

LAMPIRAN

Lampiran 1 Surat Keterangan Bebas Plagiat



UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN
Gedung Rektorat Lantai 3,
Jl.Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : PENERAPAN METODE RANDOM FOREST UNTUK
MEMPREDIKSI INDIKASI AUTISME PADA PESERTA DIDIK
SLB
Penulis : Duha Nur Pambudi
NIM : 2021230031
Tgl pemeriksaan : 25 Juli 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) **15%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 25 Juli 2025

Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan
dan Pasca Sarjana

Lampiran 2 Hasil Turnitin

2021230031_Duha Nur Pambudi			
ORIGINALITY REPORT			
15%	14%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	www.researchgate.net Internet Source		1%
2	journal.uinjkt.ac.id Internet Source		1%
3	journal.universitasbumigora.ac.id Internet Source		1%
4	eprints.bsi.ac.id Internet Source		1%
5	123dok.com Internet Source		1%
6	docplayer.info Internet Source		<1%
7	e-journal.hamzanwadi.ac.id Internet Source		<1%
8	digilib.stiestekom.ac.id Internet Source		<1%
9	kc.umn.ac.id Internet Source		<1%
10	eprints.perbanas.ac.id Internet Source		<1%
11	repository.dinamika.ac.id Internet Source		<1%
	studentpasca.budiluhur.ac.id		

Lampiran 3 Kode Program app.py

```
1. from flask import Flask, request, jsonify # Flask digunakan
    untuk membuat API
2. from flask_cors import CORS #  Untuk mengizinkan permintaan
    dari domain lain (CORS)
3. import joblib # Untuk memuat model dan encoder dari file .pkl
4. import pandas as pd
5. import numpy as np
6. import mysql.connector # Untuk koneksi ke database MySQL
7. import shap # Untuk interpretasi model menggunakan SHAP
8. import traceback # Untuk menampilkan detail error saat
    debugging
9. app = Flask(__name__)
10. CORS(app) #  Mengizinkan akses API dari semua domain
    (boleh dibatasi nanti agar lebih aman)
11. # Konfigurasi koneksi ke database
12. db_config = {
13.     'host': 'localhost',
14.     'user': 'root',
15.     'password': '',
16.     'database': 'db_aed'
17. }
18. # Struktur untuk menyimpan artefak model, encoder, dan SHAP
    explainer
19. model_bundle = {
20.     "model": None,
21.     "encoder_bundle": None,
22.     "explainer": None
23. }
24. # Fungsi untuk memuat model, encoder, dan explainer dari
    file
25. def load_artifacts():
26.     global model_bundle
27.     try:
28.         # Koneksi ke database untuk mendapatkan model yang aktif
29.         conn = mysql.connector.connect(**db_config)
30.         cursor = conn.cursor(dictionary=True)
31.         cursor.execute("SELECT filename_model, filename_encoder FROM
            tb_model WHERE aktif = 1 LIMIT 1")
32.         active_files = cursor.fetchone()
33.         if active_files:
34.             # Path file model dan encoder
35.             model_path = f"uploads/{active_files['filename_model']}"
36.             encoder_bundle_path =
                f"uploads/{active_files['filename_encoder']}"
37.             explainer_path = "shap_explainer.pkl"
38.             # Memuat artefak dari file .pkl
39.             model_bundle["model"] = joblib.load(model_path)
40.             model_bundle["encoder_bundle"] =
                joblib.load(encoder_bundle_path)
41.             model_bundle["explainer"] = joblib.load(explainer_path)
42.             print(" Artefak (Model, Encoder, Explainer) berhasil
                dimuat.")
43.             return True
44.         else:
```

```

45. print("✘ Peringatan: Tidak ada model aktif.")
46. return False
47. except Exception as e:
48. print(f"✘ Error saat memuat artefak: {e}")
49. return False
50. finally:
51. # Menutup koneksi database jika sudah selesai
52. if 'conn' in locals() and conn.is_connected():
53. cursor.close()
54. conn.close()
55. # Muat artefak saat server pertama kali dijalankan
56. load_artifacts()
57. # Endpoint untuk me-reload model jika admin mengganti model
   dari panel web
58. @app.route('/reload-model', methods=['POST'])
59. def reload_model():
60. if load_artifacts():
61. return jsonify({"status": "success", "message": "Artefak
   berhasil dimuat ulang."}), 200
62. else:
63. return jsonify({"status": "error", "message": "Gagal memuat
   ulang artefak."}), 500
64. # Endpoint untuk memproses prediksi autisme
65. @app.route('/predict', methods=['POST'])
66. def predict():
67. if not all(model_bundle.values()):
68. return jsonify({'error': 'Artefak tidak tersedia'}), 500
69. try:
70. data = request.get_json() # Ambil data JSON dari frontend
71. df = pd.DataFrame([data]) # Konversi ke DataFrame
72. # Ambil encoder dan fitur
73. encoder_bundle = model_bundle["encoder_bundle"]
74. encoder = encoder_bundle['encoder']
75. cat_features = encoder_bundle['categorical_features']
76. num_features = encoder_bundle['numerical_features']
77. # Pisahkan data kategorikal dan numerik
78. X_cat = df[cat_features]
79. X_num = df[num_features]
80. # Encode data kategorikal dan gabungkan dengan numerik
81. X_cat_encoded = encoder.transform(X_cat)
82. X_all = np.hstack([X_num.to_numpy(), X_cat_encoded])
83. # Lakukan prediksi menggunakan model Random Forest
84. prediction = model_bundle["model"].predict(X_all)
85. hasil = 'ASD' if prediction[0] == 1 else 'Non-ASD'
86. # Interpretasi menggunakan SHAP
87. explainer = model_bundle["explainer"]
88. encoded_feature_names =
   encoder.get_feature_names_out(cat_features)
89. full_feature_names = num_features +
   list(encoded_feature_names)
90. shap_values = explainer(X_all)
91. shap_values_for_asd_class = shap_values.values[:, :, 1] #
   Ambil SHAP untuk kelas ASD
92. shap_values_for_prediction = shap_values_for_asd_class[0]
93. # Gabungkan nama fitur dan kontribusinya

