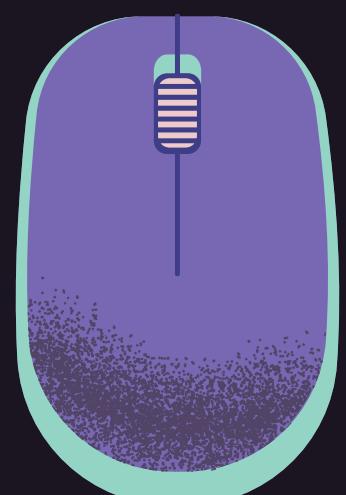
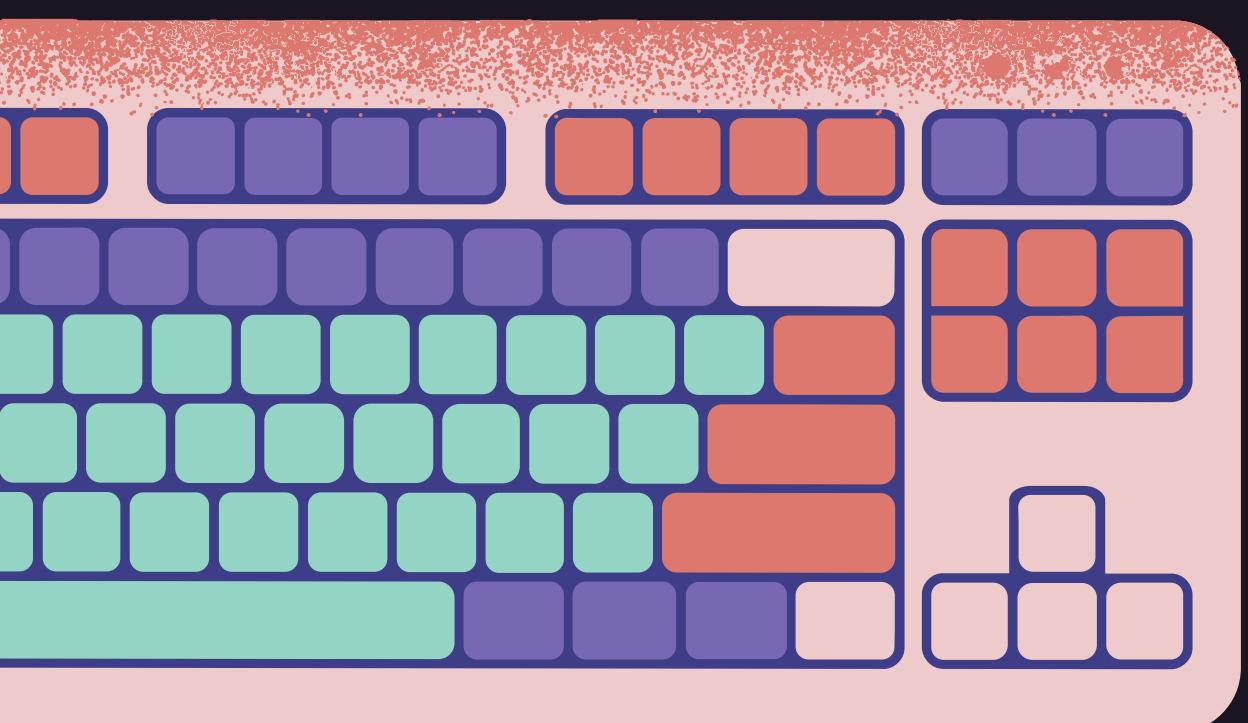


LAPORAN PROJECT QR-CODE DETECTION MENGGUNAKAN OPENCV PYTHON



Nama Kelompok

- 1 . Muhamad Athaozih (23090620043)
- 2 . Ahmad Ilman Nadziron (23090620065)
- 3 . Muhamad Fatan Nayaka (23090620076)

LATAR BELAKANG

Dalam era digital dan otomatisasi industri saat ini, kebutuhan akan sistem identifikasi yang cepat dan akurat semakin meningkat. Salah satu teknologi yang paling banyak digunakan untuk tujuan tersebut adalah QR code. QR-Code berfungsi sebagai media penyimpanan data sederhana yang dapat dibaca oleh mesin, seperti informasi produk, ID barang, atau tautan web. Teknologi ini banyak diterapkan pada bidang perdagangan, logistik, kesehatan, hingga sistem absensi karena kemampuannya dalam mempercepat proses identifikasi dan mengurangi kesalahan manusia.

