<?php

// voting.php - Halaman Proses Voting (Integrasi API Python Verifikasi + Debug Prepare)

require\_once 'config.php'; // Memuat konfigurasi & memulai session

// --- 1. Validasi Akses, Status Vote, Jadwal ---

if (!is\_login() || !isset($\_SESSION['user']['role']) || $\_SESSION['user']['role'] !== 'pemilih') {

$\_SESSION['flash\_error\_login'] = "Anda harus login sebagai pemilih untuk voting.";

header('Location: login.php'); exit;

}

$user\_id\_pemilih = $\_SESSION['user']['id'];

$status\_sudah\_vote\_session = isset($\_SESSION['user']['sudah\_vote']) ? (int)$\_SESSION['user']['sudah\_vote'] : 0;

$nama\_file\_foto\_pendaftaran\_user = null; // Akan diisi dari DB

// Cek status vote dari DB & ambil nama file foto pendaftaran

$user\_sudah\_vote\_dari\_db = false;

if ($conn) {

// Ambil status vote dan foto pendaftaran dalam satu query

$stmt\_get\_user\_data = $conn->prepare("SELECT sudah\_vote, foto\_wajah\_pendaftaran FROM users WHERE id = ? AND role = 'pemilih'");

if ($stmt\_get\_user\_data) {

$stmt\_get\_user\_data->bind\_param("i", $user\_id\_pemilih);

$stmt\_get\_user\_data->execute();

$stmt\_get\_user\_data->bind\_result($db\_sudah\_vote, $db\_foto\_pendaftaran);

if ($stmt\_get\_user\_data->fetch()) {

$status\_sudah\_vote\_session = (int)$db\_sudah\_vote;

$\_SESSION['user']['sudah\_vote'] = $status\_sudah\_vote\_session; // Update session

$nama\_file\_foto\_pendaftaran\_user = $db\_foto\_pendaftaran; // Ambil nama file

if ($status\_sudah\_vote\_session > 0) {

$user\_sudah\_vote\_dari\_db = true;

}

} else {

error\_log("Voting: User ID {$user\_id\_pemilih} tidak ditemukan / bukan pemilih.");

$\_SESSION['flash\_error\_login'] = "Data pengguna tidak valid.";

header('Location: login.php'); exit;

}

$stmt\_get\_user\_data->close();

} else {

error\_log("Voting: Gagal prepare get user data - " . $conn->error);

die("Terjadi kesalahan saat memverifikasi data pengguna.");

}

} else { die("Koneksi database bermasalah."); }

// Redirect jika sudah vote

if ($user\_sudah\_vote\_dari\_db) {

$\_SESSION['flash\_info\_dashboard'] = "Anda sudah menggunakan hak suara Anda.";

header('Location: dashboard.php'); exit;

}

// Redirect jika tidak punya foto pendaftaran

if (empty($nama\_file\_foto\_pendaftaran\_user)) {

error\_log("Voting: User ID {$user\_id\_pemilih} tidak memiliki foto pendaftaran.");

$\_SESSION['flash\_error\_dashboard'] = "Data foto pendaftaran Anda tidak ditemukan. Hubungi administrator.";

header('Location: dashboard.php'); exit;

}

// Cek Jadwal Voting Aktif

$jadwal\_voting\_aktif = false;

$pesan\_jadwal\_voting = "Informasi jadwal voting tidak tersedia atau belum diatur.";

$evoting\_start\_str\_db = get\_setting('evoting\_start', $conn);

$evoting\_end\_str\_db = get\_setting('evoting\_end', $conn);

if (!empty($evoting\_start\_str\_db) && !empty($evoting\_end\_str\_db)) {

try {

$tz = new DateTimeZone('Asia/Jakarta');

$start = new DateTime($evoting\_start\_str\_db, $tz); $end = new DateTime($evoting\_end\_str\_db, $tz); $now = new DateTime('now', $tz);

if ($now >= $start && $now <= $end) $jadwal\_voting\_aktif = true;

elseif ($now < $start) $pesan\_jadwal\_voting = "Periode voting belum dimulai (Mulai: " . $start->format('d/m/Y H:i') . ").";

else $pesan\_jadwal\_voting = "Periode voting telah berakhir (Selesai: " . $end->format('d/m/Y H:i') . ").";

} catch (Exception $e) { error\_log("Voting: Error parsing jadwal - ".$e->getMessage()); $pesan\_jadwal\_voting = "Error format jadwal."; }

}

if (!$jadwal\_voting\_aktif) {

$\_SESSION['flash\_error\_dashboard'] = $pesan\_jadwal\_voting;

header('Location: dashboard.php'); exit;

}

// Ambil Data Kandidat

$daftar\_kandidat\_voting = [];

if ($conn) {

$stmt\_get\_kdt = $conn->prepare("SELECT id, nama, jurusan, foto, visi\_misi FROM kandidat ORDER BY nama ASC");

if ($stmt\_get\_kdt) { /\* ... (ambil data kandidat seperti sebelumnya) ... \*/

$stmt\_get\_kdt->execute(); $result\_kdt = $stmt\_get\_kdt->get\_result();

while ($kdt\_row = $result\_kdt->fetch\_assoc()) { $daftar\_kandidat\_voting[] = $kdt\_row; }

$stmt\_get\_kdt->close();

} else { die("Gagal memuat data kandidat."); }

}

// --- 4. Proses Submit Voting ---

$error\_submit\_vote = '';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['submit\_vote\_action'])) {

if (!validate\_csrf\_token()) {

$error\_submit\_vote = "Sesi tidak valid. Ulangi proses voting.";

} else {

$id\_kandidat\_dipilih = filter\_input(INPUT\_POST, 'kandidat\_id\_final', FILTER\_VALIDATE\_INT);

$foto\_wajah\_lolos\_verifikasi\_base64 = $\_POST['foto\_wajah\_saat\_vote\_data'] ?? '';

if (!$id\_kandidat\_dipilih || $id\_kandidat\_dipilih <= 0) {

$error\_submit\_vote = "Pilihan kandidat tidak valid.";

} elseif (empty($foto\_wajah\_lolos\_verifikasi\_base64)) {

$error\_submit\_vote = "Data verifikasi foto wajah tidak lengkap.";

} else {

// Cek ulang status vote

$stmt\_cek\_final = $conn->prepare("SELECT COUNT(\*) FROM voting WHERE user\_id = ?");

$stmt\_cek\_final->bind\_param("i", $user\_id\_pemilih); $stmt\_cek\_final->execute();

$stmt\_cek\_final->bind\_result($count\_final); $stmt\_cek\_final->fetch(); $stmt\_cek\_final->close();

if ($count\_final > 0) {

$\_SESSION['user']['sudah\_vote'] = 1;

$\_SESSION['flash\_info\_dashboard'] = "Anda sudah tercatat voting.";

header('Location: dashboard.php'); exit;

}

// Simpan foto wajah saat voting

$nama\_file\_foto\_saat\_vote\_db = null;

$path\_simpan\_foto\_saat\_vote = null;

if (!is\_writable(UPLOAD\_FOTO\_SAAT\_VOTE\_DIR)) {

$error\_submit\_vote = "Error: Server tidak bisa menulis file foto voting.";

error\_log("Voting Submit: Direktori UPLOAD\_FOTO\_SAAT\_VOTE\_DIR tidak writable.");

} else {

if (preg\_match('/^data:image\/(png|jpeg|jpg);base64,/', $foto\_wajah\_lolos\_verifikasi\_base64, $type\_m)) {

$ext = $type\_m[1]; $data = substr($foto\_wajah\_lolos\_verifikasi\_base64, strpos($foto\_wajah\_lolos\_verifikasi\_base64, ',') + 1);

$data = base64\_decode($data);

if ($data === false) { $error\_submit\_vote = "Error: Gagal decode foto voting."; }

else {

$nama\_file\_foto\_saat\_vote\_db = 'vote\_wajah\_' . $user\_id\_pemilih . '\_' . uniqid() . '.' . $ext;

