

LAMPIRAN

Lampiran- 1 Aksi_User

```

<?php

session_start();

define('BASE_URL',
    (isset($_SERVER['HTTPS'])
    && $_SERVER['HTTPS'] ===
    'on' ? "https" : "http") .
    "://"
    $_SERVER['HTTP_HOST'] .
    "/klinikmuslimah/");

$conn = null;

try {

    $host = "localhost";

    $user = "root";

    $pass = "";

    $database =
        "db_klinik_muslimah";

    $db =
        "mysql:dbname=$database;h
        ost=$host";

    $conn = new PDO($db, $user,
        $pass);

    $conn->setAttribute(PDO::ATTR_E
        RRMODE,
        PDO::ERRMODE_EXCEPTION);

} catch (PDOException $e) {

    echo "Error !" . $e-
        >getMessage();

    die;

}

function
    compressImage($source,
        $destination, $quality)
{

    $imgInfo =

```

```

    getimagesize($source);

    $mime = $imgInfo['mime'];

    switch ($mime) {

        case 'image/jpeg':

            $image =
                imagecreatefromjpeg($sour
                    ce);

            break;

        case 'image/png':

            $image =
                imagecreatefrompng($sourc
                    e);

            break;

        case 'image/gif':

            $image =
                imagecreatefromgif($sourc
                    e);

            break;

        default:

            $image =
                imagecreatefromjpeg($sour
                    ce);

    }

    imagejpeg($image,
        $destination, $quality);

    return $destination;

}

function php_slug($b)

{

    $slug =
        preg_replace('/[^a-z0-
            9]+/', '-',
            trim(strtolower($b)));

    return $slug;
}

```

```

}

date_default_timezone_set("Asia/Makassar");

$aksi = $_GET['aksi'] ?? '';

if ($aksi == "") {
    $aksi = $_POST['aksi'] ?? '';
}

switch ($aksi) {
    case 'masuk':
        masuk($conn);
        break;

    case 'daftar':
        daftar($conn);
        break;

    case 'buatkonsultasi':
        buatkonsultasi($conn);
        break;

    // case 'hapuskonsultasi':
    //     hapuskonsultasi($conn);
    //     break;

    case 'keluar':
        keluar();
        break;
}

function masuk($conn)
{
    global $conn;

    $user = $_POST["emaillog"]
    ?? '';

    $pass = $_POST["passlog"]
    ?? '';

    $stmt = $conn->prepare("SELECT * FROM tb_pelanggan WHERE Email = :Email AND Pass = :Pass");

    $stmt->bindParam(':Email', $user);

    $stmt->bindParam(':Pass', $pass);

    $stmt->execute();

    if ($stmt->rowCount() > 0)
    {
        $row = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($pass != $row["Pass"]) {
            echo "Salah !";
        } else {

            $_SESSION["usernya"] = $row["Email"];

            $_SESSION["namanya"] = $row["Nama"];

            echo "Terimakasih !";
        }
    } else {
        echo "Maaf !";
    }
}

function daftar($conn)

```

```

{
    global $conn;

    $email =
    $_POST["emailregis"] ??
    '';

    $nohpwa =
    $_POST["nohpwaregis"] ??
    '';

    $pass =
    $_POST["passregis"] ??
    '';

    $nama =
    $_POST["namaregis"] ??
    '';

    $tgldaftar = date("d-m-Y
    H:i:s");

    $pelanggan = [
        'Email' => $email,
        'NoHP' => $nohpwa,
        'Pass' => $pass,
        'Nama' => $nama,
        'TglDaftar' =>
        $tgldaftar,
    ];

    $quercek = $conn-
    >prepare("SELECT * FROM
    tb_pelanggan WHERE
    Email=? OR NoHP=?");

    $quercek-
    >execute(array($email,
    $nohpwa));

    $cekdata = $quercek-
    >rowCount();

    if ($cekdata >= 1) {
        echo "Sudah Ada !";
    } else {
        $hasil = $conn-
        >prepare("INSERT INTO
        tb_pelanggan
        (Email,NoHP,Pass,Nama,Tgl
        Daftar) VALUES
        (:Email,:NoHP,:Pass,:Nama
        ,:TglDaftar)");

        $okok = $hasil-
        >execute($pelanggan);

        if ($okok) {
            echo "Terimakasih
            !";
        } else {
            echo "Maaf !";
        }
    }
}

function buatkonsultasi($conn)
{
    global $conn;

    $judul =
    $_POST["judulkonsul"] ??
    '';

    $keterangan =
    $_POST["keterangankonsul"
    ] ?? '';

    $tgl = date('Y-m-d
    H:i:s');

    $datakonsul = [
        'Email' =>
        $_SESSION["usernya"],
        'JudulKonsul' =>
        $judul,
        'IsiKonsul' =>
        $keterangan,
        'TanggalKonsul' =>

