

LAMPIRAN

Lampiran 1 Surat Ketersediaan Membimbing TA

SURAT KETERSEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Miftakhul Huda, M.Kom
NIDN : 0620127801
NIPY : 04.007.033
Jabatan Struktural : Dosen Program Studi DIII Teknik Komputer
Jabatan Fungsional : Asisten Ahli

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

Nama : Yuli Susilowati
NIM : 22041002
Program Studi : Diploma III Teknik Komputer

Judul TA : SISTEM INFORMASI PEMESANAN TIKET BUS PADA AGEN PO
HIBA GROUP BERBASIS WEBSITE

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, Februari 2025

Mengetahui,
Kepala Program Studi DIII Teknik Komputer



Ida Afriliana, S.T, M.Kom
NIPY. 12.013.168

Dosen Pembimbing I

Miftakhul Huda, M.Kom
NIPY. 04.007.033

SURAT KETERSEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini:

Nama : Yerry Febrian Sabanise, M.Kom
NIDN : 0613028602
NIPY : 03.012.110
Jabatan Struktural : Dosen Program Studi DIII Teknik Komputer
Jabatan Fungsional : Asisten Ahli

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

Nama : Yuli Susilowati
NIM : 22041002
Program Studi : Diploma III Teknik Komputer

Judul TA : SISTEM INFORMASI PEMESANAN TIKET BUS PADA AGEN PO
HIBA GROUP BERBASIS WEBSITE

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, Februari 2025

Mengetahui,
Kepala Program Studi DIII Teknik Komputer



Ida Afrilliana, S.T, M.Kom
NIPY. 12.013.168

Dosen Pembimbing II

Yerry Febrian Sabanise, M.Kom
NIPY. 03.012.110

Lampiran 2 Surat Permohonan Izin Observasi Tugas Akhir (TA)

		POLITEKNIK HARAPAN BERSAMA The True Vocational Campus		D-3 Teknik Komputer	
Nomor	:	428.03/KOM-PHB/VI/2025			
Lampiran	:	-			
Perihal	:	Permohonan Izin Observasi Tugas Akhir (TA)			
 Kepada Yth.					
Pimpinan Agen PO Hiba Group					
Di Ds suradadi kec kramat kab tegal					
 Dengan Hormat,					
Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester VI Program Studi Diploma III Teknik Komputer Politeknik Harapan Bersama Tegal, Maka dengan ini kami mengajukan izin observasi pengambilan data di Agen PO Hiba Group yang Bapak / Ibu Pimpin, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:					
No	NIM	NAMA	NO HP		
1	22041002	YULI SUSILOWATI	085327053592		
 Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.					
 Tegal, 30 Juni 2025 Ka. Prodi DIII Teknik Komputer Politeknik Harapan Bersama Tegal					
 Ida Afrilliana, ST, M.Kom NIPY. 12.013.168					
  Jl. Mataram No. 9 Kota Tegal 52143, Jawa Tengah, Indonesia.  02831352000					
 komputer@politektegal.ac.id  politektegal.ac.id					

Lampiran 3 Dokumentasi



Lampiran 4 Program

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');
header('Access-Control-Allow-Origin:');
header("Access-Control-Allow-Methods:GET,OPTIONS");
class Frontend extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->ul = $this->uri->segment(1);
        if (empty($this->session->userdata('log_user')) {
            $this->dt_user = null;
        } else {
            $this->db->where('id', $this->session->
            >userdata('id'));
            $this->dt_user = $this->db->get('user')->row();
            $params = array('server_key' => 'SB-Mid-server-
            WT6opkjXfDqTN3zElhbf4NHI', 'production' => false);
            $this->load->library('midtrans');
            $this->midtrans->config($params);
        }
        $this->load->model('M_Front', 'front');
    }
    private function _authentication()
    {
        if (empty($this->session->userdata('log_user')) {
            $this->session->set_flashdata('toastr-error',
            'Silahkan login dahulu!');
            redirect('auth', 'refresh');
        }
    }
    public function index()
    {
        $data = [
            'title' => 'Home',
            'page' => 'frontend/v_home',
            'kota' => $this->front->getKota()
        ];
        $this->load->view('frontend/index', $data);
    }
    public function getTerminal()
    {
        $idKota = $this->input->post('idKota');
        $terminal = $this->front->getTerminal(['idKota' =>
        $idKota]);
        if ($terminal) {
            $res = [
```