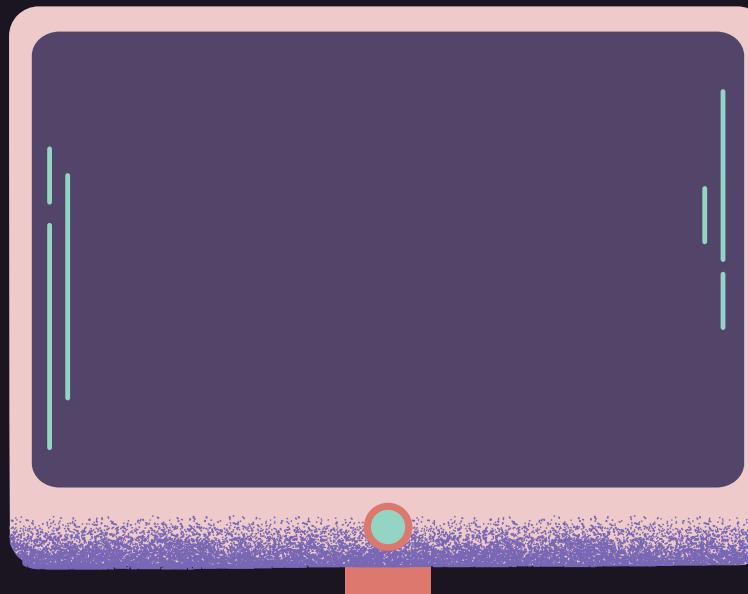
TUJUAN PROJECT

Tujuan Umum

Merancang dan mengimplementasikan sistem pendekripsi serta pembaca QR code berbasis kamera menggunakan OpenCV dan Pyzbar pada Python, yang mampu menampilkan hasil secara real-time dan dapat membuka tautan web dari hasil pemindaian.

Tujuan Khusus

- Menggunakan kamera komputer atau webcam sebagai alat input untuk menangkap citra barcode dan QR code.
- Menerapkan algoritma deteksi dan decoding menggunakan pustaka OpenCV dan Pyzbar agar sistem dapat mengenali berbagai jenis Qr-Code secara akurat.
- Menampilkan hasil deteksi dalam bentuk visual (kotak deteksi dan teks data) secara real-time di layar.
- Menambahkan fitur akses langsung ke situs web (URL) dari hasil pemindaian melalui tombol atau perintah tertentu.
- Mengembangkan sistem yang efisien, fleksibel, dan mudah digunakan tanpa memerlukan perangkat pemindai khusus.



KOMPONEN DAN TOOLS YANG DIGUNAKAN

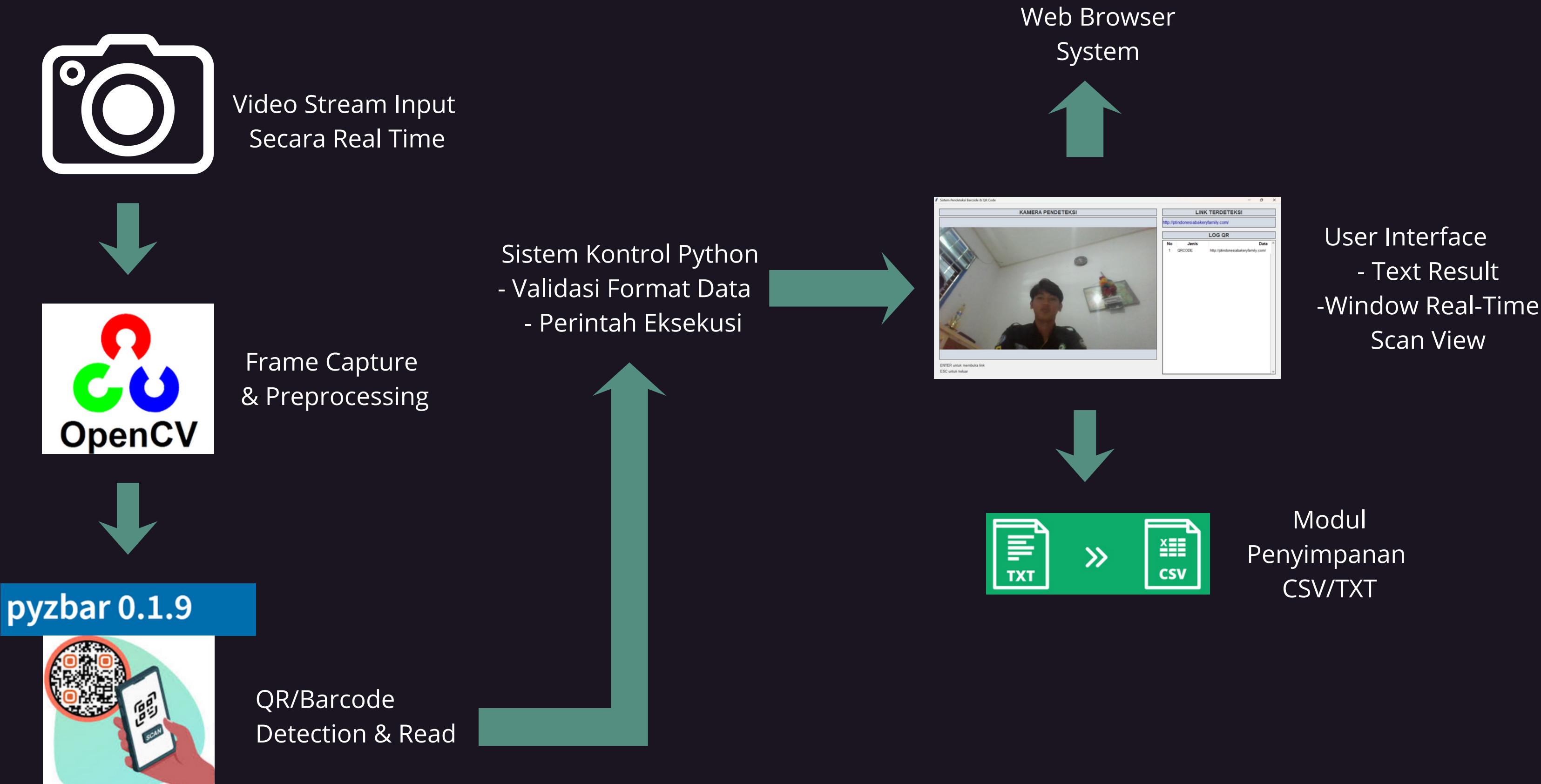
Komponen Hardware

- Laptop / PC – sebagai pusat pemrosesan data dan menjalankan program deteksi barcode.
- Webcam / Kamera Laptop – berfungsi untuk menangkap citra secara real-time yang berisi barcode atau QR code.
- QR Code Sample – digunakan sebagai objek uji coba untuk mendeteksi dan membaca data.

