

**LAPORAN PRAKTIKUM  
PEMROGRAMMAN WEB  
MODUL 9 & 10**

**PROJECT**



**NIM : 20220810052  
Nama : Ilham Hafidz  
Kelas : TINFC 2022 01**

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS KUNINGAN  
2022**

- Authentication (Login & Registrasi)
  - File login.php pada folder pages

```
<?php
session_start();
if (isset($_SESSION["id"]) && isset($_SESSION["username"]) && isset($_SESSION["role"])) {
    if ($_SESSION["role"] == "admin") {
        header("Location: ./admin/index.php");
    } else {
        header("Location: ../../index.php");
    }
}
?>
```

Analisis :

Potongan program ini akan melakukan pengecekan pada session. Jika pada variabel global \$\_SESSION terdapat key "id", "username" dan "role". Jika key tersebut sudah di set pada variabel global \$\_SESSION (artinya sudah melakukan login) maka akan melakukan redirect ke halaman sesuai dengan pages khusus role tersebut.

- File login.php pada folder backend

```
<?php
require "../connection.php";

session_start();

if (isset($_POST["submit"])) {

    extract($_POST);

    $result = mysqli_query($connect, "SELECT * FROM accounts WHERE username='$username' AND password='$password'");

    if (mysqli_num_rows($result)) {

        foreach ($result as $account) {

            $_SESSION["id"] = $account["id"];
            $_SESSION["username"] = $account["username"];
            $_SESSION["role"] = $account["role"];

            if ($account["role"] == "admin") {
                header("Location: ../../pages/admin/index.php");
            } else {
                header("Location: ../../index.php");
            }
        }
    }
}
```

```

    }
    } else {
        echo '
            <script src="https://code.jquery.com/jquery-3.7.1.slim.min.js" integrity="sha256-
            kmHvs0B+OpCW5GVHUNjv9rOmY0IvSIRcf7zGUDTDQM8="
            crossorigin="anonymous"></script>
            <script
            src="https://cdnjs.cloudflare.com/ajax/libs/sweetalert/2.1.2/sweetalert.min.js"></script>
            <script type="text/javascript">
                $(document).ready(function(){
                    swal({
                        title: "Gagal",
                        text: "Email atau Password salah",
                        icon: "error",
                        buttons: false
                    }).then((value) => {
                        window.location.href = "http://localhost/quiz/pages/login.php";
                    });
                });
            </script>
        '
    }
}

```

Analisis :

Program ini akan dijalankan ketika ada “request” POST dengan key submit. Ketika ada request tersebut. Melakukan query untuk get data dari tabel accounts yang (WHERE) username dan password-nya sama dengan username dan password request-an dari user.

#### - File register.php

```

<?
session_start();
if (isset($_SESSION["id"]) && isset($_SESSION["username"]) && isset($_SESSION["role"])) {
    if ($_SESSION["role"] == "admin") {
        header("Location: ./admin/index.php");
    } else {
        header("Location: ../index.php");
    }
}
?>

```

Analisis :

Potongan program ini akan melakukan pengecekan pada session. Jika pada variabel global \$\_SESSION terdapat key “id”, “username” dan “role”. Jika key tersebut sudah di set pada variabel global \$\_SESSION (artinya sudah melakukan login) maka akan melakukan redirect ke halaman sesuai dengan

pages khusus role tersebut.

- File register.php pada folder backend

```
<?php
require "../connection.php";

echo '
<script src="https://code.jquery.com/jquery-3.7.1.slim.min.js" integrity="sha256-
kmHvs0B+OpCW5GVHUNjv9r0mY0IvSIRcf7zGUDTDQM8=" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/sweetalert/2.1.2/sweetalert.min.js"></script>
';

if (isset($_POST["submit"])) {
    extract($_POST);

    if ($password != $confirm_password) { // cek password & confirm password
        echo '
        <script type="text/javascript">
            $(document).ready(function(){
                swal({
                    title: "Gagal",
                    text: "Password & Confirm Password tidak sama",
                    icon: "error",
                }).then((value) => {
                    window.location.href = "http://localhost/quiz/pages/register.php";
                });
            });
        </script>
        ';

        exit();
    }

    $check = mysqli_query($connect, "SELECT * FROM accounts WHERE username='$username'");

    if (mysqli_num_rows($check)) { // cek username
        echo '
        <script type="text/javascript">
            $(document).ready(function(){
                swal({
                    title: "Gagal",
                    text: "Username sudah tersedia",
                    icon: "error",
                }).then((value) => {
                    window.location.href = "http://localhost/quiz/pages/register.php";
                });
            });
        </script>
        ';

        exit();
    }
}
```

```

}

$result = mysqli_query($connect, "INSERT INTO accounts VALUES(null, '$username', '$name', '$password', 'user', 'avatar1.png')");

echo '
<script type="text/javascript">
$(document).ready(function(){
    swal({
        title: "Berhasil",
        text: "Anda sudah bisa login dengan user terdaftar",
        icon: "success",
    }).then((value) => {
        window.location.href = "http://localhost/quiz/pages/login.php";
    });
});
</script>
';
}

```

#### Analisis :

Program ini akan dijalankan ketika ada “request” POST dengan key submit. Ketika ada request tersebut. Melakukan pengecekan apakah request password dan confirm passwordnya sama. Jika tidak maka program akan diberhentikan dengan fungsi exit(). query untuk get data dari tabel accounts yang (WHERE) username dan password-nya sama dengan username dan password request- an dari user.

Menjalankan query untuk mengecek apakah ada username yang sama dengan request username dari user. Jika ada maka username tersebut berarti sudah terdaftar dan proses registrasi akan digagalkan karena username pengguna bersifat unique.

Jika semua pengecekan masing-masing bernilai false, maka akan melakukan query untuk insert data pada tabel accounts dengan value yang sama dengan requestan dari user.

#### - File backend/logout.php

```

<?php
session_start();
session_unset();
session_destroy();

header("Location: ../../index.php");

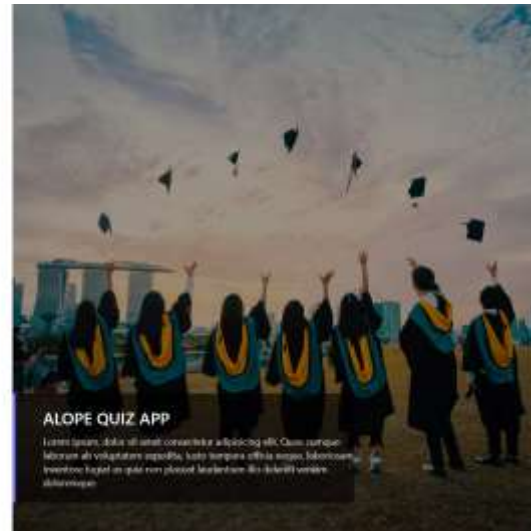
```

Analisis :

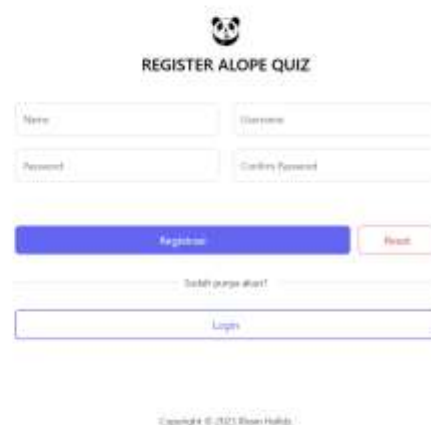
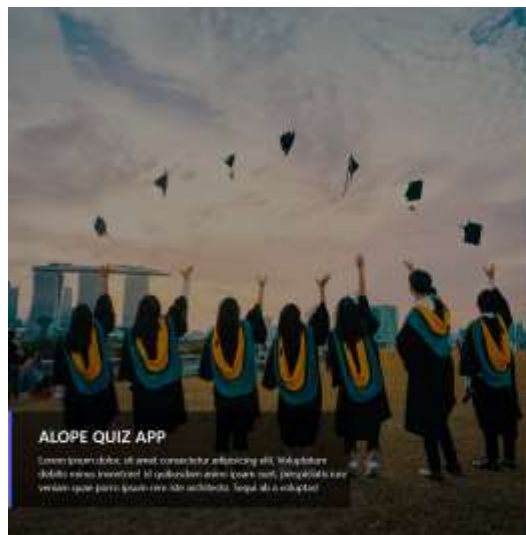
Sebelum logout (meredirect ke halaman index), session akan dihapus terlebih dahulu menandakan semua sesi telah berakhir.

Output :

- o Halaman Login



- o Halaman Register



- Admin Pages (Halaman Khusus Admin)
  - File pages/admin/index.php

```
<?php
require "../../backend/connection.php";

$userCount = mysqli_num_rows(mysqli_query($connect, "SELECT * FROM accounts WHERE
role='user'"));
$courseCount = mysqli_num_rows(mysqli_query($connect, "SELECT * FROM courses"));
$questionCount = mysqli_num_rows(mysqli_query($connect, "SELECT * FROM questions"));
$answerCount = mysqli_num_rows(mysqli_query($connect, "SELECT * FROM answers"));

$usersCountPerMonth = array_fill_keys(
    ["januari", "februari", "maret", "april", "mei", "juni", "juli", "agustus", "september",
    "oktober", "november", "desember"],
    0
);

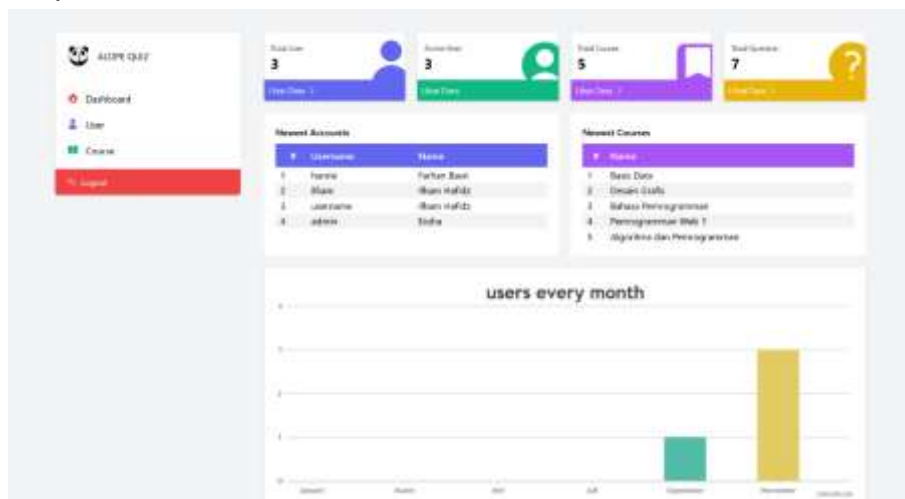
$accounts = mysqli_query($connect, "SELECT * FROM accounts ORDER BY id DESC");
$newestAccounts = mysqli_query($connect, "SELECT * FROM accounts ORDER BY id DESC LIMIT 5");
$newestCourses = mysqli_query($connect, "SELECT * FROM courses ORDER BY id DESC LIMIT 5");

foreach ($accounts as $account) {
    $month = substr($account["date"], 5, 2);
    $monthName = date("F", mktime(0, 0, 0, $month, 1));
    $usersCountPerMonth[strtolower($monthName)]++;
}
?>
```

Analisis :

- Menghitung jumlah row(data) pada tabel dengan mysqli\_rows.
- Membuat sebuah array assosiative dengan kumpulan key dari nama bulan yang masing-masing mempunyai value 0.

Output :



- File pages/admin/user.php (bagian get data dan delete data user)

```
<?php
require "../../backend/connection.php";
require "../../backend/User.php";
session_start();

if (!isset($_SESSION["id"]) && !isset($_SESSION["username"]) && !isset($_SESSION["role"])) {
    header("Location: ../pages/login.php");
}

$users = UserIndex($connect);

if (isset($_GET["deletedId"])) {
    $id = $_GET["deletedId"];
    mysqli_query($connect, "DELETE FROM accounts WHERE id=$id");

    header("Location: ../user.php");
}

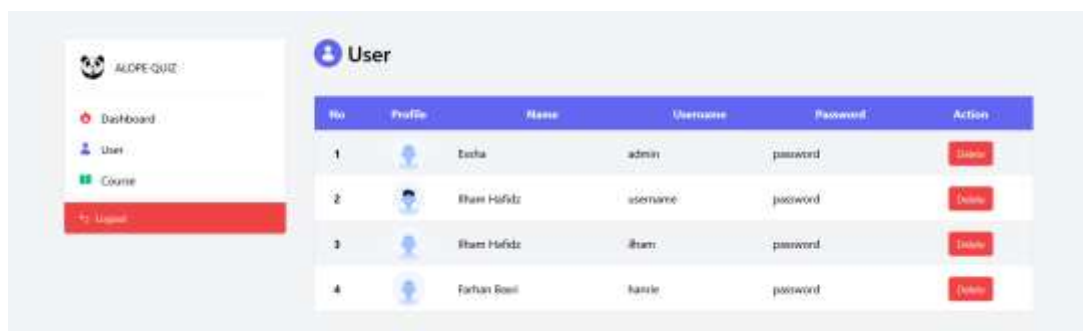
?>
```

Analisis :

Memanggil fungsi UserIndex dengan mengirimkan \$connect sebagai argument yang hasil return dari fungsi tersebut akan disimpan kedalam variabel users.

Akan melakukan pengecekan jika terdapat request GET dari user dengan key "deletedId", jika request tersebut sudah di set maka akan menjalankan query untuk mendelete data accounts sesuai dengan id yang dikirimkan pada request GET tersebut.

Output :



The screenshot shows a web application interface. On the left is a sidebar menu with a logo and the text 'ALDOPE-QUIZ'. The menu items are 'Dashboard', 'User', 'Course', and a 'Logout' button at the bottom. The 'User' item is highlighted. The main content area is titled 'User' and contains a table with the following data:

No	Profile	Name	Username	Password	Action
1		Basha	admin	password	<a href="#">Delete</a>
2		Iham Hafidz	username	password	<a href="#">Delete</a>
3		Iham Hafidz	iham	password	<a href="#">Delete</a>
4		Farhan Rendi	hansle	password	<a href="#">Delete</a>



- File pages/admin/courses/index.php

```
<?php
require "../backend/connection.php";
require "../backend/Course.php";

session_start();
if (!isset($_SESSION["id"]) && !isset($_SESSION["username"]) && !isset($_SESSION["role"])) {
    header("Location: ../login.php");
}

$courses = index($connect);

if (isset($_GET["deletedId"])) {
    $deletedId = $_GET["deletedId"];

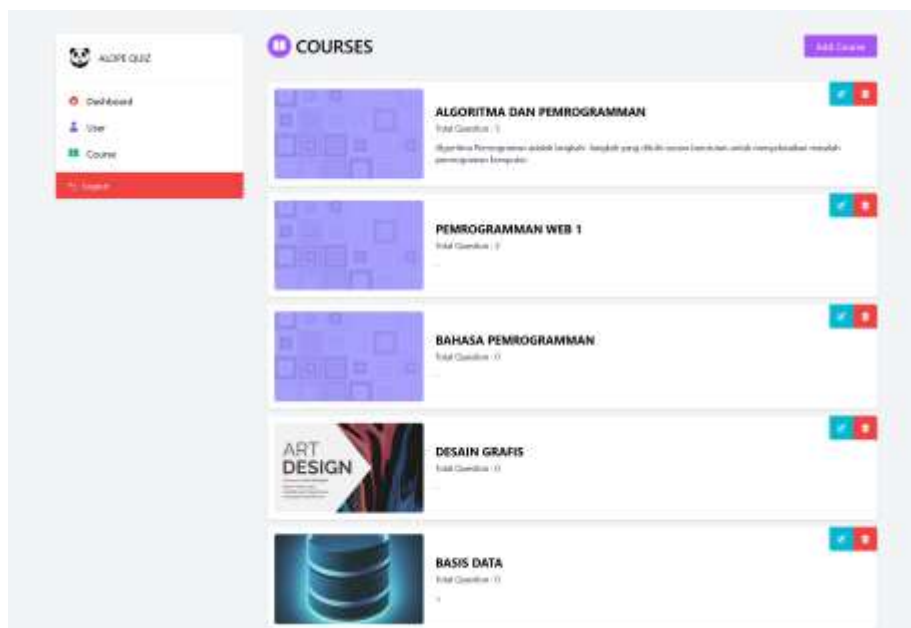
    confirmDelete();

    if (isset($_GET["yesDeleted"])) {
        delete($connect, $deletedId);
    }
}

function countTotalCourse($id)
{
    global $connect;
    $getDataByCourseId = mysqli_query($connect, "SELECT * FROM questions WHERE course_id=$id");

    return mysqli_num_rows($getDataByCourseId);
}
?>
```

Output :



- File pages/admin/courses/show.php (bagian algoritma skor)

```
<?php

function getScore($accountId)
{
    global $connect;

    $courseId = $_GET["id"];
    $answersUser = mysqli_query(
        $connect,
        "SELECT * FROM answers
        INNER JOIN questions ON questions.id = answers.question_id
        WHERE answers.course_id=$courseId AND account_id=$accountId
        "
    );

    $score = 0;
    foreach ($answersUser as $answerUser) {
        if ($answerUser["user_answer"] == $answerUser["answer"]) {
            $score++;
        }
    }

    return $score;
}

?>
```

Analisis :

Membuat sebuah fungsi getScore yang menerima 1 parameter \$accountId. Statement fungsinya akan menjalankan sebuah query untuk getdata dari tabel answers sekaligus melakukan JOIN dengan tabel questions yang questions.id nya sama dengan answers.questions\_id dengan (WHERE) answers.course\_id nya itu sama dengan request get id dari user dan accountId nya sama dengan parameternya.

Melakukan perulangan pada hasil get data tadi dan melakukan pengecekan jika user\_answer nya sama dengan answer maka variabel point akan ditambah dengan 1.

Output :

The screenshot shows a web application titled 'ALORITMA DAN PEMROGRAMMAN'. On the left is a sidebar with a logo 'ALOPE QUIZ' and navigation links: 'Dashboard', 'User', 'Course', and a 'Logout' button. The main content area has a header with the course name and buttons for 'See User Details' and 'New'. Below the header is a 'Question' form with input fields for 'Option A', 'Option B', 'Option C', and 'Option D', a 'Jawab' section with radio buttons for A, B, C, and D, and a 'Submit' button. Below the form are five question cards. Each card has a title, a question, four multiple-choice options (A, B, C, D), and a 'Submit' button. The questions are:

- Question No. 1 :** Apa itu algoritma?  
A. Sekumpulan instruksi  
B. Bahasa pemrograman  
C. Sebuah file program  
D. Sistem operasi
- Question No. 2 :** Apa fungsi dari variabel dalam pemrograman?  
A. Menyimpan data  
B. Menjalankan perhitungan  
C. Mengambil input dari pengguna  
D. Menampilkan output ke layar
- Question No. 3 :** Pernyataan berikut yang benar mengenai Big O notation?  
A. Semakin kecil nilai Big O, semakin efisien algoritma.  
B. Big O hanya berlaku untuk bahasa pemrograman tertentu.  
C. Big O digunakan untuk mengukur kecepatan eksekusi program.  
D. Semakin besar nilai Big O, semakin efisien algoritma.
- Question No. 4 :** Apa yang dimaksud dengan "bubble sort" dalam konteks algoritma pengurutan?  
A. Algoritma pengurutan yang efisien  
B. Algoritma pengurutan dengan membandingkan dua elemen sekaligus  
C. Algoritma pengurutan dengan menukar elemen terus-menerus  
D. Algoritma pengurutan yang hanya cocok untuk data besar
- Question No. 5 :** Bagaimana kompleksitas waktu algoritma QuickSort dalam kasus terburuk?  
A.  $O(n)$   
B.  $O(n \log n)$   
C.  $O(n^2)$   
D.  $O(\log n)$

- File pages/admin/courses/add.php

```
<?php
require "../backend/connection.php";
require "../backend/Course.php";

session_start();
if (!isset($_SESSION["id"]) && !isset($_SESSION["username"]) && !isset($_SESSION["role"])) {
    header("Location: ../login.php");
}

$courses = mysqli_query($connect, "SELECT * FROM courses");

if (isset($_POST["insertCourse"])) {
    insert($connect);
}
?>
```

Analisis :

Program untuk insert data terdapat pada fungsi insert pada file backend/Course.php. Fungsi insert sendiri akan dijalankan ketika terdapat request POST dengan key "insertCourse" dari user.

Output :



- File pages/admin/courses/edit.php

```
<?php
require "../../backend/connection.php";
require "../../backend/Course.php";

session_start();
if (!isset($_SESSION["id"]) && !isset($_SESSION["username"]) && !isset($_SESSION["role"])) {
    header("Location: ../../login.php");
}

$course;
if (isset($_GET["id"])) {
    $course = show($connect, $_GET["id"]);
}

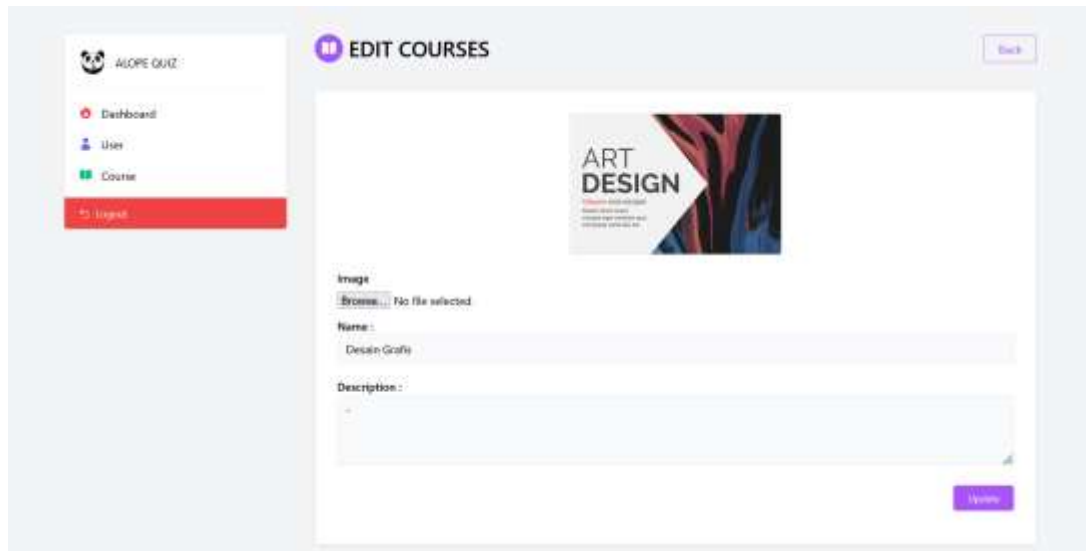
if (isset($_POST["updateCourse"])) {
    update($connect, $_GET["id"]);
}
?>
```

Analisis :

Sebelum melakukan edit data, tentunya kita harus Get data yang akan diedit terlebih dahulu. Hal itu saya handle dengan melakukan pengecekan jika terdapat request GET id maka akan membuat sebuah variabel course dengan nilai hasil return dari fungsi show. Fungsi ini adalah fungsi untuk get data by id dari tabel ccourses.

Untuk proses SQL Updatenya, akan dilakukan ketika terdapat request POST “updateCourse” dengan menjalankan fungsi update pada file backend/Course.php dengan mengirimkan value dari id pada request GET.

Output :



Berikut adalah file backendCourse.php-nya, yaitu sebuah file yang berisi fungsi-fungsi untuk CRUD dari tabel courses

```
<?php
echo '
    <script src="https://code.jquery.com/jquery-3.7.1.slim.min.js" integrity="sha256-
kmHvs0B+0pCW5GVHUNjv9rOmY0IvSIRcf7zGUDTDQM8="
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/sweetalert/2.1.2/sweetalert.min.js"></script>
';

function index($connect)
{
    $courses = mysqli_query($connect, "SELECT * FROM courses");

    return $courses;
}

function show($connect, $id)
{
    $course = mysqli_query($connect, "SELECT * FROM courses WHERE id=$id");

    return $course;
}
```

```

function insert($connect)
{
    extract($_POST);

    $uniqueFilename;
    if ($_FILES["file"]["name"]) {

        $targetDirectory = "C:/laragon/www/quiz/images/courses/";
        // $targetDirectory = "./images/courses/";
        $uniqueFilename = uniqid() . '_' . basename($_FILES["file"]["name"]);
        $targetFile = $targetDirectory . $uniqueFilename;

        move_uploaded_file($_FILES["file"]["tmp_name"], $targetFile);
    } else {
        $uniqueFilename = "course.svg";
    }

    mysqli_query($connect, "INSERT INTO courses VALUES(null, '$name', '$description', '$uniqueFilename')");

    echo '
    <script type="text/javascript">
        $(document).ready(function(){
            swal({
                title: "SUCCESS",
                text: "Course Created Successfully",
                icon: "success",
            }).then((value) => {
                window.location.href = "http://localhost/quiz/pages/admin/courses/index.php";
            });
        });
    </script>
    ';
}

function update($connect, $id)
{
    extract($_POST);

    mysqli_query($connect, "UPDATE courses SET name='$name', description='$description' WHERE id=$id");

    echo '
    <script src="https://code.jquery.com/jquery-3.7.1.slim.min.js" integrity="sha256-
    kmHvs0B+OpCW5GVHUNjv9r0mY0IvSIRcf7zGUDTDQM8="
    crossorigin="anonymous"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/sweetalert/2.1.2/sweetalert.min.js"></script>
    <script type="text/javascript">
        $(document).ready(function(){
            swal({
                title: "SUCCESS",
                text: "Course Edited Successfully",
                icon: "success",
            }).then((value) => {
                window.location.href = "http://localhost/quiz/pages/admin/courses/index.php";
            });
        });
    </script>
    ';
}

```

```

    });
</script>
';
}

function confirmDelete()
{
    echo '
<script type="text/javascript">
    $(document).ready(function(){
        swal({
            title: "CONFIRM?",
            text: "The course will be deleted permanently",
            icon: "warning",
            buttons: true,
            dangerMode: true,
        }).then((willDelete) => {
            if (willDelete) {
                var currentUrl = window.location.href;

                var newUrl = currentUrl + (currentUrl.indexOf("?") === -1 ? "?" : "&") +
"yesDeleted=1";

                window.location.href = newUrl;
            } else {
                swal({
                    title: "CANCELED",
                    text: "The course is not deleted",
                    icon: "error",
                }).then((value) => {
                    window.location.href = "http://localhost/quiz/pages/admin/courses/index.php";
                });
            }
        });
    });
</script>
';

    // mysqli_query($connect, "DELETE FROM courses WHERE id=$deletedId");
}

function delete($connect, $deletedId)
{
    mysqli_query($connect, "DELETE FROM courses WHERE id=$deletedId");

    echo '
<script type="text/javascript">
    $(document).ready(function(){
        swal({
            title: "SUCCESS",
            text: "Course Successfully Deleted",
            icon: "success",
        }).then((value) => {
            window.location.href = "http://localhost/quiz/pages/admin/courses/index.php";
        });
    });
</script>
';
}

```

- User pages (Halaman khusus User)

- File ./index.php

```
<?php
require "../backend/connection.php";

session_start();
if (!isset($_SESSION["id"]) && !isset($_SESSION["username"]) && !isset($_SESSION["role"])) {
    header("Location: ../pages/login.php");
}

$userCount = mysqli_num_rows(mysqli_query($connect, "SELECT * FROM accounts WHERE role='user'"));
$courseCount = mysqli_num_rows(mysqli_query($connect, "SELECT * FROM courses"));

?>
```

Output :



- File pages/user/courses.php (Bagian fungsi untuk menghitung total questions dalam 1 courses)

```
<?php

function countTotalCourse($id)
{
    global $connect;
    $getDataByCourseId = mysqli_query($connect, "SELECT * FROM questions WHERE course_id=$id");

    return mysqli_num_rows($getDataByCourseId);
}

?>
```

Analisis :

Membuat sebuah fungsi untuk untuk menghitung total questions dalam 1 course yaitu fungsi countTotalCourse dengan menerima 1 parameter. Statement nya yaitu untuk melakukan query data SELECT pada tabel question dengan (WHERE) course\_id nya sesuai dengan parameter id pada fungsinya. Lalu hasil dari query tersebut akan di masukkan kedalam sebuah variabel dan



akan dilakukan penghitungan row (data) dengan fungsi `mysqli_num_rows` sebagai return dari fungsinya.

- File `pages/user/quiz.php` (Bagian algoritma menerapkan pie chart)

```
<?php

$myAnswers = [];
// echo "myAnswer";
foreach ($courseAnswerCount as $myAnswer) {
    $myAnswers[] = $myAnswer["user_answer"];
}

$questionAnswers = [];
$courseQuestionCount = mysqli_query($connect, "SELECT * FROM questions WHERE
course_id=$courseId");
foreach ($courseQuestionCount as $questionAnswer) {
    $questionAnswers[] = $questionAnswer["answer"];
}

$score = 0;

foreach ($myAnswers as $key => $row) {
    if ($row == $questionAnswers[$key]) {
        $score++;
    }
}

?>

<script>
    window.onload = function() {

        var chart = new CanvasJS.Chart("chartContainer", {
            animationEnabled: true,
            title: {
                text: "Result Answer"
            },
            data: [{
                type: "pie",
                startAngle: 240,
                yValueFormatString: "##0",
                indexLabel: "{label} {y}",
                dataPoints: [{
                    y: <?= $score ?>,
                    label: "Correct"
                },
                {
                    y: <?= count($myAnswers) - $score ?>,
                    label: "Wrong"
                }
            ]
        }]
    });
```

```
chart.render();
```

```
}
```

### Analisis :

Untuk pembuatan sebuah chart nya, saya menggunakan plugin canvas.js. Untuk menghitung jumlah kesalahan-nya saya menggunakan cara aritmatika “jumlah total question” – “jumlah skor”. Karena untuk saat ini score masih berbentuk “total jawaban yang benar” bukan sebuah persentase atau nilai sehingga dapat dilakukan perhitungan seperti itu untuk mengambil nilai kesalahannya.

### Output :

