

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing TA Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Rais, S.Pd., M.Kom.
NIDN : 0614108501
NIPY : 07.011.083
Jabatan Struktural : Ka. UPT Pelatihan dan Sertifikasi
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut :

Nama : Alya Nabila Hermanto
NIM : 22040087
Program Studi : DIII Teknik Komputer

Judul TA : APLIKASI IURAN DANA KEMATIAN PADA
YAYASAN TRI DHARMA MENGGUNAKAN
PAYMENT GATEWAY BERBASIS WEB

Demikian pernyataan dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 14 April 2025

Mengetahui,

Ka. Prodi DIII Teknik Komputer,



Ida Afriliana, ST., M.Kom
NIPY. 12.013.168

Dosen Pembimbing I,



Rais, S.Pd., M.Kom.
NIPY. 07.011.083

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing 2

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Nurohim, S.ST, M.Kom
NIDN : 0625067701
NIPY : 09.017.342
Jabatan Struktural : Dosen Tetap
Jabatan Fungsional : Asisten Ahli

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut :

Nama : Alya Nabila Hermanto
NIM : 22040087
Program Studi : DIII Teknik Komputer

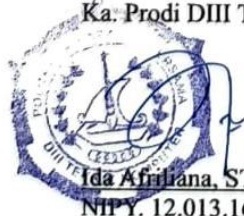
Judul TA : APLIKASI IURAN DANA KEMATIAN PADA
YAYASAN TRI DHARMA MENGGUNAKAN
PAYMENT GATEWAY BERBASIS WEB

Demikian pernyataan dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, 15 April 2025

Mengetahui,

Ka. Prodi DIII Teknik Komputer,


Ida Afrilliana, ST., M.Kom
NIPY. 12.013.168

Dosen Pembimbing II,


Nurohim, S.ST, M.Kom
NIPY. 09.017.342

Lampiran 3 Dokumentasi Observasi



Gambar Kartu Iuran Dana Kematian



Gambar dokumentasi pada saat observasi

Lampiran 4 Coding login.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Login</title>

    <!-- Custom fonts for this template-->
    <link href="{{asset('assets/vendor/fontawesome-
free/css/all.min.css')}}" rel="stylesheet" type="text/css">
    <link

href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,
300i,400,400i,600,600i,700,700i,800,800i,900,900i"
rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="{{asset('assets/css/sb-admin-2.min.css')}}"
rel="stylesheet">

    <style>
        html, body {
            height: 100%;
        }
        .center-wrapper {
            height: 100%;
            display: flex;
            align-items: center;
            justify-content: center;
        }
    </style>

</head>

<body class="bg-gradient-primary">

    <div class="container center-wrapper">

        <div class="col-xl-5 col-lg-6 col-md-8">

            <div class="card o-hidden border-0 shadow-lg my-5 mx-
auto" style="max-width: 450px; border-radius: 1rem;">
                <div class="card-body p-0">
                    <div class="p-5">
                        <div class="text-center mb-4">
                            <h1 class="h4 text-gray-900 mb-
```

```

2">Welcome!</h1>
1">Iuran Dana Duka</h2>
Tri Dharma Tegal</h2>
</div>
<form class="user" method="post"
action="{{route('auth.verify')}}">
    @csrf
    <div class="form-group mb-3">
        <input type="email" name="email"
class="form-control form-control-user shadow-sm"
        id="emailInput"
placeholder="Enter Email Address..." required>
    </div>
    <div class="form-group mb-3">
        <input type="password"
name="password" class="form-control form-control-user shadow-sm"
        id="passwordInput"
placeholder="Password" required>
    </div>
    <div class="form-group">
        <div class="custom-control custom-
checkbox small">
            <input type="checkbox"
class="custom-control-input" id="rememberMe" name="remember">
            <label class="custom-control-
label" for="rememberMe">Remember Me</label>
        </div>
    </div>
    <input type="submit" value="Login"
class="btn btn-primary btn-user btn-block shadow">
    </form>
</div>
</div>
</div>

<!-- Bootstrap core JavaScript-->
<script
src="{{asset('assets/vendor/jquery/jquery.min.js')}}"></script>
<script
src="{{asset('assets/vendor/bootstrap/js/bootstrap.bundle.min.js')}}"></script>

<!-- Core plugin JavaScript-->
<script src="{{asset('assets/vendor/jquery-
easing/jquery.easing.min.js')}}"></script>

<!-- Custom scripts for all pages-->
<script src="{{asset('assets/js/sb-admin-
2.min.js')}}"></script>

```

```

    <!-- Remember Me Script -->
    <script>
        document.addEventListener("DOMContentLoaded", function ()
    {
        const emailInput =
document.getElementById("emailInput");
        const passwordInput =
document.getElementById("passwordInput");
        const rememberMe =
document.getElementById("rememberMe");

        // Load saved credentials
        const savedEmail =
localStorage.getItem("rememberedEmail");
        const savedPassword =
localStorage.getItem("rememberedPassword");
        const savedRemember =
localStorage.getItem("rememberMe");

        if (savedRemember === "true" && savedEmail &&
savedPassword) {
            emailInput.value = savedEmail;
            passwordInput.value = savedPassword;
            rememberMe.checked = true;
        }

        // Save credentials on form submit

document.querySelector("form.user").addEventListener("submit",
function () {
            if (rememberMe.checked) {
                localStorage.setItem("rememberedEmail",
emailInput.value);
                localStorage.setItem("rememberedPassword",
passwordInput.value);
                localStorage.setItem("rememberMe", "true");
            } else {
                localStorage.removeItem("rememberedEmail");
                localStorage.removeItem("rememberedPassword");
                localStorage.removeItem("rememberMe");
            }
        });
    });
    </script>

</body>

</html>

```

Lampiran 5 Coding dashboard.blade.php (Admin)

```
@extends('backend/layout/main-admin')

@section('title', 'Dashboard Admin')

@section('content')
<style>
    .card-summary {
        height: 100px;
        display: flex;
        justify-content: center;
        align-items: center;
        font-weight: bold;
        font-size: 20px;
        color: white;
        border-radius: 12px;
    }

    .card-blue {
        background-color: #1f3a93;
    }

    .card-red {
        background-color: #e74c3c;
    }

    .table thead {
        background-color: #f8f9fa;
        font-weight: bold;
    }
</style>

<div class="container mt-4">

    <h3 class="text-center mb-4">Dashboard Admin</h3>

    <div class="row mb-4 justify-content-center">

        <div class="col-md-3">
            <div class="card card-summary card-blue shadow text-center">
                <div>
                    <div>Total Peserta</div>
                    <div style="font-size: 28px;">{{ $totalPeserta
                }}</div>
            </div>
        </div>

        <div class="col-md-3">
            <div class="card card-summary card-red shadow text-center">
                <div>
                    <div>Pembayaran Tertunda</div>
                </div>
            </div>
        </div>
    </div>
</div>
```



```


{{
$jumlahTertunda }}</div>
</div>
</div>
</div>

</div>

{{-- Tampilkan total tertunda per kategori --}}
<div class="card shadow mb-4">
  <div class="card-header bg-light font-weight-bold">
    Pembayaran Tertunda per Kategori
  </div>
  <div class="card-body table-responsive">
    @if ($tertundaPerKategori->isEmpty())
      <p class="text-center">Belum ada data pembayaran
tertunda per kategori.</p>
    @else
      <table class="table table-bordered">
        <thead>
          <tr>
            <th>Kategori</th>
            <th>Total Pembayaran Tertunda (Rp)</th>
          </tr>
        </thead>
        <tbody>
          @foreach ($tertundaPerKategori as $item)
            <tr>
              <td>{{ $item->kategori }}</td>
              <td>{{ number_format($item-
>total_tertunda, 0, ',', '.') }}</td>
            </tr>
          @endforeach
        </tbody>
      </table>
    @endif
  </div>
</div>

<div class="card shadow">
  <div class="card-header bg-light font-weight-bold">
    Laporan
  </div>
  <div class="card-body table-responsive">
    <table class="table table-bordered">
      <thead>
        <tr>
          <th>No</th>
          <th>Nama</th>
          <th>Status Pembayaran</th>
          <th>Metode Pembayaran</th>
          <th>Waktu Pembayaran</th>
        </tr>
      </thead>
      <tbody>
        @forelse ($laporanTerbaru as $index => $lapor)


```


Lampiran 6 Coding dashboard.blade.php (Peserta)

```
@extends('backend/layout/main-peserta')

@section('title', 'Dashboard Peserta')

@section('content')
<style>
    .hero-section {
        position: relative;
        height: 300px;
        background: url('{{ asset('assets/img/klenteng1.jpg') }}')
center center/cover no-repeat;
        background-color: #333;
        border-radius: 10px;
        overflow: hidden;
        margin-bottom: 30px;
    }

    .hero-overlay {
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        background-color: rgba(0, 0, 0, 0.5);
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        color: white;
        padding: 20px;
    }

    .hero-overlay h1 {
        font-size: 2rem;
        font-weight: bold;
        text-shadow: 1px 1px 4px rgba(0, 0, 0, 0.6);
    }

    @media (min-width: 768px) {
        .hero-overlay h1 {
            font-size: 3rem;
        }
    }

    .card-custom {
        border: 1px solid #ccc;
        border-radius: 6px;
        height: 100%;
    }

    .card-header-custom {
        background-color: #eee;
        padding: 10px;
    }
```

```

        font-weight: bold;
        text-align: center;
        border-bottom: 1px solid #ccc;
    }

    .badge-status {
        padding: 2px 8px;
        border-radius: 4px;
        font-size: 12px;
        background-color: green;
        color: white;
        vertical-align: middle;
    }

    .total-setoran {
        font-size: 28px;
        font-weight: bold;
        color: #2f55d4;
    }
}
</style>

<div class="container mt-4">

    {{-- Header Gambar --}}
    <div class="hero-section">
        <div class="hero-overlay">
            <div class="hero-text">
                <h1 class="mb-2">Iuran Dana Duka</h1>
                <h1>Yayasan Tri Dharma Kota Tegal</h1>
            </div>
        </div>
    </div>

    {{-- Isi 3 kolom dashboard --}}
    <div class="row mb-4 justify-content-center">

        <!-- Informasi Peserta -->
        <div class="col-md-4 mb-3">
            <div class="card card-custom">
                <div class="card-header-custom">Informasi
Peserta</div>
                <div class="card-body">
                    @php
                        $peserta = Auth::guard('peserta')->user()-
>peserta ?? null;
                    @endphp

                    <p class="mb-1">
                        <strong>{{ $peserta->name ?? 'Peserta'
}}</strong>
                        @if(isset($peserta->status) &&
strtolower($peserta->status) === 'aktif')
                            <span class="badge-
status">aktif</span>
                        @endif
                    </p>
                </div>
            </div>
        </div>
    </div>

```

```

        <p class="mb-1"><strong>Alamat:</strong> {{
$peserta->alamat ?? '-' }}</p>
        <p class="mb-1"><strong>No. Telepon:</strong>
{{ $peserta->telepon ?? '-' }}</p>
        <p class="mb-1"><strong>Kategori:</strong> {{
$peserta->kategori ?? '-' }}</p>
    </div>
</div>
</div>

<!-- Tagihan Bulan Ini -->
<div class="col-md-4 mb-3">
    <div class="card card-custom">
        <div class="card-header-custom">Tagihan Bulan
Ini</div>
        <div class="card-body">
            @if($tagihanBulanIni)
                <p class="mb-1"><strong>Rp {{
number_format($tagihanBulanIni->besaran_tagihan, 0, ',', '.')
}}</strong></p>
                <p class="mb-1"><strong>Status:</strong>
                @if($tagihanBulanIni->status ===
'belum_dibayar')
                    <span class="text-danger">Belum
Dibayar</span>
                @else
                    <span class="text-success">Sudah
Dibayar</span>
                @endif
            </p>
            <p class="mb-0"><strong>Jatuh
Tempo:</strong> {{ \Carbon\Carbon::parse($tagihanBulanIni-
>jatuh_tempo)->format('d F Y') }}</p>
            @else
                <p class="mb-1"><strong>Tidak ada tagihan
bulan ini</strong></p>
            @endif
        </div>
    </div>
</div>

<!-- Jumlah Total Setoran -->
<div class="col-md-4 mb-3">
    <div class="card card-custom">
        <div class="card-header-custom">Jumlah Total
Setoran</div>
        <div class="card-body text-center d-flex flex-
column justify-content-center">
            <p class="mb-2">Total iuran yang sudah
dibayarkan</p>
            <div class="total-setoran">
                Rp {{ number_format($totalSetoran, 0, ',',
'.' ) }}
            </div>
        </div>
    </div>
</div>

```

```

        </div>

</div>

{{-- Informasi Tambahan Lebar Penuh --}}
<div class="row mb-4">
    <div class="col-md-12">
        <div class="card card-custom">
            <div class="card-header-custom">Informasi</div>
            <div class="card-body">
                <ol class="pl-3 mb-0">
                    <li>Status akun akan nonaktif sementara
jika tagihan iuran terlambat dibayar selama 2 bulan berturut-
turut.</li>
                    <li>Jika ada kendala dalam pembayaran atau
pertanyaan terkait iuran, silakan hubungi admin yayasan untuk
bantuan lebih lanjut.</li>
                </ol>
            </div>
        </div>
    </div>
</div>

</div>
@endsection

```

Lampiran 7 Coding DashboardController.php (Admin)

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\View\View;
use App\Models\Peserta;
use App\Models\Tagihan;
use App\Models\RiwayatIuran;
use Illuminate\Support\Facades\DB;

class DashboardController extends Controller
{
    public function index(): View
    {
        $totalPeserta = Peserta::count();

        $jumlahTertunda = Tagihan::where('status',
        'belum_dibayar')->count();

        $laporanTerbaru =
        RiwayatIuran::orderBy('waktu_pembayaran', 'desc')->take(5)->get();

        $tertundaPerKategori = \DB::table('tagihans')
            ->join('pesertas', 'tagihans.user_id', '=',
            'pesertas.user_id')
            ->select('pesertas.kategori',
            \DB::raw('SUM(tagihans.besaran_tagihan) as total_tertunda'))
            ->where('tagihans.status', 'belum_dibayar')
            ->groupBy('pesertas.kategori')
            ->get();

        return view('backend.content.admin.dashboard',
        compact('totalPeserta', 'jumlahTertunda', 'laporanTerbaru',
        'tertundaPerKategori'));
    }

    public function profil(): View
    {
        return view('backend.content.admin.profil');
    }
}
```

Lampiran 8 Coding DashboardController.php (Peserta)

```
<?php

namespace App\Http\Controllers\Peserta;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\View\View;
use App\Models\Tagihan;
use App\Models\RiwayatIuran;
use Carbon\Carbon;

class DashboardController extends Controller
{
    // Halaman dashboard peserta
    public function index(): View
    {
        $user = Auth::guard('peserta')->user();

        // Ambil tagihan bulan ini yang belum dibayar untuk
peserta ini
        $tagihanBulanIni = Tagihan::where('user_id', $user->id)
            ->where('status', 'belum_dibayar')
            ->whereYear('bulan_tagihan', Carbon::now()->year)
            ->whereMonth('bulan_tagihan', Carbon::now()->month)
            ->first();

        // Hitung jumlah total setoran peserta
        $totalSetoran = RiwayatIuran::with('tagihan')
            ->where('user_id', $user->id)
            ->where('status_pembayaran', 'berhasil')
            ->get()
            ->sum(function ($riwayat) {
                return $riwayat->tagihan->besaran_tagihan ?? 0;
            });

        return view('backend.content.peserta.dashboard',
compact('user', 'tagihanBulanIni', 'totalSetoran'));
    }

    // Halaman profil peserta
    public function profil(): View
    {
        $user = Auth::guard('peserta')->user();
        $peserta = $user->peserta;

        return view('backend.content.peserta.profil',
compact('user', 'peserta'));
    }

    // Halaman edit profil peserta
    public function editProfil(): View
    {

```



```

        $user = Auth::guard('peserta')->user();
        $peserta = $user->peserta;

        return view('backend.content.peserta.edit-profil',
compact('peserta'));
    }

    // Update data profil peserta
    public function updateProfil(Request $request)
    {
        $user = Auth::guard('peserta')->user();
        $peserta = $user->peserta;

        $request->validate([
            'name'      => 'required|string|max:255',
            'nik'       => 'required|string',
            'alamat'    => 'required|string',
            'telepon'   => 'required|string',
            'kategori'  => 'required|string',
        ]);

        $peserta->update($request->only(['name', 'nik', 'alamat',
'telepon', 'kategori']));

        return redirect()->route('peserta.profil')-
>with('success', 'Profil berhasil diperbarui.');
```

Lampiran 9 Coding TagihanController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Tagihan;
use App\Models\User;

class TagihanController extends Controller
{
    public function index(Request $request)
    {
        // Ambil data tagihan beserta relasi user/peserta
        $query = Tagihan::with('peserta');

        // Filter berdasarkan bulan_tagihan (format: YYYY-MM)
        if ($request->filled('bulan')) {
            $query->whereMonth('bulan_tagihan', date('m',
                strtotime($request->bulan)))
                ->whereYear('bulan_tagihan', date('Y',
                strtotime($request->bulan)));
        }

        // Filter status (sudah_dibayar/belum_dibayar)
        if ($request->filled('status')) {
            $query->where('status', $request->status);
        }

        // Filter berdasarkan nama peserta
        if ($request->filled('search')) {
            $query->whereHas('peserta', function ($q) use
                ($request) {
                    $q->where('name', 'like', '%' . $request->search .
                    '%');
                });
        }

        // Ambil hasil query
        $tagihans = $query->get();

        // Kirim ke view
        return view('backend.content.admin.tagihan.tagihan',
            compact('tagihans'));
    }
}
```

Lampiran 10 Coding LaporanController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\RiwayatIuran;

class LaporanController extends Controller
{
    public function index(Request $request)
    {
        $query = RiwayatIuran::with('tagihan');

        if ($request->filled('periode')) {
            $query->whereYear('waktu_pembayaran', substr($request->periode, 0, 4))
                ->whereMonth('waktu_pembayaran',
substr($request->periode, 5, 2));
        }

        if ($request->filled('search')) {
            $query->where('created_by', 'like', '%' . $request->search . '%');
        }

        $laporan = $query->orderBy('waktu_pembayaran', 'desc')->get();

        $periodes =
RiwayatIuran::selectRaw("DATE_FORMAT(waktu_pembayaran, '%Y-%m') as
periode")
            ->whereNotNull('waktu_pembayaran')
            ->distinct()
            ->orderBy('periode', 'desc')
            ->pluck('periode');

        return view('backend.content.admin.laporan-riwayat',
compact('laporan', 'periodes'));
    }

    public function cetak(Request $request)
    {
        $query = RiwayatIuran::with('tagihan');

        if ($request->filled('periode')) {
            $query->whereYear('waktu_pembayaran', substr($request->periode, 0, 4))
                ->whereMonth('waktu_pembayaran',
substr($request->periode, 5, 2));
        }

        if ($request->filled('search')) {
```

```

        $query->where('created_by', 'like', '%' . $request-
>search . '%');
    }

    $laporan = $query->orderBy('waktu_pembayaran', 'desc')-
>get();

    $totalPemasukan = $laporan->sum(function ($item) {
        return $item->>tagihan->besaran_tagihan ?? 0;
    });

    $periodeText = 'Semua Periode';

    if ($request->filled('periode')) {
        $periodeText = \Carbon\Carbon::createFromFormat('Y-m',
$request->periode)
            ->timezone('Asia/Jakarta')
            ->translatedFormat('F Y');
    }

    return view('backend.content.admin.cetak-laporan-riwayat',
compact(
        'laporan',
        'totalPemasukan',
        'periodeText'
    ));
}
}

```

Lampiran 11 Coding PembayaranController.php

```
<?php

namespace App\Http\Controllers\Peserta;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use App\Models\Tagihan;
use App\Models\Pembayaran;
use Carbon\Carbon;
use Midtrans\Snap;
use Midtrans\Config;

class PembayaranController extends Controller
{
    public function index()
    {
        $user = Auth::guard('peserta')->user();

        $tagihanList = Tagihan::where('user_id', $user->id)-
>get();
        $pembayaranList = Pembayaran::where('user_id', $user->id)-
>get();

        return view('backend.content.peserta.pembayaran',
compact('tagihanList', 'pembayaranList'));
    }

    public function bayar($id)
    {
        $user = Auth::guard('peserta')->user();

        $tagihan = Tagihan::where('id', $id)->where('user_id',
$user->id)->firstOrFail();

        if ($tagihan->status == 'sudah_dibayar') {
            return redirect()->route('peserta.pembayaran.index')-
>with('error', 'Tagihan sudah dibayar.');
```

```

        'customer_details' => [
            'first_name' => $user->name,
            'email' => $user->email,
        ]
    ];

    $snapToken = Snap::getSnapToken($params);

    return view('backend.content.peserta.snap',
compact('snapToken', 'tagihan'));
    }

    public function createSnapToken(Request $request)
    {
        $user = Auth::guard('peserta')->user();
        $tagihan = Tagihan::where('id', $request->tagihan_id)-
>where('user_id', $user->id)->firstOrFail();

        Config::$serverKey = config('midtrans.server_key');
        Config::$isProduction = config('midtrans.is_production');
        Config::$isSanitized = config('midtrans.is_sanitized');
        Config::$is3ds = config('midtrans.is_3ds');

        $params = [
            'transaction_details' => [
                'order_id' => 'TAGIHAN-' . $tagihan->id . '-' .
time(),
                'gross_amount' => $tagihan->besaran_tagihan,
            ],
            'customer_details' => [
                'first_name' => $user->name,
                'email' => $user->email,
            ]
        ];

        try {
            $snapToken = Snap::getSnapToken($params);
            return response()->json(['token' => $snapToken]);
        } catch (\Exception $e) {
            \Log::error('Gagal membuat Snap Token: ' . $e-
>getMessage());
            return response()->json(['error' => 'Gagal membuat
token: ' . $e->getMessage()], 500);
        }
    }

    public function notificationHandler(Request $request)
    {
        $payload = json_decode($request->getContent(), true);

        \Log::info('NOTIFIKASI DITERIMA:', $payload);

        $orderId = $payload['order_id'] ?? null;
        $transactionStatus = $payload['transaction_status'] ??
null;
    }

```

```

        if (!$orderId || !$transactionStatus) {
            \Log::warning('Payload tidak lengkap: order_id atau
transaction_status null');
            return response()->json(['message' => 'Invalid
payload'], 400);
        }

        // Struktur TAGIHAN-16-123456789
        $explode = explode('-', $orderId);
        if (count($explode) < 2) {
            \Log::warning("Format order_id tidak sesuai:
$orderId");
            return response()->json(['message' => 'Invalid order
ID format'], 400);
        }

        $tagihanId = $explode[1];

        $tagihan = Tagihan::find($tagihanId);

        if (!$tagihan) {
            \Log::warning("Tagihan ID $tagihanId tidak
ditemukan");
            return response()->json(['message' => 'Tagihan not
found'], 404);
        }

        if (in_array($transactionStatus, ['settlement',
'capture'])) {
            $tagihan->update(['status' => 'sudah_dibayar']);

            Pembayaran::updateOrCreate(
                [
                    'tagihan_id' => $tagihan->id,
                    'user_id' => $tagihan->user_id
                ],
                [
                    'nominal' => $tagihan->besaran_tagihan,
                    'status' => 'Lunas',
                    'tanggal_bayar' => now(),
                ]
            );

            \Log::info("Tagihan ID $tagihanId berhasil diupdate
jadi 'sudah_dibayar'");
        } else {
            \Log::info("Transaksi dengan status $transactionStatus
tidak diproses (hanya settlement/capture)");
        }

        return response()->json(['message' => 'Notification
received']);
    }
}

```

Lampiran 12 Coding RiwayatPembayaranController.php

```
<?php

namespace App\Http\Controllers\Peserta;

use App\Http\Controllers\Controller;
use App\Models\RiwayatIuran;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class RiwayatPembayaranController extends Controller
{
    public function index(Request $request)
    {
        // Mendapatkan data peserta yang sedang login
        $user = Auth::guard('peserta')->user();

        // Query awal: hanya riwayat milik user login
        $query = RiwayatIuran::with('tagihan')
            ->where('user_id', $user->id)
            ->orderBy('waktu_pembayaran', 'desc');

        // Jika ada pencarian
        if ($request->has('search') && $request->search != '') {
            $search = $request->search;
            $query->where(function ($q) use ($search) {
                $q->where('metode_pembayaran', 'like',
                    "%$search%")
                    ->orWhere('status_pembayaran', 'like',
                    "%$search%")
                    ->orWhere('id_tagihan', 'like', "%$search%");
            });
        }

        // Total setoran peserta yang login
        $totalSetoran = RiwayatIuran::with('tagihan')
            ->where('user_id', $user->id)
            ->where('status_pembayaran', 'berhasil')
            ->get()
            ->sum(function ($riwayat) {
                return $riwayat->tagihan->besaran_tagihan ?? 0;
            });

        // Ambil data dengan pagination
        $riwayats = $query->paginate(10);

        // Kirim data ke view
        return view('backend.content.peserta.riwayat-pembayaran',
            compact('riwayats', 'totalSetoran'));
    }

    public function cetak($id)
    {
        // Mendapatkan data peserta yang sedang login
    }
}
```



```

        $user = Auth::guard('peserta')->user();

        // Ambil 1 data riwayat pembayaran sesuai ID dan milik
user login
        $riwayat = RiwayatIuran::with('tagihan')
            ->where('id', $id)
            ->where('user_id', $user->id)
            ->firstOrFail();

        // Ambil data peserta
        $peserta = $user->peserta;

        // Kirim ke halaman cetak
        return view('backend.content.peserta.cetak-riwayat-
pembayaran', compact('riwayat', 'peserta', 'user'));
    }
}

```