Komponen Software & Library

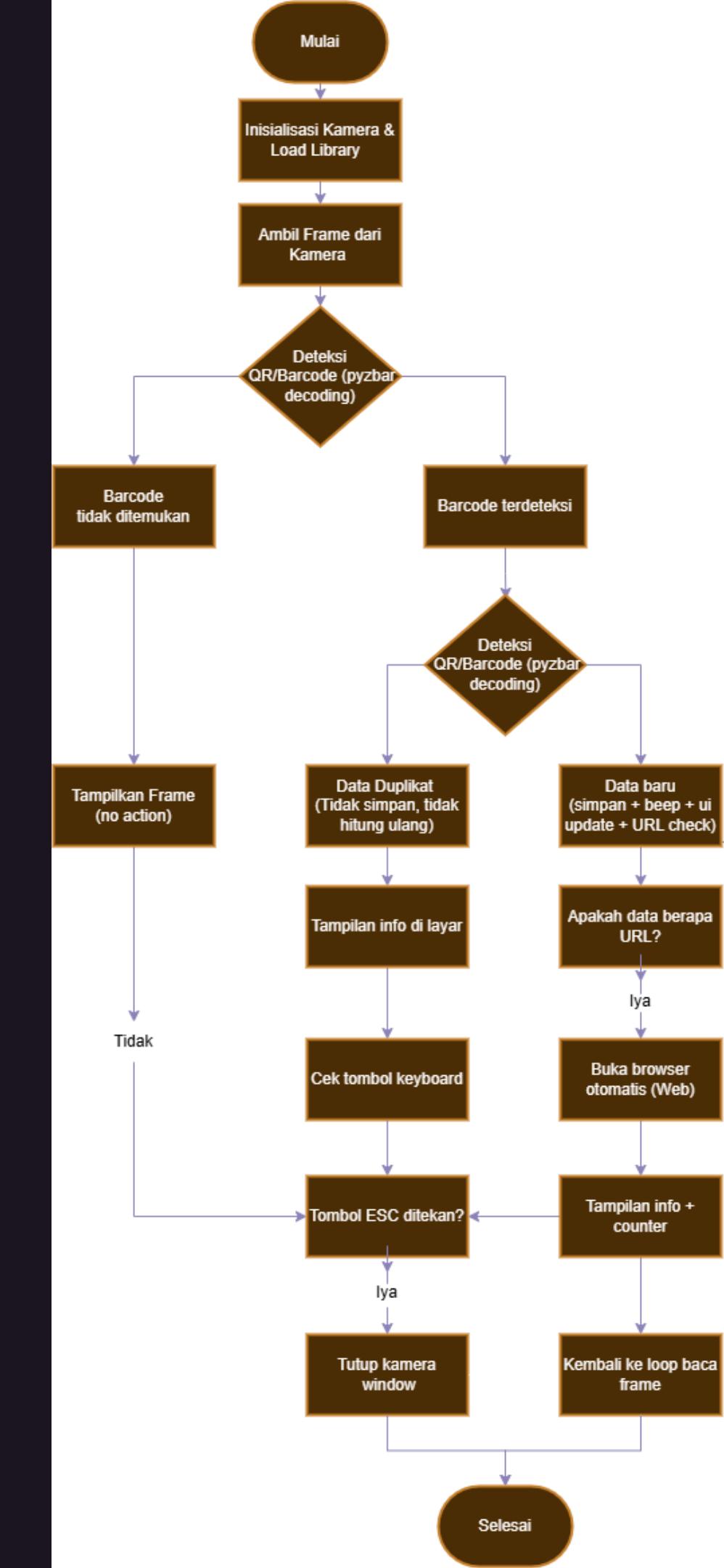
- Python 3.x – bahasa pemrograman utama yang digunakan dalam pengembangan sistem.
- OpenCV (Open Source Computer Vision Library) – digunakan untuk pengolahan citra dan menampilkan hasil deteksi secara visual.
- Pyzbar – pustaka Python untuk membaca dan mendekode berbagai jenis barcode dan QR code.
- Imutils – digunakan untuk mempermudah manipulasi gambar seperti resize atau rotasi frame kamera.
- Webbrowser (Python built-in) – digunakan untuk membuka tautan (URL) hasil pemindaian secara langsung di browser.
- Visual Studio Code / Jupyter Notebook – sebagai IDE (Integrated Development Environment) untuk menulis dan menjalankan kode program.

ARSITEKTUR SISTEM



FLOWCHART

Sistem bekerja dimulai dengan menginisialisasi kamera dan memuat library yang diperlukan. Setelah siap, program akan terus-menerus mengambil frame gambar untuk mendeteksi keberadaan QR Code atau Barcode. Jika kode ditemukan, sistem melakukan validasi untuk membedakan antara data baru atau data duplikat. Untuk data yang teridentifikasi baru, sistem akan menyimpannya, membunyikan notifikasi (beep), serta secara cerdas membuka browser otomatis jika data tersebut berupa tautan URL. Sebaliknya, jika data merupakan duplikat, sistem hanya menampilkan informasi tanpa melakukan pencatatan ulang. Proses scanning ini berjalan secara real-time dalam sebuah siklus (loop) dan hanya akan berhenti serta menutup aplikasi ketika pengguna menekan tombol ESC.