```

```

        $tgl,

    ];

    $hasil = $conn->pre...

Lampiran II Aksi-Admin

[8.05 AM, 16/2/2025] SANS.iD:
    <?php

session_start();

define('BASE_URL',
    (isset($_SERVER['HTTPS'])
    && $_SERVER['HTTPS'] ===
    'on' ? "https" : "http") .
    "://"
    $_SERVER['HTTP_HOST'] .
    "/klinikmuslimah/");

$conn = null;

try {

    $host = "localhost";

    $user = "root";

    $pass = "";

    $database =
        "db_klinik_muslimah";

    $db =
        "mysql:dbname=$database;h
        ost=$host";

    $conn = new PDO($db, $user,
        $pass);

    $conn->setAttribute(PDO::ATTR_E
        RRMODE,
        PDO::ERRMODE_EXCEPTION);

} catch (PDOException $e) {

    echo "Error !" . $e-
        >getMessage();

    die;

}

```

```

function
    compressImage($source,
        $destination, $quality)

{

    $imgInfo =
        getimagesize($source);

    $mime = $imgInfo['mime'];

    switch ($mime) {

        case 'image/jpeg':

            $image =
                imagecreatefromjpeg(...

file_aksi_admin.txt

[8.09 AM, 16/2/2025] SANS.iD:
    <?php

session_start();

define('BASE_URL',
    (isset($_SERVER['HTTPS'])
    && $_SERVER['HTTPS'] ===
    'on' ? "https" : "http") .
    "://"
    $_SERVER['HTTP_HOST'] .
    "/klinikmuslimah/admin/")
    ;

$conn = null;

try {

    $host = "localhost";

    $user = "root";

    $pass = "";

    $database =
        "db_klinik_muslimah";

    $db =
        "mysql:dbname=$database;h
        ost=$host";

```

```

$conn = new PDO($db, $user,
    $pass);

$conn->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Error !" . $e->getMessage();

    die;
}

function
    compressImage($source,
        $destination, $quality)
{
    $imgInfo =
        getimagesize($source);

    $mime = $imgInfo['mime'];
    switch ($mime) {
        case 'image/jpeg':
            $image =
                imagecreatefromjpeg($source);

            break;

        case 'image/png':
            $image =
                imagecreatefrompng($source);

            break;

        case 'image/gif':
            $image =
                imagecreatefromgif($source);

            break;

        default:
            $image =
                imagecreatefromjpeg($source);

            imagejpeg($image,
                $destination, $quality);

            return $destination;
        }
    }

function php_slug($b)
{
    $slug =
        preg_replace('/[^a-z0-9]+/', '-',
            trim(strtolower($b)));

    return $slug;
}

date_default_timezone_set("Asia/Makassar");

$aksi = $_GET['aksi'] ?? '';
if ($aksi == "") {
    $aksi = $_POST['aksi'] ??
        '';
}

switch ($aksi) {
    case 'masuk':
        masuk($conn);

        break;

    case 'viewdokter':
        viewdokter($conn);

        break;

    case 'tambahdokter':

