

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Database

4.1.1 Pembuatan Database

Berikut ini merupakan hasil pembuatan database baru yang telah dibuat



Gambar 4.1 Pembuatan Database

4.1.2 Pembuatan Tabel pada Database

A. Tabel Admin

Pada tabel admin terdapat empat atribut yaitu *id_admin*, *username*, *password*, dan *nama*. Atribut *id_admin* digunakan sebagai *primary key* dengan tipe data yaitu integer. Untuk atribut *username*, *password*, dan *nama* menggunakan tipe data *varchar*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_admin	int(10)			No	None		AUTO_INCREMENT
2	username	varchar(20)	latin1_swedish_ci		No	None		
3	password	varchar(20)	latin1_swedish_ci		No	None		
4	nama	varchar(30)	latin1_swedish_ci		No	None		

Gambar 4.2. Tabel Admin

B. Tabel User

Pada tabel *user* atribut NIK digunakan sebagai *primary key* karena termasuk dalam nomor unik. Selain atribut NIK terdapat atribut lainnya yaitu nama, alamat, telp, *email*, dan *password*. Semua atribut pada tabel ini menggunakan tipe data *varchar*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	NIK	varchar(20)	latin1_swedish_ci		No	None		
2	nama	varchar(50)	latin1_swedish_ci		No	None		
3	alamat	varchar(100)	latin1_swedish_ci		No	None		
4	telp	varchar(15)	latin1_swedish_ci		No	None		
5	email	varchar(50)	latin1_swedish_ci		No	None		
6	password	varchar(50)	latin1_swedish_ci		No	None		

Gambar 4.3. Tabel User

C. Tabel Paket

Pada tabel ini atribut *id_paket* merupakan *primary key* menggunakan *auto increment*. Selain itu terdapat atribut foto, harga, dan nama_paket yang menggunakan tipe data *varchar*. Dan terdapat satu atribut yang memiliki tipe data *text*, yaitu atribut *deskripsi*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_paket	int(10)			No	None		AUTO_INCREMENT
2	foto	varchar(300)	latin1_swedish_ci		No	None		
3	deskripsi	text	latin1_swedish_ci		No	None		
4	harga	varchar(30)	latin1_swedish_ci		No	None		
5	nama_paket	varchar(50)	latin1_swedish_ci		No	None		

Gambar 4.4. Tabel Paket

D. Tabel Kendaraan

Pada tabel kendaraan hanya berisi tiga atribut. Yaitu `id_kendaraan` yang berfungsi sebagai *primary key*, atribut `nama_kendaraan` yang memiliki tipe data *varchar*, dan atribut `orang` yang memiliki tipe data *integer*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<code>id_kendaraan</code>	<code>int(5)</code>			No	None		
2	<code>nama_kendaraan</code>	<code>varchar(20)</code>	<code>latin1_swedish_ci</code>		No	None		
3	<code>orang</code>	<code>int(11)</code>			No	None		

Gambar 4.5. Tabel Kendaraan

E. Tabel Pesanan

Pada tabel pesanan memiliki sembilan atribut yaitu `id_pesanan` yang berfungsi sebagai *primary key*, atribut `NIK` dan `id_paket` yang berfungsi sebagai *foreign key*. Dengan atribut pendukung lainnya yaitu `lokasi`, `telp`, `tgl_pesanan`, `tgl_jemput`, `total_pemesanan`, dan `status`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<code>id_pesanan</code>	<code>int(10)</code>			No	None		AUTO INCREMENT
2	<code>NIK</code>	<code>varchar(20)</code>	<code>latin1_swedish_ci</code>		No	None		
3	<code>id_paket</code>	<code>int(5)</code>			Yes	NULL		
4	<code>lokasi</code>	<code>varchar(10)</code>	<code>latin1_swedish_ci</code>		No	None		
5	<code>telp</code>	<code>varchar(15)</code>	<code>latin1_swedish_ci</code>		No	None		
6	<code>tgl_pesanan</code>	<code>date</code>			No	None		
7	<code>tgl_jemput</code>	<code>varchar(50)</code>	<code>latin1_swedish_ci</code>		No	None		
8	<code>total_pemesanan</code>	<code>int(20)</code>			No	None		
9	<code>status</code>	<code>varchar(30)</code>	<code>latin1_swedish_ci</code>		No	None		

