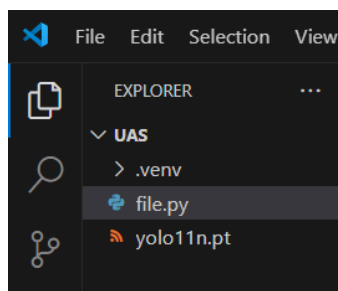


Nama : Spora Della Rosa
Matkul : Kecerdasan Tiruan
Prodi : Ilmu Komputer
Fakultas : Teknik dan Sains

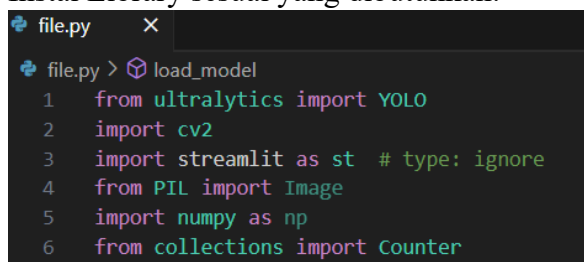
- PERTAMA



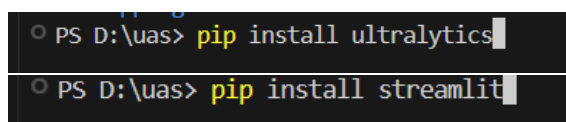
1. Sebelum melakukan deteksi objek, pertama buatlah sebuah folder sesuai nama di Drive C/Drive D, kemudian di isi dengan file yolo11n yang sudah diberikan ,Setelah itu buat file dengan nama File.py.
2. Buka Visual Studio Code dan masukan kode yang sudah di kirim di grup Whatsapp.

- KEDUA

1. Instal Library sesuai yang dibutuhkan:

A screenshot of a code editor window titled 'file.py'. The editor shows the following Python code:

```
1 from ultralytics import YOLO
2 import cv2
3 import streamlit as st # type: ignore
4 from PIL import Image
5 import numpy as np
6 from collections import Counter
```

A screenshot of a terminal window. It shows two commands being executed in a PowerShell prompt (PS):

```
PS D:\uas> pip install ultralytics
PS D:\uas> pip install streamlit
```

Penjelasan:

1. from ultralytics import YOLO:

- **Deskripsi:** Mengimpor kelas YOLO dari paket ultralytics.
 - **Fungsi:** YOLO biasanya digunakan untuk deteksi objek menggunakan algoritma You Only Look Once (YOLO).
2. `import cv2:`
- **Deskripsi:** Mengimpor pustaka cv2.
 - **Fungsi:** cv2 adalah bagian dari OpenCV (Open Source Computer Vision Library) yang digunakan untuk pemrosesan gambar dan video.
3. `import streamlit as st # type: ignore:`
- **Deskripsi:** Mengimpor pustaka streamlit dan memberikan alias st. Komentar `# type: ignore` menginstruksikan alat pemeriksa tipe (seperti mypy) untuk mengabaikan baris ini.
 - **Fungsi:** streamlit adalah pustaka yang digunakan untuk membuat aplikasi web interaktif dengan mudah menggunakan Python. Alias st digunakan untuk mempersingkat penulisan kode.
4. `from PIL import Image:`
- **Deskripsi:** Mengimpor kelas Image dari pustaka PIL (Python Imaging Library).
 - **Fungsi:** PIL digunakan untuk membuka, memanipulasi, dan menyimpan berbagai format gambar. Image adalah kelas utama untuk bekerja dengan gambar.
5. `import numpy as np:`
- **Deskripsi:** Mengimpor pustaka numpy dan memberikan alias np.
 - **Fungsi:** numpy adalah pustaka fundamental untuk komputasi ilmiah di Python. Ini menyediakan array multidimensi dan fungsi matematika yang efisien.
6. `from collections import Counter:`
- **Deskripsi:** Mengimpor kelas Counter dari modul collections.
 - **Fungsi:** Counter adalah subclass dari dictionary yang digunakan untuk menghitung objek yang dapat di-hash. Ini sangat berguna untuk menghitung kemunculan elemen dalam iterable.
- KETIGA

```
# Load YOLO model
@st.cache_resource
def load_model(model_path):
    return YOLO(model_path)
```

Penjelasan :

1. # Load YOLO model:
 - **Deskripsi:** Ini adalah komentar yang menjelaskan bahwa bagian kode ini bertujuan untuk memuat model YOLO.
2. @st.cache_resource:
 - **Deskripsi:** Ini adalah dekorator dari pustaka streamlit yang digunakan untuk menyimpan (cache) hasil dari fungsi yang didekorasi.
 - **Fungsi:** @st.cache_resource memastikan bahwa fungsi load_model hanya akan dijalankan sekali dengan argumen yang sama, dan hasilnya akan disimpan. Jika fungsi dipanggil lagi dengan argumen yang sama, hasil yang disimpan akan digunakan, sehingga meningkatkan efisiensi.
3. def load_model(model_path)::
 - **Deskripsi:** Ini adalah definisi fungsi load_model yang menerima satu parameter model_path.
 - **Fungsi:** Fungsi ini bertujuan untuk memuat model YOLO dari path yang diberikan.
4. return YOLO(model_path):
 - **Deskripsi:** Ini adalah pernyataan di dalam fungsi load_model yang mengembalikan instance dari kelas YOLO yang diinisialisasi dengan model_path.
 - **Fungsi:** Baris ini memuat model YOLO dengan menggunakan path yang diberikan dan mengembalikan instance dari model tersebut.

• KEEMPAT

```
13 # Process and display the detection results
14 def display_results(image, results):
15     boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
16     scores = results.boxes.conf.cpu().numpy() # Confidence scores
17     labels = results.boxes.cls.cpu().numpy() # Class indices
18     names = results.names # Class names
19
20     detected_objects = []
21
22     for i in range(len(boxes)):
23         if scores[i] > 0.5: # Confidence threshold
24             x1, y1, x2, y2 = boxes[i].astype(int)
25             label = names[int(labels[i])]
26             score = scores[i]
27             detected_objects.append(label)
28             cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
29             cv2.putText(image, f"{label}: {score:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
30
31     return image, detected_objects
```

Penjelasan:

1. Kode ini mendefinisikan fungsi bernama **display_results()**. Fungsi ini menerima dua argumen: **image** (gambar input) dan **results** (hasil deteksi objek). Kemudian, kode ini mengekstrak informasi dari objek **results**
2. **boxes** adalah array NumPy yang berisi koordinat kotak pembatas untuk setiap objek yang terdeteksi dalam format **[x1, y1, x2, y2]**.
3. **scores** adalah array NumPy yang berisi skor kepercayaan untuk setiap objek yang terdeteksi.
4. **labels** adalah array NumPy yang berisi indeks kelas untuk setiap objek yang terdeteksi.
5. **names** adalah daftar yang berisi nama kelas dari model deteksi objek.
6. Kode ini mendeklarasikan variabel **detected_objects** sebagai daftar kosong, yang akan digunakan untuk menyimpan nama objek yang terdeteksi.
7. Kode ini adalah loop **for** yang berulang melalui setiap kotak pembatas (**boxes**).
8. **for i in range(len(boxes)):** - Loop ini akan berulang untuk setiap kotak pembatas dalam **boxes**.
9. **if scores[i] > 0.5:** - Kode ini memeriksa apakah skor kepercayaan (**scores[i]**) lebih besar dari 0,5. Jika ya, maka objek tersebut dianggap terdeteksi dengan cukup yakin dan kode akan memproses lebih lanjut.
10. **x1, y1, x2, y2 = boxes[i].astype(int)** - Kode ini mengekstrak koordinat kotak pembatas dari **boxes** dan mengonversinya ke integer.
11. **label = names [int(labels[i])]** - Kode ini mengambil nama kelas objek berdasarkan indeks kelas (**labels[i]**) dari daftar **names**.
12. **score = scores[i]** - Kode ini mengambil skor kepercayaan dari objek yang terdeteksi.