$path\_simpan\_foto\_saat\_vote = UPLOAD\_FOTO\_SAAT\_VOTE\_DIR . $nama\_file\_foto\_saat\_vote\_db;

if (file\_put\_contents($path\_simpan\_foto\_saat\_vote, $data) === false) {

$error\_submit\_vote = "Error: Gagal simpan file foto voting."; $nama\_file\_foto\_saat\_vote\_db = null;

}

}

} else { $error\_submit\_vote = "Error: Format data foto voting tidak valid."; }

}

// Lanjutkan ke DB jika tidak ada error sebelumnya DAN foto berhasil disimpan

if (empty($error\_submit\_vote) && $nama\_file\_foto\_saat\_vote\_db !== null) {

$conn->begin\_transaction();

try {

// 1. Insert ke tabel voting

$stmt\_ins = $conn->prepare("INSERT INTO voting (user\_id, kandidat\_id, foto\_saat\_vote) VALUES (?, ?, ?)");

// Tambahkan pengecekan prepare insert vote

if ($stmt\_ins === false) {

throw new Exception("MySQL Prepare Error (INSERT voting): " . $conn->error);

}

$stmt\_ins->bind\_param("iis", $user\_id\_pemilih, $id\_kandidat\_dipilih, $nama\_file\_foto\_saat\_vote\_db);

if (!$stmt\_ins->execute()) throw new Exception("Gagal execute insert data voting: " . $stmt\_ins->error);

$stmt\_ins->close();

// 2. Update status 'sudah\_vote' di tabel 'users'

$stmt\_upd\_u = $conn->prepare("UPDATE users SET sudah\_vote = 1 WHERE id = ?");

// === PENGECEKAN EKSPLISIT SETELAH PREPARE ===

if ($stmt\_upd\_u === false) {

// Jika prepare GAGAL, lempar exception dengan error MySQL

throw new Exception("MySQL Prepare Error (UPDATE users): " . $conn->error);

}

// ===========================================

// Jika prepare berhasil, baru bind\_param (ini baris ~136 yang error sebelumnya)

$stmt\_upd\_u->bind\_param("i", $user\_id\_pemilih);

if (!$stmt\_upd\_u->execute()) throw new Exception("Gagal execute update status user: " . $stmt\_upd\_u->error);

$stmt\_upd\_u->close();

// 3. Update jumlah suara di tabel 'kandidat'

$stmt\_upd\_k = $conn->prepare("UPDATE kandidat SET jumlah\_suara = jumlah\_suara + 1 WHERE id = ?");

// Tambahkan pengecekan prepare update kandidat

if ($stmt\_upd\_k === false) {

throw new Exception("MySQL Prepare Error (UPDATE kandidat): " . $conn->error);

}

$stmt\_upd\_k->bind\_param("i", $id\_kandidat\_dipilih);

if (!$stmt\_upd\_k->execute()) throw new Exception("Gagal execute update suara kandidat: " . $stmt\_upd\_k->error);

$stmt\_upd\_k->close();

$conn->commit(); // Commit jika semua berhasil

$\_SESSION['user']['sudah\_vote'] = 1;

$\_SESSION['flash\_success\_dashboard'] = "Terima kasih! Suara Anda telah berhasil direkam.";

header('Location: dashboard.php'); exit;

} catch (Exception $e) {

$conn->rollback(); // Rollback jika ada error di try block

// Simpan pesan error spesifik dari exception untuk ditampilkan

$error\_submit\_vote = "Terjadi kesalahan teknis: " . htmlspecialchars($e->getMessage());

error\_log("Voting Submit Transaksi Gagal: " . $e->getMessage()); // Log error lengkap

// Hapus file foto yang mungkin sempat tersimpan jika transaksi DB gagal

if ($path\_simpan\_foto\_saat\_vote && file\_exists($path\_simpan\_foto\_saat\_vote)) {

@unlink($path\_simpan\_foto\_saat\_vote);

error\_log("Voting Submit: File foto '{$nama\_file\_foto\_saat\_vote\_db}' dihapus karena transaksi DB gagal.");

}

} // Akhir blok try-catch

} // Akhir if ($error\_submit\_vote empty && file foto tersimpan)

} // Akhir else (cek sudah vote final)

} // Akhir validasi input dasar & foto

} // Akhir POST check

// Definisikan URL API Python untuk JavaScript

define('PYTHON\_API\_URL\_VOTING', 'http://localhost:5000/api');

// Muat Template Header

require 'template/header.php';

?>

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet">

<style>

/\* ... (CSS dari versi sebelumnya diletakkan di sini) ... \*/

body { font-family: 'Inter', Arial, sans-serif; background-color: #f0f2f5; }

.voting-main-container { max-width: 1000px; margin: 1.5rem auto; padding: 0 1rem; }

.voting-page-title { text-align: center; font-weight: 700; color: var(--primary-color, #4f46e5); margin-bottom: 2rem; font-size: 2rem; }

.face-verification-step { background-color: #ffffff; padding: 2rem; border-radius: 12px; box-shadow: 0 6px 20px rgba(0,0,0,0.07); margin-bottom: 2.5rem; text-align: center; }

.face-verification-step h4 { font-weight: 600; color: #333; margin-bottom: 0.75rem; }

.face-verification-step p.instruction { color: #555; margin-bottom: 1.5rem; font-size: 0.95rem; }

#cameraPreviewVotingEl { width: 100%; max-width: 320px; height: auto; aspect-ratio: 4 / 3; border-radius: 8px; background-color: #333; margin: 0 auto 1rem auto; display: block; border: 3px solid #e0e0e0; }

#captureCanvasVotingEl { display: none; }

#photoStatusVotingEl { font-style: italic; color: #6c757d; margin-top: 0.75rem; min-height: 22px; font-size: 0.9rem;}

#verificationApiErrorEl { font-weight: bold; color: #dc3545; } /\* Pesan error API \*/

#verificationSpinner { display: none; margin-right: 5px;} /\* Spinner loading \*/

.snapshot-preview-voting img { max-width: 120px; max-height: 90px; border:1px solid #ccc; border-radius:5px; margin-top:0.5rem;}

.candidate-selection-step { display: none; }

.kandidat-grid-container { display: grid; grid-template-columns: repeat(auto-fit, minmax(280px, 1fr)); gap: 1.8rem; }

.kandidat-card-vote { background: #ffffff; border-radius: 12px; box-shadow: 0 5px 15px rgba(0,0,0,0.06); padding: 1.5rem; text-align: center; display: flex; flex-direction: column; transition: transform 0.2s ease, box-shadow 0.2s ease; }

.kandidat-card-vote:hover { transform: translateY(-6px); box-shadow: 0 10px 25px rgba(0,0,0,0.1); }

.kandidat-card-vote .kandidat-photo-vote { width: 100px; height: 100px; object-fit: cover; border-radius: 50%; margin: 0 auto 1rem auto; border: 4px solid var(--primary-color, #4f46e5); padding: 3px; background-color: #e9ecef; }

.kandidat-card-vote .kandidat-name-vote { font-size: 1.2rem; font-weight: 600; color: #212529; margin-bottom: 0.25rem; }

.kandidat-card-vote .kandidat-jurusan-vote { color: #6c757d; font-size: 0.9rem; margin-bottom: 0.75rem; }

.kandidat-card-vote .kandidat-visimisi-vote { font-size: 0.85rem; color: #495057; margin-bottom: 1rem; text-align: left; background-color: #f8f9fa; border-radius: 8px; padding: 0.75rem; white-space: pre-line; overflow-y: auto; max-height: 100px; flex-grow: 1; border: 1px solid #e9ecef; }

.kandidat-card-vote .btn-pilih-kandidat { font-size: 1rem; padding: 0.6rem 1.2rem; border-radius: 50px; font-weight: 600; margin-top: auto; width: 90%; margin-left: auto; margin-right: auto; }

.modal-vote-confirm { display: none; position: fixed; z-index: 2050; left: 0; top: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.55); backdrop-filter: blur(4px); align-items: center; justify-content: center; opacity: 0; transition: opacity 0.25s ease; }

.modal-vote-confirm.show { display: flex; opacity: 1; }

.modal-content-vote-confirm { background: #fff; border-radius: 15px; padding: 2rem; box-shadow: 0 8px 25px rgba(0,0,0,0.2); text-align: center; max-width: 420px; width: 90%; margin: 1rem; transform: scale(0.95); transition: transform 0.25s ease; }

.modal-vote-confirm.show .modal-content-vote-confirm { transform: scale(1); }

.modal-content-vote-confirm .icon-confirm { font-size: 2.8rem; color: var(--primary-color); margin-bottom: 1rem; }

.modal-content-vote-confirm h5 { font-weight: 600; margin-bottom: 0.75rem; }

.modal-content-vote-confirm #modalTextConfirmVote { margin-bottom: 1.5rem; color: #495057; font-size: 1.05rem; }

.modal-content-vote-confirm .btn { min-width: 120px; margin: 0 0.5rem; border-radius: 50px; padding: 0.7rem 1.3rem; font-weight: 600; }

.alert-voting-page { margin-bottom: 1.5rem; }

</style>

<div class="voting-main-container">

<div class="voting-page-title"><i class="bi bi-pencil-square"></i> Halaman Pemilihan Suara</div>

<?php if (!empty($error\_submit\_vote)): ?>

<div class="alert alert-danger alert-dismissible fade show alert-voting-page" role="alert">

<i class="bi bi-exclamation-octagon-fill me-2"></i> Voting Gagal: <?= htmlspecialchars($error\_submit\_vote) ?>

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

</div>

<?php endif; ?>

<div class="face-verification-step" id="faceVerificationStepContainer">

<h4><i class="bi bi-shield-check"></i> Verifikasi Wajah untuk Voting</h4>

<p class="instruction">Ambil foto wajah Anda. Sistem akan membandingkannya dengan foto pendaftaran.</p>

<video id="cameraPreviewVotingEl" autoplay muted playsinline></video>

<canvas id="captureCanvasVotingEl"></canvas>

<div class="btn-group mt-2" role="group">

<button type="button" class="btn btn-outline-primary" id="btnStartCameraVotingEl"><i class="bi bi-camera-video"></i> Mulai Kamera</button>

<button type="button" class="btn btn-success" id="btnCaptureAndVerifyFaceEl" disabled>

<span class="spinner-border spinner-border-sm" role="status" aria-hidden="true" id="verificationSpinner"></span>

<i class="bi bi-person-bounding-box"></i> Verifikasi Wajah Saya

</button>

</div>

<div id="photoStatusVotingEl" class="mt-2">Kamera belum aktif.</div>

<div id="snapshotPreviewVotingContainerEl" class="snapshot-preview-voting"></div>

<div id="verificationApiErrorEl" class="text-danger mt-2 small"></div>

</div>

<div class="candidate-selection-step" id="candidateSelectionStepContainer">

<h4 class="text-center mb-3" style="color: #333;">Verifikasi Berhasil! Silakan Pilih Kandidat:</h4>

<?php if (empty($daftar\_kandidat\_voting)): ?>

<div class="alert alert-warning text-center" role="alert"> Belum ada data kandidat.</div>

<?php else: ?>

<div class="kandidat-grid-container">

<?php foreach ($daftar\_kandidat\_voting as $kandidat\_item\_vote):

$path\_foto\_kdt\_vote\_server = UPLOAD\_FOTO\_KANDIDAT\_DIR . basename($kandidat\_item\_vote['foto'] ?? '');

$url\_foto\_kdt\_vote\_web = BASE\_URL . 'uploads/kandidat\_foto/' . basename($kandidat\_item\_vote['foto'] ?? '');

$gambar\_kdt\_display\_vote = BASE\_URL . 'assets/img/default-user.png';

if (!empty($kandidat\_item\_vote['foto']) && file\_exists($path\_foto\_kdt\_vote\_server)) {

$gambar\_kdt\_display\_vote = $url\_foto\_kdt\_vote\_web;

}

?>

<div class="kandidat-card-vote">

<img src="<?= htmlspecialchars($gambar\_kdt\_display\_vote) ?>" class="kandidat-photo-vote" alt="Foto <?= htmlspecialchars($kandidat\_item\_vote['nama']) ?>">

<div class="kandidat-name-vote"><?= htmlspecialchars($kandidat\_item\_vote['nama']) ?></div>

<div class="kandidat-jurusan-vote"><?= htmlspecialchars($kandidat\_item\_vote['jurusan']) ?></div>

<div class="kandidat-visimisi-vote"><strong>Visi & Misi:</strong><br><?= nl2br(htmlspecialchars($kandidat\_item\_vote['visi\_misi'])) ?></div>

<button class="btn btn-primary btn-pilih-kandidat" data-kandidatid="<?= (int)$kandidat\_item\_vote['id'] ?>" data-kandidatnama="<?= htmlspecialchars($kandidat\_item\_vote['nama'], ENT\_QUOTES) ?>">

<i class="bi bi-check-circle-fill me-1"></i> Pilih Kandidat Ini

</button>

</div>

<?php endforeach; ?>

</div>

<?php endif; ?>

</div>

</div>

<div class="modal-vote-confirm" id="modalConfirmVoteChoice">

<div class="modal-content-vote-confirm">

<div class="icon-confirm"><i class="bi bi-person-check-fill"></i></div>

<h5>Konfirmasi Pilihan Suara Anda</h5>

<div id="modalTextConfirmVote"></div>

<form method="post" id="formSubmitActualVote" action="voting.php">

<input type="hidden" name="csrf\_token" value="<?= htmlspecialchars(generate\_csrf\_token()); ?>">

<input type="hidden" name="kandidat\_id\_final" id="inputKandidatIdFinalVote">

<input type="hidden" name="foto\_wajah\_saat\_vote\_data" id="inputFotoWajahSaatVoteFinal">

<input type="hidden" name="submit\_vote\_action" value="1">

<div class="d-flex justify-content-center mt-4">

<button type="submit" class="btn btn-success" id="btnConfirmAndSubmitVote">

<span id="btnTextConfirmVote"><i class="bi bi-send-check-fill"></i> Ya, Kirim Suara Saya</span>

<span id="btnSpinnerConfirmVote" class="spinner-border spinner-border-sm" role="status" aria-hidden="true" style="display: none;"></span>

</button>

<button type="button" class="btn btn-outline-secondary ms-2" id="btnCancelConfirmVote">Tidak, Batalkan</button>

</div>

</form>

</div>

</div>

<script>

const registeredFacePhotoFilename = <?= json\_encode($nama\_file\_foto\_pendaftaran\_user) ?>;

const pythonApiVerifyEndpoint = <?= json\_encode(PYTHON\_API\_URL\_VOTING . '/verify\_face\_match') ?>;

</script>

<script>

document.addEventListener('DOMContentLoaded', function () {

const faceVerificationContainerEl = document.getElementById('faceVerificationStepContainer');

const cameraPreviewVotingEl = document.getElementById('cameraPreviewVotingEl');

const captureCanvasVotingEl = document.getElementById('captureCanvasVotingEl');

const btnStartCameraVotingEl = document.getElementById('btnStartCameraVotingEl');

const btnCaptureAndVerifyFaceEl = document.getElementById('btnCaptureAndVerifyFaceEl');

const photoStatusVotingEl = document.getElementById('photoStatusVotingEl');

const snapshotPreviewVotingContainerEl = document.getElementById('snapshotPreviewVotingContainerEl');

const verificationApiErrorEl = document.getElementById('verificationApiErrorEl');

const verificationSpinnerEl = document.getElementById('verificationSpinner');

const inputFotoWajahSaatVoteFinalEl = document.getElementById('inputFotoWajahSaatVoteFinal');

const candidateSelectionContainerEl = document.getElementById('candidateSelectionStepContainer');

const modalConfirmVoteChoiceEl = document.getElementById('modalConfirmVoteChoice');

const modalTextConfirmVoteEl = document.getElementById('modalTextConfirmVote');

const inputKandidatIdFinalVoteEl = document.getElementById('inputKandidatIdFinalVote');

const btnConfirmAndSubmitVoteEl = document.getElementById('btnConfirmAndSubmitVote');

const btnCancelConfirmVoteEl = document.getElementById('btnCancelConfirmVote');

const btnTextConfirmVoteEl = document.getElementById('btnTextConfirmVote');

const btnSpinnerConfirmVoteEl = document.getElementById('btnSpinnerConfirmVote');

const formSubmitActualVoteEl = document.getElementById('formSubmitActualVote');

let currentVotingStream = null;

let cameraVotingIsActive = false;

let capturedAndVerifiedFotoWajah = null;

// --- Logika Kamera Tahap 1 ---

btnStartCameraVotingEl.addEventListener('click', async () => { /\* ... (sama seperti sebelumnya) ... \*/

if (currentVotingStream) { currentVotingStream.getTracks().forEach(track => track.stop()); cameraVotingIsActive = false; }

snapshotPreviewVotingContainerEl.innerHTML = ''; verificationApiErrorEl.textContent = ''; btnCaptureAndVerifyFaceEl.disabled = true;

try {

photoStatusVotingEl.textContent = 'Meminta izin kamera...';

const constraints = { video: { width: { ideal: 640 }, height: { ideal: 480 } }, audio: false };

currentVotingStream = await navigator.mediaDevices.getUserMedia(constraints);

cameraPreviewVotingEl.srcObject = currentVotingStream;

await cameraPreviewVotingEl.play(); cameraVotingIsActive = true;

photoStatusVotingEl.textContent = 'Kamera aktif. Klik "Verifikasi Wajah Saya".';

btnCaptureAndVerifyFaceEl.disabled = false; btnStartCameraVotingEl.innerHTML = '<i class="bi bi-camera-video-off"></i> Stop Kamera';

} catch (err) {

console.error('Error starting camera for voting:', err);

photoStatusVotingEl.textContent = Error kamera: ${err.name}.; verificationApiErrorEl.textContent = Gagal memulai kamera: ${err.message}. Harap izinkan akses.;

btnStartCameraVotingEl.innerHTML = '<i class="bi bi-camera-video"></i> Mulai Kamera';

}

});

// --- Logika Ambil Foto & Panggil API Python ---

btnCaptureAndVerifyFaceEl.addEventListener('click', async () => { /\* ... (sama seperti sebelumnya, termasuk fetch ke API) ... \*/

if (!cameraVotingIsActive || !currentVotingStream || !currentVotingStream.active) { photoStatusVotingEl.textContent = 'Kamera tidak aktif.'; verificationApiErrorEl.textContent = 'Mulai kamera terlebih dahulu.'; return; }

if (!registeredFacePhotoFilename) { verificationApiErrorEl.textContent = 'Error: Data foto pendaftaran tidak ditemukan.'; return; }

photoStatusVotingEl.textContent = 'Mengambil foto...'; verificationSpinnerEl.style.display = 'inline-block'; btnCaptureAndVerifyFaceEl.disabled = true; btnStartCameraVotingEl.disabled = true; verificationApiErrorEl.textContent = '';

captureCanvasVotingEl.width = cameraPreviewVotingEl.videoWidth; captureCanvasVotingEl.height = cameraPreviewVotingEl.videoHeight;

const context = captureCanvasVotingEl.getContext('2d'); context.drawImage(cameraPreviewVotingEl, 0, 0, captureCanvasVotingEl.width, captureCanvasVotingEl.height);

const voteImageDataURL = captureCanvasVotingEl.toDataURL('image/jpeg', 0.85);

snapshotPreviewVotingContainerEl.innerHTML = <img src="${voteImageDataURL}" alt="Preview Foto Verifikasi">;

photoStatusVotingEl.textContent = 'Mengirim foto untuk verifikasi...';

const formData = new FormData(); formData.append('image\_vote\_base64', voteImageDataURL); formData.append('registered\_filename', registeredFacePhotoFilename);

try {

const response = await fetch(pythonApiVerifyEndpoint, { method: 'POST', body: formData });

if (!response.ok) { throw new Error(API Verifikasi (${response.status}): ${response.statusText}); }

const result = await response.json();

if (result.success && result.match) {

photoStatusVotingEl.textContent = 'Verifikasi wajah berhasil!';

capturedAndVerifiedFotoWajah = voteImageDataURL; inputFotoWajahSaatVoteFinalEl.value = capturedAndVerifiedFotoWajah;

faceVerificationContainerEl.style.display = 'none'; candidateSelectionContainerEl.style.display = 'block';

if (currentVotingStream) { currentVotingStream.getTracks().forEach(track => track.stop()); cameraVotingIsActive = false; }

} else { throw new Error(result.message || "Verifikasi wajah gagal."); }

} catch (error) {

console.error('Error calling verification API:', error); photoStatusVotingEl.textContent = 'Verifikasi Gagal.'; verificationApiErrorEl.textContent = Error: ${error.message}. Coba lagi.;

btnCaptureAndVerifyFaceEl.disabled = false; btnStartCameraVotingEl.disabled = false; capturedAndVerifiedFotoWajah = null; inputFotoWajahSaatVoteFinalEl.value = '';

} finally { verificationSpinnerEl.style.display = 'none'; }

});

// --- Logika Modal Konfirmasi & Submit Form ---

document.querySelectorAll('.btn-pilih-kandidat').forEach(button => { /\* ... (sama seperti sebelumnya) ... \*/

button.addEventListener('click', function () {

if (!capturedAndVerifiedFotoWajah) { alert("Verifikasi wajah belum berhasil."); faceVerificationContainerEl.style.display = 'block'; candidateSelectionContainerEl.style.display = 'none'; verificationApiErrorEl.textContent = 'Harap lakukan verifikasi wajah.'; return; }

const kandidatId = this.dataset.kandidatid; const kandidatNama = this.dataset.kandidatnama;

inputKandidatIdFinalVoteEl.value = kandidatId; modalTextConfirmVoteEl.innerHTML = Anda akan memilih <strong>${kandidatNama}</strong>. Pastikan pilihan Anda sudah benar.;

modalConfirmVoteChoiceEl.classList.add('show');

});

});

function closeModalVoteConfirm() { /\* ... (sama seperti sebelumnya) ... \*/

modalConfirmVoteChoiceEl.classList.remove('show'); btnConfirmAndSubmitVoteEl.disabled = false; btnTextConfirmVoteEl.style.display = 'inline'; btnSpinnerConfirmVoteEl.style.display = 'none';

}

btnCancelConfirmVoteEl.addEventListener('click', closeModalVoteConfirm);

modalConfirmVoteChoiceEl.addEventListener('click', (event) => { if (event.target === modalConfirmVoteChoiceEl) closeModalVoteConfirm(); });

formSubmitActualVoteEl.addEventListener('submit', function(event) { /\* ... (sama seperti sebelumnya) ... \*/

if (!capturedAndVerifiedFotoWajah || !inputFotoWajahSaatVoteFinalEl.value) { alert("Data foto verifikasi wajah hilang. Ulangi proses."); event.preventDefault(); closeModalVoteConfirm(); faceVerificationContainerEl.style.display = 'block'; candidateSelectionContainerEl.style.display = 'none'; verificationApiErrorEl.textContent = 'Harap ulangi verifikasi wajah.'; return false; }

btnConfirmAndSubmitVoteEl.disabled = true; btnTextConfirmVoteEl.style.display = 'none'; btnSpinnerConfirmVoteEl.style.display = 'inline';

});

window.addEventListener('beforeunload', () => { if (currentVotingStream) { currentVotingStream.getTracks().forEach(track => track.stop()); } });

});

</script>

<?php

require 'template/footer.php';

?>

<?php

// voting.php - Halaman Proses Voting (Integrasi API Python Verifikasi + Debug Prepare)

require\_once 'config.php'; // Memuat konfigurasi & memulai session

// --- 1. Validasi Akses, Status Vote, Jadwal ---

if (!is\_login() || !isset($\_SESSION['user']['role']) || $\_SESSION['user']['role'] !== 'pemilih') {

$\_SESSION['flash\_error\_login'] = "Anda harus login sebagai pemilih untuk voting.";

header('Location: login.php'); exit;

}

$user\_id\_pemilih = $\_SESSION['user']['id'];

$status\_sudah\_vote\_session = isset($\_SESSION['user']['sudah\_vote']) ? (int)$\_SESSION['user']['sudah\_vote'] : 0;

$nama\_file\_foto\_pendaftaran\_user = null; // Akan diisi dari DB

// Cek status vote dari DB & ambil nama file foto pendaftaran

$user\_sudah\_vote\_dari\_db = false;

if ($conn) {

// Ambil status vote dan foto pendaftaran dalam satu query

$stmt\_get\_user\_data = $conn->prepare("SELECT sudah\_vote, foto\_wajah\_pendaftaran FROM users WHERE id = ? AND role = 'pemilih'");

if ($stmt\_get\_user\_data) {

$stmt\_get\_user\_data->bind\_param("i", $user\_id\_pemilih);

$stmt\_get\_user\_data->execute();

$stmt\_get\_user\_data->bind\_result($db\_sudah\_vote, $db\_foto\_pendaftaran);

if ($stmt\_get\_user\_data->fetch()) {

$status\_sudah\_vote\_session = (int)$db\_sudah\_vote;

$\_SESSION['user']['sudah\_vote'] = $status\_sudah\_vote\_session; // Update session

$nama\_file\_foto\_pendaftaran\_user = $db\_foto\_pendaftaran; // Ambil nama file

if ($status\_sudah\_vote\_session > 0) {

$user\_sudah\_vote\_dari\_db = true;

}

} else {

error\_log("Voting: User ID {$user\_id\_pemilih} tidak ditemukan / bukan pemilih.");

$\_SESSION['flash\_error\_login'] = "Data pengguna tidak valid.";

header('Location: login.php'); exit;

}

$stmt\_get\_user\_data->close();

} else {

error\_log("Voting: Gagal prepare get user data - " . $conn->error);

die("Terjadi kesalahan saat memverifikasi data pengguna.");

}

} else { die("Koneksi database bermasalah."); }

// Redirect jika sudah vote

if ($user\_sudah\_vote\_dari\_db) {

$\_SESSION['flash\_info\_dashboard'] = "Anda sudah menggunakan hak suara Anda.";

header('Location: dashboard.php'); exit;

}

// Redirect jika tidak punya foto pendaftaran

if (empty($nama\_file\_foto\_pendaftaran\_user)) {

error\_log("Voting: User ID {$user\_id\_pemilih} tidak memiliki foto pendaftaran.");

$\_SESSION['flash\_error\_dashboard'] = "Data foto pendaftaran Anda tidak ditemukan. Hubungi administrator.";

header('Location: dashboard.php'); exit;

}

// Cek Jadwal Voting Aktif

$jadwal\_voting\_aktif = false;

$pesan\_jadwal\_voting = "Informasi jadwal voting tidak tersedia atau belum diatur.";

$evoting\_start\_str\_db = get\_setting('evoting\_start', $conn);

$evoting\_end\_str\_db = get\_setting('evoting\_end', $conn);

if (!empty($evoting\_start\_str\_db) && !empty($evoting\_end\_str\_db)) {

try {

$tz = new DateTimeZone('Asia/Jakarta');

$start = new DateTime($evoting\_start\_str\_db, $tz); $end = new DateTime($evoting\_end\_str\_db, $tz); $now = new DateTime('now', $tz);

if ($now >= $start && $now <= $end) $jadwal\_voting\_aktif = true;

elseif ($now < $start) $pesan\_jadwal\_voting = "Periode voting belum dimulai (Mulai: " . $start->format('d/m/Y H:i') . ").";

else $pesan\_jadwal\_voting = "Periode voting telah berakhir (Selesai: " . $end->format('d/m/Y H:i') . ").";

} catch (Exception $e) { error\_log("Voting: Error parsing jadwal - ".$e->getMessage()); $pesan\_jadwal\_voting = "Error format jadwal."; }

}

if (!$jadwal\_voting\_aktif) {

$\_SESSION['flash\_error\_dashboard'] = $pesan\_jadwal\_voting;

header('Location: dashboard.php'); exit;

}

// Ambil Data Kandidat

$daftar\_kandidat\_voting = [];

if ($conn) {

$stmt\_get\_kdt = $conn->prepare("SELECT id, nama, jurusan, foto, visi\_misi FROM kandidat ORDER BY nama ASC");

if ($stmt\_get\_kdt) { /\* ... (ambil data kandidat seperti sebelumnya) ... \*/

$stmt\_get\_kdt->execute(); $result\_kdt = $stmt\_get\_kdt->get\_result();

while ($kdt\_row = $result\_kdt->fetch\_assoc()) { $daftar\_kandidat\_voting[] = $kdt\_row; }

$stmt\_get\_kdt->close();

} else { die("Gagal memuat data kandidat."); }

}

// --- 4. Proses Submit Voting ---

$error\_submit\_vote = '';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['submit\_vote\_action'])) {

if (!validate\_csrf\_token()) {

$error\_submit\_vote = "Sesi tidak valid. Ulangi proses voting.";

} else {

$id\_kandidat\_dipilih = filter\_input(INPUT\_POST, 'kandidat\_id\_final', FILTER\_VALIDATE\_INT);

$foto\_wajah\_lolos\_verifikasi\_base64 = $\_POST['foto\_wajah\_saat\_vote\_data'] ?? '';

if (!$id\_kandidat\_dipilih || $id\_kandidat\_dipilih <= 0) {

$error\_submit\_vote = "Pilihan kandidat tidak valid.";

} elseif (empty($foto\_wajah\_lolos\_verifikasi\_base64)) {

$error\_submit\_vote = "Data verifikasi foto wajah tidak lengkap.";

} else {

// Cek ulang status vote

$stmt\_cek\_final = $conn->prepare("SELECT COUNT(\*) FROM voting WHERE user\_id = ?");

$stmt\_cek\_final->bind\_param("i", $user\_id\_pemilih); $stmt\_cek\_final->execute();

$stmt\_cek\_final->bind\_result($count\_final); $stmt\_cek\_final->fetch(); $stmt\_cek\_final->close();

if ($count\_final > 0) {

$\_SESSION['user']['sudah\_vote'] = 1;

$\_SESSION['flash\_info\_dashboard'] = "Anda sudah tercatat voting.";

header('Location: dashboard.php'); exit;

}

// Simpan foto wajah saat voting

$nama\_file\_foto\_saat\_vote\_db = null;

$path\_simpan\_foto\_saat\_vote = null;

if (!is\_writable(UPLOAD\_FOTO\_SAAT\_VOTE\_DIR)) {

$error\_submit\_vote = "Error: Server tidak bisa menulis file foto voting.";

error\_log("Voting Submit: Direktori UPLOAD\_FOTO\_SAAT\_VOTE\_DIR tidak writable.");

} else {

if (preg\_match('/^data:image\/(png|jpeg|jpg);base64,/', $foto\_wajah\_lolos\_verifikasi\_base64, $type\_m)) {

$ext = $type\_m[1]; $data = substr($foto\_wajah\_lolos\_verifikasi\_base64, strpos($foto\_wajah\_lolos\_verifikasi\_base64, ',') + 1);

$data = base64\_decode($data);

if ($data === false) { $error\_submit\_vote = "Error: Gagal decode foto voting."; }

else {

$nama\_file\_foto\_saat\_vote\_db = 'vote\_wajah\_' . $user\_id\_pemilih . '\_' . uniqid() . '.' . $ext;

$path\_simpan\_foto\_saat\_vote = UPLOAD\_FOTO\_SAAT\_VOTE\_DIR . $nama\_file\_foto\_saat\_vote\_db;

if (file\_put\_contents($path\_simpan\_foto\_saat\_vote, $data) === false) {

$error\_submit\_vote = "Error: Gagal simpan file foto voting."; $nama\_file\_foto\_saat\_vote\_db = null;

}

}

} else { $error\_submit\_vote = "Error: Format data foto voting tidak valid."; }

}

// Lanjutkan ke DB jika tidak ada error sebelumnya DAN foto berhasil disimpan

if (empty($error\_submit\_vote) && $nama\_file\_foto\_saat\_vote\_db !== null) {

$conn->begin\_transaction();

try {

// 1. Insert ke tabel voting

$stmt\_ins = $conn->prepare("INSERT INTO voting (user\_id, kandidat\_id, foto\_saat\_vote) VALUES (?, ?, ?)");

// Tambahkan pengecekan prepare insert vote

if ($stmt\_ins === false) {

throw new Exception("MySQL Prepare Error (INSERT voting): " . $conn->error);

}

$stmt\_ins->bind\_param("iis", $user\_id\_pemilih, $id\_kandidat\_dipilih, $nama\_file\_foto\_saat\_vote\_db);

if (!$stmt\_ins->execute()) throw new Exception("Gagal execute insert data voting: " . $stmt\_ins->error);

$stmt\_ins->close();

// 2. Update status 'sudah\_vote' di tabel 'users'

$stmt\_upd\_u = $conn->prepare("UPDATE users SET sudah\_vote = 1 WHERE id = ?");

// === PENGECEKAN EKSPLISIT SETELAH PREPARE ===

if ($stmt\_upd\_u === false) {

// Jika prepare GAGAL, lempar exception dengan error MySQL

throw new Exception("MySQL Prepare Error (UPDATE users): " . $conn->error);

}

// ===========================================

// Jika prepare berhasil, baru bind\_param (ini baris ~136 yang error sebelumnya)

$stmt\_upd\_u->bind\_param("i", $user\_id\_pemilih);

if (!$stmt\_upd\_u->execute()) throw new Exception("Gagal execute update status user: " . $stmt\_upd\_u->error);

$stmt\_upd\_u->close();

// 3. Update jumlah suara di tabel 'kandidat'

$stmt\_upd\_k = $conn->prepare("UPDATE kandidat SET jumlah\_suara = jumlah\_suara + 1 WHERE id = ?");

// Tambahkan pengecekan prepare update kandidat

if ($stmt\_upd\_k === false) {

throw new Exception("MySQL Prepare Error (UPDATE kandidat): " . $conn->error);

}

$stmt\_upd\_k->bind\_param("i", $id\_kandidat\_dipilih);

if (!$stmt\_upd\_k->execute()) throw new Exception("Gagal execute update suara kandidat: " . $stmt\_upd\_k->error);

$stmt\_upd\_k->close();

$conn->commit(); // Commit jika semua berhasil

$\_SESSION['user']['sudah\_vote'] = 1;

$\_SESSION['flash\_success\_dashboard'] = "Terima kasih! Suara Anda telah berhasil direkam.";

header('Location: dashboard.php'); exit;

} catch (Exception $e) {

$conn->rollback(); // Rollback jika ada error di try block

// Simpan pesan error spesifik dari exception untuk ditampilkan

$error\_submit\_vote = "Terjadi kesalahan teknis: " . htmlspecialchars($e->getMessage());

error\_log("Voting Submit Transaksi Gagal: " . $e->getMessage()); // Log error lengkap

// Hapus file foto yang mungkin sempat tersimpan jika transaksi DB gagal

if ($path\_simpan\_foto\_saat\_vote && file\_exists($path\_simpan\_foto\_saat\_vote)) {

@unlink($path\_simpan\_foto\_saat\_vote);

error\_log("Voting Submit: File foto '{$nama\_file\_foto\_saat\_vote\_db}' dihapus karena transaksi DB gagal.");

}

} // Akhir blok try-catch

} // Akhir if ($error\_submit\_vote empty && file foto tersimpan)

} // Akhir else (cek sudah vote final)

} // Akhir validasi input dasar & foto

} // Akhir POST check

// Definisikan URL API Python untuk JavaScript

define('PYTHON\_API\_URL\_VOTING', 'http://localhost:5000/api');

// Muat Template Header

require 'template/header.php';

?>

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&display=swap" rel="stylesheet">

<style>

/\* ... (CSS dari versi sebelumnya diletakkan di sini) ... \*/

body { font-family: 'Inter', Arial, sans-serif; background-color: #f0f2f5; }

.voting-main-container { max-width: 1000px; margin: 1.5rem auto; padding: 0 1rem; }

.voting-page-title { text-align: center; font-weight: 700; color: var(--primary-color, #4f46e5); margin-bottom: 2rem; font-size: 2rem; }

.face-verification-step { background-color: #ffffff; padding: 2rem; border-radius: 12px; box-shadow: 0 6px 20px rgba(0,0,0,0.07); margin-bottom: 2.5rem; text-align: center; }

.face-verification-step h4 { font-weight: 600; color: #333; margin-bottom: 0.75rem; }

.face-verification-step p.instruction { color: #555; margin-bottom: 1.5rem; font-size: 0.95rem; }

#cameraPreviewVotingEl { width: 100%; max-width: 320px; height: auto; aspect-ratio: 4 / 3; border-radius: 8px; background-color: #333; margin: 0 auto 1rem auto; display: block; border: 3px solid #e0e0e0; }

#captureCanvasVotingEl { display: none; }

#photoStatusVotingEl { font-style: italic; color: #6c757d; margin-top: 0.75rem; min-height: 22px; font-size: 0.9rem;}

#verificationApiErrorEl { font-weight: bold; color: #dc3545; } /\* Pesan error API \*/

#verificationSpinner { display: none; margin-right: 5px;} /\* Spinner loading \*/

.snapshot-preview-voting img { max-width: 120px; max-height: 90px; border:1px solid #ccc; border-radius:5px; margin-top:0.5rem;}

.candidate-selection-step { display: none; }

.kandidat-grid-container { display: grid; grid-template-columns: repeat(auto-fit, minmax(280px, 1fr)); gap: 1.8rem; }

.kandidat-card-vote { background: #ffffff; border-radius: 12px; box-shadow: 0 5px 15px rgba(0,0,0,0.06); padding: 1.5rem; text-align: center; display: flex; flex-direction: column; transition: transform 0.2s ease, box-shadow 0.2s ease; }

.kandidat-card-vote:hover { transform: translateY(-6px); box-shadow: 0 10px 25px rgba(0,0,0,0.1); }

.kandidat-card-vote .kandidat-photo-vote { width: 100px; height: 100px; object-fit: cover; border-radius: 50%; margin: 0 auto 1rem auto; border: 4px solid var(--primary-color, #4f46e5); padding: 3px; background-color: #e9ecef; }

.kandidat-card-vote .kandidat-name-vote { font-size: 1.2rem; font-weight: 600; color: #212529; margin-bottom: 0.25rem; }

.kandidat-card-vote .kandidat-jurusan-vote { color: #6c757d; font-size: 0.9rem; margin-bottom: 0.75rem; }

.kandidat-card-vote .kandidat-visimisi-vote { font-size: 0.85rem; color: #495057; margin-bottom: 1rem; text-align: left; background-color: #f8f9fa; border-radius: 8px; padding: 0.75rem; white-space: pre-line; overflow-y: auto; max-height: 100px; flex-grow: 1; border: 1px solid #e9ecef; }

.kandidat-card-vote .btn-pilih-kandidat { font-size: 1rem; padding: 0.6rem 1.2rem; border-radius: 50px; font-weight: 600; margin-top: auto; width: 90%; margin-left: auto; margin-right: auto; }

.modal-vote-confirm { display: none; position: fixed; z-index: 2050; left: 0; top: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.55); backdrop-filter: blur(4px); align-items: center; justify-content: center; opacity: 0; transition: opacity 0.25s ease; }

.modal-vote-confirm.show { display: flex; opacity: 1; }

.modal-content-vote-confirm { background: #fff; border-radius: 15px; padding: 2rem; box-shadow: 0 8px 25px rgba(0,0,0,0.2); text-align: center; max-width: 420px; width: 90%; margin: 1rem; transform: scale(0.95); transition: transform 0.25s ease; }

.modal-vote-confirm.show .modal-content-vote-confirm { transform: scale(1); }

.modal-content-vote-confirm .icon-confirm { font-size: 2.8rem; color: var(--primary-color); margin-bottom: 1rem; }

.modal-content-vote-confirm h5 { font-weight: 600; margin-bottom: 0.75rem; }

.modal-content-vote-confirm #modalTextConfirmVote { margin-bottom: 1.5rem; color: #495057; font-size: 1.05rem; }

.modal-content-vote-confirm .btn { min-width: 120px; margin: 0 0.5rem; border-radius: 50px; padding: 0.7rem 1.3rem; font-weight: 600; }

.alert-voting-page { margin-bottom: 1.5rem; }

</style>

<div class="voting-main-container">

<div class="voting-page-title"><i class="bi bi-pencil-square"></i> Halaman Pemilihan Suara</div>

<?php if (!empty($error\_submit\_vote)): ?>

<div class="alert alert-danger alert-dismissible fade show alert-voting-page" role="alert">

<i class="bi bi-exclamation-octagon-fill me-2"></i> Voting Gagal: <?= htmlspecialchars($error\_submit\_vote) ?>

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

</div>

<?php endif; ?>

<div class="face-verification-step" id="faceVerificationStepContainer">

<h4><i class="bi bi-shield-check"></i> Verifikasi Wajah untuk Voting</h4>

<p class="instruction">Ambil foto wajah Anda. Sistem akan membandingkannya dengan foto pendaftaran.</p>

<video id="cameraPreviewVotingEl" autoplay muted playsinline></video>

<canvas id="captureCanvasVotingEl"></canvas>

<div class="btn-group mt-2" role="group">

<button type="button" class="btn btn-outline-primary" id="btnStartCameraVotingEl"><i class="bi bi-camera-video"></i> Mulai Kamera</button>

<button type="button" class="btn btn-success" id="btnCaptureAndVerifyFaceEl" disabled>

<span class="spinner-border spinner-border-sm" role="status" aria-hidden="true" id="verificationSpinner"></span>

<i class="bi bi-person-bounding-box"></i> Verifikasi Wajah Saya

</button>

</div>

<div id="photoStatusVotingEl" class="mt-2">Kamera belum aktif.</div>

<div id="snapshotPreviewVotingContainerEl" class="snapshot-preview-voting"></div>

<div id="verificationApiErrorEl" class="text-danger mt-2 small"></div>

</div>

<div class="candidate-selection-step" id="candidateSelectionStepContainer">

<h4 class="text-center mb-3" style="color: #333;">Verifikasi Berhasil! Silakan Pilih Kandidat:</h4>

<?php if (empty($daftar\_kandidat\_voting)): ?>

<div class="alert alert-warning text-center" role="alert"> Belum ada data kandidat.</div>

<?php else: ?>

<div class="kandidat-grid-container">

<?php foreach ($daftar\_kandidat\_voting as $kandidat\_item\_vote):

$path\_foto\_kdt\_vote\_server = UPLOAD\_FOTO\_KANDIDAT\_DIR . basename($kandidat\_item\_vote['foto'] ?? '');

$url\_foto\_kdt\_vote\_web = BASE\_URL . 'uploads/kandidat\_foto/' . basename($kandidat\_item\_vote['foto'] ?? '');

$gambar\_kdt\_display\_vote = BASE\_URL . 'assets/img/default-user.png';

if (!empty($kandidat\_item\_vote['foto']) && file\_exists($path\_foto\_kdt\_vote\_server)) {

$gambar\_kdt\_display\_vote = $url\_foto\_kdt\_vote\_web;

}

?>

<div class="kandidat-card-vote">

<img src="<?= htmlspecialchars($gambar\_kdt\_display\_vote) ?>" class="kandidat-photo-vote" alt="Foto <?= htmlspecialchars($kandidat\_item\_vote['nama']) ?>">

<div class="kandidat-name-vote"><?= htmlspecialchars($kandidat\_item\_vote['nama']) ?></div>

<div class="kandidat-jurusan-vote"><?= htmlspecialchars($kandidat\_item\_vote['jurusan']) ?></div>

<div class="kandidat-visimisi-vote"><strong>Visi & Misi:</strong><br><?= nl2br(htmlspecialchars($kandidat\_item\_vote['visi\_misi'])) ?></div>

<button class="btn btn-primary btn-pilih-kandidat" data-kandidatid="<?= (int)$kandidat\_item\_vote['id'] ?>" data-kandidatnama="<?= htmlspecialchars($kandidat\_item\_vote['nama'], ENT\_QUOTES) ?>">

<i class="bi bi-check-circle-fill me-1"></i> Pilih Kandidat Ini

</button>

</div>

<?php endforeach; ?>

</div>

<?php endif; ?>

</div>

</div>

<div class="modal-vote-confirm" id="modalConfirmVoteChoice">

<div class="modal-content-vote-confirm">

<div class="icon-confirm"><i class="bi bi-person-check-fill"></i></div>

<h5>Konfirmasi Pilihan Suara Anda</h5>

<div id="modalTextConfirmVote"></div>

<form method="post" id="formSubmitActualVote" action="voting.php">

<input type="hidden" name="csrf\_token" value="<?= htmlspecialchars(generate\_csrf\_token()); ?>">

<input type="hidden" name="kandidat\_id\_final" id="inputKandidatIdFinalVote">

<input type="hidden" name="foto\_wajah\_saat\_vote\_data" id="inputFotoWajahSaatVoteFinal">

<input type="hidden" name="submit\_vote\_action" value="1">

<div class="d-flex justify-content-center mt-4">

<button type="submit" class="btn btn-success" id="btnConfirmAndSubmitVote">

<span id="btnTextConfirmVote"><i class="bi bi-send-check-fill"></i> Ya, Kirim Suara Saya</span>

<span id="btnSpinnerConfirmVote" class="spinner-border spinner-border-sm" role="status" aria-hidden="true" style="display: none;"></span>

</button>

<button type="button" class="btn btn-outline-secondary ms-2" id="btnCancelConfirmVote">Tidak, Batalkan</button>

</div>

</form>

</div>

</div>

<script>

const registeredFacePhotoFilename = <?= json\_encode($nama\_file\_foto\_pendaftaran\_user) ?>;

const pythonApiVerifyEndpoint = <?= json\_encode(PYTHON\_API\_URL\_VOTING . '/verify\_face\_match') ?>;

</script>

<script>

document.addEventListener('DOMContentLoaded', function () {

const faceVerificationContainerEl = document.getElementById('faceVerificationStepContainer');

const cameraPreviewVotingEl = document.getElementById('cameraPreviewVotingEl');

const captureCanvasVotingEl = document.getElementById('captureCanvasVotingEl');

const btnStartCameraVotingEl = document.getElementById('btnStartCameraVotingEl');

const btnCaptureAndVerifyFaceEl = document.getElementById('btnCaptureAndVerifyFaceEl');

const photoStatusVotingEl = document.getElementById('photoStatusVotingEl');

const snapshotPreviewVotingContainerEl = document.getElementById('snapshotPreviewVotingContainerEl');

const verificationApiErrorEl = document.getElementById('verificationApiErrorEl');

const verificationSpinnerEl = document.getElementById('verificationSpinner');

const inputFotoWajahSaatVoteFinalEl = document.getElementById('inputFotoWajahSaatVoteFinal');

const candidateSelectionContainerEl = document.getElementById('candidateSelectionStepContainer');

const modalConfirmVoteChoiceEl = document.getElementById('modalConfirmVoteChoice');

const modalTextConfirmVoteEl = document.getElementById('modalTextConfirmVote');

const inputKandidatIdFinalVoteEl = document.getElementById('inputKandidatIdFinalVote');

const btnConfirmAndSubmitVoteEl = document.getElementById('btnConfirmAndSubmitVote');

const btnCancelConfirmVoteEl = document.getElementById('btnCancelConfirmVote');

const btnTextConfirmVoteEl = document.getElementById('btnTextConfirmVote');

const btnSpinnerConfirmVoteEl = document.getElementById('btnSpinnerConfirmVote');

const formSubmitActualVoteEl = document.getElementById('formSubmitActualVote');

let currentVotingStream = null;

let cameraVotingIsActive = false;

let capturedAndVerifiedFotoWajah = null;

// --- Logika Kamera Tahap 1 ---

btnStartCameraVotingEl.addEventListener('click', async () => { /\* ... (sama seperti sebelumnya) ... \*/

if (currentVotingStream) { currentVotingStream.getTracks().forEach(track => track.stop()); cameraVotingIsActive = false; }

snapshotPreviewVotingContainerEl.innerHTML = ''; verificationApiErrorEl.textContent = ''; btnCaptureAndVerifyFaceEl.disabled = true;

try {

photoStatusVotingEl.textContent = 'Meminta izin kamera...';

const constraints = { video: { width: { ideal: 640 }, height: { ideal: 480 } }, audio: false };

currentVotingStream = await navigator.mediaDevices.getUserMedia(constraints);

cameraPreviewVotingEl.srcObject = currentVotingStream;

await cameraPreviewVotingEl.play(); cameraVotingIsActive = true;

photoStatusVotingEl.textContent = 'Kamera aktif. Klik "Verifikasi Wajah Saya".';

btnCaptureAndVerifyFaceEl.disabled = false; btnStartCameraVotingEl.innerHTML = '<i class="bi bi-camera-video-off"></i> Stop Kamera';

} catch (err) {

console.error('Error starting camera for voting:', err);

photoStatusVotingEl.textContent = Error kamera: ${err.name}.; verificationApiErrorEl.textContent = Gagal memulai kamera: ${err.message}. Harap izinkan akses.;

btnStartCameraVotingEl.innerHTML = '<i class="bi bi-camera-video"></i> Mulai Kamera';

}

});

// --- Logika Ambil Foto & Panggil API Python ---

btnCaptureAndVerifyFaceEl.addEventListener('click', async () => { /\* ... (sama seperti sebelumnya, termasuk fetch ke API) ... \*/

if (!cameraVotingIsActive || !currentVotingStream || !currentVotingStream.active) { photoStatusVotingEl.textContent = 'Kamera tidak aktif.'; verificationApiErrorEl.textContent = 'Mulai kamera terlebih dahulu.'; return; }

if (!registeredFacePhotoFilename) { verificationApiErrorEl.textContent = 'Error: Data foto pendaftaran tidak ditemukan.'; return; }

photoStatusVotingEl.textContent = 'Mengambil foto...'; verificationSpinnerEl.style.display = 'inline-block'; btnCaptureAndVerifyFaceEl.disabled = true; btnStartCameraVotingEl.disabled = true; verificationApiErrorEl.textContent = '';

captureCanvasVotingEl.width = cameraPreviewVotingEl.videoWidth; captureCanvasVotingEl.height = cameraPreviewVotingEl.videoHeight;

const context = captureCanvasVotingEl.getContext('2d'); context.drawImage(cameraPreviewVotingEl, 0, 0, captureCanvasVotingEl.width, captureCanvasVotingEl.height);

const voteImageDataURL = captureCanvasVotingEl.toDataURL('image/jpeg', 0.85);

snapshotPreviewVotingContainerEl.innerHTML = <img src="${voteImageDataURL}" alt="Preview Foto Verifikasi">;

photoStatusVotingEl.textContent = 'Mengirim foto untuk verifikasi...';

const formData = new FormData(); formData.append('image\_vote\_base64', voteImageDataURL); formData.append('registered\_filename', registeredFacePhotoFilename);

try {

const response = await fetch(pythonApiVerifyEndpoint, { method: 'POST', body: formData });

if (!response.ok) { throw new Error(API Verifikasi (${response.status}): ${response.statusText}); }

const result = await response.json();

if (result.success && result.match) {

photoStatusVotingEl.textContent = 'Verifikasi wajah berhasil!';

capturedAndVerifiedFotoWajah = voteImageDataURL; inputFotoWajahSaatVoteFinalEl.value = capturedAndVerifiedFotoWajah;

faceVerificationContainerEl.style.display = 'none'; candidateSelectionContainerEl.style.display = 'block';

if (currentVotingStream) { currentVotingStream.getTracks().forEach(track => track.stop()); cameraVotingIsActive = false; }

} else { throw new Error(result.message || "Verifikasi wajah gagal."); }

} catch (error) {

console.error('Error calling verification API:', error); photoStatusVotingEl.textContent = 'Verifikasi Gagal.'; verificationApiErrorEl.textContent = Error: ${error.message}. Coba lagi.;

btnCaptureAndVerifyFaceEl.disabled = false; btnStartCameraVotingEl.disabled = false; capturedAndVerifiedFotoWajah = null; inputFotoWajahSaatVoteFinalEl.value = '';

} finally { verificationSpinnerEl.style.display = 'none'; }

});

// --- Logika Modal Konfirmasi & Submit Form ---

document.querySelectorAll('.btn-pilih-kandidat').forEach(button => { /\* ... (sama seperti sebelumnya) ... \*/

button.addEventListener('click', function () {

if (!capturedAndVerifiedFotoWajah) { alert("Verifikasi wajah belum berhasil."); faceVerificationContainerEl.style.display = 'block'; candidateSelectionContainerEl.style.display = 'none'; verificationApiErrorEl.textContent = 'Harap lakukan verifikasi wajah.'; return; }

const kandidatId = this.dataset.kandidatid; const kandidatNama = this.dataset.kandidatnama;

inputKandidatIdFinalVoteEl.value = kandidatId; modalTextConfirmVoteEl.innerHTML = Anda akan memilih <strong>${kandidatNama}</strong>. Pastikan pilihan Anda sudah benar.;

modalConfirmVoteChoiceEl.classList.add('show');

});

});

function closeModalVoteConfirm() { /\* ... (sama seperti sebelumnya) ... \*/

modalConfirmVoteChoiceEl.classList.remove('show'); btnConfirmAndSubmitVoteEl.disabled = false; btnTextConfirmVoteEl.style.display = 'inline'; btnSpinnerConfirmVoteEl.style.display = 'none';

}

btnCancelConfirmVoteEl.addEventListener('click', closeModalVoteConfirm);

modalConfirmVoteChoiceEl.addEventListener('click', (event) => { if (event.target === modalConfirmVoteChoiceEl) closeModalVoteConfirm(); });

formSubmitActualVoteEl.addEventListener('submit', function(event) { /\* ... (sama seperti sebelumnya) ... \*/

if (!capturedAndVerifiedFotoWajah || !inputFotoWajahSaatVoteFinalEl.value) { alert("Data foto verifikasi wajah hilang. Ulangi proses."); event.preventDefault(); closeModalVoteConfirm(); faceVerificationContainerEl.style.display = 'block'; candidateSelectionContainerEl.style.display = 'none'; verificationApiErrorEl.textContent = 'Harap ulangi verifikasi wajah.'; return false; }

btnConfirmAndSubmitVoteEl.disabled = true; btnTextConfirmVoteEl.style.display = 'none'; btnSpinnerConfirmVoteEl.style.display = 'inline';

});

window.addEventListener('beforeunload', () => { if (currentVotingStream) { currentVotingStream.getTracks().forEach(track => track.stop()); } });

});

</script>

<?php

require 'template/footer.php';

?>