```

```

        tambahdokter($conn);
        break;
case 'ubahdokter':
    ubahdokter($conn);
    break;
case 'hapusdokter':
    hapusdokter($conn);
    break;
case 'viewtreatment':
    viewtreatment($conn);
    break;
case 'tambahtreatment':
    tambahtreatment($conn);
    break;
case 'ubahtreatment':
    ubahtreatment($conn);
    break;
case 'hapustreatment':
    hapustreatment($conn);
    break;
case 'viewproduk':
    viewproduk($conn);
    break;
case 'tambahproduk':
    tambahproduk($conn);
    break;
case 'ubahproduk':
        tambahdokter($conn);
        break;
case 'hapusproduk':
    hapusproduk($conn);
    break;
case 'viewpelanggan':
    viewpelanggan($conn);
    break;
case 'keluar':
    keluar();
    break;
}

function textBinASCII($text)
{
    $bin = array();
    for ($i = 0; strlen($text)
        > $i; $i++)
        $bin[] =
            decbin(ord($text[$i]));
    return implode(' ', $bin);
}

function ASCIIIBinText($bin)
{
    $text = array();
    $bin = explode(" ", $bin);
    for ($i = 0; count($bin) >
        $i; $i++)
        $text[] =
            chr(bindec($bin[$i]));
}

```

```

        return implode($text);
    }

function masuk($conn)
{
    global $conn;

    $user = $_POST["userlog"]
    ?? '';

    $pass = $_POST["passlog"]
    ?? '';

    $usernya =
        textBinASCII($user);

    $passnya =
        textBinASCII($pass);

    $kategori = substr($user,
        0, 5);

    if ($kategori == 'admin')
    {

        if
        (ASCIIIBinText($usernya) ==
        "adminklinikmuslimah" &&
        ASCIIIBinText($passnya) ==
        "BismillahAlhamdulillah") {

            $_SESSION['usernya'] =
                "adminklinikmuslimah";

            $_SESSION['namanya'] =
                "Admin Klinik Muslimah
                Kota Parepare";

            echo "Terimakasih
            !";

        } else {

            echo "Salah !";
        }
    }
}

    }

} else {

    $stmt = $conn-
    >prepare("SELECT * FROM
    tb_dokter WHERE
    UserDokter = :UserDokter
    AND PassDokter =
    :PassDokter");

    $stmt-
    >bindParam(':UserDokter',
    $user);

    $stmt-
    >bindParam(':PassDokter',
    $pass);

    $stmt->execute();

    if ($stmt->rowCount()
    > 0) {

        $row = $stmt-
        >fetch(PDO::FETCH_ASSOC);

        if ($pass !=
        $row["PassDokter"]) {

            echo "Salah
            !";

        } else {

            $_SESSION["usernya"] =
            $row["UserDokter"];

            $_SESSION["namanya"] =
            $row["NamaDokter"];

            echo
            "Terimakasih !";

        }

    } else {

        echo "Maaf !";

    }
}

```



```

    }

function viewdokter($conn)
{
    global $conn;

    $draw = $_POST['draw'];
    $row = $_POST['start'];

    $rowperpage =
        $_POST['length'];

    $columnIndex =
        $_POST['order'][0]['column'];

    $columnName =
        $_POST['columns'][$columnIndex]['data'];

    $columnSortOrder =
        $_POST['order'][0]['dir'];

    $searchValue =
        $_POST['search']['value'];

    $searchArray = array();

    $searchQuery = " ";

    if ($searchValue != '') {
        $searchQuery = " AND (

            UserDokter LIKE
            :UserDokter OR

            PassDokter LIKE
            :PassDokter OR

           >NamaDokter LIKE
            :NamaDokter ) ";

        $searchArray = array(

            'UserDokter' =>
            "%$searchValue%",

            'PassDokter' =>
            "%$searchValue%",

            'NamaDokter' =>
            "%$searchValue%"

        );

        $stmt = $conn->prepare("SELECT COUNT(*)
        AS allcount FROM
        tb_dokter");

        $stmt->execute();

        $records = $stmt->fetch();

        $totalRecords =
            $records['allcount'];

        $stmt = $conn->prepare("SELECT COUNT(*)
        AS allcount FROM tb_dokter
        WHERE 1 " . $searchQuery);

        $stmt->execute($searchArray);

        $records = $stmt->fetch();

        $totalRecordwithFilter =
            $records['allcount'];

        $stmt = $conn->prepare("SELECT * FROM
        tb_dokter WHERE 1 " .
        $searchQuery . " ORDER BY
        " . $columnName . " " .
        $columnSortOrder . " LIMIT
        :limit,:offset");

        foreach ($searchArray as
        $key => $search) {

            $stmt->bindValue(':'.
            $key, $search,
            PDO::PARAM_STR);

        }
    }
}