```

        'status' => true,
        'message' => 'Terminal found',
        'data'    => $terminal
    ];
} else {
    $res = [
        'status' => false,
        'message' => 'Terminal not found'
    ];
}
echo json_encode($res);
}
public function getJam()
{
    $idBus      = $this->input->post('idBus');
    $idTerminal = $this->input->post('idTerminal');
    $tanggal    = $this->input->post('tanggal');
    if (empty($idBus) || empty($idTerminal) ||
empty($tanggal)) {
        echo json_encode(['status' => false, 'message' =>
'Invalid input']);
        return;
    }
    $jadwal = $this->front->getJadwalBus($idBus, $idTerminal,
$tanggal);
    if (!empty($jadwal)) {
        echo json_encode([
            'status' => true,
            'message' => 'Schedule found',
            'data'    => $jadwal
        ]);
    } else {
        echo json_encode([
            'status' => false,
            'message' => 'No schedule found'
        ]);
    }
}
public function placeOrder()
{
    $this->_authentication();
    $this->form_validation->set_rules('kota',      'Destination
City', 'required');
    $this->form_validation->set_rules('tanggal',    'Departure
Date', 'required');
    $this->form_validation->set_rules('jumlah',     'Quantity',
'required|numeric');
    if ($this->form_validation->run() == FALSE) {
        $this->session->set_flashdata('toastr-error',
validation_errors());
    }
}

```

```

        redirect($_SERVER['HTTP_REFERER'], 'refresh');
    }
    $idKota = $this->input->post('kota');
    $tanggal = $this->input->post('tanggal');
    $jumlah = $this->input->post('jumlah');
    $data = [
        'group_id' => group_id(),
        'idUser' => $this->dt_user->id,
        'idKota' => $idKota,
        'tanggal' => $tanggal,
        'jumlah' => $jumlah
    ];
    $cek = $this->db->get_where('orders', [
        'idUser' => $this->dt_user->id,
        'status_code' => null
    ])->row();
    if ($cek) {
        $data['group_id'] = $cek->group_id;
        $this->db->where('id', $cek->id);
        $proses = $this->db->update('orders', $data);
    } else {
        $proses = $this->db->insert('orders', $data);
    }
    if ($proses) {
        $this->session->set_flashdata('toastr-success', 'Order
success');
        redirect('search/' . $data['group_id'], 'refresh');
    } else {
        $this->session->set_flashdata('toastr-error', 'Order
failed');
        redirect($_SERVER['HTTP_REFERER'], 'refresh');
    }
}
public function search($group_id)
{
    $this->_authentication();
    $this->db->where('group_id', $group_id);
    $order = $this->db->get('orders')->row();
    $this->_cekGroupId($order);

    $this->db->select('jadwal.*, bus.namaBus, bus.kelas,
terminal.namaTerminal, kota.namaKota');
    $this->db->join('bus', 'jadwal.idBus = bus.id', 'inner');
    $this->db->join('terminal', 'jadwal.idTerminal =
terminal.id', 'inner');
    $this->db->join('kota', 'kota.id = terminal.idKota',
'inner');
    $this->db->where('terminal.idKota', $order->idKota);
    $this->db->group_start();
    $this->db->where("('$order->tanggal' BETWEEN tanggal_mulai
AND tanggal_akhir)

```

```

        OR ('$order->tanggal' BETWEEN tanggal_mulai AND
tanggal_akhir)
        OR (tanggal_mulai BETWEEN '$order->tanggal' AND
'$order->tanggal')", null, false);
        $this->db->group_end();
        $this->db->order_by('jadwal.tanggal_mulai,        jadwal.jam,
terminal.namaTerminal, bus.namaBus', 'desc');
        $jadwal = $this->db->get('jadwal')->result();
        $data = [
            'title' => 'Search Ticket',
            'page'   => 'frontend/v_search',
            'kota'   => $this->front->getKota(),
            'order'  => $order,
            'jadwal' => $jadwal
        ];
        $this->load->view('frontend/index', $data);
    }
    public function updateJadwalOrder()
    {
        $this->_authentication();
        $data = [
            'idKota' => $this->input->post('idKota'),
            'tanggal' => $this->input->post('tanggal')
        ];
        $this->db->where('group_id',                $this->input-
>post('group_id'));
        $this->db->update('orders', $data);
        redirect('search/' . $this->input->post('group_id'),
'refresh');
    }