• KELIMA

```

33 # Main Streamlit app
34 def main():
35     st.title("Real-time Object Detection with YOLO")
36     st.sidebar.title("Settings")
37
38     model_path = "yolo11n.pt" # Path to your YOLO model
39     model = load_model(model_path)
40

```

Penjelasan :

1. **# Main Streamlit app:**
 - **Fungsi:** Ini adalah komentar yang menandakan awal program Streamlit.
 - **Penjelasan:** Streamlit adalah framework Python yang memungkinkan Anda membangun aplikasi web interaktif dengan cepat dan mudah. Baris ini adalah komentar yang membantu Anda memahami tujuan program.
2. **def main():**
 - **Fungsi:** Mendefinisikan fungsi utama bernama **main** di mana kode aplikasi utama akan dieksekusi.
 - **Penjelasan:** Dalam Python, fungsi **main** adalah titik awal eksekusi program.
3. **st.title("Real-time Object Detection with YOLO"):**
 - **Fungsi:** Menetapkan judul "Real-time Object Detection with YOLO" untuk aplikasi Streamlit.

- **Penjelasan:** `st` adalah objek Streamlit yang menyediakan berbagai fungsi untuk membangun elemen antarmuka pengguna. Fungsi **`title`** menampilkan judul aplikasi.
4. **`st.sidebar.title("Settings")`:**
 - **Fungsi:** Menampilkan judul "Settings" di sidebar aplikasi Streamlit.
 - **Penjelasan:** **sidebar** adalah fitur di Streamlit yang memungkinkan Anda menambahkan panel samping ke aplikasi Anda. Fungsi **`title`** digunakan untuk menetapkan judul di sidebar.
 5. **`model_path = "yolo11n.pt"`:**
 - **Fungsi:** Mendefinisikan variabel **`model_path`** dan menetapkan jalur ke model YOLO (model "yolo11n.pt").
 - **Penjelasan:** Variabel ini menyimpan jalur ke model YOLO yang akan digunakan untuk mendeteksi objek. Jalur ini kemungkinan mengarah ke file yang berisi model yang terlatih.
 6. **`# Path to your YOLO model`:**
 - **Fungsi:** Ini adalah komentar yang menjelaskan tujuan dari variabel **`model_path`**.
 - **Penjelasan:** Komentar ini membantu Anda memahami bahwa variabel ini menyimpan jalur ke file model YOLO yang Anda gunakan.
 7. **`model = load_model(model_path)`:**
 - **Fungsi:** Memuat model YOLO dari jalur yang ditentukan oleh variabel **`model_path`** dan menetapkan model yang dimuat ke variabel **`model`**.
 - **Penjelasan:** Fungsi **`load_model`** biasanya merupakan fungsi yang disediakan oleh pustaka pembelajaran mesin yang digunakan dalam program ini. Fungsi ini membaca model yang terlatih dari file dan membuatnya siap

- KEENAM

```
# Create the checkbox once
run_detection = st.sidebar.checkbox("Start/Stop Object Detection", key="detection_control")
```

Penjelasan:

1. **`# Create the checkbox once`:** Ini adalah komentar yang menjelaskan tujuan kode berikutnya. Komentar ini menandakan bahwa baris kode berikutnya akan membuat sebuah checkbox.
2. **`run_detection = st.sidebar.checkbox("Start/Stop Object Detection", key="detection_control")`:** Baris kode ini merupakan inti dari kode yang Anda berikan. Ini adalah baris kode yang membuat checkbox yang dapat diaktifkan atau dinonaktifkan. Berikut adalah penjelasan detail masing-masing bagian:
 - **`run_detection`:** Ini adalah variabel yang akan menyimpan status checkbox. Jika checkbox aktif, nilai variabel ini akan menjadi **True**, jika tidak, **False**.
 - **`st.sidebar`:** Ini mengacu pada modul Streamlit yang digunakan untuk membuat interface antarmuka pengguna (UI). **sidebar** adalah elemen UI

Streamlit yang memungkinkan Anda menampilkan widget dan konten di sisi kiri aplikasi Anda.

- **checkbox:** Ini adalah fungsi Streamlit yang digunakan untuk membuat checkbox.
- **"Start/Stop Object Detection":** Ini adalah teks yang akan ditampilkan di sebelah checkbox.
- **key="detection_control":** Atribut ini digunakan untuk memberikan identifikasi unik pada checkbox. Identifikasi ini penting jika Anda memiliki beberapa checkbox dalam aplikasi Anda.