```

```

94.     contributions = dict(zip(full_feature_names,
    shap_values_for_prediction))
95.     # Kembalikan hasil prediksi dan kontribusi fitur ke frontend
96.     return jsonify({
97.         'prediksi': hasil,
98.         'kontribusi_fitur': contributions
99.     })
100. except Exception as e:
101.     # Tangani error jika terjadi saat prediksi
102.     print("\n✘ TERJADI ERROR PADA PROSES PREDIKSI ✘")
103.     traceback.print_exc()
104.     return jsonify({'error': f'Terjadi kesalahan saat prediksi:
    {str(e)}'}), 400
105. # Jalankan server Flask di port 5000
106. if __name__ == '__main__':
107.     app.run(port=5000, debug=True)

```

Lampiran 4 Kode Program form_identitas.php

```

1. <?php
2. session_start();
3. // Keamanan: Pastikan pengguna sudah login
4. if (!isset($_SESSION['user_id']) || !isset($_SESSION['role'])
    || $_SESSION['role'] !== 'user') {
5.     header("Location: ../login.php");
6.     exit;
7. }
8. require '../db/koneksi.php';
9. $sweet_alert_script = '';
10. // PROSES FORM SAAT DISUBMIT
11. if ($_SERVER["REQUEST_METHOD"] == "POST") {
12.     // Ambil semua data dari form
13.     $id_user = $_SESSION['user_id'];
14.     $nama_lengkap = trim($_POST['nama_lengkap']);
15.     $gender = $_POST['gender'] ?? null;
16.     $age = filter_input(INPUT_POST, 'age', FILTER_VALIDATE_INT);
17.     $ethnicity = $_POST['ethnicity'] ?? null;
18.     $jaundice = $_POST['jaundice'] ?? null;
19.     $austim = $_POST['austim'] ?? null;
20.     $contry_of_res = $_POST['contry_of_res'] ?? null;
21.     $relation = $_POST['relation'] ?? null;
22.     // Validasi input wajib
23.     if (empty($nama_lengkap) || empty($gender) || !$age || $age
        <= 0 || empty($ethnicity) || empty($jaundice) || empty($austim)
        || empty($contry_of_res) || empty($relation)) {
24.         $sweet_alert_script = "Swal.fire('Error!', 'Semua kolom
            wajib diisi dengan benar. Mohon periksa kembali isian Anda.',
            'error');";
25.     } else {
26.         $photo_path = null;
27.         $upload_ok = true; // Flag untuk status upload
28.         // Proses upload foto hanya jika file dipilih
29.         if (isset($_FILES['photo']) && $_FILES['photo']['error'] ==
            0) {
30.             $target_dir = "../uploads/photos/";