KODE PROGRAM

```
1 import cv2
2 from pyzbar import pyzbar
3 import imutils
4 import webbrowser
5 import csv
6 import os
7 import winsound # Untuk bunyi beep di Windows
8 import tkinter as tk
9 from tkinter import ttk, font
10 from PIL import Image, ImageTk
11
12 # === Variabel global ===
13 last_data = None
14 last_type = None
15 scanned_data = set() # Menyimpan data unik yang sudah terbaca
16 scan_count = 0 # Jumlah total QR/Barcode unik
17
18 # === File output ===
19 TXT_FILE = "hasil_scan.txt"
20 CSV_FILE = "hasil_scan.csv"
21
22 # === Variabel Global GUI ===
23 root = None
24 camera_label = None
25 link_var = None
26 log_tree = None
27 camera = None
28 last_data_displayed = None # Untuk optimasi update link
29
30 # --- PALET WARNA BIRU ---
31 BLUE_BG = "#EAF2FF" # Background umum
32 BLUE_LIGHT = "#D4E4FF" # Untuk frame/panel
33 BLUE_MEDIUM = "#B4D2FF" # Untuk header label
34 BLUE_DARK = "#A7EE2" # Untuk item terpilih
35 BLUE_TEXT = "#001B4D" # Warna teks gelap
36 BLUE_ENTRY_BG = "#F5F9FF" # Warna background entry readonly
37 BLUE_LINK = "#0000AA" # Warna teks link
38
39 # Buat file CSV jika belum ada
40 if not os.path.exists(CSV_FILE):
41     with open(CSV_FILE, mode="w", newline="", encoding="utf-8") as f:
42         writer = csv.writer(f)
43         writer.writerow(["No", "Jenis", "Data"])
44
45
```

```
46 def save_scan_data(code_type, data):
47     """Simpan hasil scan ke file .txt, .csv, dan update GUI log"""
48     global scan_count, log_tree
49     scan_count += 1
50
51     # Simpan ke .txt
52     with open(TXT_FILE, "a", encoding="utf-8") as txt_file:
53         txt_file.write(f"{scan_count}. [{code_type}] {data}\n")
54
55     # Simpan ke .csv
56     with open(CSV_FILE, "a", newline="", encoding="utf-8") as csv_file:
57         writer = csv.writer(csv_file)
58         writer.writerow([scan_count, code_type, data])
59
60     # --- MODIFIKASI: Update GUI Treeview ---
61     if log_tree:
62         log_tree.insert("", tk.END, values=(scan_count, code_type, data))
63         log_tree.yview_moveto(1) # Auto-scroll ke bawah
64
65
66 def beep_sound():
67     """Bunyi beep ketika ada kode baru"""
68     try:
69         duration = 200 # ms
70         freq = 1000 # Hz
71         winsound.Beep(freq, duration)
72     except Exception as e:
73         print(f"Gagal membunyikan beep: {e}")
74
75
76 def detect_codes(frame):
77     """Mendeteksi barcode/QR di frame kamera"""
78     global last_data, last_type
79     codes = pyzbar.decode(frame)
80
81     for code in codes:
82         (x, y, w, h) = code.rect
83         data = code.data.decode("utf-8")
84         code_type = code.type
85
86         # Gambar kotak hijau di sekitar kode
87         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
88         text = f"{code_type}: {data}"
89         cv2.putText(frame, text, (x, y - 10),
90                     cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
```

KODE PROGRAM

```
92     # Jika data baru (belum pernah discan)
93     if data not in scanned_data:
94         scanned_data.add(data)
95         last_data = data
96         last_type = code_type
97         beep_sound() # Bunyi beep
98         save_scan_data(code_type, data) # Simpan ke file & update GUI
99         print(f"[INFO] Data baru terbaca: {data}")
100
101    return frame, codes
102
103
104 def handle_enter_key(event=None):
105     """Fungsi untuk menangani penekanan tombol ENTER."""
106     global last_data, last_type
107
108     if last_data and (last_data.startswith("http://") or last_data.startswith("https://")):
109         print(f"[INFO] Membuka tautan: {last_data}")
110         webbrowser.open(last_data, new=2)
111     elif last_data:
112         print(f"[INFO] Jenis: {last_type}")
113         print(f"[INFO] Data: {last_data}")
114         print("[INFO] Tidak ada URL yang valid untuk dibuka.")
115     else:
116         print("[INFO] Belum ada data yang terbaca.")
117
118
119 def update_frame():
120     """Loop utama untuk mengambil frame kamera dan memperbarui GUI."""
121     global camera, camera_label, last_data, last_data_displayed, link_var, root
122
123     if not camera.isOpened():
124         print("[ERROR] Kamera tidak terbuka, mencoba lagi...")
125         root.after(1000, update_frame)
126         return
127
128     ret, frame = camera.read()
129     if not ret:
130         print("[ERROR] Gagal membaca frame dari kamera.")
131         root.after(1000, update_frame) # Coba lagi setelah 1 detik
132         return
133
134     # Resize frame untuk ditampilkan di GUI (agar tidak terlalu besar)
135     frame = imutils.resize(frame, width=800)
136
137     # Deteksi kode
138     frame_processed, codes = detect_codes(frame)
```

