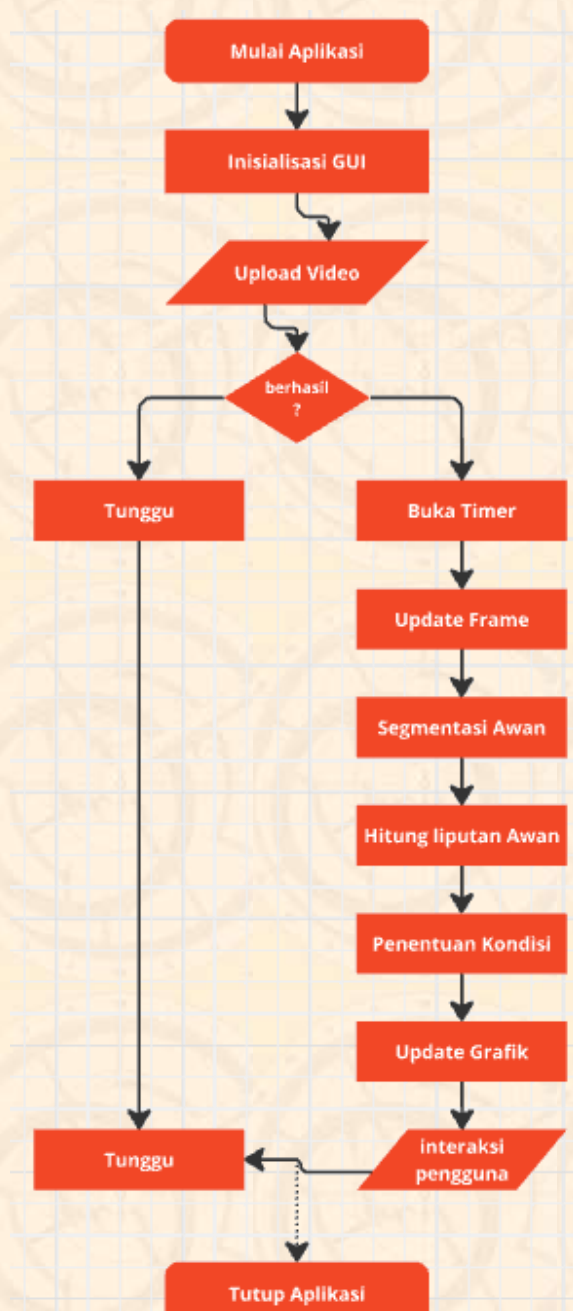
PyQt adalah toolkit populer yang digunakan untuk membangun antarmuka grafis (GUI) dengan Python.

- PyQt memiliki lebih dari 600 kelas dan ribuan fungsi yang mencakup elemen-elemen GUI dari tombol sederhana hingga grafik 2D/3D, editor teks canggih, dan tabel data interaktif.
- Aplikasi yang dibangun dengan PyQt bisa dijalankan di berbagai sistem operasi tanpa perubahan kode, termasuk Windows, macOS, dan Linux.
- PyQt mendukung QML (Qt Modeling Language), bahasa deklaratif yang mirip dengan JavaScript untuk membuat GUI dinamis. Ini memungkinkan pengembang menggunakan QML untuk animasi yang mulus dan interaktif, serta mengintegrasikan Python untuk logika aplikasinya.

5. Tombol kontrol memungkinkan pengguna untuk mengunggah video, memainkan/menghentikan video, menyimpan frame atau grafik, dan mengambil tangkapan layar. Tombol ini memberikan kemudahan akses ke fitur utama aplikasi.

b. Flowchart

berikut adalah flowchart dari **Dewan V1.0**



GAMBAR 2, FLOWCHART

Sekilas Wawasan

Linimasa penelitian deteksi Awan Part 1

Pada tahun 1960-an, satelit cuaca pertama diluncurkan dan mulai merekam gambar Bumi dari luar angkasa, memungkinkan para ilmuwan melihat pola awan secara langsung. Data ini memberikan wawasan yang belum pernah ada sebelumnya tentang awan dan pola atmosfer. Namun, analisis data ini sangat lambat dan memerlukan campur tangan manusia yang intensif.

Di dekade 1980-an, dengan kemajuan dalam ilmu komputer, para ilmuwan mulai mengembangkan algoritma dasar untuk mendeteksi bentuk awan. Mereka menggunakan teknik analisis citra digital untuk mendeteksi pola tertentu dalam foto satelit, seperti bentuk yang menunjukkan awan badai atau cuaca cerah. Meski masih sederhana, teknologi ini menjadi dasar dari sistem deteksi awan modern.

Berikut adalah penjelasannya :

1. Membuka aplikasi
2. Iniasialisasi GUI, Membuka jendela aplikasi PyQt dengan tombol kontrol (Upload Video, Play/Pause, Save Frame, Save Graph, Screenshot) serta area tampilan video, ikon cuaca, dan grafik.
3. Pilih file dan upload video
4. Apakah berhasil ?
 - Jika Ya: Lanjutkan proses dengan inialisasi parameter seperti FPS dan jumlah total frame, serta mengaktifkan tombol kontrol.
 - Jika Tidak: Tampilkan pesan error/tunggu
5. Timer aktif berdasarkan nilai FPS video yang diunggah, dan akan memicu fungsi `update_frame` setiap interval waktu.
6. Update frame, Baca frame dari video. Meliputi segmentasi awan, Ubah frame dari BGR ke HSV, Buat mask untuk mendeteksi awan terang, awan gelap, dan mengabaikan langit biru. Hitung Cakupan Awan (%) dengan menghitung area mask awan.
7. Tentukan Kondisi Cuaca berdasarkan presentase cakupan awan :
 - <10%: Cerah
 - 10-30%: Sedikit Awan
 - 30-50%: Cerah Berawan
 - 50-80%: Berawan
 - 80%: Sangat Berawan (berpotensi mendung)
 - Tampilkan ikon cuaca dan informasi kondisi di GUI
8. Perbarui grafik menggunakan data cakupan awan terhadap waktu, yang disimpan pada setiap frame
9. Tunggu Interaksi Pengguna
 - Play/Pause: Mengontrol timer.

Sekilas Wawasan

Linimasa penelitian deteksi Awan Part 2

Barulah di abad ke-21, ketika teknologi AI dan machine learning berkembang pesat, lahir program deteksi awan berbasis AI. Dengan memanfaatkan jaringan saraf tiruan (neural networks), program-program ini bisa menganalisis gambar satelit secara otomatis, mengidentifikasi jenis awan, memprediksi pergerakannya, dan memperkirakan dampak cuaca yang mungkin terjadi. Deteksi awan modern bisa membedakan awan tipis seperti cirrus dan awan tebal seperti cumulonimbus, bahkan memberikan peringatan dini untuk fenomena cuaca ekstrem.