    public function booking_detail($group_id, $idJadwal)
    {
        $this->_authentication();
        $this->db->select('orders.*,        jadwal.jam,        jadwal.harga,
terminal.namaTerminal, kota.namaKota, jadwal.idBus, bus.namaBus,
bus.kelas, bus.seat');
        $this->db->join('jadwal', 'orders.idJadwal = jadwal.id',
'left');
        $this->db->join('terminal', 'jadwal.idTerminal        =
terminal.id', 'left');
        $this->db->join('kota', 'kota.id        =        terminal.idKota',
'left');
        $this->db->join('bus', 'jadwal.idBus = bus.id', 'left');
        $order = $this->db->get_where('orders', ['orders.group_id'
=> $group_id])->row();
        $this->_cekGroupId($order);
        if ($order->idJadwal != $idJadwal) {

```

```

        $jadwal = $this->db->get_where('jadwal', ['id' =>
$idJadwal])->row();
        $dt = [
            'idJadwal' => $idJadwal,
            'totalHarga' => $jadwal->harga * $order->jumlah
        ];
        $this->db->where('group_id', $group_id);
        $update = $this->db->update('orders', $dt);
        if ($update) {
            $this->db->select('orders.*,          jadwal.jam,
jadwal.harga, terminal.namaTerminal, kota.namaKota, jadwal.idBus,
bus.namaBus, bus.kelas, bus.seat');
            $this->db->join('jadwal', 'orders.idJadwal =
jadwal.id', 'inner');
            $this->db->join('terminal', 'jadwal.idTerminal =
terminal.id', 'inner');
            $this->db->join('kota', 'kota.id =
terminal.idKota', 'inner');
            $this->db->join('bus', 'jadwal.idBus = bus.id',
'inner');
            $order = $this->db->get_where('orders',
['orders.group_id' => $group_id])->row();
            $this->db->where('group_id', $group_id);
            $delete = $this->db->delete('penumpang');
            if ($delete) {
                foreach (range(1, $order->jumlah) as $jumlah)
                {
                    $data = [
                        'id' => date('ymdhis') .
$jumlah,
                        'idUser' => $this->dt_user->id,
                        'group_id' => $group_id
                    ];
                    $this->db->insert('penumpang', $data);
                }
            }
        } else {
            $jadwal = $this->db->get_where('jadwal', ['id' =>
$idJadwal])->row();
            $dt = [
                'totalHarga' => $jadwal->harga * $order->jumlah
            ];
            $this->db->where('group_id', $group_id);
            $update = $this->db->update('orders', $dt);
        }
        $this->db->order_by('id', 'asc');
        $penumpang = $this->db->get_where('penumpang', ['group_id'
=> $group_id])->result();

```

```

        $data = [
            'title'      => 'Booking Detail',
            'page'        => 'frontend/v_bookingDetail',
            'order'       => $order,
            'penumpang'  => $penumpang
        ];
        $this->load->view('frontend/index', $data);
    }
    public function updateCustomer()
    {
        $this->_authentication();
        $data = [
            'nama'    => $this->input->post('nama'),
            'email'   => $this->input->post('email'),
            'no_hp'   => $this->input->post('no_hp')
        ];
        $this->db->where('group_id', $this->input->
>post('group_id'));
        $update = $this->db->update('orders', $data);
        if ($update) {
            $res = [
                'status' => true
            ];
        } else {
            $res = [
                'status' => false
            ];
        }
        echo json_encode($res);
    }
    public function updatePassenger()
    {
        $this->_authentication();
        $data = [
            'nama'    => $this->input->post('nama'),
            'no_hp'   => $this->input->post('no_hp')
        ];
        $this->db->where('id', $this->input->post('id'));
        $update = $this->db->update('penumpang', $data);
        if ($update) {
            $res = [
                'status' => true
            ];
        } else {
            $res = [
                'status' => false
            ];
        }
        echo json_encode($res);
    }
    public function updateSeat()
    {
        $this->_authentication();
        $data = [
            'no_kursi' => $this->input->post('no_kursi')
        ];
    }

```