KODE PROGRAM

```
187     camera.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
188     camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
189     print("Kamera berhasil diinisialisasi.")
190
191     # === Setup Jendela Utama ===
192     root = tk.Tk()
193     root.title("Sistem Pendekripsi Barcode & QR Code")
194     root.configure(bg=BLUE_BG) # Warna background
195
196     main_frame = tk.Frame(root, bg=BLUE_BG, padx=20, pady=20)
197     main_frame.pack(fill=tk.BOTH, expand=True)
198
199     # === Kolom Kiri (Kamera & Instruksi) ===
200     left_frame = tk.Frame(main_frame, bg=BLUE_BG)
201     left_frame.pack(side=tk.LEFT, fill=tk.Y, padx=(0, 20))
202
203     # Label untuk "KAMERA PENDEKTRSI"
204     tk.Label(left_frame, text="KAMERA PENDEKTRSI", bg=BLUE_MEDIUM, fg=BLUE_TEXT,
205             font=("Arial", 14, "bold"), anchor="center", relief="solid", borderwidth=1
206             ).pack(fill=tk.X, pady=(0, 5))
207
208     # Label untuk Video Feed
209     camera_label = tk.Label(left_frame, bg=BLUE_LIGHT, relief="solid", borderwidth=1)
210     camera_label.pack(fill=tk.BOTH, expand=True)
211
212     # Instruksi
213     font_instruksi = ("Arial", 10, "bold")
214     tk.Label(left_frame, text="ENTER untuk membuka link", bg=BLUE_BG, fg=BLUE_TEXT, font=font_instruksi).pack(anchor="w", pady=(10, 0))
215     tk.Label(left_frame, text="ESC untuk keluar", bg=BLUE_BG, fg=BLUE_TEXT, font=font_instruksi).pack(anchor="w")
216
217     # === Kolom Kanan (Link & Log) ===
218     right_frame = tk.Frame(main_frame, bg=BLUE_BG)
219     right_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)
220
221     # --- Bagian "LINK TERDETEKSI" ---
222     tk.Label(right_frame, text="LINK TERDETEKSI", bg=BLUE_MEDIUM, fg=BLUE_TEXT,
223             font=("Arial", 14, "bold"), anchor="center", relief="solid", borderwidth=1
224             ).pack(fill=tk.X, pady=(0, 5))
225
226     link_var = tk.StringVar()
227     link_entry = tk.Entry(right_frame, textvariable=link_var, state="readonly",
228                           font=("Arial", 12), bg=BLUE_ENTRY_BG, relief="solid", borderwidth=1,
229                           readonlybackground=BLUE_ENTRY_BG, fg=BLUE_LINK)
230     link_entry.pack(fill=tk.X, ipady=8, pady=(0, 15))
```