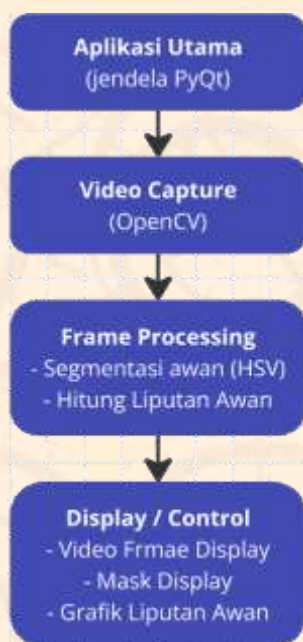
Saat ini, program deteksi awan berbasis AI digunakan oleh badan meteorologi di seluruh dunia. Selain membantu memperbaiki prediksi cuaca harian, teknologi ini juga berperan penting dalam mengatasi perubahan iklim, membantu memantau kebakaran hutan, dan memprediksi pola curah hujan yang berdampak pada pertanian serta kehidupan masyarakat.

- **Save Frame:** Menyimpan frame saat ini sebagai gambar.
- **Save Graph:** Menyimpan grafik cakupan awan.
- **Screenshot:** Menyimpan tampilan GUI sebagai screenshot.

10. Saat pengguna menutup aplikasi, lepaskan sumber daya video dan timer, kemudian akhiri aplikasi.

c. Diagram Arsitektur Komponen

Diagram berikut merinci aliran data utama dan proses inti dalam aplikasi deteksi awan berbasis PyQt dan OpenCV, dengan alur



data yang dibagi menjadi beberapa tahap kunci. Berikut adalah penjelasan setiap bagian :

1. **Aplikasi utama (PyQt)**
 - **Komponen Utama:** "Aplikasi Utama" berperan sebagai jendela utama aplikasi yang dibuat dengan PyQt merupakan antarmuka yang menampilkan output visual dari proses deteksi awan, menyediakan area bagi pengguna untuk berinteraksi, seperti memilih video dan melihat hasil deteksi.
 - **Fungsi Utama:** Mengelola interaksi pengguna dan menampilkan komponen visual yang dihasilkan oleh modul pemrosesan lainnya, seperti tampilan video, masker awan, dan grafik cakupan awan.
2. **Video Capture (OpenCV)**
 - **Komponen Pengambilan Video:** Bagian ini memanfaatkan OpenCV untuk menangkap frame dari file video atau kamera. Setiap frame yang diambil diteruskan ke modul pemrosesan untuk dideteksi awannya.

Sekilas Wawasan

Bagaimana pemrosesan citra Frame by Frame ?

Pertama, video dipecah menjadi frame-frame individual. Misalnya, video dengan 30 frame per detik (fps) akan menghasilkan 30 gambar setiap detiknya

Setiap frame dianalisis menggunakan teknik pemrosesan citra seperti deteksi tepi, pengenalan objek, atau analisis warna. Misalnya, dalam video keamanan, setiap frame dianalisis untuk mendeteksi pergerakan atau objek tertentu.

Algoritma seperti pengenalan wajah, pelacakan objek, atau segmentasi gambar diterapkan pada frame-frame ini. Proses ini memungkinkan komputer untuk mengenali elemen-elemen spesifik di setiap frame, seperti mobil di jalan atau awan di langit.

Setelah setiap frame dianalisis, data dari frame-frame tersebut digabungkan untuk mendapatkan pemahaman yang lebih besar tentang video secara keseluruhan. Misalnya, frame-by-frame dapat membantu menganalisis pergerakan awan secara bertahap dalam video satelit, sehingga kita bisa melihat perubahan cuaca dalam skala waktu.

- **Fungsi Utama:** Menyediakan aliran data secara terus-menerus dari video yang akan diproses oleh aplikasi, sehingga memungkinkan analisis waktu nyata.

3. Frame Processing

- **Segmentasi Awan (HSV):** Modul pemrosesan frame ini menganalisis setiap frame dengan metode segmentasi warna berbasis HSV untuk mendeteksi area awan. Dalam proses ini, area yang berwarna putih dan abu-abu dianalisis sebagai area yang mungkin merupakan awan, sementara warna biru langit disaring.
- **Hitung Cakupan Awan:** Setelah segmentasi, area awan dihitung sebagai persentase dari keseluruhan frame, yang menjadi indikator cakupan awan di langit.
- **Fungsi Utama:** Mengidentifikasi dan menghitung area awan dalam setiap frame video sehingga hasil ini dapat ditampilkan sebagai visualisasi cakupan.

4. Display / Control

- **Video Frame Display dan Mask Display:** Bagian ini menampilkan frame video asli dan frame masker hasil segmentasi, sehingga pengguna dapat melihat perbedaan antara video mentah dan area yang terdeteksi sebagai awan.
- **Grafik Cakupan Awan:** Grafik ini memplot persentase cakupan awan terhadap waktu, menunjukkan perubahan cakupan awan seiring waktu.
- **Fungsi Utama:** Menampilkan hasil analisis dan menyediakan grafik visual cakupan awan, memberikan pengguna informasi tentang cuaca berdasarkan analisis visual awan.

Sekilas Wawasan

Apa itu HSV dalam pengolahan Citra ?

Skala warna HSV adalah sistem warna yang digunakan untuk mendeskripsikan warna dalam pengolahan citra digital berdasarkan tiga parameter: Hue (warna dasar), Saturation (kejenuhan), dan Value (kecerahan).

Hue merepresentasikan warna dasar, seperti merah, hijau, biru, kuning, dll. Hue diukur dalam derajat pada roda warna, dengan skala dari 0° hingga 360° :

- 0° untuk merah,
- 120° untuk hijau,
- 240° untuk biru,
- Nilai-nilai lainnya mewakili transisi warna antara titik-titik ini.

Saturation menunjukkan intensitas atau kemurnian warna. Diukur dari 0% hingga 100%:

- 0% berarti warna yang sangat tidak jenuh atau abu-abu (tidak ada warna).
- 100% berarti warna penuh atau sangat jenuh (intens).

Value merepresentasikan kecerahan warna, dari 0% hingga 100%. Saat Value mendekati 0%, warna akan tampak gelap atau hitam. Saat mendekati 100%, warna menjadi lebih terang.