Fungsi Kode

Fungsi utama kode ini adalah untuk membuat sebuah checkbox yang dapat digunakan untuk mengontrol status "Object Detection" .

- **Jika checkbox diaktifkan (di-tick):** Variabel **run_detection** akan bernilai **True**, yang menandakan bahwa "Object Detection" diaktifkan.
- **Jika checkbox dinonaktifkan (tidak di-tick):** Variabel **run_detection** akan bernilai **False**, yang menandakan bahwa "Object Detection" dinonaktifkan.

• KETUJUH

```
# Open video capture if checkbox is active
if run_detection:
    cap = cv2.VideoCapture(0)
    st_frame = st.empty() # Placeholder for video frames
    st_detection_info = st.empty() # Placeholder for detection information

    while True:
        ret, frame = cap.read()
        if not ret:
            st.warning("Failed to capture image.")
            break
```

Penjelasan:

1. Potongan kode yang diberikan ditulis dalam bahasa Python dan menggunakan **cv2**(OpenCV) dan **streamlit** pustaka untuk mengimplementasikan perekaman video dan deteksi objek. Tampaknya ini merupakan bagian dari aplikasi Streamlit untuk deteksi objek secara real-time dari sumber video.
2. **# Open video capture if checkbox is active** Ini adalah komentar yang menjelaskan tujuan blok kode. Komentar ini menunjukkan bahwa kode berikut hanya akan dijalankan saat kotak centang tertentu dicentang, yang memungkinkan proses perekaman video.
3. **if run_detection:** Ini memeriksa apakah **run_detection** variabel, yang mungkin mewakili status kotak centang, adalah Benar. Jika ya, kode berikutnya akan dieksekusi.

4. **cap = cv2.VideoCapture(0)** Baris ini menginisialisasi objek perekaman video (**cap**) menggunakan OpenCV. **0** Argumen menentukan kamera default sebagai sumber video.
5. **st_frame = st.empty()** Baris ini membuat tempat penampung untuk menampilkan bingkai video di aplikasi Streamlit. **st** adalah objek dari **streamlit** perpustakaan, dan **st.empty()** membuat elemen kosong dalam aplikasi.
6. **st_detection_info = st.empty()** Baris ini menciptakan placeholder lain untuk menampilkan hasil deteksi objek. Mirip dengan **st_frame**, baris ini menyediakan ruang khusus di aplikasi Streamlit untuk menampilkan informasi terkait deteksi.
7. **while True:** Ini memulai pengulangan tak terbatas. Pengulangan ini terus-menerus membaca bingkai dari sumber video hingga menemui kondisi untuk keluar dari pengulangan.
8. **ret, frame = cap.read()** Baris ini membaca satu bingkai dari perangkat penangkap video. **ret** adalah bendera boolean yang menunjukkan apakah bingkai berhasil dibaca, sementara **frame** berisi data gambar sebenarnya.
9. **if not ret:** Ini memeriksa apakah **ret** benderanya Salah, yang berarti bingkai tidak berhasil ditangkap.
10. **st.warning("Failed to capture image.")** Jika pengambilan bingkai gagal, baris ini akan menampilkan pesan peringatan di aplikasi Streamlit, yang menunjukkan kesalahan.
11. **break** Baris ini **while** segera keluar dari loop jika terjadi kesalahan penangkapan bingkai, mencegah pemrosesan lebih lanjut terhadap bingkai yang tidak valid.

- KEDELAPAN

```
# Run YOLO detection
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB for display
results = model.predict(frame, imgsz=640) # Perform detection

# Draw results and collect detected objects
frame, detected_objects = display_results(frame, results[0])
```

Penjelasan :

1. **frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)**
 - Fungsi: Mengubah ruang warna gambar dari BGR ke RGB.
 - Penjelasan: YOLO (You Only Look Once) biasanya dibor dengan gambar dalam format BGR (Blue, Green, Red), tetapi untuk menampilkannya di layar, kita perlu mengubahnya ke format RGB (Red, Green, Blue). Fungsi **cv2.cvtColor()** digunakan untuk mengubah format ruang warna.
2. **results = model.predict(frame, imgsz=640)**
 - Fungsi: Melakukan deteksi objek menggunakan model YOLO.
 - Penjelasan: Model YOLO dibor untuk memprediksi kotak pembatas (bounding box) dan kelas objek dalam gambar. Fungsi **model.predict()** digunakan untuk menjalankan model YOLO dan mendapatkan hasil deteksi objek. Parameter **imgsz=640** menentukan ukuran gambar masukan yang akan diproses oleh model.
3. **frame, detected_objects = display_results(frame, results[0])**

- Fungsi: Menggambar hasil deteksi objek pada gambar dan mengumpulkan informasi tentang objek yang terdeteksi.