Gambar 4.6. Tabel Pesanan

F. Detail Pesanan

Pada tabel ini terdapat empat atribut, yaitu `id_detail` yang digunakan sebagai *primary key*, atribut `id_pesanan` dan `id_kendaraan` yang digunakan

sebagai *foreign key*, dan atribut jumlah. Semua atribut pada tabel ini menggunakan tipe data *integer*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_detail	int(10)			No	None		AUTO_INCREMENT
2	id_pesanan	int(10)			No	None		
3	id_kendaraan	int(5)			No	None		
4	jumlah	int(20)			No	None		

Gambar 4.7. Tabel Detail Pesanan

G. Pembayaran

Pada tabel pembayaran terdapat atribut *id_pembayaran* yang digunakan sebagai *primary key*, dan atribut *id_pesanan* yang digunakan sebagai *foreign key*. Dan juga atribut pendukung lainnya yaitu atribut *nama_bank*, *no_rekening*, *jumlah_pembayaran*, *nama_pengirim*, *tanggal_pengiriman*, dan *konfirmasi*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id_pembayaran	int(10)			No	None		AUTO_INCREMENT
2	id_pesanan	int(10)			No	None		
3	nama_bank	varchar(30)	latin1_swedish_ci		No	None		
4	no_rekening	varchar(30)	latin1_swedish_ci		Yes	None		
5	jumlah_pembayaran	int(10)			No	None		
6	nama_pengirim	varchar(30)	latin1_swedish_ci		No	None		
7	tanggal_pengiriman	varchar(30)	latin1_swedish_ci		No	None		
8	konfirmasi	varchar(20)	latin1_swedish_ci		Yes	None		

Gambar 4.8. Tabel Pembayaran

4.1.3 Pembuatan Relasi Tabel

Pada relasi tabel ini setiap tabel yang memiliki tipe *foreign key* akan selalu terhubung dengan atribut yang menggunakan tipe *primary key* dengan tipe data dan juga batas maksimal kartakter yang sama.



Gambar 4.9. Relasi Antar Tabel

4.2 Implementasi Sistem

4.2.1 Koneksi Database

Pada *source code* koneksi ini program akan melakukan koneksi ke database dengan nama *db_trimurti*.

```

1 </php
2 session_start();
3 $koneksi = new mysqli("localhost","root","","db_trimurti");
4 ?>

```

Gambar 4.10. Koneksi Database

4.2.2 Halaman Login

Pada tampilan *login* ini *user* dapat memasukkan *email* dan *password* mereka yang digunakan untuk masuk kedalam akun mereka. Selain itu untuk melakukan pemesanan wisata pengguna juga harus melakukan *login* terlebih dahulu agar dapat memesan.



Gambar 4.11. Halaman Login.

4.2.3 Halaman Utama

Pada halaman ini sistem akan menampilkan tiga paket terbaru yang dimasukkan oleh admin, pada halaman ini *user* juga dapat memesan paket dan juga melihat deskripsi paket yang tersedia. *User* juga disediakan menu pilihan untuk melihat semua paket yang tersedia.



Gambar 4.12. Halaman Utama

4.2.4 Popup Deskripsi Paket Wisata

Saat *user* menekan tombol deskripsi maka *popup* ini akan muncul. *Popup* ini berisi deskripsi secara lengkap mengenai paket wisata



Gambar 4.13. Popup Deskripsi Paket Wisata

4.2.5 Halaman Paket Wisata

Pada halaman ini *user* dapat mendapatkan informasi mengenai semua paket yang tersedia, *user* dapat melihat deskripsi paket dan *user* juga dapat memesan paket wisata melalui halaman ini.



Gambar 4.14. Halaman Paket Wisata

4.2.6 Halaman Pemesanan Paket Wisata

Pada halaman ini *user* dapat melakukan pemesanan paket wisata sesuai dengan paket yang dipilih melalui halaman utama atau halaman daftar paket wisata. Pada halaman ini *user* dapat memilih kendaraan yang akan digunakan serta memasukkan data berupa tanggal penjemputan, lokasi pejemputan, dan juga nomor telepon yang dapat dihubungi. Pada bagian bawah terdapat jumlah harga yang harus dibayar untuk mendapatkan pelayanan jasa wisata.



Gambar 4.15. Halaman Pemesanan Paket Wisata

4.2.7 Halaman Akun

Pada halaman ini *user* dapat melihat riwayat pemesanan paket wisata yang pernah dipesan sebelumnya. *User* juga dapat melihat dan mengubah data diri mereka pada halaman ini.

