

4. IMPLEMENTASI SISTEM

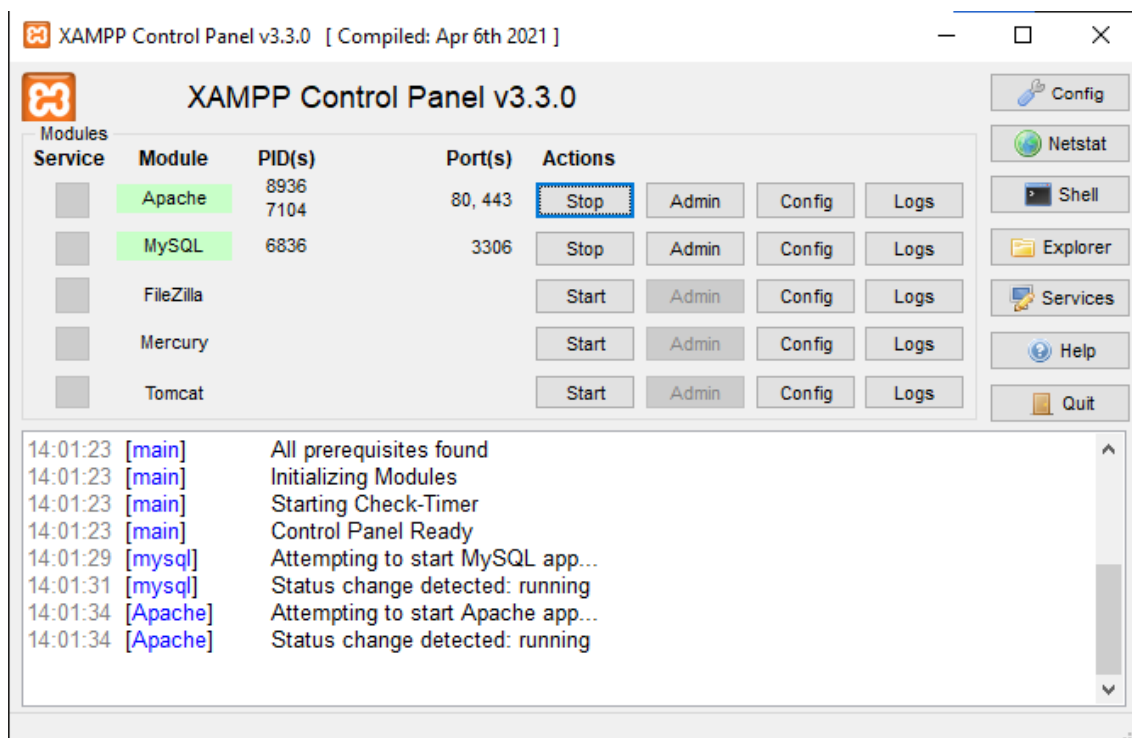
Pada bab ini membahas mengenai implementasi dari teori yang digunakan dan desain sistem yang telah dibahas pada bab-bab sebelumnya.

4.1 Implementasi Perangkat Lunak yang Digunakan

Pengembangan tampilan aplikasi *website* dibuat menggunakan HTML, CSS, dan Js pada *Microsoft visual studio*. Sistem penghubung dibangun dengan menggunakan bahasa pemrograman *PHP* sebagai *API* dan dihubungkan dengan menggunakan *MySQL* sebagai *database*. *Localhost* yang digunakan dalam pengembangan sistem yaitu *PHPMyAdmin*.