BAB IV Analisa Program

a. Pemanggilan Library

Dewan.py

```
import cv2
import numpy as np
from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout,
QLabel, QPushButton, QSpacerItem, QSizePolicy, QFileDialog
from PyQt5.QtCore import QTimer, Qt
from PyQt5.QtGui import QImage, QPixmap, QFont
import matplotlib.pyplot as plt
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
FigureCanvas
import time
```

zidCreative

Kode diatas adalah bagian import dari berbagai modul yang akan digunakan dalam aplikasi untuk melakukan beberapa tugas, yaitu pengolahan citra, antarmuka pengguna grafis, dan pembuatan grafik. Berikut adalah penjelasan untuk setiap pustaka yang diimport:

- OpenCV adalah pustaka *Computer Vision* untuk pengolahan citra dan video. Dalam aplikasi ini, OpenCV digunakan untuk membaca video, memproses setiap frame, melakukan segmentasi awan berdasarkan warna, dan menemukan kontur awan.
- NumPy adalah pustaka yang mendukung komputasi numerik. Ini sering digunakan untuk operasi array dan manipulasi data dalam aplikasi berbasis citra karena citra diwakili sebagai array numerik. Dalam kode ini, NumPy digunakan untuk membuat masker warna dalam segmentasi awan dan operasi bitwise untuk menyaring awan.
- PyQt5 adalah pustaka GUI untuk Python, dan di sini digunakan untuk membuat antarmuka pengguna.
- Matplotlib adalah pustaka plotting 2D yang digunakan untuk membuat grafik dalam aplikasi.
- Modul time adalah pustaka standar Python yang digunakan untuk operasi terkait waktu. Di sini, fungsi `time.strftime` digunakan untuk menghasilkan timestamp saat menyimpan grafik agar nama file unik berdasarkan waktu saat grafik tersebut disimpan.

b. Set Up Aplikasi

Dewan.py

```
class CloudDetectionApp(QWidget):
```

zidCreative

Kode diatas adalah bagian dari Python yang mendefinisikan sebuah kelas bernama CloudDetectionApp yang didalamnya terdapat beberapa fungsi

Dewan.py

```
def __init__(self):
    super().__init__()

    self.cap = None # Tidak ada video yang dimuat pertama kali
    self.fps = 30 # Default FPS
    self.timer_interval = int(1000 / self.fps)
    self.total_frames = 0 # Total frame dari video
    self.current_frame = 0 # Frame saat ini

    # Data untuk grafik
    self.coverage_data = []
    self.time_data = []

    # Layout dan widget PyQt
    self.initUI()

    # Timer untuk membaca frame secara berkala
    self.timer = QTimer(self)
    self.timer.timeout.connect(self.update_frame)
```

zidCreative

kode ini adalah setup awal untuk aplikasi yang akan:

1. Memuat dan memproses video, frame demi frame, menggunakan OpenCV.
2. Menghitung persentase cakupan awan per frame.
3. Menggunakan timer untuk mengatur interval pemrosesan frame, sesuai FPS video.
4. Menyimpan data cakupan awan dari setiap frame untuk ditampilkan dalam grafik.

Dewan.py

```
def initUI(self):
    self.setWindowTitle('zidCreative - Dewan | Deteksi Awan v1.0')

    # Layout utama
    self.layout = QVBoxLayout()

    # Layout untuk video frame dan mask
    self.hbox = QHBoxLayout()
```

```

# QLabel untuk menampilkan video frame dan mask
self.video_frame_label = QLabel(self)
self.mask_frame_label = QLabel(self)

# Layout untuk ikon cuaca dan logo
self.weather_layout = QHBoxLayout()

# QLabel untuk menampilkan ikon cuaca
self.weather_icon_label = QLabel(self)

# Layout untuk teks cuaca
self.weather_text_layout = QVBoxLayout()

# QLabel untuk menampilkan cakupan awan
self.awan_label = QLabel(self)
self.coverage_label = QLabel(self)
self.condition_label = QLabel(self)
self.garis_label = QLabel(self)

# Set font untuk label
font = QFont('Arial', 12, QFont.Bold)
self.awan_label.setFont(font)
self.coverage_label.setFont(font)
self.condition_label.setFont(font)
self.garis_label.setFont(font)

# Tambahkan label ke layout
self.weather_text_layout.addWidget(self.awan_label)
self.weather_text_layout.addWidget(self.coverage_label)
self.weather_text_layout.addWidget(self.condition_label)
self.weather_text_layout.addWidget(self.garis_label)

# Tambahkan pengurangan jarak antar label
self.weather_text_layout.addSpacing(2)

# QLabel untuk menampilkan logo
self.logo_label = QLabel(self)

# Tambahkan QLabel ikon cuaca dan logo ke layout weather
self.weather_layout.addWidget(self.weather_icon_label)
self.weather_layout.addLayout(self.weather_text_layout)

# Tambahkan spacer ke layout
self.weather_layout.addItem(QSpacerItem(40, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum))

# Muat logo dan set ukurannya
logo_pixmap = QPixmap('source/dewan.png')

```

```

        logo_pixmap = logo_pixmap.scaled(280, 280, Qt.KeepAspectRatio,
Qt.SmoothTransformation)
        self.logo_label.setPixmap(logo_pixmap)

        # Tambahkan logo ke layout weather
        self.weather_layout.addWidget(self.logo_label)

        # Tambahkan layout weather ke layout utama
        self.layout.addLayout(self.weather_layout)

        # Tambahkan QLabel ke layout horizontal
        self.hbox.addWidget(self.video_frame_label)
        self.hbox.addWidget(self.mask_frame_label)

        # Tambahkan hbox ke layout utama
        self.layout.addLayout(self.hbox)

        # Layout untuk tombol kontrol
        self.control_layout = QHBoxLayout()

        # Tombol untuk kontrol (misalnya, Play, Pause)
        self.control_btn = QPushButton('Pause', self)
        self.control_btn.clicked.connect(self.pause_video)
        self.control_btn.setEnabled(False) # Dinonaktifkan hingga video
dimuat
        self.control_layout.addWidget(self.control_btn)

        # Tombol Simpan Frame
        self.save_frame_btn = QPushButton('Simpan Frame', self)
        self.save_frame_btn.clicked.connect(self.save_frame)
        self.save_frame_btn.setEnabled(False) # Dinonaktifkan hingga video
dimuat
        self.control_layout.addWidget(self.save_frame_btn)

        # Tombol Simpan Grafik
        self.save_graph_btn = QPushButton('Simpan Grafik', self)
        self.save_graph_btn.clicked.connect(self.save_graph)
        self.save_graph_btn.setEnabled(False) # Dinonaktifkan hingga video
dimuat
        self.control_layout.addWidget(self.save_graph_btn)

        # Tombol Screenshot
        self.screenshot_btn = QPushButton('Screenshot', self)
        self.screenshot_btn.clicked.connect(self.take_screenshot)
        self.screenshot_btn.setEnabled(False) # Dinonaktifkan hingga video
dimuat
        self.control_layout.addWidget(self.screenshot_btn)

        # Tombol Upload Video