KODE PROGRAM

```
232 # --- Bagian "LOG QR" ---
233 tk.Label(right_frame, text="LOG QR", bg=BLUE_MEDIUM, fg=BLUE_TEXT,
234     font=("Arial", 14, "bold"), anchor="center", relief="solid", borderwidth=1
235     ).pack(fill=tk.X, pady=(0, 5))
236
237 log_frame = tk.Frame(right_frame, bg=BLUE_LIGHT, relief="solid", borderwidth=1)
238 log_frame.pack(fill=tk.BOTH, expand=True)
239
240 # Konfigurasi style Treeview
241 style = ttk.Style()
242 style.theme_use("clam") # 'clam' atau 'alt' lebih mudah dikustomisasi
243 style.configure("Treeview.Heading", font=("Arial", 11, "bold"), background=BLUE_MEDIUM, foreground=BLUE_TEXT)
244 style.configure("Treeview", font=("Arial", 10), rowheight=25, background=BLUE_ENTRY_BG, foreground=BLUE_TEXT, fieldbackground=BLUE_ENTRY_BG)
245 style.map('Treeview', background=[('selected', BLUE_DARK)]) # Warna saat dipilih
246 style.configure("Vertical.TScrollbar", background=BLUE_LIGHT, troughcolor=BLUE_BG)
247
248
249 # Buat Treeview (Log)
250 log_tree = ttk.Treeview(log_frame, columns=("No", "Jenis", "Data"), show="headings")
251 log_tree.heading("No", text="No")
252 log_tree.heading("Jenis", text="Jenis")
253 log_tree.heading("Data", text="Data")
254
255 log_tree.column("No", width=50, anchor="center")
256 log_tree.column("Jenis", width=120)
257 log_tree.column("Data", width=400)
258
259 # Scrollbar untuk Treeview
260 scrollbar = ttk.Scrollbar(log_frame, orient=tk.VERTICAL, command=log_tree.yview)
261 log_tree.configure(yscrollcommand=scrollbar.set)
262
263 scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
264 log_tree.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)
265
266 # === Binding Keyboard & Penutupan Jendela ===
267 root.bind('<Return>', handle_enter_key)
268 root.bind('<Escape>', lambda e: on_closing())
269 root.protocol("WM_DELETE_WINDOW", on_closing)
270
271 # === Mulai Loop ===
272 print("Tekan 'ENTER' untuk membuka link (jika URL)")
273 print("Tekan 'ESC' untuk keluar")
274 print("-----\n")
275
276 update_frame() # Mulai video loop
277 root.mainloop()
278
279
280 if __name__ == "__main__":
281     main_gui()
```

KESIMPULAN

Berdasarkan hasil perancangan dan implementasi sistem pemindaian QR Code secara real-time menggunakan Python, OpenCV, dan Pyzbar, dapat disimpulkan bahwa sistem berhasil menjalankan fungsi deteksi dan decoding dengan baik. Kamera berperan sebagai perangkat akuisisi data untuk menangkap objek QR code, kemudian data visual tersebut diproses oleh OpenCV sebagai media akuisisi dan pengelolaan citra. Pyzbar mampu membaca dan menerjemahkan pola QR maupun barcode dalam berbagai format menjadi teks digital. Selain itu, sistem juga mampu menyimpan hasil pemindaian secara otomatis ke dalam file .txt atau .csv serta memberikan umpan balik berupa bunyi beep setiap barcode baru terdeteksi. Fitur tambahan seperti perhitungan jumlah barcode yang berhasil discan dan pembukaan hyperlink otomatis melalui browser juga bekerja sesuai tujuan. Dengan demikian, sistem ini dapat digunakan sebagai solusi pemindaian kode sederhana yang efektif, responsif, serta berpotensi dikembangkan lebih lanjut untuk aplikasi absensi, inventaris, identifikasi produk, dan sistem otomatisasi lainnya.

REFERENSI

Referensi:

1. OpenCV Documentation:

<https://docs.opencv.org>

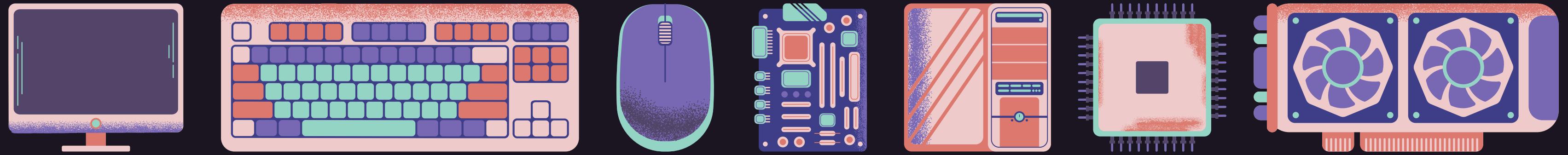
2. Pyzbar Library:

<https://pypi.org/project/pyzbar/>

3. Github Example:

<https://github.com/NaturalHistoryMuseum/pyzbar>





DEMONSTRASI