4.2 Instalasi Aplikasi Pendukung

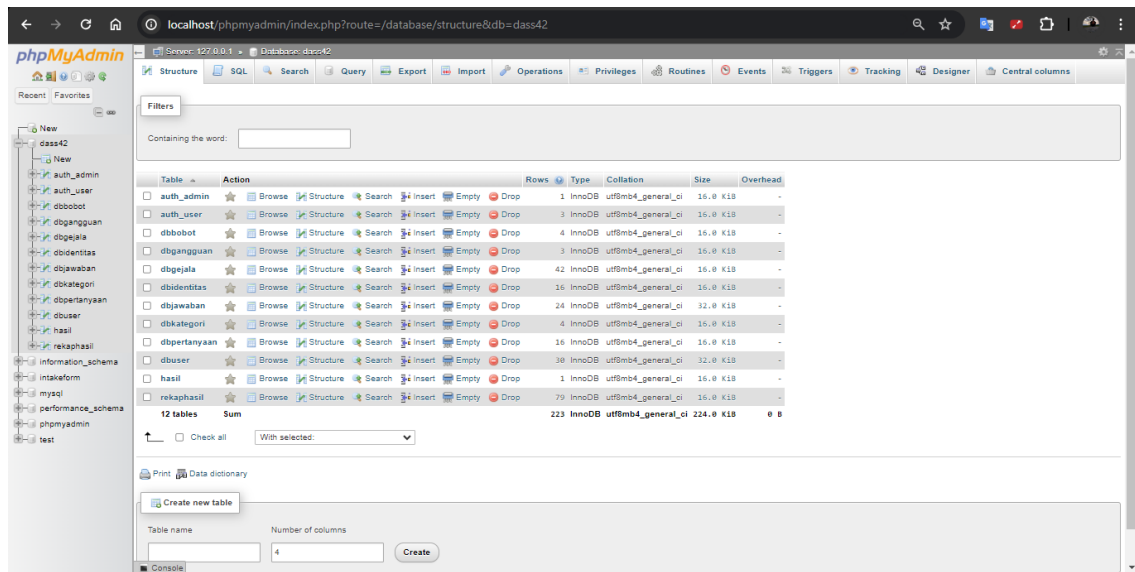
4.2.1 Instalasi XAMPP



Gambar 4. 1 Instalasi XAMPP

XAMPP kependekan dari X (cross platform) A (Apache), M (MySQL), P (PHP), dan P (Perl). Pada gambar 4.1 Instalasi XAMPP menjelaskan proses hubungan antara *website* dan *database* yang menggunakan *server localhost* sebagai proses pengembangan sistem.

4.2.2 Setting PHPMyAdmin



Gambar 4. 2 Setting PHPMyAdmin

Gambar 4.2 Setting PHPMyAdmin menjelaskan bahwa server *localhost* yang digunakan merupakan *phpMyAdmin* dan *MySQL* digunakan sebagai *database* dalam proses penyimpanan data sistem.

4.3 Implementasi Sistem User

Tabel 4.1

Tabel Pemetaan Segmen dan *Website* Sistem User

SEGMENT	PROSES	KETERANGAN
4.3.1	Koneksi Database	Menghubungkan <i>database MySQL</i> dengan <i>website</i> .
4.3.2	Implementasi GET Data Gejala	Proses pengambilan data gejala dari <i>database</i> agar dapat digunakan pada tes deteksi awal atau DASS-42 pada sistem yang dikembangkan.
4.3.3	Implementasi Fuzzy logic dan Forward Chaining pada Tes Deteksi Awal	Pengambilan gejala pada database, perhitungan dengan fuzzy logic, sampai menentukan hasil yang akan keluar dengan metode Forward Chaining berdasarkan apa yang sudah dipilih dan diproses pada pilihan-pilihan yang diberikan di tes deteksi awal masalah psikologis.

4.3.4	Implementasi Pencatatan Hasil Tes Deteksi Awal	Mencatat ke dalam database <i>id_hasil</i> yang telah dibuat dan hasil yang telah didapatkan.
4.3.5	Implementasi Pencatatan Total Tes Deteksi Awal	Mencatat ke dalam database seberapa banyak tes deteksi awal ini sudah dilakukan oleh siapapun.
4.3.6	Implementasi Pendataan Intake Form	Proses pengisian data yang dilakukan, guna dapat digunakan pada proses konseling.
4.3.7	Implementasi GET Pertanyaan Tes Intake Form	Proses pengambilan data pertanyaan dari database untuk digunakan di dalam tes <i>Intake Form</i> .
4.3.8	Implementasi Perhitungan Hasil Tes Intake Form	Proses perhitungan dari setiap opsi yang sudah dipilih dan dijumlahkan, kemudian diubah menjadi sebuah persentase dari faktor risiko klien.

4.3.1 Koneksi Database

Segmen Program 4.3. 1 Koneksi Database

```
<?php
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Headers: *");
$host    = 'localhost';
$user    = 'root';
$password = '';
$databse = 'dass42';

$conn = mysqli_connect($host, $user, $password);

$db = mysqli_select_db($conn, $databse) or
die(mysqli_error($conn));

?>
```

Pada implementasi kali ini berikut merupakan bagaimana agar koneksi sistem dapat terkoneksi dengan database.

