

BAB 4

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

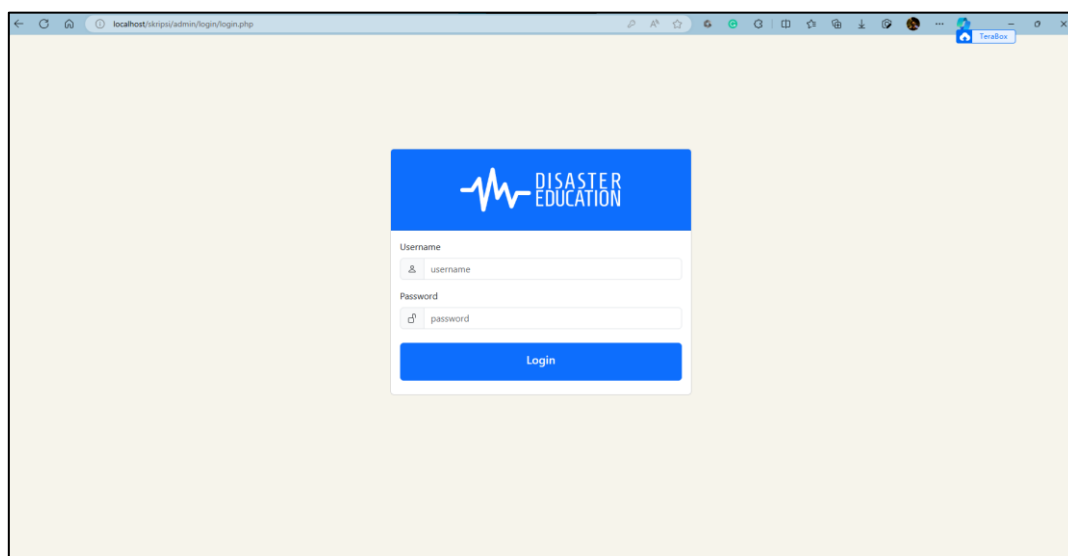
Dalam perancangan dan pengembangan sistem yang dilakukan dengan menggunakan komputer yang dilengkapi dengan sistem operasi Windows 10. Kemudian menggunakan *web browser Microsoft Edge* dan pengembangan kode program menggunakan bahasa pemrograman PHP dan menggunakan aplikasi *Visual Studio Code* yang dapat memudahkan dalam pengerjaannya. Selain itu, penulis menggunakan control panel PHP *web server XAMPP* untuk memudahkan pembuatan sistem *web server* lokal dalam membangun website.

4.2 Implementasi Aplikasi Website

Implementasi yang dilakukan adalah pengembangan sistem dengan menerapkan perancangan diagram dan desain interface pada bab sebelumnya. Pada bab ini yaitu mengimplementasikan perancangan struktur data kedalam bentuk tabel dan database, pembuatan kode program dan lain sebagainya. Berikut ini adalah menu yang tersedia pada aplikasi *website* Edukasi Kebencanaan, yaitu

4.2.1 Tampilan Halaman Admin Login

Halaman *login* atau halaman yang digunakan untuk meminta pengguna untuk memasukkan informasi yang diperlukan agar dapat mengakses akun atau layanan tertentu. Halaman ini meminta pengguna untuk memasukkan kombinasi username dan kata sandi yang sesuai dengan akun



Gambar 0.1 *Tampilan Halaman Admin Login*

Pada tampilan ini, fungsi yang digunakan yaitu pengecekan input *username* dan *password*. Serta fungsi kesesuaian antara hasil *inputan* dengan yang terdapat pada *database* agar bisa masuk ke dalam halaman *admin*.

Tabel 0.1 Source Code Tampilan Halaman Admin Login

```
<?php
// koneksi database dan session
include '../koneksi/koneksi.php';
include '../koneksi/session.php';

// fungsi login
if (isset($_POST['btn_login'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];

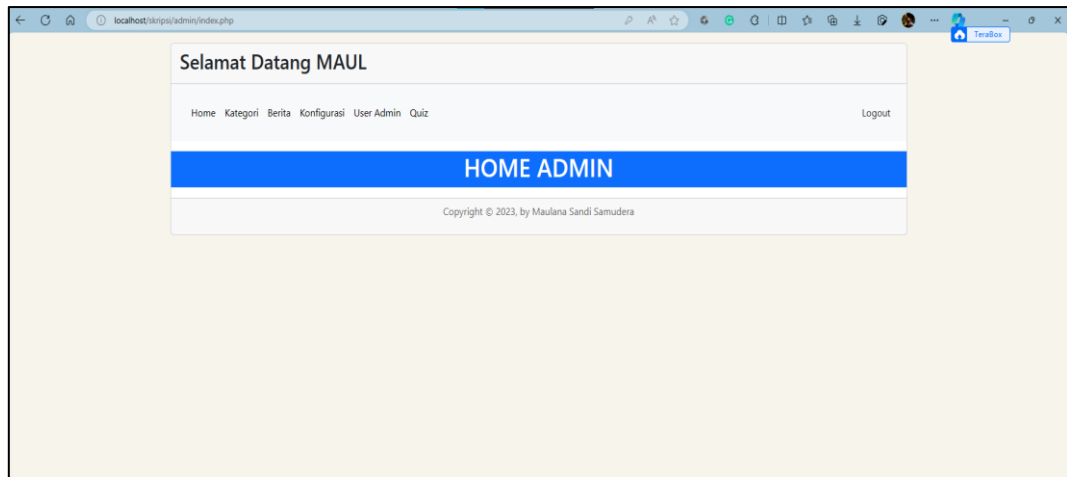
    $sql = mysqli_query($conn, "SELECT * FROM tb_admin WHERE username =
'$_POST[username]' AND password = '$_POST[password]'");

    $cek = mysqli_num_rows($sql);

    if (empty($username)) {
        header("Location: login.php?error=Username Is Required");
    } else if (empty($password)) {
        header("Location: login.php?error=Password Is Required");
    } else {
        if ($cek > 0) {
            $_SESSION['username'] = $_POST['username'];
            header("Location: ../index.php");
        } else {
            header("location: login.php");
        }
    }
}
?>
```

4.2.2 Tampilan Halaman Admin Home

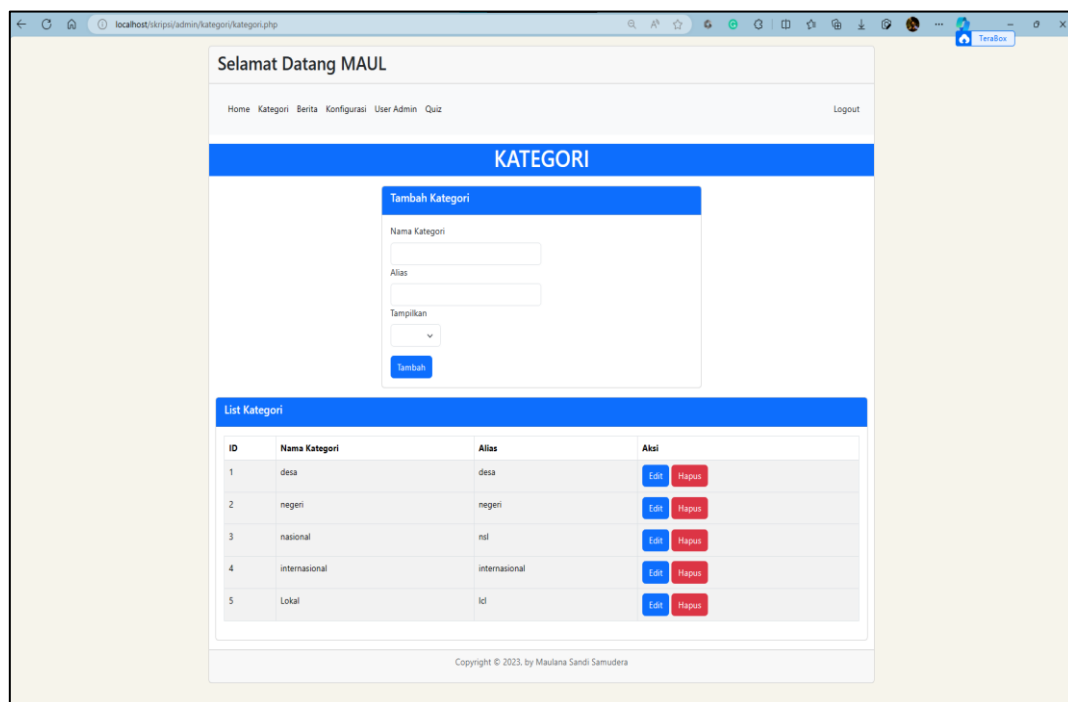
Halaman *admin home* ini adalah halaman awal dari bagian *admin*. Fungsi *login* sebelumnya akan mengarahkan pada halaman *admin*. Jika sudah masuk kedalam halaman ini maka login sebagai *admin* telah berhasil.



Gambar 0.2 *Tampilan Halaman Admin Home*

4.2.3 Tampilan Halaman Admin Menu Kategori

Halaman menu kategori ini merupakan halaman dengan fitur olah data *Create, Read, Update, Delete (CRUD)* terkait dengan jenis kategori yang akan ditampilkan dalam berita. Kategori ini yang akan ditampilkan pada bagian kategori berita maupun artikel.



Gambar 0.3 *Tampilan Halaman Admin Menu Kategori*

Berikut ini adalah *Source Code* dari sistem menu kategori. Terdapat beberapa fungsi seperti pengecekan inputan, tambah data, perbarui data, serta hapus data untuk macam kategori yang akan di olah datanya.

Tabel 0.2 *Source Code* Tampilan Halaman Admin Menu Kategori

```
<?php
include '../koneksi/koneksi.php';

if (isset($_POST['aksi'])) {

    // variable
    $id_kategori = $_POST['id_kategori'];
    $kategori = $_POST['kategori'];
    $alias = $_POST['alias'];
    $terbit = $_POST['terbit'];

    if ($_POST['aksi'] == 'add') {
        echo 'tambah';

        if (empty($kategori)) {
            header("Location: kategori.php?error=Kategori Is Required");
        } else if (empty($alias)) {
            header("Location: kategori.php?error=Alias Is Required");
        } else if (empty($terbit)) {
            header("Location: kategori.php?error=Terbit Is Required");
        } else {

            // jalannya fungsi
            $query = mysqli_query($conn, "INSERT INTO tb_kategori
(kategori, alias, terbit) VALUES ('$kategori','$alias','$terbit')");
            if ($query) {
                header("Location: kategori.php?scc=Data Berhasil
Disimpan");
            } else {
                echo "<script>
                    alert('Simpan data Gagal');
                    document.location='kategori.php';
                </script>";
            }
        }

    } else if ($_POST['aksi'] == 'ubah') {

        // jalannya fungsi
        $sql = "UPDATE tb_kategori SET kategori = '$kategori', alias =
'$alias', terbit = '$terbit' WHERE id_kategori = '$id_kategori'";
        $run = mysqli_query($conn, $sql);

        if ($run) {
            header("Location: kategori.php?edt=Data berhasil diubah");
        }
    }
} else if (isset($_GET['aksi'])) {
    if (($_GET['aksi'] == 'delete')) {
        echo 'hapus';

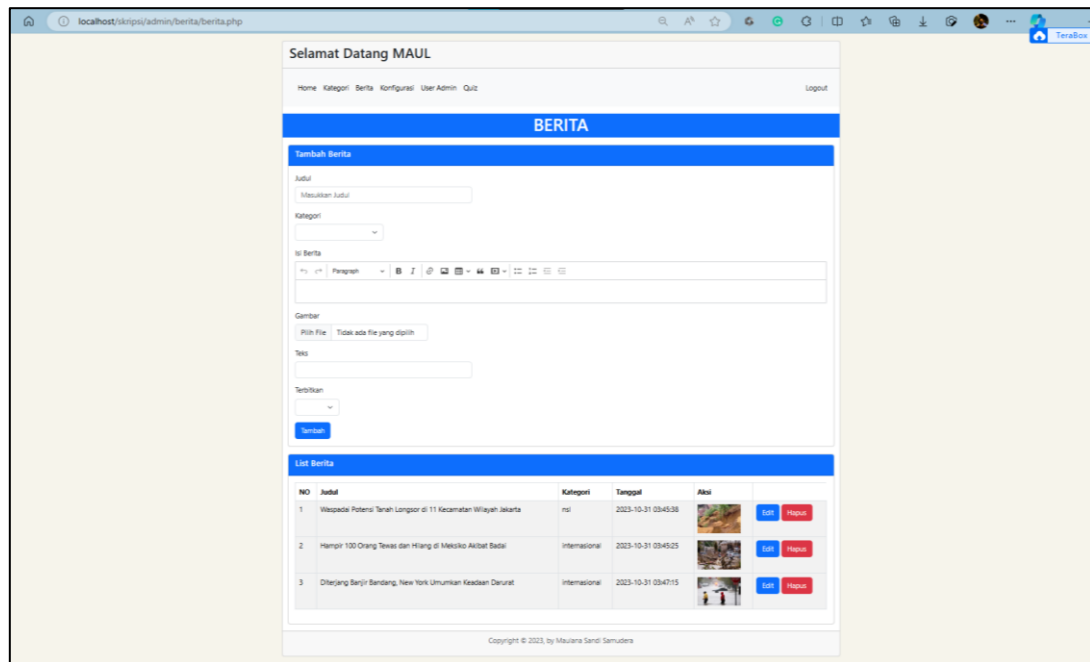
        $id_kategori = mysqli_real_escape_string($conn, $_GET['id']);
        $delete_kategori = mysqli_query($conn, "DELETE FROM tb_kategori
WHERE id_kategori = '$id_kategori'");

        header("Location: kategori.php?dlt=Data berhasil dihapus");
    }
}
```

?>

4.2.4 Tampilan Halaman Admin Menu Berita

Halaman menu ini merupakan halaman dengan fitur olah data *CRUD* tentang artikel ataupun berita yang nantinya akan ditampilkan. Nantinya akan ditampilkan pada menu berita dari sisi pengguna.



Gambar 0.4 Tampilan Halaman Admin Menu Berita

Berikut ini adalah *Source Code* dari sistem menu berita. Terdapat beberapa fungsi seperti pengecekan inputan, tambah data, perbarui data, serta hapus data untuk macam berita yang akan di olah datanya.

Tabel 0.3 *Source Code* Tampilan Halaman Admin Menu Berita

```
<?php
include "../koneksi/koneksi.php";
include "../koneksi/session.php";

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

    // variable
    $id_berita = $_POST["id_berita"];
    $judul = $_POST['judul'];
    $kategori = $_POST['kategori'];
    $isi = $_POST['isi'];
    $gambar = $_FILES['gambar']['name'];
    $teks = $_POST['teks'];
    $terbit = $_POST['terbit'];
    $author = $_SESSION['username'];

    date_default_timezone_set("Asia/Jakarta");
```

```

// echo "tanggal ". date("Y-m-d H:i:s");
$tanggal = date("Y-m-d H:i:s");

$dir = "../..../src/img/";
$tmpFile = $_FILES["gambar"]["tmp_name"];

if (($_POST['aksi'] == 'add')) {

    if (empty($judul)) {
        header("Location: berita.php?error=Judul Is Required");
    } else if (empty($kategori)) {
        header("Location: berita.php?error=Kategori Is Required");
    } else if (empty($isi)) {
        header("Location: berita.php?error=Isi Is Required");
    } else {

        // jalannya fungsi
        move_uploaded_file($tmpFile, $dir . $gambar);
        $query = mysqli_query($conn, "INSERT INTO tb_berita (judul,
isi, kategori, gambar, teks, tanggal, view, author, post_type, terbit )
VALUES ('$judul','$isi','$kategori',
'$gambar', '$teks', '$tanggal', '0', '$author','berita','$terbit')");
        if ($query) {
            header("Location: berita.php?scc=Data Berhasil Disimpan");
        } else {
            header("Location: berita.php?error=Data Gagal Disimpan");
        }
    }
} else if ($_POST['aksi'] == 'ubah') {
    // echo "edit";
    // die();
    $query_gam = "SELECT * FROM tb_berita WHERE id_berita =
'$id_berita'";
    $sql_gam = mysqli_query($conn, $query_gam);
    $result = mysqli_fetch_assoc($sql_gam);

    // var_dump($result);
    // die();

    if($_FILES['gambar']['name'] == '') {
        // echo 'kosoong';
        $gambar = $result['gambar'];
    } else {
        // echo 'ada';
        $gambar = $_FILES['gambar']['name'];
        unlink("../..../src/img/".$result['gambar']);
        move_uploaded_file($tmpFile, $dir . $gambar);
    }

    // jalannya fungsi
    $sql = "UPDATE tb_berita SET judul = '$judul', isi = '$isi',
kategori = '$kategori', gambar = '$gambar', teks = '$teks', terbit =
'$terbit' WHERE id_berita = '$id_berita'";
    $run = mysqli_query($conn, $sql);

    if ($run) {
        header("Location: berita.php?edt=Data berhasil diubah");
    }
}
} else if (isset($_GET['aksi']) ) {

```

```

if($_GET['aksi'] == 'delete') {
    // echo "delete";
    // die();

    $id_berita = mysqli_real_escape_string($conn, $_GET['id']);

    $query_gam = "SELECT * FROM tb_berita WHERE id_berita = '$id_berita'";
    $sql_gam = mysqli_query($conn, $query_gam);
    $result = mysqli_fetch_assoc($sql_gam);

    // var_dump($result);
    // die();

    unlink("../src/img/".$result['gambar']);

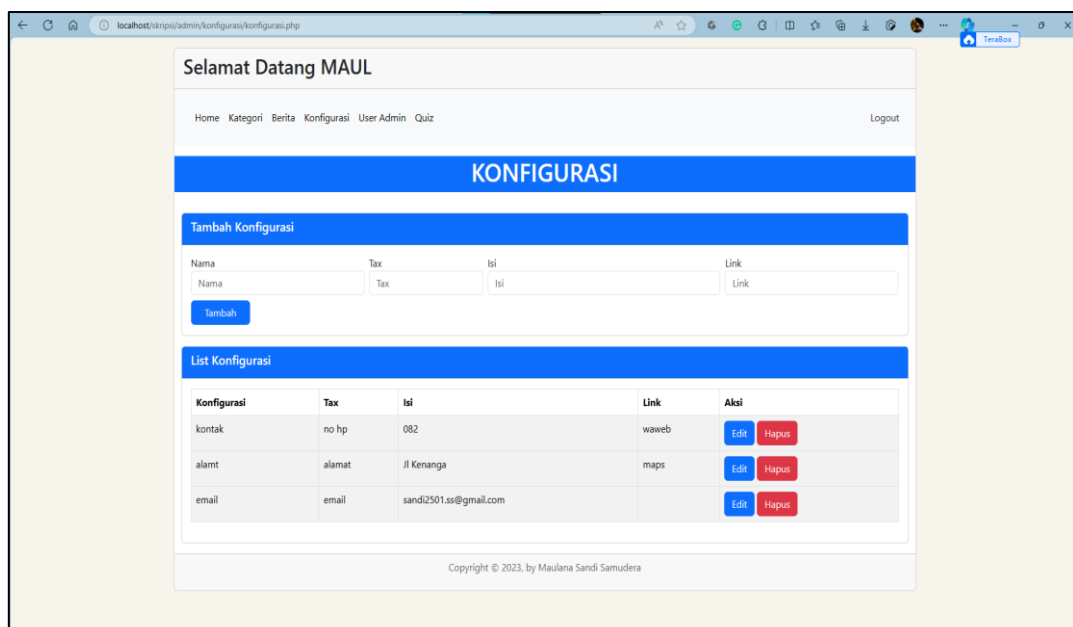
    $delete_admin = mysqli_query($conn, "DELETE FROM tb_berita WHERE id_berita = '$id_berita'");

    header("Location: berita.php?dlt=Data berhasil dihapus");
}
}
?>

```

4.2.5 Tampilan Halaman Admin Menu Konfigurasi

Halaman ini merupakan halaman dengan fitur olah data *CRUD* tentang informasi paten apa yang akan ditampilkan. Memuat informasi penting terkait dari lembaga. Data informasi ini akan masuk dalam *footer* sistem.



Gambar 0.5 Tampilan Halaman Admin Menu Konfigurasi

Berikut ini adalah *Source Code* dari sistem menu konfigurasi. Terdapat beberapa fungsi seperti pengecekan inputan, tambah data, perbarui data, serta hapus data untuk macam konfigurasi yang akan di olah datanya.

Tabel 0.4 *Source Code* Tampilan Halaman *Admin* Menu Konfigurasi

```
<?php
include("../koneksi/koneksi.php");

if (isset($_POST['aksi'])) {

    // variable
    $id_konfi = $_POST['id_konfi'];
    $nama_konfi = $_POST['nama_konfi'];
    $tax = $_POST['tax'];
    $isi_konfi = $_POST['isi_konfi'];
    $link = $_POST['link'];

    if ($_POST['aksi'] == 'add') {
        echo 'tambah';

        if (empty($nama_konfi)) {
            header("Location: konfigurasi.php?error=Konfigurasi Is
Required");
        } else if (empty($tax)) {
            header("Location: konfigurasi.php?error=Tax Is Required");
        } else if (empty($isi_konfi)) {
            header("Location: konfigurasi.php?error=Isi Is Required");
        } else {

            // jalannya fungsi
            $query = mysqli_query($conn, "INSERT INTO tb_konfigurasi
(nama_konfi, tax, isi_konfi, link, tipe) VALUES
('$nama_konfi', '$tax', '$isi_konfi', '$link', 'konfigurasi')");
            if ($query) {
                header("Location:konfigurasi.php?scc=Data Berhasil
Disimpan");
            } else {
                header("Location:konfigurasi.php?error=Data Gagal
Disimpan");
            }
        }
    } else if ($_POST['aksi'] == 'ubah') {
        // echo 'ubah';

        // jalannya fungsi
        $sql = "UPDATE tb_konfigurasi SET nama_konfi = '$nama_konfi', tax
= '$tax', isi_konfi = '$isi_konfi', link = '$link' WHERE id_konfi =
'$id_konfi'";
        $run = mysqli_query($conn, $sql);

        if ($run) {
            header("Location:konfigurasi.php?edt=Data berhasil diubah");
        }
    }
} else if ($_GET['aksi']) {
    if ($_GET['aksi'] == 'delete') {
        echo 'HAPUS';
    }
}
```



```

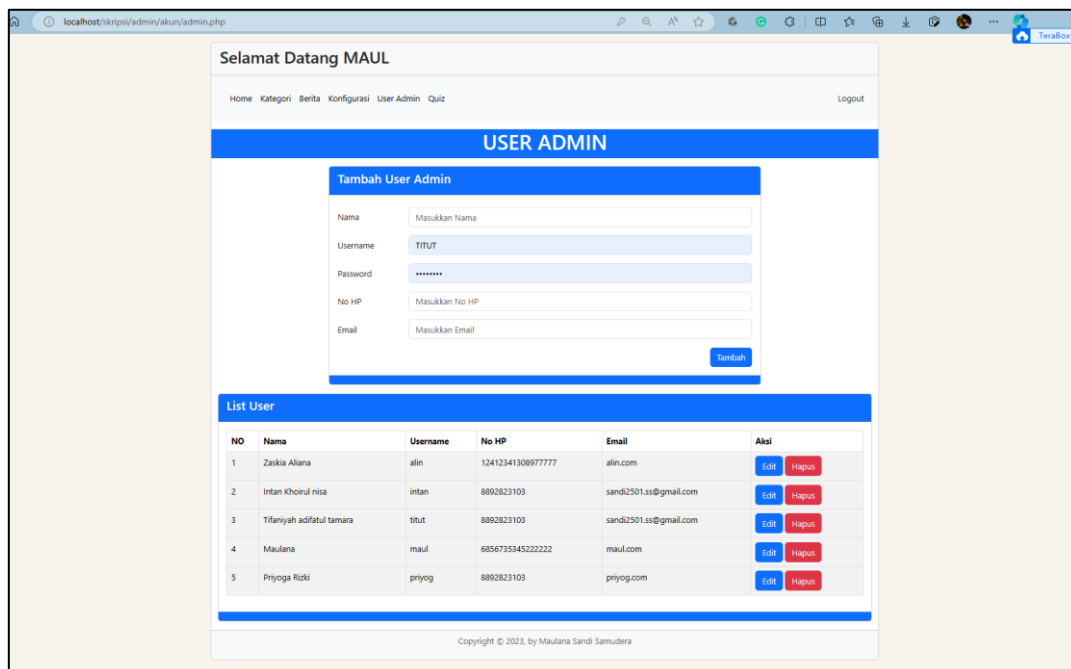
        $id_konfi = mysqli_real_escape_string($conn, $_GET['id']);
        $delete_konfi = mysqli_query($conn, "DELETE FROM tb_konfigurasi
WHERE id_konfi = '$id_konfi'");

        header("Location:konfigurasi.php?dlt=Data berhasil dihapus");
    }
}
?>

```

4.2.6 Tampilan Halaman Admin Menu User

Halaman ini merupakan halaman dengan fitur olah data *CRUD user admin*. Dalam pembuatan akun untuk sisi *admin* ini tidak menggunakan fitur register karena nantinya user yang bukan diperuntukkan sebagai admin bisa mendaftarkan dirinya sendiri sebagai *admin*. Maka dari itu, untuk tambah *user admin* akan lakukan oleh *admin* sendiri.



Gambar 0.6 Tampilan Halaman Admin Menu User

Berikut ini adalah *Source Code* dari sistem menu *user admin*. Terdapat beberapa fungsi seperti pengecekan inputan, tambah data, perbarui data, serta hapus data untuk akun *user admin* yang akan di olah datanya.

Tabel 0.5 *Source Code* Tampilan Halaman Admin Menu Konfigurasi

```

<?php
include("../koneksi/koneksi.php");

if (isset($_POST['aksi'])) {

    // variable
    $id_admin = $_POST['id_admin'];
    $nama_admin = $_POST['nama_admin'];

```

```

$username = $_POST['username'];
$password = $_POST['password'];
$no_hp_admin = $_POST['no_hp_admin'];
$email_admin = $_POST['email_admin'];

if (($_POST['aksi'] == 'add')) {

    if (empty($nama_admin)) {
        header("Location: admin.php?error>Nama Admin Is Required");
    } else if (empty($username)) {
        header("Location: admin.php?error=Username Is Required");
    } else if (empty($password)) {
        header("Location: admin.php?error=Password Is Required");
    } else if (empty($no_hp_admin)) {
        header("Location: admin.php?error=No Hp Is Required");
    } else if (empty($email_admin)) {
        header("Location: admin.php?error=Email Is Required");
    } else {

        // jalannya fungsi
        $query = mysqli_query($conn, "INSERT INTO tb_admin
(nama_admin, username, password, no_hp_admin, email_admin) VALUES
('$nama_admin', '$username', '$password', '$no_hp_admin', '$email_admin')");
        if ($query) {
            header("Location: admin.php?scc=Data Berhasil Disimpan");
        } else {
            header("Location: admin.php?error=Data Gagal Disimpan");
        }
    }
} else if (($_POST['aksi'] == 'ubah')) {

    // jalannya fungsi
    $sql = "UPDATE tb_admin SET nama_admin = '$nama_admin', username
= '$username', password = '$password', no_hp_admin = '$no_hp_admin',
email_admin = '$email_admin' WHERE id_admin = '$id_admin'";
    $run = mysqli_query($conn, $sql);

    if ($run) {
        header("Location: admin.php?edt=Data berhasil diubah");
    }
}

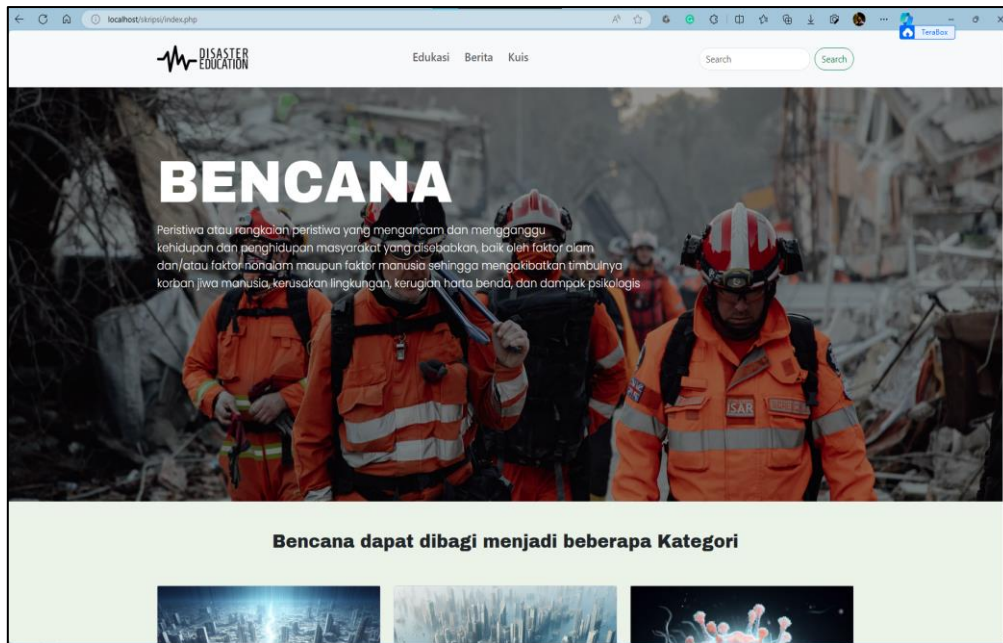
} else if (isset($_GET['aksi'])) {
    if ($_GET['aksi'] == 'delete') {
        $id_admin = mysqli_real_escape_string($conn, $_GET['id']);
        $delete_admin = mysqli_query($conn, "DELETE FROM tb_admin WHERE
id_admin = '$id_admin'");

        header("Location: admin.php?dlt=Data berhasil dihapus");
    }
}
?>

```

4.2.7 Tampilan Halaman *Home* Pengunjung

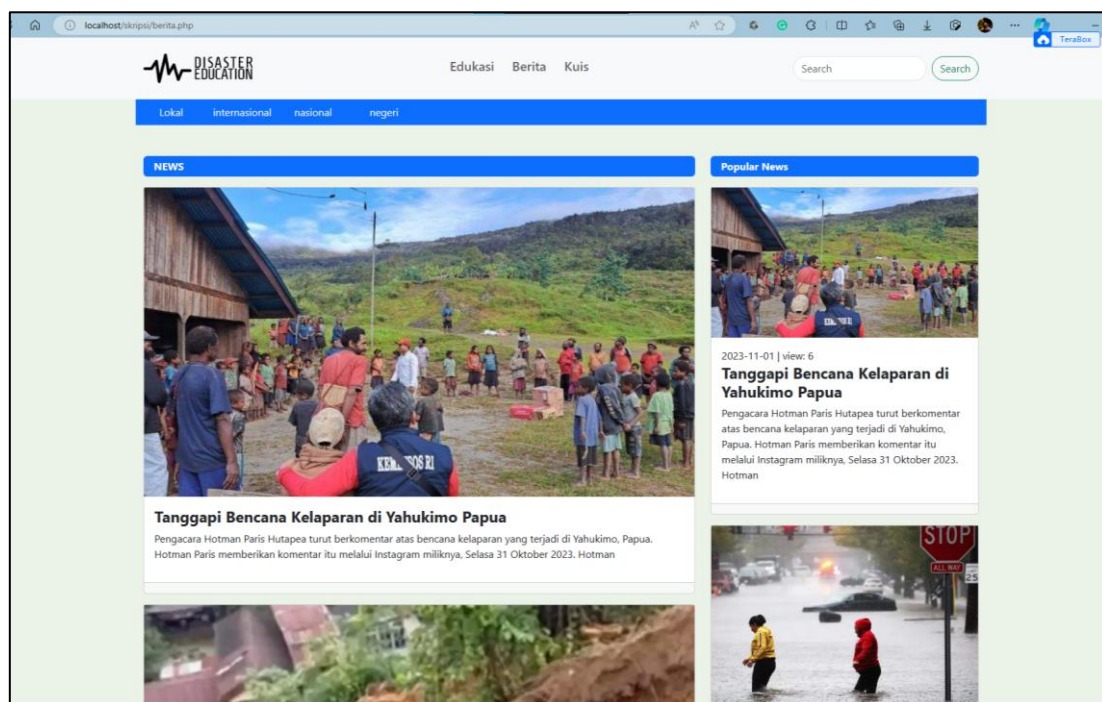
Halaman ini merupakan halaman awal pengunjung, dimana pada halaman ini berisikan dengan edukasi-edukasi terkait kebencanaan. Pada *navbar* terdapat beberapa fitur seperti pencarian dan untuk mengarahkan pada halaman lainnya.



Gambar 0.7 Tampilan Halaman Home Pengunjung

4.2.8 Tampilan Halaman Berita Pengunjung

Halaman ini merupakan halaman yang berisikan artikel maupun berita untuk pengunjung. Pada halaman ini akan ditampilkan seluruh artikel maupun berita *up to date* yang dibuat oleh *admin*. Pada sisi kiri adalah adalah berita dan artikel, dan sisi kanan berita maupun artikel terpopuler berdasarkan dari *view*-nya.



Gambar 0.8 Tampilan Halaman Berita Pengunjung

Berikut ini adalah *Source Code* dari sistem menu berita pengunjung. Disini fungsi yang dijelaskan yaitu untuk menampilkan sebuah data pada sisi kiri gambar sebelumnya, dan menampilkan data populer berdasarkan jumlah *view* pada sisi kanan gambar sebelumnya. Data yang ditampilkan adalah data berupa berita dan artikel pada menu *admin* berita.

Tabel 0.6 *Source Code* Tampilan Halaman Berita Pengunjung

```
<?php
include 'admin/koneksi/koneksi.php';
?>

<div class="container">
    <div class="row bg-primary py-2">
        <?php
            $menu = mysqli_query($conn, "SELECT * FROM tb_kategori WHERE
            terbit = '1' ORDER BY id_kategori ASC");
            while ($row = mysqli_fetch_array($menu)):
                // $id_kategori = $row['id_kategori'];
                $kategori = $row['kategori'];
                ?>
                <div class="col-sm-1 d-flex justify-content-center">
                    <a href="#" class="text-white" style="text-
                    decoration:none">
                        <?= $kategori ?>
                    </a>
                </div>
            <?php
            endwhile;
            ?>
        </div>
    </div>
    <div class="container mt-5">
        <div class="row">
            <div class="col-md-8">
                <div class="bg-primary text-white py-1 rounded">
                    <div class="ms-3">
                        <strong>NEWS</strong>
                    </div>
                </div>

                <?php
                    $sql = mysqli_query($conn, "SELECT * FROM tb_berita WHERE
                    terbit = '1' ORDER BY id_berita DESC LIMIT 0,10");
                    while ($row = mysqli_fetch_array($sql)):
                        extract($row);

                        // $result = mysqli_fetch_assoc($sql);

                        $id_berita = $row['id_berita'];
                        $judul = $row['judul'];
                        $kategori = $row['kategori'];
                        $isi = $row['isi'];
                        $gambar = $row['gambar'];
                        $teks = $row['teks'];
                        $terbit = $row['terbit'];
                        $author = $row['author'];
                        $tanggal = $row['tanggal'];
                        $view = $row['view'];
```

```

        ?>
        <a href="isi-berita.php?aksi=klik&id=<?= $id_berita
?>" style="text-decoration:none">
            <div class="card my-3">
                
                <div class="card-body">
                    <!-- judul -->
                    <h4><strong>
                        <?= $judul ?>
                    </strong></h4>
                    <p>
                        <?= substr(strip_tags($isi), 0, 200)
?>

                    </p>
                </div>
                <div class="card-footer">

            </div>
        </div>
    </a>

    <?php
    endwhile;
    ?>
</div>
<div class="col-md-4">
    <div class="bg-primary text-white py-1 rounded">
        <div class="ms-3">
            <strong>Popular News</strong>
        </div>
    </div>

    <?php
    $pop = mysqli_query($conn, "SELECT * FROM tb_berita
WHERE terbit = '1' ORDER BY view DESC LIMIT 0,10");

    while ($run = mysqli_fetch_array($pop)):
        extract($run);
    ?>
        <a href="isi-berita.php?aksi=klik&id=<?= $id_berita
?>" style="text-decoration:none">
            <div class="card my-3">
                
                <div class="card-body">
                    <!-- judul -->
                    <span><?= substr($tanggal, 0,10) ?> |
view: <?= $view ?></span>
                    <h4><strong>
                        <?= $judul ?>
                    </strong></h4>
                    <p>
                        <?= substr(strip_tags($isi), 0, 200)
?>

                    </p>
                </div>
                <div class="card-footer">

            </div>
        </div>
    </a>
    <?php

```

```

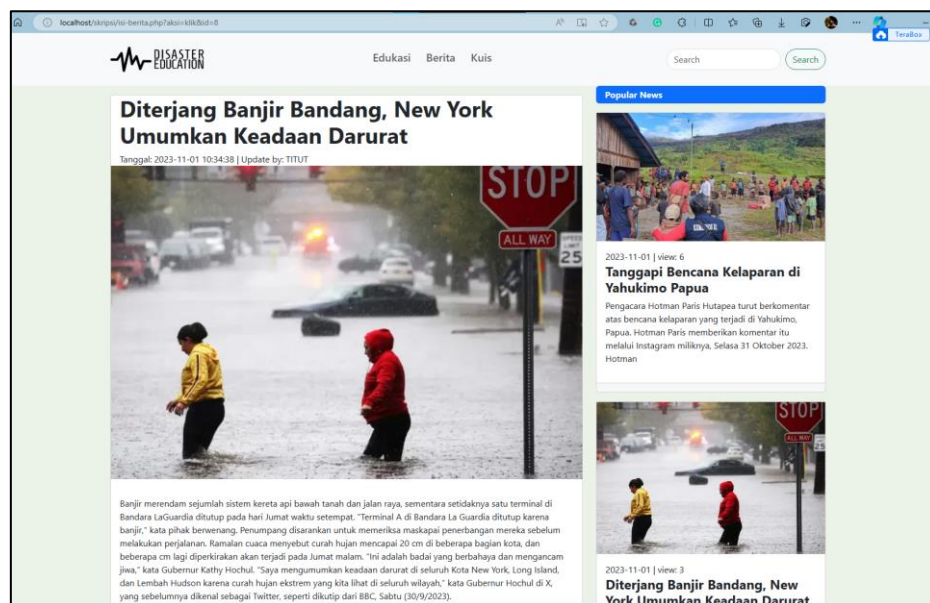
endwhile;

?>
</div>
</div>
</div>

```

4.2.9 Tampilan Halaman *Home* Berita

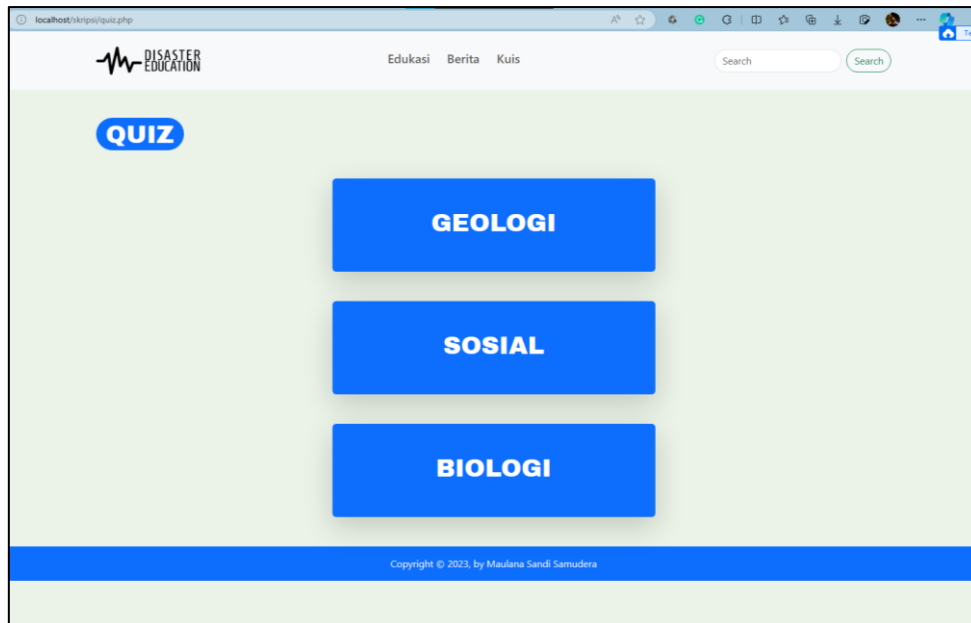
Halaman ini merupakan halaman *home* dari berita maupun artikel. Pada halaman ini, sebuah artikel maupun berita akan ditampilkan secara menyeluruh isinya secara lengkap dan pengunjung bisa memahaminya dengan lebih jelas.



Gambar 0.9 Tampilan Halaman *Home* Berita

4.2.10 Tampilan Halaman *Home* Kuis Pengunjung

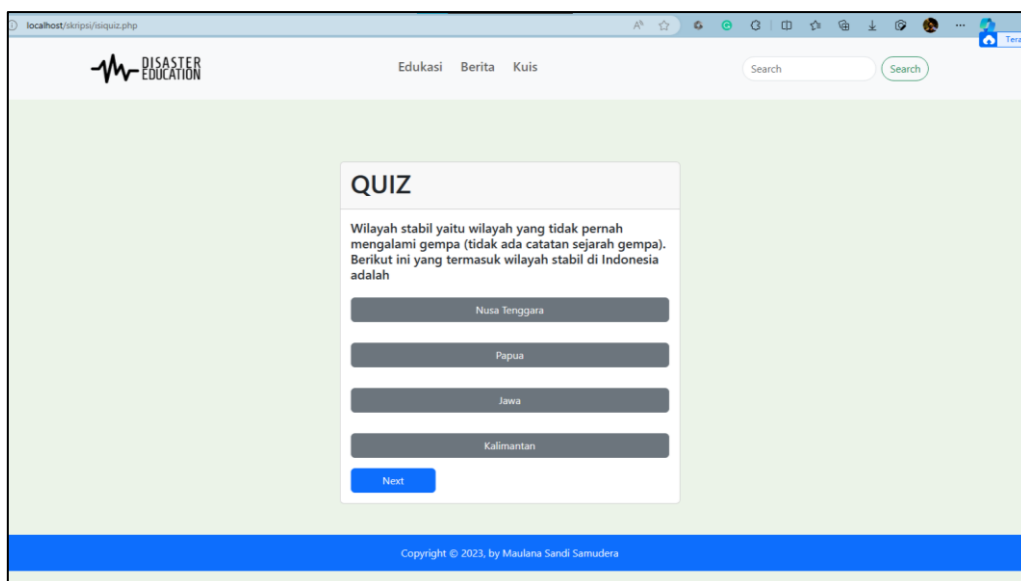
Halaman ini merupakan halaman *home* kuis untuk pengunjung. Pada halaman ini akan disediakan beberapa kategori yang nanti dapat dipilih salah satu. Dimana nanti pengunjung akan diarahkan pada kuis sesuai dengan kategori yang pengunjung pilih.



Gambar 0.10 *Tampilan Halaman Home Kuis Pengunjung*

4.2.11 Tampilan Halaman Kuis Pengunjung

Halaman ini merupakan halaman kuis yang telah dipilih oleh pengunjung. Kuis ini sesuai dengan kategori yang diawal sebelumnya dipilih. Kuis ini bisa untuk menguji pemahaman pengunjung terkait kebencanaan.



Gambar 0.11 *Tampilan Halaman Kuis Pengunjung*

4.3 Pengujian Sistem

Sistem yang sudah selesai dirancang, selanjutnya akan melalui tahapan pengujian sistem. Pada pengujian sistem, akan dilakukan dengan menilai dari segi fungsionalitas sistem dan dari sudut pandang pengguna. Disini pengujian akan menggunakan dua metode yaitu metode pengujian *Black Box Testing* dan pengujian UAT (*User Acceptance Testing*). Tujuan pengujian

ini adalah untuk mengetahui sistem yang telah didesain apakah sudah sesuai kebutuhan atau belum, dan untuk menghindari terjadinya kesalahan saat sistem digunakan.

4.3.1 Pengujian Black Box Testing

Sebelum melakukan pengujian dengan Blackbox Testing, terdapat skenario pengujian untuk menentukan langkah – langkah dalam melakukan pengujian sistem aplikasi *website* Edukasi Kebencanaan, diantaranya adalah sebagai berikut :















- a. *Load* halaman *Login Admin*
- b. *Button* “*Login*” *Admin*
- c. *Load* halaman *Dashboard home Admin*
- d. *Button* “*Kategori*” pada *navbar menu Admin*
- e. *Button* “*Tambah*” menu kategori
- f. *Button* “*Edit*” menu kategori
- g. *Button* “*Hapus*” menu kategori
- h. *Button* “*Berita*” pada *navbar menu Admin*
- i. *Button* “*Tambah*” menu berita
- j. *Button* “*Edit*” menu berita
- k. *Button* “*Hapus*” menu berita
- l. *Button* “*Konfigurasi*” pada *navbar menu Admin*
- m. *Button* “*Tambah*” menu konfigurasi
- n. *Button* “*User*” pada *navbar menu Admin*
- o. *Button* “*Tambah*” menu *user*
- p. *Button* “*Edit*” menu *user*
- q. *Button* “*Hapus*” menu *user*
- r. *Button* “*Log Out*” pada *navbar menu Admin*
- s. *Load* halaman *home Pengunjung*
- t. *Button* “*Edukasi*” pada *navbar menu Pengunjung*
- u. *Button* “*Search*” pada *navbar menu Pengunjung*
- v. *Button* “*Berita*” pada *navbar menu Pengunjung*
- w. *Card* *Berita* pada halaman *Pengunjung*
- x. *Button* “*Kuis*” pada *navbar menu Pengunjung*
- y. *Card* *Kuis* pada halaman *Pengunjung*
- z. *Button* jawaban pada halaman *kuis*
- aa. *Button* “*Next*” pada halaman *kuis*

Setelah menemukan skenario untuk pengujian, langkah selanjutnya adalah melakukan pengujian pada sistem menggunakan tabel *Black Box Testing*. Tujuan pengujian menggunakan metode ini adalah untuk memeriksa secara menyeluruh fungsi pada sistem yang telah dikembangkan apakah beroperasi dengan baik dan sesuai harapan pengguna.

Tabel 0.7 Pengujian *Black Box Testing*

No	Fitur	Hasil Yang Diharapkan	Hasil
1	<i>Load</i> halaman <i>Login Admin</i>	Dapat menampilkan halaman beranda <i>Login Admin</i>	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
2	<i>Button</i> “Login” <i>Admin</i>	Dapat memvalidasi data berdasarkan inputan <i>button</i> “Login”	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
3	<i>Load</i> halaman <i>Dasboard home Admin</i>	Dapat menampilkan halaman <i>dasboard home Admin</i>	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
4	<i>Button</i> “Kategori” pada navbar menu <i>Admin</i>	Dapat mengalihkan ke halaman kategori pada <i>admin</i>	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
5	<i>Button</i> “Tambah” menu kategori	Dapat menyimpan hasil <i>inputan</i> dari <i>form</i> kategori	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
6	<i>Button</i> “Edit” menu kategori	Dapat mengubah data <i>form</i> kategori	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
7	<i>Button</i> “Hapus” menu kategori	Dapat menghapus data <i>form</i> kategori	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
8	<i>Button</i> “Berita” pada navbar menu <i>Admin</i>	Dapat mengalihkan ke halaman berita pada <i>admin</i>	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>
9	<i>Button</i> “Tambah” menu berita	Dapat menyimpan hasil <i>inputan</i> dari <i>form</i> berita	Sukses <input checked="" type="checkbox"/> Gagal <input type="checkbox"/>

10	Button “Edit” menu berita	Dapat mengubah data <i>form</i> berita	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
11	Button “Hapus” menu berita	Dapat menghapus data <i>form</i> berita	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
12	Button “Konfigurasi” pada <i>navbar</i> menu Admin	Dapat mengalihkan ke halaman konfigurasi pada <i>admin</i>	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
13	Button “Tambah” menu konfigurasi	Dapat menyimpan hasil <i>inputan</i> dari <i>form</i> konfigurasi	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
14	Button “User” pada <i>navbar</i> menu Admin	Dapat mengalihkan ke halaman user pada <i>admin</i>	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
15	Button “Tambah” menu <i>user</i>	Dapat menyimpan hasil <i>inputan</i> dari <i>form user</i>	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
16	Button “Edit” menu <i>user</i>	Dapat mengubah data <i>form user</i>	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
17	Button “Hapus” menu <i>user</i>	Dapat menghapus data <i>form user</i>	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
18	Button “Log Out” pada <i>navbar</i> menu Admin	Dapat keluar dari sistem <i>admin</i> dan mengalihkan ke halaman awal <i>log in</i>	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
19	Load halaman <i>home</i> Pengunjung	Dapat menampilkan halaman beranda <i>home</i> pengunjung	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>
20	Button “Edukasi” pada <i>navbar</i> menu Pengunjung	Dapat menampilkan halaman beranda <i>home</i> pengunjung	Sukses <input checked="" type="checkbox"/>	Gagal <input type="checkbox"/>

21	<i>Button</i> “Search” pada <i>navbar</i> menu Pengunjung	Dapat menampilkan pencarian di halaman beranda <i>home</i> pengunjung	Sukses  Gagal 
22	<i>Button</i> “Berita” pada <i>navbar</i> menu Pengunjung	Dapat menampilkan halaman beranda berita pengunjung	Sukses  Gagal 
23	<i>Card</i> Berita pada halaman Pengunjung	Dapat menampilkan halaman berita pengunjung	Sukses  Gagal 
24	<i>Button</i> “Kuis” pada <i>navbar</i> menu Pengunjung	Dapat menampilkan halaman beranda kuis pengunjung	Sukses  Gagal 
25	<i>Card</i> Kuis pada halaman Pengunjung	Dapat menampilkan halaman kuis pengunjung	Sukses  Gagal 
26	<i>Button</i> jawaban pada halaman kuis	Dapat mengunci pilihan jawaban dari kuis	Sukses  Gagal 
27	<i>Button</i> “Next” pada halaman kuis	Dapat menampilkan soal kuis selanjutnya	Sukses  Gagal 

Pada skenario ini terdapat 27 poin pengujian dan dimana hasil dari pengujiannya tidak ada kesalahan dan semuanya dapat dijalankan dengan baik dan sukses.

4.3.2 Pengujian User Acceptance Test (UAT)

Setelah melakukan pengujian *Black Box Testing*, maka selanjutnya akan dilakukan pengujian dari sisi pengguna dengan metode UAT *testing*. Pengguna akan diberikan pernyataan yang sesuai dengan format pengujian UAT.

Tabel 0.8 Pengujian *User Acceptance Test* (UAT)

No	Pengujian	Langkah	Hasil Yang Diharapkan	Hasil		
				Diterima	Diterima dengan catatan	Ditolak

1	Login	Memasukkan <i>username</i> dan <i>password</i>	Login sukses dan masuk menu admin	✓		
2	Kategori	Memasukkan data pada form	Data berhasil disimpan dan masuk pada list data	✓		
3	Kategori	Merubah data pada form dengan “ <i>edit</i> ”	Data berhasil dirubah dan masuk pada list data	✓		
4	Kategori	Menghapus data pada list data dengan “hapus”	Data berhasil dihapus dan tidak ada di list data	✓		
5	Berita	Memasukkan data pada form	Data berhasil disimpan dan masuk pada list data	✓		
6	Berita	Merubah data pada form dengan “ <i>edit</i> ”	Data berhasil dirubah dan masuk pada list data	✓		
7	Berita	Menghapus data pada list data dengan “hapus”	Data berhasil dihapus dan tidak ada di list data	✓		
8	Konfigurasi	Memasukkan data pada form	Data berhasil disimpan dan masuk pada list data	✓		
9	User	Memasukkan data pada form	Data berhasil disimpan dan masuk pada list data	✓		

10	User	Merubah data pada form dengan “ <i>edit</i> ”	Data berhasil dirubah dan masuk pada list data	✓		
11	User	Menghapus data pada list data dengan “hapus”	Data berhasil dihapus dan tidak ada di list data	✓		
12	<i>Log Out</i>	Klik tombol “ <i>Log Out</i> ”	Dapat keluar dari sistem admin dan mengalihkan ke halaman awal <i>login</i>	✓		
13	Penunjung	Masuk ke menu utama pengunjung	Dapat menampilkan menu utama pengunjung	✓		
14	Menu	Pencarian di menu <i>navbar</i> pencarian	Dapat menampilkan kata yang dicari	✓		
15	Berita	Klik <i>card</i> berita pada halaman berita pada pengunjung	Dapat menampilkan berita secara keseluruhan	✓		
16	Kuis	Klik kuis pada halaman kuis pada pengunjung	Dapat menampilkan halaman kuis	✓		
17	Kuis	Mengerjakan kuis	<i>Score</i> ditampilkan	✓		

Berdasarkan pada pengujian UAT *testing* yang dilakukan, setiap fitur pada *website* Edukasi Kebencanaan telah sesuai dengan kebutuhan pengguna dan mampu diterima dan dipahami dengan baik.

4.4 Hasil Dan Pembahasan

4.4.1 Hasil

Aplikasi edukasi kebencanaan berhasil dibuat dengan menggunakan pengembangan metode *prototype*. Dalam pengembangannya, sudah mengikuti masing-masing alur di setiap proses pengerjaannya dari perencanaan, pengumpulan kebutuhan, perancangan hingga pengujian. Dalam pengujian aplikasi dengan metode *Black Box Testing* dan UAT, pada *Black Box Testing* terdapat 27 *point* dengan rata-rata mendapatkan hasil sukses seluruhnya begitu juga pada UAT dengan 17 *point* hasilnya telah diterima semua. Untuk pemahaman terhadap edukasi kebencanaan dapat dilakukan dengan fitur kuis pada sistem edukasi kebencanaan ini.

4.4.2 Pembahasan

Melihat dari penelitian sebelumnya bahwa, penggunaan metode *prototype* ini dalam pengembangan sistemnya menyesuaikan dengan keinginan pengguna dan memberikan gambaran aplikasi terlebih dahulu agar sesuai dengan keinginan pengguna. Dengan dilakukannya pengujian *Black Box Testing* dan UAT yang hasilnya sukses dan diterima membuktikan bahwa pengembangan sistem dengan metode *prototype* ini berhasil dengan melakukan pendekatan pada pengguna.