

LAMPIRAN

Lampiran 1. Gambar Game Cat Runner



Lampiran 2. Fungsi Main

- Inisialisasi Kamera: Membuka kamera menggunakan cv2.VideoCapture.

```

16 def main():
17     cap = cv2.VideoCapture(0)
18     pose_detector = mp.solutions.pose.Pose()
19
20     mode = 0
21
22     pose_classifier = PoseClassifierWithAngle()
23
24     while True:
25
26         # Process Key (ESC: end)
27         key = cv2.waitKey(10)
28         if key == 27: # ESC
29             break
30
31         mode = select_mode(key, mode)
32
33         ret, image = cap.read()
34         if not ret:
35             break
36
37         image = cv2.flip(image, 1)
38         debug_image = copy.deepcopy(image)

```

- get_landmarks() : Mengambil landmark tubuh yang dideteksi oleh MediaPipe

```

108
109 def get_landmarks(image, poses):
110     landmark_list = []
111     landmark_pose_list = []
112
113     landmarks = poses.landmark
114     frame_height, frame_width, _ = image.shape
115
116     # iterasi setiap landmark pada pose
117     for idh, landmark in enumerate(landmarks):
118         x = int(landmark.x * frame_width)
119         y = int(landmark.y * frame_height)
120
121         # Draw circles on specific landmarks
122         if idh in [11, 12, 13, 14, 15, 16, 23, 24]: # 10s for specific landmarks
123             landmark_list.append([x, y])
124             cv2.circle(image, (x, y), 5, (0, 255, 255), -1)
125
126     landmark_pose_list.append(calc_angle(landmark_list[4], landmark_list[2], landmark_list[0]))
127     landmark_pose_list.append(calc_angle(landmark_list[2], landmark_list[0], landmark_list[1]))
128
129     landmark_pose_list.append(calc_angle(landmark_list[5], landmark_list[3], landmark_list[1]))
130     landmark_pose_list.append(calc_angle(landmark_list[3], landmark_list[1], landmark_list[4]))
131
132     return landmark_pose_list
133

```

- calc_angle() : Menghitung sudut antara tiga titik (landmark) dalam bidang kartesian

```

135 def calc_angle(landmark1, landmark2, landmark3):
136     x1, y1 = landmark1
137     x2, y2 = landmark2
138     x3, y3 = landmark3
139
140     angle = math.degrees(math.atan2(y3 - y2, x3 - x2) - math.atan2(y1 - y2, x1 - x2))
141
142     if angle < 0:
143         angle += 360
144
145     return int(angle)

```

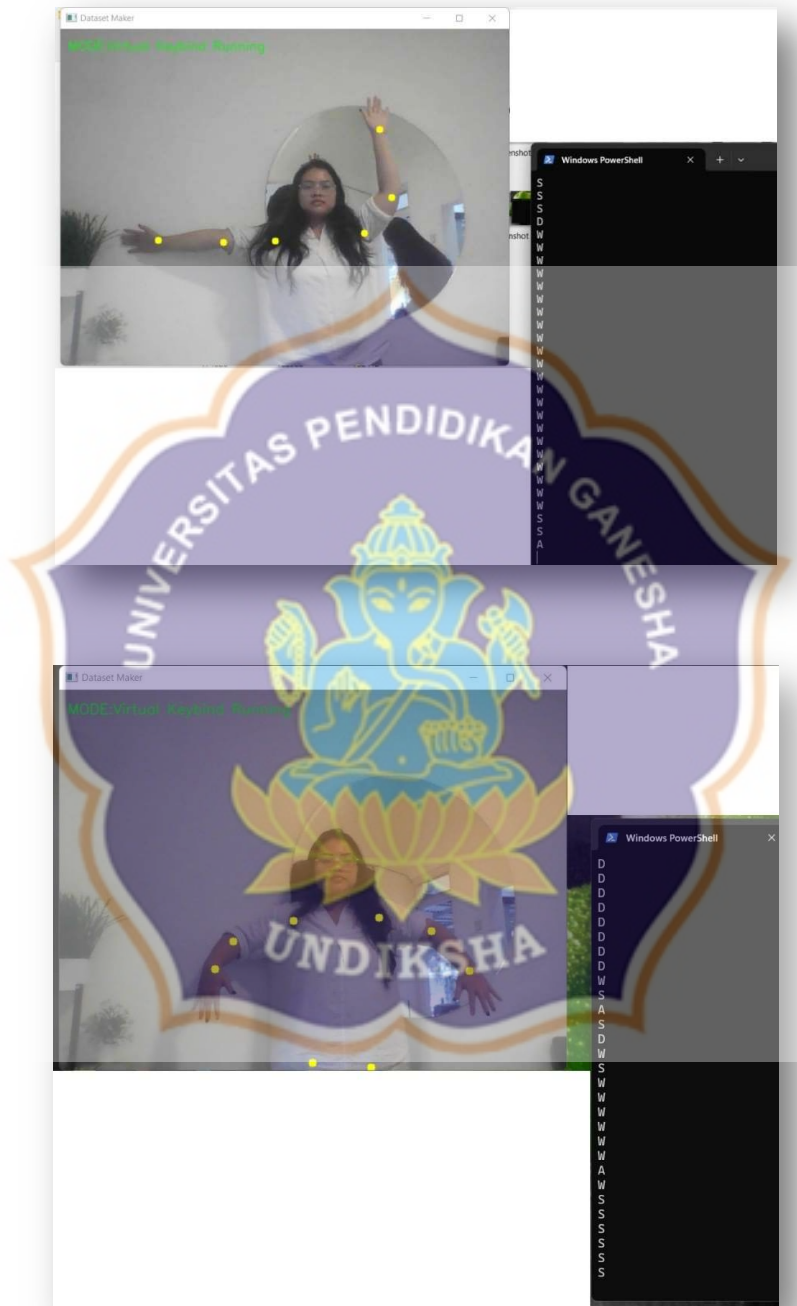
- `draw_info()` : Menampilkan teks mode (berjalan atau tidak) di atas video

```
def draw_info(image, mode):  
    mode_string = ['Virtual Keybind Not Running', 'Virtual Keybind Running']  
    if mode == 0:  
        cv2.putText(image, "MODE:" + mode_string[mode], (10, 30),  
                     cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 1,  
                     cv2.LINE_AA)  
    if mode == 1:  
        cv2.putText(image, "MODE:" + mode_string[mode], (10, 30),  
                     cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 1,  
                     cv2.LINE_AA)  
    return image
```



Lampiran 3. Print Kontrol Pada Terminal

Terlihat di terminal bahwa sistem mampu mendeteksi pose dan mengklasifikasikan sesuai dengan yang diinginkan.





Lampiran 3. Hasil Kontrol Game Dengan Pose

Sistem mampu mendeteksi dan mengklasifikasi pose serta mengaplikasikannya pada game, terlihat bahwa 'cat' memberikan aksi yang sama dengan arahan pose.