```

```

self.upload_btn = QPushButton('Upload Video', self)
self.upload_btn.clicked.connect(self.upload_video)
self.control_layout.addWidget(self.upload_btn)

# Tambahkan layout tombol kontrol ke layout utama
self.layout.addLayout(self.control_layout)

# Inisialisasi grafik
self.figure = plt.figure(figsize=(10, 5))
self.canvas = FigureCanvas(self.figure)
self.layout.addWidget(self.canvas)

# Set layout untuk aplikasi
self.setLayout(self.layout)

```

zidCreative

Kode ini membentuk antarmuka pengguna yang terdiri dari:

- Bagian tampilan video dan mask hasil deteksi awan.
- Bagian cuaca yang menampilkan ikon cuaca, informasi cakupan awan, dan logo aplikasi.
- Tombol kontrol untuk memutar, menjeda, menyimpan frame, menyimpan grafik, dan mengambil screenshot.
- Grafik untuk menunjukkan data cakupan awan pada tiap frame video.

c. Fungsi - fungsi

Dewan.py

```

def upload_video(self):
    file_name, _ = QFileDialog.getOpenFileName(self, "Pilih Video", "",
"Video Files (*.mp4 *.avi *.mov)")
    if file_name:
        self.cap = cv2.VideoCapture(file_name)
        if not self.cap.isOpened():
            print("Error: Video tidak dapat dibuka.")
            return
        self.fps = self.cap.get(cv2.CAP_PROP_FPS)
        self.timer_interval = int(1000 / self.fps)
        self.total_frames = int(self.cap.get(cv2.CAP_PROP_FRAME_COUNT))
        self.current_frame = 0

        # Reset data grafik
        self.coverage_data = []
        self.time_data = []

        # Aktifkan tombol kontrol setelah video diunggah

```

```

self.control_btn.setEnabled(True)
self.save_frame_btn.setEnabled(True)
self.save_graph_btn.setEnabled(True)
self.screenshot_btn.setEnabled(True)

# Mulai timer untuk memproses video
self.timer.start(self.timer_interval)

```

zidCreative

metode `upload_video` memungkinkan pengguna untuk memilih file video, membuka video tersebut, mendapatkan informasi dasar (seperti FPS dan total frame), mengaktifkan kontrol antarmuka untuk pemutaran video, dan memulai timer yang memungkinkan video diputar dan diproses frame-by-frame dalam aplikasi.

Dewan.py

```

def update_frame(self):
    if self.cap is None:
        return

    ret, frame = self.cap.read()
    if not ret:
        self.timer.stop()
        self.cap.release()
        return

    # Segmentasi awan menggunakan HSV
    cloud_mask_hsv = self.segment_cloud_sky_hsv(frame)

    # Cari kontur awan dan tambahkan keterangan cuaca
    self.find_cloud_contours(cloud_mask_hsv, frame)
    cloud_coverage_percentage =
self.calculate_cloud_coverage(cloud_mask_hsv)

    # Simpan data untuk grafik
    self.coverage_data.append(cloud_coverage_percentage)
    self.time_data.append(self.current_frame / self.fps)

    weather_text, icon_path =
self.get_weather_info(cloud_coverage_percentage)

    # Tambahkan keterangan cuaca pada frame
    self.awan_label.setText('-----
-----')

    self.coverage_label.setText(f'Liputan awan :
{cloud_coverage_percentage:.2f} %')
    self.condition_label.setText(weather_text)

```

```

self.garis_label.setText('-----')

# Tampilkan ikon cuaca di QLabel
self.display_icon(icon_path)

# Konversi frame BGR ke RGB untuk PyQt
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
rgb_mask = cv2.cvtColor(cloud_mask_hsv, cv2.COLOR_GRAY2RGB)

# Tampilkan frame yang sudah dimodifikasi di PyQt
self.display_frame(rgb_frame, self.video_frame_label)
self.display_frame(rgb_mask, self.mask_frame_label)

# Update grafik
self.update_graph()

# Update frame count
self.current_frame += 1

```

zidCreative

Fungsi `update_frame` adalah bagian utama dari aplikasi yang melakukan pemrosesan setiap frame dalam video. Fungsinya mencakup:

- Membaca frame dari video.
- Melakukan segmentasi awan dan menghitung cakupan awan.
- Mengambil dan menampilkan informasi cuaca, ikon cuaca, dan persentase cakupan awan.
- Mengkonversi dan menampilkan frame di antarmuka PyQt.
- Memperbarui grafik yang menunjukkan perubahan cakupan awan selama pemutaran video.

Dengan mengulang fungsi ini untuk setiap frame, aplikasi dapat menganalisis dan menampilkan informasi cuaca secara real-time.

Dewan.py

```

def display_frame(self, frame, label):
    # Konversi frame ke QImage
    h, w, ch = frame.shape
    bytes_per_line = ch * w
    q_img = QImage(frame.data, w, h, bytes_per_line,
QImage.Format_RGB888)

```

```
# Tampilkan di QLabel
label.setPixmap(QPixmap.fromImage(q_img).scaled(500, 1000,
Qt.KeepAspectRatio))
```

zidCreative

Fungsi `display_frame` ini mengonversi gambar (frame) dari format OpenCV ke QImage, mengubah ukuran gambar agar sesuai dengan label tanpa mengubah rasio aspek, dan menampilkannya pada elemen QLabel. Dengan menggunakan fungsi ini, gambar hasil pemrosesan video dapat ditampilkan secara dinamis pada aplikasi PyQt.

Dewan.py

```
def display_icon(self, icon_path):
    # Muat ikon cuaca
    pixmap = QPixmap(icon_path)

    # Set ukuran ikon
    pixmap = pixmap.scaled(100, 100, Qt.KeepAspectRatio,
Qt.SmoothTransformation)

    # Tampilkan ikon di QLabel
    self.weather_icon_label.setPixmap(pixmap)
```

zidCreative

Fungsi `display_icon` ini memuat ikon cuaca dari path yang diberikan, mengatur ukurannya agar pas dengan dimensi 100 x 100 piksel (menjaga rasio aspek), dan menampilkannya pada label `self.weather_icon_label`. Fungsi ini digunakan untuk memperbarui ikon cuaca di antarmuka aplikasi, sehingga pengguna dapat melihat cuaca terkini dalam bentuk gambar atau simbol visual

Dewan.py

```
def pause_video(self):
    if self.timer.isActive():
        self.timer.stop()
        self.control_btn.setText('Play')
    else:
        self.timer.start(30)
        self.control_btn.setText('Pause')
```

zidCreative

Fungsi `pause_video` berfungsi sebagai tombol toggle play/pause:

- Saat video diputar, fungsi ini menghentikannya dan mengubah teks tombol menjadi "Play".
- Saat video sedang berhenti, fungsi ini memulai kembali pemutaran video dan mengubah teks tombol menjadi "Pause".