```

        $this->db->where('id', $this->input->post('id'));
        $update = $this->db->update('penumpang', $data);
        if ($update) {
            $res = [
                'status' => true
            ];
        } else {
            $res = [
                'status' => false
            ];
        }
        echo json_encode($res);
    }
    public function getSeat()
    {
        $idJadwal = $this->input->post('idJadwal');
        $tanggal = $this->input->post('tanggal');
        $id = $this->input->post('id');
        $seat = $this->front->getSeat([
            'orders.idJadwal' => $idJadwal,
            'orders.tanggal' => $tanggal,
            'penumpang.id !=' => $id,
            'penumpang.no_kursi !=' => null
        ]);
        if (!empty($seat)) {
            echo json_encode([
                'status' => true,
                'message' => 'Seat found',
                'data' => $seat
            ]);
        } else {
            echo json_encode([
                'status' => false,
                'message' => 'No seat found'
            ]);
        }
    }
    private function _cekGroupId($order)
    {
        if (empty($order)) {
            $this->session->set_flashdata('toastr-error', 'Order
not found');
            redirect('home', 'refresh');
        }
        if ($order->status_code != null) {
            $this->session->set_flashdata('toastr-error', 'Order
has been processed');
            redirect('orders', 'refresh');
        }
    }
    public function orders()
    {
        $this->_authentication();
        $data = [
            'title' => 'List Orders',
            'page' => 'frontend/v_orders',

```

```

        'orders' => $this->front->getAllOrders([
            'orders.idUser' => $this->dt_user->id
        ])
    ];
    $this->load->view('frontend/index', $data);
}
public function detail($id)
{
    $this->_authentication();
    $order = $this->front->getAllOrders([
        'orders.id' => $id
    ]);
    if ($order) {
        $order = $order[0];
        $this->db->order_by('id', 'asc');
        $penumpang = $this->db->get_where('penumpang',
['group_id' => $order->group_id])->result();
    } else {
        $this->session->set_flashdata('toastr-error', 'Order
not found');
        redirect('orders', 'refresh');
    }
    $data = [
        'title'      => 'Order Detail',
        'page'       => 'frontend/v_detail',
        'order'      => $order,
        'penumpang' => $penumpang
    ];
    $this->load->view('frontend/index', $data);
}
public function cancel($group_id)
{
    $this->_authentication();
    $this->db->where('group_id', $group_id);
    $delete = $this->db->delete('orders');

    if ($delete) {
        $this->db->where('group_id', $group_id);
        $this->db->delete('penumpang');
        $this->session->set_flashdata('toastr-success', 'Order
has been canceled');
    } else {
        $this->session->set_flashdata('toastr-error', 'Order
has not been canceled');
    }
    redirect('orders', 'refresh');
}
public function pay()
{
    $this->_authentication();
    $group_id = $this->input->post('group_id');
    $result = json_decode($this->input->post('result_data'),
true);
    if ($result['payment_type'] == 'bank_transfer') {
        $dt = [
            'order_id'      => $result['order_id'],

```

```

        'transaction_id' => $result['transaction_id'],
        'payment_type'   => $result['payment_type'],
        'transaction_time' => $result['transaction_time'],
        'bank'           =>
$result['va_numbers'][0]["bank"],
        'va_numbers'     =>
$result['va_numbers'][0]["va_number"],
        'pdf_url'        => $result['pdf_url'],
        'status_code'    => $result['status_code'],
    ];
    $this->db->where('group_id', $group_id);
    $update = $this->db->update('orders', $dt);
    if ($update) {
        $this->session->set_flashdata('toastr-success',
'The product has been successfully ordered');
    } else {
        $this->session->set_flashdata('toastr-error',
'Product failed to order!!');
    }
    } else {
        $this->session->set_flashdata('toastr-error', 'Please,
use the payment method by bank transfer only (BNI, BRI, BCA)');

        redirect($_SERVER['HTTP_REFERER'], 'refresh');
    }
    redirect('orders', 'refresh');
}
public function token()
{
    $group_id = $this->input->post('group_id');
    $order = $this->front->getAllOrders([
        'orders.group_id' => $group_id
    ]);
    $this->db->order_by('id', 'asc');
    $penumpang = $this->db->get_where('penumpang', ['group_id'
=> $group_id])->result();
    $order_id = $this->generateOrderID(5);
    $this->db->where('group_id', $group_id);
    $this->db->update('orders', ['order_id' => $order_id]);
    // Required
    $transaction_details = array(
        'order_id'      => $order_id,
        'gross_amount' => $order[0]->totalHarga,
    );
    $item_details = [];
    foreach ($penumpang as $p) {
        array_push($item_details, [
            'id'         => $p->id,
            'price'      => $order[0]->harga,
            'quantity'   => 1,
            'name'       => ' x ' . $order[0]->namaBus . ' - ' .
$order[0]->kelas
        ]);
    }
    $customer_details = array(
        'first_name' => $order[0]->nama,

```