```

```

$stmt->bindValue(':limit',
(int)$row,
PDO::PARAM_INT);

$stmt->
>bindValue(':offset',
(int)$rowperpage,
PDO::PARAM_INT);

$stmt->execute();

$empRecords = $stmt->fetchAll();

$data = array();

foreach ($empRecords as $row) {

    $data[] = array(

        "UserDokter" =>
$row['UserDokter'],

        "PassDokter" =>
$row['PassDokter'],

        "NamaDokter" =>
$row['NamaDokter']

    );

}

$response = array(

    "draw" =>
intval($draw),

    "iTotalRecords" =>
$totalRecords,

    "iTotalDisplayRecords"
=>
$totalRecordwithFilter,

    "aaData" => $data

);

echo
json_encode($response);

```

```

}

function tambahdokter($conn)
{
    global $conn;

    $userdokter =
$_POST["userdokter"] ??
'';

    $passdokter =
$_POST["passdokter"] ??
'';

    $namadokter =
$_POST["namadokter"] ??
'';

    $dokter = [

        'UserDokter' =>
$userdokter,

        'PassDokter' =>
$passdokter,

        'NamaDokter' =>
$namadokter,

    ];

    $quercek = $conn->prepare("SELECT * FROM
tb_dokter WHERE
UserDokter=?");

    $quercek->execute(array($userdokter));

    $cekdata = $quercek->rowCount();

    if ($cekdata >= 1) {

        echo "Sudah Ada !";

    } else {

        $hasil = $conn->prepare("INSERT INTO
tb_dokter
(UserDokter,PassDokter,Na

```

```

maDokter)          VALUES
(:UserDokter,:PassDokter,
:NamaDokter)");

    $okok      =      $hasil-
>execute($dokter);

    if ($okok) {

        echo "Terimakasih
!";

    } else {

        echo "Maaf !";

    }

}

}

function ubahdokter($conn)
{

    global $conn;

    $kode = $_POST["idnya"] ??
    '';

    //      $userdokter      =
    $_POST["userdokter"]    ??
    '';

    $passdokter      =
    $_POST["passdokter"]    ??
    '';

    $namadokter      =
    $_POST["namadokter"]    ??
    '';

    $dokter = [

        'PassDokter'      =>
        $passdokter,

        'NamaDokter'      =>
        $namadokter,

        'UserDokter' => $kode,

    ];

```

```

$hasil      =      $conn-
>prepare("UPDATE
tb_dokter      SET
PassDokter=:PassDokter,
NamaDokter=:NamaDokter
WHERE
UserDokter=:UserDokter");

$okok      =      $hasil-
>execute($dokter);

if ($okok) {

    echo "Terimakasih !";

} else {

    echo "Maaf !";

}

}

function hapusdokter($conn)
{

    global $conn;

    $kode = $_GET['kode'] ??
    '';

    $hasil      =      $conn-
>prepare("DELETE      FROM
tb_dokter      WHERE
UserDokter=?");

    $okok      =      $hasil-
>execute(array($kode));

    if ($okok) {

        echo "Terimakasih !";

    } else {

        echo "Maaf !";

    }

}

}

```

```

function viewtreatment($conn)
{
    global $conn;

    $draw = $_POST['draw'];
    $row = $_POST['start'];

    $rowperpage = $_POST['length'];

    $columnIndex = $_POST['order'][0]['column'];
    $columnName = $_POST['columns'][$columnIndex]['data'];

    $columnSortOrder = $_POST['order'][0]['dir'];

    $searchValue = $_POST['search']['value'];

    $searchArray = array();
    $searchQuery = " ";

    if ($searchValue != '') {
        $searchQuery = " AND (

            IdTreatment LIKE
            :IdTreatment OR

            TindakanTreatment
            LIKE :TindakanTreatment
            OR

            NamaTreatment LIKE
            :NamaTreatment OR

            BiayaTreatment
            LIKE :BiayaTreatment ) ";

        $searchArray = array(

            'IdTreatment' =>
                "$searchValue",

            'TindakanTreatment' =>
                "$searchValue",

            'NamaTreatment' =>
                "$searchValue",

            'BiayaTreatment'
            => "$searchValue"

        );
    }

    $stmt = $conn->prepare("SELECT COUNT(*)
    AS allcount FROM
    tb_treatment");
    $stmt->execute();

    $records = $stmt->fetch();

    $totalRecords = $records['allcount'];

    $stmt = $conn->prepare("SELECT COUNT(*)
    AS allcount FROM
    tb_treatment WHERE 1 " .
    $searchQuery);

    $stmt->execute($searchArray);

    $records = $stmt->fetch();

    $totalRecordwithFilter = $records['allcount'];

    $stmt = $conn->prepare("SELECT * FROM
    tb_treatment WHERE 1 " .
    $searchQuery . " ORDER BY
    " . $columnName . " " .
    $columnSortOrder . " LIMIT
    :limit,:offset");