Dewan.py

```
def segment_cloud_sky_hsv(self, frame):
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # Rentang awan terang (putih)
    lower_cloud_bright = np.array([0, 0, 200])
    upper_cloud_bright = np.array([180, 60, 255])

    # Rentang awan gelap (abu-abu)
    lower_cloud_dark = np.array([0, 0, 100])
    upper_cloud_dark = np.array([180, 100, 255])

    # Rentang langit biru cerah yang perlu diabaikan
    lower_sky_blue = np.array([100, 40, 50]) # Disesuaikan untuk langit
    upper_sky_blue = np.array([140, 255, 255])

    # Masker untuk awan terang
    mask_bright = cv2.inRange(hsv, lower_cloud_bright,
upper_cloud_bright)

    # Masker untuk awan gelap
    mask_dark = cv2.inRange(hsv, lower_cloud_dark, upper_cloud_dark)

    # Masker untuk langit biru
    mask_sky = cv2.inRange(hsv, lower_sky_blue, upper_sky_blue)

    # Gabungkan masker awan terang dan gelap
    cloud_mask = cv2.bitwise_or(mask_bright, mask_dark)

    # Hilangkan area langit dari masker awan
    cloud_mask = cv2.bitwise_and(cloud_mask, cloud_mask,
mask=cv2.bitwise_not(mask_sky))

    return cloud_mask
```

zidCreative

Fungsi ini melakukan segmentasi awan berdasarkan warna menggunakan ruang warna HSV:

- Mendapatkan area yang berpotensi awan dengan mendeteksi warna putih terang dan abu-abu.
- Menghilangkan area langit biru yang tidak diperlukan dari hasil deteksi.

Hasilnya adalah mask yang menandai area awan dalam gambar atau frame video.

Dewan.py

```
def find_cloud_contours(self, cloud_mask_hsv, frame):
    contours, _ = cv2.findContours(cloud_mask_hsv, cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
    cv2.drawContours(frame, contours, -1, (0, 255, 0), 2) # Gambar
    kontur
```

zidCreative

Fungsi ini:

- Mencari kontur area awan pada masker biner yang dihasilkan.
- Menggambar kontur tersebut pada frame asli dalam bentuk garis hijau.

Hasilnya adalah tampilan frame dengan garis hijau di sekitar area awan, sehingga area awan menjadi lebih terlihat dan dapat dipantau dalam video atau gambar.

Dewan.py

```
def calculate_cloud_coverage(self, cloud_mask_hsv):
    cloud_area = cv2.countNonZero(cloud_mask_hsv)
    total_area = cloud_mask_hsv.shape[0] * cloud_mask_hsv.shape[1]
    coverage_percentage = (cloud_area / total_area) * 100
    return coverage_percentage
```

zidCreative

Fungsi ini menghitung persentase cakupan awan dalam gambar dengan:

- Menghitung jumlah piksel area awan (cloud_area).
- Menghitung total piksel dalam gambar (total_area).
- Menghitung persentase cakupan dengan (cloud_area / total_area) * 100.

Nilai persentase ini bisa digunakan untuk menunjukkan seberapa besar area gambar yang tertutup oleh awan, yang dapat ditampilkan dalam UI atau digunakan dalam analisis lebih lanjut.

Dewan.py

```
def get_weather_info(self, cloud_coverage_percentage):
    if cloud_coverage_percentage < 10:
```

```

        return 'Cerah', 'source/cerah.png'
    elif cloud_coverage_percentage < 30:
        return 'Sedikit Awan', 'source/sedawan.png'
    elif cloud_coverage_percentage < 50:
        return 'Cerah Berawan', 'source/cerawan.png'
    elif cloud_coverage_percentage < 80:
        return 'Berawan', 'source/awan.png'
    else:
        return 'Sangat berawan berpotensi Mendung', 'source/mendung.png'

```

zidCreative

Fungsi ini menggunakan nilai cakupan awan untuk menentukan kondisi cuaca dan mengembalikan deskripsi serta ikon yang sesuai. Ini memungkinkan antarmuka pengguna untuk menampilkan deskripsi cuaca yang relevan dan ikon yang sesuai berdasarkan data cakupan awan yang dianalisis dari video atau gambar.

Dewan.py

```

def update_graph(self):
    self.figure.clear()
    ax = self.figure.add_subplot(111)
    ax.plot(self.time_data, self.coverage_data, label='Cakupan Awan',
color='b', marker='o')
    ax.set_xlabel('Waktu (detik)')
    ax.set_ylabel('Cakupan Awan (%)')
    ax.set_title('Grafik Cakupan Awan')
    ax.set_ylim(0, 100)
    ax.legend()
    ax.grid()

    self.canvas.draw()

```

zidCreative

Fungsi `update_graph` bertugas untuk menggambar atau memperbarui grafik yang menampilkan cakupan awan seiring waktu. Dengan menggunakan data waktu dan cakupan, serta pengaturan yang sesuai, fungsi ini membuat visualisasi yang jelas dan informatif. Fungsi ini bisa digunakan dalam aplikasi untuk menampilkan data yang berubah seiring waktu, seperti dalam monitoring cuaca atau analisis lingkungan.

Dewan.py

```

def save_frame(self):
    # Mengatur posisi frame ke current_frame untuk mengambil frame yang
    tepat

```

```

self.cap.set(cv2.CAP_PROP_POS_FRAMES, self.current_frame)
ret, frame = self.cap.read()
if ret:
    options = QFileDialog.Options()
    filename, _ = QFileDialog.getSaveFileName(self, "Simpan Frame",
    "", "Images (*.png *.jpg *.jpeg);;All Files (*)", options=options)
    if filename:
        cv2.imwrite(filename, frame)
        print(f"Frame disimpan sebagai {filename}")
    else:
        print("Tidak dapat menyimpan frame.")

```

zidCreative

Fungsi `save_frame` bertugas untuk menyimpan frame tertentu dari video yang sedang diproses ke dalam file gambar. Fungsi ini melakukan pengaturan posisi frame, membaca frame dari video, membuka dialog untuk meminta nama dan lokasi penyimpanan, dan kemudian menyimpan frame tersebut jika proses berhasil. Ini sangat berguna dalam aplikasi yang berkaitan dengan pengolahan video, analisis, atau aplikasi yang memerlukan pengambilan gambar dari video.