4.2.9 Halaman Pembayaran

Pada halaman ini *user* dapat mengetahui kekurangan pembayaran dan juga riwayat pembayaran mereka. Pada bagian riwayat pembayaran terdapat kolom status yang berguna untuk menunjukkan pembayaran mereka telah disetujui atau belum diproses. Setiap pelanggan juga dapat melakukan konfirmasi pembayaran melalui halaman ini jika mereka sudah melakukan pembayaran.



Gambar 4.18. Halaman Pembayaran

4.3 *White Box Testing*

Pengujian *white box* bertujuan untuk mengamati proses dari kode program dan menganalisis jalannya kode program apakah ada kesalahan atau tidak. Hasil pengujian adalah sebagai berikut:

4.3.1 Pengujian *White Box* Halaman Login

Berikut ini adalah source code untuk melakukan pengujian *white box testing* untuk halaman login.

```

<?php
if (isset($_POST["login"]))
{
    $email = $_POST["email"];
    $password = $_POST["password"];
    $ambil = $koneksi->query("SELECT * FROM user WHERE email='".$email.'" AND password='".$password.'");
    $count = $ambil->num_rows;
    if ($count==1)
    {
        $sakan = $ambil->fetch_assoc();
        $_SESSION["pelanggan"] = $sakan;
        echo "<script>location='index.php';</script>";
    }
    else
    {
        echo "<script>alert('Email atau password anda salah!');</script>";
        echo "<script>location='login.php';</script>";
    }
}
</?php
if(isset($_POST["daftar"]))
{
    echo "<script>location='daftar.php';</script>";
}
}

```

Gambar 4.19. Script Halaman Login

Pada *source code* berikut jika user menekan tombol *login* maka sistem akan mengambil data *email* dan *password* yang diketik oleh *user*. Dari data tersebut sistem akan mencari data yang sesuai dengan *email* dan *password* yang diketik oleh pelanggan. Jika terdapat satu data yang cocok maka sistem akan menyimpan data user kedalam *session* dan halaman akan dialihkan ke halaman utama.

Namun jika tidak ada data yang sesuai sistem akan menampilkan notifikasi bahwa *email* dan *password* yang dimasukkan salah, dan halaman tetap berada di halaman *login*.

4.3.2 Pengujian *White Box* Halaman Pemesanan

Berikut ini adalah *source code* untuk melakukan pengujian *white box testing* untuk halaman pemesanan paket wisata.

```

<?php
if (isset($_POST['save']))
{
    $NIK = $_SESSION['pelanggan']['NIK'];
    $id = $_SESSION['idpaket'];
    $tgl_pesan = date("Y-m-d");
    $koneksi->query("INSERT INTO pemesanan ($NIK,$id_paket,$tgl_pesan,$tgl_jemput,
    total_pemesanan,status,pesan_nama) VALUES ('$NIK','$id','$_POST[lokasi]',
    '$_POST[telepon]','$tgl_pesan','$_POST[tgl_jemput]','$total12','dibayar Rp. 0','tidak')");
    $id_pesan = $koneksi->insert_id;
    foreach ($SESSION['kendaraan'] as $id_kendaraan => $jumlah)
    {
        $koneksi->query("INSERT INTO detail_pesanan ($id_pesan,$id_kendaraan,$jumlah)
        VALUES ('$id_pesan','$id_kendaraan','$jumlah')");
    }
    echo "<script>alert('Data tersimpan')</script>";
    echo "<script>location='pembayaran.php?id=$id_pesan'</script>";
}
?>
</div>
</div>

```

Gambar 4.20. Script Halaman Pemesanan

Pada *source code* diatas jika user menekan tombol simpan maka sistem akan mengambil data NIK pelanggan, id paket, dan tanggal pemesanan lalu disimpan dalam variabel yang berbeda. Setelah itu sistem akan menyimpan semua data tersebut dan data pemesanan yang pelanggan masukkan kedalam tabel pemesanan yang berada di *database*. Lalu sistem akan mengambil id pesanan yang terakhir, id tersebut digunakan untuk menyimpan data pesanan kendaraan kedalam tabel detail_kendaraan yang berada di *database*. Lalu yang terakhir, sistem akan menampilkan halaman pembayaran.