4.3.2 Implementasi Get Data Gejala

Segmen Program 4.3. 2 Get Data Gejala

```
<?php
    include "conn.php";

    $arr=[];
    $q="SELECT * FROM dbgejala";
    $res=mysqli_query($conn,$q);
    while ($row=mysqli_fetch_assoc($res))
    {
        $arr[]=$row;
    }
    echo json_encode(["pertanyaan"=>$arr]);
?>
```

Segmen Program 4.3.2 Get Data Gejala menjelaskan proses pengambilan data gejala dari *database* agar dapat digunakan pada tes deteksi awal atau DASS-42 pada sistem yang dikembangkan.

4.3.3 Implementasi Fuzzy logic dan Forward Chaining pada Tes Deteksi Awal

Segmen Program 4.3. 3 Implementasi Fuzzy logic dan Forward Chaining pada Tes Deteksi Awal

```
<?php

    $pertanyaan=[];

    $q="SELECT * FROM dbgejala";

    $res=mysqli_query($conn,$q);

    while ($row=mysqli_fetch_assoc($res))

    {

        $pertanyaan[]=$row;
```

```

    }

    ?>

    var pertanyaan2=[];

```

Segmen Program 4.3.3 Implementasi Fuzzy logic dan Forward Chaining pada Tes Deteksi Awal menjelaskan proses yang dimulai dari pengambilan gejala pada database, penggunaan skala nilai dari 0-1 dengan fuzzy logic, sampai menentukan hasil yang akan keluar dengan metode Forward Chaining berdasarkan apa yang sudah dipilih dan diproses pada pilihan-pilihan yang diberikan di tes deteksi awal masalah psikologis.

```

var nilai = 0;

    if (jawaban == "Sering terjadi") {
        nilai = 1;
    }

    if (jawaban == "Kadang terjadi") {
        nilai = 0.67;
    }

    if (jawaban == "Jarang terjadi") {
        nilai = 0.34;
    }

```

Membuat *variable* nilai dengan skala 0-1 sesuai dengan *fuzzy logic*.

```

$(document).ready(function()
{
    var html="";
    pertanyaan2=[];
    for (var i=0;i<pertanyaan.length;i++)
    {

        if (i == 28 || i == 14 || i == 0) {
            pertanyaan2.push(pertanyaan[i]);
        }
    }
}

```

```

    }

    renderPertanyaan();
  });

```

Program akan diawali dengan menampilkan tiap gejala pertama dari setiap gejala inti untuk menentukan decision pertanyaan berikutnya.

```
var pertanyaanBerikut = [];
```

Kemudian variable ini akan digunakan untuk menginisialisasi pertanyaan berikutnya.

```

for (var i=0;i<pertanyaan2.length;i++) {
    var jawaban=
    $("input[name='pertanyaan'+pertanyaan2[i].id_gejala+'']:checked"
    ).val();

    // ...

    if (idgejala[0] == "s") {
        bobotS += nilai;
    }

    if (idgejala[0] == "d") {
        bobotD += nilai;
    }

    if (idgejala[0] == "k") {
        bobotK += nilai;
    }

    // ...
}

```

Program ini mengumpulkan jawaban dari setiap pertanyaan yang telah dijawab dan menghitung bobot untuk setiap kategori gejala (s, d, k).

```

if (bobotS >= 2) {
    for (var i = 0; i < nextS.length; i++) {

```

```

        pertanyaanBerikut.push(getPertanyaanByID(nextS[i]));
    }
}
if (bobotD >= 2) {
    for (var i = 0; i < nextD.length; i++) {
        pertanyaanBerikut.push(getPertanyaanByID(nextD[i]));
    }
}
if (bobotK >= 2) {
    for (var i = 0; i < nextK.length; i++) {
        pertanyaanBerikut.push(getPertanyaanByID(nextK[i]));
    }
}
}

```

Berdasarkan jawaban dari pertanyaan sebelumnya, program menentukan pertanyaan berikutnya yang harus diajukan.

```

if (idgejala == "s2") {
    nextS.push("s5");
    nextS.push("s6");
    nextS.push("s7");
}
if (idgejala == "s5") {
    nextS.push("s8");
    nextS.push("s9");
    nextS.push("s10");
}
if (idgejala == "s10") {
    nextS.push("s11");
    nextS.push("s12");
    nextS.push("s13");
    nextS.push("s14");
}

```

```

    }

    if (idgejala == "s14") {
        akhir = true;
        kesimpulan = "s";
        //alert("xx");
    }

    if (idgejala == "k2") {
        nextK.push("k5");
        nextK.push("k6");
        nextK.push("k7");
    }

    if (idgejala == "k5") {
        nextK.push("k8");
        nextK.push("k9");
        nextK.push("k10");
    }

    if (idgejala == "k10") {
        nextK.push("k11");
        nextK.push("k12");
        nextK.push("k13");
        nextK.push("k14");
    }

    if (idgejala == "k14") {
        akhir = true;
        kesimpulan = "k";
    }

    if (idgejala == "d2") {
        nextD.push("d5");
        nextD.push("d6");
        nextD.push("d7");
    }
}

```

```

        if (idgejala == "d5") {
            nextD.push("d8");
            nextD.push("d9");
            nextD.push("d10");
        }
        if (idgejala == "d10") {
            nextD.push("d11");
            nextD.push("d12");
            nextD.push("d13");
            nextD.push("d14");
        }
        if (idgejala == "d14") {
            akhir = true;
            kesimpulan = "d";
        }

```

Jika pertanyaan tertentu dijawab, program akan menambahkan pertanyaan-pertanyaan spesifik ke dalam daftar pertanyaan berikutnya.

```

if (pertanyaanBerikut.length <= 0) {
    if (akhir && kesimpulan == "s" && bobotS >= 1) {
        lkesimpulan = "Hasil deteksi awal menunjukkan ada indikasi gejala <mark>stress</mark>.";
    } else if (akhir && kesimpulan == "k" && bobotK >= 1) {
        lkesimpulan = "Hasil deteksi awal menunjukkan ada indikasi gejala <mark>kecemasan</mark>.";
    } else if (akhir && kesimpulan == "d" && bobotD >= 1) {
        lkesimpulan = "Hasil deteksi awal menunjukkan ada indikasi gejala <mark>depresi</mark>.";
    } else {
        lkesimpulan = "Untuk saat ini tidak muncul indikasi adanya gejala masalah psikologis yang dominan.";
    }
}

```

```
// Tampilkan hasil  
}
```

Jika tidak ada pertanyaan berikutnya yang ditemukan, program mengambil keputusan berdasarkan bobot yang telah dihitung.

```
function createpertanyaan(pt) { // createpertanyaan ke 2 = opsi  
radio jadi button, style di atas  
  
    var html = "<div class='row' style='margin-bottom:10px'>";  
  
    html += "<div class='col-md-12'>";  
  
    html += "<div style='font-weight:bold'>" + pt.id_gejala + "  
" + pt.nama_gejala + "</div>"; //HILANGKAN ID GEJALA NANTI  
  
    html += "<div class='radio-options'>";  
  
    html += "<input type='radio' id='sering'+pt.id_gejala+'  
name='pertanyaan" + pt.id_gejala + "' value='Sering terjadi'>";  
  
    html += "<label for='sering'+pt.id_gejala+'>Sering  
terjadi</label>";  
  
    html += "<input type='radio' id='kadang'+pt.id_gejala+'  
name='pertanyaan" + pt.id_gejala + "' value='Kadang terjadi'>";  
  
    html += "<label for='kadang'+pt.id_gejala+'>Kadang  
terjadi</label>";  
  
    html += "<input type='radio' id='jarang'+pt.id_gejala+'  
name='pertanyaan" + pt.id_gejala + "' value='Jarang terjadi'>";  
  
    html += "<label for='jarang'+pt.id_gejala+'>Jarang  
terjadi</label>";  
  
    html += "<input type='radio' id='tidak'+pt.id_gejala+'  
name='pertanyaan" + pt.id_gejala + "' value='Tidak terjadi'>";  
  
    html += "<label for='tidak'+pt.id_gejala+'>Tidak  
terjadi</label>";  
  
    html += "</div>";  
  
    html += "</div>";  
  
    html += "</div>";  
  
    return html;  
}
```

Adapun function yang digunakan seperti function ini yang berfungsi untuk menampilkan nama-nama gejala beserta opsi pilihan didalam sebuah container.

```
function renderPertanyaan()
{
    var html="";
    for (var i=0;i<pertanyaan2.length;i++)
    {
        html=html+createpertanyaan(pertanyaan2[i]);
    }
    $("#containerPertanyaan").html(html);
}

function getPertanyaanByID(id)
{
    for (var i=0;i<pertanyaan.length;i++)
    {
        if (pertanyaan[i].id_gejala==id)
        {
            return pertanyaan[i];
        }
    }
}
```

Kedua function diatas masing-masing berfungsi untuk merender pertanyaan dan mengambil pertanyaan berdasarkan id.

```
function next()
{

    // Validate that all questions are answered
```

```

        for (var i=0; i<pertanyaan2.length; i++) {
            var jawaban =
$("input[name='pertanyaan"+pertanyaan2[i].id_gejala+"']:checked"
).val();

            if (!jawaban) {
                alert("Harap jawab semua pertanyaan sebelum
melanjutkan.");
                return;
            }
        }
    }
}

```

Function *next* digunakan untuk pengecekan dari opsi setiap gejala yang harus dipilih sebelum dapat melanjutkan untuk memunculkan gejala berikutnya.

```

function generateUserId() {
    return 'user-' + Math.random().toString(36).substr(2, 9);
}

var userId = generateUserId(); // Generate a unique user ID
when the page loads

function rekaphasil() {
    var result = lkesimpulan; // Assuming `lkesimpulan` holds
the final result text

    $.ajax({
        url: 'rekaphasil.php',
        type: 'POST',
        data: {
            id_hasil: userId,
            hasil: result
        },
        dataType: 'json',
        success: function(response) {

```

```

        if (response.status === 'success') {
            console.log('Result recorded:',
response.message);

            $("#containerID").html(response.user_Id);

            $("#containerID").show();

        } else {
            console.log('Error:', response.message);

        }

    },

    error: function(error) {
        console.log('Error:', error);

    }

});

```

Ditambahkan function untuk membuat sebuah id_hasil untung menampung hasil yang didapatkan dari seorang klien yang tidak kita ketahui, sehingga ketika klien tersebut ingin melanjutkan untuk proses pendaftaran konseling nantinya, kita dapat menggunakan id_hasil ini untuk pengecekan yang dibutuhkan.

```

function catathasil()
{
    $.ajax({
        url: 'catathasil.php', // URL to which the request
is sent

        type: 'GET', // Type of request to be made ("POST"
or "GET"), default is "GET"

        data:{},

        dataType: 'json',

        success: function(result) {
            console.log("result",result);

            $("#containerCount").html(result.total);

            $("#containerCount").show();

```

```

    },
    error: function(error) {
        console.log('Error:', error);
    }
});
}

```

Function terakhir yaitu sebuah function yang berguna untuk mencatat total tes yang sudah diselesaikan dan kemudian dimunculkan pada pojok kanan bawah halaman hasil dari Tes Deteksi Awal. Ini berguna agar Staff PKPP Universitas Kristen Petra mengetahui seberapa banyak minat dan antusias terhadap Tes Deteksi Awal masalah psikologis pada seseorang.

4.3.4 Implementasi Pencatatan Hasil Tes Deteksi Awal

Segmen Program 4.3. 4 Pencatatan Hasil Tes Deteksi Awal

```

<?php
include "conn.php";

if (isset($_POST['id_hasil']) && isset($_POST['hasil'])) {
    $id_hasil = mysqli_real_escape_string($conn,
    $_POST['id_hasil']);
    $hasil = mysqli_real_escape_string($conn, $_POST['hasil']);

    $query = "INSERT INTO rekaphasil (id_hasil, hasil) VALUES
    ('$id_hasil', '$hasil')";
    if (mysqli_query($conn, $query)) {
        echo json_encode(['status' => 'success', 'message' =>
        'hasil recorded successfully']);
    } else {
        echo json_encode(['status' => 'error', 'message' =>
        'Failed to record hasil']);
    }
} else {
    echo json_encode(['status' => 'error', 'message' => 'Invalid
    parameters']);
}

```

```
}  
?>
```

Segmen Program 4.3.4 Pencatatan Hasil Tes Deteksi Awal akan mencatat ke dalam database *id_hasil* yang telah dibuat dan hasil yang telah didapatkan agar dapat digunakan untuk proses selanjutnya.

4.3.5 Implementasi Pencatatan Total Tes Deteksi Awal

Segmen Program 4.3. 5 Pencatatan Total Tes Deteksi Awal

```
<?php  
  
    include "conn.php";  
  
    $q="UPDATE hasil SET total=total+1";  
  
    mysqli_query($conn,$q);  
  
    $total=0;  
  
    $q="SELECT total FROM hasil";  
  
    $res=mysqli_query($conn,$q);  
  
    while ($row=mysqli_fetch_assoc($res))  
    {  
        $total=$row["total"];  
    }  
  
    echo json_encode(["total"=>$total]);  
  
?>
```

Segmen Program 4.3.5 Pencatatan Total Tes Deteksi Awal akan mencatat ke dalam database seberapa banyak tes deteksi awal ini sudah dilakukan oleh siapapun, akan terhitung count +1 pada tes ini bagi siapapun yang menyelesaikan tes nya.

4.3.6 Implementasi Pendataan Intake Form

Segmen Program 4.3. 6 Pendataan Intake Form

```
<?php
    include "conn.php";

    $penyakit=mysqli_real_escape_string($conn,$_POST["penyakit"]);

    $harapan=mysqli_real_escape_string($conn,$_POST["harapan"]);

    $kunjungan=mysqli_real_escape_string($conn,$_POST["kunjungan"]);
    $alasan=mysqli_real_escape_string($conn,$_POST["alasan"]);

    $id_hasil=mysqli_real_escape_string($conn,$_POST["id_hasil"]);

    $q="INSERT INTO dbuser
    (penyakit,harapan,kunjungan,alasan,id_hasil)
    VALUES
    ('$penyakit','$harapan','$kunjungan','$alasan','$id_hasil')";

    mysqli_query($conn,$q);

    $id_user=mysqli_insert_id($conn);

    header("location:isipertanyaan.php?id=".$id_user);

?>
```

Segmen Program 4.3.6 Pendataan Intake Form menjelaskan proses pengisian data yang dilakukan, guna dapat digunakan pada proses konseling nantinya oleh konselor / psikolog karena data yang diberikan akan di catat di dalam *database*.

4.3.7 Implementasi GET Pertanyaan Tes Intake Form

Segmen Program 4.3. 7 GET Pertanyaan Tes Intake Form

```
<?php
    include "conn.php";
    include "top.php";

    $id = mysqli_real_escape_string($conn, $_GET["id"]);
```

```

        // Query untuk mengambil pertanyaan dari database
        $query = "SELECT id_pertanyaan, pertanyaan FROM
dbpertanyaan";

        $result = mysqli_query($conn, $query);

        // Buat array untuk menyimpan pertanyaan
        $questions = [];

        while($row = mysqli_fetch_assoc($result)) {
            $questions[$row['id_pertanyaan']] = $row['pertanyaan'];
        }

?>

```

Segmen Program 4.3.7 GET Pertanyaan Tes Intake Form menjelaskan proses pengambilan data pertanyaan dari database untuk digunakan di dalam tes *Intake Form*.

4.3.8 Implementasi Perhitungan Hasil Tes Intake Form

Segmen Program 4.3. 8 Perhitungan Hasil Tes Intake Form

```

<?php

include "conn.php";

        $beratMasalah = mysqli_real_escape_string($conn,
$_POST["beratMasalah"]);

        $durasiMasalah = mysqli_real_escape_string($conn,
$_POST["durasiMasalah"]);

        $sebelumnya = mysqli_real_escape_string($conn,
$_POST["sebelumnya"]);

        $gangguanMental = mysqli_real_escape_string($conn,
$_POST["gangguanMental"]);

        $tekananEmosional = mysqli_real_escape_string($conn,
$_POST["tekananEmosional"]);

        $pengaruhMasalah = mysqli_real_escape_string($conn,
$_POST["pengaruhMasalah"]);

        $hobi = mysqli_real_escape_string($conn, $_POST["hobi"]);

```

```

    $keyakinan = mysqli_real_escape_string($conn,
$_POST["keyakinan"]);

    $pasangan = mysqli_real_escape_string($conn,
$_POST["pasangan"]);

    $ayah = mysqli_real_escape_string($conn, $_POST["ayah"]);

    $ibu = mysqli_real_escape_string($conn, $_POST["ibu"]);

    $saudara = mysqli_real_escape_string($conn,
$_POST["saudara"]);

    $teman = mysqli_real_escape_string($conn,
$_POST["teman"]);

    $sahabat = mysqli_real_escape_string($conn,
$_POST["sahabat"]);

    $komunitas = mysqli_real_escape_string($conn,
$_POST["komunitas"]);

    $kontribusi = mysqli_real_escape_string($conn,
$_POST["kontribusi"]);

    $id = mysqli_real_escape_string($conn, $_POST["id"]);

    $hasil = ($beratMasalah * 1) + ($durasiMasalah * 1) +
($sebelumnya * 1) + ($gangguanMental * 1) + ($tekananEmosional *
1) + ($pengaruhMasalah * 1) + ($hobi * 1) + ($keyakinan * 1) +
($pasangan * 1) + ($ayah * 1) + ($ibu * 1) + ($saudara * 1) +
($teman * 1) + ($sahabat * 1) + ($komunitas * 1) + ($kontribusi
* 1);

    $percentage = ($hasil / 32) * 100;

    ?>

```

Segmen Program 4.3.8 Perhitungan Hasil Tes Intake Form menjelaskan proses seorang klien dalam mengisi beberapa pertanyaan-pertanyaan yang dibutuhkan. Hasil perhitungan per *item* pertanyaan dan juga hasil totalnya ada namun tidak dimunculkan bagi klien, tetapi diberikan sebuah persentase yang dapat menjadi gambaran hasil bagi klien tersebut agar kita dapat mengetahui gambaran faktor risiko yang dimiliki oleh klien sesuai dengan perhitungan dari setiap *item* pertanyaan yang sudah dijawab. Opsi dari setiap pilihan pertanyaan yang ada didalam tes ini juga dicatat ke dalam database sebagai bentuk jawaban yang dipilih oleh klien.

4.4 Implementasi Sistem Admin

Tabel 4.2

Tabel Pemetaan Segmen Sistem Admin

SEGMENT	PROSES	KETERANGAN
4.4.1	Implementasi GET Login Data	Melakukan pengecekan <i>login</i> data.
4.4.2	Implementasi Edit Tes Deteksi Awal (DASS-42)	Melakukan Pengambilan gejala dari database dan memberikan kewenangan bagi admin untuk merubah nama gejala jika terjadi kesalahan penulisan.
4.4.3	Implementasi Edit Tes Intake Form	Melakukan Pengambilan pertanyaan dari database dan memberikan kewenangan bagi admin untuk merubah pertanyaan jika terjadi kesalahan penulisan.
4.4.4	Implementasi GET Hasil Deteksi Awal	Menampilkan data id_hasil dan juga hasil dari <i>database</i> dan kemudian dapat digunakan oleh admin untuk melihat dan menggunakannya untuk keperluan konseling.
4.4.5	Implementasi GET Hasil Intake Form	Menampilkan data jawaban dari user secara lengkap pada tabel melalui pencarian id_user.

4.4.1 Implementasi GET Login Data

Segmen Program 4.4. 1 GET Login Data

```
<?php
include "conn.php";
session_start(); // Pastikan session_start() dipanggil di
awal

// Cek apakah form login telah disubmit
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Ambil nilai username dan password dari form
    $uname_admin = $_POST['uname_admin'];
    $password_admin = $_POST['password_admin'];
```

```

        // Query ke database untuk mendapatkan data admin
dengan username yang sesuai

        $query = "SELECT * FROM auth_admin WHERE uname_admin =
'$uname_admin'";

        $result = mysqli_query($conn, $query);

        if ($result) {
            // Jika data ditemukan
            if (mysqli_num_rows($result) > 0) {
                $auth_admin = mysqli_fetch_assoc($result);

                // Bandingkan password yang dimasukkan dengan
password yang disimpan di database
                if (password_verify($password_admin,
$auth_admin['password_admin'])) {
                    // Jika password benar, set session
admin_logged_in menjadi true dan redirect ke halaman admin
                    $_SESSION['admin_logged_in'] = true;
                    echo "<script>

                        alert('Login berhasil');

                        window.location.href =
'admin_dashboard.php';

                    </script>";

                    exit;
                } else {
                    // Jika password salah, tampilkan pesan
error
                    $error = "Password salah!";
                }
            } else {
                // Jika username tidak ditemukan, tampilkan
pesan error
                $error = "Username tidak ditemukan!";
            }
        }
    }
}

```

```

    }

    } else {

        // Jika terjadi kesalahan saat mengambil data dari
        database, tampilkan pesan error

        $error = "Terjadi kesalahan. Silakan coba lagi!";

    }

}

?>

```

Pada implementasi ini menunjukkan bagaimana data akun admin dari database di ambil dan di cek untuk verifikasinya sebelum dapat masuk ke halaman selanjutnya yaitu *dashboard admin*.

4.4.2 Implementasi Edit Tes Deteksi Awal (DASS-42)

Segmen Program 4.4. 2 Edit Tes Deteksi Awal (DASS-42)

```

<?php
include "conn.php";
//include "top.php";
// Mencegah caching
header("Cache-Control: no-cache, must-revalidate"); // HTTP 1.1.
header("Pragma: no-cache"); // HTTP 1.0.
header("Expires: 0"); // Proxies.

if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST['update'])) {

    // Ambil nilai dari form

    $id_gejala = $_POST['id_gejala'];

    $nama_gejala = $_POST['nama_gejala'];

    // Update database

    $query = "UPDATE dbgejala SET nama_gejala = ? WHERE
id_gejala = ?";

    $stmt = mysqli_prepare($conn, $query);

```

```

        mysqli_stmt_bind_param($stmt, "ss", $nama_gejala,
$id_gejala);

        if (mysqli_stmt_execute($stmt)) {
            echo "<script>

                alert('Data berhasil diupdate!');

                window.location.href = window.location.href; //
Reload halaman untuk menampilkan data terbaru

            </script>";
        } else {
            echo "Error: " . mysqli_error($conn);
        }
    }

    // Query untuk mendapatkan data dari dbgejala
    $query = "SELECT * FROM dbgejala";
    $result = mysqli_query($conn, $query);
?>

```

Implementasi ini ditujukan untuk menampilkan data gejala dari *database* dan kemudian dapat di *edit* oleh seorang *admin* jika kedapatan adanya kesalahan penulisan pada nama gejala yang digunakan di dalam Tes Deteksi Awal.

4.4.3 Implementasi Edit Tes Intake Form

Segmen Program 4.4. 3 Edit Tes Intake Form

```

<?php
include "conn.php";
//include "top.php";

// Mencegah caching
header("Cache-Control: no-cache, must-revalidate"); // HTTP 1.1.
header("Pragma: no-cache"); // HTTP 1.0.
header("Expires: 0"); // Proxies.

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST['update'])) {

    // Ambil nilai dari form

    $id_pertanyaan = $_POST['id_pertanyaan'];

    $pertanyaan = $_POST['pertanyaan'];

    // Update database

    $query = "UPDATE dbpertanyaan SET pertanyaan = ? WHERE
id_pertanyaan = ?";

    $stmt = mysqli_prepare($conn, $query);

    mysqli_stmt_bind_param($stmt, "ss", $pertanyaan,
$id_pertanyaan);

    if (mysqli_stmt_execute($stmt)) {

        echo "<script>

            alert('Data berhasil diupdate!');

            window.location.href = window.location.href; //
Reload halaman untuk menampilkan data terbaru

        </script>";

    } else {

        echo "Error: " . mysqli_error($conn);

    }

}

// Query untuk mendapatkan data dari dbpertanyaan
$query = "SELECT * FROM dbpertanyaan";
$result = mysqli_query($conn, $query);
?>

```

Implementasi ini ditujukan untuk menampilkan data pertanyaan dari *database* dan kemudian dapat di *edit* oleh seorang *admin* jika kedapatan adanya kesalahan penulisan pada pertanyaan yang digunakan di dalam Tes Intake Form.

4.4.4 Implementasi GET Hasil Deteksi Awal

Segmen Program 4.4. 4 GET Hasil Deteksi Awal

```
<?php
include "conn.php";

// Inisialisasi variabel
$id_hasil_search = '';
$query = "SELECT * FROM rekaphasil";

// Cek jika form pencarian telah dikirimkan
if (isset($_GET['id_hasil']) && !empty($_GET['id_hasil'])) {
    $id_hasil_search = mysqli_real_escape_string($conn,
    $_GET['id_hasil']);
    $query = "SELECT * FROM rekaphasil WHERE
    id_hasil='$id_hasil_search'";
}
// Jalankan query
$result = mysqli_query($conn, $query);
?>
```

Implementasi ini ditujukan untuk menampilkan data `id_hasil` dan juga hasil dari *database* dan kemudian dapat digunakan oleh admin untuk melihat dan menggunakannya untuk keperluan konseling.

4.4.5 Implementasi GET Hasil Intake Form

Segmen Program 4.4. 5 GET Hasil Intake Form

```
<?php
include "conn.php";

// Inisialisasi variabel
$id_user_search = '';
$query = "SELECT * FROM dbjawaban";
```

```
// Cek jika form pencarian telah dikirimkan
if (isset($_GET['id_user']) && !empty($_GET['id_user'])) {
    $id_user_search = mysqli_real_escape_string($conn,
$_GET['id_user']);
    $query = "SELECT * FROM dbjawaban WHERE
id_user='$id_user_search'";
}

// Jalankan query
$result = mysqli_query($conn, $query);
?>
```

Implementasi ini ditujukan untuk menampilkan data jawaban dari user secara lengkap pada tabel melalui pencarian *id_user*.