Dewan.py

```

def save_graph(self):
    timestamp = time.strftime("%Y%m%d-%H%M%S")
    graph_filename = f'grafik_cakupan_awan_{timestamp}.png'
    self.figure.savefig(graph_filename)
    print(f"Grafik disimpan sebagai {graph_filename}")

```

zidCreative

Fungsi `save_graph` bertugas untuk menyimpan grafik yang telah dibuat sebagai file PNG dengan nama yang mencakup timestamp saat grafik disimpan. Dengan cara ini, setiap kali fungsi ini dipanggil, pengguna dapat menyimpan versi baru dari grafik tersebut tanpa menimpa file yang sudah ada, serta mendapatkan informasi waktu yang tepat saat grafik tersebut disimpan. Ini sangat berguna untuk dokumentasi dan pelacakan perubahan grafik dari waktu ke waktu.

Dewan.py

```

def take_screenshot(self):
    screenshot_filename, _ = QFileDialog.getSaveFileName(self, "Ambil Screenshot", "", "Images (*.png);;All Files (*)")
    if screenshot_filename:

```

```
self.grab().save(screenshot_filename)
print(f"Screenshot disimpan sebagai {screenshot_filename}")
```

zidCreative

Fungsi `take_screenshot` bertugas untuk mengambil tangkapan layar dari aplikasi dan menyimpannya sebagai file gambar dalam format PNG. Dengan membuka dialog untuk memilih lokasi dan nama file, pengguna dapat dengan mudah menyimpan tangkapan layar yang diambil. Pesan di konsol juga memberikan umpan balik bahwa proses penyimpanan telah berhasil, sehingga pengguna tahu di mana untuk menemukan file yang disimpan. Ini sangat berguna dalam konteks aplikasi yang memerlukan dokumentasi visual atau berbagi informasi yang ditampilkan dalam antarmuka pengguna.

Dewan.py

```
def closeEvent(self, event):
    if self.cap:
        self.cap.release()
    event.accept()
```

zidCreative

Fungsi `closeEvent` bertujuan untuk menangani peristiwa penutupan jendela aplikasi dengan cara yang aman dan efisien. Fungsi ini memastikan bahwa objek capture (`self.cap`) yang digunakan untuk menangkap video dilepaskan sebelum aplikasi ditutup, sehingga semua sumber daya dibersihkan dengan benar. Dengan cara ini, fungsi ini membantu menjaga kestabilan aplikasi dan mencegah potensi masalah yang dapat terjadi akibat sumber daya yang tidak terlepas.

d. Lokasi dan waktu pelaksanaan

Dewan.py

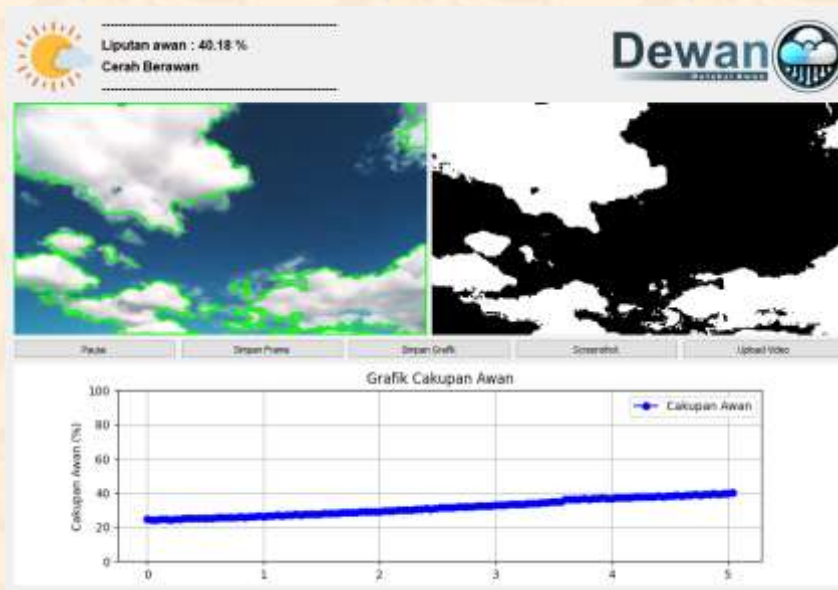
```
if __name__ == '__main__':
    app = QApplication([])
    window = CloudDetectionApp()
    window.show()
    app.exec_()
```

zidCreative

Kode di atas merupakan struktur dasar untuk menjalankan aplikasi GUI menggunakan PyQt (atau PySide). Kode ini memastikan bahwa aplikasi dijalankan dengan benar, jendela yang diperlukan dibuat dan ditampilkan, serta aplikasi siap untuk berinteraksi dengan pengguna. Dengan menggunakan pola ini, Anda dapat mengembangkan aplikasi GUI yang kompleks dengan banyak fitur dan antarmuka interaktif.

BAB V Pengujian

a. Tampilan perangkat lunak



GAMBAR 3, TAMPILAN DEWAN

Dewan memiliki tampilan antarmuka yang sederhana namun tetap fungsional sebagai berikut :

- Nama dan logo dewan menunjukkan nama dan singkatan perangkat lunak yang berarti “ Deteksi Awan “
- Logo dan keterangan liputan awan dan kondisi cuaca akan otomatis berubah berdasarkan kondisi liputan awan yang berhasil dideteksi oleh sistem, menampilkan informasi cuaca, presentase liputan awan dan gambar kondisi cuaca
- Dua frame utama. Sebelah kanan menunjukkan frame hasil pemrosesan segmentasi HSV dan Contour sedangkan frame sebelah kiri mengimplementasikan hasil deteksi ke dalam video utama.
- Dibawah frame terdapat tombol kontrol yang meliputi play / pause, simpan frame, simpan grafik, screenshot dan upload video.
- Grafik presentasi liputan awan terhadap waktu yang diperbarui seiring dengan berjalannya video.

Sekilas Wawasan

Aplikasi cuaca dengan tampilan / antarmuka terbaik



Windy (atau Windy.com) adalah aplikasi dan platform web yang memberikan data cuaca secara real-time dan prediksi cuaca yang cukup akurat, dengan visualisasi yang interaktif dan menarik.

2 fitur utama yang sangat menarik adalah :

- **Peta Cuaca Interaktif:** Menyajikan data cuaca dalam bentuk peta global yang dapat diperbesar dan diperkecil, dengan tampilan visual untuk angin, curah hujan, dan elemen cuaca lainnya.
- **Animasi Aliran Angin dan Ombak:** Memvisualisasikan aliran angin, gelombang, dan arus air yang dinamis dan terperinci.

b. instalasi dan Penggunaan

Instalasi

Berikut adalah informasi singkat penggunaan perangkat lunak Dewan :

1. Download software di situs resmi zidcreative <https://www.zidcreative.com/app/>



GAMBAR 4, WEB UNTUK MENGUNDUH DEWAN

2. Instal file mysetup.exe di laptop / komputer kamu dengan cara klik “next” hingga proses instalasi selesai
3. Software dewan siap digunakan

Penggunaan

Berikut adalah informasi singkat penggunaan perangkat lunak Dewan :

1. Buka dan jalankan **Dewan**, ketika pertama kali dibuka, maka tampilan akan seperti berikut



GAMBAR 5, TAMPILAN DEWAN KETIKA PERTAMA KALI DIBUKA

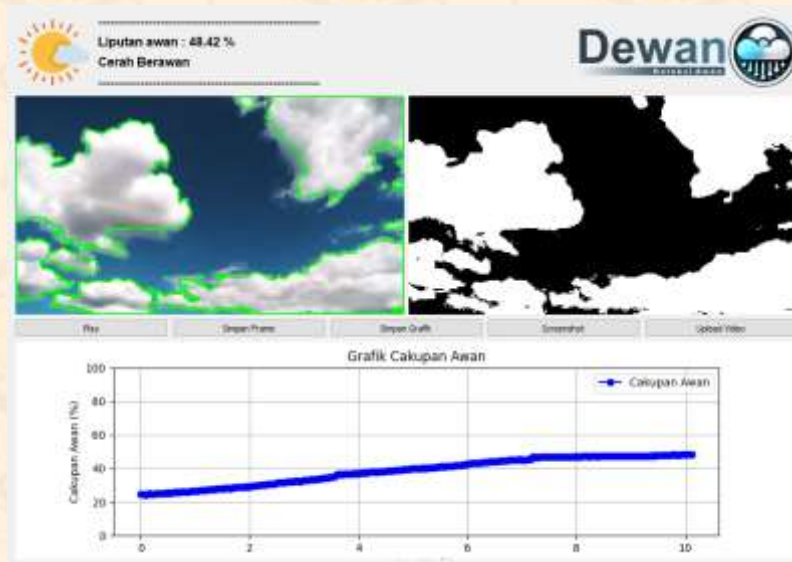
Sekilas Wawasan

.exe

Format aplikasi .exe, atau "executable," adalah format file yang digunakan pada sistem operasi Windows untuk menjalankan program.

- Dikembangkan untuk Sistem Operasi DOS: Format .exe pertama kali muncul untuk sistem operasi MS-DOS di tahun 1980-an. Kemudian, format ini diadopsi oleh Windows, sehingga tetap digunakan secara luas hingga saat ini.
- Portable Executable (PE): File .exe pada Windows berbasis pada struktur Portable Executable (PE), yang memungkinkan file ini dapat dijalankan di berbagai versi Windows, dari Windows NT hingga Windows 10 dan seterusnya.
- Kemampuan untuk Memuat Libraries Eksternal: File .exe dapat memuat file pustaka eksternal atau Dynamic Link Libraries (DLL) untuk menjalankan fungsi tambahan. Ini memungkinkan aplikasi untuk berbagi kode dan mengurangi ukuran file, yang membuat .exe fleksibel dan efisien.

2. Klik tombol upload video dan pilih file video yang bergambar awan
3. Maka secara otomatis video akan diputar dan software akan mendeteksi keberadaan awan dengan tampilan seperti berikut



GAMBAR 6, TAMPILAN DEWAN SETELAH BERHASIL DIJALANKAN

c. Analisa

Uji coba Dewan dilakukan dengan memasukkan sampel awan dari video *time lapse* pergerakan awan dengan durasi waktu tertentu untuk dilakukan pendeteksian terhadap keberadaan an liputan awan. Berikut ini adalah beberapa cuplikan hasil :



Sekilas Wawasan

Time Lapse



Time-lapse adalah teknik video yang menangkap perubahan lambat di lingkungan atau objek dengan memperlambatnya, menciptakan ilusi bahwa waktu berjalan jauh lebih cepat.

Menggabungkan Fotografi dan Film: Time-lapse melibatkan pengambilan serangkaian foto yang diambil secara berkala dengan interval waktu yang tetap (misalnya, setiap detik atau menit) lalu dijadikan video.

Penggunaan di Berbagai Bidang: Time-lapse digunakan tidak hanya dalam film atau seni visual tetapi juga dalam sains, kedokteran, dan dokumentasi. Misalnya, dalam biologi, time-lapse digunakan untuk mengamati sel yang membelah, pertumbuhan organisme, atau proses kimia yang lambat. Dalam meteorologi, teknik ini sering digunakan untuk merekam pergerakan awan dan pola cuaca.



GAMBAR 7, SAMPEL PENDETEKSIAN AWAN

Dari gambar tersebut terlihat bahwa :

1. Dewan berhasil mendeteksi keberadaan awan beserta presentase liputan awannya terhadap langit cerah di sekitarnya
2. Menandai bagian awan dengan garis berwarna hijau setiap kali video bergerak
3. Memberikan keterangan informasi cuaca berdasarkan presentase liputan awan
4. Pada gambar 6 menunjukkan grafik presentase liputan awan terhadap waktu (dalam hal ini durasi video) yang bergerak mengikuti video.

Meskipun begitu, terdapat beberapa kekurangan yang perlu diperbaiki diantaranya yaitu :

1. Karena pendeteksian menggunakan teknik segmentasi HSV yang sangat bergantung pada tingkat warna, maka :
 - Untuk bagian – bagian tertentu pada awan khususnya di bagian yang berwarna mirip warna langit atau bagian tepi yang berwarna tidak begitu kontras maka pendeteksian menjadi tidak akurat

Sekilas Wawasan

Gara-gara mendung jadi Bad mood ?



Mendung dan hujan memang memiliki pengaruh yang cukup kuat terhadap suasana hati atau mood seseorang. Ini terjadi karena beberapa alasan:

Mengurangi Cahaya Matahari: Awan mendung menghalangi sebagian besar cahaya matahari, yang menurunkan kadar serotonin di otak. Serotonin adalah neurotransmitter yang terkait dengan perasaan bahagia dan nyaman.

Rasa Nyaman dan Relaksasi: Banyak orang merasa bahwa suara hujan yang jatuh menenangkan. Suara hujan memiliki ritme alami yang lembut, yang bisa mengurangi kecemasan dan membuat perasaan lebih nyaman dan rileks.

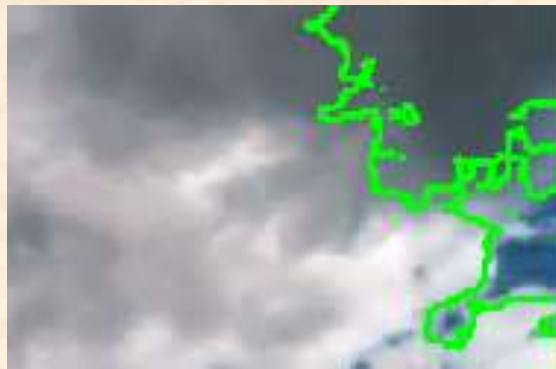
Meningkatkan Kreativitas dan Refleksi: Kondisi mendung dan hujan seringkali membuat suasana lebih tenang dan mendorong refleksi diri



GAMBAR 8, KETIDAKAKURATAN KARENA KONTRAS

Hal ini bisa diatasi dengan mengubah rentang kecerlangan awan pada fungsi segmentasi atau menambahkan metode deteksi tepi / garis tepi.

- Pada awan mendung / gelap keakuratan menjadi berkurang karena fungsi akan mengenali awan mendung sama dengan langit karena sama-sama gelap



GAMBAR 9, KETIDAKAKURATAN AWAN MENDUNG

- Keakuratan pendeteksian akurat jika warna langit cerah (kebiru-biruan), sedangkan jika warna langit putih, senja atau fajar maka keakuratan berkurang



GAMBAR 10, KETIDAKAKURATAN LANGIT SENJA

Sekilas Wawasan

Senja atau Fajar ?



Senja dan fajar adalah dua momen transisi cahaya alami yang terjadi setiap hari, tetapi keduanya memiliki perbedaan yang unik.

Efek Warna Langit: perbedaan partikel udara dan polusi yang ada membuat Senja sering terlihat lebih merah atau oranye karena lebih banyak panjang gelombang merah saat matahari terbenam. Fajar cenderung lebih ke arah biru muda atau ungu karena atmosfer lebih bersih pada malam hari.

Temperatur Lingkungan: Suhu udara cenderung lebih dingin saat fajar dibandingkan saat senja. Hal ini disebabkan oleh efek pendinginan malam hari, dimana panas matahari yang tersimpan di permukaan bumi hilang selama malam.

Efek Psikologis yang Berbeda: Senja sering kali diasosiasikan dengan perasaan tenang atau bahkan melankolis, karena ini menandai akhir dari hari. Di sisi lain, fajar sering diasosiasikan dengan harapan dan awal baru, yang membuat orang merasa termotivasi.

BAB VI Penutup

a. Kesimpulan

Dewan V1.0 telah dapat membantu mendeteksi keberadaan awan secara visual dengan bantuan segmentasi pada pengolahan citra digital, hal ini tentu diharapkan agar pemantauan awan dengan teknologi secara digital / otomatis tidak hanya dilakukan dari atas / satelit saja namun juga dapat dilakukan di permukaan bumi sehingga dapat memantau lebih akurat tentang kondisi cuaca khususnya perubahan dan liputan awan.

b. Saran pengembangan

Terdapat beberapa kekurangan yang telah disebutkan dalam pengembangan dewan V1.0. Diharapkan kedepannya teknologi semacam ini bisa terus dikembangkan sehingga mencapai keakuratan yang diinginkan. Terdapat beberapa metode yang sangat mungkin dikembangkan untuk meningkatkan keakuratan pendeteksian seperti pendeteksian garis tepi, thresholding yang lebih kompleks hingga menggunakan *Machine Learning*.

Fitur-fitur lanjutan juga sangat mungkin dikembangkan seperti konversi citra ke bentuk CSV / Spreadsheet agar dapat dianalisa, generate video pada frame yang telah terolah sehingga menciptakan visual yang lebih baik dan sebagainya untuk meningkatkan fungsi dan antarmuka yang lebih baik.

Sekilas Wawasan

Pemantauan awan oleh satelit



Pemantauan awan oleh satelit adalah salah satu teknologi yang memberikan kita informasi penting mengenai cuaca, iklim, dan kondisi atmosfer:

Menggunakan Berbagai Spektrum Cahaya: Satelit cuaca tidak hanya mengandalkan cahaya tampak; mereka juga menggunakan spektrum inframerah dan gelombang mikro untuk mendeteksi awan.

Mengukur Ketinggian dan Suhu Awan: Satelit dapat memperkirakan ketinggian awan dengan mengukur suhu puncak awan. Semakin tinggi awan, biasanya semakin dingin suhunya.

Pengamatan dengan Resolusi Tinggi: Satelit modern dapat memotret awan dengan resolusi hingga 500 meter atau kurang, memungkinkan ilmuwan untuk melihat detail kecil dalam formasi awan.

Pemantauan Polusi Awan: Satelit dapat mendeteksi partikel aerosol, seperti debu dan polutan yang terperangkap di awan.

Daftar Pustaka

- [1] A. Anjasman, “Pemanfaatan Data Radar Cuaca Untuk Membuat Peringatan Dini Cuaca Secara Spasial Menggunakan Metode K-Means Cluster,” *Newton-Maxwell J. Phys.*, vol. 4, no. 2, pp. 37–41, 2023, doi: 10.33369/nmj.v4i2.29588.
- [2] S. B. Sipayung, “Pemanfaatan data satelit mtsat untuk identifikasi awal musim kering di Indonesia,” vol. 6, no. 1, pp. 18–26, 2011.
- [3] W. Hanggoro, “Terhadap Beberapa Parameter Cuaca,” *J. Meteorol. Dan Geofis.*, vol. 2, NOMOR 2, pp. 137–144, 2011.
- [4] M. A. Massinai, “Analisis Liputan Awan Berdasarkan Citra Satelit Penginderaan Jauh,” *Pertem. Ilm. Tah. MAPIN XIV*, no. September, pp. 208–213, 2019.
- [5] D. Nugraheny, “Metode Nilai Jarak Guna Kesamaan Atau Kemiripan Ciri Suatu Citra (Kasus Deteksi Awan Cumulonimbus Menggunakan Principal Component Analysis),” *Angkasa J. Ilm. Bid. Teknol.*, vol. 7, no. 2, p. 21, 2017, doi: 10.28989/angkasa.v7i2.145.
- [6] B. Raharjo, *Python Untuk Aplikasi Desktop dan Web*, Edisi Revi. Bandung: Informatika, 2019.
- [7] Rohman Dijaya, *Buku Ajar Pengolahan Citra Digital*. Sidoarjo: Umsida Press, 2019.
- [8] D. T. Ananto et al., “Edukasi dan Pelatihan Pengenalan Machine Learning dan Computer Vision Untuk Mengeksplorasi Potensi Visual,” *Pros. Semin. Nas. Pengabd. Masy. LPPM UMJ*, vol. 1, no. 1, pp. 1–8, 2023, [Online]. Available: <https://jurnal.umj.ac.id/index.php/semnaskat/article/view/19491>
- [9] M. Orisa and T. Hidayat, “Analisis Teknik Segmentasi Pada Pengolahan Citra,” *J. Mnemon.*, vol. 2, no. 2, pp. 9–13, 2019, doi: 10.36040/mnemonic.v2i2.84.



www.zidcreative.com