• KESEMBILAN

```
# Display video feed
st_frame.image(frame, channels="RGB", use_column_width=True)

# Display detection information
if detected_objects:
    object_counts = Counter(detected_objects)
    detection_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()])
else:
    detection_info = "No objects detected."

st_detection_info.text(detection_info) # Update detection info text
```

Penjelasan:

1. # Tampilkan umpan video

- **st_frame.image (frame, channel="RGB", use_column_width=True) :** Kode ini menampilkan frame video dengan menggunakan streamlit.
- **st_frame.image:** Fungsi ini digunakan untuk menampilkan gambar.
- **frame:** Argumen ini adalah frame video yang akan ditampilkan.
- **channels="RGB":** Argumen ini menunjukkan format warna yang digunakan untuk frame video. Pada bagian ini, formatnya adalah RGB (Merah, Hijau, Biru).
- **use_column_width=True:** Argumen ini menandakan bahwa gambar akan menggunakan lebar kolom secara penuh.

2. # Menampilkan informasi deteksi

- **if terdeteksi_objek: :** Kode ini melakukan pengecekan apakah ada objek yang terdeteksi.
- **object_counts = Counter(detected_objects):** Kode ini menghitung jumlah objek yang terdeteksi.
- **Counter:** Fungsi ini menghitung jumlah kemunculan objek yang terdeteksi.
- **detected_objects:** Argumen ini adalah daftar objek yang terdeteksi.
- **deteksi_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()]):** Kode ini menyusun string teks yang akan ditampilkan sebagai informasi deteksi.
- **join:** Fungsi ini menggabungkan semua item dalam daftar menjadi sebuah string.
- **"\n":** Argumen ini menunjukkan bahwa setiap item dalam list akan dipisahkan dengan newline (\n).
- **[f"{obj}: {count}" for obj, count in object_counts.items()]:** Kode ini mengiterasi semua objek yang terdeteksi dan membuat sebuah string dengan format "obj: count" untuk setiap objek.
- **else: :** Kode ini akan dijalankan jika tidak ada objek yang terdeteksi.

- **deteksi_info = "Tidak ada objek yang terdeteksi."**: Kode ini mengatur string teks "Tidak ada objek yang terdeteksi." sebagai informasi deteksi.
- **st_detection_info.text(detection_info)**: Kode ini menampilkan informasi deteksi yang telah dibentuk di bagian sebelumnya dengan menggunakan streamlit.
- **st_detection_info.text**: Fungsi ini digunakan untuk menampilkan teks.
- **detection_info**: Argumen ini adalah string teks yang akan ditampilkan.

- KESEPULUH

```

        # Break the loop if checkbox is unchecked
        if not st.session_state.detection_control:
            break

    cap.release()

if __name__ == "__main__":
    main()

```

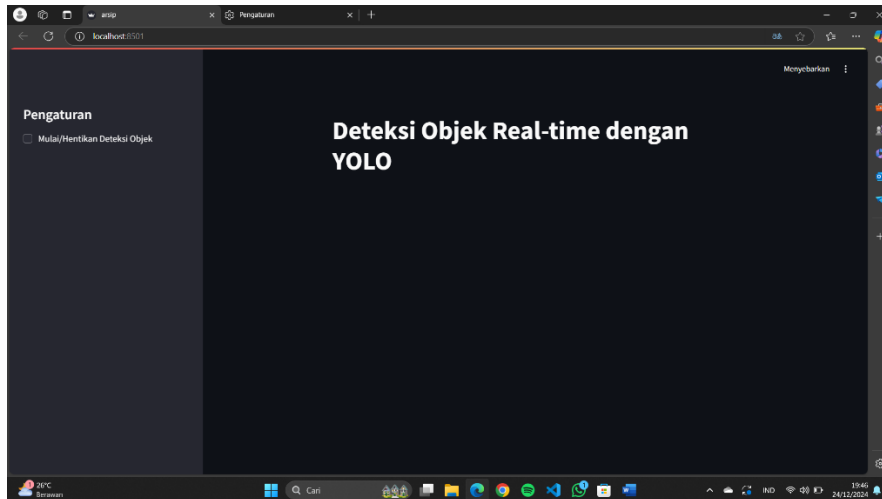
Penjelasan:

1. **# Break the loop if checkbox is unchecked** Baris ini adalah komentar yang menjelaskan tujuan kode berikut.
 - **if not st.session_state.detection_control**: Baris ini memeriksa nilai kontrol kotak centang dalam aplikasi Streamlit.
 - **St** adalah objek Streamlit yang digunakan untuk berinteraksi dengan aplikasi.
 - **session_state** adalah objek seperti kamus yang menyimpan data persisten di seluruh sesi.
 - **detection_control** adalah variabel dalam status sesi yang kemungkinan mewakili status kotak centang.
 - Kode ini memeriksa apakah **detection_control** adalah **False** (misalnya, kotak centang tidak dicentang).
2. **Break**
 - Jika kondisi pada baris sebelumnya benar (kotak centang tidak dicentang), baris ini keluar dari loop yang disertakannya.
3. **cap.release()** Baris ini kemungkinan memanggil fungsi untuk melepaskan sumber daya yang terkait dengan objek penangkapan.
 - Tampaknya kode ini merupakan bagian dari rutinitas pemrosesan atau perekaman video, yang **cap** merepresentasikan objek perekaman.
4. **if name == "main"**: Baris ini memeriksa apakah skrip dijalankan sebagai program utama.
5. **Name** adalah variabel khusus yang menyimpan nama modul atau skrip saat ini.
 - Ini adalah pola umum dalam Python untuk mendefinisikan fungsi yang dieksekusi hanya saat skrip dijalankan secara langsung, bukan saat diimpor sebagai modul.
6. **main()**
 - Baris ini memanggil fungsi bernama **main()**.
 - Fungsi tersebut **main()** mungkin berisi logika inti aplikasi Streamlit atau tugas lainnya.

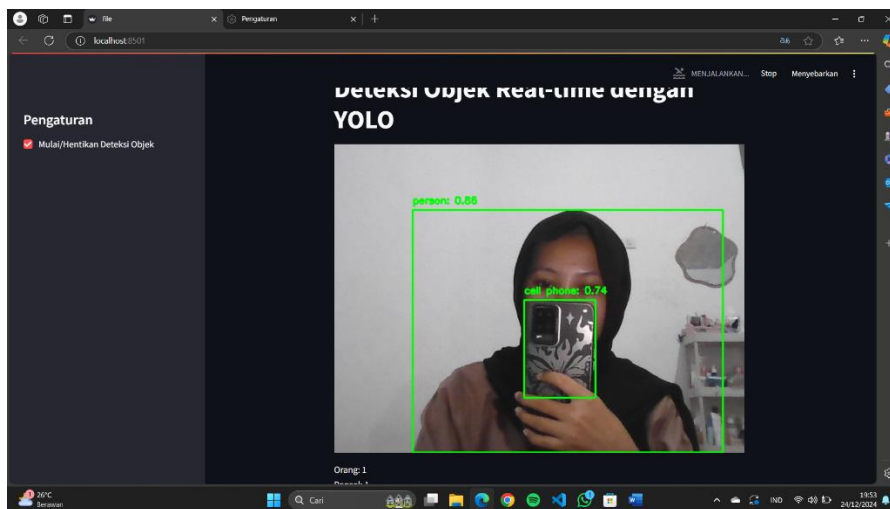
Selesai memasukan semua kode , kita akan menjalankan :

```
PS D:\uas> streamlit run file.py
```

Setelah menjalankannya,kita akan di alihkan ke sebuah website dengan tampilan:



Klik mulai/Hentikan Deteksi Objek, maka tampilannya akan berubah:



SEKIAN TERIMA KASIH