```

```

31.  if (!file_exists($target_dir)) {
32.  mkdir($target_dir, 0777, true);
33.  }
34.  $file_extension =
    strtolower(pathinfo($_FILES["photo"]["name"],
    PATHINFO_EXTENSION));
35.  $unique_filename = "input_" . uniqid() . '.' .
    $file_extension;
36.  $target_file = $target_dir . $unique_filename;
37.  $allowed_types = ['jpg', 'jpeg', 'png', 'gif'];
38.  if (!in_array($file_extension, $allowed_types)) {
39.  $sweet_alert_script = "Swal.fire('Error!', 'Hanya file
    gambar (JPG, JPEG, PNG, GIF) yang diizinkan.', 'error');";
40.  $upload_ok = false;
41.  }
42.  if ($upload_ok) {
43.  if (move_uploaded_file($_FILES["photo"]["tmp_name"],
    $target_file)) {
44.  $photo_path = "uploads/photos/" . $unique_filename;
45.  } else {
46.  $sweet_alert_script = "Swal.fire('Error!', 'Terjadi
    kesalahan saat mengunggah file foto.', 'error');";
47.  $upload_ok = false;
48.  }
49.  }
50.  }
51.  // Lanjutkan ke database HANYA jika tidak ada error upload
52.  if ($upload_ok) {
53.  $stmt = $conn->prepare(
54.  "INSERT INTO tb_input (id_user, nama_lengkap, gender, age,
    ethnicity, jaundice, austim, contry_of_res, relation, photo)
55.  VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
56.  );
57.  $stmt->bind_param("ississssss", $id_user, $nama_lengkap,
    $gender, $age, $ethnicity, $jaundice, $austim, $contry_of_res,
    $relation, $photo_path);
58.  if ($stmt->execute()) {
59.  $new_input_id = $conn->insert_id;
60.  $_SESSION['current_input_id'] = $new_input_id;
61.  $sweet_alert_script = "
62.  Swal.fire({
63.  icon: 'success',
64.  title: 'Data Identitas Tersimpan!',
65.  text: 'Anda akan diarahkan ke halaman kuesioner.',
66.  showConfirmButton: false,
67.  timer: 2000
68.  }).then(() => {
69.  window.location.href = 'kuisisioner.php';
70.  });
71.  ";
72.  } else {
73.  $sweet_alert_script = "Swal.fire('Error!', 'Gagal menyimpan
    data ke database: " . $conn->error . "', 'error');";
74.  }
75.  $stmt->close();
76.  }

```

```

77.     }
78.     $conn->close();
79.     }
80.     ?>
81.     <!DOCTYPE html>
82.     <html lang="id">
83.     <head>
84.     <meta charset="UTF-8">
85.     <meta name="viewport" content="width=device-width, initial-
      scale=1.0">
86.     <title>Langkah 1: Identitas Peserta - Autism Early
      Detector</title>
87.     <link rel="icon" href="../../../images/aed.png" type="image/png">
88.     <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/boo
      tstrap.min.css" rel="stylesheet">
89.     <link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap-
      icons@1.11.3/font/bootstrap-icons.min.css">
90.     <link rel="stylesheet" href="../../../css/styledash.css">
91.     <script
      src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
92.     </head>
93.     <body>
94.     <?php include '../includes/navbar.php'; ?>
95.     <main>
96.     <div class="container my-5">
97.     <div class="row justify-content-center">
98.     <div class="col-lg-9">
99.     <div class="card shadow-sm">
100.    <div class="card-header bg-primary text-white">
101.    <h4 class="mb-0"><i class="bi bi-person-vcard me-
      2"></i>Langkah 1: Isi Identitas Peserta Tes</h4>
102.    </div>
103.    <div class="card-body p-4">
104.    <p class="card-text text-muted mb-4">Mohon isi data pada
      form di bawah ini sesuai dengan pertanyaan dan pilihan yang
      disediakan.</p>
105.    <form method="POST" action="form_identitas.php"
      enctype="multipart/form-data">
106.    <div class="mb-3">
107.    <label for="nama_lengkap" class="form-label">Nama Lengkap
      Peserta</label>
108.    <input type="text" class="form-control" id="nama_lengkap"
      name="nama_lengkap" required>
109.    </div>
110.    <div class="row">
111.    <div class="col-md-6 mb-3">
112.    <label for="age" class="form-label">Umur (Tahun)</label>
113.    <input type="number" class="form-control" id="age"
      name="age" min="1" required>
114.    </div>
115.    <div class="col-md-6 mb-3">
116.    <label class="form-label">Jenis Kelamin</label>
117.    <div>
118.    <div class="form-check form-check-inline">