```

        'phone'          => $order[0]->no_hp
    );
    $credit_card['secure'] = true;

    $time          = time();
    $custom_expiry = array(
        'start_time' => date("Y-m-d H:i:s O", $time),
        'unit'       => 'day',
        'duration'   => 1
    );
    $transaction_data = array(
        'transaction_details' => $transaction_details,
        'item_details'       => $item_details,
        'customer_details'   => $customer_details,
        'credit_card'        => $credit_card,
        'expiry'             => $custom_expiry
    );
    error_log(json_encode($transaction_data));
    $snapToken          = $this->midtrans-
>getSnapToken($transaction_data);
    error_log($snapToken);
    echo $snapToken;
}

public function validasi($id)
{
    $this->load->model('M_Pemesanan', 'pemesanan');
    $penumpang          = $this->pemesanan-
>getPenumpang(['penumpang.id' => $id]);
    $data = [
        'title'          => 'Ticket Validation',
        'page'           => 'frontend/v_validasi',
        'penumpang'      => $penumpang,
        'valid'          => ($penumpang) ? true : false
    ];
    $this->load->view('frontend/index', $data);
}

public function profile()
{
    $data = [
        'title' => 'Profile',
        'page'  => 'frontend/profile',
    ];
    $this->load->view('frontend/index', $data);
}

public function changeProfile()
{
    $img = $_FILES['image']['name'];
    if ($img) {
        $config['upload_path']   = 'upload/profile';
        $config['allowed_types'] = 'jpg|jpeg|png';
        $config['max_size']      = 2000;
        $config['remove_spaces'] = TRUE;
        $config['encrypt_name']  = TRUE;
        $this->load->library('upload', $config);
        $this->upload->initialize($config);
    }
}

```

```

        if (!$this->upload->do_upload('image')) {
            $this->session->set_flashdata('toastr-error',
$this->upload->display_errors());
            redirect('profile');
        } else {
            $upload_data = $this->upload->data();
            $previmage = $this->input->post('previmage');
            $data = [
                'name' => $this->input->post('name'),
                'noHp' => $this->input->post('noHp'),
                'alamat' => $this->input->post('alamat'),
                'image' => $upload_data['file_name']
            ];
            $this->db->where('id', $this->dt_user->id);
            $insert = $this->db->update('user', $data);
            if ($insert) {
                if ($previmage != 'default.jpg') {
                    unlink(FCPATH . 'upload/profile/' .
$previmage);
                }
                $this->session->set_flashdata('toastr-
success', 'success !');
                redirect('profile');
            } else {
                $this->session->set_flashdata('toastr-error',
'failed!');
                redirect('profile');
            }
        }
    } else {
        $data = [
            'name' => $this->input->post('name'),
            'noHp' => $this->input->post('noHp'),
            'alamat' => $this->input->post('alamat'),
        ];
        $this->db->where('id', $this->dt_user->id);
        $insert = $this->db->update('user', $data);
        if ($insert) {
            $this->session->set_flashdata('toastr-success',
'success !');
            redirect('profile');
        } else {
            $this->session->set_flashdata('toastr-error',
'failed!');
            redirect('profile');
        }
    }
}

public function changePassword()
{
    $current_password = $this->input-
>post('current_password');
    if (password_verify($current_password, $this->dt_user-
>password)) {
        $this->form_validation->set_rules('password', 'New
Password', 'trim|required|matches[retype_password]');
    }
}

```

```

        $this->form_validation->set_rules('retype_password',
'Retype New Password', 'trim|required|matches[password]');
        if ($this->form_validation->run() == TRUE) {
            $data = [
                'password' => password_hash($this->input-
>post('password'), PASSWORD_BCRYPT)
            ];
            $this->db->where('id', $this->dt_user->id);
            $this->db->update('user', $data);
            $this->session->set_flashdata('toastr-success',
'Success! Your Password Changed!');
            redirect('profile');
        } else {
            $this->session->set_flashdata('toastr-error',
'Failed! Password Dont Match!');
            redirect('profile');
        }
    } else {
        $this->session->set_flashdata('toastr-error', 'Failed!
Wrong Current Password');
        redirect('profile');
    }
}

private function _generateOrderID($length = 10)
{
    $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $charactersLength = strlen($characters);
    $randomString = '';
    for ($i = 0; $i < $length; $i++) {
        $randomString .= $characters[rand(0, $charactersLength
- 1)];
    }
    return $randomString . '-' . date('His');
}
}

```