```

```

foreach ($searchArray as
$key => $search) {

    $stmt->bindValue(':'.
$key, $search,
PDO::PARAM_STR);
}

$stmt->bindValue(':limit',
(int)$row,
PDO::PARAM_INT);

$stmt-
>bindValue(':offset',
(int)$rowperpage,
PDO::PARAM_INT);

$stmt->execute();

$empRecords = $stmt-
>fetchAll();

$data = array();

foreach ($empRecords as
$row) {

    $data[] = array(

        "IdTreatment" =>
$row['IdTreatment'],

        "TindakanTreatment" =>
$row['TindakanTreatment']
,

        "NamaTreatment" =>
$row['NamaTreatment'],

        "BiayaTreatment"
=> $row['BiayaTreatment']

    );
}

$response = array(

    "draw" =>

        intval($draw),

        "iTotalRecords" =>
$totalRecords,

        "iTotalDisplayRecords"
=>
$totalRecordwithFilter,

        "aData" => $data

    );

    echo
    json_encode($response);
}

function
tambahtreatment($conn)

{

    global $conn;

    $tindakan =
$_POST["tindakantreatment"] ?? '';

    $nama =
$_POST["namatreatment"] ?? '';

    $harga =
$_POST["biayatrement"] ?? '';

    $treatment = [

        'TindakanTreatment' =>
$tindakan,

        'NamaTreatment' =>
$nama,

        'BiayaTreatment' =>
$harga,

    ];

    $hasil = $conn-
>prepare("INSERT INTO
tb_treatment

```

```

        (TindakanTreatment,>NamaTreatment,
        BiayaTreatment)
VALUES
(:TindakanTreatment,:NamaTreatment,:BiayaTreatment
);

$okok      =      $hasil-
>execute($treatment);

if ($okok) {

    echo "Terimakasih !";

} else {

    echo "Maaf !";

}

}

function ubahtreatment($conn)
{

    global $conn;

    $kode = $_POST["idnya"] ??
    '';

    $tindakan      =
    $_POST["tindakantreatment"] ??
    '';

    $nama      =
    $_POST["namatreatment"] ??
    '';

    $harga      =
    $_POST["biayatreatment"] ??
    '';

    $treatment = [

        'TindakanTreatment' =>
        $tindakan,

        'NamaTreatment'      =>
        $nama,

        'BiayaTreatment'     =>
        $harga,

        'IdTreatment'        =>

```

```

    $kode,

    ];

    $hasil      =      $conn-
    >prepare("UPDATE
    tb_treatment      SET
    TindakanTreatment=:TindakanTreatment,
    NamaTreatment=:NamaTreatment,
    BiayaTreatment=:BiayaTreatment
    WHERE
    IdTreatment=:IdTreatment"
    );

    $okok      =      $hasil-
    >execute($treatment);

    if ($okok) {

        echo "Terimakasih !";

    } else {

        echo "Maaf !";

    }

}

function hapustreatment($conn)
{

    global $conn;

    $kode = $_GET['kode'] ??
    '';

    $hasil      =      $conn-
    >prepare("DELETE      FROM
    tb_treatment      WHERE
    IdTreatment=?");

    $okok      =      $hasil-
    >execute(array($kode));

    if ($okok) {

        echo "Terimakasih !";

    } else {