```

```

119. <input class="form-check-input" type="radio" name="gender"
      id="male" value="Male" required>
120. <label class="form-check-label" for="male">Laki-laki</label>
121. </div>
122. <div class="form-check form-check-inline">
123. <input class="form-check-input" type="radio" name="gender"
      id="female" value="Female">
124. <label class="form-check-label"
      for="female">Perempuan</label>
125. </div>
126. </div>
127. </div>
128. </div>
129. <div class="row">
130. <div class="col-md-6 mb-3">
131. <label for="ethnicity" class="form-label">Etnis/Suku</label>
132. <select class="form-select" id="ethnicity" name="ethnicity"
      required>
133. <option value="Asian" selected>Asian</option>
134. <option value="White-European">White-European</option>
135. <option value="Middle Eastern">Middle Eastern</option>
136. <option value="South Asian">South Asian</option>
137. <option value="Black">Black</option>
138. <option value="Hispanic">Hispanic</option>
139. <option value="Latino">Latino</option>
140. <option value="Pasifika">Pasifika</option>
141. <option value="Turkish">Turkish</option>
142. <option value="others">others</option>
143. <option value="Others">Others</option>
144. <option value="Unknown">Unknown</option>
145. </select>
146. </div>
147. <div class="col-md-6 mb-3">
148. <label for="contry_of_res" class="form-label">Negara Tempat
      Tinggal</label>
149. <select class="form-select" id="contry_of_res"
      name="contry_of_res" required>
150. <option value="Indonesia" selected>Indonesia</option>
151. <option value="Afghanistan">Afghanistan</option>
152. <option value="AmericanSamoa">American Samoa</option>
153. <option value="Angola">Angola</option>
154. <option value="Argentina">Argentina</option>
155. <option value="Armenia">Armenia</option>
156. <option value="Aruba">Aruba</option>
157. <option value="Australia">Australia</option>
158. <option value="Austria">Austria</option>
159. <option value="Azerbaijan">Azerbaijan</option>
160. <option value="Bahamas">Bahamas</option>
161. <option value="Bangladesh">Bangladesh</option>
162. <option value="Belgium">Belgium</option>
163. <option value="Bolivia">Bolivia</option>
164. <option value="Brazil">Brazil</option>
165. <option value="Burundi">Burundi</option>
166. <option value="Canada">Canada</option>
167. <option value="China">China</option>
168. <option value="Cyprus">Cyprus</option>

```

```

169. <option value="Czech Republic">Czech Republic</option>
170. <option value="Egypt">Egypt</option>
171. <option value="Ethiopia">Ethiopia</option>
172. <option value="France">France</option>
173. <option value="Germany">Germany</option>
174. <option value="Hong Kong">Hong Kong</option>
175. <option value="Iceland">Iceland</option>
176. <option value="India">India</option>
177. <option value="Iran">Iran</option>
178. <option value="Iraq">Iraq</option>
179. <option value="Ireland">Ireland</option>
180. <option value="Italy">Italy</option>
181. <option value="Japan">Japan</option>
182. <option value="Jordan">Jordan</option>
183. <option value="Kazakhstan">Kazakhstan</option>
184. <option value="Malaysia">Malaysia</option>
185. <option value="Mexico">Mexico</option>
186. <option value="Netherlands">Netherlands</option>
187. <option value="New Zealand">New Zealand</option>
188. <option value="Nicaragua">Nicaragua</option>
189. <option value="Niger">Niger</option>
190. <option value="Oman">Oman</option>
191. <option value="Pakistan">Pakistan</option>
192. <option value="Romania">Romania</option>
193. <option value="Russia">Russia</option>
194. <option value="Saudi Arabia">Saudi Arabia</option>
195. <option value="Serbia">Serbia</option>
196. <option value="Sierra Leone">Sierra Leone</option>
197. <option value="South Africa">South Africa</option>
198. <option value="Spain">Spain</option>
199. <option value="Sri Lanka">Sri Lanka</option>
200. <option value="Sweden">Sweden</option>
201. <option value="Tonga">Tonga</option>
202. <option value="Ukraine">Ukraine</option>
203. <option value="United Arab Emirates">United Arab
    Emirates</option>
204. <option value="United Kingdom">United Kingdom</option>
205. <option value="United States">United States</option>
206. <option value="Viet Nam">Viet Nam</option>
207. </select>
208. </div>
209. </div>
210. <div class="row">
211. <div class="col-md-6 mb-3">
212. <label class="form-label">Apakah peserta pernah mengalami
    sakit kuning saat lahir?</label>
213. <div>
214. <div class="form-check form-check-inline"><input
    class="form-check-input" type="radio" name="jaundice"
    id="jaundice_yes" value="yes" required><label class="form-
    check-label" for="jaundice_yes">Ya</label></div>
215. <div class="form-check form-check-inline"><input
    class="form-check-input" type="radio" name="jaundice"
    id="jaundice_no" value="no"><label class="form-check-label"
    for="jaundice_no">Tidak</label></div>
216. </div>

```

```

217. </div>
218. <div class="col-md-6 mb-3">
219. <label class="form-label">Apakah ada riwayat autisme dalam
    keluarga dekat?</label>
220. <div>
221. <div class="form-check form-check-inline"><input
    class="form-check-input" type="radio" name="austim"
    id="family_yes" value="yes" required><label class="form-check-
    label" for="family_yes">Ya</label></div>
222. <div class="form-check form-check-inline"><input
    class="form-check-input" type="radio" name="austim"
    id="family_no" value="no"><label class="form-check-label"
    for="family_no">Tidak</label></div>
223. </div>
224. </div>
225. </div>
226. <div class="mb-3">
227. <label for="relation" class="form-label">Anda mengisi
    kuesioner ini sebagai?</label>
228. <select class="form-select" id="relation" name="relation"
    required>
229. <option value="">-- Pilih Hubungan --</option>
230. <option value="Parent">Orang Tua</option>
231. <option value="Self">Diri Sendiri</option>
232. <option value="Relative">Kerabat</option>
233. <option value="Health care professional">Tenaga Kesehatan
    Profesional</option>
234. <option value="Others">Lainnya</option>
235. <option value="">Tidak Diketahui</option>
236. </select>
237. </div>
238. <div class="mb-4">
239. <label for="photo" class="form-label">Upload Foto Peserta
    (Opsional)</label>
240. <input class="form-control" type="file" id="photo"
    name="photo" accept="image/*">
241. </div>
242. <div class="d-grid">
243. <button type="submit" class="btn btn-primary btn-lg">
244. Simpan dan Lanjutkan ke Kuesioner <i class="bi bi-arrow-
    right-circle-fill"></i>
245. </button>
246. </div>
247. </form>
248. </div>
249. </div>
250. </div>
251. </div>
252. </div>
253. </main>
254. <footer class="footer text-center">
255. <div class="container">
256. <span>&copy; <?= date('Y') ?> - Duha Nur Pambudi
    (2021230031) - Universitas Darma Persada</span>
257. </div>
258. </footer>

```

```

259. <?php if (!empty($sweet_alert_script)): ?>
260. <script>
261. document.addEventListener('DOMContentLoaded', function() {
262. <?php echo $sweet_alert_script; ?>
263. });
264. </script>
265. <?php endif; ?>
266. <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootst
    trap.bundle.min.js"></script>
267. </body>
268. </html>

```

Lampiran 5 Kode Program Kuisisioner.php

```

1. <?php
2. session_start();
3. // Keamanan: Pastikan pengguna sudah login DAN sudah melalui
   form_identitas.php
4. if (!isset($_SESSION['user_id']) ||
    !isset($_SESSION['current_input_id'])) {
5. header("Location: form_identitas.php");
6. exit;
7. }
8. require '../db/koneksi.php';
9. $id_input = $_SESSION['current_input_id'];
10. $sweet_alert_script = '';
11. // Kunci skoring final & akurat berdasarkan pertanyaan
    standar AQ-10
12. // Ini adalah ID pertanyaan di mana jawaban "Tidak" / "Tidak
    Setuju" mendapat skor 1
13. $reverse_scored_ids = [3, 6, 9, 10];
14. if ($_SERVER["REQUEST_METHOD"] == "POST") {
15. $answers = $_POST['answers'] ?? [];
16. if (count($answers) < 10) {
17. $sweet_alert_script = "Swal.fire('Error!', 'Harap jawab
    semua 10 pertanyaan.', 'error');";
18. } else {
19. $conn->begin_transaction();
20. try {
21. // Hapus jawaban lama jika ada (untuk kasus user back dan
    submit ulang)
22. $stmt_delete = $conn->prepare("DELETE FROM tb_answer WHERE
    id_input = ?");
23. $stmt_delete->bind_param("i", $id_input);
24. $stmt_delete->execute();
25. $stmt_delete->close();
26. $stmt_insert = $conn->prepare("INSERT INTO tb_answer
    (id_input, id_question, jawaban, skor) VALUES (?, ?, ?, ?)");
27. $total_score = 0;
28. $scores_for_model = [];
29. // Ambil semua kode soal dari DB untuk mapping
30. $all_questions = [];
31. $result_q = $conn->query("SELECT id_question, kode_soal FROM
    tb_question ORDER BY id_question ASC");

```

```

32. while($row_q = $result_q->fetch_assoc()) {
33.     $all_questions[$row_q['id_question']] = $row_q['kode_soal'];
34. }
35. foreach ($answers as $id_question => $jawaban) {
36.     $skor = 0;
37.     // Logika skoring
38.     if (in_array($id_question, $reverse_scored_ids)) {
39.         $skor = ($jawaban === 'Tidak') ? 1 : 0;
40.     } else {
41.         $skor = ($jawaban === 'Ya') ? 1 : 0;
42.     }
43.     $total_score += $skor;
44.     $kode_soal = $all_questions[$id_question];
45.     $scores_for_model[$kode_soal] = $skor;
46.     // Simpan jawaban ke tb_answer
47.     $stmt_insert->bind_param("iisi", $id_input, $id_question,
    $jawaban, $skor);
48.     $stmt_insert->execute();
49. }
50. $stmt_insert->close();
51. // Ambil data demografis dari tb_input untuk dikirim ke API
52. $stmt_input = $conn->prepare("SELECT age, gender, ethnicity,
    jaundice, austim, contry_of_res, relation FROM tb_input WHERE
    id_input = ?");
53. $stmt_input->bind_param("i", $id_input);
54. $stmt_input->execute();
55. $input_data = $stmt_input->get_result()->fetch_assoc();
56. $stmt_input->close();
57. // Gabungkan semua data fitur untuk dikirim ke API
58. $data_for_api = array_merge($input_data, $scores_for_model);
59. // Panggil API Python
60. $ch = curl_init('http://127.0.0.1:5000/predict');
61. curl_setopt_array($ch, [
62.     CURLOPT_RETURNTRANSFER => true,
63.     CURLOPT_POST => true,
64.     CURLOPT_POSTFIELDS => json_encode($data_for_api),
65.     CURLOPT_HTTPHEADER => ['Content-Type: application/json'],
66.     CURLOPT_TIMEOUT => 30,
67.     CURLOPT_SSL_VERIFYPEER => false,
68.     CURLOPT_SSL_VERIFYHOST => false
69. ]);
70. $api_response = curl_exec($ch);
71. $httpcode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
72. $error_curl = curl_error($ch);
73. curl_close($ch);
74. // Proses hasil dari API
75. if ($httpcode == 200 && $api_response) {
76.     $response_data = json_decode($api_response, true);
77.     if (isset($response_data['error'])) {
78.         throw new Exception("API Error: " .
    $response_data['error']);
79.     }
80.     $hasil_prediksi = $response_data['prediksi'] ?? 'Error';
81.     $feature_contributions =
    json_encode($response_data['kontribusi_fitur'] ?? []);
82. } else {

```

```

83. // Fallback jika API gagal
84. $hasil_prediksi = ($total_score > 6) ? 'ASD' : 'Non-ASD';
85. $feature_contributions = null;
86. if ($error_curl) {
87.     $message = "Gagal menghubungi server prediksi: " .
        $error_curl;
88.     throw new Exception($message);
89. }
90. }
91. $keterangan = "Total skor dari kuesioner adalah " .
    $total_score . ".";
92. // Simpan hasil akhir, termasuk data kontribusi SHAP
93. $stmt_result = $conn->prepare("INSERT INTO tb_result
    (id_input, total_score, hasil_prediksi, keterangan,
    feature_contributions) VALUES (?, ?, ?, ?, ?)");
94. $stmt_result->bind_param("issss", $id_input, $total_score,
    $hasil_prediksi, $keterangan, $feature_contributions);
95. $stmt_result->execute();
96. $stmt_result->close();
97. $conn->commit();
98. unset($_SESSION['current_input_id']);
99. // Redirect ke halaman hasil
100. header("Location: hasil_prediksi.php?id_input=" .
    $id_input);
101. exit;
102. } catch (Exception $e) {
103.     $conn->rollback();
104.     $sweet_alert_script = "Swal.fire('Error!', 'Terjadi
        kesalahan sistem: " . addslashes($e->getMessage()) . "',
        'error');";
105. }
106. }
107. }
108. // Ambil pertanyaan untuk ditampilkan di form
109. $questions = [];
110. $result = $conn->query("SELECT id_question, isi_pertanyaan
    FROM tb_question ORDER BY id_question ASC");
111. if ($result && $result->num_rows >= 10) {
112.     while ($row = $result->fetch_assoc()) {
113.         $questions[] = $row;
114.     }
115. } else {
116.     die("Error: Konfigurasi pertanyaan di database tidak
        lengkap. Pastikan Anda sudah menjalankan skrip SQL yang
        benar.");
117. }
118. $conn->close();
119. ??
120. <!DOCTYPE html>
121. <html lang="id">
122. <head>
123. <meta charset="UTF-8">
124. <meta name="viewport" content="width=device-width, initial-
    scale=1.0">
125. <title>Langkah 2: Kuesioner AQ-10 - Autism Early
    Detector</title>

```

```

126. <link rel="icon" href="../../../images/aed.png" type="image/png">
127. <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
128. <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
129. <link rel="stylesheet" href="../../../css/styledash.css">
130. <script
    src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
131. </head>
132. <body>
133. <?php include '../includes/navbar.php'; ?>
134. <main>
135. <div class="container my-5">
136. <div class="row justify-content-center">
137. <div class="col-lg-9">
138. <div class="card shadow-sm">
139. <div class="card-header bg-primary text-white"><h4
    class="mb-0"><i class="bi bi-pencil-square me-2"></i>Langkah 2:
    Jawab Kuesioner</h4></div>
140. <div class="card-body p-4">
141. <p class="card-text text-muted mb-4">Jawablah 10 pertanyaan
    berikut dengan memilih "Ya" (jika Anda setuju) atau "Tidak"
    (jika tidak setuju).</p>
142. <form method="POST" action="kuisisioner.php">
143. <?php foreach ($questions as $index => $q): ?>
144. <div class="mb-4 p-3 border rounded bg-light">
145. <p class="fw-bold mb-2"><?=( $index + 1 ) . ". " .
    htmlspecialchars($q['isi_pertanyaan']) ?></p>
146. <div>
147. <div class="form-check form-check-inline">
148. <input class="form-check-input" type="radio"
    name="answers[<?=$q['id_question'] ?>]" id="q<?=$q['id_question'] ?>_yes" value="Ya" required>
149. <label class="form-check-label" for="q<?=$q['id_question']
    ?>_yes">Ya / Setuju</label>
150. </div>
151. <div class="form-check form-check-inline">
152. <input class="form-check-input" type="radio"
    name="answers[<?=$q['id_question'] ?>]" id="q<?=$q['id_question'] ?>_no" value="Tidak">
153. <label class="form-check-label" for="q<?=$q['id_question']
    ?>_no">Tidak / Tidak Setuju</label>
154. </div>
155. </div>
156. </div>
157. <?php endforeach; ?>
158. <div class="d-grid mt-4"><button type="submit" class="btn
    btn-success btn-lg">Selesai & Lihat Hasil <i class="bi bi-check-circle-fill"></i></button></div>
159. </form>
160. </div>
161. </div>
162. </div>
163. </div>

```

```
164. </div>
165. </main>
166. <footer class="footer text-center">
167. <div class="container">
168. <span>&copy; <?= date('Y') ?> - Duha Nur Pambudi
    (2021230031) - Universitas Darma Persada</span>
169. </div>
170. </footer>
171. <?php if (!empty($sweet_alert_script)): ?>
172. <script>
173. document.addEventListener('DOMContentLoaded', function() {
    <?php echo $sweet_alert_script; ?> });
174. </script>
175. <?php endif; ?>
176. <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootst
    rap.bundle.min.js"></script>
177. </body>
178. </html>
```