4.3.3 Pengujian *White Box* Halaman Pembayaran

Berikut ini adalah *source code* untuk melakukan pengujian *white box testing* untuk halaman pembayaran.

```

<label>Nama Rekening Pengirim:</label>
<input type="text" class="form-control" name="nama">
</div>
<div class="form-group">
<label>Tanggal Pengiriman:</label>
<input type="text" class="form-control" name="tanggal">
</div>

<button class="btn btn-primary" name="save">Konfirmasi</button>
</form>
<?php
if (isset($_POST['nama']))
{
    $operasi=>query("INSERT INTO pembayaran (id_pemohon,nama_bank,
    no_rekening, jumlah_pembayaran,nama_pengirim,tanggal_pengiriman,konfirmasi )
    VALUES ('".$_POST[bank].','.$_POST[rekening].','.$_POST[asaya].','.$_POST[nama].',
    '.$_POST[tanggal].','Belum diproses')");
}

echo "<script>alert('Konfirmasi diterima, harap tunggu Akun segera kami proses!');</script>";
echo "<script>location='admin.php';</script>";
}

```

Gambar 4.21. *Script* Halaman Pembayaran

Pada *source code* diatas jika user menekan tombol simpan, maka sistem akan memasukkan data pembayaran yang telah diketik oleh user kedalam tabel pembayaran yang berada di *database*. Setelah itu sistem akan memunculkan notifikasi bahwa konfirmasi pembayaran telah diterima dan halaman akan diarahkan ke halaman akun.




4.4 *Black Box* Testing

Pengujian menggunakan *black box* bertujuan untuk mengamati hasil dari jalannya sistem apakah telah sesuai dengan rancangan yang telah dibuat. Hasil dari pengujian adalah sebagai berikut :

4.4.1 Pengujian *Black Box* Halaman Login

Berikut ini merupakan pengujian *Black Box* Pada halaman login. Terdapat tiga pengujian yang akan dilakukan pada halaman login. Berikut ini merupakan hasil pengujian halaman login.

Tabel 4.1. Pengujian *Black Box* Halaman Login

No	Skenario Pengujian	Hasil yang Harapkan	Hasil Pengujian	Keterangan
1	Mengkosongan <i>email</i> dan <i>password</i> lalu menekan tombol <i>login</i>	Sistem akan menolak dan tetap berada pada halaman <i>login</i>	 <p>Sistem menolak dan tetap berada pada halaman <i>login</i></p>	<i>Valid</i>
2	Mengisi <i>email</i> dan <i>password</i> dengan benar	Sistem akan menerima <i>email</i> dan <i>password</i> lalu halaman akan diarahkan ke halaman utama	 <p>Sistem menerima <i>email</i> dan <i>password</i> lalu halaman mengarah ke halaman utama</p>	<i>valid</i>
3	Mengisi <i>email</i> dan <i>password</i> yang salah	Sistem akan menolak dan tetap berada pada halaman <i>login</i>	 <p>Sistem akan menolak dan tetap berada pada halaman <i>login</i></p>	<i>valid</i>

4.4.2 Pengujian *Black Box* Halaman Pemesanan

Berikut ini merupakan pengujian *Black Box* Pada halaman pemesanan. Terdapat dua pengujian yang akan dilakukan pada halaman login. Berikut ini merupakan hasil pengujian halaman pemesanan.

Tabel 4.2 Pengujian *Black Box* Halaman Pemesanan

No	Skenario Pengujian	Hasil yang Harapkan	Hasil Pengujian	Keterangan
1	Mengisi semua <i>text input</i> yang tersedia	Sistem menerima pesanan dan halaman akan diarahkan ke halaman pembayaran	 <p>Sistem menerima pesanan dan menuju halaman pembayaran</p>	<i>valid</i>
2	Mengkosongkan salah satu <i>text input</i> yang ada	Sistem menolak pesanan dan halaman tetap berada di halaman pemesanan	 <p>Sistem menolak pesanan dan halaman tetap berada di halaman pemesanan</p>	<i>valid</i>

4.4.3 Pengujian *Black Box* Halaman Pembayaran

Berikut ini merupakan pengujian *Black Box* Pada halaman pembayaran. Terdapat dua pengujian yang akan dilakukan pada halaman login. Berikut ini merupakan hasil pengujian halaman pembayaran.

Tabel 4.3. Pengujian *Black Box* Halaman Pembayaran

No	Skenario Pengujian	Hasil yang Harapkan	Hasil Pengujian	Keterangan
1	Mengisi semua <i>text input</i> yang tersedia	Sistem menerima bukti pembayaran dan data pembayaran masuk kedalam database	 <p>Sistem menerima bukti pembayaran dan data pembayaran masuk kedalam database.</p>	<i>valid</i>
2	Mengkosongkan salah satu <i>text input</i> yang ada	Sistem menolak bukti pembayaran	 <p>Sistem menolak bukti pembayaran</p>	<i>valid</i>