```

```

        echo "Maaf !";
    }
}

function viewproduk($conn)
{
    global $conn;

    $draw = $_POST['draw'];
    $row = $_POST['start'];

    $rowperpage =
        $_POST['length'];

    $columnIndex =
        $_POST['order'][0]['column'];

    $columnName =
        $_POST['columns'][$columnIndex]['data'];

    $columnSortOrder =
        $_POST['order'][0]['dir'];

    $searchValue =
        $_POST['search']['value'];

    $searchArray = array();

    $searchQuery = " ";

    if ($searchValue != '') {
        $searchQuery = " AND (

            IdProduk      LIKE
            :IdProduk OR

            NamaProduk     LIKE
            :NamaProduk OR

            GambarProduk   LIKE
            :GambarProduk OR

            HargaProduk    LIKE
            :HargaProduk OR

```

```

            StokProduk     LIKE
            :StokProduk ) ";

        $searchArray = array(

            'IdProduk'      =>
            "%$searchValue%"

        );

        echo
            json_encode($response);
    }

    function keluar()
    {
        session_destroy();

        unset($_SESSION['usernya'],
            $_SESSION['namanya']);
    }
}

```



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

MAHASISWA : ARFIANI ADENAN	Pembimbing I : Ade Hastuty, ST., S.Kom., MT.
NIM : 218 280 093	Pembimbing II : Wahyuddin, S.Kom., M.Kom.
Judul Skripsi : APLIKASI E-KONSELING TATA KECANTIKAN WAJAH BERBASIS MOBILE SMARTPHONE	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 • Rancangan Output Judul • Algoritma E-konseling • Rancangan Basis Data • Rancangan Input	Selasa, 7 Mei 2024 <i>[Signature]</i>	Konsultasi 1 - Ikut penduan skripsi	<i>[Signature]</i>
Konsultasi 2 • Sistem Informasi Klinik Muslimah Kota Parepare • Desain Kartu anggota • Desain layanan jasa • Konsultasi Daring	Sabtu, 18 Mei 2024 <i>[Signature]</i>	Konsultasi 2 - Teori pada BAB II • & tambahkan	<i>[Signature]</i>
Konsultasi 3 • Buku yg digunakan • Nama Pembimbing I • Desain Kartu Anggota	Rabu 29 Mei 2024 <i>[Signature]</i>	Konsultasi 3 - Desain sistem • & perbaikan	<i>[Signature]</i>
Konsultasi 4 ACC proposal	<i>[Signature]</i>	Konsultasi 4 - ACC	<i>[Signature]</i>
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

Lanjutan...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6 1.	Selasa, 7 Mei 2024	Konsultasi 6 D. Fancangan	
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Parepare, 03 Mei 2024

Mengetahui
Ketua Program Studi

Mahasiswa

Marlina, S.Kom.,M.Kom.
NBM. 1162 680

ARFIANI ADENAN
NIM. 218280093

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI
~~HASIL~~

MAHASISWA : ARFIANI ADENAN	Pembimbing I : Ade Hastuty, ST., S.Kom., MT.
NIM : 218 280 093	Pembimbing II : Wahyuddin, S.Kom., M.Kom.
Judul Skripsi : SISTEM INFORMASI KLINIK KECANTIKAN MUSLIMAH KOTA PARE PARE	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Buat Teori algoritma, penrograman dan flowchart di Bab 2	27/08/24	Konsultasi 1 - perbaiki aplikasi	21/08/24
Konsultasi 2 Acc	30/08/2024	Konsultasi 2 Acc	
Konsultasi 3		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

Lanjutan...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6		Konsultasi 6	
Konsultasi 7		Konsultasi 7	
Konsultasi 8		Konsultasi 8	
Konsultasi 9		Konsultasi 9	
Konsultasi 10		Konsultasi 10	

Parepare, 21 Agustus 2024

Mengetahui
Ketua Program Studi

Mahasiswa

Marlina, S.Kom.,M.Kom.
NBM. 1162 680

ARFIANI ADENAN
NIM. 218280093

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : ARFIANI ADENAN	Pembimbing I : Ade Hastuty, ST., S.Kom., M.T.
NIM : 218280093	Pembimbing II : Wahyuddin, S.Kom., M.Kom.
Judul Skripsi : SISTEM INFORMASI KLINIK KECANTIKAN MUSLIMAH KOTA PARE PARE	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Tambahkan Kartu Kontrol pada Saat registrasi	Senin/10/03/25 	Konsultasi 1	
Konsultasi 2 Acc	Kamis/13/03/25 	Konsultasi 2	
Konsultasi 3		Konsultasi 3	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik